



**Politecnico  
di Torino**

# POLITECNICO DI TORINO

Corso di Laurea Magistrale in Ingegneria Informatica

Ottobre, 2022

## IMPLEMENTAZIONE DI UN'INTERFACCIA WEB ACCESSIBILE PER UNA SCUOLA FORMATIVA MUSICALE ONLINE

Relatore

Prof ANTONIO SERVETTI

Candidato

DARIO MONTELLA

*Get your facts first, then you can distort them as you please.*

*—Mark Twain*

Name: DARIO MONTELLA

Date of Degree: Ottobre, 2022

Title of Study: IMPLEMENTAZIONE DI UN'INTERFACCIA WEB ACCESSIBILE PER UNA SCUOLA FORMATIVA MUSICALE ONLINE

Major Field: INGEGNERIA INFORMATICA

Abstract: Oggi, nel 2022, in seguito anche alle difficoltà sorte nei precedenti anni a causa dell'insorgere della pandemia di Covid19, la tecnologia ha dovuto rispondere alle necessità degli uomini di remotizzare molte delle attività che fino a poco tempo fa venivano svolte esclusivamente in presenza. Uno degli ambiti che ha subito maggiore impulso da suddette circostanze è sicuramente l'ambito formativo, poiché in questo contesto l'interazione tra personale didattico e studenti presenta una serie di sfide per quel che riguarda l'efficacia, la semplicità e la completezza della comunicazione. Questa adozione crescente nell'utilizzo di sistemi di comunicazione remota ha evidenziato che lo sviluppo di applicazioni web non ha ancora raggiunto la giusta consapevolezza per quel che riguarda la distribuzione di questi servizi a tutti i tipi di utenti che possano trovarsi dall'altro lato dello schermo. L'accessibilità del web rimane ancora un tema da molti ignorato o quanto meno sottovalutato.

Questa tesi consiste in una applicazione web che ha lo scopo di garantire una gestione completa di una scuola di formazione in ambito musicale. Gli aspetti di interesse che l'applicazione si propone di affrontare sono la gestione di tutte le risorse associate al contesto scolastico in questione quali personale studentesco, docenti, personale amministrativo e classi di apprendimento di diverso tipo. L'applicazione deve inoltre proporre agli utenti un'interfaccia per utilizzare una comunicazione interattiva che permetta a studenti e insegnanti di comunicare in maniera remota e di performare a distanza esibizioni musicali mediante Browser. Una considerevole percentuale dello sforzo implementativo si è concentrata nel rendere il servizio fruibile al maggior spettro possibile di utenti, in cui in particolare ricadono gli utenti che per disabilità o altre problematiche sono costretti ad interagire con il web utilizzando tecnologie assistive o altri strumenti, piuttosto che la normale interazione mediante interfaccia grafica e mouse.

Sviluppare un'applicazione web mantenendo il focus sulla sua accessibilità rappresenta oggi una sfida, è necessario mettersi nei panni degli utenti che interagiscono mediante strumenti alternativi quali Screen Reader, tastiere o altre tecnologie e quindi testare l'utilizzabilità dell'applicazione attraverso tali strumenti, in tutti gli scenari possibili. Si tratta di uno sforzo non trascurabile che si aggiunge a quello già considerevole di sviluppo dell'applicazione, ma che nessuno sviluppatore può più permettersi di trascurare in tempi moderni.

## TABLE OF CONTENTS

Chapter	Page
<b>I Introduzione</b> . . . . .	<b>1</b>
1.1 Contesto applicativo . . . . .	1
1.2 Il Web é per tutti . . . . .	2
1.3 Educazione di qualità . . . . .	3
1.4 Strumenti utilizzati . . . . .	3
<b>II Obiettivi</b> . . . . .	<b>5</b>
2.1 Requisiti dell'applicazione . . . . .	5
2.2 Esibizioni musicali in interazione remota . . . . .	8
2.2.1 Box Raspberry . . . . .	8
2.3 Accessibilità . . . . .	11
2.3.1 Accessibilità tramite tastiera . . . . .	11
2.3.2 Accessibilità da Screen Reader . . . . .	14
<b>III Tecnologie Abilitanti</b> . . . . .	<b>21</b>
3.1 Vue . . . . .	21
3.1.1 Vantaggi di Vue . . . . .	21
3.1.2 Strumenti aggiuntivi di Vue . . . . .	23
3.2 Tailwind . . . . .	27
3.2.1 Vantaggi del framework . . . . .	27
<b>IV Architettura dell'applicazione</b> . . . . .	<b>31</b>
4.1 Principali componenti . . . . .	31
<b>V Conclusioni</b> . . . . .	<b>42</b>
<b>References</b> . . . . .	<b>44</b>

## LIST OF FIGURES

Figure	Page
2.1 Element focus in MEVO Application . . . . .	13
2.2 Barra di navigazione nell'applicazione . . . . .	14
2.3 Aria-label example from MEVO . . . . .	15
2.4 Aria-required example . . . . .	15
2.5 Aria-hidden esempio . . . . .	17
2.6 Esempio utilizzo di "role" . . . . .	18
2.7 Notifiche nell'applicazione . . . . .	19
2.8 Aria-live example from MEVO . . . . .	20
2.9 Detail Knob . . . . .	20
3.1 Vuex big view . . . . .	23
3.2 breve codice dal vue store dell'applicazione . . . . .	24
3.3 breve estratto dal router dell'applicazione . . . . .	26
3.4 Tailwind breakpoints . . . . .	28
3.5 Tailwind responsive esempio . . . . .	29
4.1 Authentication . . . . .	32
4.2 Navigation Bar . . . . .	33
4.3 Access to settings . . . . .	33
4.4 Classes . . . . .	34
4.5 Create class . . . . .	34
4.6 Create class: classes detail . . . . .	35
4.7 Create room . . . . .	35

4.8	Class page: utenti iscritti . . . . .	36
4.9	Rimozione utenti . . . . .	37
4.10	Iscrivere utenti a classi . . . . .	37
4.11	Vista amministratore degli utenti . . . . .	38
4.12	Vista segretario degli utenti . . . . .	38
4.13	Selezione Box . . . . .	39
4.14	Configurare Box . . . . .	40
4.15	Mixing View . . . . .	41

## ABBREVIATIONS

API	Application Programming Interface
SFC	Single File Components
ARIA	Accessible Rich Internet Applications

# CHAPTER I

## Introduzione

### 1.1 Contesto applicativo

Il Web rappresenta oggi il mezzo di interazione principale tra gli attori di qualsiasi ambito, sia esso professionale, formativo, ricreativo o qualunque altro. Nello scenario moderno é estremamente raro trovare servizi la cui efficacia non passi per la distribuzione di funzionalità accessibili mediante Internet.

Gli istituti di formazione non fanno eccezione. Adesso non solo le scuole private ma anche gli istituti pubblici sono dotati di strumenti web per poter migliorare l'accesso alle risorse e la fruizione di servizi. Questa consapevolezza ha trovato un fortissimo impulso negli ultimi anni a causa dell'epidemia di COVID19 che si é diffusa a partire dall'anno 2019 e che ha costretto gli istituti formativi di tutto il mondo ad adottare soluzioni innovative per poter rendere ancora piú immediata l'interazione tra studenti e insegnanti da remoto e renderla quanto piú possibile equivalente funzionalmente ad un'interazione in presenza, cioé garantendo una comunicazione audio e video che fosse opportuna per le esigenze degli utenti.

Il progetto in analisi si pone in questo tipo di contesto, si propone di implementare un'interfaccia web per la gestione di una scuola di formazione in ambito musicale. L'applicazione offre un controllo diretto sulle risorse di tipo amministrativo che un istituto di formazione ha bisogno di tracciare, come la gestione degli studenti, insegnanti, personale tecnico e segretario e consente ad ogni tipo di utente di usufruire istantaneamente delle informazioni e dei servizi disponibili sul sistema, creando un mezzo di controllo che sia facile e intuitivo per il personale amministrativo e propo-

nendo un mezzo di accesso a tutte le informazioni utili al personale studentesco e docenti.

