Politecnico di Torino

Politecnico di Torino

Corso di Laurea Magistrale in Ingegneria Informatica

A.a. 2023/2024

Sessione di laurea Aprile 2024

Domus: creazione di un'interfaccia web reattiva per Musictogo

Relatore:	Tutor Aziendale:	Candidato:
Luigi De Russis	Denis Bellotti	Antonio Colelli

Ringraziamenti

Non penso di essere in grado di trovare le parole adatte a ringraziare mio padre Luigi, mia madre Adriana e mio fratello Jacopo Lorenzo per avermi sempre supportato in questo lungo e faticoso percorso. Voi dite che quello che ho ottenuto è solo merito mio, ma consentitemi di correggervi: senza voi tre al mio fianco, non sarei mai riuscito a raggiungere gli obiettivi che ho raggiunto, e non sarei mai riuscito ad essere la persona che sono. Vi devo tutto. Lasciate, semplicemente, che vi dica **GRAZIE**, dal profondo del mio cuore.

Voglio ringraziare anche tutte quelle persone che sono state al mio fianco in tutti questi anni: dai miei zii, ai miei cugini, ai miei più cari amici, alle persone che adesso vegliano su di me dal Cielo, come le mie nonne Cosima e Oronza, mio nonno Antonio, mio zio Giorgio e il mio amico Diego.

A mio fratello e ai miei genitori

Indice

Elenco delle tabelle

Elenco delle figure

Capitolo 1

Introduzione

La nostra società vive in un'epoca dominata dalla tecnologia digitale. È sempre più comune che le persone svolgano, in parte o totalmente, le proprie attività giornaliere con l'ausilio di dispositivi quali computer e smartphone.

In questo contesto, con l'obiettivo di ampliare il proprio raggio d'azione verso un pubblico sempre più ampio, la maggior parte dei business locali sta iniziando ad avere la necessità di espandere e/o trasferire (parzialmente o addirittura completamente), le proprie attività su internet.

Dal negozio di vendita al dettaglio, al ristorante, alla bottega artigiana, una pratica comune per tutti questi business è l'utilizzo di mezzi e piattaforme terze che permettano all'attività commerciale di avere una presenza attiva in rete. Si pensi a piattaforme come JustEat [1], o eBay [2], che fungono da tramite tra attività commerciali e clienti, velocizzando e semplificando le procedure di scambio di beni e servizi.

Un settore fino a poco tempo fa per lo più estraneo alle novità e alle opportunità messe a disposizione dalla rivoluzione digitale è quello delle strutture di affitto di sale prova e/o studi di registrazione musicali. Per prenotare una sessione musicale era necessario recarsi in sede o chiamare telefonicamente il gestore per accordarsi su una data/ora in cui fosse possibile riservare una sala per suonare. È qui che entra in gioco Musictogo [3].

Musictogo è la prima piattaforma italiana di prenotazione nel settore musicale che permette a strutture come le precedentemente citate sale prove di offrire ai propri clienti un servizio login-less di prenotazione di fasce orarie in totale autonomia attraverso il proprio portale di Booking [4], senza necessità di chiamate o interazioni umane.

1.1 Obiettivo della tesi

Musictogo fonda il proprio modello di business sul portale di Booking messo a disposizione dei musicisti per accedere ai servizi che questa start-up offre. Il portale di Booking attualmente online ha accompagnato Musictogo, nel suo processo di crescita, fin dalla nascita della società. Non appena fondata, l'azienda aveva l'urgente necessità di sviluppare e rilasciare velocemente in produzione un portale funzionante che permettesse ai potenziali utenti di effettuare le prime prenotazioni. La comprensibile fretta con cui il portale è stato sviluppato ha fatto sì che aspetti come l'usabilità e la responsività dell'interfaccia fossero messe in secondo piano, a favore di una rapida realizzazione del portale stesso. L'obiettivo della presente tesi è il rinnovamento del portale Booking di Musictogo attraverso lo studio, lo sviluppo e la validazione di una nuova interfaccia utente basata su tecnologie all'avanguardia, che tenga in considerazione anche gli aspetti messi in secondo piano dall'attuale portale.

Il lavoro di tesi ha avuto inizio studiando e, successivamente, analizzando, mediante ben definite tecniche di valutazione euristica, il portale attuale, al fine di ottenere un resoconto scritto con dettagli circa le problematiche emerse. Questo processo di analisi ha rappresentato il punto di partenza per la progettazione di una nuova interfaccia che non cada vittima delle stesse problematiche riscontrate nella versione corrente.

Musictogo ha reso disponibili alcuni mockup, sviluppati verso la fine del 2021, nati con lo scopo di aggiornare, già allora, l'interfaccia grafica del portale. Tale aggiornamento non venne però mai implementato. I mockup sono stati analizzati per individuarne le carenze e le difformità rispetto al portale attuale. Per carenze, si intendono elementi, quali componenti specifici, funzionalità o pagine, non illustrati all'interno dei mockup ma presenti all'interno del portale attuale. Per difformità si intendono, invece, quelle funzionalità che, nel corso del tempo, sono state aggiunte, rivalutate o, addirittura, rimosse rispetto a quando furono sviluppati i mockup, e quegli elementi che, nonostante siano rappresentati nei mockup, non sono attualmente disponibili a back-end nel formato desiderato o non sono disponibili affatto. I mockup sono stati, poi, oggetto di un'ulteriore analisi, questa volta con le medesime tecniche utilizzate per analizzare l'interfaccia dell'attuale portale, al fine di far emergere eventuali violazioni euristiche. A partire dai risultati delle due analisi, e tenendo in considerazione anche i risultati dell'analisi del portale attuale, è stata svolta un'opera di aggiornamento dei mockup con lo scopo di revisionare gli elementi presenti e crearne, talvolta, di nuovi, per compensare le carenze dei prototipi.

A seguito dell'aggiornamento dei mockup, è iniziata una fase di ricerca incentrata sulle tecnologie da adottare per lo sviluppo della nuova interfaccia. Le principali tecnologie impiegate includono la libreria React [5] e il framework Next.js [6].

Conclusa la fase di sviluppo, l'interfaccia prodotta è stata valutata mediante una metodologia sperimentale, condotta con la collaborazione di potenziali utenti, denominata "Test di usabilità"; sono state effettuate cinque sessioni di test di usabilità (tre in ambiente Mobile, due in Desktop), al fine di ottenere una validazione del lavoro svolto. I risultati delle cinque sessioni di test di usabilità hanno validato la bontà del lavoro svolto per rinnovare l'interfaccia e hanno offerto diversi spunti per potenziali miglioramenti; tuttavia hanno anche evidenziato l'urgente necessità di alcune modifiche. L'interfaccia è stata conseguentemente revisionata, con l'obiettivo di risolvere le maggiori criticità, e successivamente valutata nella sua versione aggiornata, attraverso tre ulteriori sessioni di test (due in ambiente Mobile, una in Desktop) che hanno confermato un miglioramento della qualità delle interazioni a seguito della revisione.

1.2 Struttura della tesi

Il documento di tesi si scompone nei seguenti capitoli, rappresentativi delle diverse fasi precedentemente illustrate:

- Il Capitolo *2* offre al lettore un'introduzione dettagliata all'attuale portale di Booking esplorandone le pagine e i principali Task Flow degli utenti del servizio.

- Il Capitolo *3* tratta l'analisi dell'interfaccia utente dell'attuale portale Booking di Musictogo. Vengono dapprima descritti i principi teorici alla base della metodologia di analisi utilizzata, ovvero la Valutazione euristica, per poi svolgere l'analisi del portale al fine di individuare le violazioni di usabilità presenti nell'attuale interfaccia. Questo processo rappresenta il punto di partenza per la progettazione di una nuova interfaccia che non cada vittima delle stesse problematiche riscontrate nella versione corrente del portale.

- Il Capitolo *4* descrive il lavoro svolto per revisionare i mockup, sviluppati nel 2021, che Musictogo ha messo a disposizione come punto di partenza del lavoro di progettazione. Se ne studiano i limiti, cercando poi di sviluppare degli aggiornamenti adeguati.

- Il Capitolo *5* offre una panoramica sulle tecnologie disponibili per lo sviluppo di interfacce web, per poi entrare nello specifico delle tecnologie che si è deciso di impiegare per lo sviluppo di questo progetto.

- Il Capitolo *6* descrive la fase di scrittura del codice del progetto. La descrizione di quanto fatto nella pratica viene, saltuariamente, accompagnata da una descrizione più teorica dei principi sui quali si basano le scelte fatte.

- il Capitolo *7* tratta la validazione della nuova interfaccia utente sviluppata a partire dai mockup revisionati. Le sezioni di cui questo capitolo si compone, descrivono, rispettivamente, i principi teorici alla base della metodologia di valutazione utilizzata, l'applicazione di tale metodologia per validare il lavoro svolto e le modifiche, anch'esse validate, apportate al progetto a seguito dei risultati ottenuti dalla prima validazione.
- il Capitolo *8* trae le conclusioni del lavoro di tesi, descrivendone poi le limitazioni e i possibili lavori futuri.

Capitolo 2

Il portale Booking di Musictogo

Questo capitolo fornisce al lettore un'esauriente descrizione del portale Booking di Musictogo, evidenziandone le pagine (Sezione 2.1) e i principali Task Flow degli utenti del servizio (Sezione 2.2).

Si rende necessario, ai fini di una più facile comprensione da parte del lettore, un breve preambolo per distinguere le tre tipologie di strutture attualmente disponibili all'interno del portale:

- *Sale prove ad affitto orario.* Sono strutture che consentono di affittare le proprie salette musicali per periodi di tempo nell'ordine delle ore. Le salette possono, occasionalmente, entrare in uno stato denominato *Promo* [1]: in tal caso, l'utente che prenoterà tali salette avrà diritto ad una percentuale di sconto (tipicamente il 30%) sul prezzo di listino dell'affitto orario.

- *Studi di registrazione.* Sono strutture che offrono i propri servizi sotto forma di pacchetti acquistabili ed utilizzabili successivamente. L'utente interessato a tali servizi deve acquistare un pacchetto e successivamente concordare con la struttura quando utilizzarlo.

- *Sale prove ad affitto mensile.* Strutture simili alle già citate sale prove ad affitto orario, dalle quali però differiscono per la durata dell'affitto delle salette, in questo caso mensile.

Particolarmente rilevante è il fatto che, attualmente, la totalità delle sale prove ad affitto mensile, così come una parte delle sale prove ad affitto orario, utilizzino

[1] La dicitura *Promo* sta ad indicare uno stato di temporanea promozione della saletta.

il portale solo come vetrina, non permettendo dunque all'utente di effettuare prenotazioni.

Ciascuna sottosezione del presente capitolo illustrerà immagini delle schermate del portale. Le immagini verranno mostrate al solo scopo illustrativo, per renderle note al lettore. Non verranno analizzati, in questo capitolo, se non superficialmente, eventuali problemi, che saranno invece oggetto di discussione nel Capitolo 3.

2.1 Il portale

Il portale è stato sviluppato dal team di Musictogo utilizzando la libreria React [5] e la libreria di strumenti Bootstrap [7]. Le successive sottosezioni esploreranno le quattro principali schermate che compongono il portale, raggiungibili tramite la Sidebar: Sale prove, Assistenza, Contatti e Cancellazioni.

2.1.1 Sale prove

L'homepage del portale di Booking, illustrata in Figura 2.1, consiste nella schermata "Sale Prove". Questa pagina elenca, mediante una lista di card, le strutture disponibili su Musictogo, e ne indica le posizioni geografiche mediante una mappa.

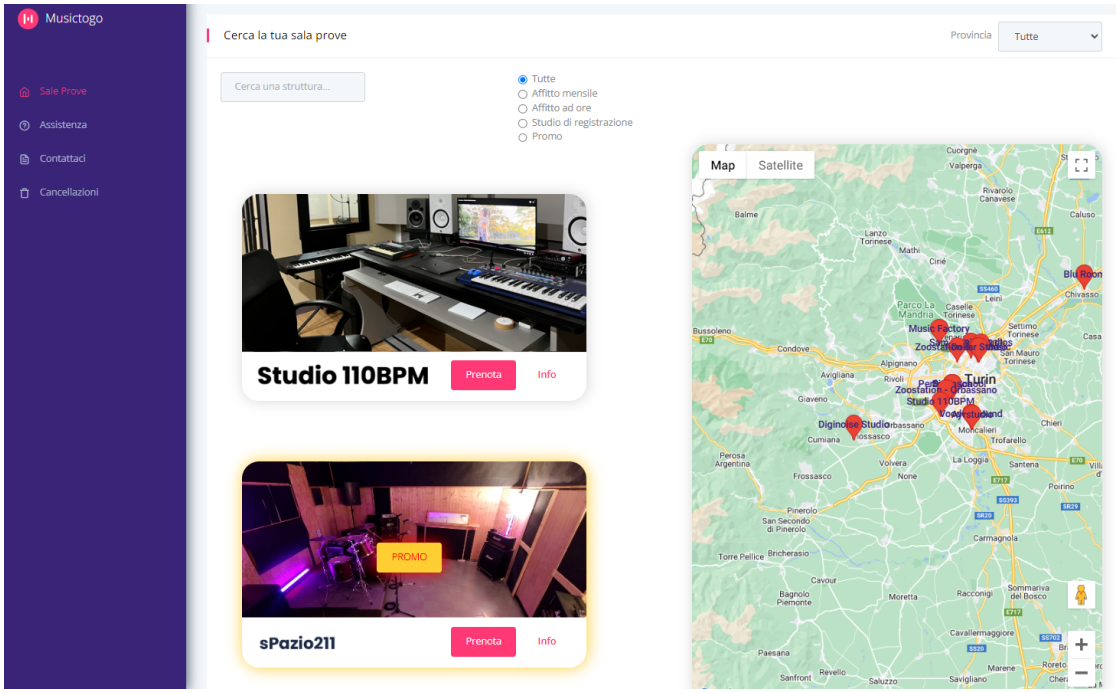

Figura 2.1: Homepage, portale attuale

La lista delle strutture è filtrabile facendo uso degli strumenti posti nella parte superiore della pagina:

- una *barra di ricerca* che filtra la lista sulla base di ciò che riceve in input. L'input desiderato sono informazioni come il nome della struttura o il suo indirizzo;
- una serie di *radio button* che consentono di filtrare la lista di strutture in base al tipo di struttura ("Affitto mensile", "Affitto ad ore" e "Studio di registrazione") e per promozioni in corso ("Promo");
- un *menu a tendina* che consente di filtrare la lista per provincia.

Ogni elemento della lista corrisponde ad una singola struttura. L'elemento grafico utilizzato per rappresentare le strutture è la Card. Ogni card include un pulsante "Info" che, se cliccato, fa apparire un Modale contenente informazioni sulla struttura (si veda Figura 2.2) e un pulsante "Prenota" il cui click apre la pagina della struttura (che verrà esaminata in seguito, durante lo studio dei Task Flow, nella Sezione 2.2). Le strutture in stato "Promo" presentano, come visibile in Figura 2.1, un testo "Promo" su sfondo giallo al centro della card ed un'ombreggiatura esterna gialla.

Figura 2.2: Modale con informazioni struttura, Homepage, portale attuale

La lista delle strutture è suddivisa, come illustrato in Figura 2.3, in varie pagine, ciascuna delle quali visualizza un sub-set della lista (circa tre strutture per pagina). Ad oggi sono presenti dodici pagine.

Figura 2.3: Paginazione della lista di strutture, Homepage, portale attuale

La mappa consente, invece, di localizzare le strutture tramite dei marker posizionati nelle coordinate geografiche delle strutture stesse. Ciascun marker è contrassegnato da un titolo (il testo in blu sul marker), identificatore della struttura che sta localizzando. I confini iniziali della mappa mostrano tutte le strutture in provincia di Torino.

Mappa e lista non sono tra di loro interconnesse: un'interazione con la mappa non provoca un aggiornamento della lista delle strutture e, viceversa, generalmente, filtrando la lista non si hanno riscontri sulla mappa. L'unico componente di filtraggio legato in qualche modo alla mappa è il menu a tendina delle provincie: cambiando provincia, la mappa si aggiorna spostando i suoi confini su tale provincia, mostrando di conseguenza i marker delle strutture in quella zona. Una volta rimosso il filtro però, la mappa non reagisce, rimanendo centrata sull'ultima provincia selezionata.

Sale prove - versione Mobile

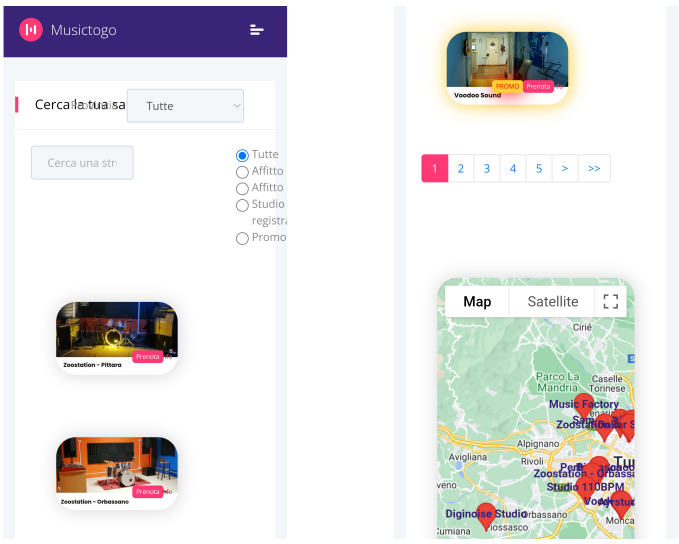

Figura 2.4: Homepage, portale attuale Mobile

2.1.2 Assistenza

La pagina "Assistenza", illustrata in Figura 2.5, ospita una serie di Q&A.

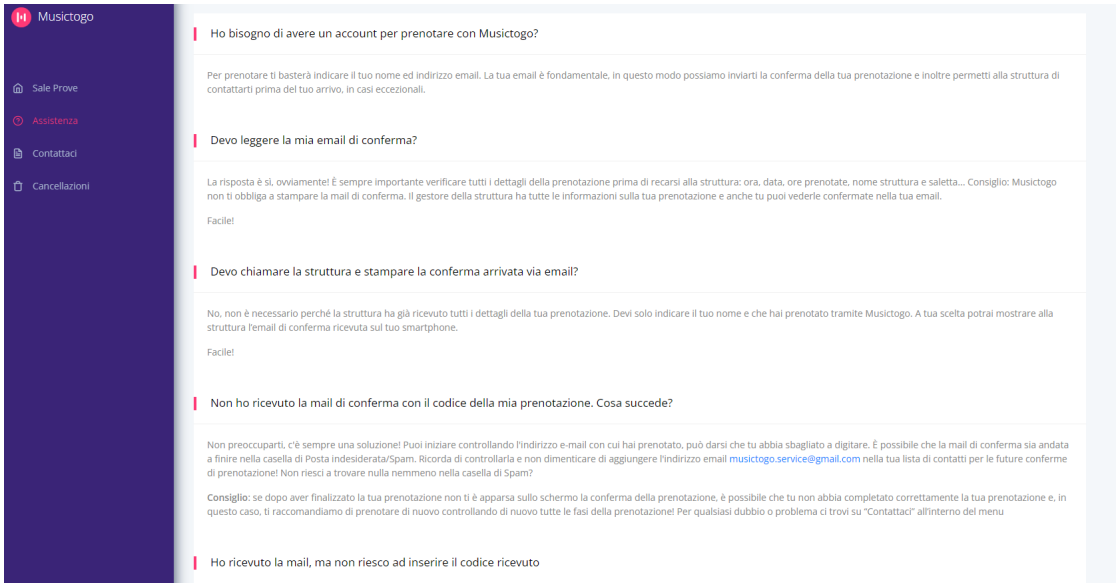

Figura 2.5: Pagina Assistenza, portale attuale

Assistenza - versione Mobile

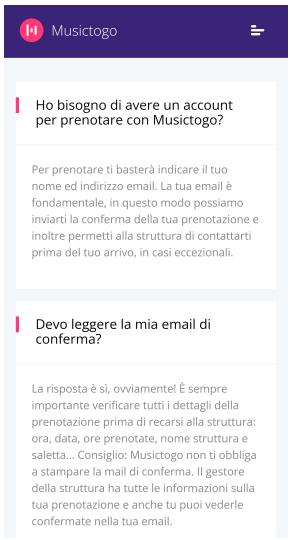

Figura 2.6: Pagina Assistenza, portale attuale Mobile

2.1.3 Contatti

La pagina "Contatti", illustrata in Figura 2.7, presenta informazioni utili per contattare lo staff di Musictogo, ed un collegamento all'indirizzo https://musictogo.it.

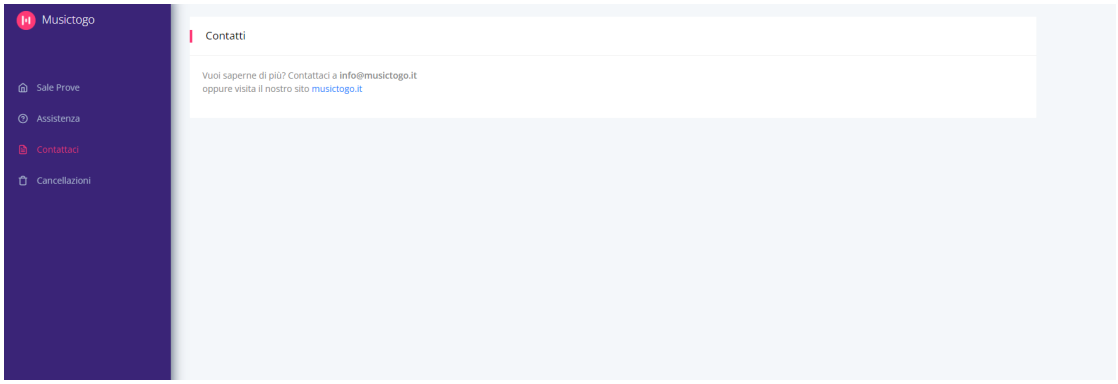

Figura 2.7: Pagina Contatti, portale attuale

Contatti - versione Mobile

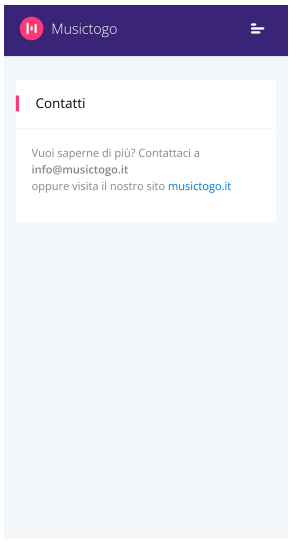

Figura 2.8: Pagina Contatti, portale attuale Mobile

2.1.4 Cancellazioni

Offrendo un servizio login-less, Musictogo ha dovuto sviluppare, tramite la pagina "Cancellazioni", illustrata in Figura 2.9, una procedura che permettesse agli utenti di annullare prenotazioni effettuate in precedenza senza la necessità di un accesso al sistema tramite procedure di Login. La pagina è composta da un form tramite il quale l'utente, dopo aver inserito i dati richiesti (la email utilizzata in fase di prenotazione ed il codice della prenotazione), potrà cancellare la prenotazione tramite la pressione del pulsante "Cancella".

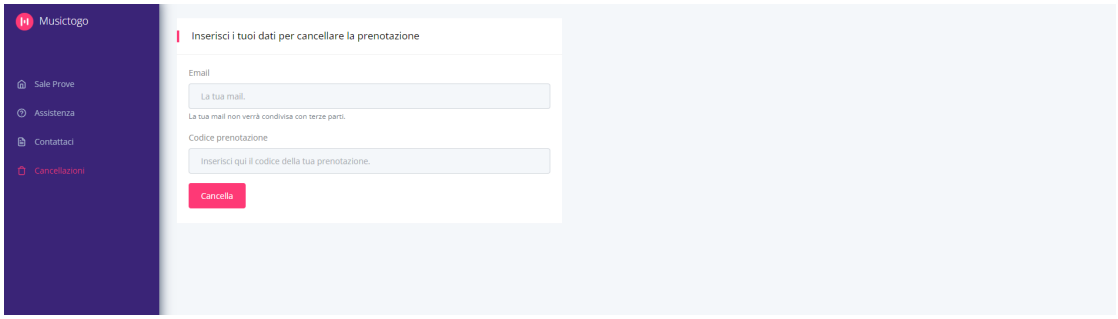

Figura 2.9: Pagina Cancellazioni, portale attuale

Cancellazioni - versione Mobile

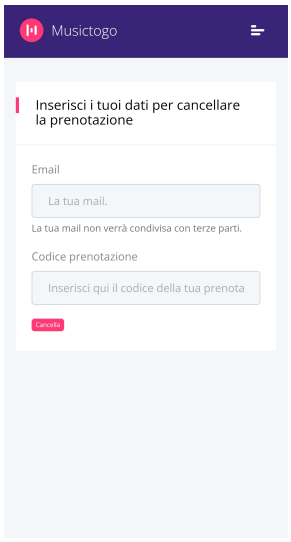

Figura 2.10: Pagina Cancellazioni, portale attuale Mobile

2.2 I principali Task Flow

Il Task Flow [8] è una forma specifica di User Flow che rappresenta il percorso che gli utenti devono seguire per completare un determinato task. Questa sezione esamina i principali Task Flow del portale Booking di Musictogo analizzando, nell'ordine, l'affitto orario di salette, l'acquisto di pacchetti e la cancellazione di prenotazioni. Nella descrizione dei tre Task Flow si considererà sempre, come punto di partenza, la Homepage (Figura 2.1).

Prima di procedere con la descrizione dei Task Flow è necessario fare una premessa: le singole card della schermata iniziale (Figura 2.1) distinguono le strutture per nome, ma non forniscono informazioni sulla tipologia di queste ultime (le tre tipologie di struttura sono state ampiamente descritte a inizio capitolo). Questo fa sì che, generalmente, un utente non sia consapevole di ciò che clicca quando preme il pulsante "Prenota" su una card. Tale problematica verrà approfondita nel Capitolo 3. In questo capitolo si assume, pertanto, che, nei primi due Task Flow, la prima azione dell'utente sia stata filtrare la lista tramite, rispettivamente, i radio button "Affitto ad ore" e "Studio di registrazione" per ottenere solo strutture facenti parte di tali categorie.

2.2.1 Affitto orario di salette

Partendo dal contesto precedentemente descritto nell'introduzione della presente sezione, l'utente intenzionato a prenotare una saletta di una sala prove ad affitto orario dovrà scegliere una struttura dalla lista e cliccare il pulsante "Prenota" sulla card di riferimento (figura 2.1).

Sorge, purtroppo, un ulteriore problema, sempre legato alla poca chiarezza delle Card: l'utente, ancora una volta, non è generalmente consapevole di ciò che sta per aprire in quanto le card non distinguono le sale prove ad affitto orario che permettono di effettuare prenotazioni delle proprie salette tramite il portale, da quelle che non lo permettono e che usano il portale solo come vetrina. A seguito del click su "Prenota", sono possibili due scenari:

1. l'apertura della pagina di una sala prove ad affitto orario avente salette prenotabili online (Figura 2.11);

2. l'apertura della pagina di una sala prove ad affitto orario avente salette **non** prenotabili online (Figura 2.12).

Anche questo problema verrà approfondito nel Capitolo 3.

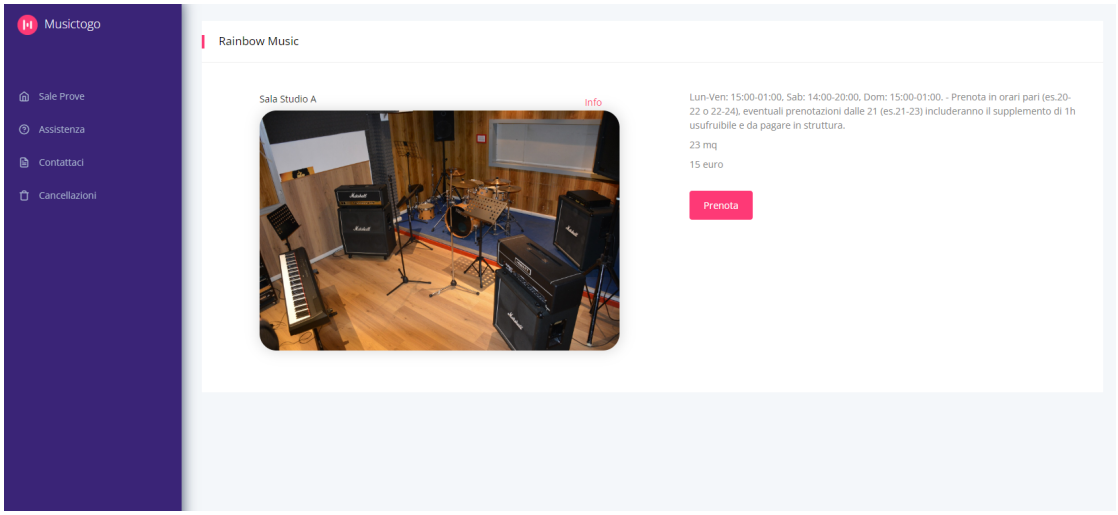

Figura 2.11: Pagina Sala prove ad affitto orario prenotabile online, portale attuale

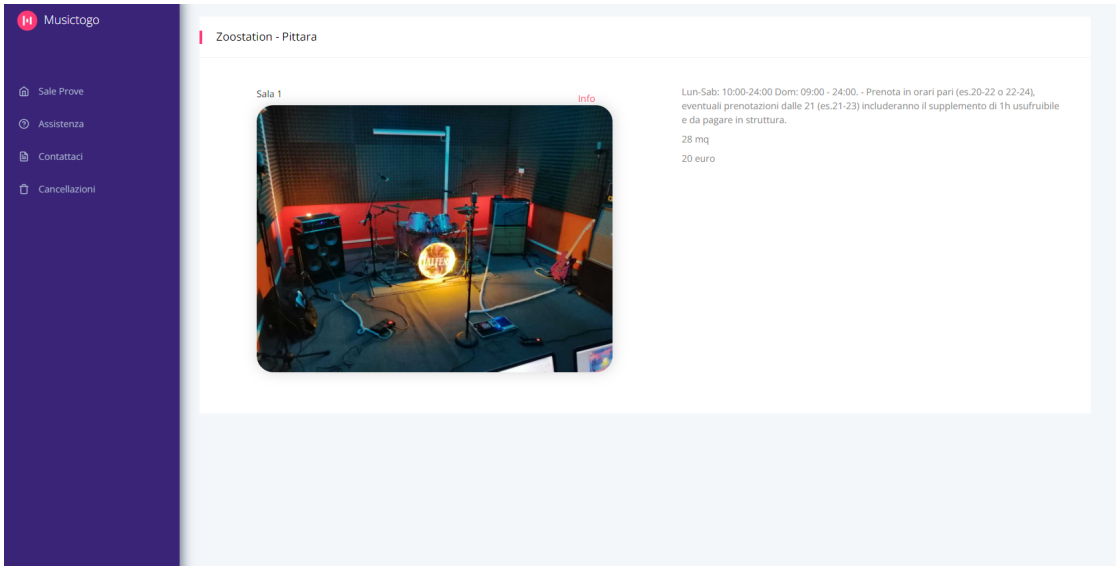

Figura 2.12: Pagina Sala prove ad affitto orario non prenotabile online, portale attuale

In questa sezione, per la corretta esecuzione del Task Flow, si assume che l'utente si trovi nel primo scenario, dunque all'interno di una schermata come quella in Figura 2.11.

Tale schermata mostra informazioni riguardanti la struttura, come il nome e gli orari di apertura, oltre a dettagli specifici sulla saletta, come il suo nome, una sua foto, la dimensione in mq e il costo orario. Inoltre, sopra la foto è presente

un testo cliccabile denominato “Info”, che fa apparire un modale simile a quello illustrato in Figura 2.2, contenente ulteriori dettagli sulla saletta, come ad esempio la strumentazione disponibile.

È inoltre presente il pulsante “Prenota”, la cui pressione permette all’utente di proseguire con il Task Flow; la pressione del pulsante apre l’ultima schermata di questo flusso: il calendario di prenotazione della saletta.

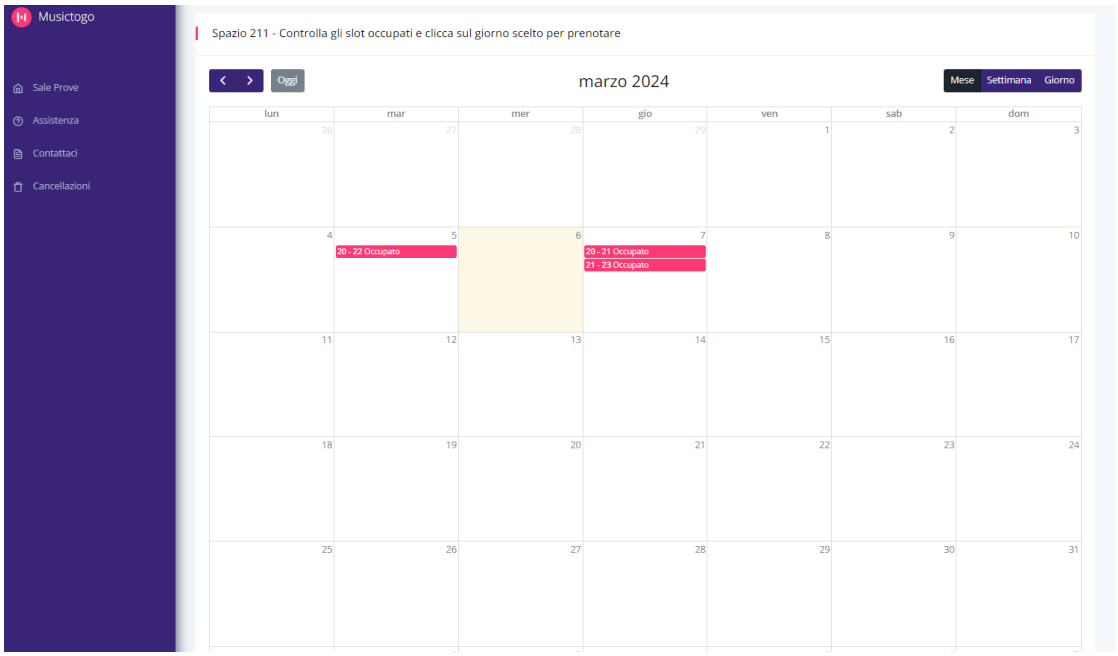

Figura 2.13: Pagina Calendario di prenotazione, portale attuale

La schermata del calendario, illustrata in Figura 2.13, comprende un componente che visualizza lo stato di occupazione giornaliero della saletta in un formato che può essere mensile, settimanale o giornaliero. L’utente, visualizzando nel calendario gli slot orari occupati, può decidere in autonomia la giornata e l’orario in cui desidera riservare per sé la saletta. Scelto il giorno, è necessario cliccarci sopra per far apparire un modale di prenotazione.

Il modale, evidenziato in Figura 2.14, richiede all’utente, attraverso un form, i propri dati personali (nome, email, numero di telefono) e la selezione di dettagli come l’orario, la durata della prenotazione e il metodo di pagamento desiderato. In aggiunta, richiede all’utente di convalidare l’indirizzo email precedentemente inserito, mediante l’invio di un codice OTP, da riportare nell’apposito campo dedicato. Una volta completata la procedura di convalida, sarà possibile finalizzare la prenotazione attraverso la pressione del pulsante posto in basso a destra nel modale, denominato “Conferma” o “Paga” a seconda dell’opzione di pagamento precedentemente selezionata, sia essa “Contanti” o “Online”.

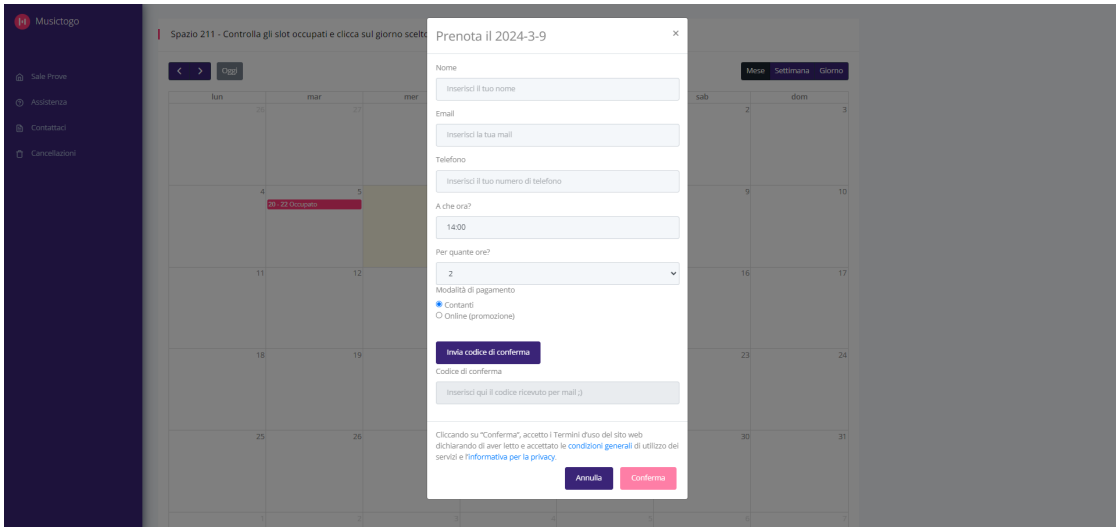

Figura 2.14: Modale di prenotazione di una di saletta, pagina Calendario di prenotazione, portale attuale

In caso di pagamento online, l'utente sarà indirizzato verso un servizio di pagamento terzo. Al contrario, in caso di pagamento in contanti, gli aspetti finanziari verranno gestiti direttamente in struttura. Particolare rilevanza ha il caso in cui un tentativo di prenotazione fallisca a causa di problemi legati agli orari selezionati.

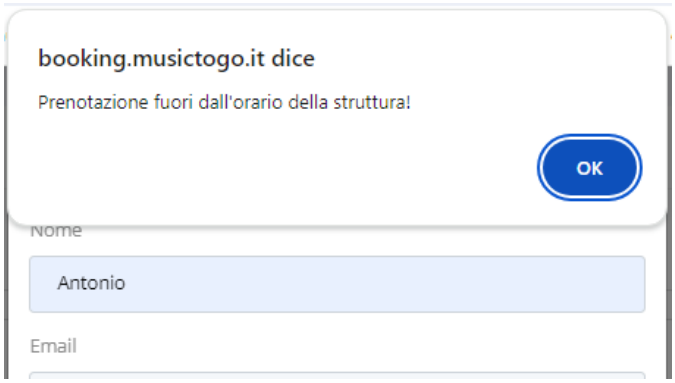

Figura 2.15: Modale di prenotazione compilato con errore, pagina Calendario di prenotazione, portale attuale

Ne è un esempio Figura 2.15: l'utente ha scelto un orario non compatibile con gli orari di apertura della struttura, causando la comparsa del messaggio di errore. A questo punto sarà necessario modificare l'orario tramite il modale (sperando di selezionarne uno che vada bene) oppure, nei casi peggiori, cambiare la data tramite

il ritorno al calendario. Tuttavia, in questo caso, si renderà necessario ripetere la procedura di validazione dell'indirizzo email. Tale problematica verrà affrontata nel Capitolo 3.

Assumendo che tale problematica non sia emersa in fase di prenotazione, questo primo Task Flow del portale Booking può ritenersi concluso.

Schermate del Task Flow "Affitto orario di salette" - versione Mobile

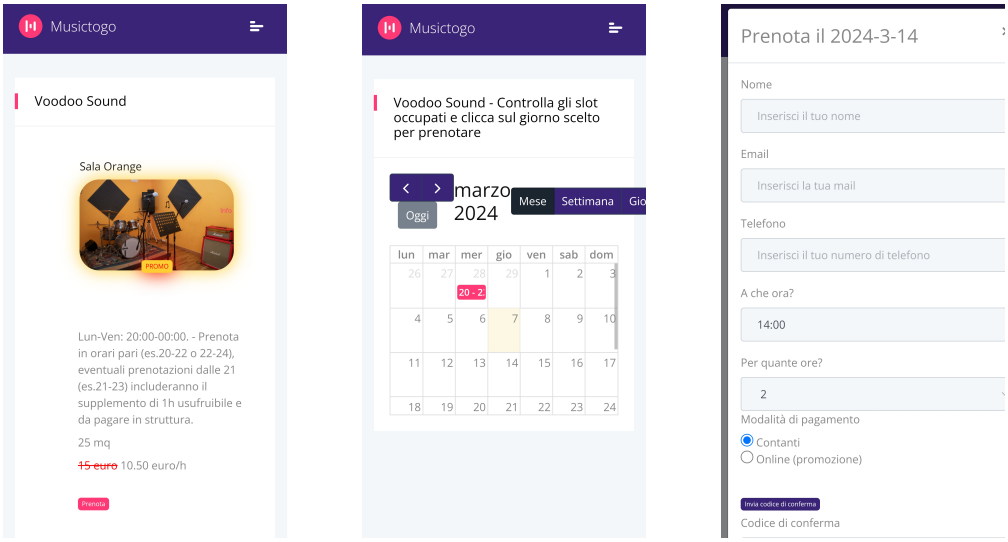

Figura 2.16: Schermate del Task Flow "Affitto orario di salette", portale attuale Mobile

2.2.2 Acquisto di pacchetti

Partendo dal contesto precedentemente descritto nell'introduzione della presente sezione, l'utente intenzionato ad acquistare un pacchetto di uno studio di registrazione dovrà scegliere lo studio desiderato dalla lista e cliccare il pulsante "Prenota" sulla card di riferimento (Figura 2.1).

A seguito della pressione del pulsante "Prenota", verrà aperta la pagina dello studio selezionato. Tale pagina, illustrata in Figura 2.17, ha un layout del tutto simile a quello di una sala prove ad affitto orario (Figura 2.11) con la sola differenza che il pulsante non è denominato "Prenota", ma bensì "Compra".

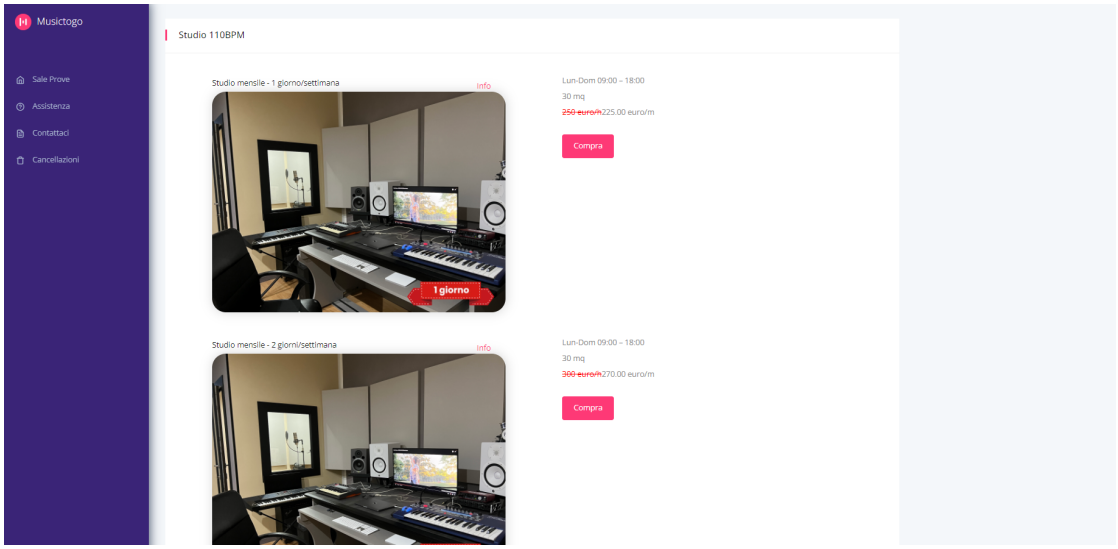

Figura 2.17: Pagina Studio di registrazione, portale attuale

Come precedentemente descritto ad inizio capitolo, gli studi di registrazione non consentono la prenotazione di slot orari, bensì offrono l'acquisto di pacchetti utilizzabili successivamente, in un periodo di tempo che da concordare direttamente con lo studio. Il ruolo di Musictogo, in questo Task Flow, si limita al permettere all'utente di acquistare un pacchetto. La pressione del pulsante, dunque, non comporta l'apertura di un calendario, ma di un semplice modale, illustrato in Figura 2.18.

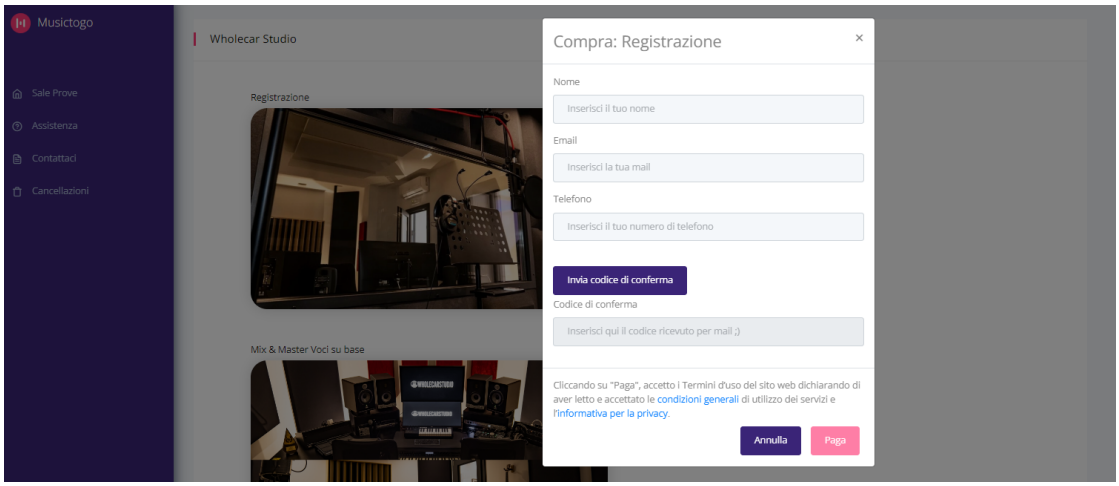

Figura 2.18: Modale di acquisto di un servizio, pagina Studio di registrazione, portale attuale

Il modale richiede all'utente i propri dati personali (nome, email, telefono) e la convalida della email tramite codice OTP. La pressione del pulsante "Paga" indirizzerà l'utente verso un servizio di pagamento terzo.

