

POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica



**Politecnico
di Torino**

Corso di Laurea Magistrale

Progettazione e sviluppo di software basato su microservizi

Relatori

Prof. Luca ARDITO

Prof. Maurizio MORISIO

Candidato

Veronica PUGGIONI

Dicembre 2021

Abstract

Contesto: Il progetto DME, Digital Management of Events, nasce con lo scopo di realizzare una piattaforma digitale per l'organizzazione di eventi di vario tipo (Matrimoni, Compleanni, Conferenze, Concerti, etc.). La piattaforma DME si propone di superare le limitazioni dei prodotti esistenti, fornendo un unico strumento in grado di coinvolgere tutti gli stakeholder del processo di organizzazione degli eventi, dai fornitori agli utenti finali. Con la collaborazione del Politecnico di Torino, Università degli studi di Bari e dell'Azienda Linear System, il progetto finanziato dalla regione Puglia, ha come obiettivo iniziale, l'ingresso nel mercato dell'organizzazione di eventi regionale, per poi espandersi sia nel territorio nazionale, che europeo. Nei primi mesi di progettazione sono stati sviluppati due macro moduli: Front End e Back End. La piattaforma si basa su un'architettura a microservizi e sull'utilizzo di altre tecnologie innovative come distributed ledger per smart contract.

Obiettivi: Il seguente lavoro di tesi ha come obiettivi l'analisi e la verifica mediante test sulla piattaforma progettata. In particolare, sono stati effettuati: test alpha delle funzionalità; valutazione del funzionamento basato sulla simulazioni di casi reali all'interno della piattaforma; progettazione ed esecuzione di test di usabilità con utenti esterni, il cui scopo è quello di migliorarle la piattaforma attraverso i feedback ricevuti. Altro obiettivo della tesi è stato la realizzazione del prototipo con interfacce UI accattivanti.

Metodo: I test sono stati svolti periodicamente per verificare la correttezza delle funzionalità, il che ha permesso di risolvere gli eventuali bug e completare le funzionalità mancanti all'interno della piattaforma. A seguito di questi è stato possibile eseguire i test su casi reali, di cui una parte basata su dati storici. Sono stati progettati i test di gradimento, poi sottoposti agli utenti, con l'obiettivo di rispondere alla domanda "ai clienti piace la piattaforma DME?". Gli utenti selezionati sono esterni al progetto, non a conoscenza della piattaforma e in grado dare informazioni utili per l'inserimento della stessa sul mercato di destinazione. Per quanto riguarda la realizzazione della UI, il modulo di Front End è stato sviluppato con il framework Angular, con l'uso di media query, bootstrap, e alcune componenti di Angular Material per rendere l'applicativo fruibile sia su desktop che su dispositivi mobile e tablet. Nella fase preliminare sono stati realizzati dei Mockup mediante l'uso di Adobe Illustrator e Adobe XD, con la collaborazione di un esperto nel settore della grafica dell'azienda Linear System, e successivamente questi sono stati inseriti all'interno del progetto tramite codice Html e Css.

Conclusioni: Il lavoro svolto presso l'azienda Linear ha portato alla conclusione di una buona parte dei requisiti richiesti. A partire dalla progettazione dei test sarà possibile effettuare ulteriori analisi sulla piattaforma al fine di renderla completamente pronta all'uso.

Ringraziamenti

Ringrazio professor L. Ardito, per tutta la disponibilità e l'aiuto fornitomi nella stesura di questa tesi.

Ringrazio l'Azienda Linear System per avermi dato la possibilità di svolgere il mio lavoro di tesi presso la loro sede di Roma, luogo interessante che mi ha permesso di mettermi in gioco e fare un'esperienza che sarà preziosa per il mio futuro.

Ringrazio tutte le persone conosciute durante questo percorso, in particolare Dario, Eleonora, Marika e Alessio per avermi fatta sentire a casa e avermi guidata nella stesura di questa tesi.

Indice

Elenco delle tabelle	VI
Elenco delle figure	VII
1 Introduzione	1
1.1 Progetto DME	1
1.2 Sviluppo Web Responsive	1
1.3 User Experience	1
1.4 Test usabilità	2
1.5 Test pagine web	2
1.6 Progettazione Front End	2
1.7 Risultati	2
2 Progetto DME	3
2.1 Scenario di riferimento	3
2.2 Tecnologie per il supporto di eventi	4
2.2.1 Benefici attesi	4
2.3 Stato dell'arte	5
2.4 Obiettivi	6
2.5 Back End	6
2.6 Front End	7
2.7 Fase di Validazione	8
2.7.1 Alpha Testing	8
2.7.2 Simulazione casi reali	8
3 Sviluppo Web Responsive	9
3.1 Introduzione	9
3.2 Storia	10
3.2.1 Design a dimensione fissa	10
3.2.2 Website mobile	10
3.2.3 Responsive Web Design e siti web differenti a confronto . . .	12

3.3	Creare un sito Responsive	13
3.3.1	HTML	13
3.3.2	Css	14
3.3.3	Media Query	18
3.3.4	Immagini	20
3.3.5	Bootstrap	22
3.3.6	Angular	22
3.3.7	React	24
3.4	Workflow Responsive Web Design	24
3.4.1	Design Lineare	25
3.4.2	Contenuto gerarchico	26
3.4.3	Pensare al Layout	26
3.4.4	Prototipi	27
3.4.5	Visual Design	28
3.4.6	Responsive Design Tools	29
3.4.7	Responsive nella pratica	30
4	User experience	31
4.1	Introduzione	31
4.2	UX, UI, e usabilità	32
4.3	Progettare una buona UX	32
4.3.1	Digital Branding	34
4.4	Psicologia della homepage	36
4.4.1	Homepage	36
4.4.2	Landing page	37
5	Test di usabilità	39
5.1	Introduzione	39
5.1.1	Piano di test	39
5.1.2	Report	40
5.2	Metodologie di User Centered Evaluation	42
5.2.1	Tecniche di valutazione	43
6	Test pagine web	48
6.1	Introduzione	48
6.2	Strumenti di Test	48
6.2.1	Problemi	49
6.3	Tecniche di test del software	49
6.3.1	Test di correttezza	50
6.3.2	White box testing	50
6.3.3	Black box testing	50

6.3.4	Grey box testing	50
6.4	Test Statici e Test Dinamici	50
6.4.1	Test statici	50
6.4.2	Test dinamici	51
6.5	Metodologie di test	51
6.6	Casi di test	52
6.6.1	Test automatizzati	52
6.6.2	Caratteristiche test automatizzati	53
6.7	Selenium	53
6.7.1	Componenti	53
6.7.2	Installazione e utilizzo di selenium	55
7	Progettazione Front End	56
7.1	Descrizione generale	56
7.1.1	Tipologie di utenti	56
7.2	Specifica dei Requisiti	56
7.2.1	Requisiti non funzionali	57
7.2.2	Requisiti funzionali	57
7.3	Strumenti e linguaggi utilizzati	57
7.4	Schema Sito DME	58
7.4.1	Organizzatore	59
7.4.2	Fornitore	64
7.4.3	UI	64
8	Risultati	65
8.1	Alpha Test	65
8.1.1	Risultati Alpha Test	65
8.2	Sperimentazione sul Campo con utenti finali	67
8.2.1	Preparazione test con gli utenti	68
8.2.2	Risultati Questionari	69
8.3	Simulazione Caso reale	71
8.3.1	Realizzazione di un matrimonio sulla piattaforma DME	72
8.3.2	Risultati test simulazione casi reali	74
9	Conclusioni	75
9.1	Sviluppi futuri	75
	Bibliografia	77

Elenco delle tabelle

8.1	Alpha Test	66
-----	----------------------	----

Elenco delle figure

3.1	Box model in Css	16
3.2	Schizzi preliminari possono essere realizzati direttamente su dispositivi che permettono il disegno.	26
3.3	Esempi di layout di siti web.	27
3.4	Le tre style tiles sono state utilizzate per arrivare al design finale del sito "examiner"	29
4.1	Biunivocità dei contenuti tra homepage e pagine secondarie	37
4.2	Generica struttura landing page	38
5.1	Fasi Human Centered Design	43
5.2	Esempio di questionario QUIS	46
5.3	Esempio di questionario QUIS	46
5.4	Esempio di questionario SUMI	47
6.1	Tecniche di test	49
6.2	Tecniche di test	51
6.3	Modello UML applicazione e-commerce	53
7.1	Architettura Organizzatore	59
7.2	Architettura Fornitore	60
7.3	Creazione evento	61
7.4	Gestione evento	61
7.5	Dettagli vetrina	62
7.6	Gestione ordini	62
7.7	Calendario e gestione appuntamenti	63
7.8	Gestione tavoli	63
7.9	Creazione blog	63
7.10	Creazione Servizi	64
8.1	Form Registrazione fornitore	67
8.2	Informazioni partecipanti	69

8.3	Utilizzo devices	70
8.4	Difficoltà e semplicità utilizzo	70
8.5	Considerazioni sulla piattaforma	71
8.6	Creazione Nuovo Evento	73

Capitolo 1

Introduzione

1.1 Progetto DME

Con la collaborazione del Politecnico di Torino, Università di Bari, DS Graphic Engineering S.r.l, STRADE S.r.l, Linear System nasce il progetto DME, Digital Management Events, finanziato dalla regione Puglia. Lo scopo è la realizzazione di una piattaforma digitale atta all'organizzazione di eventi, quali matrimoni, compleanni, conferenze, convegni e altro. L'obiettivo sarà quello espandere il progetto dalla regione Puglia, al territorio nazionale.

1.2 Sviluppo Web Responsive

Nel capitolo verrà introdotto lo sviluppo web responsive, verranno illustrate le principali tecnologie utilizzate per lo sviluppo di quest'ultimo. Di recente sono stati realizzati nuovi linguaggi e framework in grado di rispondere alla richiesta di interattività da parte degli utenti, oltre che alla diffusione maggiore di dispositivi mobili con connessione web. I principali team sono Microsoft e google che hanno contribuito alla creazione di strumenti in grado di aiutare gli sviluppatori nella realizzazione di applicativi sempre più avanzati.

1.3 User Experience

L'obiettivo del capitolo è quello di fornire principi e i concetti relativi alla user experience, fondamentali per la realizzazione di un sito web. Verrà approfondita la differenza tra user experience e usabilità, termini spesso confusi.

1.4 Test usabilità

Il capitolo tratterà una metodologia di ricerca della User Experience, ovvero, i test di usabilità. Lo scopo del capitolo è spiegare come avviene genericamente una sessione di test, quali sono gli scopi e come realizzare i documenti essenziali per una buona riuscita dei test.

1.5 Test pagine web

Le pagine web sono spesso soggette ad arresti anomali generalmente dovuti alla generazione dinamica delle pagine. Questo tipo di errori incide seriamente sull'usabilità delle applicazioni. Esistono varie tecniche e software che permettono l'esecuzione dei test in maniera automatizzata. Nel capitolo verranno discusse tali tecniche.

1.6 Progettazione Front End

Il capitolo tratterà lo sviluppo del front end riguardante la piattaforma DME, e la realizzazione di una UI accattivante che ne stimoli l'utilizzo da parte degli utenti per la realizzazione di eventi.

1.7 Risultati

Nel capitolo verranno mostrati i risultati ottenuti dai casi di test, relativi al progetto dme. Quali test alpha per la rilevazione di bug e incompletezze nel sito, simulazione di casi reali volti a trarre vantaggi e svantaggi nell'organizzazione di eventi online e, test di usabilità con gli utenti, che permetteranno di vedere il sito da un punto di vista differente con il fine di capire se la piattaforma piace.

Capitolo 2

Progetto DME

Il progetto DME, Digital Management of Events, nasce con lo scopo di realizzare una piattaforma digitale per l'organizzazione di eventi di vario tipo (Matrimoni, Compleanni, Conferenze, Concerti, etc..). La piattaforma DME si propone di superare le limitazioni dei prodotti esistenti, fornendo un unico strumento in grado di coinvolgere tutti gli stakeholder del processo di organizzazione degli eventi, dai fornitori agli utenti finali. Con la collaborazione del Politecnico di Torino, Università degli studi di Bari e dell'Azienda Linear System, il progetto finanziato dalla regione Puglia, ha come obiettivo iniziale, l'ingresso nel mercato dell'organizzazione di eventi regionale, per poi espandersi sia nel territorio nazionale, che europeo.

2.1 Scenario di riferimento

Il progetto DME riguarda l'organizzazione di eventi, che negli ultimi dieci anni ha avuto un forte sviluppo, dando un input importante anche nell'ambito del marketing e della interazione con le aziende. Questo ha permesso un aumento del numero di professionisti richiesti nel settore, e ha incentivando da una parte la loro collaborazione e dall'altra un miglioramento della produttività. Al momento attuale non è possibile stimare il valore di mercato riferito agli eventi in Italia, ma i dati nazionali mostrano un aumento negli investimenti aziendali.

Oggi le nuove strategie di marketing, che derivano specialmente dall'ambito digitale, portano al concetto di evento, ad adattarsi a nuovi contesti e interlocutori, infatti, una definizione più recente è "live communication", ovvero comunicazione viva, che coinvolga le persone. Il termine "evento" risulta incompleto, perché indica un qualcosa che dovrà succedere, che si è verificato o che può verificarsi, con live communication, invece, si ha un maggior coinvolgimento dei consumatori nel meccanismo di marketing, il che è in grado di dare risultati migliori al business stesso. Gli eventi diventano, parte integrante nella strategia comunicativa delle

aziende, permettendo una maggiore collaborazione tra brand e fruitori del servizio. Per Brand si intende una squadra molto efficiente, con condivisione di premesse strategiche e un livello di esigenza condiviso, con sfide sempre nuove e ambiziose rispetto alle precedenti, per creare un'esperienza memorabile per i propri consumatori. La progettazione degli eventi ha dei requisiti strategici, che come altri rami della comunicazione, iniziano dal brand, dal suo ruolo e dai suoi valori. Il brand, ha il compito di capire quali sono i target del cliente, e deve cercare di realizzare un'esperienza che sia nuova e soddisfacente per esso [1]. Il nuovo termine "live communication", designa una maggior professionalità da parte dei fruitori del servizio, che rispetto al passato, necessitano di dotare l'evento di strumenti sempre più interconnessi tra loro, per questo motivo l'uso della tecnologia è fondamentale, infatti, la portata dell'evento, viene amplificata tramite l'uso dei social network. L'evento diventa quindi sempre più strategico, ed un elemento integrante della comunicazione aziendale, perde la sua essenza del qui e ora, perché tramite i social il tempo di vita dell'evento è molto più lungo, poiché inizia prima e continua a vivere nei contenuti digitali [2].

2.2 Tecnologie per il supporto di eventi

La stima dei dati del settore prevede un incremento dell'organizzazione di eventi, con conseguente investimento da parte dei fornitori in soluzioni più efficaci, che portano all'utilizzo di software per la gestione organizzativa. Le aziende per poter sostenere il ritmo del mercato hanno la necessità di trovare novità per i propri business. Secondo un'analisi effettuata da Frost & Sullivan, circa l'80% delle aziende che si occupano organizza eventi mediante l'utilizzo di software di gestione. Questa percentuale si prospetta in aumento nei prossimi anni, ed inoltre si prevede una maggior diffusione della intelligenza artificiale piuttosto che non ai Big data.

Il vantaggio maggiore è dato dal fatto che l'utilizzo di interfacce utente, semplici e moderne, sia agli organizzatori che ai partecipante, rende l'organizzazione più semplice. L'uso di intelligenza artificiale permette di soddisfare i clienti garantendo loro un servizio mirato, dato da un'organizzazione guidata che fornisce le indicazioni sulle azioni e le tempistiche per il raggiungimento degli obiettivi preposti.

2.2.1 Benefici attesi

Sotto il punto di vista dell'azienda Linear System possiamo distinguere quattro principali vantaggi:

1. Varietà delle proprie offerte: il progetto DME permetterà a Linear System di addentrarsi anche nel mercato dell'organizzazione di eventi, che come abbiamo visto precedentemente, è un mercato in continua crescita. L'idea sarà quella

di partire dalla regione Puglia per poi espandersi sia campo nazionale che europeo.

2. Apprendimento nuove tecnologie:

- Distributed ledger;
- Cloud;
- Architetture microservizi ;
- Produzione in modalità Agile;
- Deployment con DevOps;
- Machine Learning

che permetteranno all'azienda una maggiore competizione nel mercato ITC. Le tecnologie elencate verranno acquisite da Linear System, Politecnico di Torino e Università di Bari con la modalità *learning on the job*.

3. Aumento occupazionale nella filiera dell'organizzazione di eventi: La piattaforma DME richiederò per la progettazione, implementazione e test sulla piattaforma, un team dedicato, il che implica un aumento sotto il profilo occupazionale.
4. Riduzione del Consumo: Il funzionamento in cloud permette di ridurre il consumo sia per l'azienda sopra citata, sia per i suoi clienti, che possono evitare l'acquisto di attrezzature informatiche e relativi consumi energetici.

2.3 Stato dell'arte

Il progetto DME prevede l'utilizzo di tecnologie e metodologie di punta: Cloud e virtualizzazione. La piattaforma DME è progettata per operare su macchine virtuali all'interno di infrastruttura cloud. Il cloud computing consente la creazione di pool di risorse da assegnare a progetti ed è possibile fornire maggiore capacità di elaborazione assegnando ulteriori risorse, grazie ai carichi di lavoro virtualizzati.

- **Architetture a microservizi:** Le architetture a microservizi prevedono l'organizzazione dell'applicazione in un insieme di servizi elementari indipendenti. Questo comporta vantaggi in termini di scalabilità e flessibilità di sviluppo rispetto al tradizionale approccio monolitico. D'altro canto, le architetture a microservizi presentano una serie di problematiche, tra cui la complessità di orchestrazione, la consistenza dei dati, l'overhead e la complessità dei test di integrazione. Quando si adottano architetture a microservizi è necessario adottare soluzioni tecnologiche per ovviare a questi problemi.

- **Recommender System:** Grazie all'utilizzo di tecniche di machine learning la piattaforma DME analizza i dati relativi all'organizzazione di eventi passati al fine di generare proposte di fornitori e servizi agli utilizzatori futuri. Produzione di servizi multicanale. La piattaforma DME è pensata per essere fruibile da ogni tipo di dispositivo (pc, tablet, telefono) in modo da consentire la più grande facilità d'uso possibile per gli utenti finali.
- **Sviluppo Agile e DevOps:** La metodologia di sviluppo Agile prevede frequenti rilasci del software, attraverso tecniche di Integrazione Continua e Distribuzione Continua, che prevedono testing e deployment automatico del codice. Ciò va di pari passo con l'approccio DevOps, che prevede una stretta collaborazione tra sviluppatori e operativi, per rendere più veloce l'intero ciclo di vita del prodotto.

2.4 Obiettivi

Il progetto DME prevede la realizzazione di otto Obiettivi Realizzativi (OR), suddivisi tra Ricerca Industriale (3), Sviluppo Sperimentale (4) e Project Management (1):

- OR1 - Stato dell'arte delle Metodologie e delle Tecnologie.
- OR2 - Definizione delle Tecnologie e degli Standard
- OR3 - Progettazione della Piattaforma Cloud e dei servizi Software
- OR4 - Sviluppo dell'Infrastruttura cloud
- OR5 - Sviluppo della Piattaforma DME
- OR6 - Integrazione delle componenti e test di laboratorio
- OR7 - Sperimentazione sul campo con utenti finali
- OR8 - Project Management

Ciascun obiettivo è composto da una serie di attività ed ogni attività ha lo scopo di produrre un deliverable, ad esempio un documento o un prototipo.

2.5 Back End

Il layer di back end dell'applicazione prevede l'impiego di un'architettura a microservizi. È previsto il deployment su infrastruttura cloud privata all'interno del CED (Centro Elaborazione Dati). Il modulo principale del back end implementa la

maggior parte dei processi di business dell'applicazione e mantiene le informazioni relative agli utenti, ai fornitori, ecc. Inoltre, tale modulo implementa soluzioni per l'autenticazione e l'autorizzazione degli utenti. Lo sviluppo di questa parte è a carico dell'azienda Linear System. La comunicazione tra varie componenti del layer di back end può avvenire, a seconda delle necessità e del tipo di servizi coinvolti, in modalità asincrona o sincrona. La comunicazione tra il layer di front end e quello di back end è sincrona, mediante il protocollo HTTP/HTTPS. Questo modulo principale offre delle API interrogabili dal layer di front end e da altri servizi che compongono il layer di back end. Alcune di queste sono private, quindi solo ad uso interno alla rete aziendale, altre saranno pubbliche, per l'uso da parte di utenti e tester. Al modulo principale si vanno ad aggiungere altri due moduli, ciascuno realizzato attraverso un microservizio:

- Modulo Recommender System
- Modulo Distributed Ledger / Smart Contract

Il modulo Recommender System ha l'obiettivo di aiutare gli utenti nelle scelte, fornendo loro suggerimenti personalizzati. Questi possono essere basati su cronologia acquisti, valutazioni espresse, preferenze di altri utenti, ecc. Esistono vari approcci per la realizzazione di Recommender System, quali content-based, collaborative filtering, ibridi, ecc. Lo scopo è quello di avvicinare il più possibile la domanda all'offerta. Il modulo Distributed Ledger / Smart Contract ha lo scopo di registrare in un ledger distribuito le transazioni tra clienti e fornitori, e di gestire contratti digitali dematerializzati. Un distributed ledger è un registro di transazioni immutabile, distribuito su più nodi. Ciascun nodo mantiene una copia del registro e gli aggiornamenti avvengono in modo indipendente, ma convalidati secondo un algoritmo di consenso. Una tecnologia comune per la realizzazione di distributed ledger è la Blockchain, che raggruppa le transazioni in blocchi, ciascuno con un hash che lo concatena al blocco precedente. Per Smart Contract si intende un contratto digitale in grado di determinare automaticamente l'esecuzione delle clausole al verificarsi delle condizioni previste dal contratto.