L'ambito di formazione in cui si pone il progetto, cioè quello musicale, fa sì che l'applicazione debba rispondere in maniera ancora più diretta alla necessità di una comunicazione veloce tra studenti e insegnanti. La principale sfida dell'applicazione risiede infatti in una comunicazione audio a bassa latenza che consenta a studenti ed insegnanti di ricevere e inviare flusso audio e di eseguire delle modulazioni su di esso mediante la stessa interfaccia web dell'applicazione, che si comporta in questo caso come un vero e proprio Mixer Audio.

## **1.2 Il Web é per tutti**

Le applicazioni web sono uno strumento altamente pervasivo e in alcuni ambiti il loro uso non é più opzionale. Questa consapevolezza deve essere sempre tenuta presente dagli sviluppatori che devono fornire un'interfaccia facilmente accessibile e utilizzabile. La complessità delle tecnologie utilizzate deve sempre nascondersi all'utente, al quale deve essere sempre fornita un'esperienza di navigazione ed utilizzo gradevole. Per questo motivo l'applicazione offre sempre un'interfaccia semplice che possa essere concettualmente compresa anche da un utente non esperto del dominio applicativo o da un utente non abituato ad utilizzare questo tipo di tecnologie online. L'applicazione di questa tesi nasce fin dall'inizio con una filosofia ben determinata: quella che il web deve essere uno strumento accessibile a qualsiasi utente, dall'utente medio capace di interagire con l'applicazione mediante i canali di input output più noti come mouse e schermo, all'utente che interagisce tramite tastiera e strumenti di tecnologia assistiva come gli Screen Reader. Pertanto un discreto sforzo nella fase di sviluppo della Web Interface si é concentrata proprio nel rendere le funzionalità dell'applicazione totalmente fruibili anche da parte di chi si avvale di tali tecnologie.

### 1.3 Educazione di qualità

L'aspetto piú importante di un sistema online addetto alla formazione di studenti é sicuramente la qualità dell'interazione tra personale studentesco e personale docenti. Gli alunni hanno bisogno di un contatto diretto per poter apprendere i concetti esposti dagli insegnanti, non basta esclusivamente fornire loro del materiale da studiare, ma occorre che il docente possa intervenire nel processo di apprendimento se necessario. Questo vale per qualsiasi ambito formativo, ma nel contesto applicativo a cui questo progetto riferisce la qualità dell'interazione non é un semplice requisito ma un'assoluta necessità. Parlando di studenti che devono apprendere come eseguire delle performance utilizzando degli strumenti musicali é evidente che l'interazione remota audio, tra studenti e insegnanti o anche tra studenti e studenti in caso di apprendimento non supervisionato, debba essere di grande qualità per consentire effettivamente una crescita qualitativa agli allievi coinvolti nel processo.

### 1.4 Strumenti utilizzati

Tale scenario é stato concretizzato implementando una applicazione composta da un client Web che interagisce con un server che espone i propri servizi mediante una REST API. Il client Web stato realizzato in Javascript sfruttando il framework Vue, strumento relativamente recente all'interno dello scenario delle Web Application ma la cui adozione é in rapida crescita grazie alla sua filosofia moderna e intuitiva sulla reattività delle pagine web nonché grazie alla presenza di una molteplicitá di strumenti per gestire tutti i possibili requisiti di una moderna applicazione. Al fine di garantire una corretta usabilitá dell'applicazione anche da parte di utenti che navigano il web mediante tecnologie assistive, é stato necessario testare l'applicazione utilizzando mezzi di navigazione alternativi, quali tastiera in sostituzione del mouse e Screen Reader per la lettura di pagine web in caso di utenti non vedenti. Questo sforzo aggiuntivo rappresenta un valore aggiunto dell'applicazione che sicuramente

puó essere considerata come un servizio pienamente accessibile.

## CHAPTER II

### Obiettivi

L'obiettivo del progetto é quello di implementare un sistema per la gestione di una scuola di formazione in ambito musicale che distribuisce i suoi servizi online.

#### 2.1 Requisiti dell'applicazione

L'interfaccia si propone di offrire agli utenti la totalitá delle funzionalitá necessarie per rendere pienamente operativo un servizio collocato in questo dominio applicativo.

I requisiti che l'applicazione deve soddisfare sono i seguenti:

- **Distinzione di diversi tipi di utenti all'interno del sistema.**

Data la vastitá di servizi e risorse presenti nel contesto applicativo, é necessario distinguere diversi tipi di utenti e discriminare le funzionalitá accessibili in base al ruolo dell'utente all'interno del sistema, in modo tale da garantire ad ognuno di essi esclusivamente la disponibilitá di operazioni conformi al proprio ruolo all'interno dell'ambito.

L'accesso al servizio puó avvenire infatti esclusivamente previa autenticazione.

Ogni utente é caratterizzato da una matricola che funge da identificativo, nome, cognome, email e un ruolo.

Possiamo distinguere le seguenti categorie di utenti all'interno del sistema, in ordine gerarchico crescente:

- Studenti: gli studenti hanno un accesso di tipo sola lettura, possono unicamente vedere le classi a cui essi sono iscritti e partecipare alle Room create

per queste stesse classi.

Non hanno alcun diritto di modifica su nessun tipo di risorsa per quanto riguarda classi e utenti. Fanno eccezione gli studenti iscritti a classi di tipo "Self-management" che possono creare delle Room per tali classi e aggiungere altri studenti.

Deve essere invece possibile per lo studente accedere a tutte le funzioni disponibili riguardanti le Box Raspberry, vale a dire selezionare, rilasciare, configurare una Box. Lo studente può modificare i parametri di Mixing della Box in caso di connessione ad una Room mediante Box Raspberry (vedere sezione apposita).

- Insegnanti: gli insegnanti hanno gli stessi diritti degli studenti per quanto riguarda l'accesso alle risorse di tipo classi e utenti, essi possono creare le Room per le classi a cui sono iscritti e accedere alle funzionalità delle Box Raspberry.
- Segretari: i segretari possono modificare tutte le classi, questo comprende sia la modifica delle informazioni della classe sia la modifica, aggiunta o cancellazione di studenti e insegnanti iscritti alla classe stessa.

Hanno inoltre diritto di modifica sugli utenti che si trovano gerarchicamente al di sotto di essi per quanto riguarda l'accesso a risorse. Possono pertanto aggiungere, eliminare e modificare utenti di tipo studente e di tipo insegnanti.

- Tecnici: hanno gli stessi diritti dei segretari, con l'estensione di poter gestire anche gli utenti di tipo "segretario".
- Amministratori: hanno accesso in lettura e scrittura a tutte le risorse dell'applicazione.

- **Gestione del personale:** aggiunta, modifica, cancellazione degli utenti pre-

sentì nel sistema

- **Gestione delle classi di apprendimento:** aggiunta, modifica, cancellazione di classi e iscrizione degli utenti (unicamente studenti ed insegnanti) ad esse.

Le classi sono caratterizzate da un tipo tra quelli indicati in seguito:

- Classi di Gruppo: composte da almeno uno studente e almeno un insegnante
- Classi Individuali: composte da un solo studente e almeno un insegnante
- Classi Musical: associate ad un evento descritto da una data e da una URL e composte da almeno uno studente
- Classi di Self-Management: composte esclusivamente da studenti

Le classi può trovarsi in uno dei due seguenti stati:

- Classe attiva: le informazioni della classe non possono più essere modificate, mentre possono ancora essere aggiunti/rimossi gli utenti iscritti
- Classe non attiva: la classe è ancora completamente editabile

- **Implementazione di una comunicazione audio remota per gli utenti iscritti ad una classe:** per performare esibizioni musicali anche a distanza utilizzando tecnologia WebRTC oppure strumentazione aggiuntiva, in particolare dispositivi Raspberry. Si veda la sezione apposita 2.2.

- **Accessibilità dei servizi:** a completare il quadro degli obiettivi del lavoro in oggetto, si aggiunge la volontà di distribuire il servizio in modalità pienamente accessibile a tutti i tipi di utenti, garantendo la navigabilità dell'applicazione mediante le principali tecnologie assistive usate in ambito web, cioè navigazione da tastiera e navigazione con supporto di lettori di schermo. Si veda la sezione apposita sull'accessibilità 2.3.

## 2.2 Esibizioni musicali in interazione remota

L'applicazione si propone come principale obiettivo quello di implementare un sistema di comunicazione audio di altá qualità che possa soddisfare i requisiti di apprendimento di studenti che si avvicinano alla formazione in ambito musicale, vale a dire apprendimento nell'utilizzo di strumenti musicali.