Il secondo Task Flow del portale Booking può ritenersi concluso.

Schermate del Task Flow "Acquisto di pacchetti" - versione Mobile

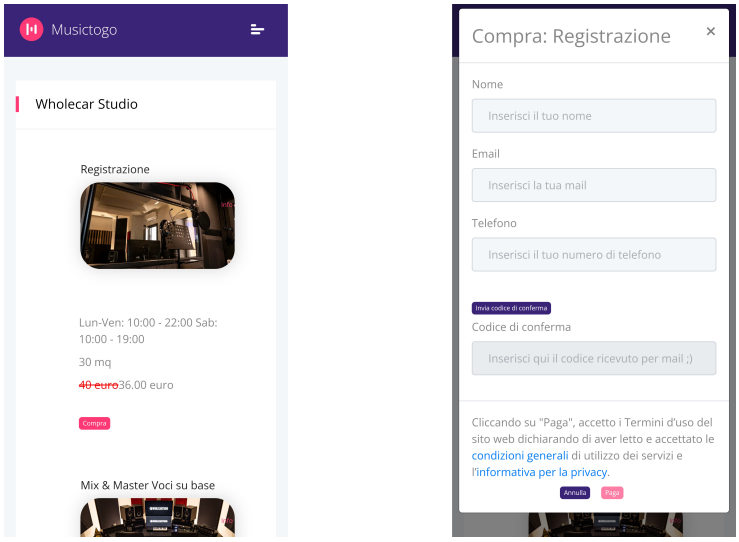

Figura 2.19: Schermate del Task Flow "Acquisto di pacchetti", portale attuale Mobile

2.2.3 Cancellazione di prenotazioni

L'utente intenzionato a cancellare una prenotazione effettuata su una saletta ad affitto orario dovrà prima di tutto fare uso della Sidebar per aprire la pagina "Cancellazioni" (Figura 2.9). Dovrà quindi inserire i dati necessari e premere il pulsante "Cancella". A seguito della pressione del pulsante "Cancella", l'utente riceverà un feedback visivo riguardante l'esito della cancellazione, come illustrato in Figura 2.20.

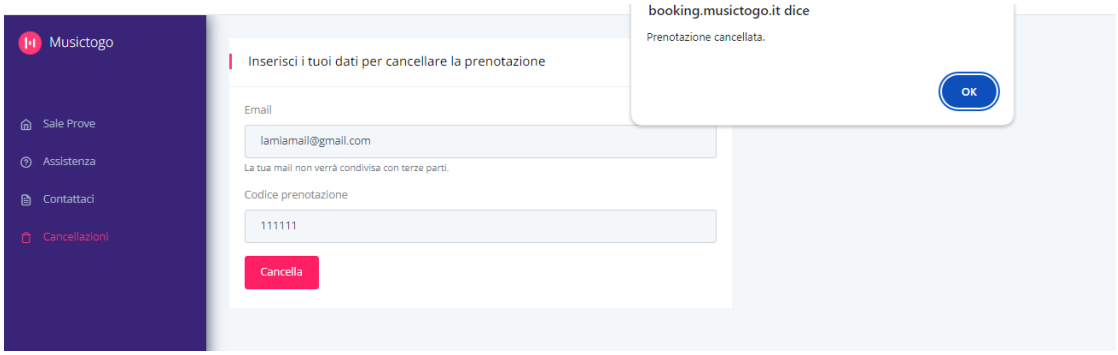

Figura 2.20: Avvenuta cancellazione di una prenotazione, pagina Cancellazioni, portale attuale

Nel caso la procedura di cancellazione non dovesse andare a buon termine, l'utente riceverà nel feedback la motivazione dietro tale fallimento.

Assumendo un esito positivo della procedura, anche questo (breve) Task Flow può ritenersi concluso.

Schermate del Task Flow "Cancellazione di prenotazioni" - versione Mobile

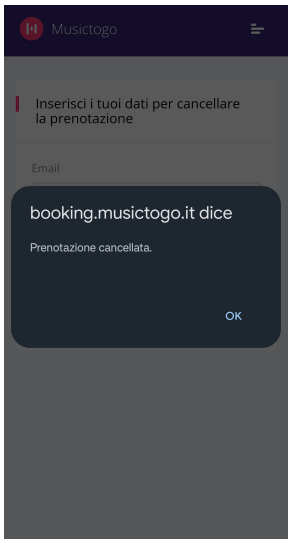

Figura 2.21: Schermate del Task Flow "Cancellazione di prenotazioni", portale attuale Mobile

2.3 Altre schermate

Di seguito, altre schermate del portale non ancora rese note dalle precedenti sezioni.

2.3.1 Sale prove ad affitto mensile

Le sale prove ad affitto mensile, così come alcune sale prove ad affitto orario, utilizzano Musictogo solo come vetrina. Di seguito, in Figura 2.22, è illustrata la pagina di una sala prove ad affitto mensile.

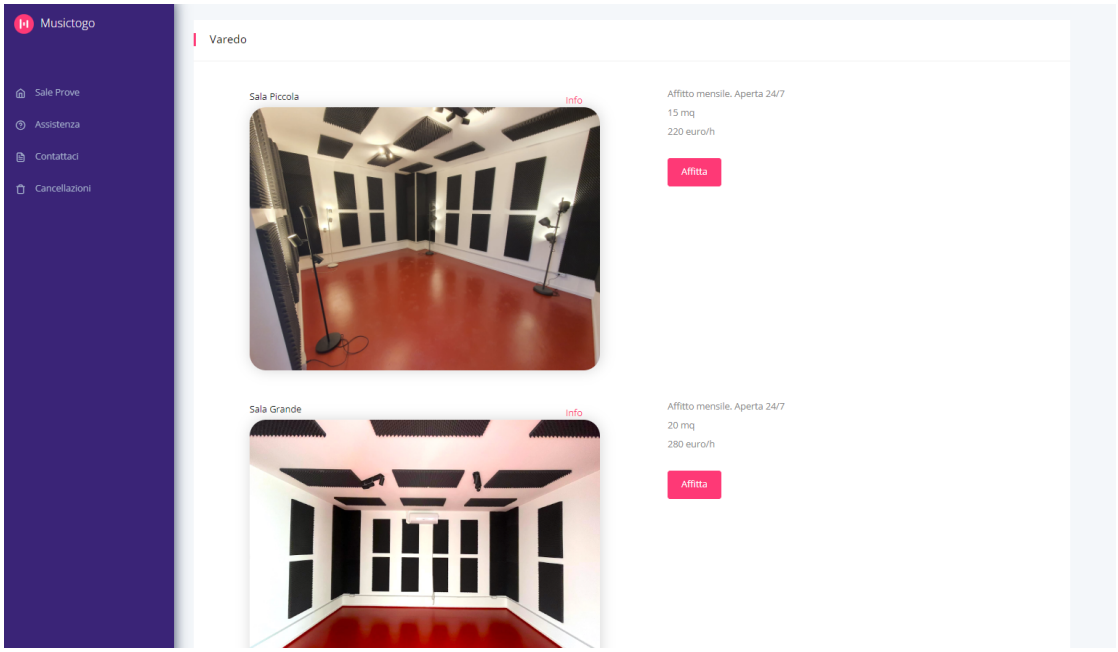

Figura 2.22: Pagina Sala prove ad affitto mensile, portale attuale

A seguito della pressione di "Affitta", l'utente vedrà aprirsi un modale informativo, come illustrato in Figura 2.23.

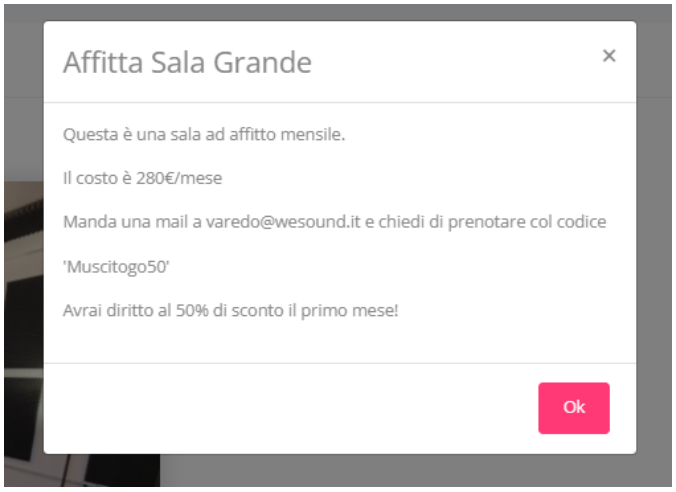

Figura 2.23: Modale "Affitta", pagina Sala prove ad affitto mensile, portale attuale

Sale prove ad affitto mensile - versione Mobile

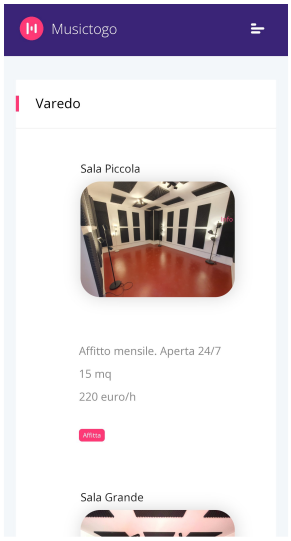

Figura 2.24: Pagina Sala prove ad affitto mensile, portale attuale Mobile

Capitolo 3

Analisi dell'interfaccia

Questo capitolo tratta l'analisi dell'interfaccia utente dell'attuale portale Booking di Musictogo. Le due sezioni di cui si compone descrivono, rispettivamente, i principi teorici alla base della metodologia di analisi utilizzata (Sezione 3.1) e l'analisi vera e propria (Sezione 3.2)

3.1 Aspetti teorici

La valutazione dell'interfaccia utente (*UI Evaluation*) testa l'*usabilità*, la *funzionalità* e l'*accettabilità* di un sistema interattivo [9]. Nel dettaglio [10]:

- l'*usabilità* rappresenta la facilità con cui gli utenti possono utilizzare le funzionalità del sistema per raggiungere i propri obiettivi;
- la *funzionalità* rappresenta la capacità del sistema di permettere agli utenti di svolgere i task desiderati;
- l'*accettabilità* rappresenta il grado di accettazione del sistema da parte dell'utente.

Esistono principalmente due approcci per la valutazione di interfacce utente [9]:

- *Analisi svolte da esperti.* Particolarmente utili per valutare le prime versioni di un nuovo design, o per identificare le parti da riprogettare in una vecchia versione dell'interfaccia.
- *Partecipazione degli utenti.* Test e valutazioni sull'interfaccia tramite l'aiuto di potenziali utenti. Richiede un prototipo funzionante. Tale approccio verrà descritto nel dettaglio all'interno del Capitolo 7.

È molto importante condurre spesso valutazioni sull'interfaccia che si vuole sviluppare. In particolare, la valutazione iniziale dovrebbe avvenire prima di avviare qualsiasi lavoro di implementazione, consentendo così modifiche al progetto prima di investire significative risorse nell'implementazione reale. In generale, più tardi si scopre un errore, più è costoso correggerlo e, di conseguenza, minori sono le probabilità di farlo. Tuttavia, effettuare test con la partecipazione degli utenti durante tutto il processo di progettazione può risultare oneroso in termini di tempo e risorse. Inoltre, è difficile ottenere una valutazione accurata dell'esperienza utente se si parte da progetti e prototipi incompleti. Di conseguenza, sono stati proposti diversi metodi per valutare le interfacce utente attraverso analisi svolte da esperti. L'obiettivo principale di tali analisi è individuare ciò che potrebbe causare difficoltà all'utente durante l'utilizzo del sistema. Queste metodologie sono applicabili in qualsiasi fase dello sviluppo. Sono inoltre relativamente economiche, poiché non richiedono il coinvolgimento degli utenti. Tuttavia, non valutano l'uso reale del sistema, ma solo se esso rispetta o meno i principi di usabilità [9].

Nella successiva sottosezione verrà descritta, a livello teorico, la metodologia di analisi che prende il nome di *Valutazione euristica*; saranno descritti i principi teorici alla base di tale metodologia e le modalità di esecuzione. La valutazione euristica verrà poi applicata, nella Sezione 3.2, al portale Booking di Musictogo.

3.1.1 Valutazione euristica

La *Valutazione euristica* è una metodologia di valutazione di interfacce utente, ideata da Jakob Nielsen e Rolf Molich, il cui obiettivo è individuare, mediante l'utilizzo di un set di criteri euristici, problemi nel design di un sistema al fine di migliorarne l'usabilità. Un'euristica [9] è una guideline, che può guidare una decisione di design o essere utilizzata per criticarne una già presa. La valutazione euristica è snella, flessibile, veloce e relativamente poco costosa; ciò l'ha resa subito molto popolare e di conseguenza molto utilizzata.

Uno dei punti di forza della valutazione euristica è la sua versatilità, poiché può essere applicata in qualsiasi fase dello sviluppo e del ciclo di vita di un sistema. Essa si può applicare, ad esempio [10]:

- *Durante la progettazione del sistema.* Ciò permette di identificare e risolvere immediatamente i problemi che, se ignorati, potrebbero diventare più costosi da risolvere in futuro.
- *Prima di un re-design del sistema.* Per comprendere chiaramente le problematiche presenti nella versione attuale del sistema, in modo da sviluppare una nuova versione che eviti, per quanto possibile, gli stessi problemi.
- *Prima del rilascio del sistema.* Consente di "smussare gli angoli" prima che il sistema raggiunga gli utenti finali.

- *Durante il normale funzionamento del sistema.* Utile per avere una documentazione scritta, dei problemi noti o sospetti, da poter utilizzare come punto di partenza per future revisioni.

L'idea di base è che, in maniera indipendente, diversi valutatori ricerchino problemi di usabilità nel sistema, utilizzando come riferimento un set di dieci euristiche fornite da Nielsen e Molich. Viene richiesta la presenza di più valutatori in quanto statisticamente è poco probabile che un unico valutatore riesca a trovare più di 1/3 dei problemi. I valutatori sono liberi di modificare a proprio piacimento, in base al contesto, le dieci euristiche proposte, ignorandone alcune o definendone di nuove [9]. Più nel dettaglio, una valutazione euristica completa si divide in quattro fasi principali [10]:

1. *Apprendimento del sistema.* Ai valutatori vengono date informazioni sul dominio e sullo scenario da valutare.

2. *Valutazione individuale.* I singoli valutatori applicano, in maniera indipendente l'uno dall'altro, la valutazione euristica al sistema.

3. *Assegnazione del grado di severità.* Ogni valutatore assegna, autonomamente, un livello di gravità a ciascuna violazione da lui individuata. Successivamente, i risultati ottenuti vengono combinati per creare un rapporto finale condiviso da tutti i valutatori.

4. *Fase di debriefing.* Analisi, in collaborazione con il team di design, del rapporto precedentemente prodotto.

Particolarmente rilevante è la fase di valutazione individuale [10]. In questa fase, ogni valutatore, basandosi su un set di task definiti dal team di design, esamina l'interfaccia, i suoi componenti e le sensazioni durante l'interazione. Le regole euristiche vengono utilizzate come promemoria di ciò che si deve cercare. La valutazione individuale si concretizza in un documento che elenca i problemi di usabilità identificati dal valutatore. Ogni problema deve essere trattato separatamente. Per ciascun problema, il documento deve indicare in modo chiaro e oggettivo quali euristiche sono state violate, per quale motivo e il grado di severità.

Come precedentemente accennato, Nielsen raccomanda l'uso di un set di dieci regole euristiche. Tali euristiche, strettamente legate a comuni principi e guideline di design, sono [9]:

1. *Visibility of system status.* L'utente dovrebbe essere sempre aggiornato su ciò che sta accadendo nel sistema, tramite dei feedback appropriati in base alla durata dell'operazione in corso. Ne sono un esempio le progress bar, che indicano graficamente la percentuale di completamento di un'operazione in corso.

2. *Match between system and the real world.* Il sistema deve comunicare con l'utente utilizzando un linguaggio ad esso familiari ed evitando l'uso di un gergo tecnico. È importante seguire le convenzioni del mondo reale, facendo apparire le informazioni in un ordine naturale e logico. Un esempio può essere l'associazione di un'icona ad un determinato comportamento.

3. *User control and freedom.* Capita frequentemente che l'utente scelga una funzionalità del sistema per errore. È essenziale fornire sempre un'opzione di "uscita di emergenza" per permettere all'utente di abbandonare una situazione non voluta, ad esempio attraverso un pulsante "Torna indietro".

4. *Consistency and standards.* Le diverse interfacce di un sistema dovrebbero mantenere, tra di loro, una certa coerenza, per prevenire confusione da parte dell'utente, che potrebbe altrimenti domandarsi se parole, contesti o azioni differenti abbiano lo stesso significato. È raccomandato aderire a convenzioni e standard riconosciuti.

5. *Error prevention.* La parola chiave è "prevenzione". Puntare alla creazione di interfacce intuitive che mirino a prevenire gli errori prima che si verifichino, evitando situazioni che potrebbero trarre in inganno l'utente e fornendo opzioni di conferma o annullamento prima di finalizzare determinate operazioni. Ne è un esempio è il classico modale "Sei sicuro di voler...?".

6. *Recognition rather than recall.* Minimizzare il carico cognitivo dell'utente rendendo visibili oggetti, azioni ed opzioni. Far sì che l'utente non debba ricordare informazioni nel passaggio da una schermata all'altra dell'interfaccia. Questa regola viene spesso applicata in operazioni bancarie multistep, come la realizzazione di un bonifico. In ogni step, all'utente viene presentato un resoconto delle azioni compiute nelle fasi precedenti.

7. *Flexibility and efficiency of use.* Disporre il sistema di "acceleratori" (banalmente, delle scorciatoie) che permettano agli utenti di velocizzare ed efficientare la propria esperienza d'uso del sistema. L'utilizzo di tali acceleratori va visto come opzionale, un'alternativa all'esperienza d'uso prevista di default.

8. *Aesthetic and minimalist design.* Le interfacce dovrebbero avere un design esteticamente piacevole e non dovrebbero contenere informazioni irrilevanti o raramente necessarie. L'effetto dell'aggiunta di un'informazione superflua è semplicemente quello di offuscare e diminuire la visibilità di un'informazione che è invece rilevante.

9. *Help users recognize, diagnose and recover from errors.* Nell'eventualità di un errore è opportuno che il corrispondente messaggio informativo sia presentato

in un linguaggio comprensibile, che evidenzi in maniera precisa l'errore e fornisca una proposta di risoluzione.

10. *Help and documentation.* Nonostante un sistema possa vantare un'usabilità ottimale, è improbabile che non vi siano utenti che possano incontrare difficoltà. Di conseguenza, è essenziale mettere a disposizione dell'utente documentazione e assistenza. L'aiuto dovrebbe essere facilmente individuabile e focalizzato sulle attività dell'utente. Dovrebbe inoltre fornire dettagli sui passaggi da compiere e mantenere dimensioni contenute.

Il grado di severità è, invece, un indicatore che valuta la gravità di un problema di usabilità. È una misura che si basa sulle caratteristiche di *frequenza*, *impatto* e *persistenza* del problema [10]. Il valutatore deve assegnare, ad ogni problema trovato, un grado di severità su una scala che va da 0 a 4 :

Score	**Severità**	**Descrizione**
0	Nessun problema	Non sono affatto d'accordo che si tratti di un problema di usabilità
1	Solo un problema estetico	Non è necessario ripararlo a meno che non sia disponibile del tempo extra sul progetto
2	Problema di usabilità minore	La correzione di questo problema dovrebbe avere una priorità bassa
3	Grave problema di usabilità	Da correggere rapidamente, andrebbe data la massima priorità
4	Catastrofe di usabilità	È imperativo risolvere questo problema prima del rilascio del prodotto

Tabella 3.1: Severity rating scale [9]

Al termine delle valutazioni individuali, i valutatori, come precedentemente accennato, si riuniranno per discutere delle proprie valutazioni e per produrre un rapporto finale condiviso da sottoporre al team di design.

3.2 Valutazione euristica del portale Booking

All'interno della presente sezione si applicherà la *Valutazione euristica* al portale Booking di Musictogo. L'obiettivo è identificare potenziali problemi di usabilità, utilizzando come riferimento le dieci euristiche di Nielsen precedentemente illustrate nella Sottosezione 3.1.1, per evitare di ripetere tali problemi nel nuovo design.

3.2.1 Modalità di esecuzione

Opererò seguendo una versione "semplificata" del processo di Valutazione euristica. La valutazione sarà condotta da me in solitaria, consapevole dei limiti che comporta l'esecuzione di una valutazione in autonomia. Ho iniziato ad esplorare il portale mettendomi nei panni di un utente che lo visita per la prima volta. Ho subito provato ad effettuare prenotazioni per diverse sale prove. Tuttavia, mi sono imbattuto in alcuni ostacoli: alcune sale non consentivano la prenotazione delle proprie salette, mentre in altri casi ho avuto problemi con il calendario ed il modale di prenotazione a causa delle date e degli orari selezionati. In linea generale, ho cercato di seguire i percorsi descritti, nella Sezione 2.2, dai principali Task Flow. Ho quindi abbozzato la mia valutazione annotando in un documento tutte le violazioni che ho riscontrato mentre eseguivo i vari task e, più in generale, mentre sfogliavo le pagine del sito. Successivamente, ho organizzato i miei pensieri, classificando e ordinando le violazioni in base all'euristica che ho ritenuto fosse stata violata.

3.2.2 Violazioni euristiche rilevate

Il dettaglio delle singole violazioni rilevate è disponibile all'Appendice A, in calce al presente documento. Questa sottosezione vuole riassumere i risultati ottenuti descrivendoli all'interno delle tabelle 3.2 e 3.3.

La Tabella 3.2 fornisce una descrizione dettagliata del numero di violazioni rilevate per ciascuna euristica, ed il relativo impatto, in percentuale, sul totale. Sono state rilevate un totale di 101 violazioni, distribuite come segue:

Euristica violata	**Numero di violazioni**	**Percentuale**
H1	11	10,89%
H2	11	10,89%
H3	1	0,99%
H4	40	39,60%
H5	12	11,88%
H6	8	7,92%
H7	4	3,96%
H8	3	2,97%
H9	1	0,99%
H10	10	9,90%
Totale	**101**	**100,00%**

Tabella 3.2: Distribuzione violazioni euristiche del portale attuale

L'euristica che risulta essere la più violata è *H4 Consistency and standards*, con un totale di 40 violazioni. Questo numero costituisce il 39,60% del totale delle violazioni.

Considerando invece le severità associate a ciascuna violazione, la Tabella 3.3 illustra, per ogni euristica, il numero di violazioni per ciascun grado di severità e la media ponderata del grado di severità delle violazioni:

Euristica violata	N° violazioni per severità					Media ponderata
	0	**1**	**2**	**3**	**4**	**severità**
H1	0	0	4	6	1	2,73
H2	0	2	4	3	2	2,45
H3	0	0	1	0	0	2,00
H4	0	8	12	15	5	2,42
H5	0	0	1	8	3	3,17
H6	0	0	1	6	1	3,00
H7	0	1	2	1	0	2,00
H8	0	1	2	0	0	1,67
H9	0	0	0	1	0	3,00
H10	0	0	2	8	0	2,80
Totale	**0**	**12**	**29**	**48**	**12**	**2,59**

Tabella 3.3: Severità associate alle violazioni euristiche del portale attuale

L'euristica che presenta la media ponderata più elevata in termini di gravità delle violazioni risulta essere *H5 Error prevention*, con un valore medio di 3,17. Questo valore supera del 22,39% la media ponderata complessiva, che si attesta, invece, a 2,59. Il grado di severità più comunemente riscontrato è *3 - Grave problema di usabilità*, con un totale di 48 violazioni a esso collegate.

3.2.3 Motivazioni per l'aggiornamento dell'interfaccia

I dati emersi dalla valutazione euristica evidenziano la necessità di un rinnovamento del portale. Il portale attuale presenta diverse aree in cui le euristiche vengono violate. L'euristica *H4 Consistency and standards* è quella più violata, mentre l'euristica *H5 Error prevention* presenta la media ponderata più elevata in termini di gravità delle violazioni. Questi dati, uniti agli altri emersi grazie alla valutazione euristica, vanno ritenuti un buon punto di partenza per la realizzazione della nuova versione del portale.

Capitolo 4

Revisione dei mockup esistenti

Questo capitolo descrive il lavoro svolto per aggiornare i mockup, creati nel 2021 da Musictogo, messi a disposizione come punto di partenza dello sviluppo del nuovo design.

Il capitolo tratta, nelle sue sezioni, i passi che hanno guidato la revisione di tali mockup. Le sezioni sono così suddivise:

- *I mockup iniziali* (Sezione 4.1). Questa sezione introdurrà il lettore ai mockup del 2021, illustrandone le schermate disponibili.
- *Analisi dei mockup* (Sezione 4.2). I mockup verranno analizzati per individuarne le carenze e le difformità rispetto al portale attuale. I mockup saranno poi oggetto di una valutazione euristica al fine di far emergere eventuali violazioni di usabilità.
- *Aggiornamento dei mockup* (Sezione 4.3). Descriverà la fase di aggiornamento dei mockup. Essa consisterà nella revisione degli elementi inizialmente presenti nei mockup e nella creazione, talvolta, di nuovi elementi, dei quali i mockup erano sprovvisti. Si punterà inoltre a correggere le problematiche che emergeranno nella Sezione 4.2. In questa fase, sarà necessario tenere conto anche della valutazione euristica fatta nella Sezione 3.2, per assicurare che la nuova interfaccia non cada vittima degli stessi problemi del portale attuale.

4.1 I mockup iniziali

I mockup sono stati sviluppati dal team di Musictogo utilizzando il software di design digitale Figma [11].

Le successive sottosezioni esploreranno le schermate di quella che può essere definita la prima versione dei mockup, ovvero quella sviluppata nel 2021. Ciascuna sottosezione presenterà immagini estratte dai mockup. Le immagini verranno mostrate al solo scopo illustrativo, per renderle note al lettore. In questa fase, non verranno analizzati i mockup. L'analisi sarà oggetto di discussione nella Sezione 4.2.

4.1.1 Homepage

La pagina, illustrata in Figura 4.1 assieme a dettagli circa la sua interattività, presenta uno stile che risulta rinnovato rispetto a quello del portale attuale.

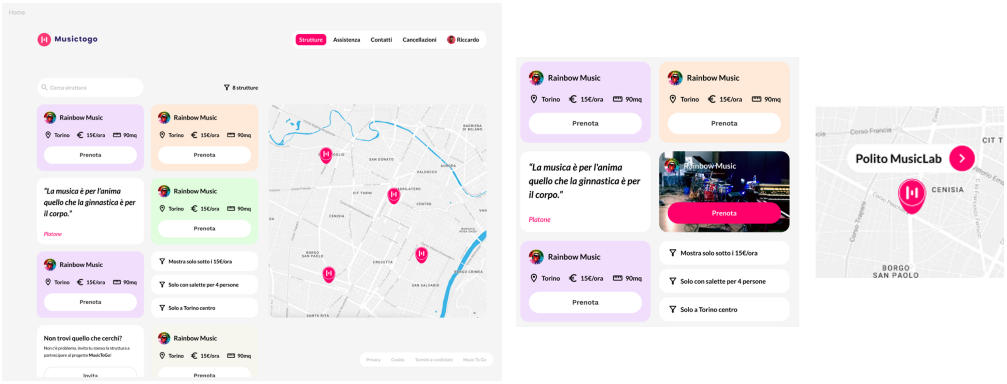

Figura 4.1: Homepage, mockup iniziali

A grandi linee, la struttura interna della pagina è rimasta invariata. Essa presenta, infatti, come la Homepage del portale attuale (Figura 2.1), una struttura a due colonne: la colonna sinistra contenente la lista delle strutture ed i filtri, la colonna destra contenente la mappa. Le differenze sostanziali tra le due versioni della pagina stanno nella forma con cui viene presentato il contenuto all'interno delle due colonne:

- la lista delle strutture non è più organizzata mediante paginazione, ma tramite un più semplice scrolling lineare. Il suo layout passa da uno a singola colonna, ad uno a doppia colonna. Oltre alle card delle strutture, all'interno della lista sono stati aggiunti degli shortcut per effettuare specifiche operazioni di filtraggio con un solo click;

- la card delle singole strutture hanno un aspetto totalmente rivisto, e contengono al loro interno un maggior numero di informazioni. A differenza di quanto avviene nel portale attuale, si prevede di rendere le card interattive tramite la variazione del loro aspetto al passaggio del mouse, come illustrato nella seconda immagine della figura;
- la mappa risulta avere un aspetto gradevole, rinnovato e minimale. I suoi marker non sono più quelli di default di Google Maps, ma sono stati "customizzati". Nella terza immagine della Figura viene illustrata l'interattività dei singoli marker, data dall'apparizione, al passaggio del cursore del mouse, di una "info windows" sopra ai marker stessi.

Homepage dei mockup iniziali - versione Mobile

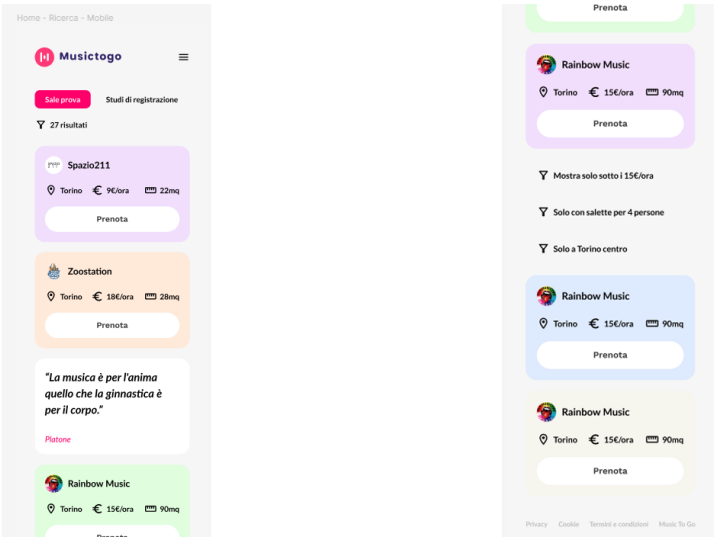

Figura 4.2: Homepage, mockup iniziali Mobile

4.1.2 Pagina Struttura

La pagina Struttura, illustrata in Figura 4.3, ha subito notevoli modifiche rispetto alla pagina del portale attuale.

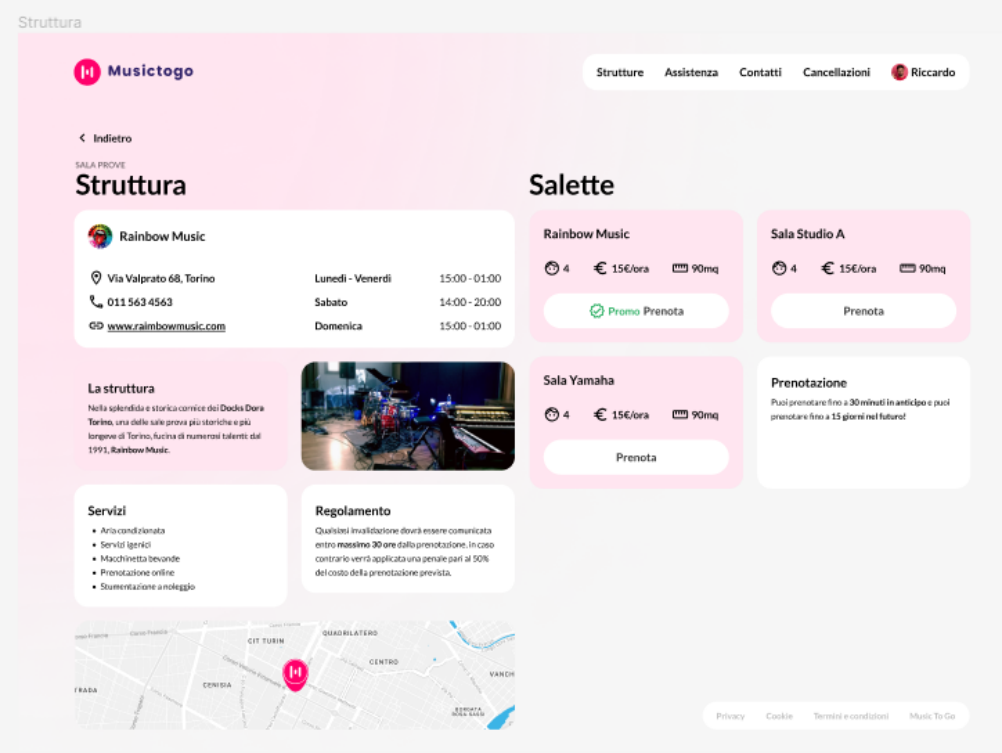

Figura 4.3: Pagina Struttura, mockup iniziali

A differenza di quanto avviene nel portale attuale, è presente una netta distinzione, espressa tramite la creazione di un layout a due colonne, tra gli elementi che descrivono la struttura nel dettaglio e gli elementi che presentano le sue salette. È stato inoltre aggiunto il pulsante “Indietro”, non presente nel portale attuale. Gli orari sono stati riorganizzati in una struttura ben definita.

Pagina Struttura dei mockup iniziali - versione Mobile

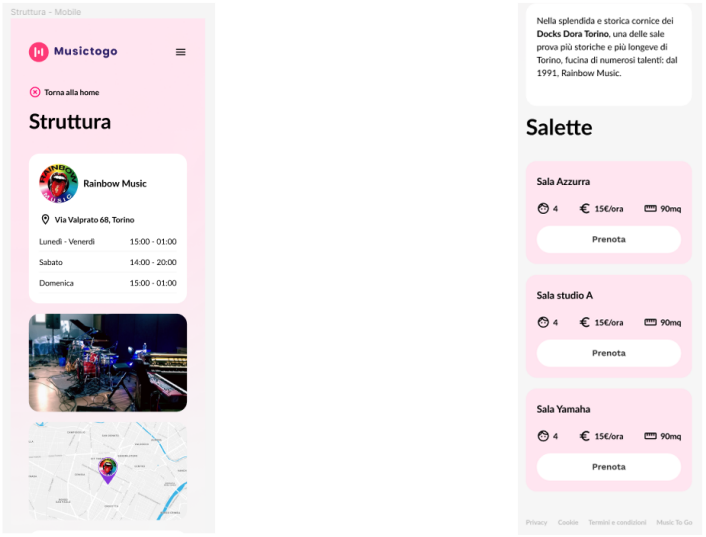

Figura 4.4: Pagina Struttura, mockup iniziali Mobile

4.1.3 Pagina Saletta

La pagina Saletta, illustrata in Figura 4.5, sostituisce la pagina calendario del portale attuale e il relativo modale. Il suo stile, a primo impatto, sembra coerente con lo stile utilizzato nelle altre pagine.

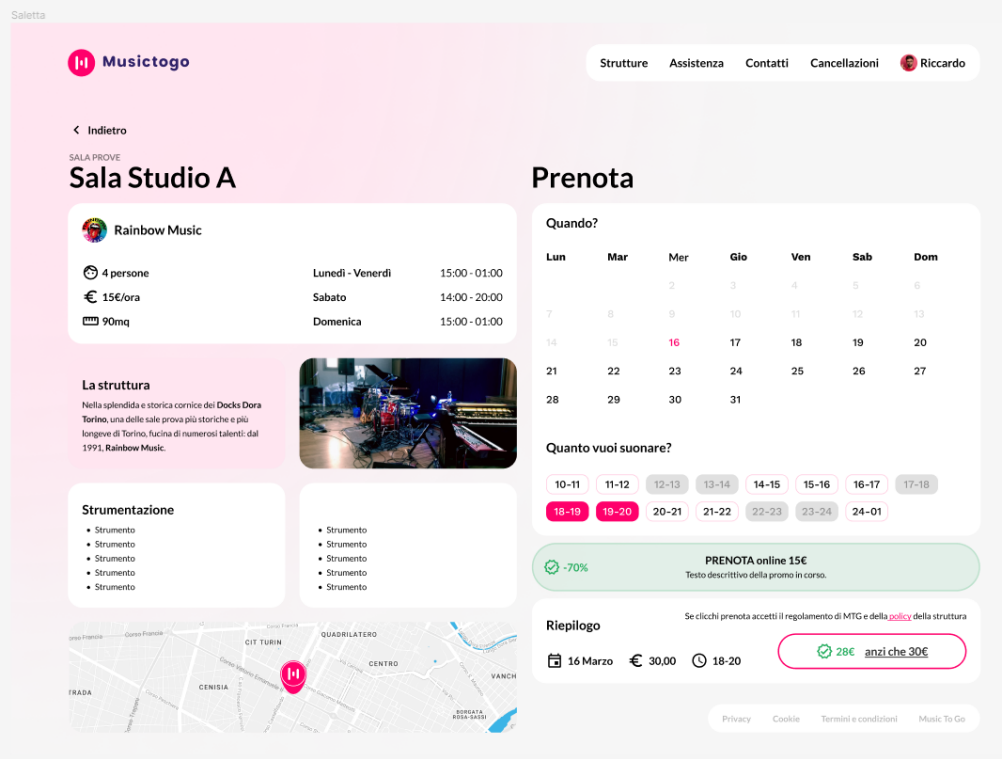

Figura 4.5: Pagina Saletta, mockup iniziali

Mantenendo lo stile della pagina Struttura, viene utilizzato un layout a due colonne. La colonna sinistra contiene dettagli sulla saletta e informazioni ereditate dalla pagina della relativa struttura. La colonna destra è invece dedicata alla procedura di prenotazione: presenta un calendario, gli slot orari occupati/liberi/scelti per la data del calendario selezionata e un breve riepilogo.

Pagina Saletta dei mockup iniziali - versione Mobile

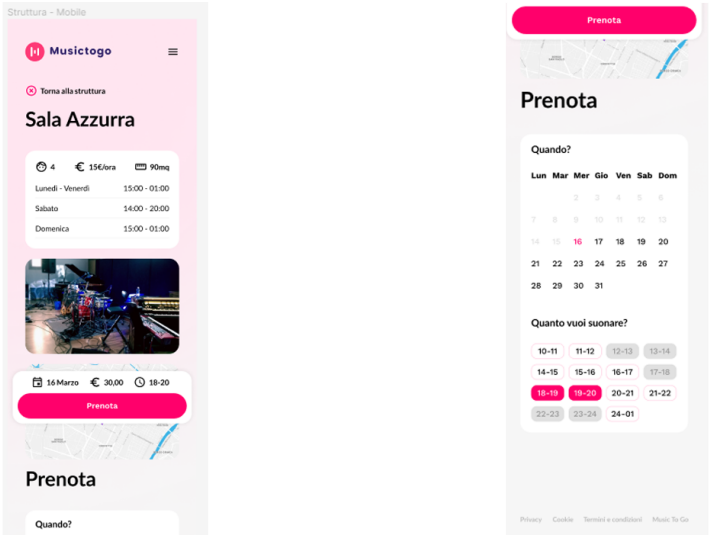

Figura 4.6: Pagina Saletta, mockup iniziali Mobile

4.1.4 Pagine Conferma e Fallimento

Le pagine Conferma e Fallimento, illustrate in Figura 4.7, sono pensate per gestire i casi in cui la prenotazione va o meno a buon fine.

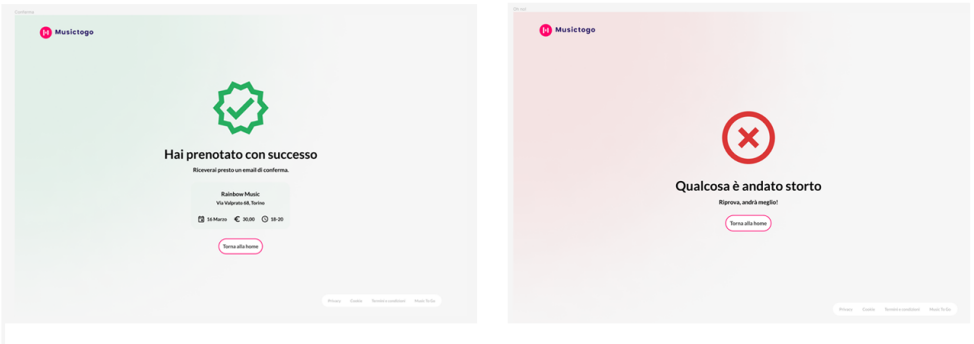

Figura 4.7: Pagine Conferma e Fallimento, mockup iniziali

La pagina Conferma viene visualizzata dopo il completamento della procedura di prenotazione; essa presenta un riepilogo con i dettagli della prenotazione. La pagina Fallimento, al contrario, viene mostrata quando la procedura di prenotazione non va a buon fine o quando, più generalmente, avviene un problema.

4.2 Analisi dei mockup

I mockup forniti da Musictogo costituiscono il punto di partenza di un lavoro più ampio. Essendo stati concepiti nel 2021, ad oggi risultano parzialmente obsoleti poiché non tengono conto delle modifiche apportate nel tempo al portale di booking. Inoltre, presentano lacune dovute alla parziale assenza di componenti, funzionalità e pagine. Risulta dunque necessaria una prima attenta analisi per individuare le carenze e le difformità presenti. Successivamente, verrà effettuata un'analisi euristica dei prototipi, prendendo in esame le schermate disponibili.

4.2.1 Carenze e difformità rispetto al portale attuale

I mockup sono stati creati dal team di Musictogo nel 2021. Ad oggi, essi risultano incompleti, in quanto non presentano prototipi che illustrino le versioni aggiornate di alcune delle pagine che sono invece attualmente presenti nel portale; inoltre,

i prototipi di cui si compongono non vengono descritti nella totalità delle loro funzionalità.

Da quando sono stati creati questi mockup, il portale ha visto l'aggiunta di nuove funzionalità e la rivalutazione, o addirittura la rimozione, di altre. Il back-end di Musictogo si è adattato a questi cambiamenti per poter servire al meglio il portale; i prototipi, invece, no. Essi non tengono in considerazione i "limiti" imposti dal back-end di Musictogo, rappresentando informazioni che nella realtà non sono disponibili nel formato desiderato o non sono disponibili affatto.

Le problematiche appena introdotte, rappresentative delle carenze e delle difformità dei mockup iniziali rispetto al portale attuale, sono state oggetto della prima analisi effettuata sui mockup iniziali, volta ad individuare, nel dettaglio, le specifiche carenze e difformità.

Le carenze

Per carenze, si intendono elementi, quali componenti specifici, funzionalità o pagine, non illustrati all'interno dei mockup. Messi a confronto con le pagine del portale attuale, i mockup risultano carenti delle pagine Assistenza, Contatti, Cancellazioni, Struttura nelle varianti descrittive di sale prove ad affitto mensile e studi di registrazione, Saletta nella variante descrittiva di salette non prenotabili, Servizio di studio di registrazione, 404 e descrittive di errori di comunicazione con il server. Risulta inoltre assente un prototipo che descriva come rappresentare gli elementi della Navbar nella versione Mobile dell'applicazione. Nel dettaglio dei componenti e delle funzionalità assenti nelle singole pagine, è emersa l'assenza, tra i componenti illustrati nei prototipi di Homepage, di uno che descriva il menu dei filtri aperto; non viene dato inoltre alcun dettaglio circa la risposta del sistema a seguito dell'applicazione di filtri o dell'interazione con la mappa. Nei prototipi della pagina Struttura non viene data invece alcuna indicazione circa l'eventuale interattività della mappa e delle card delle salette. Nei prototipi della pagina Saletta, allo stesso modo della pagina Struttura, non vengono date indicazioni circa l'eventuale interattività della mappa; non viene indicato cosa mostrare nel caso in cui la saletta non fosse prenotabile [1] e non vengono descritti molti degli elementi che, se al contrario, la saletta risultasse prenotabile, sarebbero necessari per la procedura di prenotazione, quali i form di inserimento dei dati, i componenti per la verifica dell'Email e gli elementi atti alla gestione di casi limite come la perdita di disponibilità di un determinato slot orario.

[1] caso *Vetrina*, descritto nel Capitolo 2

Le difformità

Le difformità si riferiscono a quelle funzionalità che, nel corso del tempo, sono state aggiunte, rivalutate o, addirittura, rimosse rispetto a quando furono sviluppati i mockup, e quegli elementi che, nonostante siano rappresentati nei mockup, non sono attualmente disponibili a back-end nel formato desiderato o non sono disponibili affatto. La funzionalità di Login è la prima difformità presente nei mockup in quanto, seppur in passato si sia probabilmente discusso sull'inserimento di tale funzione, attualmente Musictogo preferisce offrire un servizio login-less.

Entrando nello specifico delle singole pagine, sono presenti diverse difformità dovute all'impossibilità tecnica di sviluppare determinate funzionalità.