2.6 Front End

La progettazione pone al centro dell'intero sistema l'utente, il confronto ne ha permesso uno sviluppo basato sulla User Experience Design. La piattaforma sarà destinata a qualsiasi tipo di utente, per questo motivo il punto cardine di questo macro-modulo sarà la semplicità. La UI presenta interfacce semplici e chiare che permettono di raggiungere le sezioni in modo rapido. Altro aspetto fondamentale della progettazione è dato dall'accesso multi-device, il che permette l'utilizzo della piattaforma in qualsiasi device (desktop, tablet, smartphone). L'architettura

basata su microservizi ha richiesto che la struttura comprendesse la realizzazione dei comportamenti per ciascun microservizio. Il layer di front End permette tre tipi di registrazione: Organizzatore, Fornitore e Invitato, e una modalità di Login che riconosce in automatico la tipologia di utente, reindirizzandolo alle sezioni ad esso dedicate, come ad esempio la sezione di creazione servizio per un fornitore, e creazione di evento per un organizzatore.

2.7 Fase di Validazione

La piattaforma DME sarà il risultato finale del progetto, su di essa sarà possibile eseguire delle verifiche sia tramite controllo documentale, sia mediante i prototipi. Al termine verranno resi disponibili i risultati di ciascuna attività.

2.7.1 Alpha Testing

Lo scopo degli Alpha testing è quello di rilevare Bug, anomalie e funzionamenti mancanti, in una prima fase, la seconda fase prevede un controllo della qualità ed ulteriori verifiche. Si prevede che questi vengano effettuati quando la piattaforma non è ancora completamente pronta all'uso, ma in uno stato tale che ne permetta una valutazione. I Test prevederanno diverse fasi:

- Revisione specifiche e requisiti funzionali di progettazione;
- Sviluppo piano di test
- Esecuzione del piano di test definito
- Registrazione dei difetti

Il procedimento sarà iterativo, fino a che tutti i problemi non saranno risolti.

2.7.2 Simulazione casi reali

A seguito degli Alpha Testing, verranno svolti i test con gli utenti. I test su casi reali permetteranno di valutare la soddisfazione potenziale dell'utente, attraverso una visita guidata nella piattaforma. Il fine sarà quello di capire se agli utenti piace la piattaforma DME. Gli utenti selezionati non conoscono la piattaforma, e sono esterni al progetto, questo permette una maggiore oggettività da parte della loro esperienza sulla piattaforma.

Capitolo 3

Sviluppo Web Responsive

3.1 Introduzione

L'aggettivo inglese "responsive" designa tutto ciò che reagisce o risponde dinamicamente e in modo appropriato ad uno stimolo; In italiano viene tradotto come *responsivo*, adattivo o adattativo, ovvero un qualcosa capace di adattamento, capace di modificarsi per soddisfare i nuovi requisiti, dal vocabolario treccani [3]:

Responsivo agg. In informatica, capace di adattarsi alla navigazione su computer o su dispositivi mobili, con particolare riferimento alla dimensione e alla forma dello schermo.

La definizione applicata al mondo web consiste nell'adattare, mediante regole, un sito web in base a vari fattori, come la dimensione dello schermo, piattaforma o orientamento del dispositivo. Responsive Web Design è una delle principali tendenze nello sviluppo web, esso è fondamentale per la creazione di applicazioni web fruibili su diversi dispositivi utilizzati per la navigazione. Il primo a scrivere sul Responsive Web Design fu Ethan Marcotte, nel 2010, in "A list Apart" [4]:

I designers hanno bramato la stampa per via dei suoi layout precisi, lamentandosi dei vari contesti d'uso dei propri utenti sul web che compromettono i loro design. Ethan Marcotte sostiene che si debba cambiare il modo in cui pensiamo i nostri design per superare questi limiti: usando le flex grid, le immagini flessibili e le media queries, ci mostra come adottare il flusso ed il riflusso delle cose con il web design reattivo.

Negli ultimi dieci anni è diventato lo standard *de facto* utilizzato per adattare le pagine web ad un ampio numero di dispositivi, ma sebbene abbia migliorato l'usabilità delle pagine web non è privo di limitazioni e inconvenienti, infatti, gli sviluppatori devono progettare manualmente i layout web per più dimensioni e

implementare le regole di adattamento [5].

Il RWD si compone di moltissime regole specifiche, una di queste consiste nel considerare le misure in senso relativo, e non misure fisse in pixel, che, invece, non assicurano una flessibilità del layout, in quanto l'elemento html conserverà sempre la stessa dimensione passando da una visualizzazione su grandi display a dispositivi mobile, o tablet [6]. Un sito web responsive è un approccio al web design che fa sì che i contenuti si adattino ai dispositivi mobili come tablet e smartphone e ai dispositivi più tradizionali come Computer e Desktop, in questo modo viene offerta un'esperienza user-friendly agli utenti, che possono utilizzare il sito web indipendentemente da "dove" e "quando" il cliente utilizza il sito [7]. Altro aspetto rilevante per il Responsive Web Design è l'accessibilità sia dal punto di vista degli utenti, che dalle caratteristiche del dispositivo utilizzato.

3.2 Storia

3.2.1 Design a dimensione fissa

Negli anni 2000 i siti venivano realizzati con una dimensione di circa 960px, dimensione tipica di desktop e laptop in quegli anni, ma che permetteva di adattarsi anche a schermi di dimensione più ampia (1024 px) lasciando un maggiore margine sui bordi. Sempre negli stessi anni nacquero le idee del *fluid design* e del *liquid layout* [8].

3.2.2 Website mobile

Nel 2007, con il lancio da parte della Apple del primo iPhone, viene inaugurata per la prima volta la navigazione da telefono. Prima di questo i telefoni cellulari non avevano accesso ai siti internet nello stesso modo in cui era possibile accedervi dai computer. Spesso avevano i primi browser per dispositivi mobili potevano visualizzare solo pagine bianco e nero, utilizzando la versione base di HTML. Questo avveniva, ad esempio, con le previsioni del tempo. L'iPhone è stato il primo dispositivo mobile capace di visualizzare una pagina internet come quelle visibili nei computer. L'iPhone fu anche il primo dispositivo ad avere il multi-touch che permetteva di zoomare e ridurre le dimensioni della pagina, progettata per stare su uno schermo di dimensioni maggiori (960px rispetto a 320px dell'iPhone). Questa non era una soluzione ottimale, era quindi necessario trovare altre soluzioni per la visualizzazione corretta dei siti web su dispositivi mobili. Una prima soluzione fu quella di creare dei siti web separati per desktop e mobile. Gli utenti da smartphone venivano reindirizzati alla pagina web mobile. Un approccio di questo tipo presentava diversi problemi, in primo luogo richiedeva un lavoro extra per i web designer, e in secondo luogo, i siti venivano realizzati e ottimizzati per dispositivi

con misure specifiche (iPhone), il che non permette di adattarsi bene a tutti i dispositivi. Per ridurre il carico su i web design, i siti mobile presentavano meno contenuti rispetto a quella desktop, con la giustificazione che venivano usati in movimento, o per alcune attività di base, e quindi non necessitavano di tutte le funzionalità. Con il passare del tempo però i cellulari hanno avuto una crescita esponenziale, e diventava così necessario l'utilizzo di tutte le funzionalità, come su computer e laptop. Altro momento cruciale fu nel 2010 con l'uscita da parte della Apple del primo iPad. I siti mobile risultavano troppo piccoli, e i siti desktop troppo grandi, alcuni designer decisero quindi di creare una nuova versione specifica per questo dispositivo. Nel momento in cui nel mercato iniziarono ad arrivare dispositivi di diverse dimensioni, non era più possibile realizzare tanti siti diversi, per le diverse dimensioni dei dispositivi. Alcuni designer reagirono creando una versione web specifica per iPad, ma questo non diventava più sostenibile nel momento in cui nel mercato iniziavano ad arrivare tanti dispositivi di diverse dimensioni, non era più possibile realizzare dei siti per ogni dimensione dei dispositivi.[8] Il Responsive nasce con la necessità di rendere i siti facilmente accessibili con ogni dispositivo e risoluzione, in questo modo si ha un'unica versione che si adatta su schermi di qualsiasi dimensione, esso consolida tre tecniche esistenti in un unico approccio.

- Flex grid layout: griglia flessibile;
- Flex image: immagini flessibili;
- Media query

Il numero di persone che utilizzano dispositivi con schermi più piccoli è sempre maggiore, e ora non vi è più differenza tra schermi grandi e piccoli. Inizialmente la filosofia del Responsive web design consiste nell'implementare prima i design desktop a larghezza fissa e poi ridimensionare il design secondo le necessità per gli schermi più piccoli. Tuttavia, il modo migliore consiste nello sviluppare prima i design per gli schermi più piccoli e poi migliorarlo progressivamente per schermi più grandi e dispositivi più capaci. Prima di procedere alla realizzazione di siti è necessario decidere su quali piattaforme saranno supportati al 100%, e su quali altri invece avranno qualche anomalia funzionale/visiva. Il sito <http://caniuse.com>, permette quali funzionalità sono supportate dai vari browser/dispositivi.[9]

Flex Grid Layout

L'approccio fluido/proporzionale, consiste nell'utilizzare le percentuali al posto dei pixel, questo fa sì che il contenuto cresca e si rimpicciolisca per riempire la finestra del browser, indipendentemente da quanto questo sia grande o piccolo. L'uso delle percentuali funziona bene tra dispositivi con dimensioni simili, ma tra dispositivi con schermi significativamente diversi, si creano disallineamenti evidenti di visualizzazione.

Media Query

Le Media Query sono una regola dei Css, grazie alla quale il sito viene renderizzato con layout diversi a seconda delle caratteristiche dello schermo del dispositivo utilizzato, senza modificarne il codice HTML. Questo metodo permette di avere un unico sito, visualizzabile correttamente su diversi dispositivi. Per poter funzionare correttamente le Media Query necessitano misure flessibili, per questo motivo le colonne sono misurate in percentuale e il testo in *em*, unità di misura relativa.

3.2.3 Responsive Web Design e siti web differenti a confronto

Il Responsive Web Design permette di avere un'unica pagina web che viene visualizzata sui vari dispositivi. Avere più siti web per diversi dispositivi, poteva presentare problemi di visualizzazione, nel caso si utilizzassero dei rilevamenti del dispositivi, poteva essere inviata infatti una versione non corrispondente al dispositivo, o poteva volerci un tempo di caricamento maggiore per la pagina, o se si utilizzava un URL separato poteva determinare un rendering non corretto dei contenuti. Un ulteriore problema, con i siti con differente URL poteva essere determinato dalla condivisione dei link da parte degli utenti, ad esempio se un utente su desktop manda il link ad un utente che poi dovrà visualizzare il contenuto su dispositivo mobile, dovrà prima passare alla versione mobile per poterne visualizzare correttamente il contenuto, o viceversa. Con il RWD si ha un unico Url, il vantaggio maggiore è dato dal fatto che si dovrà creare un unico progetto per la pagina web, e dunque un unico codice HTML. Avere siti separati, significava effettuare le modifiche su più codici HTML, che se pur allineati nel set dei contenuti potevano creare dei problemi nella visualizzazione dei contenuti, con il responsive web design, le modifiche di progettazione mirate per i dispositivi, avverranno semplicemente nei fogli di stile.

Problemi nella ricerca

Un sito con Url separate possono creare problemi con il posizionamento del sito nei motori di ricerca, quest'ultimi infatti hanno bisogno di sapere che sono la stessa pagina in modo da poter indicizzare correttamente, e visualizzarli come un'unica voce nella lista dei risultati di ricerca. Se non si riesce a farlo correttamente si possono avere più visualizzazioni della stessa pagina nei risultati di ricerca confondendo gli utenti, ed influenzando negativamente il ranking di ricerca. È stato Google nel 2012 a raccomandarne l'utilizzo, per tre motivi fondamentali:

- Migliorare la user experience;
- Rendere più efficiente la ricerca sul web;
- Aggiornamento del sito in modo più rapido.

3.3 Creare un sito Responsive

Per Realizzare un sito Responsive sono necessari Html e Css. Al crescere della complessità aumenta però la necessità di coinvolgere linguaggi come javascript, le cui ultime tecnologie sono Angular, React etc, queste permettono un front-end più preciso sia in termini di prestazioni che di facilità d'uso. [10]

3.3.1 HTML

HTML è un linguaggio di markup, il che significa che permette di specificare la posizione degli elementi nella pagina. HTML è un linguaggio in continua evoluzione che si adatta in base alle esigenze del tempo e della tecnologia. Nell'ottobre 2014 è stata pubblicata come W3C Recommendation la versione HTML5 [11]. Il World Wide Web Consortium (W3C) è uno standard volontario che decide cosa deve essere presenta nella versione ufficiale di HTML. Volontario significa che nonostante si accettino le regole non è necessario seguirle. Prima del W3C i browser creavano le loro regole, il che rendeva le pagine molto differenti a seconda del browser utilizzato. Seguire le regole assicura una maggiore probabilità che il sito sia compatibile con dispositivi e tecnologie futuri. La versione nativa per il web e per la creazione di applicazioni (mobile e desktop) è HTML5, che si compone della definizione di:

- Sintassi per il markup: con controlli form, adatti alle esigenze moderne;
- Api che consentono di gestire network, multimedia, e hardware dei dispositivi;

Il linguaggio è composto da elementi, che sono dei componenti che servono a descrivere gli elementi all'interno della pagina. La pagina HTML è un file di testo con estensione `text/html`, in cui vengono salvate tutte le specifiche relative alla pagina web. Per poter realizzare un documento HTML occorre, per prima cosa, dichiarare in cima al file la dichiarazione del doctype che dice al browser la versione di HTML utilizzata, all'interno poi si trovano i **tag**, delle etichette, che hanno la caratteristica di stare all'interno di parentesi angolate. Attraverso i tag; quindi, si indica quali elementi dovranno apparire e in che modo saranno disposti. [12].

```
<!doctype html>
<html lang="it">
<head><title>Ciao Mondo!</title></head>
<body>
<h1>Ciao Mondo!</h1>
<p>Questa è la nostra prima pagina HTML!</p>
</body>
</html>
```

3.3.2 Css

Cascading Style Sheets, anche detti *Css*, sono il complemento del linguaggio *HTML*, essi forniscono le istruzioni sullo stile della pagina. I *Css* modellano l'intero layout di un sito web, dalla formattazione del testo al posizionamento degli elementi, alla disposizione che gli elementi avranno rispetto ad altri elementi. Chiamati fogli di stile in cascata perché la cascata determina l'ordine con cui gli stili vengono applicati. Una regola determina lo stile applicato ad un elemento html, ogni regola consta di due elementi, il selettore e la dichiarazione. Il selettore è l'elemento *HTML*, nella dichiarazione viene definita la qualità della proprietà da modificare, come ad esempio colore, dimensione, etc. Ogni proprietà è seguita ":" e il valore, nel momento in cui è necessario dichiarare più di una proprietà questa viene seguita dal ";".

```
p {  
  color: red;  
  font-size: 1.5em;  
}
```

Gli stili possono essere applicati anche ad un sottoinsieme di elementi, utilizzando *id* o classi.

```
.nomeClasse{  
  color: blue;  
}  
#id{  
  color: green;  
}
```

questo codice influenzerà tutti gli elementi html con *classe="nomeClasse"* o con *id="id"*, classe e id funzionano in modo simile, ma è possibile applicare le classi a tutti gli elementi mentre gli id solo a un elemento. Come *Html* anche dei *Css* esistono diverse versioni, la più recente è *Css3*, ogni versione di *Css* include nuovi componenti stilizzabili, ma la più grande novità di *Css3* sono l'introduzione delle *media query*, che permettono alla pagina di adattarsi alle diverse dimensioni dei dispositivi. Inoltre, come accade anche per l'*html* non tutte le pagine supportano i *Css* nello stesso modo; quindi potrebbero esserci dei rendering differenti a seconda del browser utilizzato. I *Css* possono essere applicati alla pagina web in vari modi: possono trovarsi in un file separato, o essere inclusi nel file della pagina web, nel secondo caso verranno applicati solo alla pagina in questione, mentre nel primo caso saranno applicati a tutti i file. È comunque buona norma scrivere i *Css* in un file separato, anche nel caso in cui questi siano relativi ad una sola pagina, si avrà infatti un file *Css* relativo a quella singola pagina, e il file *Css* relativo all'intero progetto. Per avere una dichiarazione ancora più specifica, è possibile utilizzare

i Css "inline", che si trovano nell'elemento hmtl, è però una tecnica sconsigliata, poiché si perde facilmente traccia della dichiarazione del Css, può però essere utilizzata nei casi di test.

Css Cascade

I Css seguono delle regole molto dettagliate per applicare gli stili ad un elemento, in particolare, se un elemento ha degli stili che vanno in conflitto i cascade determinano l'ordine con cui verranno eseguiti i fogli di stile. I browser seguono una gerarchia molto specifica per decidere quale proprietà applicare all'elemento:

1. Regole contrassegnate come importanti: proprietà seguita da !important;
2. Css inline;
3. Regole contenenti id;
4. Regole contenenti classi, attributi e pseudo-classi;
5. Regole contenenti elementi e pseudo-elementi;
6. Regole ereditate;
7. Valori predefiniti;

L'ordine è chiamato *Css specificity* e va dalle regole più specifiche a quelle meno specifiche; quindi andrà a verificare che siano presenti prima le regole ritenute importanti rispetto a quelle contenenti id. Per ogni elemento della pagina il browser passa attraverso una gerarchia, in ordine finché non trova uno stile per ogni proprietà. Nel momento in cui, due regole definiscono la stessa proprietà ma con valori diversi, verrà eseguita la proprietà scritta per ultima all'interno del foglio, ovvero con priorità maggiore.

```
h1{
  color: blue;
}
h2{
  color: red;
}
```

In questo caso la regola che verrà applicata sarà la seconda perché è l'ultima ad apparire nel codice.

Box Model

Un concetto importante per i Css è il Box modeling, che descrive lo spazio preso dall'elemento HTML. Il box-model è costituito da quattro box:

- content;
- padding;

- border;
- margin.

Ciascuno di questi può essere modificabile mediante Css. Nell'immagine in figura 3.1

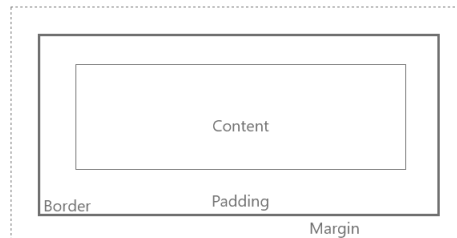


Figura 3.1: Box model in Css

partendo dall'interno abbiamo la zona in cui si trova il **contenuto** vero e proprio, che può essere testo, immagine etc. Le proprietà **width** e **height** modificano rispettivamente le dimensioni orizzontali e verticali dell'immagine. Per creare un margine all'interno del contenuto può essere utilizzata la proprietà **padding**, spazio che si trova tra il bordo dell'elemento e il contenuto. Nel momento in cui si imposta un colore di sfondo per il contenuto, questo verrà esteso fino al padding. Il **margin** è lo spazio che separa il dato dagli elementi adiacenti, anche questo modificabile nelle dimensioni. Il contenuto e il padding sono circondati dal **bordo**, una linea il cui colore e stile può essere modificato sempre mediante opportune regole di Css. [13]

Il codice HTML di default, anche senza Css imposterà delle impostazioni predefinite, introducendo , ad esempio, un certo margine tra i due elementi. Le unità di misura più comuni sono:

- %: La percentuale utilizzata in confronto all'elemento contenitore;
- em: sono relativi alla dimensione del carattere dell'elemento (**font-size**);
- rem: si riferiscono alla dimensione del font (**font-size**), questa è in assoluto l'unità di misura più comune.
- px: i pixel sono misure assolute, ma non sono necessariamente coerenti tra diversi dispositivi;
- in, cm, mm: unità di misure fisiche si mappano su i pixel, ma non sono coerenti tra dispositivi;
- pt: punti, sono uguali a 1/72 di inch, sono molto comuni nella stampa.