Questo tipo di apprendimento é sicuramente molto sensibile alla qualità della trasmissione audio da un interlocutore all'altro, di conseguenza si sono utilizzati degli strumenti per mantenere quanto possibile la comunicazione naturale e con una latenza controllata.

Questa comunicazione può avvenire in due modi:

- Mediante tecnologia WebRTC, utilizzando quindi esclusivamente il Browser
- Mediante le Box Raspberry

### 2.2.1 Box Raspberry

La comunicazione implementata utilizzando dispositivi Raspberry risponde in maniera piú specifica alle esigenze di bassa latenza. Su tali dispositivi viene avviata un'applicazione server HTTP che distribuisce dei servizi tramite un'interfaccia REST, l'applicazione Web di questa tesi interagisce con i Raspberry tramite questa interfaccia. Una volta che l'applicazione server viene avviata sul Raspberry, il dispositivo diventa una "Box" e può essere raggiunto usando gli endpoint da essa esposti.

L'applicazione esegue una scansione periodica per identificare i dispositivi Raspberry disponibili alla connessione, pertanto in ogni momento é possibile per l'utente prendere controllo della singola Box oppure rilasciare la risorsa. La Box Raspberry va connessa alla propria rete locale per rendere possibile il suo riconoscimento da parte dell'applicazione.

Nel momento in cui un utente acquista il controllo di una Box, esso può eseguire la configurazione dei parametri di base, vale a dire:

- Selezionare la scheda audio operativa tra quelle connesse alla Box dall'utente stesso
- Selezionare il numero di input (cioè di strumenti) il cui flusso si vuole inviare agli altri utenti connessi alla Room (attualmente per un massimo di due input). Ogni input è di tipo stereo.
- Selezionare il numero di output su cui si vuole ricevere il flusso audio degli altri utenti connessi alla Room. Attualmente il numero predefinito è due: corrispondente ad un output sinistro e uno destro.
- Selezionare un certo profilo audio per la scheda

Inoltre l'utente (se autorizzato) ha la possibilità di creare una Room di tipo Box per le classi a cui esso è iscritto. Una Room non è altro che un punto di accesso comune per tutti gli utenti iscritti alla classe mediante il quale essi possono unirsi ad una comunicazione interattiva.

Dopo la creazione della Room per una data classe, tutti gli studenti e insegnanti iscritti a quella classe potranno unirsi alla Room utilizzando una propria Box Raspberry.

La richiesta di unione ad una stanza da parte di un utente, comporta una richiesta alla Box da parte dell'applicazione. Questa richiesta deve specificare l'identificativo della stanza e l'indirizzo IP del Signaling Server, un server che si occupa di implementare concretamente la comunicazione delle Box Raspberry che si connettono ad esso.

Inviata questa richiesta, tutte le Box Raspberry che si sono "registrate" al Signaling Server per una determinata Room, sono connesse mediante il server e da questo momento potranno interagire scambiando contenuti audio.

Per poter inviare flusso audio ad una Box Raspberry, in modo che questa possa inviarlo a tutti gli altri utenti connessi ad una Room, é necessario che l'utente connetta ai dispositivi Raspberry una o piú schede audio a cui poter connettere la propria strumentazione. Per poter sfruttare tutte le funzionalità delle Box, cioè per ricevere e inviare flusso audio, é necessario connettere almeno uno strumento di input (come un microfono o uno strumento musicale) e almeno uno strumento di output, tipicamente delle cuffie.

Una volta che l'utente si é unito ad una stanza, può accedere alla schermata di Mixing per configurare la ricezione del flusso audio da parte degli altri utenti connessi alla Room, chiamati in questo contesto "Peers" proprio perché in background gli utenti sono connessi con connessione Peer to Peer.

La schermata di Mixing consente di effettuare delle operazioni di modulazione sul volume ricevuto dai Peer connessi. L'interfaccia in particolare consente: 1) la modulazione del volume complessivo (Master Volume) della Box Raspberry, che modula il volume ricevuto da tutti i Peer connessi; 2) la modulazione del volume di ciascun Peer connesso alla Room ; 3) la modulazione degli input di ogni Peer, per indicare alla Box come l'utente desidera ascoltare il flusso audio proveniente da quell'input 4) il volume specifico di ogni canale di input.

Nonostante la complessità delle configurazioni offerte, l'interfaccia é stata progettata in modo tale da presentare in maniera immediata le operazioni disponibili, rendendole quanto piú intuitive anche ad utenti poco familiari all'interazione mediante Browser. L'interfaccia cerca infatti di replicare in maniera sufficientemente fedele le operazioni che sono disponibili sui dispositivi hardware tipicamente usati per la configurazione e modulazione contemporanea del volume di piú strumenti, vale a dire i Mixer. Offrendo uno scenario simile a quello del contesto applicativo reale, l'applicazione vuole rendersi facilmente fruibile e soprattutto incoraggiare l'utente ad intraprendere questo nuovo approccio all'ambito applicativo, i cui vantaggi per quel

che riguarda la flessibilità del servizio e l'utilizzo in remoto sono fuori di dubbio.

## 2.3 Accessibilità

Quando si pensa all'interazione tra utente umano e applicazione tramite Desktop, è immediato figurarsi il tipico scenario dell'utilizzatore con mouse che clicca su un punto o l'altro dell'applicazione, la quale dal lato suo reagisce all'azione presentando un feedback grafico all'utente per permettere ad esso di apprezzare le conseguenze delle azioni da lui intraprese. Molte applicazioni si limitano a questa visione, considerandola l'unica possibile. Di fatto lo scenario presentato, esclude un'intera categoria di utenti. Un utente che a causa di una disabilità motoria non può utilizzare il mouse, o magari non può utilizzarlo in maniera scorrevole, sarà se non impossibilitato almeno scoraggiato dall'utilizzo della suddetta interfaccia. Un utente con una disabilità visiva invece vedrà preclusa qualsiasi possibilità di interagire con il servizio offerto. Questa è una problematica che oggi non può più essere ignorata, perché ormai tutti i servizi, anche le operazioni più comuni come eseguire un pagamento, un acquisto possono essere eseguite tramite Web e tutti gli utenti devono avere pari opportunità di eseguire queste azioni, indipendentemente dalle condizioni fisiche e motorie della persona. Il Web deve essere sviluppato anche per questa minoranza di utenti.

L'applicazione di queste tesi nasce con questa consapevolezza e l'implementazione dell'applicazione è stata condotta fin dall'inizio con l'obiettivo di garantire che tutte le funzionalità del progetto siano fruibili sia mediante i tipici strumenti che mediante quelli assistivi. Molte delle scelte implementative prese si basano sullo standard WAI-ARIA (W3C, 2022) che definisce le best practices per rendere un sito web accessibile.

### 2.3.1 Accessibilità tramite tastiera

Il primo aspetto a cui fare attenzione nel rendere un sito web accessibile è quello di rendere il sito completamente navigabile utilizzando esclusivamente la tastiera. È

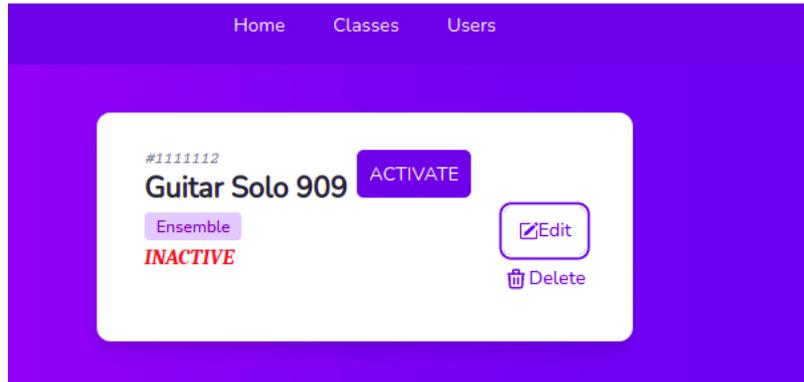
infatti condizione comune ad utenti con disabilità motorie nell'utilizzo di mani e/o muscoli quello di utilizzare tastiere o comunque strumenti assistivi che simulano il comportamento di una tastiera per interagire con la macchina.