Le difformità riscontrate all'interno della Homepage sono:

- l'impossibilità di creare, se non staticamente, gli shortcut di filtraggio presenti in lista;
- l'assenza, lato back-end, di un archivio di citazioni dal quale prelevare le singole citazioni da mostrare in pagina;
- l'assenza, lato back-end, di un supporto a procedure di "Invito" di strutture;
- l'incompatibilità di formato con cui il sistema fornisce le informazioni sull'indirizzo della struttura: non viene fornito, in maniera isolata, il nome della città di una struttura, ma solo il suo indirizzo completo, spesso in formati diversi da struttura a struttura.

Le difformità riscontrate all'interno della pagina Struttura sono:

- l'assenza, lato back-end, di informazioni circa il sito web delle strutture;
- l'Incompatibilità di formato con cui il sistema fornisce gli orari. Gli orari delle strutture, quando questa analisi è stata svolta, non erano forniti dal back-end in un formato malleabile, che permettesse cioè di trasformare le informazioni per renderle fruibili secondo lo stile dei mockup. Forniva una stringa, scritta, per ogni struttura, a mano con formati spesso diversi da una struttura all'altra. Nonostante fosse importante far emergere la difformità inizialmente presente, questa è stata successivamente risolta grazie alla creazione di una nuova API appositamente sviluppata per comunicare gli orari delle strutture in un formato più adatto alle diverse esigenze del front-end;
- l'assenza, lato back-end, di informazioni circa il regolamento e i servizi delle strutture. Viene fornita una stringa, descrittiva della struttura che, variando da struttura a struttura, a volte contiene informazioni sul regolamento e sui servizi, a volte no.

L'unica difformità riscontrata all'interno della pagina Saletta è l'assenza, lato back-end, di informazioni circa la strumentazione della saletta. Vale il discorso fatto poc'anzi per il regolamento e i servizi della struttura.

4.2.2 Valutazione euristica dei mockup iniziali

All'interno della presente sottosezione si applicherà l'approccio della *Valutazione euristica* ai mockup del 2021. Anche in questo caso, similmente a quanto fatto nella Sezione 3.2, opererò seguendo una versione "semplificata" del processo di Valutazione euristica. La valutazione sarà, ancora una volta, condotta da me in solitaria, consapevole dei limiti che comporta l'esecuzione di una valutazione in autonomia. I mockup sono semplici schermate. La procedura seguita per identificare le violazioni è consistita principalmente nell'osservazione di tali schermate.

Il dettaglio delle singole violazioni rilevate è disponibile all'appendice B, in calce al presente documento. Questa sottosezione vuole riassumere i risultati ottenuti descrivendoli all'interno delle tabelle 4.1 e 4.2.

La Tabella 4.1 fornisce una descrizione dettagliata del numero di violazioni rilevate per ciascuna euristica, ed il relativo impatto, in percentuale, sul totale. Sono state rilevate un totale di 29 violazioni, distribuite come segue:

Euristica violata	**Numero di violazioni**	**Percentuale**
H1	2	6,90%
H2	3	10,34%
H3	0	0,00%
H4	12	41,38%
H5	2	6,90%
H6	3	10,34%
H7	0	0,00%
H8	6	20,69%
H9	0	0,00%
H10	1	3,45%
Totale	**29**	**100,00%**

Tabella 4.1: Distribuzione violazioni euristiche dei mockup

L'euristica che risulta essere la più violata è *H4 Consistency and standards*, con un totale di 12 violazioni. Questo numero costituisce il 41,38% del totale delle violazioni.

Considerando invece le severità associate a ciascuna violazione, la Tabella 4.2 illustra, per ogni euristica, il numero di violazioni per ciascun grado di severità e la media ponderata del grado di severità delle violazioni:

Euristica violata	**N° violazioni per severità**					**Media ponderata**
	0	**1**	**2**	**3**	**4**	**severità**
H1	0	0	1	1	0	2,50
H2	0	0	0	3	0	3,00
H3	0	0	0	0	0	0,00
H4	0	1	10	1	0	3,00
H5	0	0	0	2	0	3,00
H6	0	0	2	1	0	2,33
H7	0	0	0	0	0	0,00
H8	0	1	4	1	0	2,00
H9	0	0	0	0	0	0,00
H10	0	0	0	1	0	3,00
Totale	**0**	**2**	**17**	**10**	**0**	**2,28**

Tabella 4.2: Severità associate alle violazioni euristiche dei mockup

Il valore più alto tra le diverse medie ponderate è 3,00. A questo valore si assestano le euristiche H2, H4, H5 e H10. Questo valore supera del 31,58% la media ponderata complessiva, che si attesta, invece, a 2,28. Il grado di severità più comunemente riscontrato è il 2, con un totale di 17 violazioni a esso collegate.

Rispetto ai risultati ottenuti dalla valutazione euristica applicata al portale attuale (Sottosezione 3.2.2), si nota un generale miglioramento:

- il numero delle violazioni riscontrate passa da 101 a 29;
- la media ponderata complessiva del grado di severità delle violazioni passa da 2,59 a 2,28;
- il grado di severità più comunemente riscontrato passa da 3 al 2.

Nel complesso non si può che lodare il lavoro svolto dal team di Musictogo per provare a rinnovare il portale.

4.3 Aggiornamento dei mockup

Come accennato nelle precedenti sezioni, si è reso necessario un lavoro di aggiornamento dei mockup. Le successive sottosezioni descriveranno, nel dettaglio di ogni pagina, la fase di aggiornamento dei mockup. Essa consisterà nella revisione degli elementi presenti nei mockup iniziali e nella creazione, talvolta, di nuovi elementi, dei quali i mockup erano sprovvisti. Si punterà a risolvere le problematiche emerse nelle sottosezioni 4.2.1 e 4.2.2. Sarà inoltre necessario tenere conto dei risultati ottenuti dalla valutazione euristica fatta nella Sezione 3.2, per assicurare che la nuova interfaccia non cada vittima degli stessi problemi del portale attuale.

I successivi mockup includeranno, all'interno del footer, un ulteriore voce rispetto ai mockup iniziali: tale aggiunta nasce dall'esigenza di consentire il cambio della lingua del sistema, da italiano a inglese e viceversa.

4.3.1 Navbar Mobile

La Navbar, nella sua versione Mobile, non era descritta dai mockup iniziali; si è reso dunque necessario crearne una. La Navbar Mobile, illustrata in Figura 4.8, è stata progettata seguendo lo stile, comunemente utilizzato in ambito Mobile, dell'*Offcanvas Menu*.

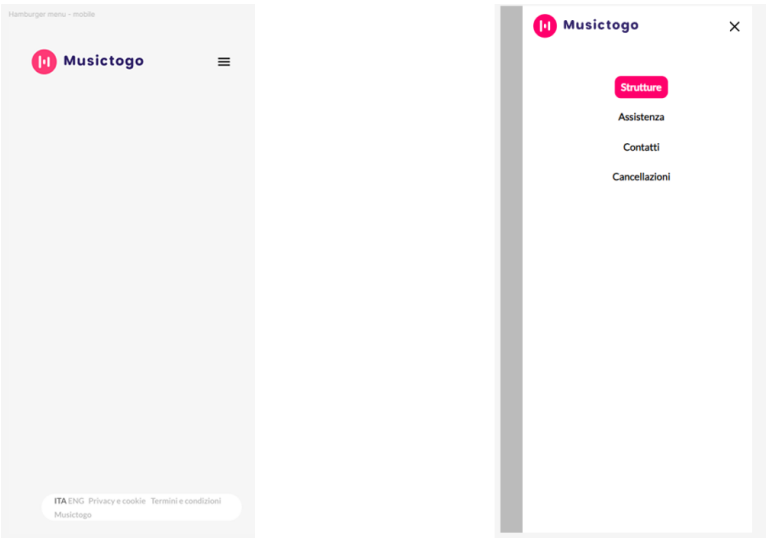

Figura 4.8: Navbar Mobile, mockup aggiornati

4.3.2 Homepage

La pagina, illustrata in Figura 4.9, mantiene a grandi linee l'aspetto che aveva nei mockup iniziali.

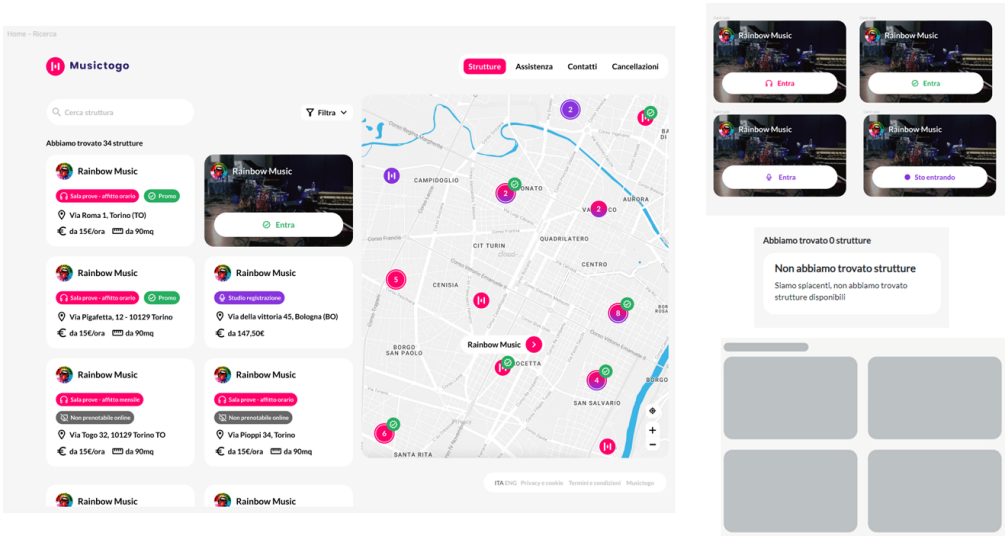

Figura 4.9: Homepage, mockup aggiornati

Le principali modifiche apportate ai mockup iniziali di questa pagina sono:

- *Modifiche all'aspetto delle card.* Ora le card hanno tutte lo stesso colore di sfondo, presentano (tramite tag) indicazioni chiare circa il tipo di struttura che rappresentano e l'eventuale stato di promozione o non prenotabilità della struttura. Le card sono ora cliccabili nella loro interezza. Si è deciso, per rappresentare gli stati "over" e "active", di aggiungere dell'ombra alla card per sopraelevarla e di mantenere lo stile proposto dai mockup iniziali seppur apportando alcune modifiche stilistiche, evidenziate in figura. Le informazioni quali il prezzo e i metri quadri sono testualmente meglio specificate. Non avendo a disposizione l'informazione sulla città della struttura, ma il solo indirizzo completo, si è preferito spostare tale dato su un nuovo rigo dedicato. Lo stato over, inoltre, nella visualizzazione Desktop fa apparire sulla mappa l'info-window del marker relativo alla struttura descritta nella card, in modo da localizzare istantaneamente la struttura.

- *Modifiche agli elementi visibili in lista.* Sono stati rimossi dalla lista tutti gli elementi che non rappresentano strutture. In particolare, la card precedentemente denominata "Non trovi quello che cerchi?" è stata ridefinita in una card più semplice visibile solo se la ricerca di strutture non porta risultati. Sono

state create delle card di tipo placeholder, da fa apparire durante i caricamenti della lista a seguito dell'applicazione dei filtri.

- *Modifiche ai componenti di filtraggio.* Si è preferito inserire un pulsante denominato "Filtri", che permetta di aprire un menu di gestione; l'indicazione sul numero di strutture trovate è stata spostata a ridosso della lista.

- *Modifiche alla mappa.* Sono stati aggiunti i pulsanti per fare zoom-in, zoom-out e per localizzare l'utente. Sono stati inseriti dei "cluster", ovvero degli elementi grafici che illustrano, in maniera compatta, la presenza, in una determinata zona, di più strutture, tramite dei marker che indicano al loro interno il numero di strutture agglomerate in ogni specifico cluster. I cluster sono particolarmente utili per scongiurare al minimo spiacevoli sovrapposizioni tra marker di strutture vicine, consentendo così al fruitore di avere sempre una chiara visione di ciò che è il contenuto della mappa. I cluster possono presentarsi in diverse varianti, ognuna rappresentativa dei tipi di strutture che agglomera al suo interno. Per consistenza con le icone circolari dei cluster, anche le icone delle strutture sono state modificate in una forma rotonda.

Le carenze dei mockup iniziali di questa pagina riguardano l'assenza di un componente che gestisca l'attivazione e la disattivazione dei filtri. La soluzione proposta è illustrata in Figura 4.10.

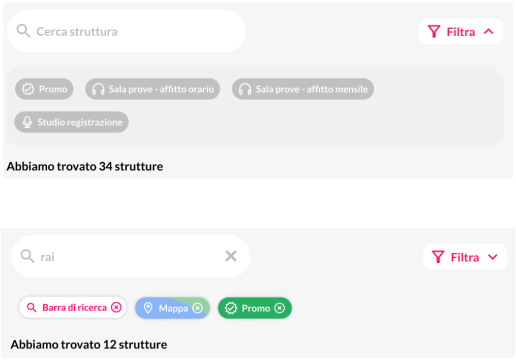

Figura 4.10: Componente filtri, Homepage, mockup aggiornati

Essa si compone di un piccolo menu la cui apertura avviene in seguito al click del pulsante "Filtri". Il menu consente di filtrare le strutture per tipo e per promo. I filtri del menu sono tra loro sovrapponibili, vale a dire che è possibile applicarne più di uno allo stesso momento, a differenza di quanto avviene nel portale attuale. A menu chiuso, gli eventuali filtri attivi sono tutti raggruppati nella stesso zona di schermo, tra la barra di ricerca e la lista delle strutture. La rimozione di un filtro attivo avviene tramite il click sul relativo tag. Viene considerato "filtro" qualsiasi

elemento della pagina che consenta di far variare il numero di strutture visualizzate nella lista. La barra di ricerca può essere, per tanto, considerata un filtro. Si prevede che anche la mappa offra la possibilità di filtrare la lista delle strutture in base allo stato di visibilità dei suoi singoli marker, che potrà variare a seguito dell'interazione dell'utente con la mappa stessa. La mappa punta a sostituire il famigerato filtro per province con un sistema di filtraggio geografico più semplice e intuitivo. Essendo, dunque, a tutti gli effetti, dei filtri, la barra di ricerca e la mappa avranno il proprio tag, come illustrato nella precedente figura. Si prevede inoltre, al fine di migliorare l'esperienza di navigazione dell'utente tra le diverse pagine, di conservare lo stato dei filtri durante la navigazione dalla Homepage alle altre pagine. In questo modo, se l'utente dovesse tornare indietro alla Homepage, troverebbe l'applicazione nello stesso stato in cui l'aveva lasciata.

Tra i vari filtri, sarebbe stato interessante averne uno che avesse potuto maneggiare gli orari di apertura delle strutture. Al momento dell'aggiornamento dei mockup, però, il back-end di Musictogo non esponeva API che descrivessero gli orari di apertura delle strutture in un formato "malleabile" attraverso il quale sarebbe stato possibile eseguire operazioni di filtraggio, ma esponeva solo una stringa il cui contenuto erano gli orari di apertura, descritti in formati diversi da struttura in struttura, e dunque non utilizzabili come input di un eventuale filtro. Tale temporanea situazione ha scoraggiato la creazione del filtro orario, facendo sì che gli sforzi si concentrassero su altro. Gli orari sono stati mantenuti con una semplice stringa fino a quando, per necessità dovute all'implementazione reale della procedura di prenotazione, sono stati inseriti in una nuova api in un formato consono. Per questioni di tempo non è stato possibile rivalutare il mockup per aggiungere il nuovo filtro; si è preferito piuttosto completare il lavoro e lasciarlo come possibile sviluppo futuro.

Homepage dei mockup aggiornati - versione Mobile

La versione Mobile, illustrata nelle sue varie schermate in Figura 4.11, è stata del tutto rivisitata rispetto a come si presentava nei mockup iniziali.

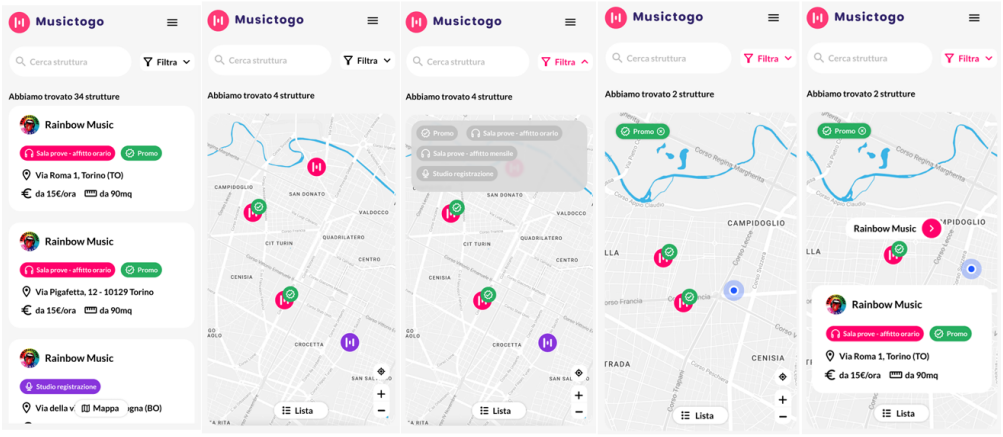

Figura 4.11: Homepage, mockup aggiornati Mobile

Allo stato attuale, essa risulta coerente con la sua controparte Desktop. La pagina consente, tramite un pulsante posto in basso, lo switch tra la lista delle strutture e la mappa. Particolarmente interessante è quest'ultima, in quanto integra bene al suo interno il menu dei filtri e, all'occorrenza, la card della struttura il cui marker è stato cliccato.

4.3.3 Pagina Struttura

Le pagine delle strutture mantengono, a grandi linee, l'aspetto che avevano nei mockup iniziali. Se ne distinguono tre varianti: *Sala prove ad affitto orario* (Figura 4.12), *Studio di registrazione* (Figura 4.13), *Sala prove ad affitto mensile* (Figura 4.14).

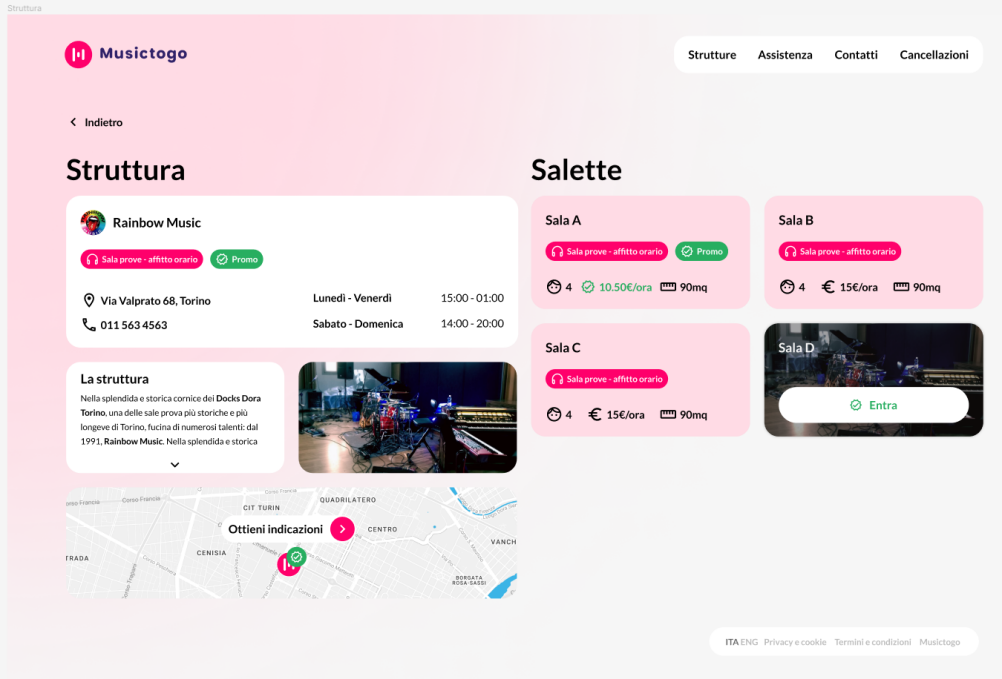

Figura 4.12: pagina Sala prove ad affitto orario, mockup aggiornati

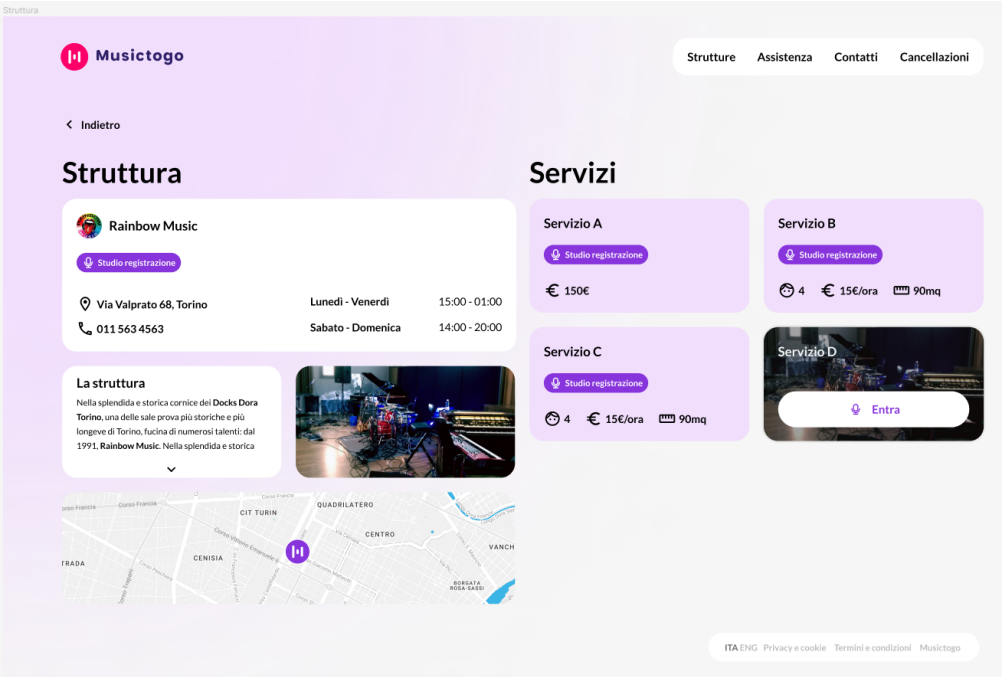

Figura 4.13: pagina Studio di registrazione, mockup aggiornati

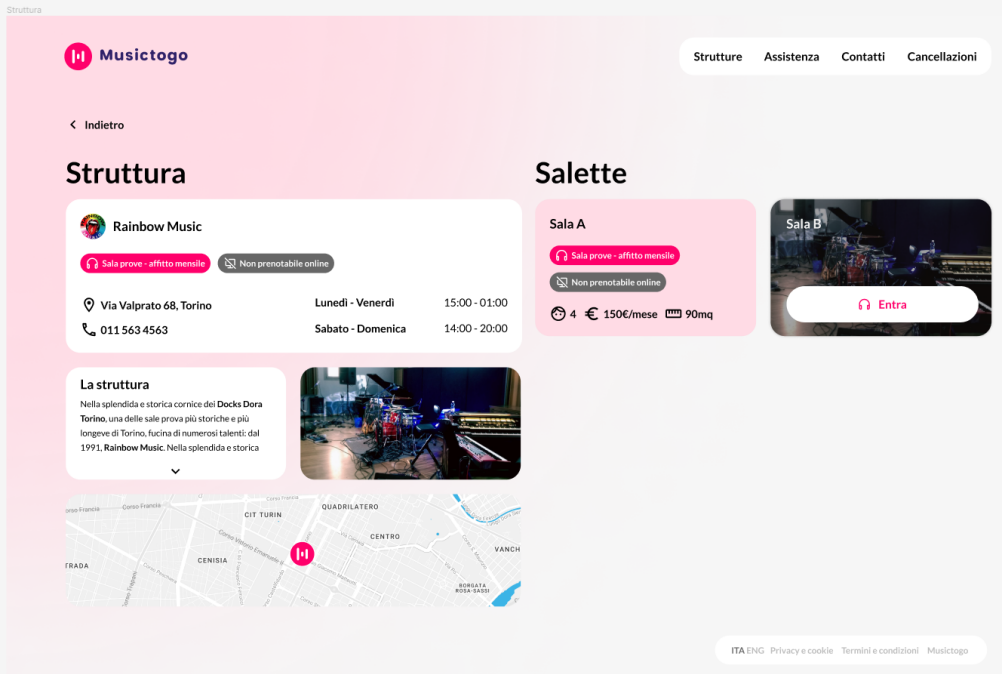

Figura 4.14: pagina Sala prove ad affitto mensile, mockup aggiornati

Il layout sul quale si basano le pagine dalle tre varianti è sempre il medesimo. Le principali differenze tra di esse si riscontrano nel colore dell'effetto di sfondo (rossiccio per le sale prove, sul lilla per gli studi di registrazione) e nel titolo della colonna di destra ("Salette" per le sale prove e "Servizi" per gli studi di registrazione). Dato il basso numero di differenze, è possibile dunque elencare le modifiche apportate al mockup iniziale avendo come riferimento il layout di base della pagina aggiornata. Nel dettaglio, le principali modifiche apportate ai mockup iniziali di questa pagina sono:

- *Rimozione di alcuni elementi.* Sono state rimosse le card "Servizi" e "Regolamento" in quanto tali informazioni non sono disponibili a back-end.

- *Modifiche alla card descrittiva della struttura.* Si è seguito lo stesso stile, basato sui tag, delle card della Homepage: le caratteristiche della struttura, ovvero tipo e stato, sono state descritte tramite tali tag. Sono state mantenute con lo stile originale le informazioni sull'indirizzo, il telefono e gli orari di apertura della struttura. È previsto che l'indirizzo e il numero di telefono siano cliccabili per agevolare l'accesso a Google Maps [12] e al dialer telefonico. È stata invece rimossa l'informazione del sito web, in quanto non disponibile a back-end.

- *Modifiche alla card "La struttura".* La card ha ora uno sfondo bianco che la uniforma alla vicina card descrittiva. È resa, all'occorrenza, espandibile: questa caratteristica permette di evitare che la card, di default, se il suo contenuto è molto lungo, risulti molto più grande in altezza rispetto agli altri componenti della pagina.

- *Modifiche alla mappa.* Il marker è stato sostituito con la sua nuova versione circolare. Inoltre, è stato reso cliccabile. Cliccando sul marker, verrà visualizzata una finestra informativa (info-window) che, se cliccata, aprirà Google Maps [12], impostando la navigazione sulla struttura.

- *Modifiche alla lista di card salette/servizi.* Le card di salette/servizi sono state modificate per farle apparire in uno stile che risultasse coerente con quello delle card delle strutture presenti in Homepage. Le card adesso espongono in modo chiaro il prezzo finale delle salette in promozione. È stata rimossa la card "Prenotazione", ritenuta, in questa pagina, superflua.

Pagina Struttura dei mockup aggiornati - versione Mobile

La versione Mobile della pagina, illustrata in Figura 4.15 nella variante "Studio di registrazione", è stata del tutto rivisitata rispetto a come si presentava nei mockup iniziali.

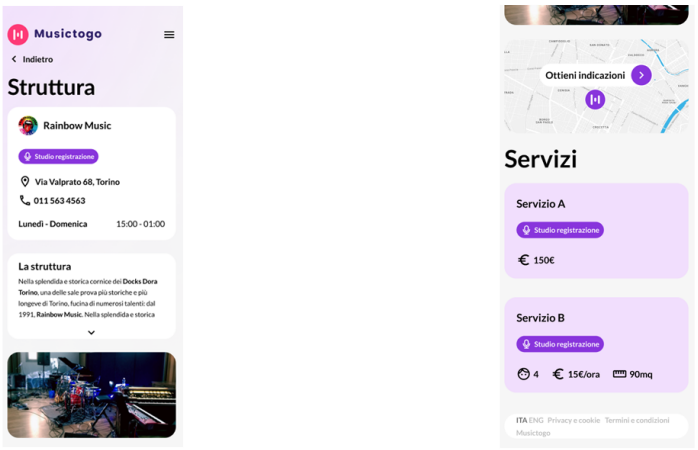

Figura 4.15: Pagina Struttura, mockup aggiornati Mobile

La versione Mobile rappresenta una riorganizzazione degli elementi presenti nella versione Desktop, opportunamente adattati per aderire alle dimensioni dello

schermo dei dispositivi cellulari, non si reputano pertanto necessarie ulteriori spiegazioni circa questa pagina.

4.3.4 Pagina Saletta

La pagina dedicata alle singole salette, illustrata in Figura 4.16, è stata la più impegnativa da aggiornare. Anche il layout di questa pagina rispecchia quello proposto nei mockup iniziali, con una colonna sinistra che fornisce dettagli sulla saletta e una colonna destra che gestisce la procedura di prenotazione. I componenti della colonna sinistra di questa pagina sono stati aggiornati per renderli coerenti con le controparti della pagina Struttura. Per quanto riguarda, invece, la colonna destra, è stato necessario ideare un sistema che permettesse la selezione di data e slot orari, l'inserimento dei dati utente, la verifica dell'Email e la successiva prenotazione, tutto all'interno dello stesso componente. La soluzione proposta nei mockup aggiornati consiste in una card dotata di *breadcrumb navigation*. In altre parole, questa card consente di eseguire al suo interno una procedura articolata in più step, dove l'avanzamento allo step successivo è concesso solo quando vengono soddisfatte specifiche condizioni nello step corrente. Nel dettaglio, la procedura di prenotazione di una saletta verrà divisa in tre step:

1. selezione della data e degli slot orari desiderati;
2. inserimento dei dati personali e selezione della modalità di pagamento desiderata;
3. verifica dell'indirizzo email inserito nel precedente step.

In seguito al completamento del terzo step, a seconda della modalità di pagamento scelta, l'utente verrà reindirizzato sulla pagina di riepilogo della prenotazione o su un servizio terzo per effettuare il pagamento. La suddivisione in più step della procedura di prenotazione consente all'utente di avere una visione chiara delle diverse fasi necessarie per completare il processo. Questo approccio offre inoltre all'utente maggiore flessibilità e controllo, contribuendo a rendere l'intero processo di prenotazione più efficiente.

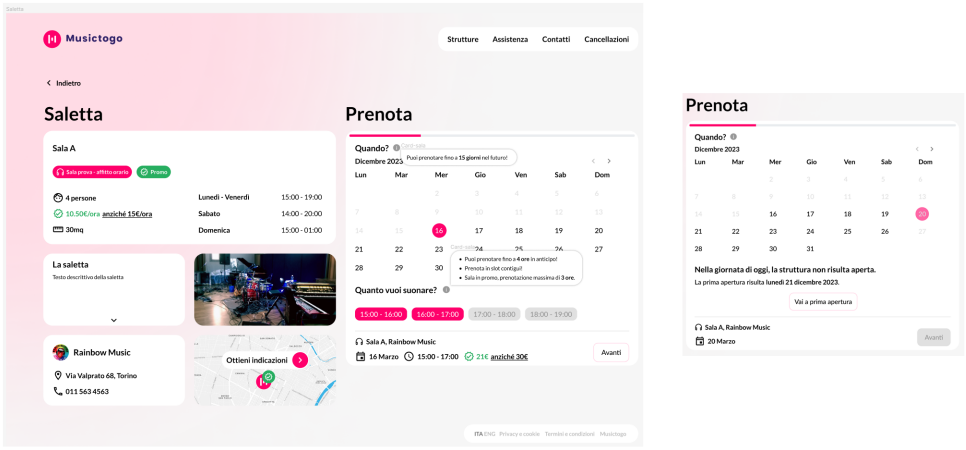

Figura 4.16: Pagina Saletta, mockup aggiornati

Le principali modifiche apportate ai mockup iniziali di questa pagina sono:

- *Modifiche alla struttura delle informazioni.* Nei mockup iniziali, le informazioni sulla saletta erano distribuite tra il titolo della colonna di sinistra, il suo occhiello e la card descrittiva. Nei mockup aggiornati, queste informazioni sono state condensate all'interno della card descrittiva, sostituendo il titolo della pagina con il più generico "Saletta", in linea con quello della pagina Struttura.

- *Modifiche alla card descrittiva della saletta.* Si è seguito lo stesso stile, basato sui tag, della card descrittiva di pagina Struttura: la parte superiore della card contiene ora il nome della saletta e le sue caratteristiche quali tipo e stato, descritte tramite tali tag. Sono state mantenute con lo stile originale le informazioni sul numero di persone, il costo, i metri quadri e gli orari di apertura. La card espone in modo chiaro il prezzo finale delle saletta. Nel caso in cui la saletta sia in promozione, viene visualizzato anche il prezzo originale non scontato, al fine di fornire all'utente un'idea di quanto stia risparmiando.

- *Rimozione di alcuni elementi.* Sono state rimosse le card "La struttura" e "Strumentazione": la prima perché fuori contesto, la seconda in quanto tali informazioni non sono disponibili a back-end.

- *Aggiunta di nuovi elementi.* Sono state create due nuove card. La prima, denominata *La saletta*, è una copia speculare della card "La struttura" di pagina Struttura, contenente informazioni generiche sulla saletta e, all'occorrenza,

espandibile. La seconda, una versione "mini" della card descrittiva della struttura, riassume le informazioni essenziali relative alla struttura per la quale si sta prenotando la saletta; in particolare, essa include il nome della struttura, l'indirizzo e il numero di telefono. È previsto che l'indirizzo e il numero di telefono siano cliccabili per agevolare l'accesso a Google Maps [12] e al dialer telefonico.

- *Modifiche alla mappa.* Il marker è stato sostituito con la sua nuova versione circolare. Inoltre, è stato reso cliccabile. Cliccando sul marker, viene visualizzata una finestra informativa (info-window) che, se cliccata, apre Google Maps [12], impostando la navigazione sulla struttura.

- *Modifiche ai componenti di prenotazione.* Come precedentemente accennato, si è deciso di gestire il processo di prenotazione all'interno di una singola card, tramite una procedura a più step. Di conseguenza, sono state eliminate la card di riepilogo e la nota verde descrittiva della promozione, originariamente presenti nei mockup iniziali. I componenti precedentemente contenuti nella card di riepilogo sono stati invece mantenuti e trasferiti all'interno della card di prenotazione. La parte superiore della card ospita una *progess bar*, concepita per fornire un feedback visivo sullo stato di avanzamento della procedura. I diversi step della procedura sono accessibili tramite i pulsanti "Avanti" e "Indietro", posti nella parte inferiore della card. Nel dettaglio, il pulsante "Indietro" è disponibile a partire dal secondo step, mentre il pulsante "Avanti" viene abilitato per procedere allo step successivo solo quando vengono soddisfatte specifiche condizioni nello step corrente. All'interno della card, nel primo step sono state aggiunte delle *info-icon* accanto ai titoli "Quando?" e "Quanto vuoi suonare". Posizionando il cursore del mouse su queste info-icon, vengono visualizzati dei tooltip contenenti, tra le altre informazioni, i dettagli che erano originariamente presenti nella card "Prenotazione" della pagina Struttura. La scelta di spostare tali informazioni qui è dettata dalla rilevanza che esse hanno nella procedura di prenotazione.

Le carenze dei mockup iniziali di questa pagina riguardano, come presumibilmente è ormai ben noto, l'assenza di dettagli sul come deve essere progettata l'intera procedura di prenotazione. Come precedentemente accennato, è stata proposta una procedura a più step.

Il primo step, come precedentemente illustrato in Figura 4.16 consiste nella scelta della data e degli slot orari. Il pulsante "Avanti" verrà abilitato dal sistema solo dopo che l'utente avrà selezionato almeno uno slot. Si prevede che il calendario sia navigabile solo nei mesi contenenti date prenotabili e che disabiliti le date corrispondenti a giorni della settimana in cui la struttura è chiusa. Il calendario partirà sempre dalla data odierna. Tale scelta è dettata dalla volontà di avere

sempre lo stesso comportamento iniziale e dalla necessità di creare una forte corrispondenza tra il sistema ed il mondo reale. Il caso limite è che la struttura sia chiusa proprio nella data odierna: in questo caso si preferisce comunque mantenere il calendario su tale data, e non spostarlo sulla prima apertura disponibile, per evitare che gli utenti più distratti prenotino una data futura pensando, abituati al comportamento di default, di prenotare per la giornata odierna: in questo caso speciale, come evidenziato in Figura 4.16, viene mostrato un messaggio d'avviso e un pulsante che, se cliccato, permette di visualizzare gli slot della prima apertura disponibile. Si prevede inoltre di mantenere cliccabili anche i giorni in cui tutti gli slot risultano già occupati da altri utenti. Questo permette di comunicare chiaramente che la struttura è aperta, evitando che gli utenti si facciano strane idee sul motivo per cui alcune date risultano disabilitate senza apparente motivo. Nella gestione degli slot orari, sarà invece fondamentale consentire solo la selezione di slot contigui. Sarà inoltre necessario gestire il caso in cui l'utente rimuova uno slot centrale, ad esempio l'ipotetico slot B, dopo aver selezionato gli slot contigui A, B e C. In questa situazione, si potrebbe prevedere che tutti gli slot successivi a quello deselezionato dall'utente vengano automaticamente deselezionati anch'essi.

Il secondo step, come illustrato in Figura 4.17, prevede l'inserimento dei dati dell'utente e la selezione della modalità di pagamento desiderata.

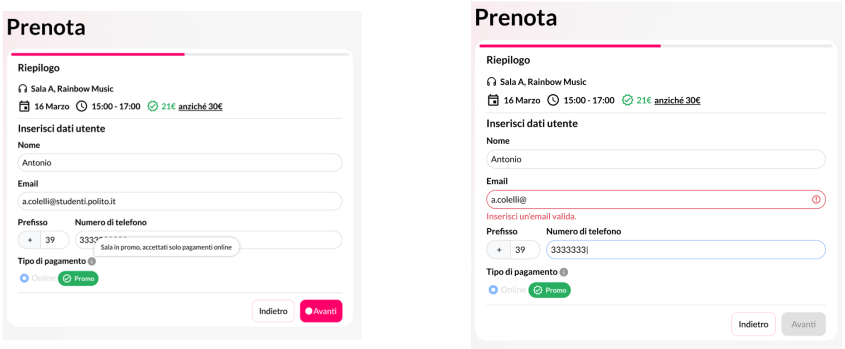

Figura 4.17: Secondo step breadcrumb, pagina Saletta, mockup aggiornati

La zona superiore della card mantiene informazioni che riepilogano quanto fatto nello step precedente. Il resto della card è invece dedicato all'inserimento dei dati dell'utente e alla selezione della modalità di pagamento desiderata. Ogni dato inserito dall'utente verrà validato in tempo reale. Sono previsti dei feedback visivi, illustrati anch'essi in figura, per informare l'utente nel caso in cui l'informazione inserita non sia stata scritta nel formato previsto dal sistema. Un aspetto rilevante riguardo l'esperienza utente (UX) è che, oltre a dover garantire al fruitore del servizio il flusso più agevole possibile, un'interfaccia deve anche venire incontro alle esigenze di business della società fornitrice di tale servizio. Ne sono un esempio i

radio button che permettono di selezionare la modalità di pagamento desiderata: nel caso in cui siano disponibili entrambe le modalità di pagamento, ovvero online e in contanti, il sistema seleziona di default il pagamento online, pur consentendo all'utente di cambiarlo a favore del pagamento in contanti; per le prenotazioni di salette in promozione, invece, Musictogo ha deciso di imporre il pagamento online come unico disponibile. Queste "forzature" nella UX del portale consentono a Musictogo, in qualità di azienda, di emettere un maggior numero di fatture. Il pulsante "Avanti" verrà abilitato dal sistema dopo che l'utente avrà inserito tutti i dati richiesti in un formato reputato corretto dal sistema stesso.

Il terzo step prevede la verifica dell'indirizzo di posta elettronica precedentemente indicato. Ciò avviene mediante l'inserimento, nell'apposito form, di un codice *OTP* inviato per mail a tale indirizzo. La procedura di verifica dell'indirizzo di posta elettronica del portale attuale si limita al semplice invio di una mail, seguita da, potenzialmente, infiniti tentativi di verifica. La procedura di verifica prevista per la nuova interfaccia, illustrata, nelle diverse schermate che la compongono, in Figura 4.18, si pone, per quanto possibile, l'obiettivo di essere più robusta rispetto alla sua controparte del portale attuale. La zona superiore della card mantiene informazioni che riepilogano quanto fatto negli step precedenti. Il resto della card è invece dedicato alla verifica dell'indirizzo email. Questo terzo step è, a tutti gli effetti, una procedura nella procedura, in quanto si è deciso, come precedentemente anticipato, di renderlo più robusto, limitando il numero di tentativi di validazione concessi all'utente. Si ritiene necessario guidare l'utente nelle diverse fasi che compongono la procedura. Il sistema necessiterà, dunque, di un'ampia varietà di schermate da poter mostrare, all'occorrenza, all'utente, per poterlo guidare al meglio. Per una rapida descrizione delle schermate presenti in Figura 4.18, si conviene di identificare ogni schermata mediante un'associazione riga-colonna: a puro scopo d'esempio, la terza schermata della seconda riga verrà identificata come "schermata 2C". La schermata **1A** rappresenta la prima schermata visualizzata quando si raggiunge il terzo step della procedura di prenotazione. Tramite la pressione del pulsante "Invia codice", l'utente fa inviare il codice OTP all'indirizzo email da verificare. Il pulsante è contornato da testo che spiega dettagliatamente cosa avverrà a seguito del click. La pressione del pulsante è seguita da una fase di attesa, illustrata dalla schermata **1B**; questa schermata tiene l'utente informato sull'operazione in corso nel sistema, ovvero l'invio del codice. Una volta inviato il codice, il sistema è pronto a mostrare all'utente la schermata **1C**, che offre un riepilogo delle azioni compiute fino a quel momento e spiega come trovare il codice inviato. Viene inoltre spiegato che il codice deve essere inserito nel form della schermata entro un determinato lasso di tempo, visibile sotto il form stesso. Il codice verrà verificato dal sistema a seguito della pressione del pulsante "Verifica codice". La pressione del pulsante è seguita da una fase di attesa, illustrata dalla schermata **2A**; questa schermata tiene l'utente informato sull'operazione in corso nel sistema, ovvero

la verifica del codice precedentemente inserito. Se il sistema ritiene che il codice inserito dall'utente sia valido, la procedura di verifica può considerarsi completata con successo. A questo punto, il sistema mostra, attraverso la schermata **2B**, un breve messaggio di conferma. A differenza di tutte le altre schermate della figura, il pulsante "Paga" non risulta disabilitato: è dunque possibile premerlo per essere reindirizzati alla schermata del servizio terzo attraverso il quale effettuare il pagamento. Invece, nel caso in cui il codice inserito fosse errato, viene mostrata la schermata **2C**, ovvero una variante della schermata 1C che illustra il caso in cui l'utente abbia tentato di verificare un codice errato. Per ogni codice inviato per email, si concedono tre tentativi di verifica (purché vengano effettuati all'interno dell'intervallo di validità del codice), dopo i quali la procedura di verifica OTP viene ritenuta fallita ed il codice viene invalidato. Dopo il terzo tentativo fallito, o dopo che l'OTP scade a seguito del termine dell'intervallo di validità, il sistema, ritenendo fallita la procedura di verifica, mostra la schermata **3A**; l'utente che visualizza questa schermata può riprovare la procedura di validazione premendo il pulsante "Invia un nuovo codice". Se l'utente fallisce per tre volte la procedura di verifica, il sistema visualizza la schermata **3B**; la comparsa di questa schermata invalida la procedura di prenotazione in toto. Se l'utente desidera avviare una nuova procedura di prenotazione, dovrà necessariamente ricaricare la pagina della saletta, perdendo tutto.