Nel box model ogni elemento ha altezza e larghezza che possono essere cambiati mediante i Css. Gli elementi di un blocco sono, per impostazione predefinita, al 100% della larghezza dell'elemento contenitore, anche se il contenuto non riempie lo spazio, mentre l'altezza è alta tanto da contenere l'elemento. Ogni elemento ha

margini che forniscono lo spazio esterno all'elemento, e il padding che ne fornisce lo spazio interno. Per il Responsive Web Design si utilizza in genere la percentuale per impostare i margini destro e sinistro, e padding, in questo modo il layout può ridimensionarsi in modo automatico agli schermi di diverse dimensioni. I margini verticali (alto e basso) possono essere invece impostati mediante pixel, poiché non è necessario che rispondano alle dimensioni dell'area visibile. I 4 lati dell'elemento possono avere valori diversi sia per il margine che per il padding, le dimensioni si definiscono in senso orario a partire dal **top**, **right**, **bottom**, **left**, possono essere anche definite singolarmente con ad esempio **margin-top**. Per centrare un elemento all'interno del contenitore si può utilizzare il valore auto per destra e sinistra.

I bordi sono posizionati tra il padding e il margine e forniscono un contorno per l'elemento, la dichiarazione deve includere la larghezza della linea, lo stile e il colore. Una nuova proprietà inserita nei Css3 è il border-box, tale proprietà può assumere due valori:

- **content-box**: il width fa riferimento all'area del contenuto;
- **border-box**: il width fa riferimento al box nella sua interezza; quindi, comprende padding e bordo.

Display

Proprietà chiave per gli elementi html è la display. Questa proprietà permette di definire come l'elemento deve occupare lo spazio all'interno della pagina. In generale gli elementi hanno un valore **block** o **inline**, la cui differenza consiste nella visualizzazione, rispettivamente, degli elementi come blocchi impilati o elementi in linea.

Position

La chiave per disporre gli elementi nel modo desiderato è l'utilizzo della proprietà **position**, essa può assumere i valori:

- **static**: tutti gli elementi per impostazione predefinita sono statici, rappresenta la posizione normale che ciascun elemento occupa nel flusso del documento.
- **relative**: Posizione relative dell'elemento rispetto al box contenitore. La posizione viene traslata con l'uso delle proprietà top, left, bottom o right, queste non indicano un punto preciso, ma lo spostamento in senso orizzontale e verticale rispetto al box contenitore [14].
- **absolute**: L'elemento viene rimosso dal flusso del documento e viene posizionato in base ai valori forniti con le proprietà top, left, bottom o right, il posizionamento assoluto avviene sempre rispetto al box contenitore, è rappresentato dal primo elemento antenato con posizionamento diverso da static, se

non presente viene preso come riferimento l'elemento radice. Se un elemento ha una posizione "absolute" sarà di default anche block-level. Ovvero è come se si dichiarasse anche `display: block;`

- **fixed:** Anche in questo caso viene la posizione viene sottratta dal normale flusso del documento, la differenza con absolute, è che il box contenitore è sempre la viewport.¹ Altra differenza fondamentale è data dal fatto che con fixed il box non scorre con il resto del documento ma rimane fisso nel punto.

Float and Clear

Float è la proprietà che permette di togliere l'elemento dal normale flusso del documento e posizionarlo ad esempio ad un lato del contenitore. Clear è invece la proprietà che impedisce che al fianco di un elemento compaiano altri elementi con il float.

3.3.3 Media Query

Le media Query sono ciò che rendono i siti reattivi ai dispositivi in cui vengono visualizzati, esse consentono di applicare stili differenti in base alla dimensioni del dispositivo su cui viene visualizzato. In genere si utilizza la larghezza della viewport. La sintassi delle Media query è molto semplice, si tratta di controllare se lo schermo su cui viene visualizzata la pagina è più grande o più piccola di una certa dimensione:

```
body { background-color: green; }
@media only screen and (min-width: 650px) {
  body { background-color: blue; }
}
```

In questo esempio lo sfondo blu verrà applicato nelle visualizzazioni di tutti i dispositivi la cui dimensione è inferiore ai 650px di larghezza, mentre per tutti gli altri dispositivi lo sfondo sarà verde. Per le regole di precedenza viste in precedenza, le media query devono trovarsi sul fondo del file. Oltre alla dimensione dello schermo nelle media query è possibile inserire altre funzionalità, ad esempio orientamento dello schermo, aspect ratio, risoluzione, etc. Anche se si tende a scrivere le media query rispetto alla dimensione dello schermo in realtà le dimensioni sono relative alla viewport, questo consente, modificando le dimensioni dello schermo, il comportamento reattivo della pagina.

¹Si intende la finestra principale del browser, ovvero l'area del contenuto.

Sistemi a Griglie

Il responsive web design è spesso associato all'uso di griglie. Una griglia comprende una serie di linee orizzontali e verticali utilizzate per organizzare il contenuto. Le griglie permettono di progettare in proporzioni, bilanciando tra tutti elementi presenti all'interno del progetto. I siti spesso utilizzano le griglie a 12 colonne, perché queste danno molta flessibilità, essendo 12 divisibile per 2,3,4,6.

Quando si lavora con le media query per adattare il contenuto alle diverse dimensioni degli schermi, è possibile partire prima dalle dimensioni più grandi per poi adattare a quelle più piccole, o viceversa. Una best practice in questi casi consiste nel progettare prima gli schermi con dimensioni inferiori, e poi quelli di dimensioni maggiori. Infatti, una volta creato il layout per uno schermo stretto è molto più semplice progettarlo per lo schermo di dimensioni maggiori, perché si avrà molto più spazio per inserire tutti gli elementi, il contrario risulta invece molto più complicato, e l'unica soluzione sarà "schiacciare" i contenuti all'interno della pagina.

Breakpoint

Un altro concetto importante del Responsive Web Design sono i breakpoint, questi stabiliscono le soglie alle quali modificare la presentazione grafica in funzione della larghezza del dispositivo. Bootstrap [15] individua quattro resolution breakpoint corrispondenti a quattro categorie di device, utilizzando il pixel come unità di misura per le diverse soglie.

- Extra small device con risoluzione inferiore a 576px;
- Small devices con risoluzione fino a 768px;
- Medium devices con risoluzione fino a 992px;
- Large devices con risoluzione superiore 1200px;

I breakpoints sono importanti, ma è importante che il sito abbia un bell'aspetto per qualsiasi dimensione, non solo per determinati punti. Quando si lavora con le media query i breakpoint sono arbitrari, possono essere di numero arbitrario, compreso lo zero e quindi non esserci. Il numero di breakpoint sarà dato dalla pianificazione iniziale del design, vedendo la pagina a partire dalle dimensioni più strette ed allargarle via via per vedere cosa accade. È buona norma non basare i breakpoint direttamente sulle dimensioni dei dispositivi, poiché come già detto in precedenza il design deve essere adatto per qualsiasi dimensione, anche se alcuni design scelgono i breakpoint rispetto alle dimensioni dei dispositivi più comuni. Alcuni designer scelgono i loro punti di rottura in base alle dimensioni dei comuni. L'obiettivo è quello di avere una pagina che si adatti a qualsiasi larghezza di viewport, possono essere pensati in maniera più generale come dimensioni desktop, tablet e smartphone, ma senza misure precise relative ad un dispositivo nello specifico. Dopo aver determinato i breakpoint ridimensionando la viewport

è possibile provare su dispositivi per vedere come appare, e nel momento in cui ci si ritrova un breakpoint con dimensioni simili ad un dispositivo noto, questo può essere modificato per renderlo un break point. Per esempio, se la larghezza dell'iPad è a pochi pixel dal breakpoint scelto, è possibile adattare questo in modo che la dimensione dell'iPad ricada nel breakpoint.

3.3.4 Immagini

L'inserimento delle immagini in un design responsive può risultare complesso, non saranno mai visualizzate con le stesse dimensioni perché flessibili. Il design responsive sulle immagini è in continua evoluzione, nella ricerca della soluzione migliore. L'aggiunta delle immagini può avvenire in diversi modi a seconda dello scopo.

Css

I Css permettono di aggiungere bordi, ombre, gradienti e pulsanti etc. Fino a poco tempo fa per ottenere lo stesso risultato si usavano le immagini, questo permette quindi di scaricare meno immagini, rendendo la pagina più leggera.

Content Image

Sono caratterizzate dall'elemento HTML `<img>`, è uno dei pochi elementi con un solo tag:

```

```

esso comprende anche due attributi:

- `scr`: che dice al browser dove prendere il file immagine;
- `alt`: fornisce un'alternativa di testo nel caso in cui il browser non sia in grado di visualizzare l'immagine. È molto importante che il testo dica all'utente cosa è presente nell'immagine.

Background Image

Proprietà Css che permette di aggiungere immagini come sfondo a scopi decorativi, senza che quindi facciano parte del contenuto.

```
div { background-image: url(flowers.png);
```

Questo tipo di immagini non saranno visibili alle persone che usano lettori di schermi basati sul testo.

Sprite

Ogni richiesta HTTP che una pagina web fa per recuperare i file necessari (file, immagini, Css) aggiunge tempo nel caricare la pagina. Per ridurre il numero di richieste le immagini possono essere caricate unendo tante immagini in una immagine unica, utilizzando poi il posizionamento Css per visualizzarle correttamente. Queste immagini sono chiamate **sprite**.

Icone

Le icone sono come qualsiasi altro font, ma i caratteri alfanumerici sono sostituiti da simboli. Come i font anche le icone sono vettoriali, il che significa che ogni carattere può essere visualizzato a qualsiasi dimensione senza modificarne la qualità. Tramite i Css è possibile modificare il colore come con il testo.

Ottimizzare le immagini

Un aspetto fondamentale riguarda i problemi con la "user experience", ovvero, se un'immagine impiega tanto tempo a caricarsi, l'utente potrebbe rinunciare ad aspettare l'intero caricamento, abbandonando la pagina, questo potrebbe essere dovuto in parte alle immagini, in parte alla connessione lenta. Una soluzione per caricare le immagini più velocemente è quella di diminuire la dimensione comprimendole, salvandole cioè con un livello di qualità inferiore o con un formato differente. Con il Responsive design, l'immagine potrebbe essere ridimensionata per diverse misure di schermo; quindi, si vorrebbe l'immagine con la dimensione maggiore, anche se questo implicherebbe scaricare un'immagine di grandi dimensioni quando non necessario.

Immagine con dimensioni flessibili

La funzione `float` viene spesso utilizzata per inserire testo e immagini all'interno di una pagina, se questa è visualizzata in uno schermo alle massime dimensioni può andar bene la dimensione massima dell'immagine, ma nel momento in cui si restringe la viewport è necessario ridurre anche le dimensioni dell'immagine. Questo può essere fatto impostando una larghezza del 50%, il che significa che occuperà metà del contenitore. Le media query in questo caso possono essere utili per impostare differenti misure dell'immagine, per diverse dimensioni della viewport. Partendo da uno schermo grande per poi rimpicciolirlo, utilizzeremo come proprietà `width: auto`, il che significa che verrà visualizzato alla sua dimensione naturale. Quando si aggiungono le immagini è spesso più semplice passare da schermo largo a stretto, ma è possibile anche il viceversa. L'uso della proprietà `max-width` è fondamentale per far funzionare correttamente un sito responsive, impostando

un `max-width=100%` diciamo al browser che l'immagine non deve essere mai più grande del contenitore che la contiene.

3.3.5 Bootstrap

Bootstrap è un toolkit, una libreria che al suo interno contiene una serie di componenti che permettono la creazione applicativi responsive e mobile first [16].

Bootstrap supporta nativamente il design responsive; quindi, permette la fruizione dell'applicazione su diversi dispositivi. Nasce nel 2011 con lo scopo di sviluppare una libreria interna che gestisse interamente i layout, oggi rappresenta la risorsa principale per disegnare lo stipo di un sito, inoltre è possibile integrare con facilità i componenti HTML5 e Css3. Una delle caratteristiche di questo framework front-end è proprio il responsive design, infatti aiuta gli sviluppatori nella realizzazione di applicativi che in automatico supportano risoluzioni differenti.

3.3.6 Angular

Angular è una piattaforma è un framework utilizzata per applicazioni client single-page, è scritto, e implementa funzionalità di base e opzionali sempre come librerie TypeScript. Per costruire una UI responsive, ci sono diversi modi, dall'uso di media query e Css, all'uso di librerie come Angular Material. Un componente utile per questo scopo è il `BreakpointObserver` dalla libreria CDK di Angular Material, tramite questo componente gli elementi verranno resi in maniera differente in base al dispositivo di destinazione.

Il componente `BreakpointObserver` permette di definire diversi tipi di schermo:

```
export declare const Breakpoints: {  
  Small: string;  
  Large: string;  
};
```

Dopo di che è possibile sottoscrivere il tipo di schermo in questo modo: Una volta che il servizio `BreakpointObserver` è stato iniettato nel componente, possiamo sottoscrivere il tipo di schermo desiderato:

```
this.cardsLayout = merge(  
  this.breakpointObserver.observe([Breakpoints.Small]).pipe(  
    map(({ matches }) => {  
      if (matches) {  
        return this.getHandsetLayout();  
      }  
      return this.getDefaultLayout();  
    })),  
  this.breakpoints$);
```

Nell'esempio vediamo la definizione del solo schermo piccolo, ma al suo interno è possibile definire anche altre dimensioni.

Per poter poi visualizzare l'UI destinata ai dispositivi mobili:

```
getSmallLayout(): Layout {
  return {
    name: 'Handset',
    gridColumns: 1,
    layoutItem: [
      { title: 'Card 1', cols: 1, rows: 1 },
      { title: 'Card 2', cols: 1, rows: 1 },
      { title: 'Card 3', cols: 1, rows: 1 }
    ]
  };
}

getDefaultLayout(): Layout {
  return {
    name: 'Web',
    gridColumns: 2,
    layoutItem: [
      { title: 'Card 1', cols: 2, rows: 1 },
      { title: 'Card 2', cols: 2, rows: 1 },
      { title: 'Card 3', cols: 2, rows: 1 }
    ]
  };
}
}
```

Sostanzialmente le funzioni permettono di modificare il numero di righe e colonne dedicate al componente card, a seconda che il dispositivo sia di tipo Small, o di qualsiasi altro tipo (Default). Per poter visualizzare le card mediante HTML si utilizzano le proprietà `colspan` e `rowspan` che permettono di aggiornare in modo asincrono il numero di colonne e righe dedicate all'elemento.

```
<div>
  ...
  <mat-grid-list [cols]="(cardsLayout | async)?.gridColumns"
    rowHeight="350px">
    <mat-grid-tile *ngFor="let card of
      (cardsLayout | async)?.layoutItem"
      [colspan]="card.cols" [rowspan]="card.rows">
      <mat-card class="dashboard-card">
        ...
```

```
    </mat-card>
  </mat-grid-tile>
</mat-grid-list>
...
</div>
```

3.3.7 React

React, libreria JavaScript sviluppata da Facebook, ha lo scopo di semplificare lo sviluppo di interfacce visive, dividendo l'UI in una collezione di componenti. Per la realizzazione di applicazioni web responsive si utilizza un pacchetto npm chiamato `react-responsive`, esso combina media queries e breakpoint per definire gli elementi nel DOM che si vogliono mostrare o nascondere a seconda del dispositivo [17]. L'installazione avviene tramite il comando `npm install react-responsive`, seguita dall'importazione `import useMediaQuery from 'react-responsive'`. Dopo l'installazione i componenti potranno utilizzare i breakpoint come nell'esempio:

```
import { useMediaQuery } from 'react-responsive'

export default function Layout({ children }) {
  const isBigScreen = useMediaQuery({
    query: '(min-device-width: 1224px)' })
  const isSmallScreen = useMediaQuery({
    query: '(max-width: 1224px)' })

  return (
    <div>
      {isSmallScreen ? (
        <LeftSidebar small={true} />
      ) : (
        <LeftSidebar />
      )}
    </div>
  )
}
```

3.4 Workflow Responsive Web Design

Il contenuto è la parte più importante di qualsiasi sito web, una volta decisi i contenuti il passo successivo consiste nell'organizzare una struttura, chiamata architettura dell'informazione (IA). I principi generali di IA si applicano anche ai

siti responsive, ma bisogna assicurarsi che l'architettura del sito funzioni su schermi piccoli, dove potrebbe non esserci lo spazio per una navigazione dettagliata. L'IA e il design devono essere abbastanza flessibili per accogliere i cambiamenti futuri. Il primo passo dell'organizzazione del contenuto consta nella creazione di un semplice schema ad alto livello che verterà sulla navigazione principale del sito. Quando si crea l'outline per la prima volta i contenuti dovranno essere generici, i dettagli saranno aggiunti via via.

Dopo di che si può iniziare a pensare a come il contenuto dovrà apparire sullo schermo, essendo la parte più importante del sito, si dovrà costruire il sito intorno al contenuto non il contrario. Il modo migliore consiste nel creare una pagina senza stile, con tutti i contenuti. Fare questo prima di avere dei prototipi o schizzi permette di pensare al ruolo che ogni contenuto avrà all'interno della pagina. È utile pensare ai contenuti che andranno nella pagina come separati in modo da poterli spostare nella pagina quando si crea il prototipo. Ecco un esempio dei principali componenti che devono essere presenti all'interno del sito:

- Logo
- Barra di ricerca
- Navigazione principale
- Navigazione secondaria
- Link informativi
- Copyright
- Pubblicità
- Link ai social media
- Login

e a seconda del tipo di pagina possono essere necessari componenti come

- Titolo
- Corpo
- Contenuti secondari
- Sinossi
- Autore
- Data
- Citazione/i

A questo punto inseriamo i contenuti all'interno degli elementi html.

3.4.1 Design Lineare

Il design lineare consiste nell'aggiungere l'HTML nell'ordine in cui dovrebbe essere letto. Utilizzare il design lineare non stilizzato, funzionerà anche su schermi stretti e anche su browser che non supportano gli stili.

3.4.2 Contenuto gerarchico

Arrivati a questo punto è necessario scegliere quali contenuti sono più importanti per inserirli prima all'interno della pagina. Non si deve ancora pensare il contenuto in termini di layout perché con il responsive web design potrebbero non essere sempre nello stesso posto; infatti alcuni elementi potrebbero trovarsi sopra o sotto altri all'interno delle pagine più piccoli, e trovarsi in posizioni differenti nelle pagine più grandi.

3.4.3 Pensare al Layout

Una volta decisi i contenuti per ogni pagina è possibile iniziare a lavorare su come sarà ogni pagina, pensando il design per diverse dimensioni, ma prima di passare ai prototipi è utile fare degli schizzi grezzi. Stephen Hay propone di iniziare con dei piccoli schizzi contenenti pochi dettagli, in modo da provare tante idee velocemente, e poi renderli più dettagliati nel momento che le idee diventano più articolare. Gli schizzi potranno essere rappresentati su carta, o su dispositivi che ne consentono il disegno (fig.3.2). Inoltre, occorre pensare subito a come la pagina

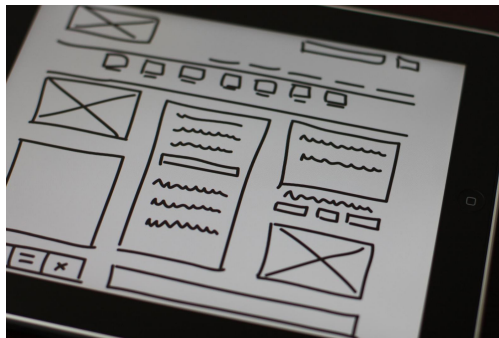


Figura 3.2: Schizzi preliminari possono essere realizzati direttamente su dispositivi che permettono il disegno.

apparirà rispetto ai vari dispositivi. Utilizzando l'approccio *small-screen-first* si parte dalla realizzazione degli schizzi prima per dispositivi più piccoli, questo perché permetterà di inserire solo i contenuti effettivamente necessari, questo non significa che non si possa passare anche alla realizzazione dei design per schermi più grandi, anzi molto spesso è utile lavorare quasi in parallelo. L'approccio *mobile first* è un approccio che invece consiste nel realizzare un design dando priorità agli utenti che usano i dispositivi mobili, in questo tipo di approccio il mobile è considerato più importante di quello non mobile. Assicurarsi che il sito funzioni con il touch screen è molto importante, ma quando si progetta un sito reattivo è meglio essere il più flessibili possibile, con l'obiettivo che il sito funzioni bene con qualsiasi tipo

di dispositivo, ormai non vi è più una distinzione anche i computer possono essere touch screen; quindi tutte le dimensioni sono ugualmente importanti.



Figura 3.3: Esempi di layout di siti web.

3.4.4 Prototipi

Una volta che si ha l'idea approssimativa del prototipo è necessario mettere tutto insieme, avendo però una chiara idea di come funzioneranno le cose sullo schermo. Ci sono due modi diversi per visualizzare il layout:

- Wireframe: disegni statici, utilizzati per lo più in passato;
- Layout responsive: con il design responsive non si ha più un design statico, ma questo cambierà in base alle dimensioni.