**Evidenziare adeguatamente gli elementi della pagina** La navigazione da tastiera avviene tipicamente mediante il tasto "TAB" che scorre gli elementi navigabili della pagina. I moderni browser capiscono autonomamente quali sono gli elementi "reattivi" della pagina, come gli input di un form, i pulsanti, i link e li rendono navigabili tramite tastiera. É possibile per lo sviluppatore rendere navigabili anche elementi che non nascono per ricevere l'input dell'utente, utilizzando l'attributo "tabindex" (MDN, 2022b).

Nel momento in cui uno di questi elementi viene raggiunto dalla navigazione, si dice che esso ha ricevuto il "focus" dell'utente. Un elemento che riceve focus viene evidenziato tipicamente cambiandone il colore dei bordi, ad ogni modo ogni Browser esegue questa operazione in un certo modo. Il ruolo dello sviluppatore Web é eventualmente sovrascrivere questo comportamento del Browser se non sufficientemente chiaro, come spesso accade e introdurre uno stile che espliciti il focus dell'elemento in maniera ben visibile e anche in coerenza con lo stile dell'applicazione stessa.

Nell'applicazione di questa tesi, gli elementi interattivi della pagina sono stati tutti dotati di uno stile conforme al tema dell'applicazione e contemporaneamente ben evidenziati in modo tale da distinguere l'elemento dal resto della pagina. Sono stati molto utili gli strumenti offerti dal framework Tailwind, vedere apposito capitolo 3.2.

Figure 2.1: Element focus in MEVO Application



**Navigabilità della pagina** Oltre al modo in cui gli elementi mostrano il focus é importante l'ordine in cui gli elementi siano navigabili.

Un dettaglio importante che fa la differenza per chi naviga le pagine web tramite tastiera é la presenza di menu di navigazione, ovvero di link che consentono di spostarsi da un punto all'altro della pagina senza dover navigare ogni singolo elemento dal punto A al punto B. Si tratta di un accorgimento di fondamentale importanza soprattutto nel caso di pagine web con tanti contenuti al proprio interno. L'approccio da preferire per il displaying dei menu di navigazione é quello di impostarli tramite CSS non visibili di default e di renderli prominenti all'interno del layout nel momento in cui essi ricevono il focus tramite tastiera. In questo modo non andranno ad avere nessun impatto stilistico sulla pagina nel caso in cui non vengano navigati, ma diventeranno ben visibili se l'utente dovesse cercarli. É poi importante che nel caso un link di navigazione sia premuto, il contenuto raggiunto riceva il focus e ciò deve essere chiaramente comprensibile. Per poter sfruttare correttamente i menu di navigazione é necessario che la pagina sia implementata utilizzando una opportuna struttura di heading e di regioni, in particolare dovrebbero sempre essere usati gli elementi `<nav>` e `<main>` e le `<section>` per raggiungere determinati punti della pagina.

Nell'applicazione corrente la navigabilità tra le principali viste del sistema é stata implementata mediante una barra di navigazione, semplice e ben visibile, posizionata

come primo elemento raggiungibile da tastiera e quindi molto semplice da accedere. Questo garantisce la rapidità di navigazione anche ai Tab Users. Si è preferita una barra di navigazione piuttosto che un menù poiché nel caso specifico di questo progetto non vi sono viste particolarmente lunghe in contenuto bensì vi sono tante diverse viste da discriminare, pertanto una barra di navigazione fissa, cioè sempre disponibile in tutte le viste, rappresenta la scelta più consona al caso corrente.

Figure 2.2: Barra di navigazione nell'applicazione



### 2.3.2 Accessibilità da Screen Reader

Implementare in maniera opportuna la navigabilità del sito tramite tastiera è una pratica che rende più accessibile le pagine web per tutti gli utenti, anche e soprattutto per coloro che leggono le pagine tramite strumenti assistivi come gli Screen Reader. Gli Screen Reader sono strumenti automatizzati che elencano gli elementi all'interno delle pagine web tramite voce elettronica in maniera tale che anche utenti non vedenti possano utilizzare i servizi offerti dalle web applications. Rendere un sito completamente visibile e comprensibile da uno Screen Reader non è un'operazione scontata, è necessario seguire delle best practice al fine di ottenere un risultato soddisfacente.

**ARIA Attributes** La prima best practise per rendere un sito accessibile ad uno Screen Reader è quella di dotare tutti gli elementi interattivi della pagina di una descrizione fornita mediante attributi ARIA (MDN, 2022a). Questi attributi forniranno delle informazioni aggiuntive che verranno annunciate dagli Screen Reader, in modo che gli utenti con disabilità visive che usano questo strumento potranno avere una chiara esposizione dell'utilità di ciascun elemento della pagina.

L'attributo **"aria-label"** permette di specificare il contenuto audio da annunciare all'utente nel momento in cui la navigazione giunge sull'elemento in questione. Tutti gli elementi interattivi dell'applicazione sono stati dotati di un tale campo, sia elementi di form, sia pulsanti che altri mezzi di input come checkbox e altro ancora.

Figure 2.3: Aria-label example from MEVO

```
1 <button
2     type="button"
3     class="h-12 w-12 flex rounded-full text-white"
4     @click="someMethod"
5     aria-label="Do some very well explained
6         operation"
7 >
```

Gli attributi ARIA vengono utilizzati anche per segnalare ai Browser eventuali campi di compilazione obbligatori, in maniera tale che gli Screen Reader annuncino la necessità di inserire dati in questi campi

Figure 2.4: Aria-required example

```
1 <div class="md:flex md:items-center mb-6">
2     <div class="md:w-1/3">
3         <label
4             class="block text-gray-500 font-bold
5                 md:text-right mb-1 md:mb-0 pr-4"
6             for="inline-class-code"
7         >
8             Class Code
9         </label>
10    </div>
11    <div class="md:w-2/3">
12        <input
13            class="bg-gray-200 ..."
14            type="text"
15            v-model="code"
16            required
17            aria-required="true"
18        />
19    </div>
```

Nel progetto di questa tesi, tutti i form presenti all'interno dell'applicazione, come

quelli per effettuare il Login o quelli per l'inserimento di nuove entità nel sistema, quali utenti o classi, sono dotati di una caratterizzazione ARIA che segnala i campi obbligatori, in modo da informare opportunamente l'utente. Il codice inserito come esempio riguarda proprio un campo di compilazione per l'inserimento di una nuova classe.

Un'altra accortezza necessaria è quella di verificare se vi sono elementi della pagina che vanno nascosti agli Screen Reader. In alcuni casi, soprattutto quando si utilizzano elementi HTML di raggruppamento con un certo ruolo, gli Screen Reader tendono ad interpretare tutto ciò che c'è al loro interno annunciandolo all'utente, comprendendo sia informazioni utili sia eventuali elementi che sono puramente visivi e che non vogliamo vadano annunciati perché avrebbero come effetto unicamente quello di confondere l'utente che ascolta la descrizione della pagina.

Per rimediare a queste evenienze spiacevoli, è stato usato l'attributo "aria-hidden" per nascondere determinati elementi.

Figure 2.5: Aria-hidden esempio

```
1 <button
2   ....
3   >
4     <span
5       class="absolute inset-x-0 bottom-2 z-20 text-white"
6       aria-hidden="true"
7       >{{ value }}</span
8     >
9     <span class="absolute bottom-5 left-1 xs:-left-2
10        text-sm" aria-hidden="true"
11      >L</span
12    >
13    <span class="absolute inset-x-0 -top-3.5 text-sm"
14      aria-hidden="true"
15    >C</span
16  >
17  <span
18    class="absolute bottom-5 right-0 xs:-right-2 text-sm"
19    aria-hidden="true"
20    >R</span
21  >
22  ....
23 </button>
```

Il codice di esempio appena riportato é stato preso dal componente implementato per realizzare una manopola customizzata. Essendo il componente implementato come un `button` personalizzato, esso può ricevere nativamente il focus del Browser ma avendo al proprio interno dei campi testuali, se essi non fossero nascosti allo Screen Reader verrebbero annunciati quando il pulsante prende il focus, avendo uno spiacevole effetto per l'utente.

Per riconoscere la necessità di questi accorgimenti é necessario eseguire un test apposito di navigazione mediante Tecnologie Assistive, solo in questo modo é possibile per uno sviluppatore applicare in maniera completa le pratiche opportune per garantire la leggibilità della pagina da parte di queste tecnologie.