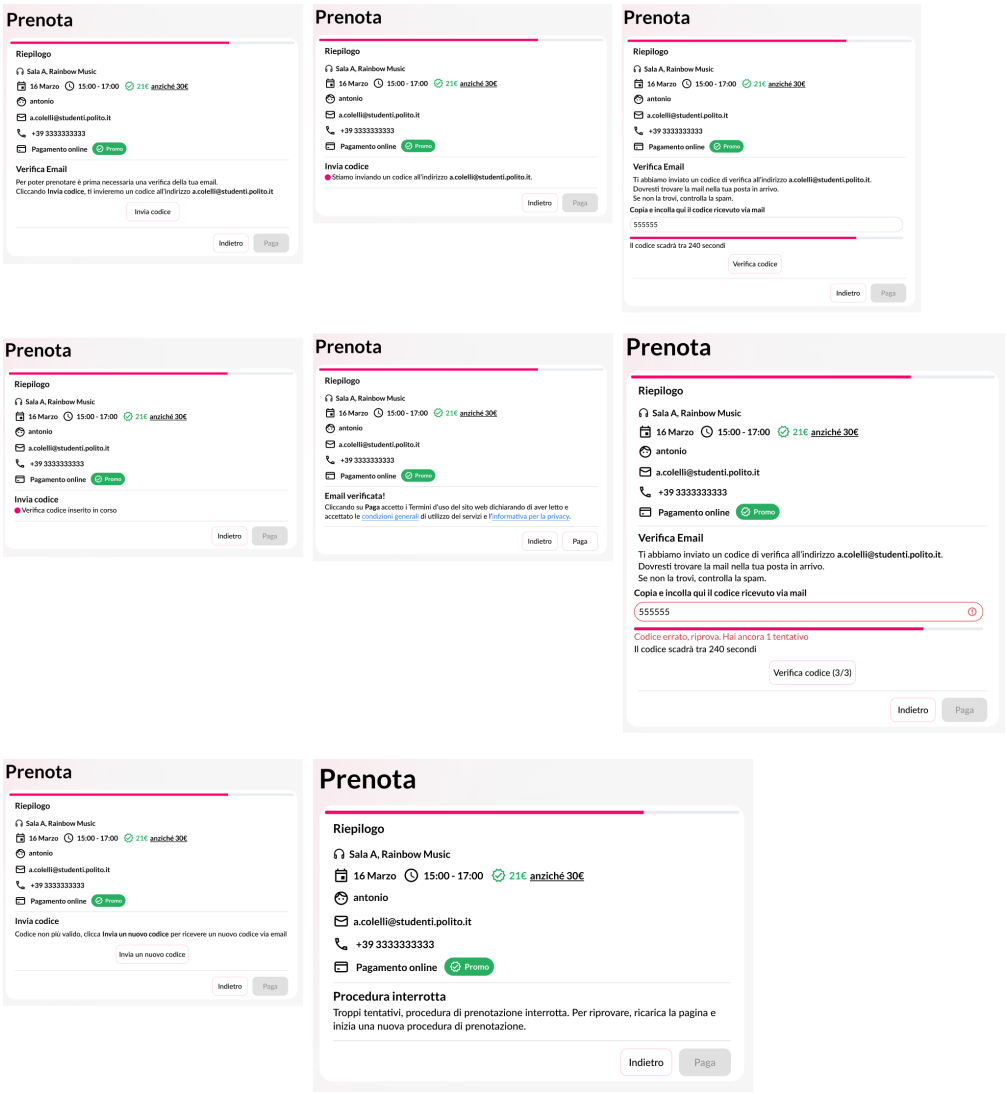

Figura 4.18: Terzo step breadcrumb, pagina Saletta, mockup aggiornati

Le schermate rappresentate in figura illustrano una procedura di prenotazione con pagamento online. Nel caso in cui invece il pagamento avvenisse in contanti, il pulsante avrà la dicitura “Prenota” e la sua pressione reindirizzerà l’utente alla pagina di riepilogo della prenotazione effettuata.

Si prevede che la pressione dei pulsanti “Avanti”,“Indietro”,“Paga”/“Prenota” scateni delle operazioni di controllo sui dati. Andrà controllato, cioè, che i parametri di prenotazione (data e slot orari) continuino ad essere compatibili con la quantità minima di minuti d’anticipo necessari per poter prenotare ogni specifico slot orario, e che gli slot orari selezionati non siano stati occupati, nel frattempo, da altri utenti

più veloci. Il sistema fornirà, mediante un *Grow Spinner* che verrà posizionato accanto al testo del pulsante cliccato, un feedback visivo per indicare che in background è in corso un operazione. Al verificarsi di una di queste due condizioni, si prevede di mostrare a schermo un modale come quello illustrato in Figura 4.19, che riporti forzatamente l'utente alla schermata di selezione degli slot orari per permettergli, qualora lo desideri, di selezionare nuovi slot.

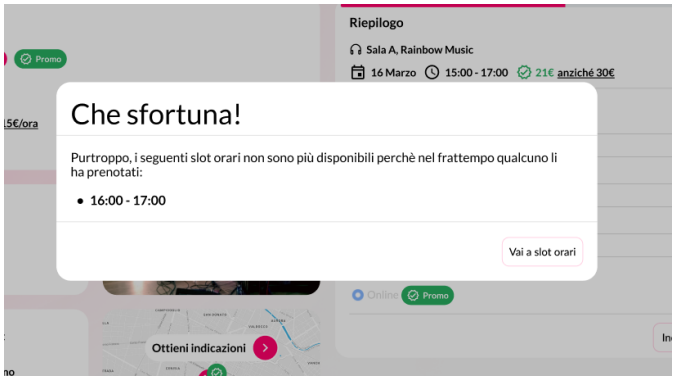

Figura 4.19: Modale slot occupati, pagina Saletta, mockup aggiornati

Si prevedono modali simili a quello precedentemente proposto anche nel caso in cui, per errori di rete, non si riesca a contattare il server di Musictogo per reperire le informazioni necessarie o per completare la procedura di prenotazione.

Sarà essenziale, per evitare di ricadere in uno dei principali problemi del portale attuale, garantire che l'utente, tornato, volontariamente, tramite la pressione del pulsante "Indietro", o forzatamente, a causa del verificarsi degli eventi poc'anzi descritti, in uno step precedente, non perda quanto aveva eventualmente fatto fino a quel momento nelle schermate successive. Pertanto, sarà importante salvare lo stato di validazione dell'email e i dati dell'utente. Inoltre, se l'utente ha già validato l'email, i suoi dati dovrebbero essere visibili in sola lettura, senza possibilità di modifica.

Pagina Saletta dei mockup aggiornati - versione Mobile

C'è poco da dire sulla versione Mobile di questa pagina, in quanto essa altro non è che il risultato di una riorganizzazione degli elementi presenti nella versione Desktop, opportunamente adattati per aderire alle dimensioni dello schermo dei dispositivi cellulari. La pagina, illustrata in figura 4.20, risulta del tutto rivisitata rispetto a come si presentava nei mockup iniziali ma non si reputano necessarie ulteriori spiegazioni dato che gli elementi che la compongono sono stati già ampiamente descritti in questa sottosezione.

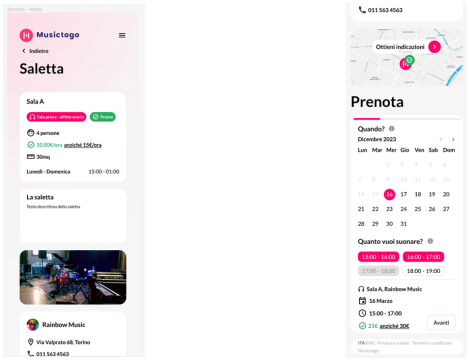

Figura 4.20: Pagina Saletta, mockup aggiornati Mobile

Pagina Saletta dei mockup aggiornati - variante "vetrina"

La pagina Saletta dispone anche della variante "vetrina", illustrata in figura 4.21.

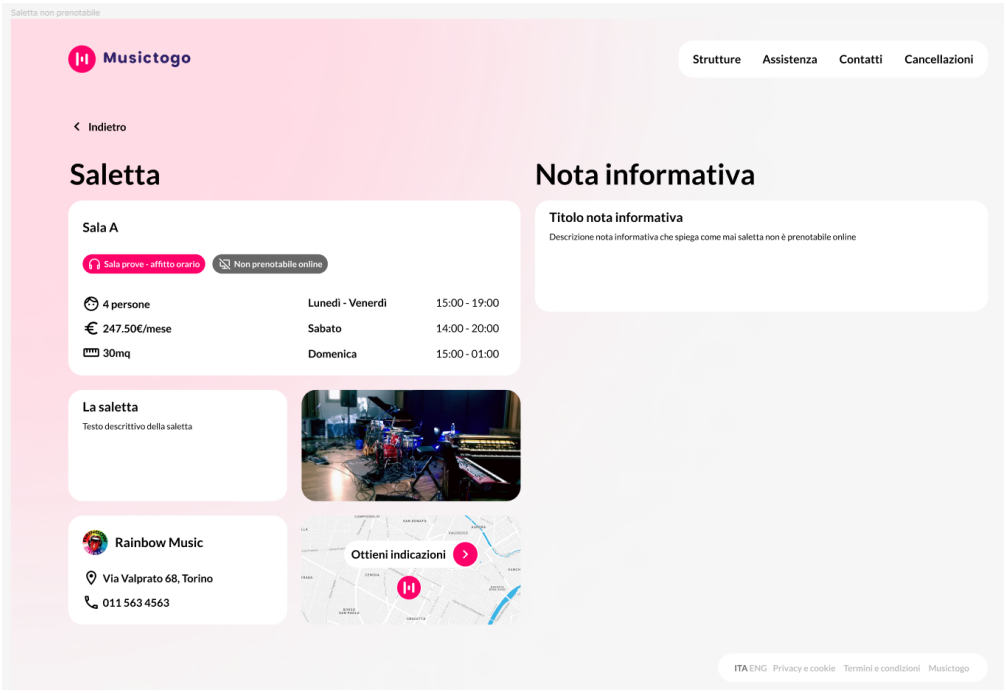

Figura 4.21: Pagina Saletta non prenotabile, mockup aggiornati

Questa variante della pagina consente di descrivere sia le salette di sale prove ad affitto mensile che le salette di alcune sale prove ad affitto orario che utilizzano Musictogo solo come vetrina.

Nella figura, la variante descrive, nello specifico, una saletta ad affitto orario; le uniche differenze che sarebbero emerse se avesse descritto una saletta ad affitto mensile sarebbero state la dicitura testuale all'interno del tag rosso ed, eventualmente, il testo all'interno della nota informativa. La principale differenza della variante rispetto alla versione di base della pagina è l'assenza, nella colonna destra, della card "Prenota", sostituita con la già citata card "Nota informativa", che spiega perché la saletta non è prenotabile online. Non si ritiene necessario illustrare e descrivere la versione Mobile di questa variante, in quanto essa sarà speculare alla rappresentazione illustrata in figura 4.20, se non per le differenze già illustrate nella descrizione della versione Desktop.

4.3.5 Pagina Servizio

La pagina dedicata ai singoli servizi degli studi di registrazione, illustrata in Figura 4.22, altro non è che una sorella della precedentemente descritta pagina Saletta, addobbata con un tema blu.

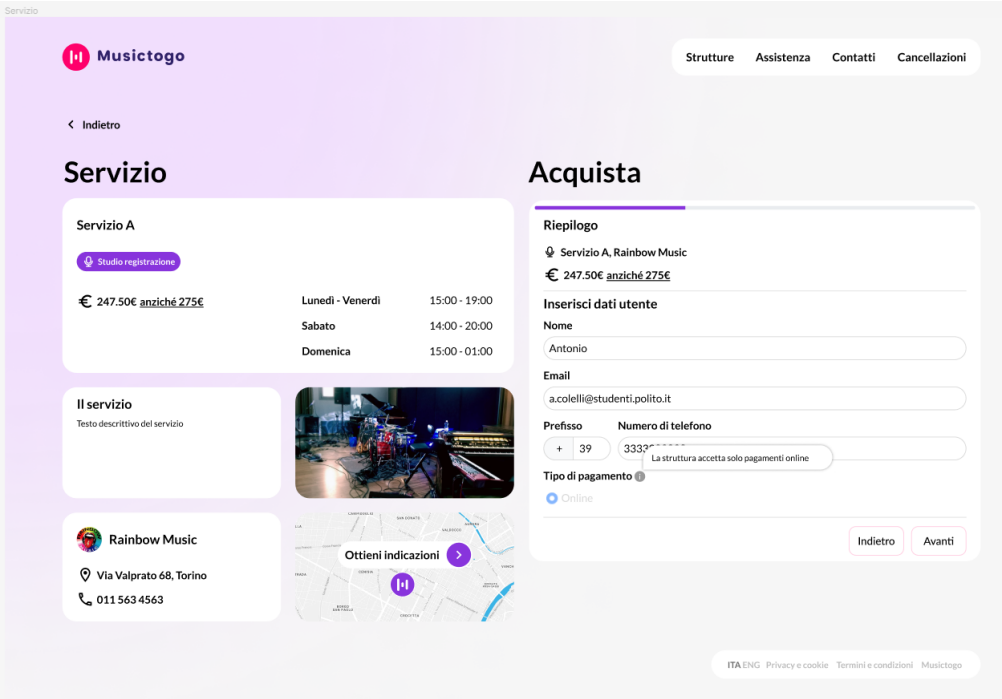

Figura 4.22: Pagina Servizio, mockup aggiornati

Essa presenta la medesima struttura a due colonne della sorella. Le due colonne vengono qui denominate "Servizio" e "Acquista". Nel dettaglio, all'interno della colonna "Servizio", l'unica differenza rilevante è l'assenza delle diciture "mq" e

"numero di persone" nella card descrittiva del servizio, poiché queste informazioni non sono ritenute rilevanti per i servizi.

La colonna destra presenta il medesimo componente di prenotazione a più step già illustrato in pagina Saletta, snellito dall'assenza del primo step di selezione di data e slot orari, dato che l'utente interessato a tale servizio dovrà semplicemente acquistare il relativo pacchetto e successivamente concordare con la struttura quando utilizzarlo.

Pagina Servizio dei mockup aggiornati - versione Mobile

Non si ritiene necessario illustrare e descrivere la versione Mobile di questa pagina, in quanto essa sarà speculare alla rappresentazione illustrata in figura 4.20, se non per le differenze già illustrate nella descrizione della versione Desktop.

4.3.6 Pagina Assistenza

I mockup iniziali non disponevano di una versione aggiornata della pagina Assistenza. Si è reso dunque necessario sviluppare un prototipo, illustrato in Figura 4.23, che risultasse coerente con lo stile dei mockup e che fungesse da aggiornamento della pagina Assistenza del portale attuale.

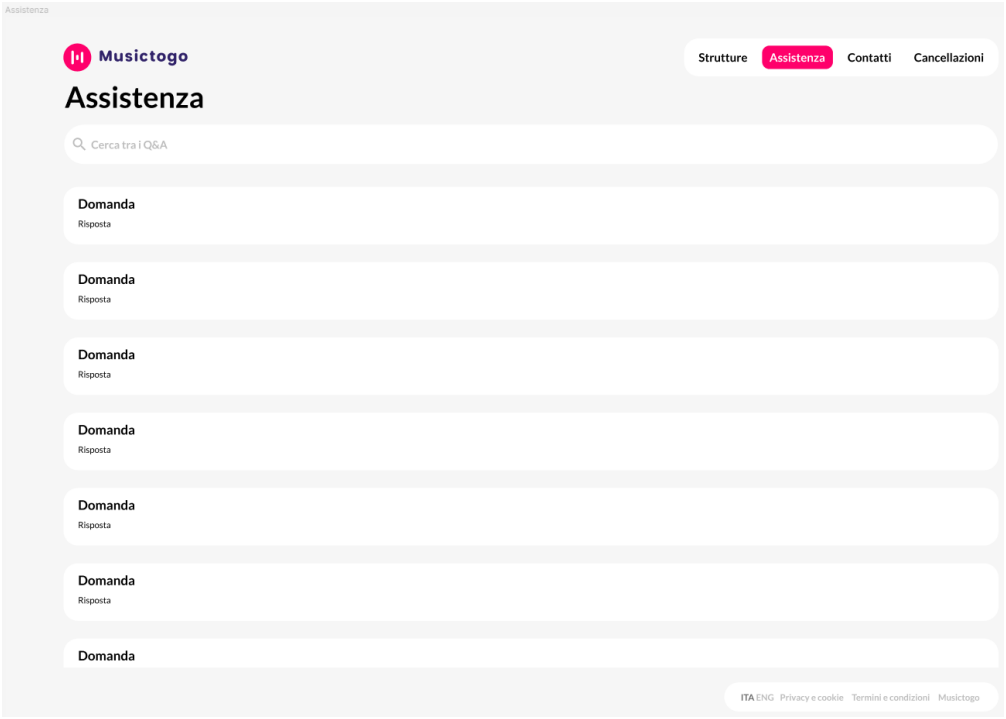

Figura 4.23: Pagina Assistenza, mockup aggiornati

Con riferimento alla pagina Assistenza del portale attuale (Figura 2.5), le singole Q&A della lista, coerentemente con lo stile adottato nei mockup, sono state inserite all'interno di card. Questo prototipo include, inoltre, una barra di ricerca, che consentirà agli utenti di filtrare la lista per trovare la Q&A più adatta alle proprie necessità.

Figura 4.24 illustra, invece, il caso in cui la ricerca non porti da alcun risultato.

Figura 4.24: Nessun risultato, pagina Assistenza, mockup aggiornati

Pagina Assistenza dei mockup aggiornati - versione Mobile

Non si ritiene necessario illustrare e descrivere la versione Mobile di questa pagina, in quanto essa altro non sarà che il risultato di una riorganizzazione degli elementi presenti nella versione Desktop, opportunamente adattati per aderire alle dimensioni dello schermo dei dispositivi cellulari.

4.3.7 Pagina Contatti

I mockup iniziali non disponevano di una versione aggiornata della pagina Contatti. Si è reso dunque necessario sviluppare un prototipo, illustrato in Figura 4.25, che risultasse coerente con lo stile dei mockup e che fungesse da aggiornamento della pagina Contatti del portale attuale.

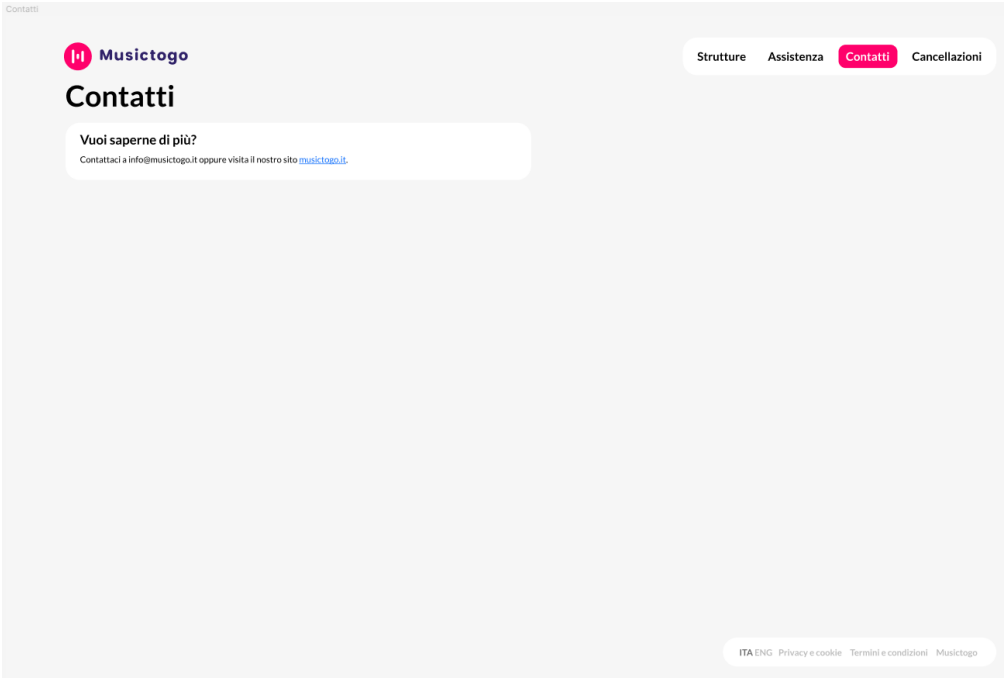

Figura 4.25: Pagina Contatti, mockup aggiornati

Con riferimento alla pagina Contatti del portale attuale (Figura 2.7), è stata creato, coerentemente con lo stile adottato nei mockup, un prototipo con un layout a due colonne. La colonna destra è, allo stato attuale, lasciata vuota. La colonna sinistra presenta invece, oltre al titolo della pagina, una card contenente informazioni di contatto.

Pagina Contatti dei mockup aggiornati - versione Mobile

Non si ritiene necessario illustrare e descrivere la versione Mobile di questa pagina, in quanto essa altro non sarà che il risultato di una riorganizzazione degli elementi presenti nella versione Desktop, opportunamente adattati per aderire alle dimensioni dello schermo dei dispositivi cellulari.

4.3.8 Pagina Cancellazioni

I mockup iniziali non disponevano di una versione aggiornata della pagina Cancellazioni. Si è reso dunque necessario sviluppare un prototipo, illustrato in Figura 4.26, che risultasse coerente con lo stile dei mockup e che fungesse da aggiornamento della pagina Cancellazioni del portale attuale.

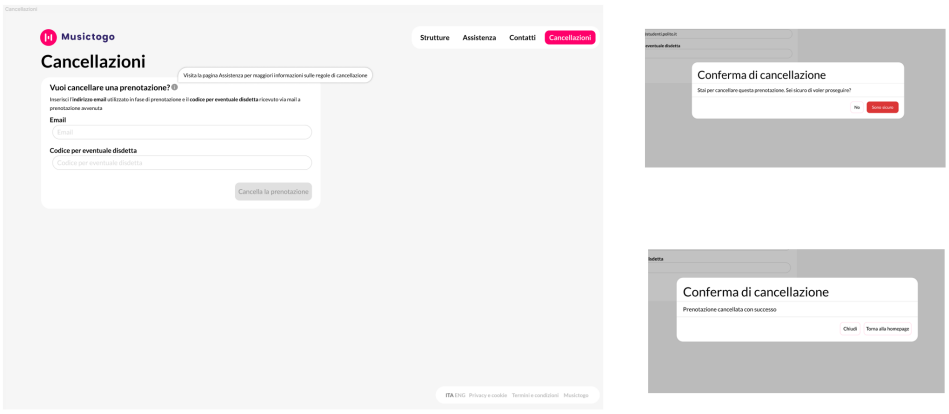

Figura 4.26: Pagina Cancellazioni, mockup aggiornati

Con riferimento alla pagina Cancellazioni del portale attuale (Figura 2.9), è stata creato, coerentemente con lo stile adottato nei mockup, un prototipo con un layout a due colonne. La colonna destra è, allo stato attuale, lasciata vuota. La colonna sinistra presenta invece, oltre al titolo della pagina, una card contenente un form per effettuare le operazioni di cancellazione. La card, rispetto alla sua controparte attuale, presenta un'info-icon situata nella parte superiore. Questa icona, quando puntata dal cursore del mouse, mostrerà a schermo un tooltip contenente del testo informativo. Inoltre, al suo interno, la card presenta un altro testo che sinteticamente spiega all'utente quali dati siano necessari per effettuare la cancellazione. Nel caso del codice di eventuale disdetta, viene anche indicato dove vada cercato. Il form dati, anche se non illustrato in figura, sarà dotato della stessa forma di validazione illustrata in Figura 4.17 e descritta nella Sottosezione 4.3.4. Il pulsante "Cancella la prenotazione" verrà abilitato dal sistema solo dopo che

l'utente avrà inserito i dati richiesti in un formato reputato corretto dal sistema stesso. Il sistema, data la natura "distruttiva" dell'operazione di cancellazione, dovrà procedere ad eseguirla solo a seguito di una conferma da parte dell'utente. Questa conferma sarà richiesta tramite il modale illustrato in alto a destra nella Figura 4.26. Una volta confermata, tramite la pressione del pulsante "Sono sicuro", l'intenzione di voler cancellare, dovrà avvenire un tentativo di cancellazione a seguito del quale il sistema fornirà, sia in caso di successo che di fallimento, un feedback mediante il modale illustrato in basso a destra nella Figura 4.26. Il sistema fornirà, mediante un *Grow Spinner* che verrà posizionato accanto al testo del pulsante "Cancella la prenotazione", un feedback visivo per indicare che in background è in corso l'operazione di cancellazione.

Pagina Cancellazioni dei mockup aggiornati - versione Mobile

Non si ritiene necessario illustrare e descrivere la versione Mobile di questa pagina, in quanto essa altro non sarà che il risultato di una riorganizzazione degli elementi presenti nella versione Desktop, opportunamente adattati per aderire alle dimensioni dello schermo dei dispositivi cellulari.

4.3.9 Altre schermate

Pagina conferma dei mockup aggiornati

Il prototipo della pagina è stato aggiornato aggiungendo, come illustrato in figura 4.27, il nome della saletta o del servizio prenotati.

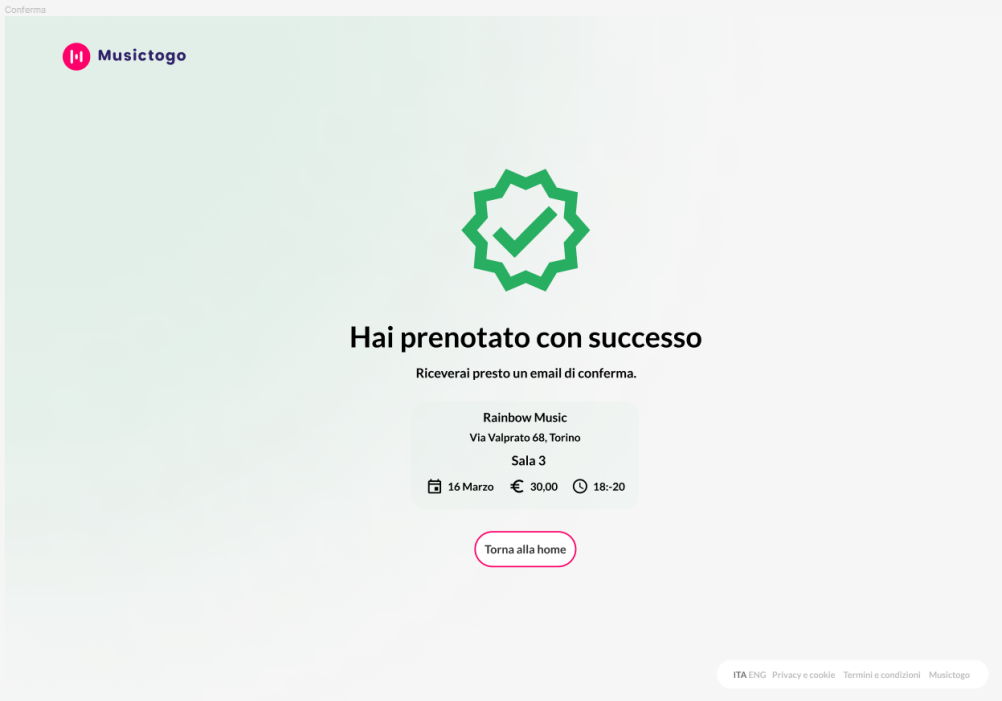

Figura 4.27: Pagina Conferma, mockup aggiornati

Pagine errori dei mockup aggiornati

Seguendo lo stile del prototipo di conferma, sono stati creati due ulteriori prototipi: la Figura 4.28 illustra due schermate che, all'occorrenza, possono essere mostrate all'utente:

- *Pagina 404.* Pagina che il sistema mostrerà qualora venga immesso un Url errato, scaduto, o non più funzionante.
- *Pagina Errore di rete.* Pagina che il sistema mostrerà qualora non fosse in grado di comunicare con il back-end.

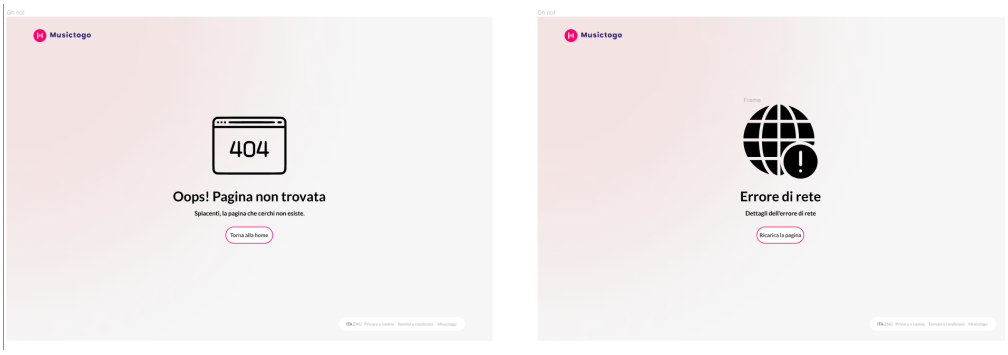

Figura 4.28: Pagine 404 e Errore di rete, mockup aggiornati

Versioni Mobile

Non si ritiene necessario illustrare e descrivere le versioni Mobile di queste pagine, in quanto esse altro non saranno che il risultato di una riorganizzazione degli elementi presenti nelle versioni Desktop, opportunamente adattati per aderire alle dimensioni dello schermo dei dispositivi cellulari.

Capitolo 5

Tecnologie utilizzate

Il portale di Booking attuale, ampiamente descritto e discusso nel Capitolo 2, è stato sviluppato dal team di Musictogo utilizzando la libreria React. Questo capitolo offre una panoramica sulle tecnologie disponibili per lo sviluppo di interfacce web, per poi entrare nello specifico delle tecnologie che si è deciso di impiegare per lo sviluppo di questo progetto.

5.1 Tecnologie disponibili

Per sviluppare pagine web, il programmatore potrebbe limitarsi ad utilizzare HTML, CSS e JavaScript:

- *HTML*. È il linguaggio di markup utilizzato per creare la struttura di base di una pagina web.
- *CSS*. È il linguaggio utilizzato per definire lo stile e l'aspetto di una pagina web.
- *JavaScript*. È il linguaggio di scripting che rende una pagina web interattiva.

Un tale approccio è, però, sempre meno frequente, in quanto richiede generalmente un grande sforzo anche per ottenere piccoli risultati. Per lo sviluppo di interfacce più complesse, può risultare pertanto conveniente utilizzare strumenti come framework e librerie che, attraverso l'uso di componenti di più alto livello, consentono di astrarre lo sviluppatore dall'utilizzo diretto di HTML, CSS e JavaScript in forma grezza. Questi strumenti hanno fin da subito riscosso un notevole successo tra gli sviluppatori.

Tra di essi, spiccano, per popolarità, Angular [13] , React [5] e Vue.js [14]. Questi strumenti sono tutti e tre basati su Javascript. Nel dettaglio [15]:

- *Angular*. È un framework open-source, sviluppato da Google, che fa uso del design pattern MVVM (Model View ViewModel) per l'organizzazione del codice. In Angular, il codice che rappresenta i diversi componenti viene diviso nei cosiddetti "chunks". Quando un utente visita una pagina, i chunk permettono al sistema di caricare solo il codice necessario per maneggiare quella determinata pagina. Angular è, però, anche noto per la sua ripida curva di apprendimento: richiede un grande investimento di tempo per poterlo padroneggiare al meglio. Inoltre, fornendo numerose funzionalità "out-of-the-box", produce bundle pesanti che generano applicazioni spesso più lente rispetto a quelle dei competitor.

- *React*. È una libreria open-source, sviluppata da Meta, per la creazione di interfacce utente, basata su un approccio dichiarativo e sull'utilizzo di componenti riutilizzabili. React implementa, al suo interno, un sistema denominato "Virtual DOM", che funge da intermediario tra lo stato interno dell'applicazione e il DOM reale del browser, consentendogli di gestire in modo efficiente le manipolazioni dirette sul DOM, migliorando così le prestazioni del sistema. Non essendo un framework, ma solo una libreria, React non definisce un pattern architetturale che il programmatore deve seguire durante la scrittura del codice, ma gli lascia totale libertà.

- *Vue.js*. È un framework progressivo open-source mantenuto dalla community. Ha in comune alcune caratteristiche con React, come il concetto di Virtual DOM e lo sviluppo di componenti riusabili. Risulta facile da imparare. Uno degli svantaggi di Vue è che, non avendo alle spalle una grossa azienda, si regge solo sulla community.

5.2 Libreria di sviluppo: React

Dopo aver esaminato le diverse alternative disponibili, si è deciso di sviluppare la nuova interfaccia utilizzando la libreria React. Il ruolo della libreria sarà quello di eseguire le istruzioni "a basso livello", come per esempio l'effettiva creazione degli elementi HTML, consentendo dunque di lavorare ad un più alto livello. Nel dettaglio, per la scrittura del codice, React prevede che gli sviluppatori utilizzino un approccio di tipo "dichiarativo": mentre il cosiddetto approccio "imperativo" prevede l'utilizzo di comandi espliciti per specificare ogni singolo step da eseguire in maniera ordinata (creazione l'elemento, poi settaggio il contenuto, poi aggiunta di un listener all'elemento e così via) per raggiungere un determinato obiettivo (nell'ambito di sviluppo di pagine web, banalmente, la creazione della pagina), la sua controparte, ovvero il precedentemente citato approccio dichiarativo, permette di descrivere cosa si voglia ottenere, quale sia l'obiettivo, senza entrare nei dettagli

su come ottenerlo. Entrando più nello specifico, React consente agli sviluppatori di definire gli oggetti da visualizzare a schermo e le loro funzionalità; la libreria si occuperà poi del resto. Questi oggetti prendono il nome di "componenti": un sistema sviluppato in React sarà composto da diversi componenti, ognuno dei quali avrà un suo specifico scopo; i componenti possono essere visti come dei blocchi di costruzione che vengono messi insieme dalla libreria. Lo sviluppatore non dovrà scrivere istruzioni step by step per creare e renderizzare gli elementi HTML ma, al contrario, definirà, tramite i componenti, cosa vuole ottenere come risultato finale e la logica che consentirà di definire quali componenti mostrare quando il sistema è in un determinato stato. React si occuperà, poi, di eseguire le istruzioni di basso livello per mettere insieme i componenti, descrivere la loro logica e renderizzarli a schermo tramite HTML, CSS e JavaScript [16].

5.2.1 Creazione di un progetto con React

Nei progetti sviluppati tramite l'approccio tradizionale, per ottenere pagine interattive e comprensibili al browser è sufficiente importare i file JavaScript dentro alle pagine HTML; React lavora diversamente. React prevede l'utilizzo della sintatti "HTML in JavaScript" per la creazione dei componenti. Essi vengono definiti all'interno di file con estensione JSX e possono, opzionalmente, ricevere dei parametri di input detti "props". Il browser non è però in grado di comprendere i file JSX. Si rende pertanto necessario che questi file subiscano un processo di build, che li trasformi in qualcosa di comprensibile al browser. Per questa ragione, i progetti sviluppati in React sono leggermente più complessi dei progetti tradizionali, e includono più file [16].

Per poter funzionare, React necessita di un tool denominato Node.js [17]: Node.js è un "JavaScript runtime environment", ovvero un tool che permette l'esecuzione di codice JavaScript al di fuori del browser. React farà uso di Node.js per generare il progetto e per trasformare, tramite il precedentemente citato processo di build, il codice sorgente in qualcosa di comprensibile al browser. Node.js si occuperà inoltre di eseguire il progetto tramite la creazione di un "development server" che ospiterà il progetto in locale [16].

5.3 Preprocessore CSS: SASS

SASS [1] [18] è un linguagio di fogli di stile preprocessato che estende il CSS tradizionale offrendo, tramite le varie sintassi che mette a disposizione degli sviluppatori, funzionalità aggiuntive per semplificare e migliorare la scrittura di codice CSS.

[1] SASS: acronimo di Syntactically Awesome Style Sheets

All'interno del progetto è stato utilizzato SASS, più nello specifico nella sua sintassi SCSS [2]. L'utilizzo di SASS ha consentito la scrittura di codice modulare e riutilizzabile. Sono state create delle variabili globali per definire colori, font, dimensioni e altre informazioni da rendere disponibili ovunque nel progetto. Si è fatto ampio uso di "mixin", per definire gruppi di stili da riutilizzare in più parti nel singolo foglio di stile, e dell'ereditarietà, per definire un insieme di stili comuni in una classe madre ed ereditarli nelle classi figlie. Il codice SCSS scritto è stato organizzato in file separati, sottoforma di diversi moduli: ogni modulo appartiene ad uno specifico componente React, semplificando così la gestione e la manutenzione degli stili [19].

SASS viene compilato in CSS prima di essere utilizzato, garantendo così la compatibilità con i browser web.

5.4 Framework di supporto: Next.js

Next.js è un framework di supporto per progetti sviluppati con la libreria React. Questo framework è divenuto fin da subito molto popolare in quanto permette di sviluppare applicazioni web basate su React con molta più facilità, proponendo soluzioni a problemi comuni che gli sviluppatori devono puntualmente risolvere in ogni progetto React sul quale si trovano a lavorare [16].

5.4.1 Perché Next.js

React è, come già ampiamente descritto, una libreria, cioè un package di terze parti che consente di creare interfacce utente interattive con molta più facilità rispetto al più classico utilizzo, diretto, di HTML, CSS e JavaScript. Data la sua natura di libreria, React si concentra su uno specifico aspetto dello sviluppo web: l'interfaccia utente. Tutto il resto, come ad esempio gli aspetti di routing o di autenticazione degli utenti, non viene coperto dalla libreria, che consente però di essere espansa tramite l'integrazione, all'interno dei singoli progetti, di ulteriori librerie di terze parti [16].

Next.js è, invece, un framework; si definisce "The React Framework for the Web" [6]. Un framework, per sua natura, offre soluzioni a problemi comuni e stabilisce linee guida chiare (guidance) per la struttura del codice e l'organizzazione dei file all'interno del progetto.

Next.js "espande" React, semplificando ulteriormente lo sviluppo di applicazioni web, soprattutto per progetti di grandi dimensioni. Il framework si pone l'obiettivo di semplificare la vita degli sviluppatori, costruendo uno strato sopra React che

[2]SCSS: acronimo di Sassy Cascading Style Sheets

aggiunga tutte quelle funzionalità fondamentali che sarebbe altrimenti stato necessario includere manualmente tramite l'ausilio di diversi pacchetti di terze parti [16].

La scelta di integrare Next.js all'interno del progetto di tesi è motivata proprio dal fatto che Next.js, includendo al suo interno soluzioni a problemi comuni, come ad esempio il già citato routing, l'ottimizzazione delle immagini, il pre-rendering delle pagine e altro ancora, consente allo sviluppatore di concentrarsi sullo sviluppo delle funzionalità del sistema piuttosto che perdere tempo e risorse a "reinventare la ruota".

5.4.2 Caratteristiche principali di Next.js

Next.js può essere visto come un "mattoncino" posto sopra React, che ha il compito di migliorare l'esperienza di sviluppo dei programmatori tramite l'introduzione di caratteristiche non presenti in React e la rivisitazione di caratteristiche presenti, ma ritenute poco pratiche. Ne segue una descrizione delle principali caratteristiche che hanno reso Next.js uno dei framework più apprezzati dagli sviluppatori front-end e che hanno avuto un ruolo rilevante nella decisione di utilizzare Next.js all'interno del progetto di tesi.

Pre-rendering

La feature più importante offerta da Next.js è, senz'ombra di dubbio, il supporto al Pre-rendering delle pagine. In un'applicazione React standard, ispezionando il codice sorgente delle pagine caricate dal client, risulta subito evidente come esse siano, praticamente, vuote all'inizio; conterranno, come illustrato in Figura 5.1, soltanto uno scheletro HTML di base. L'applicazione web verrà caricata e renderizzata all'interno del body, più specificatamente nell'elemento con id "root".