I Wireframe, per un sito web, sono modelli in cui si indica dove i vari componenti andranno all'interno della pagina, non comprende i dettagli visivi, ma stabilisce solamente l'esatta posizione per ogni componente, dato che il layout cambia a seconda delle larghezze di viewport. Un prototipo è un modello che mostra anche come il componente funzionerà, oltre che come appare. Non necessariamente i prototipi vengono realizzati con lo stesso metodo del prodotto finale; quindi, si potrebbe realizzare un prototipo con HTML, o un'applicazione di prototipazione. Generalmente, si all'inizio del processo si realizza un prototipo a bassa fedeltà, e man mano che si più avanti, questo diventerà più approfondito. Il prototipo mostra le basi del layout e di come questo risponderà ai cambiamenti di larghezza, oltre a questo è meglio che i prototipi mostrino solo l'aspetto visivo, senza aggiungere complessità per vedere come il processo funziona. Sempre in questa fase non sarà necessario pensare ai dettagli di colori, caratteri etc., questo consentirà al design di concentrarsi maggiormente sul posizionamento dei componenti. Quando si realizzano i prototipi, occorre pensare anche a quali pagine dovranno essere

prototipate, perché alcune delle pagine avranno probabilmente lo stesso layout; quindi, non necessario realizzarlo per tutte le pagine, inoltre, ci saranno elementi che sono coerenti in tutte le pagine.

Generalmente si realizzano i template più complessi (2-5 pagine) e quelle più semplici si derivano da queste. Un prototipo reattivo è una pagina web "reale", nel senso che si può vedere in un browser, ma è solo il layout di base, con un aspetto simile a un wireframe. Spesso si è tentati di fare prima la prima pagina, perché è quella che "fa" il sito, ma in realtà è meglio partire con pagine più semplici, e una volta ottenuto un layout per queste si può passare alla home page.

Framework

Esistono dei framework che permettono di scaricare una base di codice HTML e Css, il che rende la creazione dei prototipi più rapida. Per la creazione di prototipi reattivi esistono diversi strumenti, sia per applicazioni web desktop che mobile, ma la maggior parte di questi non creerà veri e propri prototipi reattivi, ma permette la realizzazione di wireframe statici separati per le diverse dimensioni.

3.4.5 Visual Design

Una volta completato il prototipo, si passa alla realizzazione del design visivo, questa è la fase in cui si aggiungono i colori, la tipografia e il branding, anche questo sarà fatto diversamente per le diverse dimensioni. Tradizionalmente venivano mostrati su una o più comps di Photoshop (composizione di livelli), ma con un sito reattivo, non si vuole mostrare loro solo una sola vista statica della pagina. Una tecnica utilizzata, per poter mostrare le interfacce ai clienti sono le *Style Tiles*:

“Un prodotto di design composto da caratteri, colori ed elementi dell’interfaccia che comunicano l’essenza di un marchio visivo per il web” [18].

Le style tiles sono il risultato di una discussione con i clienti, in cui esprimono il loro gradimento sull’interfaccia. Si tratta di opzioni campione che stimolano il confronto con gli stakeholders su un comune linguaggio visuale. Le style tiles sono un punto di partenza flessibile, permettono di comunicare in maniera più comprensibile con i clienti, spesso i mockup sono troppo dettagliati e danno al cliente un’idea di finito, rendoli poco partecipi nella scelta, le style tiles sono invece come i campioni di tessuto utilizzati dai designer di interni per la progettazione di una stanza. Per la realizzazione delle style tiles si possono usare gli strumenti di editing delle immagini come Photoshop, o lavorare direttamente nel browser. In entrambi i casi, è possibile iterare quanto necessario per arrivare ad un accordo (fig:3.4).

Il processo reattivo richiede continui test man mano che si procede, come accade

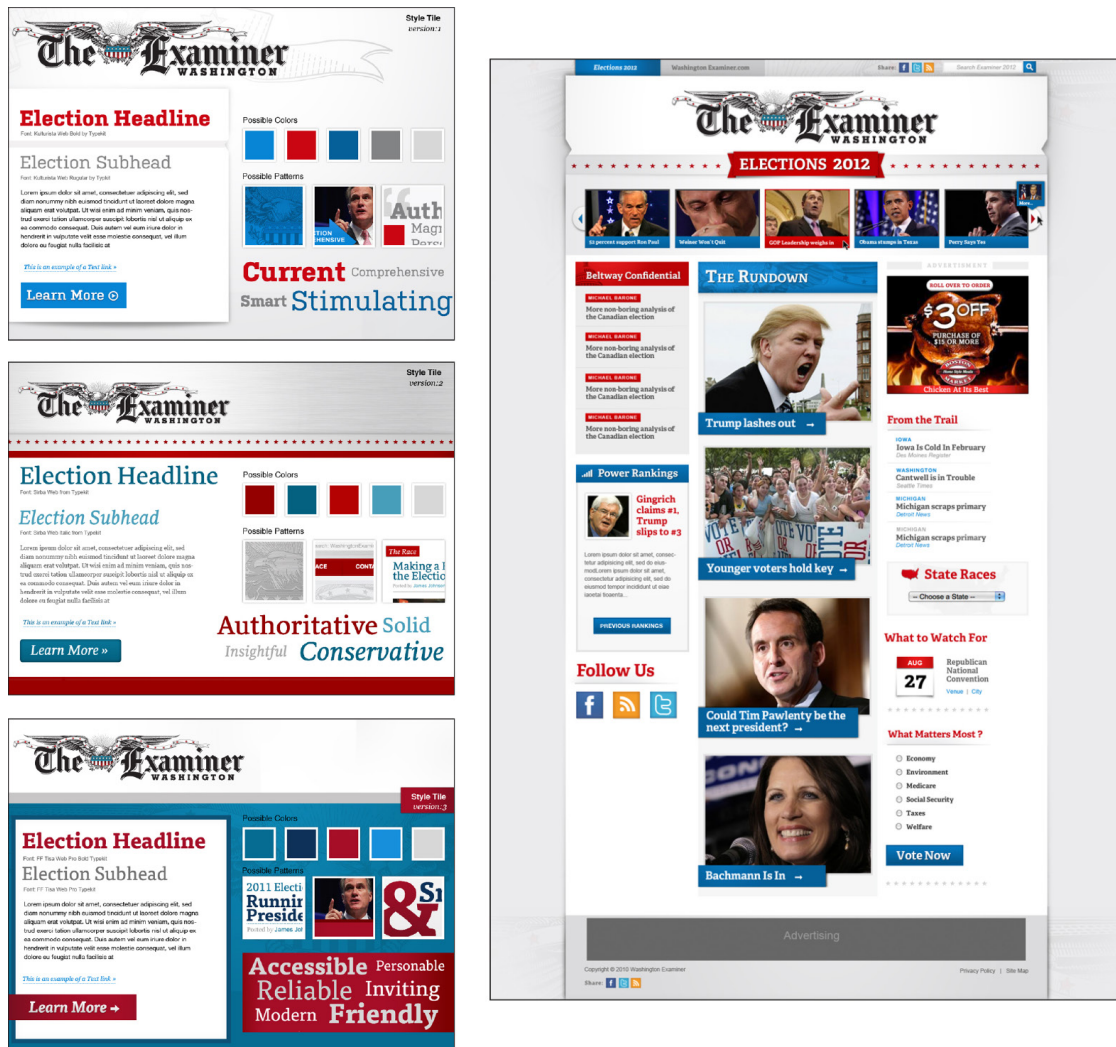


Figura 3.4: Le tre style tiles sono state utilizzate per arrivare al design finale del sito "examiner"

anche con i prototipi HTML che dovrebbero essere testati sui diversi dispositivi. Se un layout funziona bene in teoria ma poi non funziona su particolari dimensioni degli schermi è meglio scoprirlo in fase di prototipazione piuttosto che a progetto finito.

3.4.6 Responsive Design Tools

Esistono vari tools per la realizzazione di design responsive:

- Adobe Photoshop

- Adobe Illustrator
- Adobe XD
- Adobe Dreamweaver

3.4.7 Responsive nella pratica

Di base gli obiettivi minimi da porsi per realizzare un'esperienza utente positiva sono:

- Adattare il layout al maggior numero di risoluzioni schermo;
- Adattare le dimensioni delle immagini, e in generale tutti i contenuti alla risoluzione e dimensione dello schermo;
- Utilizzare immagini ottimizzate
- Semplificare il layout degli elementi presenti, per dispositivi con schermi più piccoli, e nascondere gli elementi non essenziali;
- Fornire un'interfaccia utente funzionale per il touch screen;
- Sfruttare, quando serve, funzionalità adatte al mobile;

Capitolo 4

User experience

4.1 Introduzione

La user experience, o UX, è la cosiddetta "Esperienza Utente", tutto ciò che ha a che fare con l'esperienza vissuta da una utente. Possiamo definirla come l'insieme delle emozioni, percezioni e reazioni provate da una persona, nel momento in cui entra in contatto con un'azienda, un prodotto o un servizio. L'utente, la persona o lo user, sono il fulcro della definizione, l'esperienza d'uso è quindi estremamente soggettiva, anche se si è constatato che persone simili (stesso contesto culturale, sociale) possono avere reazioni simili, usi e consumi delle persone possono cambiare le aspettative, le reazioni, avendo così una diversa percezione della Ux. Altro elemento che ne influenza la percezione è il tempo, infatti, accade spesso che un utente abbia percezioni diverse della UX a seconda della quantità di utilizzo, e altro aspetto fondamentale sono le aspettative che un utente ha su un prodotto, risulta infatti che queste provochino maggiore eccitazione all'inizio dell'uso e tendano a svanire subito dopo [19].

La Ux è una dimensione della progettazione, il cui obiettivo è quello di generare prodotti belli e piacevoli, ma allo stesso tempo ergonomici e funzionali.

Il termine UX, introdotto da Donald Norman [20] comprende le varie aree di interesse per la creazione di un sito web, quali [21]:

- Marketing;
- Comunicazione;
- Branding;
- Design;
- Usabilità.

Il concetto di User Experience è strettamente correlato alle qualità ergonomiche ed edoniche di un prodotto, quest'ultime consentono di provare emozioni tali da soddisfare il bisogno edonico ¹ di una persona. Queste caratteristiche permettono di raggiungere i "be-goals" di una persona.

4.2 UX, UI, e usabilità

Nonostante il termine UX sia in uso ormai da parecchio tempo, viene spesso confuso con i termini User interface (UI) e usabilità. In realtà questi hanno dei significati ben distinti:

- User Experience: comprende tutto ciò che va dalla facilità d'uso al coinvolgimento degli utenti, e impatto visivo. In sostanza, rappresenta il grado di esperienza vissuta da una persona nell'uso del prodotto o servizio.
- User Interface: rappresenta l'insieme degli elementi che permettono l'interazione.
- Usabilità: indica il grado di efficacia, efficienza, la semplicità d'uso con cui l'uomo può relazionarsi con il prodotto.

Si tratta di concetti strettamente correlati, ma distinti e autonomi. Un prodotto può essere usabile senza necessariamente avere un'esperienza d'uso piacevole, o viceversa.

Vantaggi usabilità

Un'esperienza d'uso piacevole crea, con gli utenti, un rapporto di fiducia che fa sì che gli utenti rimangano nel sito. Per potersi assicurare un maggior successo, si deve tenere conto delle esigenze dei suoi utenti, in questo modo gli utenti rimarranno soddisfatti.

4.3 Progettare una buona UX

Come prima cosa è necessario capire a quale tipologia di utente sarà dedicato il servizio, poiché questo potrebbe cambiare notevolmente progettazione del prodotto. Inoltre, sarebbe bene che ci fossero feedback da parte dell'utente durante tutto il processo di progettazione, questo aiuterebbe a comprenderne il grado di soddisfazione. Ma generalmente questo non avviene, anzi, la Ux viene presa in considerazione solo alla fine del processo produttivo, quando le modifiche delle funzionalità e

¹edonico: bisogno di provare gradevoli emozioni.

caratteristiche risultano più complesse. Generalmente, la progettazione di un sito web richiede un ciclo iterativo, chiamato *the wheel* (la ruota), che consta in quattro macro fasi, che possono e devono essere ripetute più volte:

1. **Ricerca/Analisi:** Fase in cui si analizza la tipologia di utenti a cui sarà destinato il sito, e si cerca di capire quali sono i loro bisogni mediante interviste, questionari, etc. Vengono definiti i requisiti e dettagli del sito;
2. **Progettazione/Design:** Fase in cui viene eseguito il brainstorming, prototipi, mappe visive etc. In questa fase verranno realizzati i disegni dell'interazione uomo-macchina, schizzi sulle possibili interfacce. Sarà richiesta l'interazione con l'utente per verificare se per lui, le interfacce realizzate, risultano semplici da utilizzare, e da capire, in caso contrario sarà, ovviamente, necessaria una revisione e modifica del design fino a che non si ottiene l'approvazione del cliente.
3. **Implementazione:** Il design verrà trasformato in un prototipo reale utilizzabile dall'utente. L'interazione con l'utente in questa fase pone le basi per la fase successiva.
4. **Valutazione:** In questa fase vengono effettuati dei test che serviranno a valutare il sito, osservando gli utenti in una sessione di navigazione, oppure mediante un'analisi dell'interfaccia, da parte di esperti che seguono determinate linee guida e parametri (valutazione euristica). Verrà chiesto all'utente lo svolgimento di qualche azione all'interno del sito, per verificare se riesce a svolgerle in modo veloce, e semplice. Fondamentale, nella fase di analisi, è verificare se l'utente alla fine dei test risulterà affaticato dalla vista dell'interfaccia, in particolare, potrebbero essere necessarie delle modifiche ai colori del sito.

La ruota è un insieme di linee guida, che non devono essere usate in modo rigido, ma che aiutano per la realizzazione di una buona UX. Un errore tipico che si commette è quello di iniziare la fase di implementazione troppo presto, è necessario effettuare prima un'analisi completa, per potersi assicurare una buona riuscita del progetto. Il processo visto è molto generico, chiaramente ogni progetto avrà le sue specifiche e direttive, per questo motivo, quando si prende in considerazione un progetto reale occorre tenere in considerazione:

- *Tolleranza al rischio:* questa indica quanto è accettabile un errore durante l'utilizzo del servizio, prodotto, etc. Per fare un esempio concreto, in un software in cui si effettuano calcoli per la scelta di un colore si potrà accettare un errore di calcolo ben maggiore rispetto ad un software per il calcolo di traiettoria di un missile. Quando la tolleranza al rischio è molto bassa, le fasi della ruota devono essere eseguite con grande accuratezza.

- *Disponibilità delle risorse*: con risorse si intendono quelle economiche, ma anche risorse umane, cioè i professionisti che parteciperanno alla realizzazione del progetto, mettendo a disposizione le loro competenze nel campo.
- *Ambiente di sviluppo*: è necessario prendere in considerazione la cultura organizzativa in cui verrà sviluppato il prodotto, infatti, mentre per una start-up si potrebbe essere maggiormente predisposti ad un UX più innovativa, in una società leader nel suo settore di mercato potrebbero non essere troppo attratti da novità e variazioni.

Un sito web per poter avere una buona esperienza utente deve poter garantire:

- **Usabilità**: il sito deve essere prima di tutto semplice da usare, ogni elemento al suo interno deve aver un ben preciso scopo, che sia chiaro agli utenti, garantendo, inoltre che l'esecuzione di azioni, come acquisto di un prodotto, o il raggiungimento di una pagina specifica, siano intuitive.
- **Utilità**: Il sito deve permettere agli utenti lo svolgimento di un qualcosa di concreto, e quindi essere utile.
- **Accessibilità**: Il sito deve essere fruibile senza difficoltà.
- **Velocità**: la navigazione del sito deve essere fluida e veloce.
- **Visibilità**: il sito deve essere disponibile per il maggior numero di dispositivi, mobile, desktop, tablet, etc.
- **Grafica**: il sito deve presentare layout, stile, colori in linea con il brand che lo identifica.

4.3.1 Digital Branding

Quando un utente naviga in un sito per la prima volta, deve riuscire a capire quale è l'identità, lo scopo e il settore di business a cui il sito fa riferimento. Questo permetterà al sito di raggiungere una maggiore credibilità e fiducia.

I siti devono quindi, essere realizzati pensando prima di tutto a quali sono gli obiettivi di comunicazione e marketing, un utente, durante la sua navigazione, sarà portato a fare confronti tra ciò che vede nel sito, e ciò che si aspettava di trovare. L'utente naviga il sito con la cosiddetta "attenzione selettiva", il che significa che se il sito non presenta i suoi obiettivi in modo efficace, l'utente sarà portato a continuare altrove la sua ricerca. Un buon User Experience Designer, deve essere capace di trasmettere messaggi facili da comprendere anche per l'utente che capita nel sito per sbaglio, infatti, nella maggior parte dei casi gli utenti capitano su un sito, per curiosità senza particolari obiettivi o aspettative. In questi casi si parla di

serendipity, ovvero, quando l'utente capita in maniera casuale in siti non conosciuti, nei quali fa scoperte inattese e non programmate, le pagine web devono saper ricevere anche questo tipo di visitatori.

Primo impatto

Starting interaction: il momento in cui l'utente entra per la prima volta in contatto con il sito. L'utente ha bisogno di feedback esaustivi che lo informino sul "chi" e "cosa" del sito in questione. Banalmente, quindi ogni sito deve avere un nome che lo identifichi, questo deve essere incalzante e descrittivo, rappresenta il biglietto da visita per la homepage. Legato al nome, altro aspetto fondamentale, è l'indirizzo del sito, è importante che questo permetta di accedere in maniera univoca al URL, alcuni accorgimenti potrebbero essere i seguenti:

- Memorabilità: l'indirizzo deve essere semplice da ricordare e allo stesso tempo credibile;
- Estensione: i domini principali da acquistare sono certamente .it e .com per l'italia, ma ultimamente altre estensioni sono diventate popolari.
- Brand identity: è buona norma registrare un dominio di secondo livello che coincida, o differisca di poco dal nome della marca;
- Brand protection: è consigliabile acquistare domini simili, e tutte le principali estensioni onde evitare azione scorrette della concorrenza.
- Trovabilità: un buon dominio attrae il traffico e ne aumenta la visibilità
- E-mail: acquistando i domini si ha diritto anche ai relativi indirizzi di posta elettronica personalizzati.

Come prima regola fondamentale un sito web deve contenere il logo, questo deve essere presente in ogni pagina (tipicamente in alto a sinistra) e al click deve rimandare l'utente alla pagina principale. Per alcuni tipi di siti è utile dichiarare in primo piano, senza necessità di scorrimento, la *value proposition*, questa conterrà una breve descrizione che serve a dare il benvenuto agli utenti, chiarisce a chi si rivolge il sito, e presenta la proposta commerciale. È una best practice aggiungere un area in cui si contengano tutte le informazioni dell'azienda, generalmente chiamato "About us" o nella versione italiana "Chi siamo", in questa sezione verranno inseriti recapiti reali dell'azienda, e serviranno a dare una maggiore credibilità al sito, facendo capire all'utente che dietro vi è un'azienda reale.

Reputazione e fiducia

Per ottenere la fiducia delle persone è fondamentale che ci sia uniformità tra le varie pagine, di stili, font, colori, questi devono corrispondere agli obiettivi di business e comunicazione. L'inserimento di foto e presentazioni del team permettono agli utenti di vedere l'azienda senza valì, il che garantirà da parte loro una maggiore fiducia. Altro elemento da tenere in considerazione sono i feedback, recensioni che i clienti possono lasciare all'interno del sito.

4.4 Psicologia della homepage

Durante la progettazione della homepage è doveroso tenere in considerazione il fatto che il cervello umano interagisce con l'interfaccia, e ne influenza l'uso sulla base di codificati principi mentali. Gli aspetti psicologici più diffusi sono:

- *Learnability*: La home deve essere semplice da capire, in modo tale da permettere all'utente di svolgere velocemente le proprie azioni.
- *Likeability*: Un utente è maggiormente attratto e quindi soddisfatto da una home piacevole, il che lo tratterrà sul sito.
- *Memorability*: La home deve essere facile da ricordare, questo permetterà all'utente di ricordarsi come usarla anche dopo lungo periodo di inutilizzo.
- *Affordance*: La home deve ridurre il carico cognitivo nell'utilizzo, tramite elementi grafici deve invitare l'utente a una certa modalità di utilizzo.
- *Consistency*: I colori, le icone, le immagini devono avere una certa coerenza all'interno del sito.
- *Errors rate*: Si deve ridurre la potenzialità di errore da parte dell'utente, non permettendo percorsi scorretti, o percorsi irreversibili.

4.4.1 Homepage

La home deve avere tratti visivi caratteristici e contemporaneamente essere coerente con le altre pagine del sito. Elementi come header, body e footer, devono essere presenti in ciascuna pagina nella stessa posizione, la modalità di scorrimento deve essere sempre coerente, o verticale o orizzontale. Le voci nel menu devono contenere sempre gli stessi nomi, adottando una convenzione di denominazione regolare, tra i vari oggetti. I font e i colori come già specificato devono essere sempre invariabili. La home e le sotto pagine devono essere strettamente correlati, ogni collegamento deve essere progettato in modo preciso, evitando di generare pagine orfane che

non permettano di andare da nessuna parte o modalità *back only*, in cui l'utente è costretto a tornare indietro con i soldi tasti del browser.