**Attributo ROLE** Gli elementi che possono ricevere il focus devono necessariamente essere dotati di una descrizione ARIA, in caso contrario qualora un utente con problemi visivi navigasse tali elementi non potrebbe in nessun modo capire a cosa servono. In aggiunta potrebbe essere una buona idea anche dotare di un attributo "role" elementi HTML di raggruppamento in modo tale che uno Screen Reader possa capire la natura del contenuto all'interno del contesto della pagina e annunciarlo all'utente nella maniera piú consona.

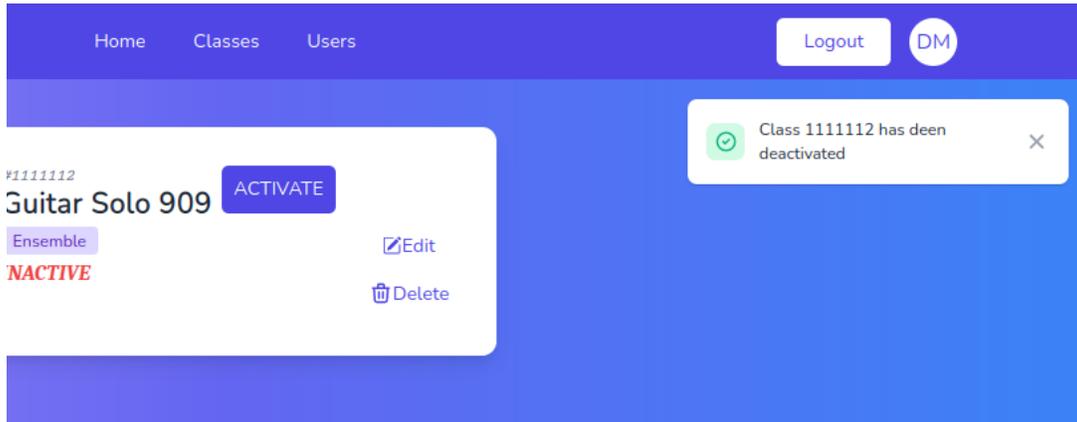
Ad esempio, se si vuole definire una notifica che avvisa l'utente di qualcosa, di un'operazione andata a buon fine o al contrario di eventuali problemi, é bene inserire il messaggio all'interno di un elemento segnato con l'apposito ruolo

Figure 2.6: Esempio utilizzo di "role"

```
1 <div
2   class="flex items-center w-full xs:max-w-xs p-4
      text-gray-800 bg-white rounded-lg shadow self-center
      md:self-end"
3   role="alert"
4 >
5 <p> Important Message! </p>
6 </div>
```

Questo é lo stesso metodo utilizzato all'interno dell'applicazione di questa tesi, inserendo il campo di notifica all'interno di un `ROLE = ALERT` la notifica risulta accessibile e annuncia il contenuto nel momento in cui l'elemento compare nella pagina

Figure 2.7: Notifiche nell'applicazione



**Regioni ARIA LIVE** Quando si implementano degli elementi pienamente accessibili che devono ricevere delle azioni dell'utente é generalmente necessario utilizzare degli strumenti per segnalare al Browser e quindi alle Tecnologie Assistive la presenza di contenuti in corso di modifica, ciò é necessario affinché l'utente con disabilità visive, esattamente come avviene all'utente senza disabilità, possa ricevere il feedback delle modifiche apportate e avere certezza della loro applicazione. Come accennato in precedenza comportamenti di questo tipo sono già inclusi in tutti gli elementi di input di HTML5, per tale motivo se un utente interagisce con una casella di testo lo Screen Reader verrà notificato dei nuovi caratteri inseriti e li annuncerà.

Per implementare un comportamento che reagisce ai cambiamenti annunciandoli allo Screen Reader é necessario definire delle regioni ARIA Live. Nella creazione di queste regioni bisogna anche fornire una configurazione sull'intrusività dello Screen Reader nell'annunciare i cambiamenti:

- Assertive mode: lo Screen Reader interrompe l'attuale task che sta svolgendo per annunciare il cambiamento
- Polite mode: lo Screen Reader annuncia il cambiamento alla prima occasione, senza interrompere i task correnti

- Off mode (default): lo Screen Reader non annuncia il cambiamento a meno che l'utente non abbia il focus sulla regione in questione

Figure 2.8: Aria-live example from MEVO

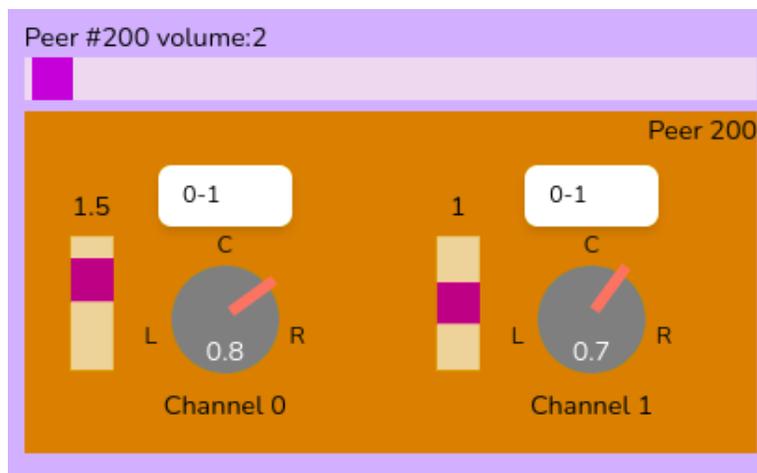
```

1 <div
2   role="status"
3   aria-live="polite"
4   aria-relevant="additions text"
5   class="sr-only"
6 >
7   <span>
8     {{ valuelivecontent }}
9   </span>
10 </div>

```

Nell'applicazione oggetto di questa tesi, una regione Aria Live é stata utilizzata nella schermata di Mixing per informare l'utente sull'effetto dell'utilizzo delle manopole per la modulazione degli input. Le manopole infatti sono un Componente grafico realizzato appositamente, non sono elementi di input html che conservano già al loro interno un comportamento accessibile. Pertanto é stato inserito un campo Aria Live che reagisce al cambiamento del valore associato alla manola per annunciare eventuali modifiche

Figure 2.9: Detail Knob



## CHAPTER III

### Tecnologie Abilitanti

#### 3.1 Vue

**Vue** é un framework per costruire in maniera semplice interfacce Web utilizzando HTML, CSS e Javascript. Si tratta di un progetto Open Source relativamente giovane, la prima versione é stata rilasciata nel 2014 da parte del creatore **Evan You**. Da allora la sua adozione é aumentata con regolarit  negli anni ed oggi la community che si occupa del Framework é aumentata di pari passo alla popolarit  di Vue. La versione pi  recente del framework nonch  quella utilizzata nel presente progetto é la versione 3.

##### 3.1.1 Vantaggi di Vue

**Framework Progressivo** Visitando la Documentazione ufficiale del framework Vue, possiamo notare come esso presenti come propria qualit  copertina quella di essere un **Progressive Framework**. Vue infatti ha la qualit  di essere molto modulare e flessibile, ci  fa si che le sue funzionalit  possano essere aggiunte in maniera incrementale alle applicazioni. Tutto ci  si traduce in pratica nel fatto che la Core Library di View si occupa esclusivamente del layer di visualizzazione, ci   permette di estendere il normale html con delle funzionalit  che rendono il codice della pagina "reattivo". Eventuali funzionalit  ulteriori possono essere adottate estendendo l'applicazione, ci   installando ulteriori librerie di Vue.js secondo esigenza. Questo approccio ha due conseguenze molto apprezzate da tutti gli sviluppatori:

1.   molto facile iniziare ad utilizzare il Framework, in quanto la difficolt  iniziale

é bassa ed incrementa man mano che aggiungiamo funzionalità. La curva di apprendimento si presenta pertanto molto lineare.