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,user-scalable=no,initial-scale=1,maximum-scale=1,minimum-scale=1">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Antonio Colelli</title>
    <script defer="defer" src="/main.js"></script>
</head>
<body>
<div id="root"></div>
</body>
</html>
```

Figura 5.1: Codice sorgente di una pagina sviluppata in React

La fase di rendering è gestita da React. Dato che React è una libreria client-side, tutto il rendering dell'applicazione avverrà di conseguenza all'interno del client, dunque nei browser degli utenti. Questo è il motivo per cui, quando un utente visita le pagine di un'applicazione web sviluppata in React, il codice HTML inviato dal server non contiene altro se non il precedentemente citato scheletro. Se deve essere renderizzata una pagina contenente una lista di dati, l'utente che la richiede dovrà attendere, visualizzando un'iniziale schermata di caricamento, che i dati vengano acquisiti dal client, ad esempio mediante una chiamata ad una *API REST*. Ciò è dovuto al fatto che la fase di recupero dei dati può iniziare solo dopo che il codice Javascript viene caricato ed eseguito sul client. Questo comportamento può risultare problematico quando si desidera che la pagina venga indicizzata dai motori di ricerca. I motori di ricerca, tramite i loro *crawler*, vedranno soltanto una pagina HTML vuota; la lista non verrà rilevata poiché, come precedentemente spiegato, essa verrà renderizzata successivamente, lato client. Questa situazione potrebbe costituire un problema [16].

Next.js, attraverso l'implementazione del concetto di *Pre-rendering*, offre una soluzione efficace a questo inconveniente. Se le pagine fossero pre-renderizzate sul server e i dati fossero, di conseguenza, recuperati direttamente dal server, quando esso viene raggiunto da una richiesta di pagina HTML, potrebbe fornire agli utenti e ai crawler dei motori di ricerca le pagine finite. In questo modo, gli utenti non dovrebbero attendere il caricamento iniziale dei dati e i motori di ricerca sarebbero in grado di visualizzare il contenuto completo delle pagine.

Sebbene React contenga una funzionalità che permette di implementare il Pre-rendering, essa richiede una configurazione aggiuntiva e può risultare complessa da gestire. Al contrario, con Next.js, tutto diventa molto più semplice, poiché il framework incorpora già tale funzionalità. Next.js, in modo automatico e senza necessità di ulteriori configurazioni, effettua di default il pre-rendering delle pagine; ciò significa che, sviluppando un'applicazione standard Next.js, le sue pagine saranno già pre-renderizzate "out-of-the-box" dal server senza la necessità di effettuare configurazioni extra. Il client riceverà tutto il contenuto della pagina richiesta già nel sorgente HTML che gli verrà servito dal server. Questa caratteristica, oltre a rappresentare un grande vantaggio per l'ottimizzazione per i motori di ricerca (la cosiddetta *SEO*), rappresenta un grande vantaggio in termini di esperienza utente, in quanto i fruitori delle pagine del sito non dovranno più attendere alcun caricamento iniziale dei dati, ma avranno tutto disponibile fin da subito[16].

File-Based Routing

Un'altra delle feature principali di Next.js è il *File-Based Routing*. React non include, tra le sue funzionalità, il Routing. Nel contesto di React, il Routing rappresenta l'illusione data agli utenti di avere più pagine. Tipicamente, per

implementare tale funzionalità, è necessario includere, nel progetto, pacchetti esterni come "React Router" [20], che gestiscono il passaggio da una "pagina" all'altra: viene controllato l'URL e, nel caso in cui cambi, si interviene per impedire che il browser, come da suo comportamento di default, invii richieste al server per richiedere una nuova pagina. React Router richiede una configurazione iniziale, fatta all'interno del codice mediante il componente <Route>, che descriva le route disponibili all'interno dell'applicazione, in modo che possa poi richiedere a React di renderizzare contenuti specifici discriminando in base all'URL richiesto dall'utente. Con Next.js tutto ciò diventa superfluo in quanto il framework definisce una chiara organizzazione dei file e delle cartelle che permetta al framework stesso di definire, in automatico, le route: si richiede la presenza, nella directory principale del progetto, della cartella "pages", illustrata con un esempio in Figura 5.2, all'interno della quale andranno creati, tramite una chiara organizzazione, i file e le cartelle che definiranno route e path [16].

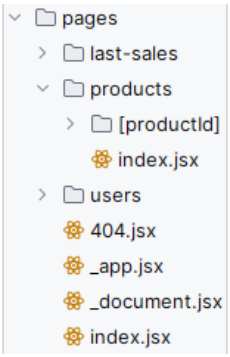

Figura 5.2: Cartella pages di un progetto Next.js che usa il File-Based Routing

Il File-Based Routing, consentendo l'aggiunta di nuove pagine in maniera semplice e diretta tramite la creazione di nuovi file e cartelle opportunamente organizzati all'interno della directory "pages", annulla la necessità di scrivere codice aggiuntivo che definisca le route, consentendo dunque allo sviluppatore di risparmiare tempo e di mantenere il codice più pulito [16].

Fullstack Capabilities

Nei progetti sviluppati utilizzando Next.js è possibile scrivere anche codice back-end creando, di fatto, progetti full-stack in cui sono presenti sia codice client, che verrà eseguito sui browser degli utenti, che codice server stand-alone sviluppato in Node.js, che verrà invece eseguito in un contesto privato. Il codice back-end, non accessibile al client, può essere impiegato per svolgere operazioni interne, stabilire connessioni con database esterni, realizzare sistemi di autenticazione e altro. Questo permette, potenzialmente, di non avere più la necessità di creare progetti separati per client e

server, consentendo invece di gestire tutto all'interno di un unico progetto Next.js, integrando sia la parte client-side che quella server-side [16].

5.5 Principali pacchetti utilizzati

Il progetto sviluppato nell'ambito di questo lavoro di tesi fa dunque uso della libreria React e del framework Next.js. L'utilizzo combinato di queste due tecnologie ha effettivamente consentito, come previsto, di concentrare il tempo e le risorse principalmente sullo sviluppo delle funzionalità del sistema. Tuttavia, per soddisfare requisiti specifici e implementare caratteristiche aggiuntive, si è reso necessario integrare pacchetti di terze parti all'interno del progetto. Ne segue una breve descrizione dei principali pacchetti integrati e di come sono stati utilizzati all'interno del progetto.

React Bootstrap

Bootstrap [7] è un framework open source per lo sviluppo di interfacce responsive per siti e applicazioni web. React Bootstrap [21] è, invece, una libreria open-source che fornisce componenti React basati su Bootstrap.

La libreria è stata utilizzata per sviluppare l'intera interfaccia del nuovo portale mediante i componenti "Container", "Navbar", "Row", "Col", "Button", "Modal", "Form", "InputGroup" e "Spinner".

MUI

MUI (actonimo di Material-UI) [22] è una libreria di componenti UI per React, che segue le guidelines di Material Design proposte da Google.

La libreria è stata utilizzata per sviluppare il calendario di prenotazione all'interno della pagina Saletta mediante il componente "DateCalendar".

Framer Motion

Framer Motion [23] è una libreria per l'animazione e la creazione di transizioni fluide e interattive nelle interfacce utente.

La libreria è stata utilizzata principalmente per animare le apparizioni dei componenti, il comportamento (in particolare per card e pulsanti) al passaggio e al click del mouse e l'apertura/chiusura dei menu a scomparsa.

Day.js

Day.js [24] è una libreria per la manipolazione delle date e degli orari in JavaScript.

La libreria è stata utilizzata per formattare gli orari di apertura delle strutture coerentemente con lo stile proposto dai mockup e per gestire tutta la logica temporale necessaria a supportare la procedura di prenotazione di salette ad affitto orario.

Axios

Axios [25] è una libreria per la gestione delle richieste HTTP basata su Promise, utilizzata principalmente per effettuare chiamate AJAX[3] da browser o da server Node.js.

La libreria è stata utilizzata per gestire le comunicazioni tra il progetto ed il server di Musictogo.

react-google-maps-api

Questa libreria, denominata react-google-maps-api [26], fornisce, tramite dei componenti preconfigurati, un'interfaccia per integrare le mappe di Google Maps nelle applicazioni React; l'utilizzo di questa libreria semplifica notevolmente il processo di integrazione di Google Maps in un progetto React.

La libreria è stata utilizzata per integrare le mappe di Google Maps in Homepage e nelle pagine Struttura, Saletta e Servizio mediante i componenti "GoogleMap", "MarkerF" e "InfoWindowF".

supercluster

Questa libreria, denominata supercluster [27], è utilizzata per il clustering di punti geospaziali.

La libreria è stata utilizzata nella mappa in Homepage per raggruppare strutture vicine all'interno di cluster (rappresentati poi in mappa mediante il componente "MarkerF" precedentemente citato), in modo da mantenere la mappa pulita anche quando si prevede di mostrare a schermo numerose strutture. Quando lo zoom della mappa consente una visualizzazione "pulita" dei marker delle strutture, il supercluster esplode automaticamente nei singoli marker che raggruppa.

React Slick

React Slick [28] è una libreria per creare caroselli di immagini o contenuti all'interno di applicazioni web basate su React.

La libreria è stata utilizzata nelle pagine Saletta e Servizio per dare vita ai processi di prenotazione e acquisto a step tramite la creazione di un carosello di

[3]AJAX: acronimo di Asynchronous JavaScript and XML

componenti movimentato su richiesta, ad esempio tramite la pressione dei pulsanti "Avanti" e "Indietro".

React Tooltip

React Tooltip [29] è una libreria per aggiungere tooltip interattivi e personalizzabili agli elementi dell'interfaccia utente.

La libreria è stata utilizzata nelle pagine Saletta e Servizio per fornire indicazioni aggiuntive quando l'utente passa il mouse sulle info-icon sparse tra i vari componenti all'interno della card di prenotazione e acquisto.

react-phone-number-input

Questa libreria, denominata react-phone-number-input [30], è utilizzata per la gestione di input di numeri telefonici.

La libreria è stata utilizzata per gestire, tramite il metodo "isValidPhoneNumber" da essa esposto, la validazione dei numeri di telefono inseriti dagli utenti durante la procedura di prenotazione/acquisto.

email-validator

Questa libreria, denominata email-validator [31], è utilizzata per la validazione degli indirizzi email.

La libreria è stata utilizzata per gestire, tramite il metodo "validate" dell'oggetto "validator" da essa esposto, la validazione degli indirizzi email inseriti dagli utenti durante la procedura di prenotazione/acquisto.

otp-client

Questa libreria, denominata otp-client [32], è utilizzata per creazione di codici OTP.

La libreria è stata utilizzata per la creazione, tramite il metodo "getToken" dell'oggetto "OTP" da essa esposto, di codici OTP della durata di 300 secondi da utilizzare nella procedura di verifica delle email degli utenti.

EmailJS

EmailJS [33] è un servizio per l'invio di email da tecnologie client-side. Consente, tramite il proprio SDK [4], di integrare facilmente le funzionalità di invio email direttamente all'interno delle applicazioni web basate su React.

[4]SDK: acronimo di Software Development Kit

Il SDK è stato utilizzato per l'invio, tramite il metodo "send" dell'oggetto "emailjs" da esso esposto, dei codici OTP alle email degli utenti, al fine di verificarle.

Capitolo 6

Implementazione dell'interfaccia

Questo capitolo offre al lettore una panoramica del lavoro svolto per implementare il nuovo portale. La descrizione di quanto fatto nella pratica verrà, saltuariamente, accompagnata da una descrizione più teorica dei principi sui quali si basano le scelte fatte. Le tecnologie utilizzate per sviluppare il nuovo portale sono quelle descritte nel Capitolo 5. Queste tecnologie sono state combinate tra loro per creare un'applicazione web che rifletta lo stile e l'interattività proposti nella Sezione 4.3 del Capitolo 4.

6.1 Ambiente di sviluppo

Al fine di agevolare i processi di sviluppo, revisione e versioning del codice, si è fatto uso della piattaforma GitLab [34] e dell'ambiente di sviluppo software IntelliJ IDEA [35].

6.1.1 Utilizzo di GitLab

Gitlab è stato utilizzato per la gestione del ciclo di vita del software sviluppato. Per strutturare in modo efficacie il flusso di lavoro, è stata adottata una strategia di branching denominata "GitFlow" [36]. Questa strategia prevede la presenza, nel repository del progetto, di due branch principali, comunemente denominati "main" e "dev", rispettivamente il branch utilizzato per contenere il codice sorgente pronto per essere rilasciato in produzione ed il branch di sviluppo principale dove le nuove funzionalità vengono integrate prima di essere unite al branch di produzione.

Dopo aver completato lo sviluppo di una nuova funzionalità all'interno di un branch di tipo feat, è necessario creare una "pull request" da tale branch verso

il branch dev. A seguito della pull request, uno o più "revisori" sono tenuti ad esaminare il codice scritto che si desidera "mergiare" in dev, fornendo, all'occorrenza, eventuali feedback per richiedere modifiche al codice prima che questo possa essere, tramite l'approvazione della richiesta, "mergiato" in dev. Una volta ottenuta l'approvazione della richiesta, è consentito il merge del nuovo codice nel branch dev.

L'aver adottato tale strategia, unitamente all'aver fissato delle "milestone" bisettimanali che guidassero l'avanzamento lavori, ha reso il processo di sviluppo del codice efficiente e trasparente agli occhi di Musictogo.

6.1.2 Utilizzo di IntelliJ IDEA

IntelliJ IDEA è stato utilizzato per la scrittura del codice. Questo IDE [1] si è dimostrato uno strumento fondamentale grazie all'intuitiva interfaccia e alla vasta gamma di funzionalità che mette a disposizione del programmatore, tra le quali spicca l'integrazione con Git per la gestione dei repository locali e remoti, l'esecuzione di comandi come push, pull, merge e rebase, la gestione dei conflitti e molto altro.

All'interno di IntelliJ IDEA è stato integrato "SonarLint" [37], ovvero uno strumento di analisi statica del codice, utilizzato per il controllo della qualità. SonarLint, eseguendo analisi automatiche prima di ogni push sul repository, ha spesso rilevato e segnalato problemi come variabili non utilizzate, import inutili e altro. Questa integrazione ha aiutato a migliorare notevolmente la qualità complessiva del codice e ha contribuito a ridurre il tempo necessario per individuare e correggere problemi ed errori.

6.2 Il ruolo di Next.js nello sviluppo del codice

6.2.1 File based routing

Il File-Based Routing di Next.js, come anticipato nella Sottosezione 5.4.2, è una feature che permette di creare un routing basato sul file system e sul concetto di pagina. Quando un file viene aggiunto nella cartella "pages", esso viene automaticamente reso disponibile come route dell'applicazione [16].

Come richiesto da Next.js, all'interno del progetto è stata creata la cartella "pages". Il primo file inserito in questa directory è "index.jsx", il quale esporrà un componente React che Next.js utilizzerà per rappresentare la pagina della route di Homepage. Per definire i path, sono state create delle sottocartelle

[1] IDE: acronimo di Integrated Development Environment

all'interno di "pages". A ciascuna sotto cartella è stato dato un nome sulla base del path che rappresenta. All'interno di ogni sotto cartella, per la generazione dei path statici, sono stati collocati ulteriori file "index.jsx", ciascuno dei quali rappresentativo della pagina associata al rispettivo path. Ad esempio, creando la cartella "assistenza" ed inserendo al suo interno il file "index.jsx", navigando, in produzione, a booking.musictogo.it/assistenza verrà caricato il componente contenuto in quel file. Per la generazione dei path dinamici, ovvero quei percorsi che possono essere utilizzati per la creazione di URL univoci per pagine basate su risorse variabili, sono state invece posizionate, all'interno delle corrispondenti sottocartelle, ulteriori cartelle denominate [id], dove "id" rappresenta l'identificatore univoco di una specifica risorsa. All'interno di queste cartelle sono stati poi inseriti i file "index.jsx", questa volta rappresentativi, in maniera dinamica, delle singole risorse. Ad esempio, creando la cartella "strutture", posizionando al suo interno la cartella "[structureId]" ed inserendo, all'interno di quest'ultima, il file "index.jsx", navigando, in produzione, a booking.musictogo.it/strutture/1 verrà caricato il componente contenuto in quel file, opportunamente idratato con le informazioni della struttura con id pari ad 1 [16].

È pratica comune che pagine corrispondenti a path diversi condividano alcuni elementi, come la Navbar. Sarebbe uno sforzo inutile aggiungere, manualmente, questi elementi in ogni pagina, dato che è possibile inserirli all'interno di un singolo file, da creare sempre nella cartella "pages" e da denominare "_app.jsx". Next.js richiede che, se creato, questo file contenga al suo interno un componente React che avrà il compito di fare da "wrapper" dell'applicazione. All'interno di questo componente sarà possibile inserire elementi comuni a tutte le pagine, come contesti e componenti tipo la Navbar. Il componente riceverà, tramite le sue props, gli oggetti "Component" e "pageProps". "Component" rappresenta la pagina attiva, quindi, quando si naviga tra le diverse route, questo oggetto conterrà sempre la nuova pagina. L'oggetto "pageProps" conterrà, invece, le props iniziali precaricate per la pagina attiva [16].

Per l'implementazione di una pagina 404 da utilizzare, ad esempio, quando l'utente prova a navigare un URL dinamico nel tentativo di visualizzare una risorsa non presente a sistema, è stato sufficiente inserire, all'interno della directory "pages", un file denominato "404.jsx".

L'organizzazione in sottocartelle ha consentito di creare facilmente path e sotto-path, sia statici che dinamici, rendendo il processo di creazione dei percorsi dell'applicazione estremamente semplice.

La navigazione tra i diversi percorsi è stata resa possibile dall'utilizzo dell'oggetto "useRouter" di Next.js. Le strategie adottate all'interno di questo progetto per gestire la navigazione verranno meglio descritte all'interno della Sottosezione 6.4.1 del presente capitolo.

6.2.2 Pre-rendering

L'idea, come anticipato nella Sottosezione 5.4.2, è che Next.js prepari le pagine in anticipo, costruendo il contenuto HTML ed eventualmente caricando dei dati, se richiesti. Next.js ha due forme di pre-rendering tra le quali il programmatore può scegliere in base alle proprie necessità: Static Site Generation (SSG) e Server-side Rendering (SSR). La prima prevede che tutte le pagine siano pre-generate in anticipo durante la fase di build iniziale del progetto. La seconda prevede, invece, che il server, a seguito delle richieste ricevute, serva le pagine creandole "on the fly" per ogni singola richiesta [16].

getStaticProps

Next.js, di default, pre-renderizza tutte le pagine a build time, ma a volte è necessario indicargli quali siano i dati che i componenti necessiteranno in input per poter pre-renderizzare un determinato contenuto. Esiste una funzione, denominata "getStaticProps", che svolge questo compito. Il tipico file "index.jsx" appartente alla cartella "pages" o ad una delle sue sottocartelle, come precedentemente descritto nella Sottosezione 6.2.1, conterrà al suo interno un componente React, descrittivo della pagina che verrà pre-renderizzata da Next.js. Questo componente potrebbe necessitare di una o più props. All'interno del medesimo file è possibile inserire la funzione "getStaticProps", che avrà il compito di recuperare i dati necessari al componente e ritornarli tramite un oggetto. Sarà poi Next.js, in maniera autonoma, a utilizzare l'oggetto ritornato da getStaticProps come input per il componente. Un esempio di quanto precedentemente descritto è illustrato in Figura 6.1: getStaticProps recupera del contenuto, lo inserisce all'interno della costante "data" e lo ritorna all'interno di un oggetto; Next.js si occuperà poi di idratare il componente "Home" con il contenuto ritornato da getStaticProps e inviare al client il componente pre-renderizzato.

```
function Home({data}) {
    return (
        <Homepage data={data}/>
    )
}


export async function getStaticProps() : Promise<{...}>  {
    const data : number  = await getData()
    return {
        props: {data},
    }
}
```

Figura 6.1: Esempio di utilizzo di getStaticProps

getStaticProps e Incremental Static Regeneration

Se i dati che devono essere contenuti all'interno di una specifica pagina cambiano frequentemente, non è sufficiente effettuare il pre-rendering della pagina a build time, ma si rende necessario implementare un meccanismo di "rinfresco". Next.js offre una feature, denominata "Incremental Static Regeneration", che permette la gestione di questi casi: oltre ad essere pre-renderizzate a build-time, le pagine interessate all'Incremental Static Regeneration verranno rigenerate, in base alle richieste ricevute, al massimo una volta ogni X secondi, dove X è un parametro da inserire nell'oggetto ritornato dalla funzione getStaticProps. Nell'esempio in Figura 6.2 è illustrato come, tramite il campo "revalidate" sia stato inserito un numero di secondi pari a 30 per gestire la rigenerazione della pagina.

```
export async function getStaticProps() : Promise<{...}>  {
    const data : number  = await getData()
    return {
        props: {data},
        revalidate: 30
    }
}
```

Figura 6.2: Esempio di applicazione di Incremental Static Regeneration in getStaticProps

All'interno del progetto di tesi, questo meccanismo è stato utilizzato nella Homepage per ottenere i dati di tutte le strutture, nella pagina Struttura per ottenere i dati della singola struttura e nelle pagine Saletta e Servizio per ottenere i dati della specifica saletta o dello specifico servizio, con un timer di rigenerazione fissato a 1200 secondi (20 minuti).

getStaticProps e parametro "notFound"

L'oggetto ritornato da getStaticProps, oltre a "props" e "revalidate", può contenere il campo "notFound" nel caso in cui si decida di far ritornare, in determinati contesti, pagina 404 al posto della pagina richiesta. Nell'esempio in Figura 6.3 è illustrato come, al verificarsi di una specifica condizione, sia possibile sfruttare il parametro "notFound" di getStaticProps per ritornare pagina 404.

```
export async function getStaticProps() : Promise<...>  {
    const data : number  = await getData()
    if (!data)
        return {notFound: true}
    return {
        props: {data},
        revalidate: 30
    }
}
```

Figura 6.3: Esempio di applicazione del parametro "notFound" in getStaticProps

All'interno del progetto di tesi, questo meccanismo è stato utilizzato per la gestione di richieste in cui, all'interno dell'URL, l'id della struttura, della saletta o del servizio risultassero non validi.

getStaticPaths

Di default, le pagine dei path dinamici (quelli create con [id]) non vengono pre-renderizzate da Next.js in quanto il framework non può sapere in anticipo il numero di pagine che dovrà creare per tali path. Di conseguenza, le pagine dei path dinamici vengono generate "on the fly" dal server a seguito di ogni richiesta. Per essere pre-renderizzate, le pagine di questi path richiedono più del solo getStaticProps. È necessario informare Next.js su quale sia l'insieme, o il sottoinsieme, di id che si vorrebbero pre-renderizzare per uno specifico URL dinamico, utilizzando il metodo "getStaticPaths", da aggiungere all'interno del file "index.jsx" della pagina di riferimento. L'esempio in Figura 6.4 illustra come viene utilizzato getStaticPath per permettere alle pagine di un URL dinamico, sulla base degli id estratti dalla lista "events", di essere pre-renderizzate. La funzione, che viene eseguita una sola volta da Next.js, in fase di build iniziale, ritorna anche un parametro denominato "fallback", che istruisce il framework circa il comportamento da adottare nel caso in cui venga richiesto un id non presente in lista. Nell'esempio, così come nel progetto di tesi, questo parametro viene settato a "blocking", ovvero una configurazione che permette la creazione "on the fly", e il successivo salvataggio in cache, della pagina se essa è richiesta per un id non presente in lista. L'id verrà successivamente aggiunto in lista e considerato, per le richieste future, al pari di tutti gli altri id precedentemente inseriti in lista a build time da getStaticPath.

```
export async function getStaticPaths() : Promise<{...}>  {
    const events : number  = await getAllEvents()
    const paths = events.map(event => ({params: {eventId: event.id}}))
    return {
        paths: paths,
        fallback: 'blocking'
    }
}
```

Figura 6.4: Esempio di utilizzo di getStaticPaths

La funzione getStaticPaths, tramite il parametro "fallback" si limita soltanto a consentire, o meno, che id fuori lista siano processabili da getStaticProps. Sarà getStaticProps a decidere se uno specifico id sia da considerarsi valido o se si renda necessario invocare la pagina 404.

Come accennato poc'anzi, getStaticPaths è stato utilizzato anche all'interno del progetto di tesi, nello specifico per la gestione del pre-rendering delle pagine Struttura, Saletta e Servizio; il parametro "fallback" è stato settato a "blocking".

6.2.3 Full stack capabilities

Grazie a Next.js, il programmatore ha la possibilità di gestire, all'interno di un unico progetto, sia un front-end React che un back-end Node.js. Durante la fase di build iniziale, per la generazione di Homepage e delle pagine Struttura, Saletta e Servizio, sono necessari dei dati recuperabili tramite chiamate API al server di Musictogo. Per tali dati, il server di Musictogo espone un'API denominata "getAllStructcures", che restituisce un elenco contenente informazioni su tutte le strutture. Di conseguenza, nella fase di build iniziale, sarebbe necessario effettuate circa 150 chiamate a questa API: la prima per ottenere i dati necessari per la homepage, le altre per le informazioni sulle singole strutture, salette e servizi. Risulta abbastanza evidente come l'utilizzo di un'API che restituisce sempre informazioni su tutte le strutture, comporti uno spreco enorme di risorse quando viene chiamata per recuperare informazioni su una singola struttura: vengono inviati dati non necessari alla richiesta. Per la generazione di ogni singola pagina viene dunque richiesta sempre la stessa API per circa 150 volte ottenendo, presumibilmente, sempre gli stessi dati: ciò risulta in uno spreco di chiamate che avranno il solo effetto di stressare il server di Musictogo dato che il contenuto delle risposte del server sarà, presumibilmente, data la poca dinamicità dei dati e lo stretto intervallo di tempo in cui tutte queste richieste vengono effettuate, sempre lo stesso. Poiché il server di Musictogo non era oggetto della presente tesi, è stata proposta una soluzione intermedia basata sulla memorizzazione temporanea dei dati sul server Node.js, all'interno di una cache. Durante la fase di build iniziale, per la generazione della prima pagina viene fatta una chiamata all'API, in modo da ottenere i dati di tutte le strutture; dopodiché, questi dati vengono salvati in una cache temporanea all'interno del server del progetto, per essere utilizzati nella generazione delle pagine successive, riducendo così il numero di chiamate fatte al server di Musictogo da circa 150 ad 1.

Questa cache risulta utile anche a run time, in quanto si combina bene con il concetto di Incremental Static Regeneration, che prevede la rigenerazione delle pagine ogni 20 minuti. Data la scarsa dinamicità dei dati delle strutture, è molto probabile che, in fase di rigenerazione, questi rimangano invariati. Si è pertanto deciso di mantenere la cache anche a run time, rivalidandola al massimo una volta ogni 10 minuti, sulla base delle richieste ricevute. In questo modo è stato possibile minimizzare il numero di richieste al server di Musictogo. Dato il timer di rigenerazione delle pagine fissato a 20 minuti, se due pagine A e B, in quest'ordine, dovessero necessitare una rigenerazione, si proverà ad accedere alla cache per il recupero dei dati. Se durante il processo di recupero dati per la rigenerazione della pagina A, Next.js dovesse notare che la cache è più vecchia di 10 minuti, effettuerà una chiamata al server di Musictogo per rinfrescarla con dati più recenti. Successivamente, rigenererà la pagina A con i dati aggiornati e, senza la necessità di effettuare ulteriori chiamate al back-end di Musictogo, rigenererà anche la pagina

B.

6.3 Organizzazione del codice

Il codice del progetto è stato suddiviso in più cartelle, ognuna rappresentativa del tipo di file contenuti.

Per archiviare i componenti React sono state utilizzate tre cartelle: la già citata "pages" richiesta da Next.js, la cartella "layout" e la cartella "components". Si è deciso di distribuire i componenti su tre cartelle per meglio rappresentare il ruolo di ciascun gruppo di componenti all'interno del progetto.

La cartella "pages", come già ampiamente descritto, ospita i file "index.jsx" che, attraverso i propri componenti, definiscono le varie pagine del sito. Questi componenti, nel dettaglio del presente progetto, si occupano di descrivere ogni pagina ad un più alto livello: definiscono i dettagli dell'header, come il titolo, la descrizione e altri "Meta-tag" e lasciano la descrizione del body ad altri componenti, figli, definiti all'interno della cartella "layout".

All'interno della cartella "layout" e delle sue sottocartelle sono stati creati diversi file .jsx con lo scopo di definire i layout delle pagine dell'applicazione. Il layout di una pagina descrive dettagliatamente come vada organizzata la griglia degli elementi di tale pagina. I vari layout sono stati definiti principalmente mediante i componenti <Row> e <Col> di React Bootstrap; tali componenti sono stati opportunamente configurati al fine di gestire al meglio la responsività delle pagine. Creata la struttura generale delle pagine, lo step successivo è quello di aggiungervi il contenuto. Ciò viene fatto mediante l'inserimento di componenti figli, definiti nella cartella "components".

All'interno della cartella "components" vengono definiti i singoli componenti che verranno poi utilizzati come figli nei precedentemente citati file .jsx della cartella "layout". Tra questi componenti spicca, per rilevanza, la card, nelle sue varie forme: il numero di file contenuti all'interno della sotto cartella "cards" è poco più di un terzo (32) del numero totale di file contenuti in tutta la directory "components" (90); questo dato permette di prendere visione dell'importanza che tale componente ha avuto all'interno di questo progetto. Più nel dettaglio, a partire dai componenti "StaticCard" e "InteractiveCard", involucri, rispettivamente, di card statiche e card interattive, sono state costruite tutta una serie di card per rappresentare specifici elementi, come ad esempio le card interattive delle strutture in Homepage, o le card statiche descrittive della singola struttura in pagina Struttura. Altri esempi di componenti descritti all'interno della cartella "components" sono le mappe, i modali, i filtri ed i relativi tag, gli spinner di caricamento, la barra di ricerca, il calendario, il tooltip e così via.

Oltre a queste tre cartelle, ne sono state create altre per organizzare il codice che definisce le chiamate alle API, il meccanismo di cache, le costanti, i context, i modelli, le funzioni "utils" di supporto, lo stile e i file pubblici come immagini ed icone.

In particolare, il codice che descrive lo stile dell'applicazione si è diviso tra le cartelle "styles","components" e "layout". Il contenuto della cartella "styles" è composto da file di stile con estensione .scss, aventi al loro interno variabili globali per definire colori, font, dimensioni e altre informazioni da rendere disponibili ovunque nel progetto, e un insieme di stili ereditabili nei moduli degli specifici componenti. Le cartelle "components" e "layout" contengono, invece, ove necessario, file con estensione .module.scss descrittivi dello stile dei singoli componenti. Questi moduli avranno a disposizione, previa importazione, le variabili globali e l'insieme di stili precedentemente definiti nella cartella "styles".

6.4 I principali context

In React, il Context è un meccanismo che consente la condivisione di dati tra diversi componenti senza dover passare manualmente le props ad ogni livello dell'albero dei componenti [38]. All'interno del progetto di tesi si è fatto ampio uso dell'hook useContext per la condivisione di informazioni tra i diversi componenti. Alcuni contesti sono stati resi accessibili a tutti i componenti dell'applicazione, altri solo ai componenti di specifiche pagine.

6.4.1 NavigationContext

Questo context gestisce in maniera centralizzata la navigazione tra le diverse pagine dell'applicazione. Il suo ruolo è quello di fare da strato intermedio intelligente tra i vari componenti e l'oggetto "useRouter" messo a disposizione da Next.js. I componenti, grazie a questo context, non dovranno più importare ripetutamente l'oggetto "useRouter", dato che esso verrà importato all'interno del contesto, ma potranno effettuare le operazioni di navigazione chiamando metodi a più alto livello esposti dal contesto stesso, che si occuperanno poi di utilizzare l'istanza interna di "useRouter" per portare effettivamente a termine le procedure di navigazione. Le chiamate come *router.push(path)*, che richiedono l'inserimento esplicito di un path, vengono dunque mascherate dal contesto dietro a chiamate, chirurgiche, più ad alto livello, come la chiamata *navigateToStructureId(structureId)* che, nello specifico, permette al componente che la esegue di far partire la navigazione verso la pagina di una struttura richiedendo in input il solo id della struttura e non l'intero path.

Le pagine Struttura, Saletta e Servizio prevedono la presenza di un pulsante per tornare alla schermata precedente. Alla pressione di questo pulsante, si prevede che avvenga ciò che avverrebbe a seguito della pressione del pulsante "indietro"

del browser, ovvero l'esecuzione del metodo *window.history.back()* per tornare alla schermata precedente. Questo comportamento è legittimo quando la navigazione tra le pagine del portale è "uniforme", ovvero quando, partendo dalla Homepage, si navigano, avanti e indietro, le altre pagine del portale. Cosa accade però nel momento in cui la prima pagina alla quale si accede non è la Homepage? Mettendo il caso che una struttura decida di condividere, sui propri account social, l'URL che porta alla sua pagina del portale, la prima schermata che l'utente vedrebbe sarebbe quella della struttura, non la homepage. Come si deve comportare, in tal caso, il pulsante "indietro" presente all'interno della pagina? La domanda è lecita, in quanto, mantenendo il comportamento di default del router di Next.js, la pressione di quel pulsante riporterebbe l'utente alla pagina navigata in precedenza, ovvero quella social della struttura. Una prima soluzione potrebbe consistere nel nascondere, in questi particolari casi, il pulsante per tornare indietro; è una soluzione poco elegante. Sarebbe invece opportuno "illudere" l'utente, mantenendo il comportamento standard, ovvero la navigazione, rimanendo all'interno del portale, verso la schermata madre della corrente, anche se in maniera artificiale. Questo speciale comportamento è stato raggiunto tramite l'ausilio, all'interno del context di navigazione, di un contatore che, a seconda del tipo di navigazione, può essere incrementato o decrementato di 1: +1 se la navigazione parte da una pagina madre (ad esempio Homepage) con destinazione una pagina figlia (ad esempio Struttura), -1 nel caso contrario. Quando si accede per la prima volta al portale, il contatore è sempre settato a 0; questo, permette di determinare se la pagina figlia è stata o meno raggiunta a partire dalla pagina madre. Nel caso dell'esempio, il contatore a 0 all'interno della pagina Struttura è l'indicatore necessario a capire che il comportamento del sistema, alla pressione del pulsante per tornare indietro in Homepage, non potrà essere quello di default, ovvero *window.history.back()*, ma dovrà consistere in una navigazione forzata verso Homepage. Lo stesso principio viene applicato se la prima pagina visualizzata è Saletta/Servizio.

La logica di navigazione appena descritta consente di avere un comportamento uniforme del pulsante "Indietro" del portale, a prescindere da come si sia arrivati nella pagina che lo contiene.

6.4.2 Language Context

Musictogo, attualmente, ha stretto legami commerciali soltanto con strutture italiane, ma nulla vieta che in futuro possa espandere il proprio business anche a strutture estere, come nulla vieta che i fruitori del servizio possano non comprendere bene la lingua italiana. Nasce quindi l'esigenza di predisporre il portale all'utilizzo in più lingue. In particolare, questa prima versione del nuovo portale permette di essere esplorata in italiano e in inglese. Grazie alla strategia adottata, il cambio di lingua risulta istantaneo e non richiede che l'applicazione web venga ricaricata,

come invece spesso accade in altri siti multilingue. I testi presenti all'interno dell'applicazione non sono mai scritti a mano nel codice dei componenti, ma sono invece organizzati sotto forma di costanti di tipo stringa all'interno di oggetti ben definiti, denominabili "dizionari", dai quali le diverse pagine estrarranno il sub-set di testi a loro necessari. Per ogni lingua disponibile a sistema, deve essere creato un dizionario dal quale si possano estrarre tutti i testi richiedibili dalle diverse pagine. L'identificativo di ogni testo, a prescindere dalla lingua nella quale viene descritto il suo contenuto, deve essere lo stesso per ogni dizionario, in modo da consentire l'intercambiabilità dei dizionari nel sistema. I componenti delle diverse pagine, per recuperare i testi a loro necessari, potrebbero accedere direttamente ai dizionari, ma ciò li esporrebbe alla responsabilità di dover scegliere il giusto dizionario a seconda della lingua impostata a sistema; si renderebbe necessaria la scrittura di molto codice "boilerplate" all'interno di quasi tutti i componenti: una soluzione poco elegante.

La soluzione proposta richiede l'utilizzo di un "intermediario" tra le pagine e i dizionari che possa, dinamicamente, inviare alle pagine il dizionario corretto a seconda della lingua selezionata dall'utente. La figura di tale intermediario è ricoperta da un context. Il context fa uso, al suo interno, dell'hook "useState" per salvare la lingua in cui deve essere visualizzata l'applicazione web (di default, l'italiano), ed espone dei metodi per poterla cambiare. Espone, inoltre, un metodo denominato "getDictionary", che ritorna l'oggetto dizionario corrispondente alla lingua impostata nello stato. Questo metodo viene utilizzato, all'interno dei componenti, per richiedere l'oggetto dizionario da utilizzare per l'estrazione dei testi. La decisione su quale sia il dizionario da utilizzare in base alla lingua di sistema non è più a carico dei singoli componenti, ma viene presa, a monte, dal metodo "getDictionary". Il cambio della lingua farà scattare l'aggiornamento del dizionario ritornato da getDictionary che, a sua volta, causerà un re-rendering dei componenti che hanno richiesto l'oggetto dizionario, per mostrare il testo nella nuova lingua. Tutto questo senza che l'applicazione debba essere ricaricata per applicare i cambiamenti.

Ai fini di una maggiore comprensione da parte del lettore, in Figura 6.5 viene riportato un esempio di applicazione del context all'interno di un componente. Esso viene utilizzato per l'estrazione, dal dizionario, alcuni testi che saranno poi utilizzati dal componente.