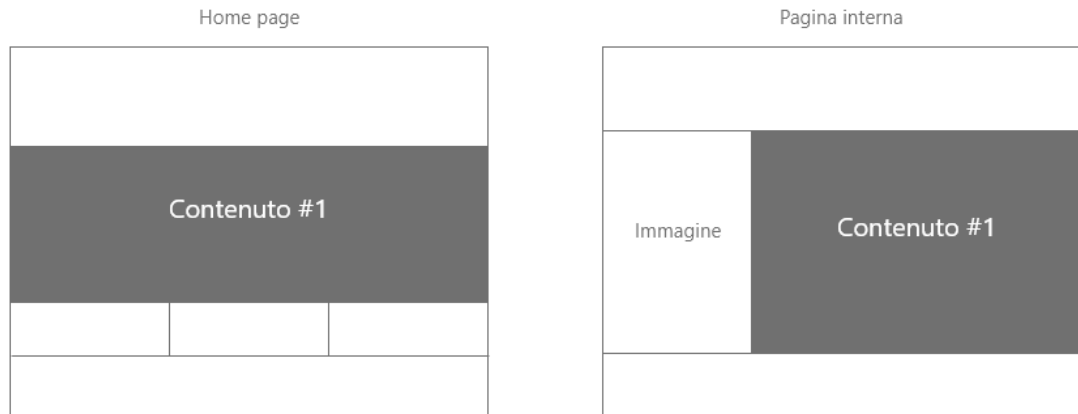


Figura 4.1: Biunivocità dei contenuti tra homepage e pagine secondarie

4.4.2 Landing page

Una best practice consiste nel realizzare ogni pagina come se fosse una landing page, letteralmente "pagina di atterraggio", in cui si fa risaltare la *Call to Action*.

La landing page è la pagina in cui arriva dopo aver cliccato un collegamento ipertestuale di una campagna pubblicitaria, nella quale si chiede all'utente di svolgere una sola azione (compilare un modulo, acquistare un prodotto, iscriversi etc.).

Nel marketing esistono due tipologie principali di landing page:

- *Reference*: landing page di consultazione, essa contiene informazioni importanti;
- *Transactional*: questa spingerà l'utente a completare un'azione;

Per realizzare una landing page efficace occorre progettare con attenzione le componenti al suo interno:

- Titolo e sottotitolo
- Hero image: immagine di alta qualità
- Descrizione accurata del prodotto, offerta, etc
- Elenchi puntati, che permettano una maggior comprensione
- Testo persuasivo che metta in evidenza l'invito all'azione

Altri fattori meno importanti ma comunque rilevanti sono:

- formattazione e impaginazione
- numero di campi da compilare
- pulsanti di condivisione



Figura 4.2: Generica struttura landing page

Call to action

La Call to action (CTA) è il bottone che permette all'utente di compiere delle azioni che sono l'unione di ciò che voi volete che l'utente faccia, e ciò che lo stesso utente vuole fare. Il pulsante è generalmente raggiungibile senza sforzo, o quasi da qualsiasi punto. È importante notare che ci sono delle convenzioni da seguire nella realizzazione delle CTA, tramite il bottone l'utente deve capire immediatamente cosa succederà dopo averlo premuto, per questo motivo si consigliano frasi semplici come "Iscriviti", "Scarica qui" e così via. Il bottone deve essere facilmente visibile, e "toccabile" dai dispositivi mobili, e se possibile, è consigliato inserirlo nella *above the fold*.²

²Area della pagina visibile immediatamente nello schermo dell'utente senza necessità di effettuare lo scrolling.

Capitolo 5

Test di usabilità

5.1 Introduzione

I test di usabilità sono una metodologia di ricerca UX e sono una parte essenziale del web design. In una generica sessione, i test vengono condotti su un partecipante alla volta, al quale verrà chiesto lo svolgimento di compiti all'interno del sito, o prodotto, in esame. I test permettono di avere dei feedback sulla progettazione del sito e di scoprire informazioni sui dati quantitativi e qualitativi.

Le metodologie, utilizzabili dai team di test, possono essere varie, tuttavia, ogni metodologia presenta sempre due documenti fondamentali per la realizzazione dei test: il piano di test, e il report [22].

5.1.1 Piano di test

Il piano di test è un documento che viene preparato prima dell'esecuzione dei test. Al fine di realizzare un buon piano di test, questo deve rispondere fondamentalmente a tre domande:

1. Cosa vogliamo da questi test?
2. Come verranno condotti?
3. Cosa verrà chiesto agli utenti durante il test?

Obiettivi

Gli obiettivi di un piano di test rispondono alla domanda "*Cosa otterremo da questo test?*", questi, quindi, delimitano il piano di progettazione del test.

Scenari di test

Nel piano di test devono essere presenti tutti gli scenari di test, questi possono andare dal generale allo specifico, e potranno essere approfonditi al fine di aiutare chi esegue il test. Gli scenari descrivono delle situazioni immaginarie cui saranno sottoposti gli utenti, generalmente composti da tre parti:

- Descrizione: Parte che viene letta al tester al fine di comprendere la situazione;
- Risultati attesi: descrizione su come affrontare correttamente la situazione;
- Domande: specifiche sullo scenario.

Limiti dei materiali di prova

Idealmente si vorrebbero eseguire i test su progetti completi e funzionanti, ma il più delle volte, i test vengono eseguiti su prototipi, schizzi, non ancora del tutto terminati.

Domande pre-test e post-test

In alcuni casi può essere utile proporre all'utente due tipi di domande: prima e dopo del test. Il primo tipo di domande verranno poste prima di mostrare il prototipo, queste possono essere utili per testare le aspettative rispetto a ciò che stanno per vedere e il livello di esperienza generale. A seconda della metodologia di test utilizzata queste possono essere questionari o domande poste di persona ai partecipanti registrando le risposte sul momento. Le domande post-test servono invece a valutare le impressioni generali e a cogliere i punti deboli o suggerimenti da parte degli utenti

5.1.2 Report

Dopo aver eseguito i test, è necessario raccogliere i dati e presentarli sotto forma di report al team di progettazione.

I report possono contenere informazioni varie, a seconda delle richieste, ad esempio, possono contenere il pensiero degli utenti, esposto in maniera esplicita, oppure possono essere presenti dei suggerimenti basati sulle osservazioni dei tester. Tuttavia, è importante che il report sia capace di superare queste sfide:

- Sintesi: i report sono spesso difficile da sintetizzare, nel momento in cui si riportano le sole osservazioni degli utenti, queste potrebbero essere complesse da captare per i progettisti, specie se non correttamente contestualizzate, si può quindi pensare di aggiungere spiegazioni. Ma spesso, gli abbellimenti sono delle distrazioni perché distolgono dai possibili miglioramenti da parte dei progettisti.

- Obiettività: cercare di sintetizzare le osservazioni degli utenti, potrebbe sembrare che queste siano solo delle critiche.
- Accessibilità: a seconda dei metodi utilizzati i test possono risultare troppo complessi, al fine di sviluppare un rapporto più comprensibile si potrebbe pensare di generalizzare le osservazioni, questo potrebbe però portare ad una visione distorta dei risultati.

Obiettivi

I report possono avere obiettivi diversi, ma, un buon rapporto di usabilità deve essere focalizzato su almeno uno di questi scopi:

- Diagnosi dei problemi
- Stabilire le priorità
- Convalidare i concetti di design
- Convalidare priorità dei requisiti
- Imparare dagli utenti e dai loro comportamenti

Affermare lo scopo all'inizio del documento può aiutare per rimanere focalizzati su di esso, e a stabilirne il contesto.

Priorità

Un buon report deve tenere bene a mente quali sono i problemi che devono essere risolti nell'immediato, e quelli invece meno rilevanti per i progettisti. Al fine di mantenere il documento fruibile, i risultati direttamente legati alle specifiche devono avere una maggiore priorità rispetto a quelli riguardanti le informazioni di supporto. Il trucco è quello di bilanciare le esigenze del team di progetto, al fine di non rendere, ad esempio, il report troppo teorico [22].

Riassunto e analisi

Probabilmente la parte fondamentale di un rapporto di usabilità è il riepilogo. I report possono sembrare pagine piene di dati, o di feedback schietti. Una buona pagina di riepilogo aiuta i lettori a concentrarsi sui risultati più importanti. Il riepilogo ha lo scopo di descrivere i meccanismi di test, scenari analizzati, e gruppi di utenti che hanno partecipato al test. La metodologia di uno studio di test deve comprendere questi sei punti [22]:

- Tecnica utilizzata;

- Numero di partecipanti;
- Tipologia di utenti partecipanti;
- Materiali utilizzati per lo studio;
- Scopo dello studio;
- Numero di scenari analizzati;

L'analisi fornisce una spiegazione unificata dei risultati, l'inserimento di questa nel documento dipende dal tipo di contenuto. Altro modo di interpretare i risultati può essere quello di offrire delle raccomandazioni per poter migliorare il sito sulla base dei dati raccolti.

5.2 Metodologie di User Centered Evaluation

User Centered Evaluation (UCE), è un metodo di valutazione empirico, che dipende dall'esperienza individuale nell'utilizzo del software, permettendo di analizzarne caratteristiche funzionali, qualità, e efficienza (ISO 1994) [23, 24]. La norma ISO 9241-210 "*Human-centered design for interactive system*" definisce 4 fasi principali (fig. 5.1), tra cui appunto la valutazione:

1. Capire e descrivere lo scenario d'uso;
2. Definizione dei requisiti
3. Progettazione delle soluzioni di design, basate sui requisiti
4. Validazione delle soluzioni

Obiettivi

Identifichiamo nella User Centered Evaluation tre principali obiettivi:

- Supporto delle decisioni da parte degli sviluppatori, in accordo con i principi della User Centered Design;
- Analisi problematiche (bug, malfunzionamenti, comportamenti indesiderati);
- Valutazione della qualità del prodotto da parte dell'utente finale.

Le tecniche di UCE permettono una maggiore qualità del prodotto software in virtù di una esperienza utente migliore, infatti, tramite i risultati ottenuti sarà possibile effettuare modifiche e migliorie sul sistema software.

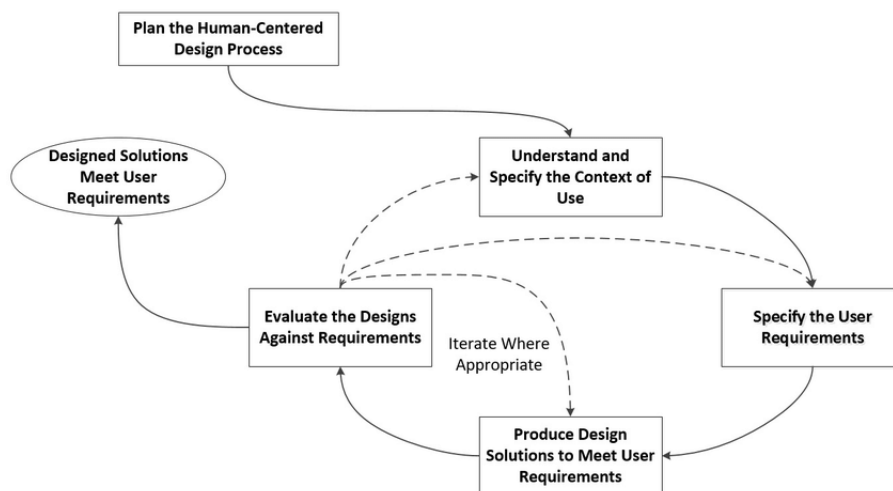


Figura 5.1: Fasi Human Centered Design

5.2.1 Tecniche di valutazione

La letteratura classifica due categorie principali di tecniche di valutazione utilizzabili in maniera indipendente, o combinata in ciascun protocollo per la valutazione:[25]

- **Tecniche descrittive:** si basano sull'analisi di un utente finale e definiscono in modo obiettivo e affidabile le problematiche presenti in un software. Questo tipo di tecnica richiede che il sistema sia in uno stato che possa essere proposto all'utente, quindi in una fase finale o un suo prototipo. I risultati delle tecniche descrittive sono dati che devono essere analizzati da esperti al fine di essere utili per sviluppi o correzioni del software. Identifichiamo tre categorie fondamentali:

- *Behaviour-based*
- *Opinion-based*
- *Usability-testing*

- **Tecniche predittive:** Il coinvolgimento dell'utente finale in questo metodo non è fondamentale, in quanto non richiedono un prototipo sofisticato, ma si basano su esperti che ispezionano il software.

A differenza delle tecniche descrittive, possono presentare problemi di oggettività e affidabilità. L'obiettivo ultimo della metodologia predittive è la raccomandazione per sviluppi futuri e prevenzione di problematiche sull'usabilità della piattaforma. Sono tecniche legate al problem solving, poiché i risultati ottenuti verranno poi utilizzati nella pratica.

Tecniche Behaviour-based

Tecniche che permettono la rilevazione di problemi riguardanti l'usabilità, si basano sull'osservare l'utente che interagisce con il sistema. Distinguiamo tre famiglie di valutazioni appartenenti a questa tipologia:

- *Tecniche basate sull'osservazione*: possono avvenire in maniera diretta (osservando l'utente durante la fase di test), o in maniera indiretta (effettuando registrazioni o log dei dati elaborati e/o immagazzinati dal software, al fine di tenere traccia dell'interazione soggetto esaminato con il sistema). Il test permette di produrre sia dati quantitativi che qualitativi.
In questa metodologia si cerca di ridurre la soggettività, standardizzando le procedure e preparando l'utente all'applicazione con una fase di training. Tali metodologie sono pensate principalmente per l'analisi di nuovi prototipi.
- *Protocolli Thinking-aloud*: La tecnica consiste nel far pensare l'utente a voce alta, nel momento in cui interagisce con il sistema, e formalizzare i pensieri per poter collezionare dei feedback qualitativi. Esistono due varianti dell'approccio CTA e RTA: il primo è il *Thinking Aloud concorrente*, ovvero i partecipanti sperimentano le operazioni e la scrittura dei loro pensieri avviene in real-time, nel secondo caso invece, *Thinking Aloud retrospettivo* i partecipanti verbalizzano i loro pensieri una volta terminati i task richiesti [26]. La prima tipologia può permettere l'identificazione delle problematiche in maniera più semplice, tuttavia, presenta due problematiche: in primo luogo l'innaturalità dell'esperienza, in secondo luogo i processi di pensiero dell'utente potrebbero essere alterati e quindi influenzare i compiti sperimentali, invalidando, quindi, anche i dati ottenuti. Il secondo metodo invece permette un ragionamento più naturale ai partecipanti, ma i dati dipendono molto dalla memoria del partecipante [27]. Una soluzione è quella di usare un approccio ibrido.
- *Tecniche di Video Confrontation*: Si basa sul thinking-aloud retrospettivo, ma in questa tipologia, vengono analizzate le parti di maggiore interesse e su queste si intervista l'utente. Le domande si concentrano sui punti salienti, creando un'intervista strutturata. Rispetto al protocollo classico, gli utenti possono formalizzare meglio le loro impressioni, avendo delle domande mirate a cui rispondere. Tramite questa metodologia è possibile ottenere una lista di problemi, e le ragioni che conducono a tali.

Tecniche Opinion-based

Le tecniche Opinion-based, si basano sull'opinione espressa dagli utenti tramite questionari o interviste. I risultati di questi permetteranno di ricavare suggerimenti di miglioramento, e valutarne la loro soddisfazione riguardo il prodotto testato.

- *Interviste*: possono essere scritte o orali, vengono generalmente utilizzate per valutare sistemi software. I principali vantaggi rispetto alle tecniche di osservazione, sono i costi e la possibilità di ottenere informazioni sull'opinione dell'utente, che difficilmente possono essere raccolte tramite la sola osservazione.
- *Questionari*: chiamati anche survey, vengono utilizzati principalmente per la valutazione dei prototipi. Essi comprendono una lista di domande strutturate a cui i tester dovranno rispondere [28]. Ogni domanda ha come scopo misurare l'usabilità di un particolare aspetto del software. Un tipico esempio è il **QUIS** (Questionnaire for User Interface Satisfaction) (fig. 5.2, 5.3), strumento che permette la valutazione della soddisfazione degli utenti tramite 5 metodi differenti ognuno valutato su scala da 1 a 9 con possibilità di non rispondere:
 - *Overall User Reactions*: con valori come terribile/meraviglioso, frustrante/-soddisfacente, noioso/stimolante, etc.
 - *Screen*: valuta l'aspetto dello schermo mostrato all'utente;
 - *Terminology and System Information*: valuta la precisione della terminologia utilizzata;
 - *Learning*: valuta la semplicità di apprendimento del software per chi lo utilizza;
 - *System Capabilities*: valuta l'efficienza del sistema;

Altro esempio di questionario molto diffuso è il SUMI (Software Usability Measurement Inventory) (fig. 5.4), utilizzato per misurare la percezione dell'utente riguardo l'usabilità del software. Si raccomanda un numero di utenti compreso tra 10 e 12. Il questionario è formato da 50 item, assegnati a cinque scale diverse, e 25 item con scala globale, questa permette di paragonare i vari progetti. La tipologia di domande proposta presenta il formato "d'accordo", "in disaccordo". Altre scale utilizzate sono [29]:

- *Efficiency*: valuta l'efficienza del software nello svolgimento dei casi d'uso da parte dell'utente;
- *Affect*: misura la risposta emotiva dell'utente;
- *Control*: misura il grado con cui l'utente si sente controllato dal sistema durante l'esecuzione del compito.
- *Lernability*: misura la facilità di apprendimento del software da parte dell'utente, ovvero il tempo che un utente impiega a capire come funziona.

3. How satisfied or dissatisfied are you with the following

	Very satisfied	Satisfied	Neither satisfied nor dissatisfied	Dissatisfied	Very dissatisfied
The interaction with the sales staff	☐	☐	☐	☐	☐
Your experience at the register	☐	☐	☐	☐	☐
The organization of the store	☐	☐	☐	☐	☐
The wait for the fitting room	☐	☐	☐	☐	☐
The products offered in the store	☐	☐	☐	☐	☐
The price of the products	☐	☐	☐	☐	☐
The sizes available at the store	☐	☐	☐	☐	☐

Figura 5.2: Esempio di questionario QUIS

Screen		1	2	3	4	5	6	7	8	9		NA
Characters on the computer screen	hard to read				x						easy to read	
Highlighting on the screen	unhelpful			x							helpful	
Screen layouts were helpful	never					x					always	
Sequence of screens	confusing									x	clear	
Terminology and System Information		1	2	3	4	5	6	7	8	9		NA
Use of terminology throughout system	inconsistent			x							consistent	
Terminology relates well to the work you are doing?	always					x					never	
Position of messages on screen	inconsistent				x						consistent	
Messages which appear on screen	confusing				x						clear	
Computer keeps you informed about what it is doing	never							x			always	
Error messages	unhelpful									x	helpful	
Learning		1	2	3	4	5	6	7	8	9		NA

Figura 5.3: Esempio di questionario QUIS

Usability Testing

Gli usability testing sono la combinazione dei due approcci con l'aggiunta di elementi empirici e sperimentali per la misurazione e il confronto dei risultati. A differenza delle altre tecniche queste sono utilizzate nello specifico per l'usabilità del sistema. Possiamo distinguere sette tecniche principali:

- *Cognitive walkthroughs*: svolta da uno specialista al fine di identificare problematiche di usabilità;
- *Prototype evaluation*: svolta da utenti finali che durante un meeting spiegheranno agli esperti le proprie aspettative riguardo il prototipo o il mockup del software.
- *Checklist verification*: In questa fase gli specialisti verificano che i requisiti per gli standard di design sono verificati, l'analisi viene condotta principalmente nell'interfaccia grafica.
- *Pencil & Paper*: valutazione da parte degli utenti sugli aspetti grafici visibili sul prototipo.

The Post-Study Usability Questionnaire Version 3		Strongly agree							Strongly disagree	NA
		1	2	3	4	5	6	7		
1	Overall, I am satisfied with how easy it is to use this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	It was simple to use this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
3	I was able to complete the tasks and scenarios quickly using this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
4	I felt comfortable using this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
5	It was easy to learn to use this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
6	I believe I could become productive quickly using this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
7	The system gave error messages that clearly told me how to fix problems.	☐	☐	☐	☐	☐	☐	☐	☐	☐
8	Whenever I made a mistake using the system, I could recover easily and quickly.	☐	☐	☐	☐	☐	☐	☐	☐	☐
9	The information (such as online help, on-screen messages and other documentation) provided with this system was clear.	☐	☐	☐	☐	☐	☐	☐	☐	☐
10	It was easy to find the information I needed.	☐	☐	☐	☐	☐	☐	☐	☐	☐
11	The information was effective in helping me complete the tasks and scenarios.	☐	☐	☐	☐	☐	☐	☐	☐	☐
12	The organization of information on the system screens was clear.	☐	☐	☐	☐	☐	☐	☐	☐	☐
13	The interface* of this system was pleasant.	☐	☐	☐	☐	☐	☐	☐	☐	☐
14	I liked using the interface of this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐
15	This system has all the functions and capabilities I expect it to have.	☐	☐	☐	☐	☐	☐	☐	☐	☐
16	Overall, I am satisfied with this system.	☐	☐	☐	☐	☐	☐	☐	☐	☐

*The "interface" includes those items that you use to interact with the system. For example, some components of the interface are the keyboard, the mouse, the microphone, and the screens (including their graphics and language).