2. Renderizzazione veloce: la natura "lightweight" di Vue fa sì che il suo processo di renderizzazione sia molto rapido

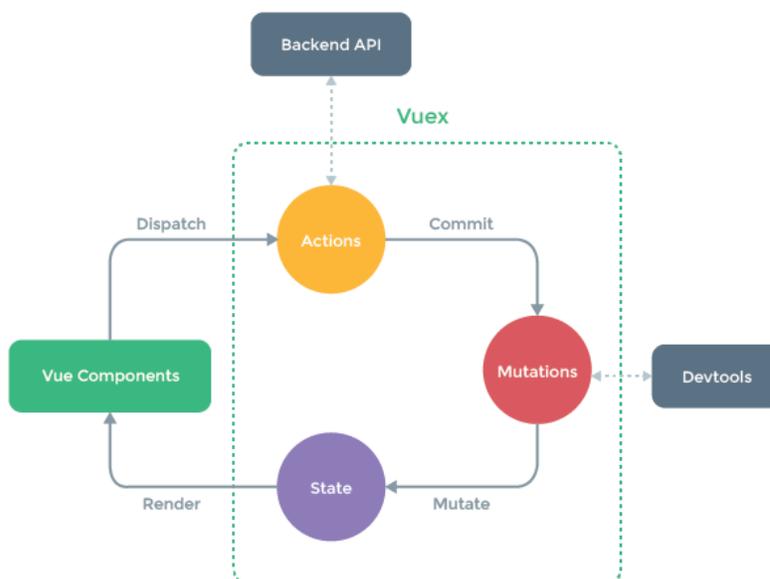
**Single File Components** Un vantaggio, seppur dibattuto, di Vue é che esso utilizza l'approccio dei Single File Components, secondo cui tutto ciò che riguarda un Componente, cioè il suo template HTML, la sua logica e il suo stile CSS, si trova all'interno di un unico file.

Il framework opera come filosofia di progettazione quella della semplicità e della utilizzabilità e i SFC sono un'espressione di tale scelta. Avere l'intera descrizione di un Componente all'interno di un unico file fa sì che tutti gli aspetti concernenti un determinato elemento possano essere facilmente individuati e capiti. La visione tradizionale degli sviluppatori web ha sempre prediletto la netta separazione di questi 3 aspetti: HTML, CSS e Javascript, ma la separazione di cui parliamo é una separazione funzionale, che rappresenta indubbiamente la scelta da incoraggiare. Quello utilizzato da Vue é ancora un approccio che mantiene funzionalmente separati questi 3 aspetti, ma la separazione funzionale non implica necessariamente una separazione in file diversi. Il fine ultimo dei principi di sviluppo delle applicazioni é quello di rendere il codice piú comprensibile e mantenibile e secondo Vue avere HTML, CSS e Javascript in file diversi é qualcosa che non migliora queste caratteristiche, bensí introduce complessità all'interno dei frontend. Concentrando invece tutti e 3 gli aspetti all'interno di un unico file, seppure ognuno nella propria sezione in maniera isolata, si raggiunge un codice molto piú coeso.

### 3.1.2 Strumenti aggiuntivi di Vue

**Vuex** Vuex (Vuex) é un pattern per la gestione dello stato nelle applicazioni Vue.js. In accordo alla filosofia incrementale di Vue, Vuex puó essere aggiunto alle applicazioni mediante la semplice installazione di una libreria. Lo scopo di questo strumento é quello di fornire un punto unico in cui conservare lo stato dell'applicazione in modo che tali informazioni siano accessibili da tutti i componenti dell'applicazione. Vuex si propone di astrarre lo stato condiviso tra piú componenti e isolarlo all'interno di uno store Singleton, in modo tale che l'intero albero di componenti possa comportarsi come un'unica grande vista che accede all'unico punto di storage di queste informazioni. Per garantire la coerenza all'interno dell'applicazione, la modifica di queste informazioni condivise puó avvenire unicamente mediante gli hook messi a disposizione dallo store, chiamati *mutations*.

Figure 3.1: Vuex big view



Al centro di ogni applicazione di Vuex é il concetto di *store*, un contenitore che contiene lo stato dell'applicazione. La particolaritá dello store é la reattivitá, cioé se

un componente reperisce informazioni da esso e ad un certo punto queste informazioni vengono modificate all'interno dello store in un qualsiasi punto dell'applicazione, allora il componente verrà immediatamente notificato del cambiamento in modo che il suo stato sia sempre aggiornato. Per garantire questo comportamento lo store mette a disposizione dei componenti dell'applicazione un unico strumento per il committing delle modifiche, quello delle Mutazioni.

Figure 3.2: breve codice dal vue store dell'applicazione

```
1 export default createStore({
2   state: {
3     refreshToken: loadRefreshToken(),
4     token: loadToken(),
5     id: loadId(),
6     username: loadUsername(),
7     firstName: "",
8     lastName: "",
9     alias: "",
10    notifications: notifications,
11    boxList: [],
12    selectedBox: undefined,
13  },
14  mutations: {
15    saveToken(state, token) {
16      localStorage.setItem("token", token);
17      state.token = token;
18      state.id = tokenUtils.getId(token);
19    },
20    saveRefreshToken(state, refreshToken) {
21      localStorage.setItem("refreshToken", refreshToken);
22      state.refreshToken = refreshToken;
23    },
24    clearToken(state) {
25      localStorage.removeItem("token");
26      state.token = undefined;
27      state.id = undefined;
28    },
29  }
}
```

Nel progetto implementato lo store Vuex é stato utilizzato per condividere tra componenti informazioni globali, tra cui:

- il nome dell'utente
- il refresh token per l'autenticazione
- la Box Raspberry attualmente selezionata
- L'eventuale profilo audio scelto per la box
- L'eventuale scheda audio selezionata tra quelle connesse alla box.

E molte altre ancora

**Vuerouter** Le applicazioni moderne seguono quasi tutte la filosofia della Single Page Application, cioè l'applicazione non si estende su più pagine web ma ne occupa una sola, controllando unicamente quali sono i Componenti da mostrare all'interno di tale pagina a seconda dello stato dell'applicazione e dalla URL caricata all'interno dello spazio indirizzi. Mappare un determinato Componente ad una specifica URL é compito dei Web Routers e Vue é dotato della propria apposita libreria, che si chiama appunto VueRouter (VueRouter). Una volta registrate all'interno del Vue Router le associazioni tra una specifica URL e il Componente da visualizzare per tale URL, il framework si occuperá autonomamente di inserire il giusto Componente nella pagina a seconda della URL nella barra degli indirizzi.

Nel presente progetto, il router di per sé é stato inserito all'interno di un file separato. Per ogni route é possibile inoltre specificare delle metainformazioni aggiuntive, esempio tipico di metainformazione é la possibilità di visualizzare la route unicamente previa autenticazione o se essa é disponibile anche per visitatori anonimi.

Specificare queste informazioni con Vue Router é facile ed intuitivo. Risulta molto semplice anche creare delle Nested Routes, cioè organizzare una struttura gerarchica andando ad indicare in una determinata route le possibili "sotto-routes" da includere nel Componente associato, a seconda del path della URL. A questo scopo si utilizza l'attributo *children*

Figure 3.3: breve estratto dal router dell'applicazione

```
1 const routes = [  
2   {  
3     path: "/",  
4     name: "Home",  
5     component: Home,  
6     meta: { forVisitors: true },  
7   },  
8   {  
9     path: "/user",  
10    name: "UserHome",  
11    // route level code-splitting  
12    // this generates a separate chunk (user.[hash].js) for  
13    // this route  
14    // which is lazy-loaded when the route is visited.  
15    component: () =>  
16    import(/* webpackChunkName: "user" */ "@views/  
17    UserHome.vue"),  
18    meta: { requiresAuth: true },  
19    children: [  
20      {  
21        path: "",  
22        name: "Main",  
23        // route level code-splitting  
24        // this generates a separate chunk (user-main.[hash  
25        ].js) for this route  
26        // which is lazy-loaded when the route is visited.  
27        component: () =>  
28        import(  
29          /* webpackChunkName: "user-main" */ "@views/  
30          user/UserMain.vue"  
31        ),  
32      },  
33    ],  
34  },  
35 ]
```

## 3.2 Tailwind

Tailwind (Tailwind, 2022) é un Framework CSS per la gestione completa del lato presentativo di interfacce web. Tailwind si definisce come un framework "utility-first", perché esso prevede l'utilizzo di classi di utility da applicare direttamente agli elementi della pagina web, senza alcuna necessità di scrivere alcun file CSS da parte dello sviluppatore. La forza di Tailwind é che la sua applicazione non impone assolutamente nessun design al sito né richiede alcun tipo di imposizione su come il sito deve sembrare. Lo sviluppatore ha piena libertà su come costruire l'interfaccia, Tailwind si limita a fornire delle classi che applicate agli elementi conferiscono caratteristiche stilistiche ben definite e facili da comprendere e modificare.