```
const {getDictionary} = useContext(LanguageContext)
const {
    SUMMARY_TEXT, PAYMENT_TEXT_EXTENDED, EMAIL_VERIFY, CODE_VERIFYING,
    EMAIL_VERIFY_INSTRUCTION, EMAIL_SEND_FAILED, SEND_AGAIN_TEXT, NOT_VALID_OTP,
    BOOKING_CANCELED, COPY_PASTE_CODE_PLACEHOLDER, NOT_VALID_CODE, CODE_EXPEDING_IN,
    VERIFY_CODE_BUTTON, COPY_PASTE_CODE_TEXT_INSTRUCTION, COPY_PASTE_CODE_TEXT, RETRY_TEXT, EMAIL_SENDING,
    VERIFY_COMPLETED, FINAL_DISCLAIMER,
} = getDictionary().PAGES.STRUCTURE_ROOM.BREADCRUMB_DATA
```

Figura 6.5: Esempio di applicazione del Language Context

6.4.3 Homepage Context

Questo context nasce dall'esigenza, espressa all'interno della Sottosezione 4.3.2, di migliorare l'esperienza di navigazione dell'utente tra le diverse pagine del portale, conservando lo stato dei filtri durante la navigazione dalla Homepage alle altre pagine, in modo tale che se l'utente dovesse tornare indietro alla Homepage, troverebbe l'applicazione nello stesso stato in cui l'aveva lasciata.

Questo requisito può essere facilmente soddisfatto utilizzando un context globale che salvi al suo interno i dati relativi ai filtri attivi e allo stato della mappa. I dati, rappresentati all'interno del context attraverso diverse istanze dell'hook "useState", vengono esposti congiuntamente ai propri setState per essere utilizzati dai componenti della Homepage. Nel caso in cui si ritorni in Homepage dopo la navigazione in altre pagine, gli stati di questo context verranno utilizzati per l'inizializzazione dei componenti della pagina.

Homepage Context si è rivelato, inoltre, molto utile per centralizzare le informazioni necessarie alla corretta gestione della complessa interconnessione tra la lista, la mappa e gli stessi filtri. Basti notare come un'interazione con uno di questi tre componenti scateni una risposta, oltre che nel componente con cui l'interazione è avvenuta, anche negli altri due. Ne sono un esempio le specifiche interazioni con la mappa che producono, come risultato, la scomparsa, dai confini visibili di quest'ultima, di alcuni marker: le conseguenze di tale interazione saranno direttamente visibili sia all'interno della mappa, in quanto, essendo stata movimentata, presenterà dei confini diversi ed un numero minore di marker, che all'interno della sezione dei filtri e della lista di strutture, dato che l'aver nascosto alcuni marker farà sì che avvenga un filtraggio geografico che avrà ripercussioni anche sul numero di strutture visualizzate tramite le card.

Come accennato prima di descrivere l'esempio di interazione, l'interconnessione tra i componenti lista, mappa e filtri è molto complessa. Ciò è principalmente dovuto al fatto che si è reso necessario sviluppare un meccanismo che consentisse una sovrapposizione non distruttiva dei diversi tipi di filtri. Si è optato per una soluzione a più livelli di filtraggio, che prevede la presenza di un array iniziale al livello 0, contenente la lista delle strutture nella sua interezza, e la presenza, ad ogni livello successivo, di array intermedi il cui contenuto è calcolato mediante l'applicazione di specifici filtri sull'array del livello precedente. Gli array sono dichiarati all'interno del presente context mediante istanze dell'hook "useState", e vengono esposti congiuntamente ai propri setState per essere utilizzati dai diversi componenti della Homepage; quando viene applicato un filtro, gli array vengono aggiornati in cascata, a partire dall'array del livello in cui quel filtro opera. Per far scattare il meccanismo di aggiornamento dell'array di uno specifico livello X, è stato utilizzato l'hook "useEffect" all'interno del componente che gestisce tale livello (ad esempio, il livello di filtraggio geografico viene gestito all'interno del

componente che descrive la mappa). Il codice all'interno della useEffect applica, all'array del livello precedente, una funzione di filtraggio basata sui filtri associati al livello corrente, di cui la useEffect è responsabile; la lista filtrata, risultato di tale operazione, viene settata all'interno dello stato rappresentante l'array di stutture del livello corrente. Il codice all'interno della useEffect di uno specifico livello X, verrà eseguito ogni qual volta che una delle dipendenze della useEffect cambia di valore. Le dipendenze di tale useEffect comprendono lo stato dei filtri che gestiscono il livello X e lo stato rappresentante l'array di strutture del livello X-1: il codice della useEffect viene quindi eseguito quando avviene un cambiamento nello stato dei filtri del livello corrente o quando, a seguito di un cambiamento nello stato dei filtri di un livello precedente, avviene una propagazione delle modifiche che investirà tutti i livelli successivi a quello in cui la procedura di aggiornamento ha visto la sua nascita.

I livelli intermedi permettono di rendere modulare la procedura di filtraggio e di definire liste intermedie da poter utilizzare in componenti grafici, visibili all'utente, come la mappa. I diversi livelli rappresentano, al loro interno, la lista delle strutture nei suoi differenti stati. I livelli sono così definiti:

Livello 0: descrive la lista di tutte le strutture. Non viene mai modificato.

Livello 1: descrive la lista delle strutture del livello precedente filtrata a seguito dell'utilizzo della barra di ricerca testuale.

Livello 2: descrive la lista delle strutture del livello precedente filtrata a seguito dell'applicazione dei filtri contenuti all'interno del menu "Filtri". Questa lista viene utilizzata per definire quali siano le strutture i cui marker saranno contenuti all'interno della mappa, a prescindere che, a seguito delle interazioni dell'utente con tale componente, siano o meno visibili al suo interno.

Livello 3: descrive la lista delle strutture del livello precedente filtrata a seguito dell'interazione con la mappa. Le interazioni con la mappa possono produrre come risultato la scomparsa, dai confini visibili di quest'ultima, di alcuni marker. In tal caso, l'array di questo livello verrà filtrato dalle strutture rappresentanti i marker non visibili.

Livello 4: questo è un livello speciale che, di per sé, non applica nessun filtro alla lista del livello precedente, ma si limita a copiarne il contenuto. La presenza di questo livello si è resa necessaria per una corretta implementazione della procedura di "fake loading", ovvero una pratica comune che prevede, al fine di dare un feedback visivo agli utenti, la simulazione di un stato di caricamento dovuto all'elaborazione della richiesta. L'array di livello 3 viene aggiunto come dipendenza in una

useEffect che attiva uno stato di loading di 300ms. Mentre è attivo lo stato di loading, il sistema mostra, al posto delle card delle strutture, una serie di card di tipo placeholder, ed in background aggiorna l'array di questo livello copiandoci il contenuto aggiornato dell'array di livello 3. L'utente, al termine dello stato di loading, vedrà, nella lista delle strutture di Homepage, il contenuto dell'array di questo livello.

Come precedentemente accennato, gli array dei diversi livelli sono contenuti all'interno del presente contesto, e sono resi disponibili all'esterno. Questo approccio è risultato vincente in quanto, dato che le operazioni di filtraggio avvengono all'interno di vari componenti sparsi per tutta la Homepage, ha permesso di centralizzare gli array dei diversi livelli, in modo da poterli richiamare alla necessità in ogni componente senza doverli condividere tramite props.

6.4.4 Breadcrumb Context

Questo context nasce per la gestione delle procedure di prenotazione/acquisto all'interno delle pagine Saletta/Servizio. I diversi step di queste procedure vengono rappresentati, nella pratica, da diversi componenti React. I singoli componenti sono, come è facilmente intuibile, molto legati tra loro, in quanto fanno uso degli stessi dati (quelli della procedura) ed effettuano al loro interno operazioni che hanno ripercussioni anche sugli altri componenti. Si è pertanto deciso di optare per l'utilizzo di un context che centralizzasse la gestione della procedura. All'interno di tale context si fa uso dell'hook "useState" per salvare i dati dei quali la procedura si compone, come ad esempio nome, email e numero di telefono, al fine di renderli disponibili a tutti i componenti che costituiscono i diversi step. Al suo interno sono inoltre presenti funzioni trasversali ad ogni componente, come ad esempio quelle che effettuano i controlli in seguito alla pressione dei pulsanti "Avanti","Indietro","Prenota"/"Paga", e funzioni che, anche venendo eseguite all'interno di uno specifico componente, necessitano dei dati generati da un altro componente, come ad esempio la funzione che gestisce l'invio della mail di validazione all'interno del componente preposto a tale scopo, che utilizza, come destinatario di tale mail, l'indirizzo email inserito all'interno del form presente nel componente dello step precedente. A differenza dei context precedentemente descritti in questa sezione, che sono visibili ovunque all'interno dell'applicazione, il Breadcrumb Context è visibile limitatamente ai soli componenti della procedura di prenotazione/acquisto.

Capitolo 7

Validazione dell'interfaccia prodotta

Questo capitolo tratta la validazione della nuova interfaccia utente sviluppata nel Capitolo 6 a partire dai mockup revisionati, oggetto di discussione nel Capitolo 4.

Le sezioni di cui questo capitolo si compone, descrivono, rispettivamente, i principi teorici alla base della metodologia di valutazione utilizzata (Sezione 7.1), l'applicazione di tale metodologia per validare il lavoro svolto (Sezione 7.2) e le modifiche, anch'esse validate, apportate al progetto a seguito dei risultati ottenuti dalla prima validazione (Sezione 7.3)

7.1 Aspetti teorici

Come accennato nella Sezione 3.1, esistono principalmente due approcci per la valutazione di interfacce utente [9]:

- *Analisi svolte da esperti.* Particolarmente utili per valutare le prime versioni di un nuovo design, o per identificare le parti da riprogettare in una vecchia versione dell'interfaccia. Tale approccio è stato descritto nel dettaglio all'interno del Capitolo 3.

- *Partecipazione degli utenti.* Test e valutazioni sull'interfaccia tramite l'aiuto di potenziali utenti. Richiede un prototipo funzionante.

Oggetto della presente sezione è lo studio dell'approccio *Partecipazione degli utenti.* Le analisi svolte da esperti sono generalmente più veloci, in quanto, per effettuare una valutazione, sono necessarie quantità di tempo nell'ordine delle poche ore; sono inoltre utili per analizzare il design delle interfacce utente sulla base di ben definite regole, che, nel caso della valutazione euristica, sono il set di dieci

regole euristiche raccomandate da Nielsen [9]. Sono, però, anche prone a generare falsi positivi e a non individuare la totalità dei problemi, dato che, ad esempio, statisticamente, mediante una valutazione euristica, il singolo valutatore è in grado di trovare circa 1/3 dei problemi di usabilità che affliggono un'interfaccia utente [9]. Al contrario, le valutazioni effettuate con la collaborazione di potenziali utenti risultano generalmente in processi più lenti, in quanto richiedono il coinvolgimento di altre persone. Sono da considerarsi, però, più accurate, in quanto richiedono di far utilizzare l'interfaccia prodotta a persone che potrebbero potenzialmente utilizzarla nella loro quotidianità. Di conseguenza, queste tecniche di valutazione di interfacce utente possono essere applicate solo nelle fasi finali dello sviluppo, ovvero quando si ha a disposizione un prototipo (più o meno) funzionante. Generalmente, per eseguire questo tipo di test, è necessario avere, come accennato poc'anzi, un prototipo funzionante dell'applicazione la cui interfaccia si vuole valutare, in quanto viene richiesto ai potenziali utenti di svolgere dei task ben definiti, attraverso l'utilizzo del prototipo [39].

Nella successiva sottosezione verrà descritta, a livello teorico, una metodologia di valutazione delle interfacce tramite l'aiuto degli utenti che prende il nome di "Test di usabilità". Questa metodologia verrà poi applicata, nella Sezione 7.2 del presente capitolo, per testare l'usabilità della nuova interfaccia prodotta.

7.1.1 Test di usabilità

I *Test di usabilità* rappresentano una metodologia sperimentale per la valutazione di interfacce utente, condotta con la collaborazione di potenziali utenti. Lo scopo principale di questa metodologia è l'identificazione di problemi che gli utenti potrebbero incontrare nell'utilizzo quotidiano del sistema, e la raccolta di feedback su come, a detta degli utenti stessi, sarebbe possibile migliorare l'esperienza utente [39].

Il Nielsen-Norman Group definisce i Test di usabilità come segue [40]:

> *In una sessione di test di usabilità, un ricercatore (chiamato "facilitatore" o "moderatore") chiede a un partecipante di eseguire dei compiti, di solito utilizzando una o più interfacce utente specifiche. Mentre il partecipante completa ogni compito, il ricercatore osserva il comportamento del partecipante e ascolta i suoi feedback.*

La definizione appena riportata descrive a pieno come vada svolta una sessione di test di usabilità. La fase di svolgimento è però solo una parte della più ampia procedura attraverso la quale avviene il processo di validazione di un'interfaccia tramite test di usabilità. L'intero processo del test dell'usabilità di una interfaccia può essere, difatti, suddiviso in tre fasi [39]:

1. La fase di *pianificazione*. In questa fase vengono definiti gli obiettivi del test, il target di utenti a cui il test deve essere sottoposto, il numero di partecipanti, i task da svolgere e le modalità di esecuzione; quest'ultime devono essere scritte all'interno di uno "script" da seguire alla lettera durante fase di svolgimento, per mantenere consistenza tra le diverse sessioni. È importante che ogni task sia rappresentativo di una attività che gli utenti potrebbero effettivamente svolgere nel quotidiano utilizzo del sistema. Piuttosto che limitarsi al chiedere ai partecipanti di "fare X", ogni task andrebbe situato all'interno di uno scenario, per fornire un po' di contesto. Per essere in grado di valutare i risultati ottenuti dallo svolgimento dei singoli task, è necessario definire delle metriche, soggettive (come domande ai partecipanti) e/o quantitative (come il tempo sul task e i criteri di successo), per apprezzare l'efficacia, l'efficienza e la soddisfazione degli utenti nell'esecuzione dei task. Durante questa prima fase, viene inoltre preparato il materiale necessario per lo svolgimento del test. Questo include gli strumenti che verranno utilizzati dai partecipanti durante il test, come computer, smartphone e così via, quelli utilizzati dai ricercatori, come taccuini, registratori vocali e così via, e i documenti da sottoporre ai partecipanti prima e dopo il test, come il modulo di consenso alla partecipazione i questionari post-test, essenziali, rispettivamente, per garantire il consenso informato dei partecipanti e raccogliere ulteriori feedback significativi sull'esperienza utente.

2. La precedentemente citata fase di *svolgimento*. In questa fase, come ampiamente descritto dalla precedente definizione, vengono svolte le singole sessioni di test. Si richiede la presenza di una persona che agisca nel ruolo del facilitatore e di almeno un'altra nel ruolo dell'osservatore. Il facilitatore deve seguire lo script rimanendo in uno stato di neutralità e fornendo istruzioni chiare, senza aiutare i partecipanti. Gli osservatori devono prendere nota del comportamento dei partecipanti durante lo svolgimento dei singoli task: i commenti, gli errori e informazioni circa il (parziale) successo o il fallimento del task di ogni utente, saranno dati fondamentali per la successiva fase di analisi.

3. La fase di *analisi*. In questa ultima fase del processo di test di usabilità, vengono analizzati i risultati ottenuti a seguito delle singole sessioni di test per individuare eventuali problemi di usabilità e spunti per possibili miglioramenti dell'interfaccia, considerando le metriche soggettive e/o quantitative definite nella prima fase del processo.

7.2 Test di usabilità dell'interfaccia prodotta

La metodologia adottata per validare il lavoro che ha portato allo sviluppo del nuovo portale Booking di Musictogo, è quella dei "Test di usabilità". Nelle successive sottosezioni verranno descritte le diverse fasi che hanno caratterizzato questo processo di validazione.

7.2.1 Fase di pianificazione

Musictogo è un'azienda che opera nel settore musicale; più nello specifico, come già largamente descritto in questo documento di tesi, consente a strutture come sale prove e studi di registrazione di offrire i loro servizi attraverso il portale di Booking il cui rinnovamento è stato oggetto di discussione nei precedenti capitoli. Pertanto, il target di popolazione a cui far svolgere i test di usabilità non può che essere costituito da potenziali clienti di tali strutture, ovvero, individui, come musicisti, sia dilettanti che professionisti, alla ricerca di un luogo appropriato per esercitarsi, sperimentare le proprie composizioni o registrare la propria musica.

Si è deciso di optare per un numero di partecipanti pari a cinque; questo numero non viene scelto a caso, in quanto esso risulta in linea con le raccomandazioni di Jakob Nielsen, ovvero un noto esperto di usabilità e di progettazione di interfacce utente, nonché uno degli ideatori della metodologia di "Valutazione euristica" descritta nella Sezione 3.1 del presente documento di tesi. Entrando più nel dettaglio, questa scelta si basa sulla ricerca di Nielsen che dimostra come, con un campione relativamente piccolo, ovvero cinque partecipanti, sia possibile individuare la maggior parte dei problemi di usabilità di un sistema. Effettuare i test con un più elevato numero di partecipanti comporta costi aggiuntivi e richiede più tempo ma, statisticamente, non si traduce in significativi miglioramenti nell'efficacia del processo di testing [41].

Sempre più utenti accedono a Internet utilizzando dispositivi come gli smartphone; si rende dunque necessario testare l'interfaccia sia nella sua versione Desktop che in quella Mobile. Dato che, ad oggi, il mercato Mobile occupa più del 59% delle quote di mercato, mentre quello Desktop si ferma a poco meno del 38% [42], risulta naturale prediligere i test di usabilità del nuovo portale nella sua versione Mobile. Delle cinque sessioni di test in programma, infatti, tre verranno effettuate attraverso la versione Mobile del nuovo portale, lasciando alla versione Desktop solo due sessioni.

Sono stati definiti un totale di sei task da far svolgere ad ognuno dei partecipanti. I sei task testano le attività che ci si aspetta che gli utenti svolgano frequentemente durante l'utilizzo quotidiano del portale. Al fine di fornire del contesto, per ognuno dei task è stato definito uno scenario. Lo script utilizzato per guidare i partecipanti attraverso i diversi task da svolgere, completo dei testi utilizzati per descrivere i

singoli task e i relativi scenari, è disponibile all'Appendice C, in calce al presente documento. Ne segue ora una breve descrizione dei task e dei loro obiettivi.

- *Task 1.* Il primo task consiste nel richiedere agli utenti di ricercare una sala prove nella zona di Torino, che sia aperta la domenica pomeriggio. Dato che gli orari delle strutture non sono disponibili in Homepage, ma solo a partire dalla pagina Struttura, gli utenti, per trovare quello che cercano, saranno obbligati ad aprire le pagine delle varie strutture. La scelta della giornata di domenica non è casuale, in quanto, la domenica, per la maggior parte delle strutture, è giorno di chiusura; quando si sono svolti i test di usabilità, nella zona di Torino solamente 3 strutture su 12 risultavano aperte di domenica. Ciò comporta che, nel 75% dei casi, la prima card cliccata in Homepage dagli utenti risulti in una struttura chiusa la domenica, rendendo dunque necessaria un'interazione con il sistema per tornare indietro e cercare altre strutture. L'obiettivo principale di questo task è proprio quello di valutare i meccanismi con cui il sistema permette la navigazione tra le varie pagine dell'applicazione. Durante lo svolgimento del task, si presterà particolare attenzione nell'osservare le diverse strategie che gli utenti adotteranno per raggiungere l'obiettivo preposto. Sarà di rilevante interesse analizzare l'eventuale uso che verrà fatto del menu dei filtri, della barra di ricerca e della mappa, per filtrare la lista delle strutture visualizzate; in tal caso, inoltre, si monitorerà se, dopo un eventuale ritorno alla Homepage a seguito della visualizzazione di una struttura chiusa la domenica, gli utenti percepiranno come "normale" il fatto che il sistema abbia salvato lo stato della Homepage, evitando così la necessità di riapplicare tutti i filtri. In altre parole, si intende verificare se tale comportamento del sistema corrisponde alle aspettative degli utenti in termini di navigazione tra le pagine. I criteri di successo di questo task riguardano dunque l'efficacia dell'interfaccia nel permettere agli utenti di navigare il sistema e di individuare e recuperare facilmente le informazioni necessarie, ovvero, più nello specifico, gli orari di apertura delle strutture.

- *Task 2.* Il secondo task richiede agli utenti di prenotate una saletta ad affitto orario di una struttura che si trovi nelle vicinanze. Non vengono espressi vincoli sulla data e sugli slot orari in cui deve avvenire la prenotazione. La definizione di "vicinanza" è lasciata alla libera interpretazione dei partecipanti. L'obiettivo del task è verificare l'efficacia dell'interfaccia nel consentire agli utenti di portare a termine una procedura di prenotazione. Come nel precedente task, sarà di rilevante interesse analizzare l'eventuale uso che verrà fatto del menu dei filtri, della barra di ricerca e della mappa, per filtrare la lista delle strutture visualizzate; in particolar modo, con la richiesta di cercare strutture nelle vicinanze dei partecipanti, si vuole osservare l'approccio di questi ultimi rispetto alla natura "geografica" del task, analizzando la frequenza con cui

avvengono le interazioni con la mappa e con il suo pulsante di geolocalizzazione. L'eventuale non utilizzo di tale funzionalità non compromette il successo del task, che, come precedentemente accennato, vuole verificare che l'interfaccia sia in grado di guidare gli utenti lungo tutta la procedura di prenotazione. Sarà dunque rilevante osservare come gli utenti approcceranno la scelta della saletta e la procedura di prenotazione nei diversi step che la compongono. I criteri di successo di questo task riguardano dunque l'efficacia dell'interfaccia nel permettere agli utenti di completare una procedura di prenotazione, rispettando il vincolo, precedentemente descritto, sulla vicinanza della struttura.

- *Task 3.* Il terzo task si pone l'obiettivo di testare la capacità del sistema di far individuare, agli utenti, strutture aventi specifiche caratteristiche. In questo task, viene richiesto di effettuare la prenotazione di una saletta in promo, ovvero di una saletta presso una struttura che abbia lo status di promozione attivo sulle proprie salette. Non vengono espressi vincoli sulla data e sugli slot orari in cui deve avvenire la prenotazione. Come nel precedente task, sarà di rilevante interesse analizzare l'eventuale uso che verrà fatto del menu dei filtri, della barra di ricerca e della mappa, per filtrare la lista delle strutture visualizzate; in particolar modo, richiedendo la presenza dello status di promozione, si vuole osservare l'approccio degli utenti alla ricerca di tale caratteristica, che sia semplicemente scorrendo la lista delle card, o applicando il filtro "Promo" dal menu dei filtri, o altro. I criteri di successo di questo task riguardano dunque l'efficacia dell'interfaccia nel permettere agli utenti di completare una procedura di prenotazione, rispettando il vincolo, precedentemente descritto, sullo stato promozionale della saletta selezionata.

- *Task 4.* Il presente task potrebbe essere inizialmente percepito come una ripetizione dei precedenti, poiché richiede un'ulteriore prenotazione di una saletta ad affitto orario. Mentre il Task 2 si pone di osservare le eventuali interazioni con la mappa e il Task 3 osserva l'eventuale utilizzo dei filtri, questo task si propone di osservare l'approccio degli utenti nella ricerca di una specifica struttura, che avvenga tramite il semplice scorrimento della lista delle card, o attraverso la barra di ricerca, o altro. Anche in questo caso, non vengono espressi vincoli sulla data e sugli slot orari in cui deve avvenire la prenotazione. Ciò che rende particolarmente interessante questo task è la presenza di un "ostacolo" durante la procedura di prenotazione. Dopo aver selezionato liberamente la data e gli slot orari, inserito i propri dati personali e verificato l'indirizzo Email, gli utenti saranno informati, al momento del click sul pulsante "Paga"/"Prenota", che un altro utente è stato più rapido nel prenotare uno degli slot orari precedentemente selezionati. L'obiettivo di questo task è valutare se il sistema consenta agli utenti di superare questo ostacolo in modo semplice. I criteri di successo di questo task riguardano

dunque l'efficacia dell'interfaccia nel permettere agli utenti di affrontare la problematica nel modo più efficiente possibile, fornendogli indicazioni chiare per completare la prenotazione nonostante l'imprevisto.

- *Task 5.* Il quinto task richiede agli utenti di eseguire l'operazione inversa rispetto a quella analizzata nei precedenti task; richiede, ossia, la cancellazione di una prenotazione. L'obiettivo è valutare l'approccio degli utenti a questa procedura. Questo task risulta particolarmente interessante in quanto, considerando che Musictogo offre un servizio login-less, è cruciale comprendere se il sistema evidenzia correttamente questa caratteristica. Il task richiede di cancellare la penultima prenotazione effettuata. I criteri di successo di questo task riguardano dunque l'efficacia dell'interfaccia nel consentire agli utenti di cancellare una prenotazione. Ciò implica l'individuazione della pagina tramite la Navbar, e il completamento della procedura.

- *Task 6.* Il sesto e ultimo task richiede agli utenti di acquistare un servizio per incidere una canzone, presso uno studio di registrazione a libera scelta. Come nei precedenti task riguardanti la prenotazione di salette, sarà di rilevante interesse analizzare l'eventuale uso che verrà fatto del menu dei filtri, della barra di ricerca e della mappa, per filtrare la lista delle strutture visualizzate. I criteri di successo di questo task riguardano dunque l'efficacia dell'interfaccia nel consentire agli utenti di completare una procedura di acquisto, rispettando il vincolo, precedentemente descritto, del tipo di servizio da acquistare.

Per quanto riguarda le modalità di esecuzione dei test, si è optato per la metodologia "Think aloud", che consiste nell'incoraggiare i partecipanti a pensare ad alta voce mentre svolgono i diversi task, in modo da poter cogliere i loro pensieri e le loro sensazioni [39].

Le metriche quantitative che verranno raccolte sono il tempo impiegato per completare il task, il numero di tentativi compiuti e l'esito del task, che sarà categorizzato come Successo, Successo parziale o Fallimento, sulla base dei criteri di successo definiti per ogni task. Al contrario, le metriche soggettive saranno raccolte attraverso domande poste ai partecipanti prima, durante e dopo la sessione di test [39].

Prima dell'esecuzione dei test, il codice del progetto è stato adattato al fine di prevenire l'effettiva acquisizione delle prenotazioni da parte delle strutture. Per il resto, è stato impiegato un sistema basato sul medesimo codice destinato all'utilizzo in produzione da parte degli utenti effettivi.

Durante questa fase di preparazione alle sessioni di test, sono stati inoltre redatti il modulo di consenso alla partecipazione ed il questionario post-test. Mentre il modulo di consenso alla partecipazione si conforma agli standard di un consenso informato convenzionale, il questionario post-test, al contrario, può essere redatto

seguendo diverse metodologie. Il questionario post-test redatto per il presente lavoro di tesi adotta la *System Usability Scale* (SUS), una scala di usabilità ampiamente utilizzata per valutare la percezione dell'usabilità di un sistema. Il SUS consiste in un questionario standard composto da una serie di affermazioni, ciascuna delle quali richiede al partecipante di indicare il proprio livello di accordo su una scala da 1 a 5, dove 1 rappresenta "Fortemente in disaccordo" e 5 "Fortemente d'accordo". Si tratta di domande a risposta chiusa che generano, attraverso un calcolo matematico un punteggio compreso tra 0 e 100 [39]. I questionari post-test saranno oggetto di successive analisi per ricavare l'usabilità percepita media del sistema.

7.2.2 Fase di svolgimento

Come anticipato nella Sottosezione 7.2.1, il target di popolazione a cui far svolgere i test di usabilità è costituito da musicisti, sia dilettanti che professionisti, alla ricerca di un luogo appropriato per esercitarsi, sperimentare le proprie composizioni o registrare la propria musica. Il reclutamento è avvenuto attraverso la ricerca di individui appartenenti a tale target di popolazione, facendo affidamento sulla mia rete di conoscenze. Sono stati reclutati due miei conoscenti e tre loro amici. I cinque individui che hanno preso parte al test hanno un'età compresa tra i 22 e i 41 anni. Il campione è composto da quattro individui di sesso maschile e una di sesso femminile.

Lo script seguito durante tutte le sessioni di test, disponibile all'Appendice C, in calce al presente documento, rappresenta il copione che il ricercatore è tenuto a seguire il più fedelmente possibile, al fine di garantire consistenza tra le diverse sessioni di test. Esso ha inizio introducendo il test, descrivendone gli obiettivi e precisando che lo scopo della sessione è testare l'interfaccia, non l'utente che la utilizza, per poi descrivere cosa è Musictogo, ciò di cui si occupa, e invitare il partecipante a pensare ad alta voce durante lo svolgimento dei diversi task. Dopo aver fatto firmare il modulo di consenso, inizia la registrazione della sessione e vengono fatti svolgere i sei task previsti. Per ciascun task, insieme al compito da svolgere, viene fornito uno scenario, al fine di contestualizzarne l'esecuzione. Vengono chiesti, di tanto in tanto, pareri sulle funzionalità utilizzate, cercando di raccogliere eventuali perplessità, considerazioni, apprezzamenti e così via. Al termine dei sei task viene fatto compilare il questionario post-test e si chiede un feedback generale sull'esperienza d'uso avuta con l'applicazione. Al termine del feedback, la sessione si ritiene conclusa e la registrazione viene interrotta.

Sebbene una corretta esecuzione dei test di usabilità richieda la presenza di almeno due figure (il facilitatore e l'osservatore), data l'assenza di un collaboratore disponibile per supportarmi in questa attività, ho deciso di procedere assumendo, in momenti diversi, il ruolo sia di facilitatore che di osservatore. Le sessioni di test, durante le quali assumerò il ruolo di facilitatore, verranno registrate. Le registrazioni

consisteranno in materiale audio e video che sarà mia premura analizzare, in un secondo momento, assumendo il ruolo di osservatore.

7.2.3 Fase di analisi

Quella che segue è un'analisi dettagliata dei risultati emersi, per ogni task, dalle cinque sessioni di test di usabilità svolte. Dove non è stata fatta distinzione esplicita, non sono stati osservati approcci diversi ai singoli task in relazione all'ambiente utilizzato per svolgere le sessioni di test, sia esso Mobile o Desktop. A seguito dei task, verranno inoltre analizzati i risultati dei questionari post-test sottoposti ai partecipanti al termine delle sessioni.

Task 1

Tutti i partecipanti sono stati fin da subito in grado di interagire con il sistema alla ricerca di una struttura che rispettasse le richieste del task. Dei cinque partecipanti, quattro hanno utilizzato la barra di ricerca come unico metodo di filtraggio, mentre uno non ha utilizzato alcuna forma di filtraggio, limitandosi a scorrere la lista delle strutture. Due partecipanti hanno deciso di utilizzare la barra di ricerca solo dopo che, a seguito dell'apertura del menu dei filtri, non hanno trovato il filtro desiderato. All'interno della barra di ricerca è stato sempre inserito il termine "Torino"; seppure i risultati ottenuti tramite la barra di ricerca comprendano tutte le strutture di Torino, essa, in questo contesto, risulta poco efficiente, in quanto fa apparire, tra i risultati, non solo le strutture torinesi, ma anche tutte quelle strutture che hanno, nel nome o nell'indirizzo, il termine "Torino". La mappa, ovvero il componente che avrebbe permesso di filtrare con precisione le strutture nella zona geografica di Torino, non è stata utilizzata da nessun partecipante. Questo approccio al task, sebbene non abbia ripercussioni sui criteri di successo del task stesso, può essere preso come spunto per eventuali riflessioni future.

Incredibilmente, il partecipante che si è limitato a scorrere la lista alla ricerca di una card che avesse la dicitura "Torino" nello spazio dedicato all'indirizzo, è stato l'unico a trovare subito una struttura aperta la domenica pomeriggio. Tutti gli altri partecipanti hanno invece dovuto aprire le pagine di almeno due strutture prima di trovarne una che soddisfacesse i requisiti del task. La navigazione tra le pagine del sistema non ha causato alcun tipo di problema: per tutti i partecipanti è risultato subito naturale cliccare, o fare tap, sulle card delle varie strutture, in modo da avere accesso alle relative pagine; dei quattro partecipanti che hanno avuto la necessità di tornare in Homepage, tre hanno fatto uso del pulsante "Indietro" presente all'interno della pagina Struttura, uno ha utilizzato la funzionalità di navigazione nativa dello smartphone.

Non sono emerse particolari differenze nell'approccio al task tra le sessioni in cui è stato testato l'ambiente Mobile e quelle in cui è testato l'ambiente Desktop. L'unica problematica emersa in questo task è quella riscontrata da un partecipante nel corso di una sessione in ambiente Mobile, dovuta al fatto che, appena entrato all'interno della pagina Struttura, l'ha fatta scorrere in basso nascondendo la card con gli orari e facendo apparire le card delle salette. Ha quindi deciso di fare tap su una delle card delle salette, aprendone la relativa pagina e notando poi, dal calendario della procedura di prenotazione, la chiusura domenicale della struttura. Tornato indietro alla pagina Struttura, ha trovato la card con gli orari di apertura ed è stato in grado di continuare il task e completarlo senza ulteriori problemi.

La Tabella 7.1 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Mobile	1 min, 42 sec	2	Successo parziale
2	Mobile	0 min, 51 sec	1	Successo
3	Mobile	0 min, 34 sec	1	Successo
4	Desktop	1 min, 36 sec	1	Successo
5	Desktop	0 min, 56 sec	1	Successo

Tabella 7.1: Metriche quantitative, Task 1

Le metriche quantitative evidenziano come i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task. L'unico "Successo parziale", ovvero quello del partecipante con Id pari ad 1, è dovuto alla problematica dettagliatamente evidenziata poc'anzi.

Al termine del task, i feedback forniti dai partecipanti sono stati, nel complesso, abbastanza positivi. Un paio di partecipanti hanno notato e apprezzato il fatto che il sistema non dimenticasse il testo cercato nella barra di ricerca. Un partecipante ha elogiato la chiarezza delle informazioni presentate all'interno delle schermate esplorate. Entrambi i partecipanti che, prima di utilizzare la barra di ricerca, avevano tentato di filtrare tramite il menu dei filtri, hanno espresso critiche riguardo all'assenza dei filtri che avrebbero voluto utilizzare. Il primo ha lamentato l'assenza di un filtro per gli orari di apertura, mentre il secondo si è lamentato dell'impossibilità di filtrare per zona.

Task 2

Nel task precedente, nonostante la richiesta fosse di natura geografica, nessun partecipante ha utilizzato la mappa. In questo task, al contrario, tre dei cinque partecipanti ne hanno fatto uso. Di questi tre, due hanno utilizzato il pulsante di geolocalizzazione, mentre uno ha effettuato lo zoom sulla propria zona di residenza. Due dei cinque partecipanti, invece, hanno continuato a utilizzare la barra di ricerca. Dei tre partecipanti che hanno utilizzato la mappa, due si trovavano su piattaforma Mobile. Il pulsante che, su Mobile, consente di cambiare la visualizzazione da lista a mappa, è stato individuato in entrambi i casi dopo un primo momento in cui è stato aperto il menu dei filtri e ne sono stati selezionati alcuni. Il menu dei filtri, durante lo svolgimento di questo task, è stato utilizzato da quattro dei cinque partecipanti, con lo scopo di applicare il filtro sulle sale prove ad affitto orario, e, a volte, anche quello per le promozioni, sebbene non fosse richiesto. Tutti i partecipanti sono riusciti a individuare, attraverso i vari metodi che hanno utilizzato, una struttura nelle proprie vicinanze. In particolare, i due partecipanti che hanno utilizzato la mappa su piattaforma Mobile hanno interagito con la doppia schermata lista/mappa in modo molto fluido: entrambi hanno aperto la mappa e utilizzato il pulsante di geolocalizzazione; uno di loro è poi tornato alla schermata della lista e ha selezionato una struttura tra quelle filtrate, mentre l'altro ha fatto tap sul marker della mappa, accedendo direttamente da lì alla scheda della struttura.

Una volta aperte le pagine delle strutture, sono emerse le prime difficoltà. Un partecipante, su Desktop, non è stato in grado di portare a termine il task poiché, dopo aver guardato la pagina per un po' di tempo senza interagirci, ha dichiarato di non capire come procedere.

Due partecipanti, uno su Desktop e uno su Mobile, hanno inizialmente espresso sensazioni simili a quelle del partecipante che ha fallito il task ma, dopo alcuni istanti di sconforto, hanno provato ad interagire con il sistema riuscendo, entrambi, in totale autonomia, ad aprire la pagina Saletta per proseguire con il task.

All'interno delle pagine delle salette, i quattro partecipanti che sono riusciti a proseguire il task sono anche stati in grado di completare, senza incontrare particolari difficoltà, la procedura di prenotazione. Durante tale procedura, un partecipante ha chiesto come mai fosse disponibile solo il pagamento online, mentre un altro ha domandato come mai non fosse possibile, all'interno del calendario, prenotare oltre una cerca data. Queste domande avrebbero avuto risposta se i partecipanti avessero fatto uso delle "info-icon" presenti nella card di prenotazione. Un partecipante ha affrontato una situazione insolita: l'email contenente il codice OTP è stata erroneamente segnalata come spam. Sono stati necessari circa 30 secondi per individuare e risolvere il problema, consentendo così la conclusione del task.

La Tabella 7.2 riporta i dettagli delle metriche quantitative raccolte per questo

task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Mobile	1 min, 47 sec	1	Successo
2	Mobile	1 min, 37 sec	1	Successo
3	Mobile	2 min, 36 sec	1	Successo
4	Desktop	3 min, 08 sec	1	Fallimento
5	Desktop	1 min, 52 sec	1	Successo

Tabella 7.2: Metriche quantitative, Task 2

Le metriche quantitative evidenziano come quasi tutti i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task. L'unico "Fallimento", ovvero quello del partecipante con Id pari ad 4, è dovuto alla problematica dettagliatamente evidenziata poc'anzi.

Al termine del task, i feedback forniti dai partecipanti sono stati, nel complesso, abbastanza positivi. Uno dei due partecipanti che ha utilizzato la mappa su piattaforma Mobile ha apprezzato il fatto che la mappa funzionasse correttamente; ha evidenziato che questo particolare, a suo parere, non è affatto trascurabile, poiché ha dichiarato di essere abituato a interagire con siti web che spesso presentano mappe non funzionanti. Ha inoltre apprezzato la presenza del filtro geografico fatto dalla mappa e l'interscambiabilità tra lista delle strutture e mappa stessa. Un partecipante ha apprezzato la presenza, all'interno della card di prenotazione, della sezione di riepilogo. Il partecipante che non è riuscito a completare il task ha espresso critiche riguardo l'assenza di indicazioni chiare all'interno della pagina Struttura per comprendere come effettuare la prenotazione.

Task 3

A differenza del precedente task, nel Task 3 tutti i partecipanti sono riusciti a completare la procedura di prenotazione. Tutti e cinque i partecipanti hanno utilizzato il filtro "Promo". Inoltre, tre partecipanti hanno utilizzato anche il filtro "Sala prove - affitto orario". Un solo partecipante, almeno inizialmente, non ha utilizzato i filtri del menu, ma ha prima tentato, in vano, di scrivere "Promo" all'interno della barra di ricerca, ottenendo zero risultati; ha poi rimosso il testo inserito, aperto il menu dei filtri e filtrato per promo. Nonostante l'ampio uso fatto della mappa nel precedente task, due partecipanti hanno scelto di effettuare un

filtraggio geografico inserendo "Torino" nella barra di ricerca, nonostante non fosse richiesto dal task. Un partecipante, dopo aver filtrato per promo e per sale prove ad affitto orario, notando l'assenza della provincia di Lecce sulla mappa, ha tentato, in vano, di spostarne i confini sul Salento, ottenendo zero risultati. Successivamente, ha posizionato i confini della mappa nella zona di Latina, ottenendo diversi risultati.

I quattro partecipanti che in precedenza erano riusciti a completare la procedura di prenotazione, hanno concluso anche questo task senza incontrare problemi. Il partecipante che aveva fallito il Task 2 è riuscito, nonostante i miei timori, ad accedere alla pagina Saletta dopo aver interagito con il sistema all'interno della pagina Struttura, spostando il cursore del mouse in diverse posizioni fino a notare che, passando sopra le card delle salette, queste rispondevano attraverso la loro interattività. È riuscito, con molta soddisfazione, a completare, anche lui, il task. Durante la procedura di prenotazione, un partecipante ha sollevato una questione riguardante la durata dei singoli slot orari, osservando che questa volta erano di due ore, a differenza del task precedente in cui erano della durata di un'ora. Questa domanda avrebbe avuto risposta se il partecipante avesse fatto uso delle "info-icon" presenti nella card di prenotazione.

La Tabella 7.3 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Mobile	1 min, 54 sec	2	Successo parziale
2	Mobile	1 min, 31 sec	1	Successo
3	Mobile	2 min, 07 sec	1	Successo
4	Desktop	2 min, 36 sec	2	Successo parziale
5	Desktop	1 min, 08 sec	1	Successo

Tabella 7.3: Metriche quantitative, Task 3

Le metriche quantitative evidenziano come tutti i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task. I due "Successi parziali", ovvero quelli dei partecipante con Id pari ad 1 e 4, sono dovuti alle problematiche, dettagliatamente evidenziata poc'anzi, che hanno causato la momentanea assenza di risultati nella Homepage.

Anche al termine di questo task, i feedback forniti dai partecipanti sono stati, nel complesso, abbastanza positivi. Il partecipante che, a seguito dell'interazione con la mappa, ha ottenuto zero risultati, ha commentato che questa esperienza gli

ha permesso di comprendere meglio il funzionamento della mappa, anche se avrebbe preferito scoprirne prima le funzionalità. Il partecipante che nel precedente task non era riuscito ad avanzare oltre la pagina Struttura, dopo aver avuto successo in questo task, ha osservato che, a suo parere, la pagina Struttura non evidenzia a sufficienza l'interattività delle card delle salette; a suo parere, il titolo "Salette", della colonna di destra, non è abbastanza chiaro. Un partecipante ha scelto di prenotare una saletta di una struttura che era chiusa nel giorno della settimana in cui è stata effettuata la sessione di test. Ha apprezzato che la card di prenotazione ha evidenziato questa informazione, permettendogli così di prendere una decisione più consapevole. Un altro partecipante ha invece apprezzato la presenza, sparsa in più card, di entrambi i prezzi di riferimento della saletta, ovvero sia quello originale che quello scontato, poiché questo gli ha consentito di capire quanto effettivamente stesse risparmiando.

Task 4

La tanto apprezzata barra di ricerca è stata utilizzata da tutti e cinque i partecipanti al fine di trovare la struttura richiesta da questo task. Nel dettaglio, quattro partecipanti hanno subito utilizzato la barra di ricerca, mentre un partecipante ha prima compiuto un breve scroll della lista, per poi decidere di utilizzare, anche lui, la barra. Nessuno dei partecipanti ha incontrato difficoltà nell'eseguire i passi, già noti, di questo compito, presumibilmente grazie all'esperienza acquisita nei task precedenti. Questo aspetto non va sottovalutato, poiché indica che l'interfaccia è di facile apprendimento.

Tutti i partecipanti hanno rapidamente superato l'ostacolo rappresentato dagli slot già occupati da altri utenti. Pertanto, nonostante l'imprevisto, sono tutti riusciti a portare a termine la procedura di prenotazione.

La Tabella 7.4 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Mobile	1 min, 48 sec	1	Successo
2	Mobile	1 min, 40 sec	1	Successo
3	Mobile	2 min, 14 sec	1	Successo
4	Desktop	2 min, 18 sec	1	Successo
5	Desktop	1 min, 28 sec	1	Successo

Tabella 7.4: Metriche quantitative, Task 4

Le metriche quantitative evidenziano come tutti i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task.

Anche al termine di questo task, i feedback forniti dai partecipanti sono stati, nel complesso, abbastanza positivi. Un partecipante ha apprezzato che, nella procedura di prenotazione, nonostante l'imprevisto dello slot occupato da altri lo abbia costretto a tornare alla prima schermata della procedura, il sistema abbia mantenuto le sue informazioni e non gli abbia chiesto di nuovo di verificare l'indirizzo Email. Un altro partecipante ha osservato che potrebbe essere utile "congelare" gli slot selezionati da un utente, in modo che altri non possano prenotarli mentre sta già provando a prenotarli lui.

Task 5

Il primo obiettivo di questo task consiste nell'individuare l'elemento "Cancellazioni" nella Navbar, associarlo alla relativa funzionalità e premerlo per accedere alla pagina corrispondente. Tra i cinque partecipanti, tre hanno correttamente identificato tale elemento, mentre due l'hanno interpretato come un modo per accedere a una pagina contenente cancellazioni effettuate in passato. Mentre i primi tre partecipanti hanno aperto la pagina consapevolmente, il quarto lo ha fatto più per curiosità; il quinto, invece, non l'ha aperta affatto.

Dei primi tre partecipanti, due sono stati in grado di completare il task senza problemi, mentre uno ha incontrato delle difficoltà che hanno comportato il fallimento del task. Questo partecipante ha letto i dati richiesti dal form tramite i placeholder posti all'interno dei campi di input, ha inserito la propria Email, ma non ha capito cosa fosse il codice di eventuale disdetta. Ha quindi tentato di trovare risposte ai suoi dubbi all'interno della pagina Assistenza, cercando "Codice per eventuale disdetta" nella barra di ricerca, senza però avere successo. Dopo essere

tornato alla pagina Cancellazioni e aver reinserito la propria Email, ha comunicato di non comprendere cosa fosse il codice richiesto, affermando di non essere in grado di completare il task. Il quarto partecipante, dopo aver aperto la pagina per curiosità, ha letto le istruzioni indicate e ha proceduto con la cancellazione, riuscendo dunque a portare a termine il task. Il quinto partecipante ha invece deciso di aprire la pagina Assistenza, cercando in vano un aiuto nei Q&A attraverso la barra di ricerca. Successivamente ha dichiarato che, in una situazione reale, avrebbe risolto il problema visitando la pagina della struttura, ottenendo il numero di telefono e contattando direttamente il gestore.

La Tabella 7.5 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Mobile	1 min, 11 sec	1	Successo
2	Mobile	0 min, 56 sec	1	Successo
3	Mobile	2 min, 05 sec	2	Fallimento
4	Desktop	1 min, 09 sec	1	Successo parziale
5	Desktop	2 min, 07 sec	1	Fallimento

Tabella 7.5: Metriche quantitative, Task 5