Figura 5.4: Esempio di questionario SUMI

- *Perspective based usability inspection:* La procedura consiste nell'analizzare le problematiche utilizzando una procedura specifica con una serie di domande che rappresentano i problemi di usabilità da verificare.
- *Eye Tracking:* In questa procedura viene misurato il movimento degli occhi dell'utente durante l'uso del software, tramite questi analisi è possibile determinare se alcune sezioni vengono ignorate.
- *Click-Scroll-Heat Maps:* Permette di identificare le sezioni in cui gli utenti hanno passato la maggior parte del tempo. Attraverso l'uso di particolari software viene creata una mappa di click, con colori diversi a seconda del tempo passato dall'utente in quell'area.

Capitolo 6

Test pagine web

6.1 Introduzione

L'uso delle applicazioni web sta prendendo sempre più piede nella vita di tutti i giorni, vista la grande importanza che il web ha assunto negli ultimi anni [30]. Tra i vari vantaggi possiamo citare:

- Costi di installazione
- Aggiornamento automatico: non è necessario mandare aggiornamenti ai client, l'aggiornamento avviene direttamente dal server.
- Multi-piattaforma: è possibile farvi accesso a prescindere dal sistema operativo utilizzato

Tuttavia, l'aspetto negativo è dovuto all'inaffidabilità, causata dagli errori cui le tecnologie server e browser sono spesso soggette. Recenti studi hanno dimostrato che 29 su 40 dei principali siti e-commerce hanno fallito i requisiti funzionali, durante i test con gli utenti [31, 32].

Al fine di ottenere una buona riuscita nello sviluppo di un software, i test GUI dovrebbero occupare circa il 30-40% del tempo totale di sviluppo [33]. Ma spesso gli errori non dipendono dalla GUI, ma piuttosto dalla logica di business sottostante. Per sopperire a tali errori i ricercatori propongono nuove tecniche per testare le applicazioni basate sul web.

6.2 Strumenti di Test

Sul mercato esistono diversi tool che permettono i test su pagine web, tuttavia, gran parte di questi si concentrano conformità del protocollo, sui test di carico, sul controllo dei collegamenti e su varie analisi statiche, sostanzialmente non verificano

direttamente che i requisiti funzionali siano rispettati [34].

Oggi giorno le pagine Web vengono generate dinamicamente, e gli strumenti disponibili non sono capaci di analizzarle, ma permettono la registrazione delle sequenze di input da utilizzare sui test, i cosiddetti *white-box*, il che facilita la costruzione di suite di test "adeguate", ma risultano comunque dispendiosi dal punto di vista della manovalanza per la generazione di casi di test che soddisfino i requisiti. I team di test devono scegliere attentamente quale tool utilizzare, perché questo potrebbe garantire il successo o meno del progetto.

6.2.1 Problemi

I principali problemi che si incontrano nel momento in cui si testano le applicazioni web possono essere diversi:

- Architetture client/server: il che significa che sono presenti chiamate HTTP asincrone, a cui ogni richiesta viene associata una risposta;
- Molteplicità dei linguaggi di programmazione: le applicazioni sono spesso eterogenee, lato client i linguaggi più comuni sono JavaScript, HTML, CSS, e su lato server e Java, PHP, etc..
- Struttura dinamica: il che determina risultati non deterministici;

6.3 Tecniche di test del software

I test del software permettono di misurare la qualità del processo, verificandone la presenza di errori [35]. Le tecniche più diffuse del test sono raffigurate nella figura 6.2

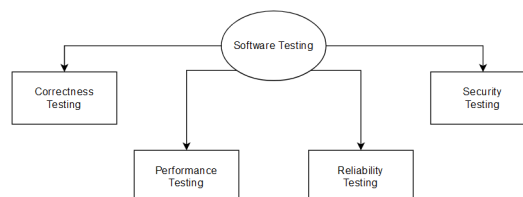


Figura 6.1: Tecniche di test

6.3.1 Test di correttezza

Lo scopo è quello di verificare la correttezza del software, verificandone i requisiti .

6.3.2 White box testing

Chiamato anche test strutturale, permette la verifica di uno o più componenti per poter trovare errori nel sistema software. Il funzionamento si basa su determinare un percorso da seguire al fine di testare una parte del programma. Non esistono criteri che permettono di testare tutti i "cammini" di un certo programma, dal momento che questi potrebbero essere infiniti, per la presenza di cicli. Ma è possibile trovare un numero finito di cammini indipendenti che combinati che permettano il raggiungimento di tutti, o quasi, i restanti. L'analisi in questa metodologia consta nell'avere a disposizione il codice che permette di identificare i cammini [36].

6.3.3 Black box testing

L'obiettivo di questo tipo di test consiste nel testare le funzionalità sulla base dei requisiti, ma senza fare riferimento alla struttura interna, a differenza del White Box, infatti, non è necessario valutare il codice interno. I test principali riguardano i possibili input e output.

6.3.4 Grey box testing

Questa metodologia viene utilizzata per testare una parte del software rispetto alle sue specifiche, ma essendo a conoscenza del suo funzionamento interno. Questo tipo di test può spesso includere il reverse engineering [37, 38].

6.4 Test Statici e Test Dinamici

La presenza di bug all'interno di un software deve essere verificata prima e durante la sua esecuzione. Per effettuare questa verifica i principali approcci utilizzabili sono test *statici* e test *dinamici*. Entrambi i metodi possono essere utilizzati per testare il software, a seconda delle disponibilità economiche, e della precisione richiesta.

6.4.1 Test statici

Questo tipo di test vengono eseguiti prima dell'esecuzione del software. Tale approccio permette di rilevare principalmente i bug presenti sul codice. Il sito può essere analizzato per determinare la presenza di pagine fantasma, ovvero pagine

che esistono, ma che non è possibile raggiungere a partire dalla pagina iniziale [39]. Gli approcci statici possono risultare per certi aspetti limitanti nel momento in cui si vanno a testare applicazioni web dinamiche in cui il codice viene generato dinamicamente, ma allo stesso tempo questi permettono di individuare errori prima della compilazione, rendendo la correzione più economica e veloce

6.4.2 Test dinamici

A differenza dei test statici quelli dinamici richiedono l'esecuzione del programma. Questo tipo di approccio verifica che a determinati input (test case) il sistema fornisca l'output desiderato, analizzando una parte del software alla volta, al fine di rilevare bug ed errori.

6.5 Metodologie di test

I metodi di test tradizionali, elencati sopra, vengono utilizzati da sei principali metodologie di test delle applicazioni web (fig. 6.2) ciascun dei quali contiene i modelli, i metodi, i criteri e gli obiettivi del test. I modelli descrivono strutture e metodi dell'applicazione, mentre i metodi riguardano il meccanismo di prova (white box, black box). Gli obiettivi invece descrivono requisiti funzionali e non funzionali, del software sul quale effettuare i test. Tuttavia, il test delle applicazioni web è più

	Models	Testing Methods		Testing Criteria		Testing Targets		References
		White Box	Black Box	Control flow	Data flow	Functional	Non-functional	
Formal Method	TestUml	x		x		x		[5,6]
	State-based		x	x		x		[21,22]
Object-oriented	Object-Based	x			x	x		[18, 19]
	Agent-Based	x	x		x	x		[24]
	OO Web Test	x			x	x		[16]
Statistical	Markov Chain	x	x				x	[7, 8, 13, 14, 37]
UML	ReWeb/ TestWeb	x		x		x		[27, 28, 31, 33]
Slicing	WebApp slicing	x			x	x		[29, 32]
User Session	User Session Data	x	x	x		x		[9, 10, 34, 35]

Figura 6.2: Tecniche di test

impegnativo rispetto ai test dei software tradizionali, il che è dovuto al non poter controllare il sistema web e dalla complessità di internet.

6.6 Casi di test

Quando si progettano i casi di test non tutte le pagine sono ugualmente interessanti, ad esempio, le pagine che contengono dati statici, che non richiedono l'inserimento o l'interazione da parte dell'utente possono essere ignorate. In questo modo il test si riduce alle sole entità rilevanti. Si richiede la definizione di tutti i percorsi indipendenti o con loop limitati. Una volta generati i casi di test, si inseriscono i valori di input per le variabili, dopo di che l'esecuzione può essere automatizzata, ed è possibile ispezionare i risultati per valutare se il caso di test è stato superato o meno. I test di regressione traggono molto vantaggio dall'automazione poiché il caso di test può essere eseguito per confrontare i vari output generati.

6.6.1 Test automatizzati

Uno dei metodi di sviluppo più utilizzati negli ultimi anni è Agile, con i quali i team devono rilasciare i software durante gli sprint (ogni 1-4 settimane), il che richiede l'uso di test di automazione per permettere a chi si occupa dei test di stare al passo con i team di sviluppo, evitando così test manuali lunghi spesso causa di errori [40]. Nel momento in cui si vanno ad utilizzare tool di test automatizzati, il team che se ne occupa deve sapere cosa andare ad automatizzare, deve avere un piano e linee guida chiare per il lavoro da svolgere. La maggior parte dei test automatizzati presenti nel mercato forniscono una infrastruttura che registra particolari sessioni dell'utente e le trasforma in una serie di script che possono essere riprodotti in seguito per l'esecuzione di test funzionali e di regressione. Altri strumenti, invece, generano casi di test combinando un qualche algoritmo di esplorazione del percorso del sito web, con input forniti dall'utente, che ne effettua il test. Di recente sono state proposte altre tecniche, queste utilizzano forme di test model based e UML classificabili come "white-box", sostanzialmente queste tecniche estendono la tradizionale generazione di test basati su percorsi. L'approccio Ricca e Tonella genera un modello basato sui nodi, i quali rappresentano gli oggetti all'interno della pagina web e le relazioni tra gli oggetti sono rappresentati dagli spigoli. In figura 6.3, vediamo un modello di un componente relativo all'acquisto di libri online [41].

Il grafico permette di generare i requisiti e i casi di test e il tester fornisce i valori di input in modo che tali sequenze possono essere eseguite. In questo approccio il test case consiste in una sequenza di pagine web da visitare insieme alle corrispondenti coppie nome-valore. Un fattore limitante nell'uso delle tecniche di test come questa è il costo nella ricerca di input, la cui selezione è lenta e deve essere eseguita manualmente [39].

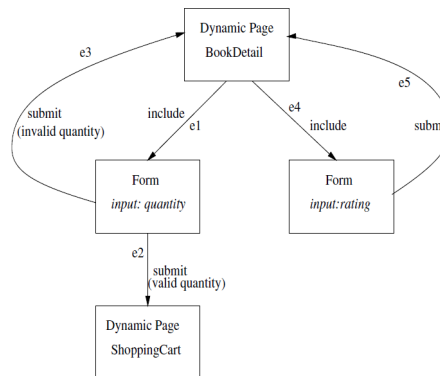


Figura 6.3: Modello UML applicazione e-commerce

6.6.2 Caratteristiche test automatizzati

Elenchiamo di seguito le caratteristiche e funzionalità fondamentali che un test automatizzato dovrebbe avere [42]:

- Registrazione e riproduzione affidabili;
- Esportazione di script programmabili;
- Supporto compatibilità tra browser;
- Capacità nel trovare gli elementi all'interno della pagina.
- Efficaci nell'attendere il caricamento completo delle pagine web, prima di accedervi e manipolare gli elementi.

6.7 Selenium

Uno dei più famosi tool open source che permette l'esecuzione dei test automatizzati è Selenium, nato nel 2004 da Huggins e da Hammant, che si aggiunse al team successivamente.

Selenium fornisce uno strumento di registrazione dei test, durante questa fase è possibile per il tester selezionare gli elementi all'interno della pagina attraverso un menu contestuale, questo semplifica la comprensione degli script [42].

6.7.1 Componenti

Selenio è composto da diversi componenti, ognuno dei quali con uno specifico compito nello sviluppo dell'automazione delle applicazioni web:

- Selenium IDE
- Selenium API Client
- Selenium Remote Control
- Selenium Web Driver
- Selenium Remote Control
- Selenium Web Driver
- Selenium Grid

Selenium IDE

Selenium IDE è un plugin di firefox ed estensione di chrome che attraverso un'interfaccia permette la registrazione, la modifica e il debug dei test. Le azioni dell'utente registrate durante la fase di test possono essere esportate come script, riutilizzabili sia per una successiva fase di test, sia per essere riutilizzate in altri linguaggi di programmazione [43].

Selenium API client

Mediante delle API di selenium Client è possibile scrivere i test nei linguaggi di programmazione più comuni come Java, C#, Javascript e Python.

Selenium Remote Control

Selenium Remote Control (RC) permette la scrittura dei test automatizzati in qualsiasi linguaggio di programmazione, fornisce driver client per PHP, Python, Ruby, .NET, Perl e Java. Selenium RC è un server scritto in java ed accetta chiamate HTTP.

Selenium WebDriver

Selenium WebDriver è il successore di Selenium RC, fornisce un'interfaccia che permette il supporto dei test automatizzati per tutti i tipi di browser. Il driver gestisce le comunicazioni da e verso Selenium e Browser. Per guidare il browser, e non avere problemi dovuti alle differenze di comandi che possono esservi tra javascript, e i comandi nativi del sistema operativo, in alcuni casi è possibile utilizzare direttamente le funzionalità a livello di sistema operativo, anziché utilizzare i comandi Javascript basati sul browser.

Selenium Grid

Selenium Grid permette l'esecuzione su macchine remote, il suo scopo è quello di hub centrale: sostanzialmente i test contattano selenium Grid al fine di ottenere l'accesso al browser, e l'hub avendo un elenco di server che ne forniscono l'accesso consente ai test di utilizzare le istanze necessarie. Vantaggio di selenium Grid è quello di permettere l'esecuzione parallela su più macchine gestendo così le diverse versioni e configurazioni del browser.

6.7.2 Installazione e utilizzo di selenium

L'installazione di Selenium è suddivisa in tre step:

1. Installazione della Libreria Selenium per il linguaggio di programmazione desiderato;
2. Configurazione del driver per automatizzare il browser;
3. Impostazione e configurazione di Selenium Grid per aumentare la dimensione dei test (facoltativo)

Selenium è uno strumento molto utile, ma è necessario capire quando è il caso di utilizzarlo. È sconsigliato nel caso di progetti di brevissima durata, o in caso in cui l'interfaccia del progetto vari spesso.

Capitolo 7

Progettazione Front End

7.1 Descrizione generale

Il software DME è un'applicazione web il cui scopo è quello di permettere l'organizzazione di Eventi, dai matrimoni, alle feste di Laurea, compleanni, eventi sportivi, Feste private etc..

7.1.1 Tipologie di utenti

L'applicazione permette la registrazione a tre tipologie di utenti:

- **Organizzatori:** utenti cui sarà permesso l'accesso a tutte le funzionalità relative all'organizzazione di eventi.
- **Fornitori:** Specifico per le aziende interessate nel mercato degli Eventi, ad essi sarà permesso di inserire i propri servizi all'interno della piattaforma, per poter essere scelti dagli organizzatori nella realizzazione del loro evento.
- **Invitati:** Gli invitati, mediante un codice di invito potranno accedere alla piattaforma e inserire foto e video dell'evento. Inoltre, potranno vedere menu, location, e altri dettagli dell'evento cui sono stati invitati.

7.2 Specifica dei Requisiti

I requisiti sono principalmente di due tipi: funzionali e non funzionali. I requisiti funzionali descrivono i servizi e/o comportamenti forniti dal sistema mentre quelli non funzionali forniscono i vincoli.

7.2.1 Requisiti non funzionali

Essi non riguardano direttamente le specifiche funzioni fornite dal sistema, ma possono, sia riferirsi a caratteristiche che si desidera il sistema presenti, sia definire i vincoli ai quali il sistema deve sottostare. I requisiti non funzionali non riguardano solo il sistema software che si sta sviluppando ma, alcuni possono vincolare anche il processo usato per sviluppare il sistema. Spesso i requisiti non funzionali sono più critici di quelli funzionali, poiché se questi non sono soddisfatti la piattaforma risulta inutilizzabile. I requisiti non funzionali della piattaforma DME prevedono la *Portabilità* del sito, ovvero la possibilità di utilizzare la piattaforma su qualsiasi dispositivo e browser.

7.2.2 Requisiti funzionali

I requisiti funzionali definiscono una funzione di sistema di uno o più componenti. Per la piattaforma DME i requisiti funzionali sono:

- Gestione tipologia utente
- Registrazione e memorizzazione dati cliente
- Registrazione e memorizzazione dati fornitore
- Memorizzazione di un servizio, la sua tipologia e la relativa vetrina da parte di un fornitore
- Creazione e memorizzazione di un evento da parte di un cliente
- Inserimento e memorizzazione nominativi invitati del cliente
- Gestione e assegnazione tavoli invitati del cliente
- Memorizzazione e gestione dello smart contract
- Memorizzazione e gestione pagamento servizio.

7.3 Strumenti e linguaggi utilizzati

Il Front End è stato realizzato utilizzando il framework Angular, con HTML5 e Css3. La realizzazione del sito responsive, come da requisiti non funzionali è stata implementata con alcuni componenti di angular material, con l'uso delle Media Query e con bootstrap. Il nuovo design grafico della piattaforma è stato realizzato prima mediante l'uso di Adobe XD e Adobe illustrator, e una volta approvato è stato inserito all'interno dell'applicazione web realizzando il relativo

codice HTML e CSS. Per la realizzazione dei mockup e siti responsive, non è stato utilizzato l'approccio "small first", ma sono stati realizzati prima le UI per desktop, poi adattate a mobile e tablet, per averne una visione più completa.

7.4 Schema Sito DME

La piattaforma DME comprende attualmente due protagonisti principali: fornitori e organizzatori. Effettuando una registrazione e successivo login come Organizzatore sarà possibile accedere alle seguenti pagine (fig.7.1):

- Event-tool
- Private-page
- Calendar
- Orders
- Services
- Services-details
- Cost-estimation
- Create-event
- Update-event
- Profile
- Contracts
- Table-managements
- Guests
- Blog

Effettuando registrazione, e Loggandosi come Fornitore invece è possibile accedere alle pagine (fig. 7.2):

- Private-page-f
- Orders
- Calendar

- Create-service
- Update-service
- Profile
- Calendar
- Services
- Service-details

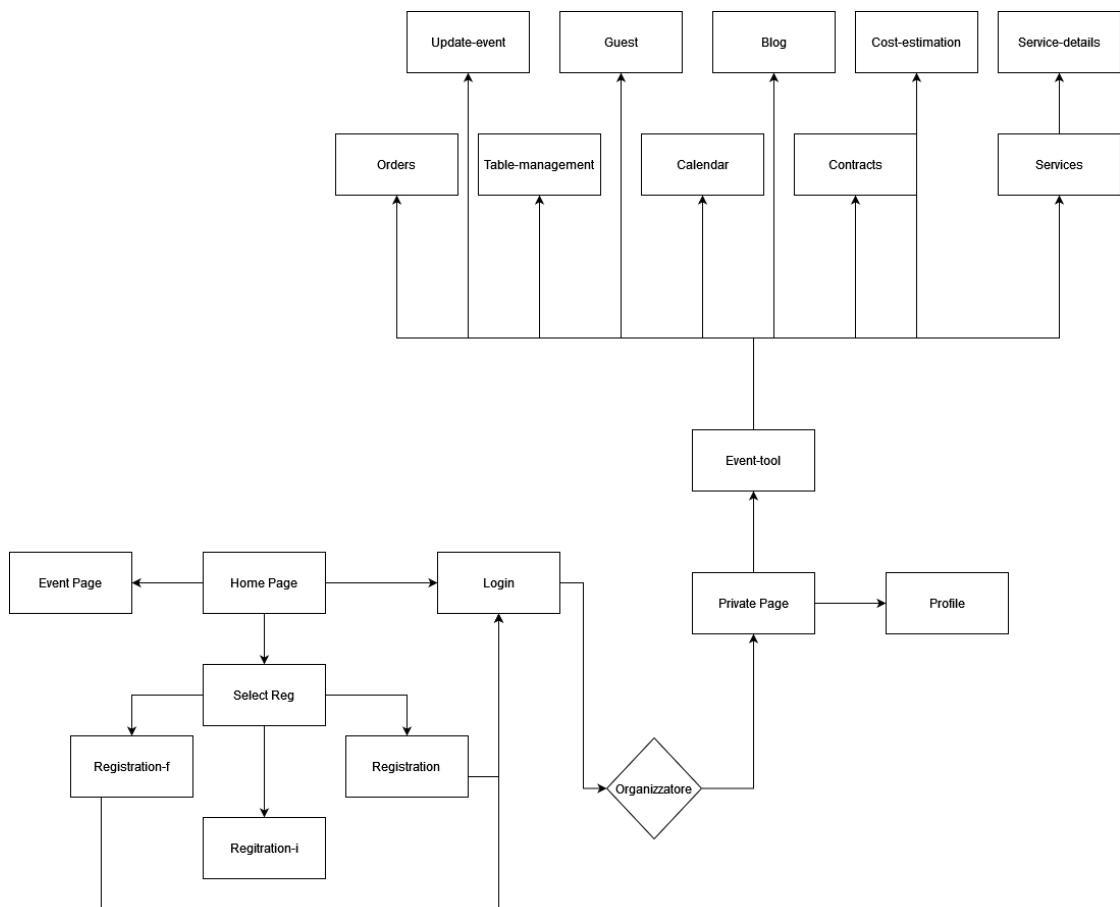


Figura 7.1: Architettura Organizzatore

7.4.1 Organizzatore

L'organizzatore ha la possibilità di creare un evento attraverso il form sulla pagina `create-event`. Il form si divide in due sottosezioni: scelta del tipo di evento (fig.