### 3.2.1 Vantaggi del framework

I vantaggi rispetto alla tradizionale applicazione di stile mediante file CSS sono evidenti:

- Nessuno spreco di tempo nel dover inventare nomi di classi CSS o applicare complicati selettori
- Non abbiamo più degli enormi file CSS, la cui dimensione e numero aumenta con l'aumentare della dimensione dell'applicazione.
- Una maggiore sicurezza nell'attuare modifiche. Se definiamo un file CSS che viene utilizzato da più componenti, nel momento in cui applichiamo una modifica avremo sempre il timore che in uno di questi componenti tale mutamento avrà effetti diversi da quello che ci aspettiamo. Le utility class di Tailwind invece sono locali all'elemento stesso, possiamo personalizzare in maniera indipendente dall'altro ogni singolo elemento della pagina in maniera estremamente semplice.

**Responsive Design** Al fine di rendere l'applicazione gradevole e fruibile su schermi di diverse dimensioni, partendo dai desktop fino ad arrivare agli smartphone, sono state sfruttate le notevoli caratteristiche di Tailwind sulla "responsivness" dello stile.

Tailwind permette di applicare classi responsive ovvero che reagiscono alla dimensione dei dispositivi su cui vengono visualizzate in modo da ottenere un comportamento adatto a tutti i tipi di dispositivi mediante l'applicazione di poche semplici classi.

Ogni classe può essere infatti applicata con un prefisso che specifica la dimensione del dispositivo a partire dalla quale tale classe dovrà intervenire attivamente nello stile.

Figure 3.4: Tailwind breakpoints

Breakpoint prefix	Minimum width	CSS
<code>`sm`</code>	640px	<code>`@media (min-width: 640px) { ... }`</code>
<code>`md`</code>	768px	<code>`@media (min-width: 768px) { ... }`</code>
<code>`lg`</code>	1024px	<code>`@media (min-width: 1024px) { ... }`</code>
<code>`xl`</code>	1280px	<code>`@media (min-width: 1280px) { ... }`</code>
<code>`2xl`</code>	1536px	<code>`@media (min-width: 1536px) { ... }`</code>

Basta far precedere la classe dall'apposito identificativo di breakpoint.

**Approccio Mobile First** L'approccio di default di Tailwind prevede l'applicazione delle classi di tipo Mobile First, questo significa che eventuali utility class senza prefisso di dimensione vengono applicate ai dispositivi di tutte le dimensioni, mentre le utility class con prefisso vengono applicate dalla dimensione specificata dal quel prefisso in su.

Figure 3.5: Tailwind responsive esempio

```
1 | 
```

Property "width" is 16 for all sizes under md, 32 from md to lg and 48 from 48 to above

**Supporto all'accessibilità** Le utility class di Tailwind possono inoltre essere applicate con un prefisso che identifica uno stato dell'elemento HTML, in maniera tale da applicare la classe unicamente quando l'elemento si trova in quello stato.

Questa potente funzionalità é stata usata massivamente nell'applicazione per specificare uno stile differente per gli elementi disabilitati e per quelli che hanno ricevuto il focus. Utilizzando proprio il prefisso "focus" é stato molto semplice dotare tutti gli elementi interattivi dell'applicazione di uno stile evidente e conforme al tema dell'applicazione, raggiungendo in questo modo uno degli obiettivi che ci siamo posti per quanto riguarda l'accessibilità della pagina web, vale a dire evidenziare correttamente lo stato della navigazione.

Tailwind inoltre propone allo sviluppatore una utility class apposita per indicare che un elemento debba essere visibile all'interno della pagina unicamente in caso di navigazione mediante Screen Reader, cioè la classe "sr-only".

**Possibili svantaggi** Il principale svantaggio quando si usa un framework composto da utility class piuttosto che una propria libreria CSS, é connesso alla mantenibilità e riusabilità degli stili. Definendo una libreria CSS personalizzata infatti possiamo racchiudere all'interno di determinate classi una serie di molteplici proprietà CSS, in tal modo l'applicazione di tutte queste proprietà ad elementi diversi richiederá unicamente l'utilizzo della classe wrapper. Usando le utility class invece se vogliamo applicare uno stesso stile a due elementi sará necessario applicare ad ogni elemento la stessa lista di utility class. Tailwind offre in verità dei metodi per racchiudere piú utility class in wrappers e applicarli liberamente, quindi l'utente puó a propria discrezione liberarsi da questo limite con Tailwind.

Un ulteriore possibile svantaggio é la scarsa possibilitá di personalizzazione lasciata dal Framework, ma anche ció con Tailwind é assolutamente fuori luogo. Il Framework infatti di default cerca automaticamente alla radice del progetto dell'applicazione un file "tailwind.config.js" in cui lo sviluppatore puó inserire tutte le personalizzazioni di cui ha bisogno. In questo file sará possibile per esempio definire dei nuovi breakpoint, nuovi colori oltre ai predefiniti, nuovi font testuali e molto altro.

## CHAPTER IV

### Architettura dell'applicazione

#### 4.1 Principali componenti

Per garantire l'adempimento degli obiettivi preposti all'applicazione, sono stati implementati numerosi componenti mediante cui accedere al totale delle funzionalità.

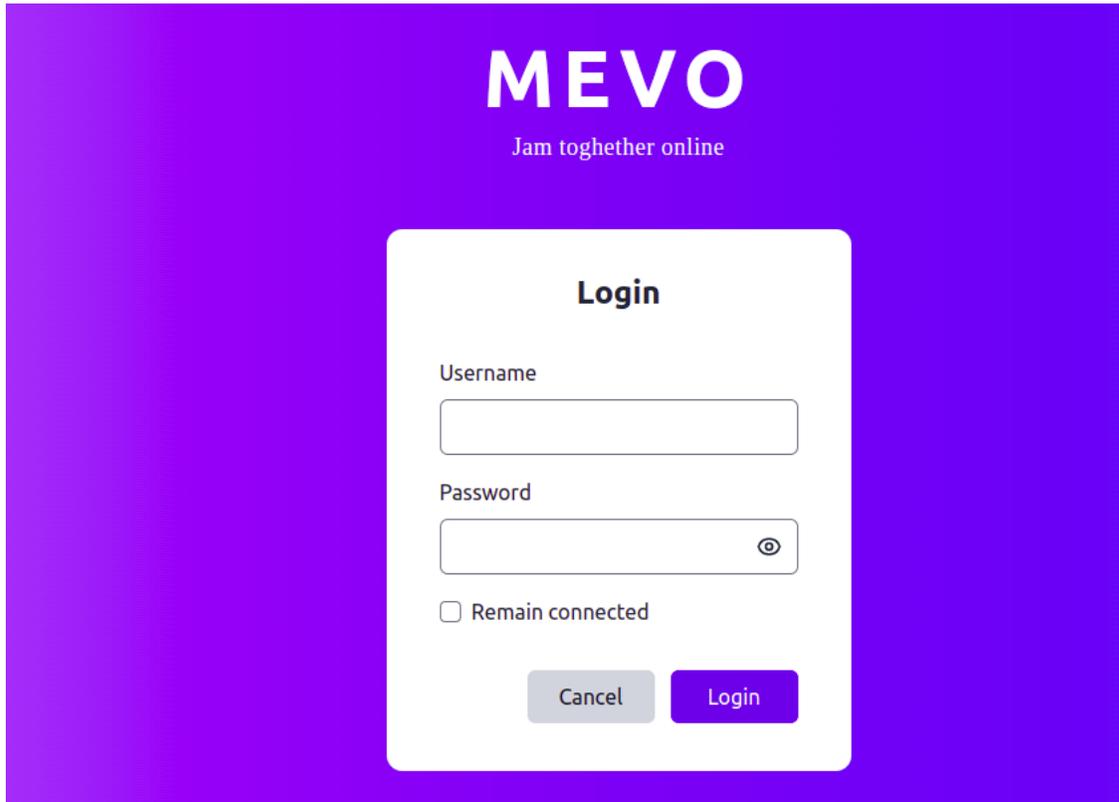
La lista delle principali viste raggiungibili mediante l'applicazione é contenuta nel file descrivente il Vue Router, dove viene registrata l'associazione tra un Componente e una URL

**Componente Home.vue** Associato alla URL `"/`

L'accesso a qualsiasi tipo di servizio offerto dal sistema puó avvenire solamente in seguito ad autenticazione. L'unico Componente a cui i visitatori hanno accesso é Home.vue ovvero il Componente nel quale viene effettuata l'autenticazione.