Le metriche quantitative evidenziano risultati contrastanti tra i diversi partecipanti. I due "Fallimenti", ovvero quelli dei partecipanti con Id pari a 3 e 5, sono stati causati dalle due diverse problematiche dettagliatamente evidenziate poc'anzi. Al partecipante con Id pari a 3 è stato assegnato solo un "Successo parziale" nonostante abbia completato il task, poiché ha ammesso di essere entrato nella pagina Cancellazioni più per curiosità che per consapevolezza. Dati i risultati di questo test, si rendono urgenti delle modifiche volte a migliorare la procedura di cancellazione.

I feedback ricevuti al termine di questo task evidenziano le difficoltà riscontrate da alcuni partecipanti. Questi feedback costituiscono spunti di riflessione per le modifiche che andranno urgentemente apportate all'interfaccia. Uno dei due partecipanti che non sono riusciti a completare il task ha ribadito che, a suo parere, non si capisca cosa sia il codice. Il partecipante che è entrato all'interno della pagina per curiosità, ha fatto notare che forse il termine "Cancellazioni" potrebbe non essere appropriato, poiché interpretabile come riferimento alle cancellazioni passate. Entrambi i partecipanti che hanno cercato informazioni nella pagina Assistenza

hanno suggerito di aggiungere ulteriori Q&A, in quanto ritengono che attualmente la pagina sia carente di contenuti.

Task 6

Il task è stato completato con successo da tutti i partecipanti. Due di loro hanno individuato uno studio di registrazione semplicemente scorrendo la lista, mentre gli altri tre hanno prima utilizzato il menu dei filtri.

All'interno della pagina dei servizi, due partecipanti hanno inizialmente chiesto come mai non fosse presente un calendario di prenotazione. Tuttavia, i loro interrogativi sono stati risolti dalla lettura della card "Il servizio" presente nella pagina. Di per sé, la procedura di acquisto, differendo dalla procedura di prenotazione di salette solo per l'assenza del calendario e degli slot, non ha presentato alcun problema per nessuno dei partecipanti.

La Tabella 7.6 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Mobile	1 min, 51 sec	1	Successo
2	Mobile	1 min, 41 sec	1	Successo
3	Mobile	2 min, 13 sec	1	Successo
4	Desktop	1 min, 58 sec	1	Successo
5	Desktop	1 min, 42 sec	1	Successo

Tabella 7.6: Metriche quantitative, Task 6

Le metriche quantitative evidenziano come tutti i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task.

Al termine di questo task, gli unici feedback ricevuti sono stati quelli da parte dei partecipanti che hanno sollevato il dubbio sull'assenza del calendario. Secondo loro, potrebbe essere utile aggiungere del testo che chiarisca perché il calendario, solitamente presente e utilizzato dagli utenti, non sia disponibile per gli studi di registrazione. Questo testo, a loro detta, fornendo una spiegazione sull'assenza del calendario, consentirebbe agli utenti di effettuare acquisti in modo più consapevole.

Risultati SUS

Come anticipato nella Sottosezione 7.2.1, il questionario post-test redatto per il presente lavoro di tesi adotta la *System Usability Scale* (SUS), una scala di usabilità ampiamente utilizzata per valutare la percezione dell'usabilità di un sistema. Il SUS consiste in un questionario standard composto da una serie di affermazioni, ciascuna delle quali richiede al partecipante di indicare il proprio livello di accordo su una scala da 1 a 5, dove 1 rappresenta "Fortemente in disaccordo" e 5 "Fortemente d'accordo". Si tratta di domande a risposta chiusa che generano, attraverso un successivo calcolo matematico, un punteggio compreso tra 0 e 100 [39].

La Tabella 7.7 presenta, per ciascuna riga, i dati relativi al SUS di ogni partecipante: il livello di accordo per ogni singola affermazione e il SUS score calcolato. L'ultima riga della tabella contiene invece la media sei singoli SUS Score, e rappresenta il risultato finale.

Id Partecipante	**q1**	**q2**	**q3**	**q4**	**q5**	**q6**	**q7**	**q8**	**q9**	**q10**	**SUS Score**
1	5	1	5	1	5	1	5	1	5	1	100,0
2	5	1	5	1	5	1	4	1	5	1	97,5
3	4	1	4	2	5	2	5	2	5	2	85,0
4	3	1	4	1	5	1	4	2	3	1	82,5
5	5	1	5	1	5	1	5	1	4	1	97,5
									Risultato finale		**92,5**

Tabella 7.7: SUS Score, test di usabilità interfaccia

Il risultato finale risulta in SUS Score complessivo di 92,5. È dunque possibile affermare, a seguito di questo risultato, che i partecipanti hanno apprezzato il lavoro svolto per rinnovare il portale.

7.3 Sviluppi a seguito dei test

I risultati delle cinque sessioni di test di usabilità e il risultato del questionario SUS hanno validato la bontà del lavoro svolto per rinnovare l'interfaccia, tuttavia hanno anche evidenziato l'urgente necessità di alcune modifiche. Queste sessioni di test hanno anche offerto diversi spunti per potenziali miglioramenti. Gli spunti sono in gran parte derivati dai feedback degli utenti, ma a volte sono emersi anche

da considerazioni fatte sull'utilizzo, o sul mancato utilizzo, di alcuni elementi dell'interfaccia. Oltre agli spunti per possibili miglioramenti, i risultati del Task 5 hanno sottolineato la necessità di ristrutturare i componenti grafici che costituiscono la procedura di cancellazione.

7.3.1 Modifiche urgenti

Le modifiche resesi urgentemente necessarie sono quelle alla procedura di cancellazione. Al fine di migliorare la raggiungibilità della pagina per effettuare le cancellazioni, essa è stata rinominata da "Cancellazioni" a "Cancella una prenotazione". Per migliorare la comprensione da parte degli utenti, è stato leggermente esteso il testo che descrive, all'interno della pagina, come effettuare una cancellazione. Le modifiche appena descritte sono illustrate in Figura 7.1.

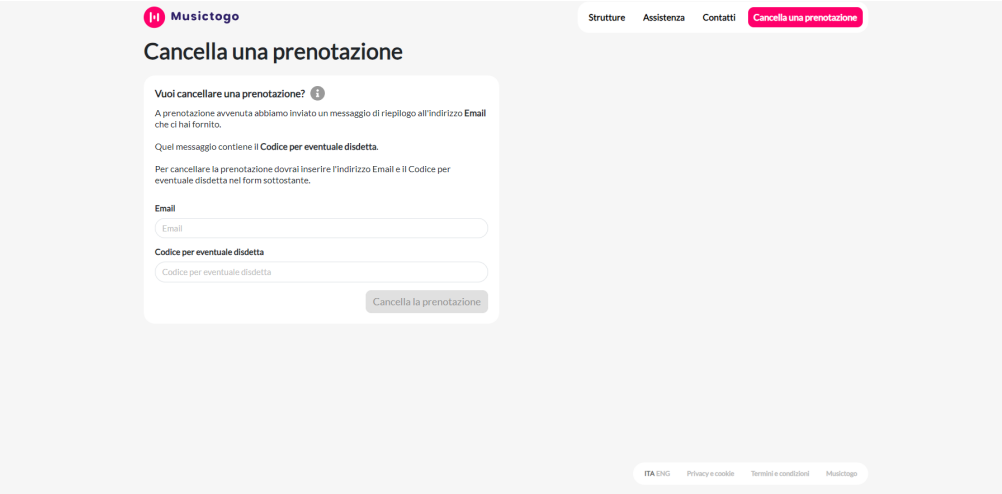

Figura 7.1: Pagina Cancella una prenotazione, nuovo portale revisionato

Al fine di evitare che gli utenti poco attenti non leggano le istruzioni, è stato inoltre inserito un modale che si apre non appena viene raggiunta la pagina. Questo modale, illustrato in Figura 7.2, spiega come effettuare la procedura di cancellazione, e rassicura gli utenti che, anche dopo la chiusura del modale, le istruzioni per effettuare la cancellazione saranno disponibili alla lettura all'interno della pagina.

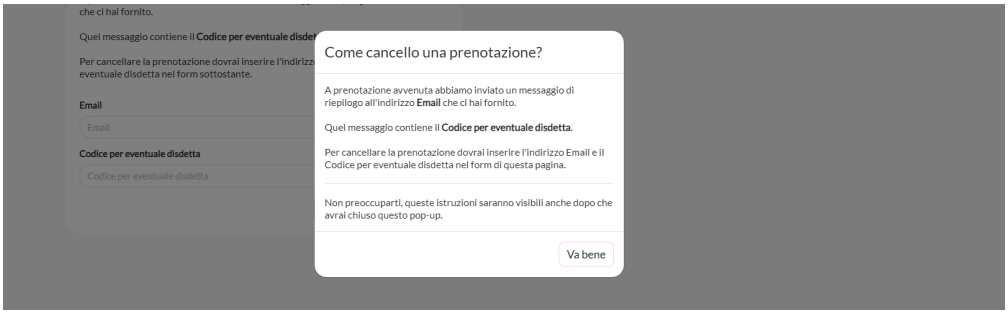

Figura 7.2: Modale informativo, Pagina Cancella una prenotazione, nuovo portale revisionato

7.3.2 Modifiche migliorative

In questa sottosezione verranno elencate le modifiche effettuate con all'interfaccia a partire dagli spunti derivanti dai feedback degli utenti e dalle considerazioni fatte sull'utilizzo, o sul mancato utilizzo, di alcuni elementi dell'interfaccia.

Interazioni con la mappa

I test hanno evidenziato come la mappa sia stata poco utilizzata. Molto spesso gli utenti le hanno preferito la barra di ricerca. Questo è un gran peccato dato che questo componente è molto potente e permette di effettuare filtraggi geografici, oltre che di localizzare l'utente. Dato che alcuni partecipanti al test, soprattutto nel primo task, hanno cercato all'interno dei filtri un modo per filtrare geograficamente, si è deciso di inserire un "suggerimento" all'interno del menu dei filtri. Il suggerimento, illustrato in Figura 7.3, consiste in un testo che spiega come, esplorando la mappa, sia possibile filtrare geograficamente la lista delle strutture.

Figura 7.3: Menu dei filtri, Homepage, nuovo portale revisionato

Si è cercato anche di migliorare l'usabilità del pulsante "Mappa"/"Lista" presente nell'interfaccia mobile. Come illustrato in Figura 7.4, i testi delle due varianti del

pulsante sono stati estesi, rendendoli più "azionabili". Al fine di dare un maggiore contrasto al pulsante quando esso è visibile sopra le card delle strutture, è stato colorato di un verde che ricorda la mappa.

Figura 7.4: Pulsante Mappa/Lista aggiornato, Homepage, nuovo portale revisionato

Titoli delle colonne in pagina Struttura

Alcuni partecipanti hanno riscontrato difficoltà nel capire, all'interno della pagina Struttura, quale fosse il passo successivo per effettuare una procedura di prenotazione. Al fine di cercare di arginare questo fenomeno, in Figura 7.5 viene proposta una soluzione che consiste nel dare, alle colonne, dei titoli che mettano più in risalto l'azione che si potrà compiere cliccando una delle card al loro interno.

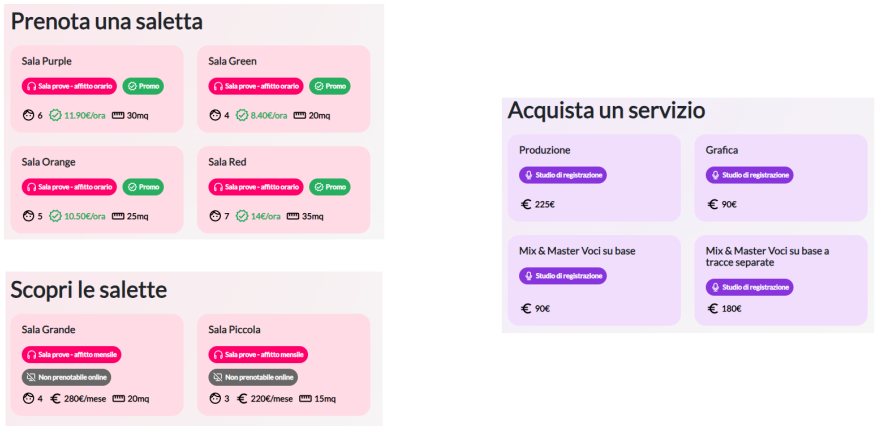

Figura 7.5: Colonne Salette e Servizi, pagina Struttura, nuovo portale revisionato

La dicitura "Salette" lascia il posto, in base al contesto, alle diciture "Prenota una saletta" oppure "Scopri le salette". La dicitura "Servizi" lascia invece il posto alla dicitura "Acquista un servizio".

Interazioni con le info-icon

Le info-icon, ovvero quelle icone che, al passaggio del cursore, mostrano a schermo un tooltip informativo, non sono mai state utilizzate dai partecipanti. Spesso, il loro utilizzo avrebbe risposto alle domande che i partecipanti hanno tentato di

fare al facilitatore. Sono state formulate varie ipotesi che spiegherebbero il perché del loro mancato utilizzo, ma l'ipotesi più accreditata è che fossero troppo piccole, e quindi poco visibili. Come illustrato in Figura 7.6, la dimensione delle icone informative è stata aumentata.

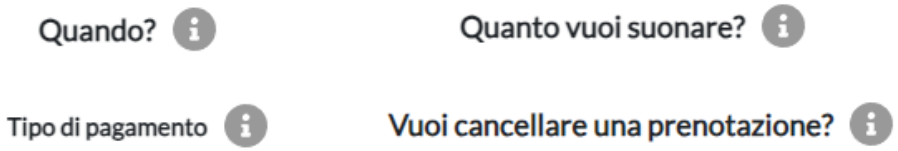

Figura 7.6: Info-icon, nuovo portale revisionato

Maggiori spiegazioni in pagina Servizio

Alcuni partecipanti hanno suggerito che potrebbe essere utile aggiungere, all'interno della card della procedura di acquisto di un servizio, del testo che chiarisca perché il calendario, solitamente presente e utilizzato dagli utenti, non sia disponibile per gli studi di registrazione. Questo feedback è stato preso alla lettera e, come illustrato in Figura 7.7, è stato aggiunto il testo richiesto.

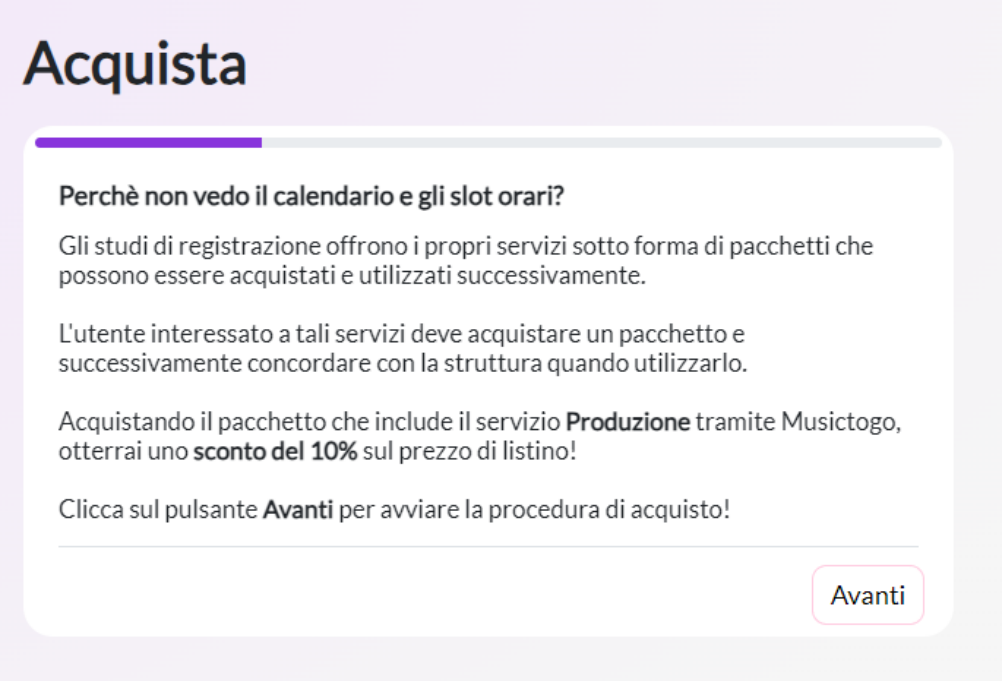

Figura 7.7: Testo informativo in card acquisto, pagina Servizio, nuovo portale revisionato

L'obiettivo del testo è fornire una spiegazione sull'assenza del calendario, consentendo così agli utenti di effettuare acquisti in modo più consapevole.

7.3.3 Validazione dell'interfaccia modificata

Per validare l'efficacia delle modifiche apportate, si è deciso di condurre tre ulteriori sessioni di test di usabilità, con le stesse modalità con cui sono state effettuate le precedenti cinque. Ai partecipanti verrà chiesto di completare solo un sottoinsieme dei task inizialmente pensati per validare l'interfaccia, ovvero i task 1, 2, 5 e 6, poiché ritenuti più rappresentativi per la successiva analisi sulle modifiche apportate.

Per le stesse motivazioni precedentemente descritte nella Sottosezione 7.2.1, due sessioni verranno effettuate attraverso la versione Mobile del nuovo portale, lasciando alla versione Desktop solo una sessione.

Sono stati reclutati un mio conoscente e due suoi amici. I tre individui che hanno preso parte al test hanno un'età compresa tra i 36 e i 45 anni. Il campione è composto da due individui di sesso maschile e una di sesso femminile.

Ne segue un'analisi dettagliata dei risultati emersi, per ogni task, dalle tre sessioni di test di usabilità svolte. A seguito dei task, verranno inoltre analizzati i risultati dei questionari post-test sottoposti ai partecipanti al termine delle sessioni.

Task 1

Tutti i partecipanti sono stati in grado di completare il compito richiesto senza problemi. Il partecipante con cui si è testato l'ambiente Desktop ha, fin da subito, effettuato un filtraggio geografico tramite la mappa. Non si è resa necessaria l'apertura del menu dei filtri. I due partecipanti che hanno interagito con l'ambiente Mobile, hanno utilizzato due approcci diversi per filtrare le strutture: il primo ha utilizzato la barra di ricerca inserendo il termine "Torino", il secondo ha invece preferito aprire il menu dei filtri. Dopo aver letto il suggerimento presente nel menu, ha premuto il filtro "Sale prove - affitto orario" per poi premere il pulsante che apre la mappa. All'interno della schermata della mappa, ha spostato i confini nella zona di Torino; è quindi tornato alla visualizzazione lista, filtrata.

Dopo aver filtrato la lista delle strutture nei diversi modi descritti, tutti e tre i partecipanti hanno dovuto aprire le pagine di almeno due strutture prima di trovarne una che soddisfacesse i requisiti del task.

La Tabella 7.8 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Desktop	0 min, 58 sec	1	Successo
2	Mobile	0 min, 47 sec	1	Successo
3	Mobile	1 min, 06 sec	1	Successo

Tabella 7.8: Metriche quantitative, Task 1 interfaccia modificata

Le metriche quantitative evidenziano come i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task.

Al termine del task, i feedback forniti dai partecipanti sono stati abbastanza positivi. Il partecipante che ha inizialmente aperto il menu dei filtri, ha apprezzato la presenza del testo informativo che, a suo dire, è stato utile per essere in grado di capire come mostrare solo le strutture di Torino. Il partecipante che ha utilizzato l'interfaccia Desktop, ha apprezzato molto la mappa.

Task 2

Tutti e tre i partecipanti hanno utilizzato la mappa. I due partecipanti che avevano già utilizzato la mappa nel precedente task, hanno immediatamente deciso di utilizzare la sua funzionalità di geolocalizzazione. Il partecipante che aveva utilizzato la barra di ricerca, ha inizialmente aperto il menu dei filtri ma, dopo aver

letto il suggerimento, ha deciso di aprire la mappa. Tuttavia, anziché utilizzare il pulsante di geolocalizzazione, ha preferito spostare i confini della mappa nella zona in cui risiede. Due partecipanti hanno utilizzato la info-window posta sopra ai marker per accedere alla pagina della struttura che desideravano aprire, mentre uno è tornato alla lista per poi aprire la nuova pagina da lì. Nessuno dei tre partecipanti ha sentito la necessità di utilizzare altre forme di filtraggio.

Nella pagina Struttura, tutti e tre i partecipanti hanno subito identificato le card delle salette come l'unico modo per proseguire con la procedura di prenotazione. Quando ho chiesto se il titolo della colonna avesse contribuito a comprendere cosa fare, due di loro hanno risposto che non è stato necessario, mentre uno ha affermato che il titolo gli è stato d'aiuto.

La procedura di prenotazione è stata completata senza problemi da tutti e tre i partecipanti. Nessuno ha incontrato difficoltà nel portare a termine il task. Tuttavia, le info-icon sono state utilizzate solo una volta, da un partecipante che non comprendeva il motivo per cui fosse disponibile solo il pagamento online.

La Tabella 7.9 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Desktop	1 min, 39 sec	1	Successo
2	Mobile	1 min, 45 sec	1	Successo
3	Mobile	2 min, 09 sec	1	Successo

Tabella 7.9: Metriche quantitative, Task 2 interfaccia modificata

Le metriche quantitative evidenziano come i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task.

Al termine del task, i feedback forniti dai partecipanti sono stati abbastanza positivi. Due partecipanti hanno apprezzato la presenza, all'interno della card di prenotazione, del riepilogo. Un partecipante ha apprezzato la presenza dei diversi step in cui è stata divisa la procedura.

Task 5

Questo task può essere considerato, senz'ombra di dubbio, il più importante tra i quattro scelti per queste sessioni di test. Il testo di questo task ha subito una piccola variazione: mentre nella sua forma originale richiede di cancellare "la penultima

prenotazione effettuata", ora, inevitabilmente, viene richiesto di cancellare "la precedente prenotazione effettuata" dato che è anche l'unica.

I due partecipanti che hanno utilizzato l'interfaccia Mobile, hanno subito aperto l'Offcanvas Menu alla ricerca della pagina per effettuare la cancellazione richiesta. L'elemento del menu che permette l'apertura di tale pagina è stato subito individuato. Aperta la pagina, hanno entrambi letto il modale informativo e sono stati poi in grado di effettuare la cancellazione senza problemi.

Il partecipante che ha utilizzato l'interfaccia Desktop ha notato la presenza dell'elemento in Navbar solo dopo essere inizialmente tornato nella pagina della struttura presso la quale aveva prenotato. Però, una volta aperta la pagina per effettuare la cancellazione, è stato abbastanza veloce a leggere il modale informativo e a completare la procedura.

La Tabella 7.10 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Desktop	1 min, 31 sec	2	Successo
2	Mobile	1 min, 18 sec	1	Successo
3	Mobile	1 min, 24 sec	1	Successo

Tabella 7.10: Metriche quantitative, Task 5 interfaccia modificata

Le metriche quantitative evidenziano come i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task.

I tempi sul task, a "causa" della necessità di leggere il modale introduttivo, sono risultati leggermente più alti rispetto a quelli delle sessioni dei primi test in cui il Task 5 è stato completato con successo. Tuttavia, il tasso di successo è aumentato, poiché tutti e tre i partecipanti sono stati in grado di portare a termine il task, mentre, nei primi test, relativamente al Task 5, si sono osservati due fallimenti e un successo parziale. Nonostante il partecipante che ha utilizzato l'ambiente Desktop non abbia immediatamente notato l'elemento della Navbar, pur avendo considerato due tentativi, si ritiene più appropriato non classificare il suo svolgimento del task come "Successo parziale", ma come "Successo". Ciò perché le azioni eseguite nel primo tentativo (apertura della pagina Struttura) non hanno impedito al partecipante di accedere comunque alla pagina per effettuare la cancellazione.

Al termine del task, i feedback forniti dai partecipanti sono stati abbastanza positivi. Un partecipante ha mostrato stupore nel sapere che questo task fosse

stato fonte di molti problemi nei precedenti test.

Task 6

Nonostante il Task 6 fosse andato molto bene, ottenendo cinque "Successo", si è deciso di testarlo di nuovo per validare la modifica fatta alla card di acquisto con l'aggiunta del testo introduttivo alla procedura.

Come avvenuto precedentemente, il task è stato completato con successo da tutti i partecipanti. Dopo aver letto il testo informativo, tutti e tre i partecipanti hanno acquistato con successo il servizio richiesto.

La Tabella 7.11 riporta i dettagli delle metriche quantitative raccolte per questo task.

Id partecipante	**Ambiente di test**	**Tempo sul task**	**Numero di tentativi**	**Esito del task**
1	Desktop	2 min, 05 sec	1	Successo
2	Mobile	2 min, 31 sec	1	Successo
3	Mobile	2 min, 24 sec	1	Successo

Tabella 7.11: Metriche quantitative, Task 6 interfaccia modificata

Le metriche quantitative evidenziano come tutti i partecipanti siano stati in grado di rispondere abbastanza bene alle richieste del task.

Al termine di questo task, quando ho chiesto ai tre partecipanti se avessero trovato utile il testo informativo iniziale, due hanno confermato che è stato utile per comprendere cosa stessero acquistando, mentre uno ha dichiarato che, sebbene non lo ritenesse di fondamentale importanza, ne ha apprezzato la presenza.

Risultati SUS

Anche in queste ulteriori tre sessioni di test, al termine dei task è stato sottoposto ai partecipanti un questionario SUS. I dati relativi ai questionari SUS compilati sono disponibili in Tabella 7.12.

Id Partecipante	**q1**	**q2**	**q3**	**q4**	**q5**	**q6**	**q7**	**q8**	**q9**	**q10**	**SUS Score**
1	5	1	5	1	5	1	5	1	5	1	100,0
2	5	1	5	1	5	1	5	1	5	1	100,0
3	5	1	5	1	5	1	5	1	5	1	100,0
									Risultato finale		**100,0**

Tabella 7.12: SUS Score, test di usabilità interfaccia modificata

Il risultato finale risulta in un sorprendente SUS Score complessivo di 100,0. Ai tre partecipanti, le funzionalità utilizzate sono piaciute davvero molto.

Capitolo 8

Conclusioni

L'obiettivo di questa tesi era il rinnovamento del portale Booking di Musictogo attraverso lo studio, lo sviluppo e la validazione di una nuova interfaccia utente basata su tecnologie all'avanguardia, che tenesse in considerazione gli aspetti messi in secondo piano dall'attuale portale, ovvero l'usabilità e la responsività.

Dopo aver studiato e analizzato l'attuale portale, elencandone i problemi di usabilità rilevati, al fine di non ripetere gli stessi errori con la nuova interfaccia, sono stati analizzati anche i mockup messi a disposizione da Musictogo, al fine di individuarne le carenze e le difformità rispetto al portale attuale. Per carenze, si sono intesi elementi, quali componenti specifici, funzionalità o pagine, non illustrati all'interno dei mockup ma presenti all'interno del portale attuale. Per difformità si sono, invece, intese, quelle funzionalità che, nel corso del tempo, sono state aggiunte, rivalutate o, addirittura, rimosse rispetto a quando furono sviluppati i mockup, e quegli elementi che, nonostante fossero rappresentati nei mockup, non erano disponibili a back-end nel formato desiderato o non erano disponibili affatto. I mockup sono stati, poi, oggetto di un'ulteriore analisi, questa volta con le medesime tecniche utilizzate per analizzare l'interfaccia dell'attuale portale, al fine di far emergere eventuali violazioni euristiche. A partire dai risultati delle due analisi, e tenendo in considerazione anche i risultati dell'analisi del portale attuale, è stata svolta un'opera di aggiornamento dei mockup con lo scopo di revisionare gli elementi presenti e crearne, talvolta, di nuovi, per compensare le carenze dei prototipi.

A seguito dell'aggiornamento dei mockup, è iniziata una fase di ricerca incentrata sulle tecnologie da adottare per lo sviluppo della nuova interfaccia. Le principali tecnologie impiegate, hanno incluso la libreria React e il framework Next.js. Conclusa la fase di sviluppo, l'interfaccia prodotta è stata valutata mediante cinque sessioni di test di usabilità (tre in ambiente Mobile, due in Desktop), al fine di ottenere una validazione del lavoro svolto. I risultati delle cinque sessioni di test di usabilità

hanno validato la bontà del lavoro svolto per rinnovare l'interfaccia e hanno offerto diversi spunti per potenziali miglioramenti; tuttavia hanno anche evidenziato l'urgente necessità di alcune modifiche. L'interfaccia è stata conseguentemente revisionata, con l'obiettivo di risolvere le maggiori criticità, e successivamente valutata nella sua versione aggiornata, attraverso tre ulteriori sessioni di test (due in ambiente Mobile, una in Desktop) che hanno confermato un miglioramento della qualità delle interazioni a seguito della revisione.

I risultati dei test hanno, nel complesso, confermato la bontà del lavoro svolto. La nuova interfaccia si adatta bene sia all'ambiente Desktop che a quello Mobile, risulta in un alto grado di usabilità percepita ed esteticamente è piaciuta molto alle persone che hanno avuto modo di usarla grazie ai test di usabilità. I test hanno, inoltre, fornito nuovi spunti per possibili miglioramenti futuri. Si ritiene per tanto di aver completato con successo gli obiettivi posti all'inizio del percorso di tesi.

8.1 Limitazioni

Le principali limitazioni affrontate durante questo percorso sono dovute all'assenza di collaboratori con cui abbia potuto svolgere, nell'ordine, le valutazioni euristiche del portale attuale, le valutazioni euristiche dei mockup iniziali, i test di usabilità del nuovo portale e i test di usabilità del nuovo portale a seguito delle modifiche. L'aver svolto queste attività in autonomia ne ha sicuramente influenzato i risultati; svolgendo queste attività in più persone, sicuramente si sarebbero potuti ottenere risultati migliori. Ciò nonostante, a seguito dei test di usabilità, tutti i partecipanti hanno apprezzato il lavoro svolto, le metriche quantitative hanno riportato dati molto soddisfacenti e i questionari SUS hanno fatto emergere un'alta percezione dell'usabilità del sistema da parte dei partecipanti.

8.2 Sviluppi futuri

Il progetto ha, a mio parere, ampi margini di miglioramento. Durante i test di usabilità sono emerse diverse funzionalità che potrebbe essere interessante sviluppare, come un filtro basato sugli orari di apertura delle strutture all'interno della Homepage.

Sempre in Homepage, vista la popolarità che ha avuto la barra di ricerca nelle diverse sessioni di test, si potrebbe pensare di potenziarne, in qualche modo, le funzionalità.

I test hanno fatto emergere una problematica che necessita dell'intervento diretto del team di Musictogo, cioè la carenza di Q&A all'interno della pagina Assistenza; i due partecipanti che hanno provato ad utilizzare quella funzionalità, non sono

infatti riusciti ad ottenere le risposte che cercavano perché non hanno trovato Q&A appropriati.

Andrebbe inoltre approfondito il "problema" delle info-icon: queste icone che, se puntate dal cursore del mouse, visualizzano a schermo un tooltip informativo, non sono mai state utilizzate nelle prime cinque sessioni di test, e nelle successive tre sessioni, a seguito di un ingrandimento delle dimensioni delle icone per renderle più visibili, sono state utilizzate solo una volta.

Lavorando in sinergia con il team che si occupa dello sviluppo del back-end, sarebbe interessante riuscire a portare nel sistema il concetto di "congelamento" degli slot selezionati, proposto da un partecipante durante i test di usabilità.

Appendice A

Lista delle violazioni euristiche rilevate nell'attuale portale di Booking

La presente appendice contiene la lista completa delle violazioni di usabilità rilevate applicando la valutazione euristica al portale di Booking attuale. Per ogni violazione verrà riportata l'euristica violata e i dettagli della violazione.

Si precisa che, a meno che non sia diversamente specificato, ogni violazione riguarda sia la versione Desktop che quella Mobile del portale. Le violazioni riguardanti esclusivamente la versione Mobile saranno contrassegnate con la dicitura "Mobile" nel contenuto del campo *Dove*. Allo stesso modo, si farà uso della dicitura "Desktop" per le violazioni riguardanti esclusivamente la versione Desktop.

La lista

1. *H1 Visibility of system status*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Feedback visivo nella mappa a seguito del filtraggio per provincia. Filtrando per una qual si voglia provincia, la mappa si sposta su quella provincia. Rimuovendo il filtro, la mappa rimane sull'ultima provincia selezionata.
    - *Perché*: Non è visibile, guardando la mappa, il cambiamento dello stato del sistema a seguito della rimozione del filtro.

- *Severità*: 2

2. *H1 Visibility of system status*

   - *Dove*: Homepage (Figura 2.1)
   - *Che cosa*: Filtrando per alcune provincie, ad esempio Caserta, la mappa indica la presenza di alcune strutture in tale provincia ma, al contrario, la lista delle strutture trova zero risultati.
   - *Perché*: Non si capisce se il sistema sia in uno stato di zero strutture trovate (come si evince dalla lista), oppure no (come si evince dalla mappa).
   - *Severità*: 3

3. *H1 Visibility of system status*

   - *Dove*: Homepage (Figura 2.1)
   - *Che cosa*: Interazioni con la mappa a seguito delle quali viene modificato lo stato di visibilità di alcuni marker, non trova corrispondente aggiornamento nella lista delle strutture.
   - *Perché*: Il sistema, a seguito dell'interazione dell'utente con la mappa, potrebbe presentarsi in uno stato di alta incoerenza tra mappa e lista.
   - *Severità*: 3

4. *H1 Visibility of system status*

   - *Dove*: Homepage (Figura 2.1)
   - *Che cosa*: L'applicazione del filtro per provincie non modifica il numero di marker visibili in mappa. La mappa, che a seguito dell'applicazione del filtro si limita a centrarsi sulla provincia selezionata, mantiene qual si voglia livello di zoom fosse presente prima del filtraggio. Ciò comporta, in caso di un basso livello di zoom, la visibilità di marker di strutture facenti parte di altre provincie e, in caso di un alto livello di zoom, l'esclusione dalla visualizzazione delle strutture più periferiche della provincia. La lista, invece, mostra sempre e soltanto tutte le strutture della provincia che si sta filtrando.
   - *Perché*: Il sistema, a seguito dell'applicazione del filtro per provincia, potrebbe presentarsi in uno stato di alta incoerenza tra mappa e lista.
   - *Severità*: 3

5. *H1 Visibility of system status*

- *Dove*: Homepage (Figura 2.1)
- *Che cosa*: L'applicazione di qual si voglia filtro, che non sia quello della provincia, non causa alcun feedback visivo sulla mappa.
- *Perché*: Il sistema, a seguito dell'applicazione di qual si voglia filtro, che non sia quello della provincia, potrebbe presentarsi in uno stato di alta incoerenza tra mappa e lista.
- *Severità*: 3

6. *H1 Visibility of system status*

   - *Dove*: Homepage (Figura 2.1)
   - *Che cosa*: I risultati che il sistema visualizza in lista a seguito dell'applicazione di qual si voglia filtro, che non sia quello promo, non danno alcuna indicazione sul perché siano usciti.
   - *Perché*: In certe circostanze, come ad esempio quando si effettua una ricerca del termine "Torino" tramite la barra di ricerca, i risultati (che, nell'esempio, si presume siano a Torino, ma che potrebbero anche trovarsi in altre città, in via Torino), mancano al loro interno di evidenze che ne giustifichino le ragioni per le quali il sistema ha scelto di visualizzarli. Ciò rende di fatto tutto molto ambiguo per gli utenti, che potrebbero non capire perché hanno ottenuto un certo risultato.
   - *Severità*: 3

7. *H1 Visibility of system status*

   - *Dove*: Homepage (Figura 2.1)
   - *Che cosa*: Le card delle strutture.
   - *Perché*: In certe circostanze, come ad esempio quando si effettua una ricerca del termine "Torino" tramite la barra di ricerca, i risultati (che, nell'esempio, si presume siano a Torino, ma che potrebbero anche trovarsi in altre città, in via Torino), mancano al loro interno di evidenze che ne giustifichino le ragioni per le quali il sistema ha scelto di visualizzarli, rendendo di fatto tutto ciò molto ambiguo.
   - *Severità*: 3

8. *H1 Visibility of system status*

   - *Dove*: Homepage (Figura 2.1)
   - *Che cosa*: La lista delle card delle strutture.

- *Perché*: L'utente non ha mai idea del numero di strutture visualizzate in lista.
- *Severità*: 2

9. *H1 Visibility of system status*

   - *Dove*: Modale prenotazione saletta (Figura 2.14)
   - *Che cosa*: L'assenza di un'indicazione riguardo al prezzo totale, basata sulle scelte di orario e modalità di pagamento.
   - *Perché*: L'utente, in fase di creazione della prenotazione, è all'oscuro del costo che il sistema gli addebiterà. Nel caso di una prenotazione online, l'importo esatto sarà rivelato solo dopo che l'utente sarà reindirizzato al servizio di pagamento terzo. Se l'utente sceglie di pagare in contanti, l'importo dovuto sarà reso noto solo quando si presenterà fisicamente presso la struttura.
   - *Severità*: 4

10. *H1 Visibility of system status*

    - *Dove*: Modale prenotazione saletta (Figura 2.14)
    - *Che cosa*: L'assenza di un feedback grafico durante il periodo d'attesa successivo alla pressione del pulsante "Invia codice di conferma".
    - *Perché*: L'invio di una mail è una procedura che potrebbe richiedere un numero non noto di secondi per essere completata. Durante questi secondi d'attesa è necessario mostrare graficamente all'utente che sono in corso delle operazioni.
    - *Severità*: 2

11. *H1 Visibility of system status*

    - *Dove*: Modale prenotazione saletta (Figura 2.14)
    - *Che cosa*: Il redirect in Homepage a seguito del completamento di una prenotazione con pagamento in contanti.
    - *Perché*: Dopo aver effettuato una prenotazione, sarebbe opportuno visualizzare un feedback di conferma. Il sistema non fornisce alcuna conferma visiva, ma semplicemente reindirizza l'utente in Homepage, lasciandolo in uno stato di incertezza riguardo l'avvenuto completamento della procedura. Almeno fino a quando non riceve l'email di conferma.
    - *Severità*: 2

12. *H2 Match between system and the real world*

- *Dove*: Sidebar e Homepage (Figura 2.1)
- *Che cosa*: L'utilizzo, a più riprese, delle diciture "Sala prove"/"Sale Prove" per indicare il contenuto della pagina.
- *Perché*: Tali diciture non riflettono accuratamente il contenuto della pagina, in quanto oltre alle sale prove sono presenti gli studi di registrazione. Suggerisco di utilizzare un termine più generico come "Strutture". In questo modo gli utenti avranno un'idea più chiara di cosa il sistema gli presenterà nella pagina.
- *Severità*: 2

13. *H2 Match between system and the real world*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Il click su alcuni radio button.
    - *Perché*: Per applicare i filtri "Tutte","Affitto mensile","Affitto ad ore" è possibile, tra le altre cose, cliccare sul testo corrispondente. Al contrario, cliccando sul testo corrispondente ai filtri "Studio di registrazione" e "Promo", viene applicato il filtro "Affitto ad ore". Di conseguenza il sistema applica un filtro diverso da quello desiderato dall'utente.
    - *Severità*: 4

14. *H2 Match between system and the real world*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: L'utilizzo del termine "Prenota" nei pulsanti che aprono le pagine delle strutture
    - *Perché*: Di per sé, il termine "Prenota" potrebbe considerarsi "eccessivo", in quanto associato ad un comportamento del sistema che non è quello di effettuare una prenotazione, ma è quello di aprire le pagine delle strutture all'interno delle quali si potranno poi scegliere le salette da prenotare. Però, dato che non tutte le strutture permettono di effettuare le prenotazioni (alcune perché, seppur sale prove, utilizzano Musictogo solo come vetrina, altre perché, in quanto studi di registrazione, permettono l'acquisto di pacchetti), il suo utilizzo è da considerarsi errato, in quanto può tendere l'utente in inganno sul tipo di struttura che aprirà cliccando il pulsante.
    - *Severità*: 3

15. *H2 Match between system and the real world*

    - *Dove*: Modali informativi in tutto il portale (come in figura 2.2)

- *Che cosa*: La scritta "Ok" dei modali
- *Perché*: La pressione del pulsante "Ok" comporta la chiusura del modale. Andrebbe associato a tale pulsante un termine (ad esempio "Chiudi") che sia coerente con il comportamento che ha il sistema dopo la pressione del pulsante
- *Severità*: 1

16. *H2 Match between system and the real world*

    - *Dove*: Pulsante "Affitta" di sale prove ad affitto mensile (Figura 2.22)
    - *Che cosa*: La scritta "Affitta" del pulsante
    - *Perché*: Ci si aspetterebbe che la pressione del pulsante "Affitta" comporti l'inizio di una procedura di affitto. Al contrario, comporta l'apertura di un modale. Andrebbe associato a tale pulsante un termine coerente con l'azione di apertura di un modale informativo.
    - *Severità*: 3

17. *H2 Match between system and the real world*

    - *Dove*: Modale prenotazione saletta (Figura 2.14)
    - *Che cosa*: Cambio di informazioni come email o nome a seguito della verifica della email
    - *Perché*: Il sistema permette, solo in parte, il cambiamento di informazioni come email o nome a seguito della verifica della email, creando una discrepanza tra ciò che l'utente crede di fare e ciò che effettivamente il sistema fa. In particolare, la mail di conferma prenotazione viene inviata al "vecchio" indirizzo email precedentemente validato, ma la prenotazione risulta a nome del nuovo nominativo.
    - *Severità*: 4

18. *H2 Match between system and the real world*

    - *Dove*: Modale prenotazione saletta (Figura 2.14)
    - *Che cosa*: La scritta "Conferma" del pulsante di prenotazione in caso di prenotazione in contanti
    - *Perché*: Il pulsante "Conferma" non conferma una prenotazione, ma ne crea una. Sarebbe più opportuno avere per tale pulsante un termine come "Prenota". Tale termine appare in linea con l'azione che viene eseguita premendo il pulsante.
    - *Severità*: 1

19. *H2 Match between system and the real world*

    - *Dove*:pagine Sale prove ad affitto orario prenotabili (Figura 2.11)
    - *Che cosa*: il formato del prezzo.
    - *Perché*: Il formato del prezzo, nelle sale prove non in promozione, è "X euro".Si intuisce che sia un prezzo "all'ora", ma non è presente un'indicazione chiara che lo confermi.
    - *Severità*: 3

20. *H2 Match between system and the real world*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: Il pulsante "Invia codice di conferma"
    - *Perché*: Le azioni svolte da questo pulsante non si limitano a quelle dichiarate nel suo nome. Esso, infatti, si occupa anche della validazione degli input dei sovrastanti campi. Il pulsante dovrebbe solo fare ciò che dichiara di fare. I campi dovrebbero "validarsi da soli" e "sbloccare" il pulsante una volta che tutti gli input sono considerati validi.
    - *Severità*: 2

21. *H2 Match between system and the real world*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: Il pulsante "Conferma"/"Paga"
    - *Perché*: Le azioni svolte da questo pulsante non si limitano a quelle di tentare di completare la procedura di prenotazione/acquisto. Esso, infatti, si occupa anche della verifica del codice di conferma inserito. Il pulsante dovrebbe solo fare ciò che dichiara di fare. Il codice di conferma andrebbe verificato da altri e il pulsante andrebbe "sbloccato" a verifica avvenuta.
    - *Severità*: 2

22. *H2 Match between system and the real world*

    - *Dove*: pagina Calendario (Figura 2.13)
    - *Che cosa*: Il pulsante "Oggi"
    - *Perché*: Il pulsante "Oggi" non fornisce una chiara indicazione di cosa comporti cliccare su di esso. Noto il contesto nel quale è inserito, si potrebbe presumere che riporti il calendario al giorno corrente, tuttavia, nelle viste settimanale e mensile, non ritorna al giorno attuale, ma alla settimana o al mese in corso.

- *Severità*: 2

23. *H3 User control and freedom*
    - *Dove*: Le schermate non raggiungibili tramite Sidebar
    - *Che cosa*: L'assenza del pulsante "Indietro"
    - *Perché*: Anche se sarebbe possibile utilizzare il pulsante "Indietro" del browser, servirebbe che il sistema, al suo interno, abbia a disposizione un uscita d'emergenza dalle schermate aperte per errore
    - *Severità*: 2
24. *H4 Consistency and standards*
    - *Dove*: Ovunque
    - *Che cosa*: Il colore dei pulsanti quando passano ad uno stato di "Focus"
    - *Perché*: Il colore di sfondo di qual si voglia pulsante nel portale cambia radicalmente quando il pulsante passa in una stato di "Focus". Ad esempio, il pulsante "Prenota" della homepage passa dal rosso al blu, creando un'inconsistenza rispetto alla palette di colori che ci si aspetterebbe abbia.
    - *Severità*: 1
25. *H4 Consistency and standards*
    - *Dove*: Homepage (Figura 2.1), pagina Cancellazioni (Figura 2.9), pagina Calendario (Figura 2.13)
    - *Che cosa*: Il "titolo" posto nella parte superiore di queste schermate
    - *Perché*: La struttura delle varie pagine del portale fa immaginare che il primo elemento visibile in alto sia il titolo della pagina. Però, nelle schermate elencate, il titolo sembra più una descrizione di cosa fare. Di conseguenza, questo elemento presenta un significato incoerente tra le diverse pagine.
    - *Severità*: 2
26. *H4 Consistency and standards*
    - *Dove*: pagina Cancellazioni (Figura 2.9)
    - *Che cosa*: Il testo "Codice di prenotazione"
    - *Perché*: La mail che l'utente riceve come conferma dell'avvenuta prenotazione presenta al suo interno un codice denominato "Codice per eventuale disdetta". All'interno della pagina Cancellazioni si fa riferimento a tale codice con la dicitura "Codice di prenotazione", creando un'inconsistenza che confonde gli utenti.

- *Severità*: 3

27. *H4 Consistency and standards*
    - *Dove*: pagina Cancellazioni Desktop (Figura 2.9)
    - *Che cosa*: La larghezza del contenitore bianco
    - *Perché*: In tutte le pagine, il contenitore bianco ha la stessa larghezza, qualsiasi sia il contenuto. All'interno della pagina Cancellazioni, tale contenitore è largo giusto lo spazio per contenere al suo interno gli elementi della pagina. Ciò crea un'inconsistenza grafica tra le varie pagine.
    - *Severità*: 1
28. *H4 Consistency and standards*
    - *Dove*: Le pagine raggiungibili tramite Sidebar
    - *Che cosa*: I titoli delle pagine
    - *Perché*: I titoli delle pagine descritti nella Sidebar non sono i titoli descritti a inizio pagine. Ne è un esempio il titolo "Contattaci" della Sidebar: aprendo la pagina, il titolo visualizzato è "Contatti".
    - *Severità*: 1
29. *H4 Consistency and standards*
    - *Dove*: Pagina Assistenza (Figura 2.5)
    - *Che cosa*: Il titolo della pagina
    - *Perché*: A differenza di tutte le altre pagine, questa pagina non presenta un titolo a inizio pagina.
    - *Severità*: 1
30. *H4 Consistency and standards*
    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Posizione della mappa rispetto al contenitore bianco.
    - *Perché*: Il modulo della mappa tende spesso a uscire fuori dai confini dettati dal contenitore bianco. Questo problema si manifesta, ad esempio, quando le operazioni di filtraggio restituiscono un solo risultato in lista. In questo caso, le dimensioni della pagina e del contenitore bianco si riducono per ospitare l'unico risultato, ma l'altezza della mappa rimane invariata, causandone così la sua fuoriuscita dal contenitore bianco. In Mobile è di default fuori dai confini bianchi. Ciò va contro le guidelines che prevedono una precisa organizzazione dei vari elementi di UI in dei confini grafici definiti.

- *Severità*: 1

31. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Il colore di testi associati ai marker.
    - *Perché*: Le icone dei marker sono di colore rosso, i testi associati ai marker sono invece di colore blu. L'associazione di questi due colori, in questo caso, non permette un alto grado di leggibilità del testo. Non vengono rispettate le buone pratiche di visual design inerenti i colori.
    - *Severità*: 2

32. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Feedback della mappa a seguito di operazioni di filtraggio.
    - *Perché*: Quando si applica il filtro per provincia, la mappa fornisce un feedback visivo centrando la vista sulla provincia scelta. Tuttavia, quando si disattiva il filtro per provincia o si applicano altri filtri sulla lista, la mappa non fornisce alcun feedback. Questo crea una discrepanza nei comportamenti della mappa in base ai filtri applicati.
    - *Severità*: 3

33. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Titolo pagina, placeholder barra di ricerca e messaggio nel caso di zero risultati.
    - *Perché*: Il titolo della pagina utilizza il termine "sala prove", la barra di ricerca utilizza il termine "struttura", il messaggio nel caso di zero risultati utilizza il termine "sala". Tutti questi termini si riferiscono alla stessa entità e dovrebbero essere quindi uniformati per evitare confusione.
    - *Severità*: 2

34. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: L'organizzazione gerarchica della pagina.

- *Perché*: Si violano le Best Practices di grid alignment in quanto non vi è una netta separazione dei differenti elementi che compongono la pagina. Ad esempio, il filtro per provincie è posizionato, sulla destra, nello stesso rigo del titolo della pagina. Gli altri filtri si trovano, invece, sulla sinistra nel rigo successivo. Per offrire all'utente una visione più chiara e complessiva dei filtri applicabili e/o applicati, sarebbe opportuno raggrupparli tutti insieme.
- *Severità*: 2

35. *H4 Consistency and standards*

    - *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
    - *Che cosa*: L'organizzazione gerarchica della pagina.
    - *Perché*: Si violano le Best Practices di grid alignment in quanto non vi è una netta separazione degli elementi descrittivi della struttura da quelli descrittivi delle sue salette. In particolare, gli orari di apertura descrivono la struttura, ma vengono ripetuti in ogni elemento della lista delle salette.
    - *Severità*: 1

36. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: La barra di ricerca.
    - *Perché*: La barra di ricerca del portale è un campo di input testuale, identificabile solo dal suo placeholder, che scompare non appena si inizia a digitare. Di solito, le barre di ricerca sono associate a icone, come una lente d'ingrandimento, che ne evidenzino le funzionalità. Questa mancanza può rendere meno intuitivo il riconoscimento della barra di ricerca.
    - *Severità*: 2

37. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1), modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: Il click sul testo di alcuni radio button.
    - *Perché*: Per applicare, in homepage, i filtri "Tutte", "Affitto mensile", "Affitto ad ore" è possibile cliccare sia sul pallino che sul corrispondente testo. Nella homepage e nei modali di prenotazione/acquisto sono presenti altri radio button, selezionabili però solo cliccando sul pallino. Cliccando sul testo non succede nulla.

- *Severità*: 3

38. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: La paginazione della lista.
    - *Perché*: Le varie pagine nelle quali la lista è divisa mostrano a volte quattro strutture in una pagina, altre volte tre, altre ancora addirittura una. Sarebbe opportuno mostrare, per ogni pagina che non sia l'ultima, sempre lo stesso numero di strutture.
    - *Severità*: 2

39. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Il passaggio da una pagina all'altra della lista.
    - *Perché*: Pratica comune vuole che, quando si cambia pagina, il browser debba visualizzare la nuova pagina partendo dal suo inizio; sarà poi l'utente a scorrere la pagina verso il basso fino alla sua fine. Sul portale invece, ciò non avviene: dopo che l'utente cambia la pagina della lista, rimane nella parte bassa della nuova pagina.
    - *Severità*: 3

40. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: L'applicazione di un filtro mentre è viene visualizzata una pagina della lista che non sia la prima.
    - *Perché*: Nel caso descritto, a seguito dell'applicazione di un qual si voglia filtro, la lista viene modificata. Sarebbe dunque opportuno far tornare la paginazione sulla pagina uno. Ciò non accade, in quanto il sistema rimane sulla pagina in cui si trovava prima dell'applicazione del filtro.
    - *Severità*: 4

41. *H4 Consistency and standards*

    - *Dove*: Homepage (Figura 2.1), pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
    - *Che cosa*: Il pulsante "Info"

- *Perché*: Il pulsante "Info" non ha l'aspetto di un pulsante. Spostando il mouse su di esso, sembra presentarsi più come un link in quanto il testo viene decorato con una sottolineatura.
- *Severità*: 3

42. *H4 Consistency and standards*

- *Dove*:pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: la posizione del pulsante "Info", del nome dell'elemento e dell'eventuale pulsante di prenotazione/affitto/acquisto
- *Perché*: La pagina della struttura organizza quasi tutti i suoi elementi in maniera totalmente diversa rispetto a come venivano organizzati nella homepage. I pulsanti "Info" e "Prenota" (o le sue varianti) non si trovano più dentro la card ma, rispettivamente, sopra di essa e alla sua destra. Il nome dell'elemento (che in homepage rappresenta il nome della struttura, e qui il nome della sala/servizio) non si trova più dentro la card, ma sopra di essa. Tutto ciò causa confusione.
- *Severità*: 2

43. *H4 Consistency and standards*

- *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: la visibilità del nome della saletta/servizio.
- *Perché*: Il nome della saletta/servizio andrebbe messo in risalto in quanto esso rappresenta un'informazione indispensabile. Viene invece messo a pari "livello" dei testi che descrivono le caratteristiche della saletta/servizio.
- *Severità*: 1

44. *H4 Consistency and standards*

- *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: il testo dell'eventuale pulsante di prenotazione/affitto/acquisto
- *Perché*: Mentre in homepage, per qualsiasi struttura, è presente il pulsante "Prenota", all'interno della pagina della struttura non sempre il testo del pulsante è, ove presente,"Prenota". A volte è "Compra", altre volte "Affitta", altre volte non c'è proprio. Questo crea un'inconsistenza e genera molta confusione.
- *Severità*: 3

45. *H4 Consistency and standards*

- *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: il testo degli orari di apertura.
- *Perché*: Gli orari di apertura delle strutture dovrebbero avere un formato "universale" applicabile a tutte le strutture. Nel portale, invece, ogni struttura ha gli orari scritti in un formato "personale". La presenza di formati diversi crea inconsistenza tra le pagine delle varie strutture.
- *Severità*: 2

46. *H4 Consistency and standards*

- *Dove*: pagine Sale prove ad affitto orario prenotabili (Figura 2.11)
- *Che cosa*: il formato del prezzo.
- *Perché*: Nelle sale prove in promozione, il prezzo è nel formato "X euro/h" mentre, senza apparente motivazione, il formato del prezzo nelle sale prove non in promozione è "X euro".