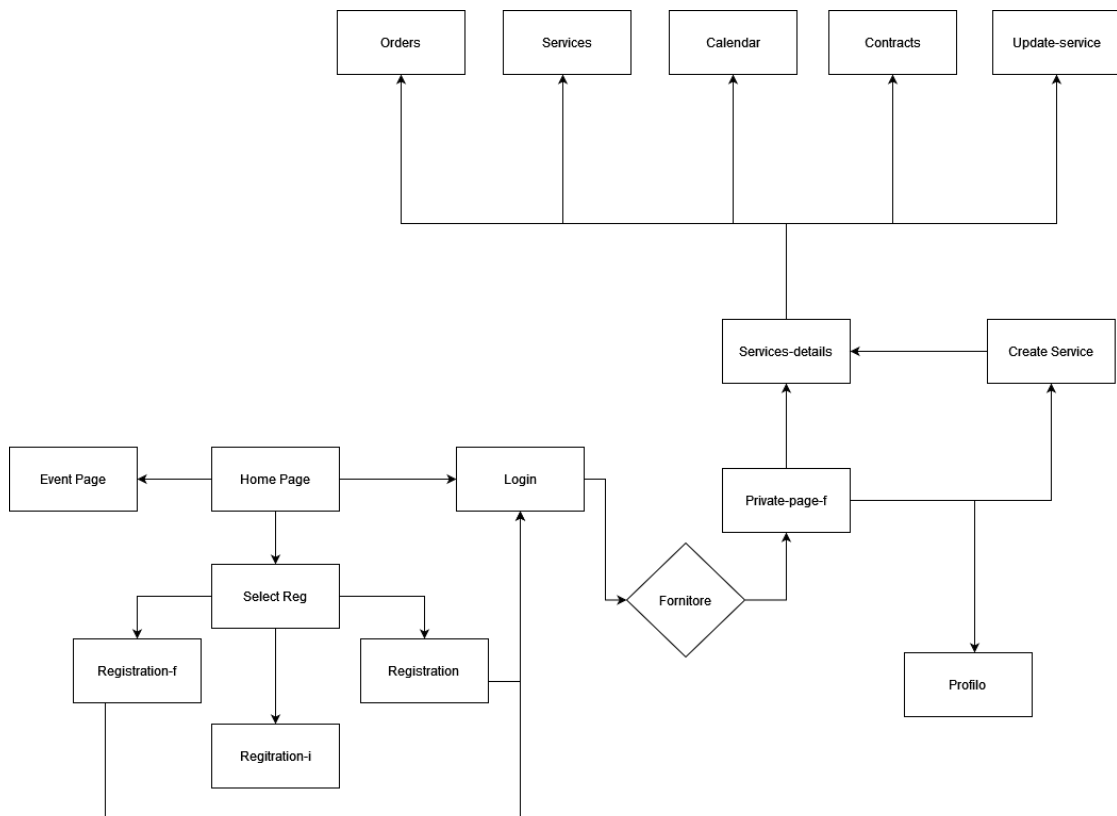


Figura 7.2: Architettura Fornitore

7.3a), e inserimento dei dati relativi all'evento (fig. 7.3b). I dati richiesti all'organizzatore sono: nome evento, data, luogo in cui si vorrebbe si svolgesse l'evento (provincia e comune), e un'e-mail di contatto, inoltre verrà chiesto all'organizzatore di scegliere almeno un servizio relativo all'organizzatore dell'evento all'interno della piattaforma, come ad esempio catering, location, parrucchiere, o altro. La scelta è obbligatoria, perché consentirà la visualizzazione di altre funzionalità.

Dopo aver completato il form relativo alla creazione dell'evento, l'utente viene reindirizzato alla pagina **private-page** in cui è presente un cont-down dell'evento che si vuole realizzare; premendo sul bottone "gestisci" (fig.7.4a) verrà reindirizzato sulla pagina **event-tool** (fig.7.4b), in cui sono presenti diversi bottoni, ognuno dei quali rimanda gli organizzatori a funzionalità diverse della pagina.

Servizi fornitori

Selezionando il bottone "fornitori" saranno visibili tutti i fornitori che offrono i servizi selezionati durante la creazione dell'evento. La pagina permette all'organizzatore di filtrare i servizi per categoria, prezzo e recensioni di altri clienti. Selezionando sul

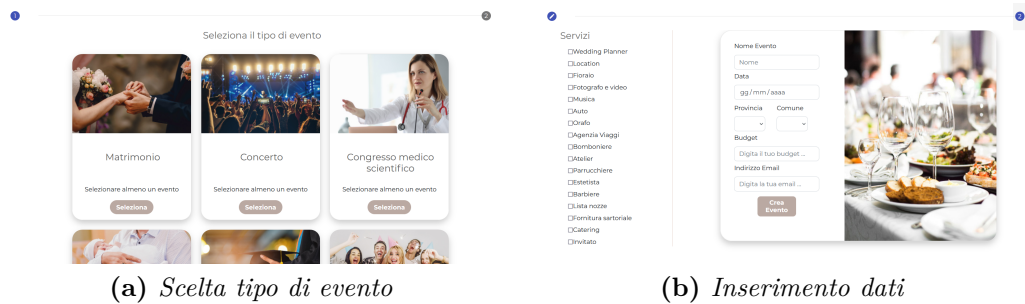


Figura 7.3: Creazione evento

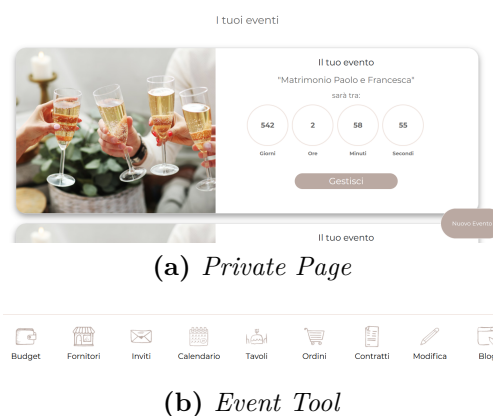


Figura 7.4: Gestione evento

bottone "vedi dettagli" (fig. 7.5a) si entra nella pagina relativa al servizio (fig. 7.5), in cui è possibile visionare prezzi e descrizione del prodotto, è poi possibile prendere un appuntamento con il fornitore (fig. 7.5c), e ordinare il prodotto desiderato (fig. 7.5d).

Ordini

Nella sezione ordini è possibile vedere gli ordini effettuati, e se confermati dal fornitore è possibile effettuare il pagamento (fig. 7.6).

Calendario

Nella sezione calendario, sono visibili tutti gli appuntamenti presi da un organizzatore, questi sono divisi in sospeso e confermati. Gli appuntamenti possono essere modificati ed eliminati (fig. 7.7).



Figura 7.5: Dettagli vetrina

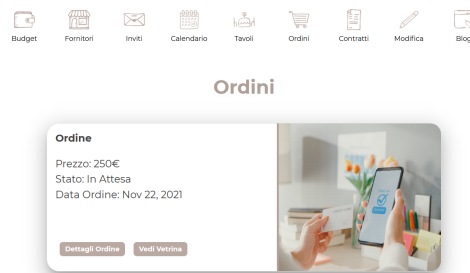


Figura 7.6: Gestione ordini

Invitati e gestione Tavoli

Nella sezione invitati, potranno essere aggiunti tutti gli invitati all'evento, è possibile inserirli in una categoria specifica, che può essere relativa ad un invitato "attivo", quindi, un servizio come i fotografi, o relativa ad un invitato "passivo" che può essere un amico o un parente. L'invitato una volta inserito nella lista può essere confermato, eliminato o modificato.

Una volta selezionati gli invitati, per le relative categorie, e confermati, è possibile creare dei tavoli e assegnargli un nome (fig.7.8).

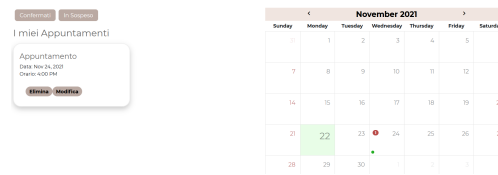


Figura 7.7: Calendario e gestione appuntamenti

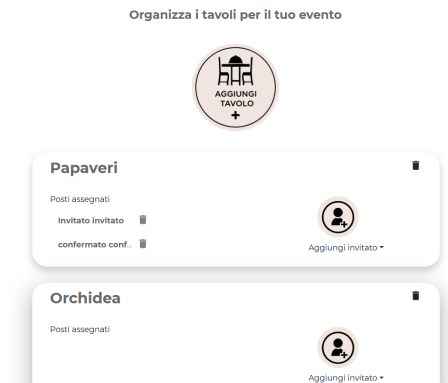


Figura 7.8: Gestione tavoli

Blog

La pagina del blog permette di selezionare una grafica e di inserire informazioni riguardanti l'evento in una pagina dedicata all'evento, in cui gli invitati potranno accedere aggiungere foto e video, e per commentare l'evento (fig. 7.9).

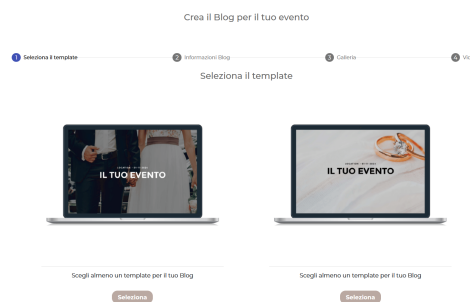


Figura 7.9: Creazione blog

7.4.2 Fornitore

Le pagine del fornitore sono molto simili a quelle già viste per l'organizzatore. L'azione principale del fornitore è la creazione di un servizio, in cui può inserire uno o più prodotti da poter proporre agli organizzatori di eventi. Il fornitore del servizio avrà la possibilità di confermare, modificare o rifiutare le varie richieste da parte dell'organizzatore (fig. 7.10).



Figura 7.10: Creazione Servizi

7.4.3 UI

La realizzazione della grafica si basa su uno stile molto semplice, con colori sul tono del beige, che possano adattarsi a tutte le tipologie di evento. Ogni pagina riporta uno stile comune, e sono sempre presenti delle card bianche con bordo sfumato, contenenti immagini rappresentative e testo. Per mantenere sempre uno stile delicato si è utilizzato il font "Montserrat" utilizzato all'interno di tutto il sito.

Capitolo 8

Risultati

8.1 Alpha Test

Parallelamente alla realizzazione dell'interfaccia UI sono state svolte varie sessioni di alpha test. Queste hanno permesso la rilevazione di bug e funzionalità incomplete nella piattaforma DME.

Sono stati analizzati 24 casi d'uso per un totale di 154 test per sessione (tabella: 8.1).

Ad ogni Use Case sono associati più casi di test, relativi alla compilazione dei campi dei form, e delle funzionalità nel caso vadano o meno a buon fine. Ogni test è caratterizzato da:

- ID progressivo (TC 1, TC 2, ..)
- Titolo;
- Descrizione del tipo di test;
- Precondizione;
- Risultati attesi;
- Step necessari per l'esecuzione del test;
- Note relative ai bug riscontrati durante l'esecuzione.

8.1.1 Risultati Alpha Test

Dalle sessioni di test è emerso che diverse pagine della sezione cliente e della sezione fornitore erano ancora incomplete in particolare:

- Agenda Fornitori

Use Case	Title
Creazione account organizzatore	UC 1
Creazione account fornitore	UC 2
Creazione account Invitato	UC 3
Login	UC 4
Creazione evento privato	UC 5
Gestione attività stima e costo	UC 6
Pagina Fornitori	UC 7
Pagina Inviti	UC 8
Pagina Calendario	UC 9
Pagina Tavoli	UC 10
Pagina Ordini	UC 11
Pagina Contratti	UC 12
Pagina Modifica Evento	UC 13
Pagina Blog	UC 14
Profilo	UC 15
Private-page-f	UC 16
Pagina Crea-servizio	UC 17
Gestione servizio	UC 18
Pagina Calendario (Fornitore)	UC 19
Pagina Ordini (Fornitore)	UC 20
Pagina Contratti (Fornitore)	UC 21
Modifica -Servizio (Fornitore)	UC 22
Profilo Fornitore	UC 23
Servizi	UC 24

Tabella 8.1: Alpha Test

- Tavoli
- Inviti
- Agenda Fornitori
- Ordini Fornitori
- Servizi Fornitori
- Creazione servizio

Sono stati verificati significativi errori e mancanza di controlli nei form di registrazione e, a seguito di questo, si è deciso di dividere il form della registrazione dei fornitori in più parti (fig. 8.1):

- Inserimento delle credenziali e dati di contatto;
- Inserimento dati aziendali;
- Informazioni sulle modalità di pagamento.

The figure displays three sequential screenshots of a web form titled 'Fornitori' (Suppliers) with the subtitle 'Inserisci i tuoi dati' (Enter your data) and a link 'Se già registrato? Login'.

Screenshot 1: Step 1 - Credenziali
 This step includes a progress bar with three items: 'Credenziali' (selected), 'Dati Azienda', and 'Modalità pagamento'. The form fields are:
 - Username *
 - Password *
 - Email *
 - Conferma Password *
 - Informazioni di Contatto:
 - Nome *
 - Cognome *
 - Data di nascita * (format: gg/mm/aaaa)
 - Sesso * (dropdown)
 - Provincia * (dropdown)
 - Luogo di nascita * (dropdown)
 - Codice Fiscale *
 - Professione *
 - IT +39 * (dropdown)
 - Cellulare *
 - IT +39 * (dropdown)
 - Telefono *
 - Buttons: 'Indietro' and 'Avanti'

Screenshot 2: Step 2 - Dati Azienda
 This step includes a progress bar with three items: 'Credenziali', 'Dati Azienda' (selected), and 'Modalità pagamento'. The form fields are:
 - Tipo di Servizio *
 - Ragione Sociale *
 - Partita IVA *
 - Indirizzo Azienda:
 - Regione * (dropdown)
 - Città *
 - Provincia * (dropdown)
 - CAP *
 - Indirizzo *
 - PEC *
 - Sito Web *
 - Nome dell'Azienda *
 - Campo: 'Descrivi la tua azienda' (text area)
 - Buttons: 'Indietro' and 'Avanti'

Screenshot 3: Step 3 - Modalità pagamento
 This step includes a progress bar with three items: 'Credenziali', 'Dati Azienda', and 'Modalità pagamento' (selected). The form fields are:
 - Selezione il tipo di Pagamento:
 - ☐ Bonifico
 - ☐ Carta di credito
 - ☐ Paypal
 - Selezione modalità pagamento:
 - ☐ Acconto
 - ☐ Rate
 - ☐ Saldo
 - Buttons: 'Indietro' and 'Invia'

Figura 8.1: Form Registrazione fornitore

Lo svolgimento dei test ha permesso oltre al completamento delle funzionalità mancanti anche l'aggiunta di maggiori controlli, alert e notifiche di successo o insuccesso dell'operazione.

8.2 Sperimentazione sul Campo con utenti finali

Per la sperimentazione della piattaforma DME sono state utilizzate tecniche Opinion-based, il cui scopo è stato quello di ottenere indicazioni sul gradimento da parte degli utenti, sia a livello di facilità d'uso che a livello di processo di business. La sperimentazione con gli utenti finali ha permesso una visione del progetto da un punto di vista differente.

Individuazione utenti finali e definizione dei case studies

I questionari sono stati proposti ad utenti privati che assumono nel test il ruolo di "organizzatori". All'interno della piattaforma sono stati inseriti dei fornitori, utili a simulare l'organizzazione di eventi. Per fare questo, sono state contattate diverse aziende del mercato locale, esperte nel settore. Si ringraziano per la collaborazione:

- Masseria Don Luigi, Location
- Vincenzo Massaro Photographer
- Antonio Colonna, Orafo
- Giulia Molinari, Wedding Planner
- Barbara Nacci, Atelier
- Rent Classica, servizio Auto
- Blu Vacanze, Agenzia di Viaggi
- Incantalupo, Parrucchiere e centro estetico

8.2.1 Preparazione test con gli utenti

Il questionario proposto si compone di 6 sezioni:

- Svolgimento di Task;
- Informazioni personali;
- Valutazione della terminologia utilizzata;
- Valutazione della semplicità di apprendimento ;
- Valutazione visione del sito sui vari schermi;
- Valutazione generale del sito e suggerimenti.

Sono state proposte diverse tipologie di domande: domanda aperta, valutazione su scala 1-5, domande a risposta multipla. Inoltre, è stato chiesto agli utenti lo svolgimento di piccoli task con lo scopo di realizzare un Evento e, di utilizzare in modo guidato tutte le funzionalità del sito.

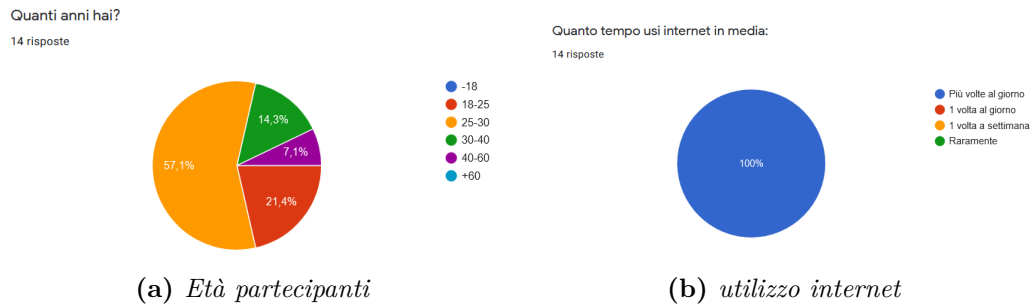


Figura 8.2: Informazioni partecipanti

8.2.2 Risultati Questionari

I questionari comprendevano in totale 40 domande sottoposte a 14 candidati di età compresa tra i 18 e i 60 anni (fig. 8.2a), tutti gli intervistati utilizzano internet più volte al giorno (fig. 8.2b). I task sono stati effettuati per il 57.1% su PC, e per il restante 42.9% su smartphone (fig. 8.3a). È stato poi chiesto di valutare la qualità dell'esperienza sia per utilizzo su smartphone che su desktop, esprimendo un punteggio da 1 a 5 (1 scarso, 5 buono).

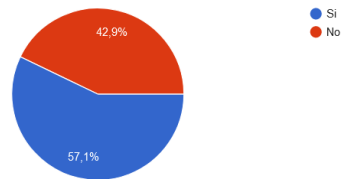
I risultati ottenuti indicano una sommariamente una buona esperienza, specialmente per l'uso su desktop (fig. 8.3b), mentre su smartphone 1 sola persona ritiene di aver avuto un'esperienza sufficiente e gli altri partecipanti si dividono in un'esperienza buona e discreta; questo potrebbe essere uno spunto per un'ulteriore analisi sul tipo di devices utilizzato per effettuare il test (fig. 8.3c). Per quanto riguarda la facilità d'uso della piattaforma DME circa l'80% dei partecipanti ritiene che la piattaforma sia facile/molto facile da utilizzare, mentre una sola persona ritiene che sia difficile (fig. 8.4a, 8.5b).

Andando ad analizzare le difficoltà riscontrate, alcuni partecipanti ritengono che diversi form siano poco chiari, in particolare indicano che non è visibile dove inserire i dati, o che non viene evidenziato in modo chiaro quali siano i dati mancanti. È stata indicata anche poca chiarezza nella scelta dei termini nella sezione ordini, riguardo alle modalità di pagamento.

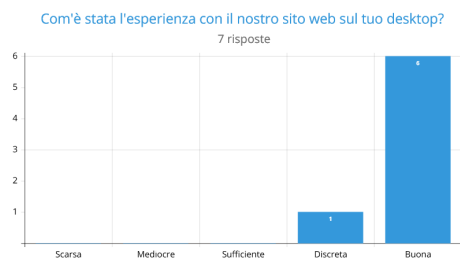
Nel complesso 10 utenti ritengono che il sito sia chiaro nei contenuti, e 4 utenti lo ritengono discreto (fig. 8.4c). Esaminando l'aspetto generale del sito web, il 64% degli utenti consiglierebbe il sito ad un amico o conoscente (fig. 8.5a).