- *Severità*: 3

47. *H4 Consistency and standards*

- *Dove*: Ovunque
- *Che cosa*: il comportamento dei modali.
- *Perché*: In homepage è possibile chiudere i modali cliccando con il mouse fuori dall'area del modale, in tutte le altre pagine tale azione non chiude i modali aperti.
- *Severità*: 2

48. *H4 Consistency and standards*

- *Dove*: pagina Calendario (Figura 2.13)
- *Che cosa*: Le date passate del calendario
- *Perché*: Le date passate non un hanno un aspetto "disabilitato" che le differenzi graficamente dalle altre date. L'utente non nota differenze.
- *Severità*: 3

49. *H4 Consistency and standards*

- *Dove*: pagina Calendario (Figura 2.13)
- *Che cosa*: Le date

- *Perché*: Le date sono elementi con cui l'utente può interagire. Cliccando su una data (che non sia nel passato), si apre il modale di prenotazione. L'elemento grafico della data, di per sé, non da nessuna indicazione circa la sua interattività. Tale elemento fa affidamento su un elemento esterno, il "titolo" della pagina, per spiegare che è possibile cliccarci sopra. Inoltre, muovendo il mouse sopra la data, non si riceve alcun feedback, nemmeno il cambio del cursore in una mano (che rappresenta uno standard circa la possibilità di cliccare su un elemento grafico).
- *Severità*: 3

50. *H4 Consistency and standards*

    - *Dove*: pagina Calendario (Figura 2.13)
    - *Che cosa*: Gli slot occupati delle date
    - *Perché*: Muovendo il mouse su uno slot occupato, il cursore si trasforma in una mano (che rappresenta uno standard circa la possibilità di cliccare su un elemento grafico), ma cliccandoci sopra non succede nulla.
    - *Severità*: 3

51. *H4 Consistency and standards*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: I colori dei pulsanti
    - *Perché*: Dando diversi colori ai pulsanti, si tendono a legare stilisticamente pulsanti con lo stesso colore. In questo caso, come mai per la prima volta si vedono dei pulsanti ("Invia codice di conferma" e "Annulla") di colore blu? Perché hanno un colore diverso rispetto a quello del pulsante "Conferma"/"Paga"?
    - *Severità*: 2

52. *H4 Consistency and standards*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: La descrizione del significato del pulsante "Invia codice di conferma"
    - *Perché*: Di per sé, il pulsante "Invia codice di conferma" può andare bene. Però, per capire dove questo codice venga inviato (email o telefono) e, in generale, cosa si debba fare con il codice, è necessario guardare il placeholder del sottostante input form. Le istruzioni andrebbero esposte meglio, magari prima del pulsante, in modo da rispettare il naturale flusso di lettura, che va dall'alto al basso.

- *Severità*: 2

53. *H4 Consistency and standards*

    - *Dove*: Modale prenotazione saletta (Figura 2.14)
    - *Che cosa*: Le distanze di alcuni elementi.
    - *Perché*: Mentre alcuni elementi del modale sono tra loro ben spaziati, altri risultano attaccati (ad esempio, "Per quante ore?" e "Modalità di pagamento"). Come mai queste inconsistenze?
    - *Severità*: 2

54. *H4 Consistency and standards*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: I titoli dei modali.
    - *Perché*: Mentre il modale di prenotazione ha come titolo un'informazione riguardo la data scelta, il modale di acquisto ha un'informazione riguardo il nome del pacchetto scelto. Per mantenere una coerenza tra i due modali, bisognerebbe inserire nel titolo del modale di prenotazione il nome della saletta e spostare l'informazione della data all'interno del body del modale.
    - *Severità*: 1

55. *H4 Consistency and standards*

    - *Dove*: Homepage Mobile (Figura 2.4)
    - *Che cosa*: Il testo "Provincia" del menu a tendina
    - *Perché*: Non vengono rispettate le linee guida riguardo la spaziatura degli elementi di una schermata in quanto il testo "Provincia" è sovrapposto al titolo della pagina
    - *Severità*: 4

56. *H4 Consistency and standards*

    - *Dove*: Homepage Mobile (Figura 2.4)
    - *Che cosa*: Posizione dei filtri radio button rispetto al contenitore bianco.
    - *Perché*: I filtri radio button escono fuori dai limiti del contenitore bianco. Ciò va contro le guidelines che prevedono una precisa organizzazione dei vari elementi di UI in dei confini grafici definiti.
    - *Severità*: 4

57. *H4 Consistency and standards*

- *Dove*: Homepage Mobile (Figura 2.4)
- *Che cosa*: La dimensione delle card delle strutture.
- *Perché*: Le card delle strutture risultano piccolissime, poco leggibili. Le loro dimensioni non sono coerenti con le dimensioni dello schermo in cui vengono visualizzate.
- *Severità*: 3

58. *H4 Consistency and standards*

- *Dove*: Ovunque in Mobile
- *Che cosa*: La dimensione dei pulsanti e del loro testo.
- *Perché*: I pulsanti sono davvero tanti piccoli. Si rischia di non riuscire a leggere il testo. Le loro dimensioni non sono coerenti con le dimensioni dello schermo in cui vengono visualizzati.
- *Severità*: 3

59. *H4 Consistency and standards*

- *Dove*: Homepage Mobile (Figura 2.4), pagine Strutture Mobile (come Figure 2.16, 2.19, 2.24)
- *Che cosa*: Il colore del pulsante "Info".
- *Perché*: Il colore rosso del pulsante "Info" lo rende poco visibile nella card Mobile, in quanto spesso non più nella zona bianca della card, ma sulla foto della struttura.
- *Severità*: 3

60. *H4 Consistency and standards*

- *Dove*: Homepage Mobile (Figura 2.4)
- *Che cosa*: La vicinanza dei pulsanti "Prenota", "Info" e, quando presente, "Promo".
- *Perché*: I pulsanti non rispettano nessuna buona pratica circa il distanziamento di elementi interattivi. Non sono abbastanza spaziati. Sono praticamente appiccicati. Usando il dito per fare tap, il rischio di cliccare quello sbagliato è molto alto.
- *Severità*: 4

61. *H4 Consistency and standards*

- *Dove*: pagina Calendario Mobile (Figura 2.16)

- *Che cosa*: I componenti che controllano il calendario.
- *Perché*: I componenti che controllano il calendario non hanno alcuna spaziatura tra loro. Il rischio di cliccare quello sbagliato è alto
- *Severità*: 3

62. *H4 Consistency and standards*

- *Dove*: pagina Calendario Mobile (Figura 2.16)
- *Che cosa*: Il componente "Mese/Settimana/Giorno" che controlla il calendario.
- *Perché*: Il componente esce fuori dai limiti del contenitore bianco. Ciò va contro le guidelines che prevedono una precisa organizzazione dei vari elementi di UI in dei confini grafici definiti.
- *Severità*: 4

63. *H4 Consistency and standards*

- *Dove*: pagina Calendario Mobile (Figura 2.16)
- *Che cosa*: I box delle date con vista mensile e settimanale.
- *Perché*: Con il calendario impostato su visualizzazione mensile o settimanale, non si riesce a leggere del tutto il testo degli orari occupati all'interno dei box giornalieri poiché, non entrando del tutto nel box, essi vengono in parte tagliati. Ciò va contro le guidelines che prevedono una precisa organizzazione dei vari elementi di UI in dei confini grafici definiti.
- *Severità*: 3

64. *H5 Error prevention*

- *Dove*: pagina Cancellazioni (Figura 2.9)
- *Che cosa*: Il pulsante "Cancella"
- *Perché*: Non è presente, in alcuna forma, un'opzione di conferma prima di eseguire l'azione di cancellazione
- *Severità*: 3

65. *H5 Error prevention*

- *Dove*: pagina Cancellazioni (Figura 2.9)
- *Che cosa*: Il pulsante "Cancella".

- *Perché*: Non è presente alcuna forma di validazione degli input che prevenga il click del pulsante in caso di input non valido. Il pulsante è sempre abilitato alla pressione, anche se i form "Email" e "Codice Prenotazione" sono vuoti.
- *Severità*: 4

66. *H5 Error prevention*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: I marker della mappa
    - *Perché*: I marker sono tutti uguali. Non vi è alcuna distinzione tra marker che rappresentano tipi diversi di strutture. Ogni marker è identificato solo dal nome della struttura che, di per sé, non basta per indicarne il tipo. L'utente può benissimo cliccare su un marker pensando di aprire, ad esempio, uno studio, ritrovandosi però in una sala prove.
    - *Severità*: 3

67. *H5 Error prevention*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: La lista delle strutture
    - *Perché*: Le card della lista sono tutte uguali. Non vi è alcuna rappresentazione grafica che distingua tipi diversi di strutture. Ogni card è identificata solo dal nome della struttura che, di per sé, non basta per indicarne il tipo. L'utente può benissimo cliccare sul pulsante "Prenota" di una card pensando di aprire, ad esempio, uno studio, ritrovandosi, però, in una sala prove.
    - *Severità*: 3

68. *H5 Error prevention*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Le card di sale prove ad affitto orario
    - *Perché*: Le card delle strutture "Affitto ad ore" (così vengono definite nei filtri) sono tutte uguali. Non vi è alcuna rappresentazione grafica che distingua le sale prove ad affitto orario prenotabili online da quelle che, presenti solo come vetrina, non lo sono. L'utente può benissimo cliccare sul pulsante "Prenota" di una card pensando di poter prenotare, ritrovandosi, però, in una sala prove non prenotabile online.
    - *Severità*: 3

69. *H5 Error prevention*

    - *Dove*: pagina Calendario (Figura 2.13)
    - *Che cosa*: La possibilità di aprire il modale di prenotazione cliccando qualsiasi data futura
    - *Perché*: Ogni struttura decide, a partire dalla data odierna, il numero di giorni d'anticipo con cui un utente può prenotare una saletta. Il calendario, al contrario, consente l'apertura del modale di prenotazione e la compilazione dei sui campi qualsiasi sia la data futura cliccata. Alla pressione del pulsante "Prenota"/"Paga", l'utente scopre però che, se la data è successiva alla finestra di prenotazione scelta dalla struttura, la procedura non potrà andare a buon fine. Sarebbe bene impedire che l'utente commetta tali, involontari, errori di scelta della data disabilitando il click per date successive alla suddetta finestra temporale.
    - *Severità*: 4

70. *H5 Error prevention*

    - *Dove*: pagina Calendario (Figura 2.13)
    - *Che cosa*: La possibilità di aprire il modale di prenotazione cliccando su giorni della settimana in cui la struttura è chiusa
    - *Perché*: Ogni struttura può avere, durante la settimana, dei giorni di chiusura. Il calendario consente l'apertura del modale di prenotazione e la compilazione dei sui campi anche in giorni della settimana in cui la struttura è chiusa. Alla pressione del pulsante "Prenota"/"Paga", l'utente scopre però che, se la data corrisponde ad una giornata di chiusura della struttura, la procedura non potrà andare a buon fine. Sarebbe bene impedire che l'utente commetta tali, involontari, errori di scelta della data disabilitando il click per i giorni della settimana in cui la struttura è chiusa.
    - *Severità*: 4

71. *H5 Error prevention*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: I form di inserimento "Email", "Nome" e "Telefono".
    - *Perché*: I form non fanno alcun controllo di validazione sull'input inserito. Il controllo dell'input dei campi "Email" e "Telefono" è demandato al pulsante "Invia codice di conferma". Per l'input del campo "Nome" non vi è proprio alcun controllo, neppure sulla lunghezza della stringa. L'utente,

non avendo un feedback istantaneo su ciò che sta inserendo, è portato, erroneamente, a pensare di stare compilando i form con dati validi fino a quando la pressione del pulsante lo "contraddice". Andava "contraddetto" prima.

- *Severità*: 3

72. *H5 Error prevention*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: Il form di inserimento dell'ora.
    - *Perché*: Il form di inserimento dell'ora non fa alcun controllo di validazione sull'input inserito. Si possono inserire orari inesistenti (ad esempio 03:99), orari passati (nel caso di prenotazioni nella data odierna), orari in cui la struttura risulta chiusa, orari che, nell'istante di apertura del modale, risultano già occupati da altri e orari che permetterebbero di prenotare con troppo poco preavviso. Il controllo dell'input di questo campo è demandato al pulsante "Prenota"/"Compra". L'utente, non avendo un feedback istantaneo su ciò che sta inserendo, è portato, erroneamente, a pensare di stare compilando il form con dati validi fino a quando la pressione del pulsante lo "contraddice". Andava "contraddetto" (per quanto possibile) prima.
    - *Severità*: 3

73. *H5 Error prevention*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: Il menu a tendina di scelta della quantità di ore.
    - *Perché*: Il menu a tendina di scelta della quantità di ore non fa alcun controllo di validazione sull'input inserito. Si possono inserire quantità di ore che, ad esempio, farebbero sforare la prenotazione fuori dagli orari di apertura della struttura oppure creerebbero sovrapposizioni con prenotazioni già esistenti. Il controllo dell'input di questo campo è demandato al pulsante "Prenota"/"Compra". L'utente, non avendo un feedback istantaneo su ciò che sta inserendo, è portato, erroneamente, a pensare di stare compilando il form con dati validi fino a quando la pressione del pulsante lo "contraddice". Andava "contraddetto" (per quanto possibile) prima.
    - *Severità*: 3

74. *H5 Error prevention*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)

- *Che cosa*: Il form di inserimento del codice di conferma.
- *Perché*: Il controllo dell'input del form di inserimento del codice di conferma è demandato al pulsante "Prenota"/"Compra". L'utente, non avendo modo di far verificare l'input inserito tramite un pulsante apposito, è portato, erroneamente, a pensare di aver inserito un codice valido fino a quando la pressione del pulsante "Conferma"/"Paga" lo "contraddice". Andava "contraddetto" prima, tramite la verifica del codice attraverso la pressione di un pulsante dedicato.
- *Severità*: 3

75. *H5 Error prevention*

- *Dove*: pagina Calendario Mobile (Figura 2.16)
- *Che cosa*: I box delle date con vista mensile e settimanale.
- *Perché*: Con il calendario impostato su visualizzazione mensile o settimanale, non si riesce a leggere del tutto il testo degli orari occupati all'interno dei box giornalieri poiché, non entrando del tutto nel box, essi vengono in parte tagliati. Si riesce a intuire quale sia l'orario che termina l'occupazione dello slot, ma ciò comunque non consente all'utente di avere ben chiari quali siano gli slot occupati, aumentando ancora di più le già alte probabilità di commettere errori in fase di scelta dell'orario e della quantità di ore nel modale di prenotazione.
- *Severità*: 2

76. *H6 Recognition rather than recall*

- *Dove*: Homepage (Figura 2.1)
- *Che cosa*: La lista delle strutture
- *Perché*: Le card della lista sono tutte uguali. Non vi è alcuna rappresentazione grafica che distingua tipi diversi di strutture. Ogni card è identificata solo dal nome della struttura che, di per sé, non basta per indicarne il tipo. L'utente, per essere sicuro di aprire una struttura del tipo desiderato, deve applicare il relativo filtro o ricordare se ha aperto tale struttura in passato e, in tal caso, di che tipo fosse.
- *Severità*: 3

77. *H6 Recognition rather than recall*

- *Dove*: Homepage (Figura 2.1)
- *Che cosa*: Le card di sale prove ad affitto orario

- *Perché*: Le card delle strutture "Affitto ad ore" (così vengono definite nei filtri) sono tutte uguali. Non vi è alcuna rappresentazione grafica che distingua le sale prove ad affitto orario prenotabili online da quelle che, presenti solo come vetrina, non lo sono. L'utente, per essere sicuro di aprire una struttura prenotabile, deve ricordare se ha aperto tale struttura in passato e, in tal caso, se fosse prenotabile.
- *Severità*: 3

78. *H6 Recognition rather than recall*

- *Dove*: Homepage (Figura 2.1)
- *Che cosa*: L'indicazione sui filtri attivi
- *Perché*: Scorrendo la pagina, si perde l'informazione sui filtri attivati. L'utente deve quindi ricordare che filtri aveva attivato per ottenere i risultati che sta visualizzando.
- *Severità*: 2

79. *H6 Recognition rather than recall*

- *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: Le informazioni che erano visibili in homepage nel modale di "Info"
- *Perché*: Alcune informazioni sulla struttura, che in homepage erano contenute nel modale di "Info", in questa pagina non si ha modo di recuperarle. Bisogna ricordarle dalla homepage
- *Severità*: 3

80. *H6 Recognition rather than recall*

- *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: L'assenza di un indicazione chiara del tipo di struttura che si è aperta
- *Perché*: L'utente ha la possibilità di identificare il tipo di struttura basandosi sul testo del pulsante "Prenota"/"Affitta"/"Compra" o sulla sua assenza. Deve associare il testo del pulsante, o la sua assenza, a un tipo di struttura. Un'operazione mnemonica.
- *Severità*: 3

81. *H6 Recognition rather than recall*

- *Dove*: pagina Calendario (Figura 2.13)

- *Che cosa*: Le informazioni che erano visibili nelle pagine precedenti
- *Perché*: In questa pagina si perdono molte informazioni, rilevanti, sulla saletta e sulla struttura, che erano state illustrate nelle precedenti pagine. Ne sono esempi il prezzo, l'eventuale dettaglio di promo attiva, gli orari di apertura della struttura, il nome della saletta, i suoi mq. Tutti questi dettagli andrebbero riportati in questa pagina per ridurre il carico cognitivo dell'utente.
- *Severità*: 3

82. *H6 Recognition rather than recall*

- *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
- *Che cosa*: Le informazioni che erano visibili nelle pagine precedenti
- *Perché*: I modali non contengono, al loro interno, informazioni essenziali per il perfezionamento delle procedure di prenotazione/acquisto. Ne sono esempi il prezzo della saletta/servizio, l'eventuale dettaglio di promo attiva della saletta, gli orari di apertura della struttura, il nome della saletta, i mq della saletta. Tutti questi dettagli andrebbero riportati nei modali per ridurre il carico cognitivo dell'utente. Particolarmente grave è il fatto che si faccia prenotare/acquistare senza far vedere il prezzo.
- *Severità*: 4

83. *H6 Recognition rather than recall*

- *Dove*: Homepage Mobile (Figura 2.4)
- *Che cosa*: Il testo "Provincia" del menu a tendina
- *Perché*: Essendo il testo "Provincia" sovrapposto al titolo della pagina, è molto difficile da leggere. Di conseguenza, l'utente è costretto a memorizzare che l'opzione "Tutte" nel menu a discesa si riferisce alle province visualizzate.
- *Severità*: 3

84. *H7 Flexibility and efficiency of use*

- *Dove*: pagina Assistenza (Figura 2.5)
- *Che cosa*: Capacità di filtraggio delle Q&A
- *Perché*: Sarebbe utile poter effettuare una qual si voglia forma di filtraggio delle Q&A, magari tramite una barra di ricerca
- *Severità*: 1

85. *H7 Flexibility and efficiency of use*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Maggiore controllo sui filtri di "Tipo" attivi
    - *Perché*: Sarebbe utile avere più controllo sulla quantità di filtri "tipo" applicabili insieme. Ad esempio, potrebbe risultare utile filtrare insieme "Affitto ad ore" e "Promo".
    - *Severità*: 2

86. *H7 Flexibility and efficiency of use*

    - *Dove*: Homepage (Figura 2.1), pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
    - *Che cosa*: Salvataggio dei filtri attivi durante navigazione tra homepage e pagine strutture
    - *Perché*: Sarebbe utile salvare i filtri attivi nel passaggio dalla homepage alle pagine delle strutture in modo che, a seguito di un eventuale ritorno in homepage, l'utente trovi di nuovi i filtri precedentemente impostati
    - *Severità*: 2

87. *H7 Flexibility and efficiency of use*

    - *Dove*: Modali di prenotazione/acquisto (Figure 2.14, 2.18)
    - *Che cosa*: L'inserimento dei dati nel modale.
    - *Perché*: L'utente scopre se la combinazione di orario e quantità di ore inserita è valida solo dopo aver cliccato il pulsante "Conferma"/"Paga". Se tale combinazione risulta non valida e non esistono altre combinazioni disponibili per la data scelta, l'utente è costretto a chiudere il modale e aprirne uno nuovo per una data diversa. Questo processo comporta la perdita di tutti i dati precedentemente inseriti dall'utente e la conseguente invalidazione dell'indirizzo email, che dovrà essere nuovamente validato tramite l'invio di un nuovo codice OTP. Sarebbe pertanto opportuno conservare i dati personali precedentemente inseriti e validati dall'utente, in modo da ritrovarli già compilati e validati nel nuovo modale che verrà aperto.
    - *Severità*:

88. *H8 Aesthetic and minimalist design*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: La sovrapposizione di tanti marker.

- *Perché*: Specialmente con un basso livello di zoom, la mappa tende a sovrapporre i marker e i relativi testi. Ciò rende il contenuto della mappa poco piacevole alla vista. Servirebbe trovare una forma per esprimere tali informazioni in modo più condensato.
- *Severità*: 2

89. *H8 Aesthetic and minimalist design*

    - *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
    - *Che cosa*: Gli orari di orari di apertura della struttura
    - *Perché*: La ripetizione, per ogni saletta, degli orari di orari di apertura della struttura, è un'inutile occupazione di spazio. Si potrebbe evitare indicando gli orari a inizio pagina prima della lista delle salette
    - *Severità*: 1

90. *H8 Aesthetic and minimalist design*

    - *Dove*: pagina Calendario (Figura 2.13)
    - *Che cosa*: L'esplorazione del passato
    - *Perché*: Poter esplorare il passato è una funzionalità inutile. Per le date passate è possibile visionare gli slot orari occupati: tale funzionalità mette a disposizione dell'utente informazioni superflue e non necessarie.
    - *Severità*: 2

91. *H9 Help users recognize, diagnose and recover from errors*

    - *Dove*: pagina Cancellazioni (Figura 2.9)
    - *Che cosa*: I form di inserimento "Email" e "Codice Prenotazione".
    - *Perché*: Il sistema non esegue alcuna validazione sugli input forniti attraverso i form.. Di conseguenza, l'utente, non ricevendo alcun feedback sulle informazioni inserite, potrebbe erroneamente presumere che i dati forniti siano validi. Il sistema non fornisce assistenza all'utente nel rilevare e comprendere eventuali errori commessi durante la compilazione dei form.
    - *Severità*: 3

92. *H10 Help and documentation*

    - *Dove*: pagina Cancellazioni (Figura 2.9)
    - *Che cosa*: Email
    - *Perché*: Il sistema non fornisce istruzioni chiare su quale indirizzo email debba essere inserito.

- *Severità*: 3

93. *H10 Help and documentation*

    - *Dove*: pagina Cancellazioni (Figura 2.9)
    - *Che cosa*: Codice prenotazione
    - *Perché*: Il sistema non fornisce indicazioni chiare su cosa rappresenti e dove sia possibile ottenere il codice di prenotazione.
    - *Severità*: 3

94. *H10 Help and documentation*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: I filtri
    - *Perché*: Il sistema dovrebbe fornire una chiara indicazione dei filtri disponibili. Tuttavia, nel portale, esistono elementi che, sebbene intuitivamente sembrino avere la funzione di filtraggio, richiederebbero una descrizione più dettagliata. Un esempio è l'etichetta "Provincia" posizionata a sinistra del relativo menu a tendina. Sarebbe più appropriato utilizzare un'espressione come "Filtra per provincia".
    - *Severità*: 2

95. *H10 Help and documentation*

    - *Dove*: Homepage (Figura 2.1)
    - *Che cosa*: Menu a tendina "Provincia"
    - *Perché*: Il sistema non fornisce una spiegazione chiara sulla ragione per cui solo determinate province italiane sono visibili in questo menu a tendina. Una possibile interpretazione potrebbe essere che le strutture disponibili su Musictogo si trovano esclusivamente in quelle province. Tuttavia, questa ipotesi viene contraddetta dal fatto che, ad esempio, la provincia di Caserta, pur essendo presente nel menu, non presenta alcuna struttura in elenco.
    - *Severità*: 2

96. *H10 Help and documentation*

    - *Dove*: Homepage (Figura 2.1), pagina Sale prove ad affitto orario in promo
    - *Che cosa*: Pulsante "Promo"

- *Perché*: Il sistema non fornisce una spiegazione chiara su cosa comporti la pressione del pulsante "Promo" situato al centro delle card delle sale prove ad affitto orario in promozione e delle relative salette.
- *Severità*: 3

97. *H10 Help and documentation*

- *Dove*: pagine Strutture (Figure 2.12, 2.11, 2.17, 2.22)
- *Che cosa*: L'eventuale presenza del pulsante "Affitta"/"Compra"/"Prenota"
- *Perché*: Il sistema non fornisce alcuna spiegazione circa l'assenza o la presenza con un determinato testo del pulsante descritto.
- *Severità*: 3

98. *H10 Help and documentation*

- *Dove*: pagina Calendario (Figura 2.13)
- *Che cosa*: L'assenza di istruzioni per la prenotazione
- *Perché*: Il sistema non offre indicazioni dettagliate riguardo al numero massimo di giorni di anticipo e al numero minimo di minuti prima dei quali è possibile effettuare una prenotazione.
- *Severità*: 3

99. *H10 Help and documentation*

- *Dove*: Modale prenotazione saletta (Figura 2.14)
- *Che cosa*: Gli elementi del menu a tendina "Per quante ore?"
- *Perché*: Il sistema non offre motivazioni dietro la scelta di inserire, per strutture diverse, elementi diversi nel menu a tendina "Per quante ore?". Ad esempio, a volte si permette di scegliere in una lista contenente gli elementi 1,2,3,4,5, altre volte in una lista contenente soltanto 2,4.
- *Severità*: 3

100. *H10 Help and documentation*

- *Dove*: Modale prenotazione saletta (Figura 2.14)
- *Che cosa*: Le ore cliccabili nel menu a tendina "Per quante ore?"
- *Perché*: Il sistema non offre motivazioni che spieghino come mai, a volte, alcuni elementi del menu a tendina siano disabilitati e quindi non selezionabili.
- *Severità*: 3

101. *H10 Help and documentation*

- *Dove*: Modale prenotazione saletta (Figura 2.14)
- *Che cosa*: Le modalità di pagamento
- *Perché*: Il sistema non offre motivazioni che spieghino come mai non sempre è consentito all'utente scegliere la modalità di pagamento.
- *Severità*: 3

Appendice B

Lista delle violazioni euristiche rilevate nei mockup

La presente appendice contiene la lista completa delle violazioni di usabilità rilevate applicando la valutazione euristica ai mockup del 2021 messi a disposizione da Musictogo. Per ogni violazione verrà riportata l'euristica violata e i dettagli della violazione.

La lista

1. *H1 Visibility of system status*
    - *Dove*: pagina Saletta (Figura 4.5)
    - *Che cosa*: Il calendario
    - *Perché*: Sebbene il componente "Riepilogo" visualizzi il giorno e il mese selezionati, l'omissione dell'informazione relativa al mese nel calendario potrebbe generare una certa confusione tra gli utenti, almeno in una fase iniziale. Questo perché potrebbero sorgere difficoltà nel comprendere lo stato attuale del calendario.
    - *Severità*: 2
2. *H1 Visibility of system status*
    - *Dove*: pagina Struttura (Figura 4.3), pagina Saletta (Figura 4.5)
    - *Che cosa*: Il prezzo nelle card

- *Perché*: Nonostante i componenti di prenotazione di pagina Saletta rendano evidente l'esistenza di una promozione in corso, l'omissione di tale informazione nel prezzo visualizzato nella card di sinistra di pagina saletta e nelle card saletta di pagina Struttura, potrebbe generare una certa confusione tra gli utenti, almeno inizialmente. Questo perché potrebbero sorgere difficoltà nel capire se quella cifra rappresenti o meno il prezzo che il sistema considera scontato.
- *Severità*: 3

3. *H2 Match between system and the real world*

   - *Dove*: Homepage (Figura 4.1)
   - *Che cosa*: Le card delle strutture
   - *Perché*: Le card contengono informazioni circa i mq e il costo all'ora. Non è ben chiaro però a cosa facciano riferimento, dato che generalmente le strutture hanno diverse salette, ognuna delle quali con un suo costo orario e mq.
   - *Severità*: 3

4. *H2 Match between system and the real world*

   - *Dove*: Homepage (Figura 4.1)
   - *Che cosa*: L'utilizzo del termine "Prenota" nei pulsanti che aprono le pagine delle strutture
   - *Perché*: Di per sé, il termine "Prenota" potrebbe considerarsi "eccessivo", in quanto associato ad un comportamento del sistema che non è quello di effettuare una prenotazione, ma è quello di aprire le pagine delle strutture all'interno delle quali si potranno poi scegliere le salette da prenotare. Però, dato che non tutte le strutture permettono di effettuare le prenotazioni (alcune perché, seppur sale prove, utilizzano Musictogo solo come vetrina, altre perché, in quanto studi di registrazione, permettono l'acquisto di pacchetti), il suo utilizzo è da considerarsi errato, in quanto può tendere l'utente in inganno sul tipo di struttura che aprirà cliccando il pulsante.
   - *Severità*: 3

5. *H2 Match between system and the real world*

   - *Dove*: pagina Saletta (Figura 4.5)
   - *Che cosa*: Il pulsante di prenotazione

- *Perché*: Non è ben chiaro quale sia il pulsante da premere per prenotare, in quanto non è presente alcun pulsante con la dicitura "Prenota" nonostante in testo nel riepilogo reciti: "Se clicchi prenota accetti...".
- *Severità*: 3

6. *H4 Consistency and standards*
   - *Dove*: Homepage (Figura 4.1)
   - *Che cosa*: Il nome della struttura quando il mouse è sulla card
   - *Perché*: Il nome della struttura, quando il mouse si trova sulla sua card, risulta poco visibile. Si potrebbe pensare di mantenere il colore bianco, diminuendo la luminosità dell'immagine di sfondo.
   - *Severità*: 2
7. *H4 Consistency and standards*
   - *Dove*: Homepage (Figura 4.1)
   - *Che cosa*: La card "Non trovi quello che cerchi?"
   - *Perché*: L'elemento, in questo caso la card, che descrive il caso in cui "non trovi quello che cerchi" andrebbe mostrato quando la ricerca e/o le operazioni di filtraggio non producono risultati. Quando sono presenti risultati è poco frequente vedere un elemento di quel tipo.
   - *Severità*: 1
8. *H4 Consistency and standards*
   - *Dove*: Homepage (Figura 4.1)
   - *Che cosa*: La mappa
   - *Perché*: Rispetto alle mappe di pagina Struttura e Saletta, la mappa della Homepage nasce per permettere interazioni più articolate. L'assenza, in questa mappa, dei pulsanti + e -, che permettono di gestire lo zoom, può essere vista come una violazione degli standard che prevedono la presenza costante di tali pulsanti nelle mappe caratterizzate da un elevato grado di interattività.
   - *Severità*: 2
9. *H4 Consistency and standards*
   - *Dove*: Homepage Mobile(Figura 4.2)
   - *Che cosa*: La struttura della pagina

- *Perché*: La struttura della pagina Mobile risulta molto differente rispetto alla controparte Desktop. Per iniziare, in Mobile è assente la barra di ricerca. L'indicatore del numero di strutture presenti passa da "X strutture" nei mockup Desktop a "X risultati" in quelli Mobile. Nei Mockup Mobile sono presenti i filtri "Sale prova", "Studi di registrazione" che nei mockup Desktop non esistono; inoltre la mappa, che nei mockup Desktop era presente, in Mobile scompare. Ciò che ne risulta è un inconsistenza generale tra versione Desktop e quella Mobile del portale.
- *Severità*: 3

10. *H4 Consistency and standards*

    - *Dove*: pagina Struttura Mobile(Figura 4.4)
    - *Che cosa*: Il marker della mappa
    - *Perché*: Il marker, nella versione Mobile della mappa, risulta inconsistente rispetto al marker della versione Desktop.
    - *Severità*: 2

11. *H4 Consistency and standards*

    - *Dove*: pagina Struttura Mobile(Figura 4.4), pagina Saletta Mobile (Figura 4.6)
    - *Che cosa*: Il pulsante per tornare alla schermata precedente
    - *Perché*: Il pulsante per tornare alla schermata precedente, in Mobile, risulta inconsistente rispetto alla sua versione Desktop.
    - *Severità*: 2

12. *H4 Consistency and standards*

    - *Dove*: pagina Struttura Mobile(Figura 4.4)
    - *Che cosa*: La card con le informazioni sulla struttura
    - *Perché*: La card con le informazioni sulla struttura, nella sua versione Mobile, risulta inconsistente con la versione Desktop in quanto informazioni come il numero di telefono e il sito web non vengono illustrate
    - *Severità*: 2

13. *H4 Consistency and standards*

    - *Dove*: pagina Struttura Mobile(Figura 4.4)
    - *Che cosa*: La card "La struttura"

- *Perché*: La card “La struttura”, in Mobile, risulta in un colore di sfondo diverso rispetto alla sua controparte Desktop. Viene inoltre omesso il titolo della card.
- *Severità*: 2

14. *H4 Consistency and standards*
    - *Dove*: pagina Struttura Mobile(Figura 4.4)
    - *Che cosa*: Le card “Servizi”,“Regolamento” e “Prenotazione”
    - *Perché*: Queste card, presenti nella versione Desktop, scompaiono in quella Mobile, creando inconsistenza.
    - *Severità*: 2
15. *H4 Consistency and standards*
    - *Dove*: pagina Saletta Mobile(Figura 4.6)
    - *Che cosa*: Le card “La struttura” e “Strumentazione”
    - *Perché*: Queste card, presenti nella versione Desktop, scompaiono in quella Mobile, creando inconsistenza.
    - *Severità*: 2
16. *H4 Consistency and standards*
    - *Dove*: pagina Saletta Mobile(Figura 4.6)
    - *Che cosa*: La card con le informazioni sulla saletta
    - *Perché*: Questa card, nella sua versione Mobile, perde il titolo, presente invece nella versione Desktop, creando inconsistenza.
    - *Severità*: 2
17. *H4 Consistency and standards*
    - *Dove*: pagina Saletta (Figura 4.5)
    - *Che cosa*: Il calendario
    - *Perché*: Nel calendario, il giorno attualmente selezionato è il 16. La selezione viene evidenziata dal sistema mediante la colorazione in rosso del corrispondente numero. Tuttavia, questa modalità di rappresentazione non sembra sufficientemente enfatizzare la selezione del giorno. Per migliorare la visibilità della selezione, una possibile soluzione potrebbe essere quella di applicare una colorazione allo sfondo del riquadrino del numero selezionato.

- *Severità*: 2

18. *H5 Error prevention*

    - *Dove*: Homepage (Figura 4.1)
    - *Che cosa*: La lista delle strutture
    - *Perché*: Le card non distinguono tipi diversi di strutture. Ogni card è identificata da informazioni, come il nome della struttura, che, di per sé, non bastano per indicarne il tipo. L'utente può benissimo cliccare sul pulsante "Prenota" di una card pensando di aprire, ad esempio, uno studio, ritrovandosi, però, in una sala prove.
    - *Severità*: 3

19. *H5 Error prevention*

    - *Dove*: Homepage (Figura 4.1)
    - *Che cosa*: Le card di sale prove ad affitto orario
    - *Perché*: Le card non distinguono le sale prove ad affitto orario prenotabili online da quelle che, presenti solo come vetrina, non lo sono. L'utente può benissimo cliccare sul pulsante "Prenota" di una card pensando di poter prenotare, ritrovandosi, però, in una sala prove non prenotabile online.
    - *Severità*: 3

20. *H6 Recognition rather than recall*

    - *Dove*: pagina Saletta (Figura 4.5)
    - *Che cosa*: L'assenza di informazioni sulla struttura
    - *Perché*: Sarebbe stato utile ereditare qualche informazione dalla struttura, come il numero di telefono e l'indirizzo.
    - *Severità*: 2

21. *H6 Recognition rather than recall*

    - *Dove*: pagina Saletta (Figura 4.5)
    - *Che cosa*: L'assenza delle regole di prenotazione
    - *Perché*: Le regole di prenotazione sono illustrate nella pagina Struttura (Figura 4.3) come ultima card della lista di salette, ma non vengono riportate nella pagina Saletta, dove servono. L'utente deve ricordare queste informazioni dalla pagina precedente
    - *Severità*: 3

22. *H6 Recognition rather than recall*

    - *Dove*: pagina Conferma (Figura 4.7)
    - *Che cosa*: Le informazioni sulla prenotazione
    - *Perché*: Non è presente, nel riepilogo, il nome della saletta prenotata
    - *Severità*: 2

23. *H8 Aesthetic and minimalist design*

    - *Dove*: Homepage (Figura 4.1)
    - *Che cosa*: I colori delle card
    - *Perché*: Generalmente si tende ad associare tra loro elementi che, seppur diversi, presentano dei tratti in comune. Sicuramente il colore di sfondo è uno di questi tratti. Sebbene molto gradevoli alla vista, le card, così colorate, potrebbero trarre in inganno gli utenti. Card diverse, tra loro scorrelate ma aventi per qualche motivo lo stesso colore di background, potrebbero venire, erroneamente, associate.
    - *Severità*: 3

24. *H8 Aesthetic and minimalist design*

    - *Dove*: Homepage (Figura 4.1), pagina Struttura (Figura 4.3)
    - *Che cosa*: Il pulsante "Prenota" delle card
    - *Perché*: Il pulsante "Prenota" viene ripetuto in ogni card. È sempre visibile e non aggiunge nulla alla card. Si potrebbe pensare di rimuovere questo pulsante, rendendo tutta la card cliccabile. Potrebbe essere interessante mantenerlo, come "fake button" puramente a scopo informativo, quando il mouse va sulla card, mantenendo però la card totalmente cliccabile.
    - *Severità*: 1

25. *H8 Aesthetic and minimalist design*

    - *Dove*: pagina Struttura (Figura 4.3)
    - *Che cosa*: Le regole di prenotazione nella lista di salette
    - *Perché*: Le regole di prenotazione sono un'informazione necessaria nella pagina Saletta, dove avviene la procedura di prenotazione. In questa pagina risultano superflue.
    - *Severità*: 2

26. *H8 Aesthetic and minimalist design*

- *Dove*: pagina Struttura (Figura 4.3)
- *Che cosa*: La card “La struttura”
- *Perché*: Generalmente si tende ad associare tra loro elementi che, seppur diversi, presentano dei tratti in comune. Sicuramente il colore di sfondo è uno di questi tratti. La card “La Struttura” e le card delle salette presentano lo stesso colore di sfondo. Sono card dal significato diametralmente opposto che condividono un colore di sfondo diverso rispetto a quello del resto delle card. Dargli lo stesso colore di sfondo potrebbero trarre in inganno gli utenti.
- *Severità*: 2

27. *H8 Aesthetic and minimalist design*

    - *Dove*: pagina Struttura (Figura 4.3)
    - *Che cosa*: La card “Servizi”
    - *Perché*: La card da l’impressione che la sua dimensione dipenda dal numero di servizi disponibili. Nel caso in cui fossero presenti molti servizi, le sue dimensioni esploderebbero, oscurando, di fatto, i componenti dell’interfaccia sotto di lei. Sarebbe meglio poter “controllare l’esplosione” magari tramite l’utilizzo di una card che si espanda/comprima a comando.
    - *Severità*: 2

28. *H8 Aesthetic and minimalist design*

    - *Dove*: pagina Saletta (Figura 4.5)
    - *Che cosa*: Le card “Strumentazione”
    - *Perché*: Le card danno l’impressione che, nei casi peggiori, potrebbero diventare troppe. In tal caso oscurerebbero, di fatto, i componenti dell’interfaccia sotto di loro. Sarebbe meglio poter “controllare l’esplosione” magari tramite l’utilizzo di una singola card che si espanda/comprima a comando.
    - *Severità*: 2

29. *H10 Help and documentation*

    - *Dove*: pagina Saletta (Figura 4.5)
    - *Che cosa*: I componenti di prenotazione
    - *Perché*: Le regole di prenotazione andrebbero spiegate all’utente negli spazi della pagina dedicati ai componenti di prenotazione saletta.
    - *Severità*: 3

Appendice C

Script del moderatore - Test di usabilità

Introduzione

"Ciao, *<nome del partecipante>*, grazie per essere qui ad aiutarci. Sono Antonio e oggi siamo qui per provare la nuova interfaccia del portale Booking di Musictogo. Eventuali problemi o errori che dovessero emergere sono colpa dell'applicazione, non tua. Stiamo testando la nuova interfaccia, non te! Il nostro obiettivo con questi test è comprendere come gli utenti interagiscono con l'applicazione e come la percepiscono. Questo ci aiuterà a individuare cosa funziona bene e cosa potrebbe essere migliorato. I dati raccolti durante questo test saranno utilizzati per migliorare l'interfaccia in modo da garantire che sia facile da usare per tutti gli utenti. La tua opinione è preziosa per noi; ti ringrazio a nome di Musictogo per il tuo contributo."

"Ti riassumo brevemente cosa è Musictogo e ciò di cui si occupa. Musictogo è la prima piattaforma italiana di prenotazione nel settore musicale che permette a strutture come sale prove e studi musicali di offrire ai propri clienti un servizio login-less di prenotazione di fasce orarie in totale autonomia attraverso il proprio portale di Booking."

"Durante il test, ti invito a condividere ad alta voce i tuoi pensieri mentre utilizzi il portale. Questo ci permetterà di comprendere il tuo modo di interagire con l'interfaccia e di individuare eventuali problematiche o difficoltà che potresti riscontrare. Ricorda che non esistono risposte giuste o sbagliate durante questo test; stiamo solo cercando di capire le tue sensazioni mentre usi il portale."

"Per favore, compila questo documento per autorizzarci a registrare le operazioni che eseguirai con l'applicazione."

[*fornire il modulo di consenso*]

"Hai domande prima di iniziare?"

Inizia la registrazione

"Questa è la homepage del portale, guardala per 5-10 secondi. Puoi esplorare la mappa, ma non premere nulla."

Partecipante guarda la pagina.

Per ciascuno dei seguenti task, descrivi prima il contesto e poi fornisci il task da completare. Durante ciascun task, per incoraggiare l'utente a condividere i propri pensieri, puoi chiedere:

- "Puoi condividere i tuoi pensieri con noi mentre utilizzi l'applicazione?"
- "Cosa stai pensando in questo momento mentre utilizzi l'applicazione?"

Alla fine del task, puoi chiedere, in base a quanto accaduto, una/due domande come:

- "Hai riscontrato difficoltà durante questo task? In caso affermativo, quali?
- "Cosa ti piace di questa funzionalità dell'applicazione?"
- "C'è qualcosa che vorresti vedere migliorato in questa funzionalità?"
- "Sei riuscito a completare il task come previsto? Perché/perché no?"
- "Come valuti la facilità d'uso dell'applicazione durante questo task?"
- "C'è qualcosa che non hai capito durante questo task?"

T1
Scenario: "Tra qualche giorno ti recherai a Torino per affari e, anche se sarai in trasferta, vorresti occupare il pomeriggio di domenica suonando."
Task: "Trova, su Torino, una sala prove economica che sia aperta domenica pomeriggio."

T2
Scenario: "Sei a casa, stai ascoltando musica e ti viene voglia di suonare un po' con la tua band."

Task: "Prenota una saletta di una sala prova nelle tue vicinanze per quando ti è più comodo."

T3
Scenario: "La scorsa settimana hai utilizzato il servizio di Musictogo con una saletta in promo che ti ha permesso di approfittare del 30% di sconto sulla prenotazione. Vuoi utilizzare di nuovo il servizio promo per approfittare di altri sconti."
Task: "Trova una struttura con salette in promo e prenota una sua saletta per quando ti è più comodo."

T4
Scenario: "Sei intenzionato a prenotare una saletta della struttura chiamata Musictogo."
Task: "Prenota una saletta della struttura Musictogo per quando ti è più comodo."

T5
Scenario: "Avevi prenotato una saletta ma ti sei ricordato di avere un impegno."
Task: "Cancella la penultima prenotazione effettuata."

T6
Scenario: "Devi registrare il tuo ultimo singolo presso uno studio di registrazione."
Task: "Prenota una sessione di registrazione presso uno studio di registrazione."

I task sono terminati

"Grazie per aver partecipato al test. Vorremmo conoscere la tua opinione sull'applicazione e sulla tua esperienza d'uso. Potresti gentilmente compilare questo questionario? Grazie mille!"

[*fornire il questionario SUS*]

L'utente completa il questionario SUS.

"Hai incontrato delle difficoltà con qualche task durante il test? C'è qualcosa che ti è piaciuto particolarmente o che pensi dovrebbe essere migliorato?"

Risposta dell'utente.

"Apprezziamo molto il tuo feedback, perché ci aiuta a perfezionare l'applicazione per tutti gli utenti. Grazie ancora per aver partecipato!"

Termina la registrazione

Bibliografia

[1] Just Eat Italy S.r.l. *Just Eat*. 2024. URL: https://www.justeat.it/ (cit. a p. 1).

[2] eBay Inc. *eBay*. 2024. URL: https://www.ebay.it/ (cit. a p. 1).

[3] MUSICTOGO S.R.L. *Musictogo*. 2024. URL: https://www.musictogo.it/ (cit. a p. 1).

[4] MUSICTOGO S.R.L. *Portale Booking*. 2024. URL: https://booking.musictogo.it/ (cit. a p. 1).

[5] Meta Platforms Inc. *React*. 2024. URL: https://react.dev/ (cit. alle pp. 2, 6, 67).

[6] Vercel Inc. *Next.js by Vercel - The React Framework*. 2024. URL: https://nextjs.org/ (cit. alle pp. 2, 70).

[7] Mark Otto e Jacob Thornton. *Bootstrap · The most popular HTML, CSS, and JS library in the world.* 2024. URL: https://getbootstrap.com/ (cit. alle pp. 6, 74).

[8] Przemyslaw Baraniak. *What is a User Flow – Everything You need to Know*. Ago. 2020. URL: https://uxmisfit.com/2020/08/17/what-is-a-user-flow-everything-you-need-to-know/ (cit. a p. 12).

[9] Alan Dix, Janet Finlay, Gregory Abowd e Russell Beale. *Human Computer Interaction*. 3ª ed. Pearson Prentice Hall (cit. alle pp. 22–24, 26, 92, 93).

[10] Alberto Monge Roffarello. *Evaluation: Introduction and Heuristics*. 2023. URL: https://polito-hci-2023.github.io/materials/slides/07-heuristic-evaluation.pdf (cit. alle pp. 22–24, 26).

[11] Figma Inc. *Figma*. 2024. URL: https://www.figma.com/ (cit. a p. 30).

[12] Alphabet Inc. *Google maps*. 2024. URL: https://maps.google.com/ (cit. alle pp. 48, 49, 52).

[13] Alphabet Inc. *Angular*. 2024. URL: https://react.dev/ (cit. a p. 67).

[14] Evan You. *Vue.js*. 2024. URL: https://vuejs.org/ (cit. a p. 67).

[15] Giuseppe Speranza. *React vs Angular vs Vue.js: cosa scegliere?* 2022. URL: https://aulab.it/notizia/323/react-vs-angular-vs-vuejs-cosa-scegliere/ (cit. a p. 67).

[16] Maximilian Schwarzmüller. *Next.js & React - The Complete Guide*. 2022. URL: https://www.udemy.com/course/nextjs-react-the-complete-guide (cit. alle pp. 69–74, 79–81).

[17] OpenJS Foundation. *Node.js*. 2022. URL: https://nodejs.org/ (cit. a p. 69).

[18] the Sass team. *Sass: Syntactically Awesome Style Sheets*. 2024. URL: https://sass-lang.com/ (cit. a p. 69).

[19] Openize Pty Ltd. *Formato file SCSS - Foglio di stile a cascata Sass*. 20242. URL: https://docs.fileformat.com/it/web/scss/ (cit. a p. 70).

[20] Remix. *React Router*. 2024. URL: https://reactrouter.com/ (cit. a p. 73).

[21] React Bootstrap. *React Bootstrap*. 2024. URL: https://react-bootstrap.netlify.app/ (cit. a p. 74).

[22] Material-UI SAS. *MUI: The React component library you always wanted*. 2024. URL: https://mui.com/ (cit. a p. 74).

[23] Framer B.V. *Framer Motion*. 2024. URL: https://www.framer.com/motion/ (cit. a p. 74).

[24] Day.js. *Day.js · 2kB JavaScript date utility library*. 2024. URL: https://day.js.org/ (cit. a p. 74).

[25] Axios Project. *Axios*. 2024. URL: https://axios-http.com/ (cit. a p. 75).

[26] Alexey Lyakhov. *react-google-maps-api*. 2024. URL: https://github.com/JustFly1984/react-google-maps-api (cit. a p. 75).

[27] Mapbox Inc. *supercluster*. 2024. URL: https://github.com/mapbox/supercluster (cit. a p. 75).

[28] NEOSTACK INNOVATIVE TECHNOLOGIES (OPC) PRIVATE LIMITED. *React Slick*. 2024. URL: https://react-slick.neostack.com/ (cit. a p. 75).

[29] React Tooltip. *React Tooltip*. 2024. URL: https://react-tooltip.com/ (cit. a p. 76).

[30] Nikolay Kuchumov. *react-phone-number-input*. 2024. URL: https://gitlab.com/catamphetamine/react-phone-number-input#readme (cit. a p. 76).

[31] Manish Saraan. *email-validator*. 2024. URL: https://github.com/manishsaraan/email-validator (cit. a p. 76).

[32] Oliver Schneider. *otp-client*. 2024. URL: https://github.com/olsio/otp-client (cit. a p. 76).

[33] EmailJS Pte Ltd. *EmailJS*. 2024. URL: https://www.emailjs.com/ (cit. a p. 76).

[34] GitLab B.V. *GitLab: The most-comprehensive AI-powered DevSecOps Platform*. 2024. URL: https://about.gitlab.com/ (cit. a p. 78).

[35] JetBrains s.r.o. *IntelliJ IDEA – the Leading Java and Kotlin IDE - JetBrains*. 2024. URL: https://www.jetbrains.com/idea/ (cit. a p. 78).

[36] Luca Murante. *Guida a GitFlow*. URL: https://devdev.it/guida-gitflow/ (cit. a p. 78).

[37] SonarSource SA. *SonarLint*. 2024. URL: https://www.sonarsource.com/products/sonarlint/ (cit. a p. 79).

[38] Fulvio Corno, Luigi De Russis e Enrico Masala. *Context. The Foundations of React*. 2023. URL: https://github.com/polito-WA1-AW1-2023/materials/blob/master/slide/3-05-Context.pdf (cit. a p. 86).

[39] Luigi De Russis. *User Evaluation: Usability Testing*. 2023. URL: https://polito-hci-2023.github.io/materials/slides/12-usability-testing.pdf (cit. alle pp. 93, 98, 99, 109).

[40] Kate Moran. *Usability Testing 101*. 2019. URL: https://www.nngroup.com/articles/usability-testing-101/ (cit. a p. 93).

[41] Jakob Nielsen. *How Many Test Users in a Usability Study?* 2012. URL: https://www.nngroup.com/articles/how-many-test-users/ (cit. a p. 95).

[42] StatCounter. *Desktop vs Mobile vs Tablet Market Share Worldwide*. 2024. URL: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/ (cit. a p. 95).