La funzionalità più apprezzata all'interno del sito è la scelta dei fornitori, alcuni partecipanti ritengono che possa essere utile potersi basare sulle recensioni di altri clienti per la scelta dei servizi mentre altri trovano interessante la gestione del budget. È stato infine chiesto quale tipo di evento organizzerebbero nel sito (fig. 8.5c) e, tra le varie risposte, il 57.1% organizzerebbe una festa di laurea. Alla

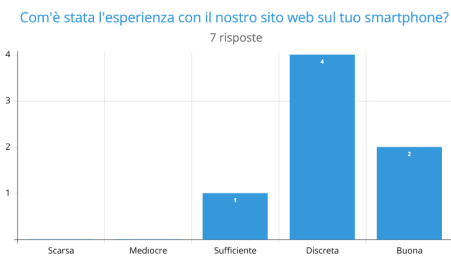
Hai utilizzato il sito web su PC
14 risposte



(a) Utilizzo Smartphone e Desktop



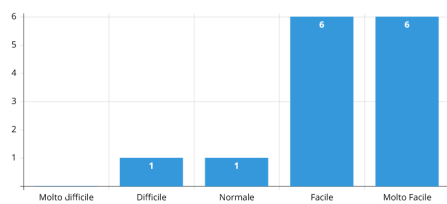
(b) Esperienza su desktop



(c) Esperienza su smartphone

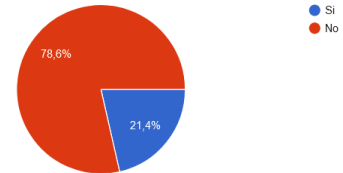
Figura 8.3: Utilizzo devices

L'uso delle pagine è stato semplice?
14 risposte

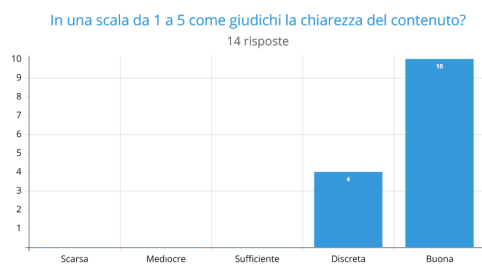


(a) Facilità utilizzo

Hai incontrato qualche difficoltà durante l'utilizzo del nostro sito web?
14 risposte



(b) Difficoltà riscontrate



(c) Chiarezza contenuto

Figura 8.4: Difficoltà e semplicità utilizzo

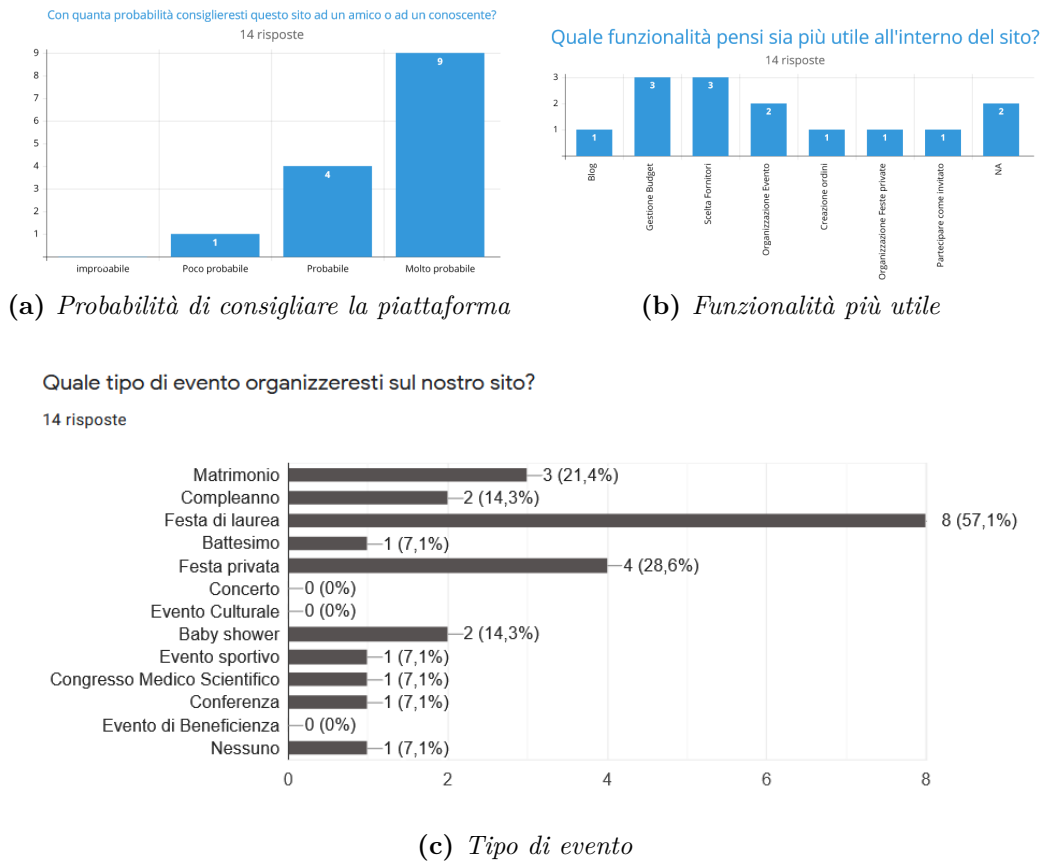


Figura 8.5: Considerazioni sulla piattaforma

domanda "Cosa ti piace del sito DME" alcune delle risposte date sono:

- Possibilità di organizzare feste private;
- Icone e immagini utilizzate;
- Grafica e scelta dei colori;
- Semplicità del sito.

8.3 Simulazione Caso reale

Lo scopo di questo tipo di test è quello di riproporre un evento passato utilizzando la piattaforma DME. Il paragone tra le due modalità di organizzazione ci permetterà di trarre vantaggi e svantaggi sull'utilizzo della piattaforma.

Per la simulazione è stata contattata la wedding planner Giulia Molinari, che ci ha

fornito come esempio il matrimonio di Francesco e Lucia:

"Gli sposi avevano già le idee chiare sulla scelta della location, desideravano un matrimonio sul mare. Una volta deciso il budget limite per l'organizzazione del matrimonio, hanno scelto una data approssimativa. Per potersi assicurare di festeggiare il loro matrimonio di sabato, hanno iniziato circa un anno prima la ricerca della location.

Hanno valutato in totale due strutture, subito dopo Lucia ha iniziato la ricerca del suo abito da sposa. Nei mesi successivi la ricerca dei vari fornitori come Musicista, fiorista, e fotografi non è stata semplice. Le scelte erano tante, e non sapevano come orientarsi nella scelta. Per questo motivo a circa 6 mesi dalla data prevista hanno deciso di affidarsi alla wedding planner Giulia Molinari. Giulia, esperta nel settore, ha saputo indirizzarli nella scelta dei dettagli mancanti. Gli ultimi tre mesi sono stati dedicati alla progettazione grafica e agli allestimenti."

8.3.1 Realizzazione di un matrimonio sulla piattaforma DME

Nel momento in cui si vuole realizzare un evento all'interno della piattaforma DME, il primo passo da compiere è registrarsi come organizzatore. Il form di registrazione richiede una serie di dati personali, quali nome, cognome, data di nascita, etc. e, una volta inseriti l'utente riceve una mail di conferma e può accedere al sito.

Gli organizzatori dopo aver effettuato il login vengono reindirizzati in una pagina che li invita ad organizzare il loro primo evento (fig. 8.6). I dati richiesti in questo caso sono:

- Nome Evento
- Data
- Luogo evento (Provincia, Comune)
- Budget
- Email di riferimento
- Scelta categorie servizi.

L'evento si chiamerà "Matrimonio sul Mare" verrà celebrato il 31/08/2022 in provincia di Bari, come budget si è scelto 25.000€. Le categorie servizi di cui usufruiremo per la realizzazione del matrimonio sono Location, Fotografo, Orafo, Wedding planner, Auto, Estetista e Parrucchiere, Atelier. Una volta creato l'evento



Figura 8.6: Creazione Nuovo Evento

nella sezione "fornitori" si ha un'ampia scelta di fornitori relativi alle categorie servizi selezionate, questi possono essere filtrati in base alle recensioni, ai prezzi e alla data. Ipotizziamo di scegliere, come nel caso dei nostri sposi, come primo servizio la location.

Selezioniamo come recensione 5 stelle, e come data il 31 agosto 2022; il numero di fornitori a disposizione si riduce permettendoci una scelta di qualità. Scegliamo come location la Masseria don Luigi, che offre un menu a 150€ a persona e una location sul mare. Dalla piattaforma DME prendiamo un appuntamento con il fornitore. Una volta scelta la location, definiamo il numero di invitati nella sezione "inviti" in questo modo è possibile fissare un accordo con il fornitore della location per la scelta del menu. Il numero di invitati sarà circa 100. A questo punto dopo l'ipotetico appuntamento con il fornitore dalla sezione ordine, nella vetrina della Masseria Don Luigi, possiamo prenotare il nostro menu per il 31 di agosto. Una volta ricevuta la conferma dal fornitore, nella sezione budget le stime dei costi vengono aggiornate in automatico. Scegliamo poi come Wedding planner Giulia Molinari, per un servizio di planning (300€), anche in questo caso prendiamo un appuntamento ed effettuiamo l'ordine all'interno della piattaforma. Il budget residuo a questo punto è di 9.700€. Scegliamo ora l'atelier, impostando filtro con prezzo inferiore ai 2.500€ scegliamo il fornitore Barbara Nacci che dopo aver preso un appuntamento sulla piattaforma ci propone diversi abiti. Scegliamo un abito della collezione "Flora", presente sulla piattaforma ed effettuiamo l'ordine. Scegliamo nello stesso atelier anche l'abito dello sposo, per un totale di 2.300€. Il budget restante per la scelta del fotografo, Orafo estetista e parrucchiere è di

7.400€. Con la funziona Tavoli, scegliamo la disposizione degli invitati. Nello stesso modo verranno scelti i servizi rimanenti (auto, estetista, orafo). Una volta che sono stati selezionati tutti i servizi, i preparativi per l'organizzazione del matrimonio sono completi.

8.3.2 Risultati test simulazione casi reali

Dalla simulazione dell'organizzazione del matrimonio mediante la piattaforma DME è possibile trarre diversi vantaggi: in primo luogo la possibilità di filtrare i fornitori in base alle proprie esigenze, questo permette sia di ridurre il grande numero di fornitori che possono esserci per un servizio, sia di garantire un servizio affidabile grazie alle recensioni di utilizzatori. Infatti, la coppia di sposi analizzata, prima di contattare la wedding planner ha avuto difficoltà nel scegliere i fornitori adatti. Un altro vantaggio del sito DME è che una volta completo, avrà un servizio di raccomandazione, che quindi suggerirà agli organizzatori, quando prenotare un servizio, piuttosto che un altro, e lo guiderà totalmente nell'organizzazione dell'evento. Aspetto molto interessante è il budget, perché permette di vedere subito come sono state distribuite le varie spese e come organizzarsi nella scelta dei costi.

Dal punto di vista dei fornitori i vantaggi possono essere:

- Maggiore visibilità da parte dei clienti
- Garanzia del servizio, data dalle recensioni
- Semplicità nella comunicazione con il cliente
- Pianificazione appuntamenti e ordini direttamente sulla piattaforma.

Dal punto di vista delle tempistiche di organizzazione, il vantaggio principale è dato dalla possibilità di accelerare i tempi di prenotazione appuntamenti e di poter visionare i prodotti mediante le foto all'interno della piattaforma.

Capitolo 9

Conclusioni

Il periodo di collaborazione con l'azienda Linear system è durato nel complesso 6 mesi da luglio 2021 a dicembre 2021.

Il lavoro di tesi è stato svolto in modalità ibrida, in presenza nell'azienda Linear system e mediante SAL settimanali che permettevano aggiornamenti sull'avanzamento della progettazione.

La fase preliminare della realizzazione della tesi si è concentrata sulla definizione di un nuovo design grafico per la piattaforma DME. Sono stati realizzati diversi mockup e prototipi, con l'aiuto di un esperto nel settore della grafica di siti web, utilizzando Adobe XD, e Adobe Illustrator. Una volta realizzati, e approvati sono stati inseriti all'interno dell'applicazione web utilizzando principalmente Html5 e Css3, inoltre per rendere la piattaforma responsive, requisito fondamentale del progetto, sono stati utilizzati media Query, alcune componenti di Angular Materials e di Bootstrap. La fase di test è stata svolta con l'aiuto di un esperto nell'analisi delle funzionalità dei siti web, realizzando casi d'uso basati sulle funzionalità della piattaforma. I test si sono svolti manualmente utilizzando il programma Excel, verificando in ogni pagina casi a buon fine e non a buon fine, non sono stati utilizzati software per l'automazione di test.

La simulazione dei casi reali è stata invece svolta con l'aiuto di una wedding planner, che ha fornito informazioni su un evento passato da poter riproporre sulla piattaforma.

9.1 Sviluppi futuri

Lo stato attuale della piattaforma si trova all'80-90%, dal punto di vista del front-end sono state modificate tutte le pagine e, ne sono state aggiunte in base alle richieste dell'azienda. Per quanto riguarda gli sviluppi futuri si prevede l'aggiunta

alla piattaforma della parte di raccomandazione, questa permetterà agli utenti iscritti di essere guidati nell'organizzazione di eventi, sia come scelta dei fornitori, sia come tempistiche. Si prevede, infatti che, una volta completato il sito raccomanderà agli organizzatori i fornitori in base alla posizione geografica, all'organizzazione di eventi passati ed alle recensioni degli utenti. Si prevede inoltre un servizio Agenda che tramite delle mail, ricorderà agli utenti le tempistiche di organizzazione per la ricerca di un fornitore piuttosto che un altro, per non essere colti impreparati per la realizzazione dell'evento. Altra funzionalità prevista sarà l'aggiunta delle pagine per gli invitati, attualmente è possibile effettuare la registrazione come invitato, inserendo un codice di invito, che viene auto-generato dagli organizzatori; si prevede che tramite questo codice gli invitati possano accedere al blog, e possano quindi aggiungervi foto e video relativi all'evento. Inoltre, gli invitati potranno accedere ad altre funzionalità come la visione del menu, la location e altro che verrà definito successivamente.

Bibliografia

- [1] *Brand*. URL: www.glossariomarketing.it/significato/brand/ (cit. a p. 4).
- [2] S. Savona S. Pattuglia. «Live Communication». In: *LIVE COMMUNICATION - Tor Vergata* (lug. 2019). URL: <https://it.readkong.com/page/live-communication-7517801> (cit. a p. 4).
- [3] *Vocabolario Treccani online*. URL: <http://www.treccani.it/vocabolario/> (cit. a p. 9).
- [4] *Responsive Web Design di Ethan Marcotte 08 Giugno, 2010*. URL: <https://alistapart.com/it/argomenti/responsive-design/page/3/> (cit. a p. 9).
- [5] Laine Zhang Santala Jokinen Oulasvirta A. «Responsive and Personalized Web Layouts with Integer Programming. Proceedings of the ACM on Human-Computer Interaction». In: (2021) (cit. a p. 10).
- [6] Simona Tocci. *Guida al Web Design: Prima Parte - HTML5 e CSS3 (Sviluppo Web Vol. 1)*. it. Institution of Engineering e Technology, gen. 2021. (Visitato il 01/07/2021) (cit. a p. 10).
- [7] *Responsive pixel*. URL: <http://www.responsivepixel.com/> (cit. a p. 10).
- [8] Clarissa Peterson. *Learning Responsive Web Design, a beginner's guide*. en. O'Reilly Media, lug. 2014. ISBN: 144936294X. (Visitato il 03/07/2014) (cit. alle pp. 10, 11).
- [9] Ben Frain. *Responsive web design with HTML5 and CSS: develop future-proof responsive websites using the latest HTML5 and CSS techniques*. Packt Publishing, 2020 (cit. a p. 11).
- [10] *How to Make your Angular Responsive Web App Design?* URL: <https://aglowiditsolutions.com/blog/angular-responsive/> (cit. a p. 13).
- [11] *Wikipedia: HTML 5*. URL: <https://it.wikipedia.org/wiki/HTML5> (cit. a p. 13).
- [12] *html.it*. URL: <https://www.html.it/> (cit. a p. 13).

- [13] *Css Box Model*. URL: <https://www.html.it/pag/14223/il-box-model/> (cit. a p. 16).
- [14] *Posizionare gli elementi con i CSS*. URL: <https://www.html.it/pag/14235/posizionamento-degli-elementi/> (cit. a p. 17).
- [15] *Responsive breakpoints*. URL: <https://getbootstrap.com/docs/4.0/layout/overview/> (cit. a p. 19).
- [16] Vincenzo Giacchina. *Web design con Bootstrap*. it. 2018. ISBN: 9788850334001 (cit. a p. 22).
- [17] Daniel E. Miller. *React Native: Scopri la guida completa alla programmazione di siti internet e web app con ReactJs, costruisci soluzioni scalabili con GraphQL e sviluppa applicazioni Full Stack. (Paperback)*. it. Institution of Engineering e Technology, lug. 2021. (Visitato il 01/07/2021) (cit. a p. 24).
- [18] *Style Tiles*. URL: <http://styletil.es/> (cit. a p. 28).
- [19] Alessandro Stella. *User Experience: Teorie, tecniche e metodologie per artefatti che regalano la felicità*. 2017. ISBN: 9781521996911 (cit. a p. 31).
- [20] Donald Norman, Jim Miller e Austin Henderson. «What You See, Some of What's in the Future, And How We Go About Doing It: HI at Apple Computer». In: gen. 1995. DOI: 10.1145/223355.223477 (cit. a p. 31).
- [21] Jacopo Pasquini e Simone Giomi. *Web usability: guida completa alla user experience e all'usabilità per comunicare e vendere online*. Hoepli, 2014 (cit. a p. 31).
- [22] Daniel M. Brown. *Communicating design: developing web site documentation for design and planning*. 2^a ed. New Riders, 2011 (cit. alle pp. 39, 41).
- [23] Asbjørn Følstad e Odd-Wiking Rahlff. «Challenges in conducting user-centered evaluations of mobile services». In: *Human-Computer Interaction International Conference, Las Vegas, USA*. 2005 (cit. a p. 42).
- [24] Lex S van Velsen, Thea M van der Geest e Rob F Klaassen. «User-centered evaluation of adaptive and adaptable systems». In: *fifth workshop on User-Centred Design and Evaluation of Adaptive Systems. June 20th, Dublin, Ireland*. 2006 (cit. a p. 42).
- [25] G Gediga, KC Hamborg e I Düntsch. *Evaluation of Software systems.-V: Encyclopedia of computer science and technology (str. 166-192)*. 2001 (cit. a p. 43).
- [26] Obead Alhadreti e Pam Mayhew. «Rethinking thinking aloud: A comparison of three think-aloud protocols». In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–12 (cit. a p. 44).

- [27] Maaïke J Van den Haak, Menno DT de Jong e Peter Jan Schellens. «Employing think-aloud protocols and constructive interaction to test the usability of online library catalogues: a methodological comparison». In: *Interacting with computers* 16.6 (2004), pp. 1153–1170 (cit. a p. 44).
- [28] Freddy Paz e José Antonio Pow-Sang. «A systematic mapping review of usability evaluation methods for software development process». In: *International Journal of Software Engineering and Its Applications* 10.1 (2016), pp. 165–178 (cit. a p. 45).
- [29] «Issues for Usability Evaluation in Industry: Seminar Discussions». In: *Usability Evaluation In Industry* (1996), pp. 255–262. DOI: 10.1201/9781498710411-42 (cit. a p. 45).
- [30] Serdar Doğan, Aysu Betin-Can e Vahid Garousi. «Web application testing: A systematic literature review». In: *Journal of Systems and Software* 91 (2014), pp. 174–201 (cit. a p. 48).
- [31] Business Internet Group di San Francisco. «The Black Friday Report on Web Application Integrity.» In: (2003) (cit. a p. 48).
- [32] Rani E. & Priyadarshini Afroz S. «Web Application – A Study on Comparing Software Testing Tools». In: (2011) (cit. a p. 48).
- [33] Derar Eleyan Samer Al-Zain e Yousef Hassouneh. «Comparing GUI Automation Testing Tools for Dynamic Web Applications». In: (2013) (cit. a p. 48).
- [34] Dr. S. M. Afroz N. Elezabeth Rani N. Indira Priyadarshini. «Web Application– A Study on Comparing Software Testing Tools». In: *International Journal of Computer Science and Telecommunications* 2.3 (giu. 2011) (cit. a p. 49).
- [35] *Software testing by Jiantao Pan available at*. URL: http://www.ece.cmu.edu/~roopman/des-899/sw_testing (cit. a p. 49).
- [36] *Wikipedia: White Box*. URL: https://it.wikipedia.org/wiki/Test_strutturale (cit. a p. 50).
- [37] *Software testing glossary available*. URL: <http://www.aptest.com/glossary.html#performance> (cit. a p. 50).
- [38] *Grey box testing*. URL: https://en.wikipedia.org/wiki/Gray_box_testing#grey_ (cit. a p. 50).
- [39] F. Ricca e P. Tonella. «Analysis and testing of Web applications». In: *Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001* (). DOI: 10.1109/icse.2001.919078 (cit. alle pp. 51, 52).
- [40] Addison-Wesley Professional. *Agile Testing: A Practical Guide for Testers and Agile Teams*. eng. 2008 (cit. a p. 52).

- [41] S. Elbaum, S. Karre e G. Rothermel. «Improving web application testing with user session data». In: *25th International Conference on Software Engineering, 2003. Proceedings.* (2003). DOI: 10.1109/icse.2003.1201187 (cit. a p. 52).
- [42] Samer Al-Zain, Derar Eleyan e Yousef Hassounah. «Comparing GUI automation testing tools for dynamic web applications». In: *Asian Journal of Computer and Information Systems* 1.2 (2013) (cit. a p. 53).
- [43] Rosnisa Abdull Razak e Fairul Rizal Fahrurazi. *Agile testing with Selenium*. 2011. DOI: 10.1109/MySEC.2011.6140672 (cit. a p. 54).