Il tentativo di accesso a qualsiasi altra URL da parte di un utente non autenticato riconduce a questo componente.

Figure 4.1: Authentication



**Componente Navbar.vue** La barra di navigazione é sempre visibile all'interno dell'intertaccia, eccezione fatta per le viste che rappresentano l'unione alla Room o alla comunicazione di tipo WebRTC.

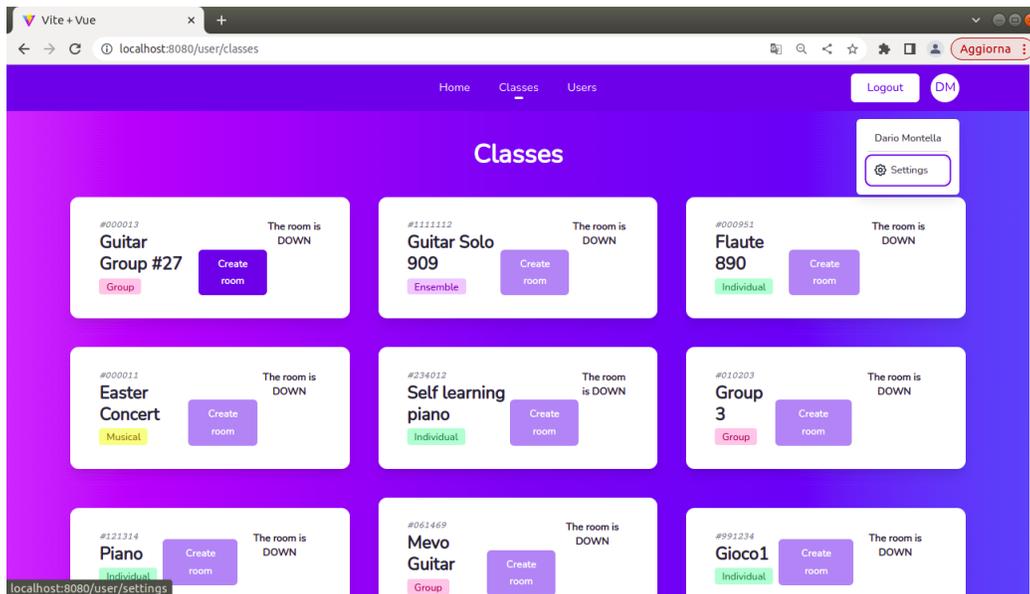
La presenza costante di questo strumento fa si che ci si possa muovere rapidamente da una vista all'altra, é completamente accessibile e le voci al suo interno dipendono dal ruolo dell'utente che ha effettuato l'accesso. Uno studente non vedrá gli stessi tab di navigazione di un amministratore, in particolare non potrà accedere alle funzionalità di gestione utenti.

Figure 4.2: Navigation Bar



Dalla barra di navigazione é inoltre possibile, tramite il pulsante dell'utente, accedere alle impostazioni "Settings" che riconduce al componente Settings.vue

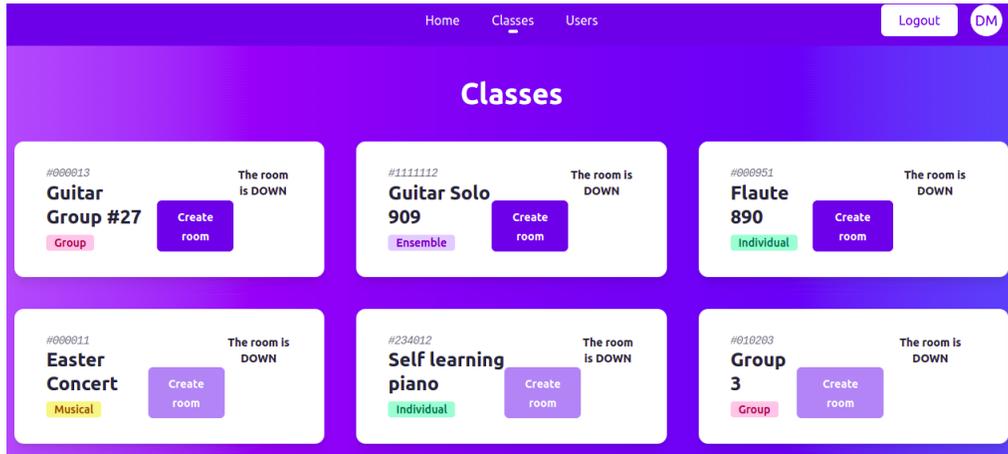
Figure 4.3: Access to settings



**Componente Classes.vue** Associato alla URL "user/classes"

Questo componente permette di visualizzare le Classi accessibili dall'utente che ha effettuato login nel sistema. Uno studente/insegnante pertanto visualizza unicamente le classi a cui esso é iscritto, gli utenti con permessi piú elevati possono invece visualizzare la totalitá delle classi.

Figure 4.4: Classes



In base al ruolo dell'utente sarà possibile o meno aggiungere una nuova classe, utilizzando un floating button.

Figure 4.5: Create class

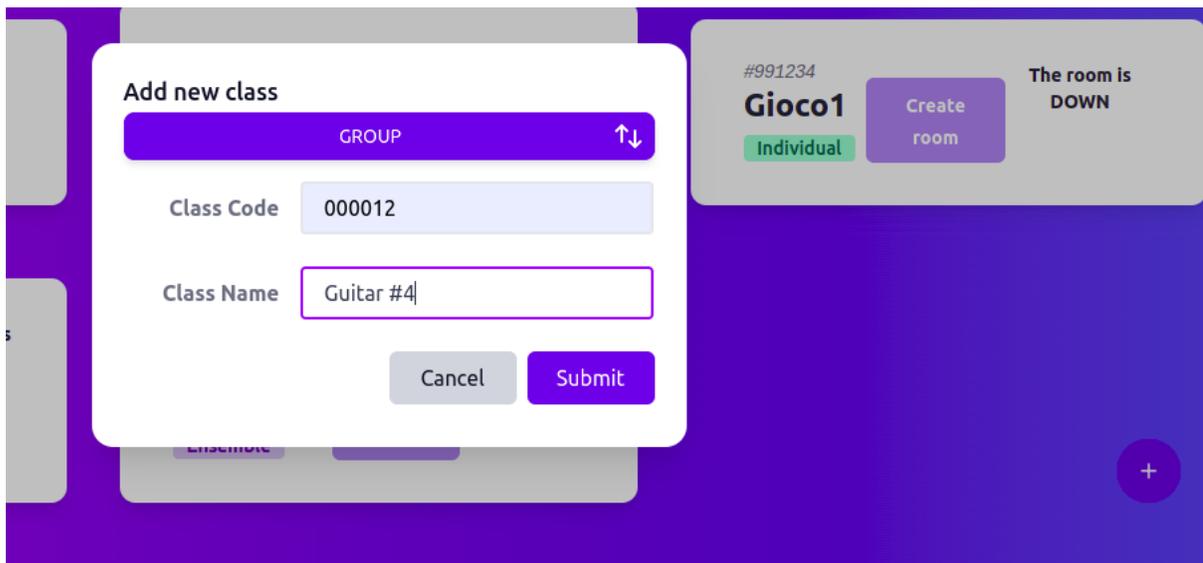
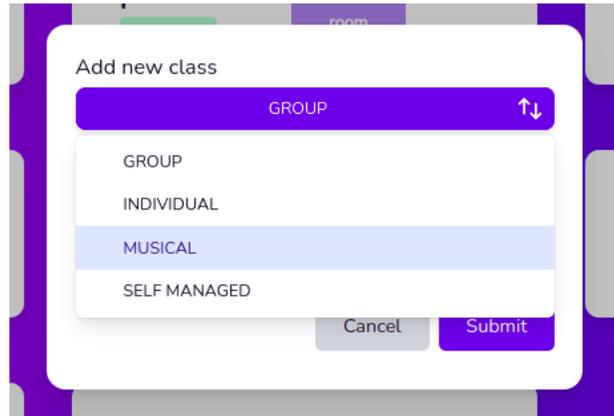
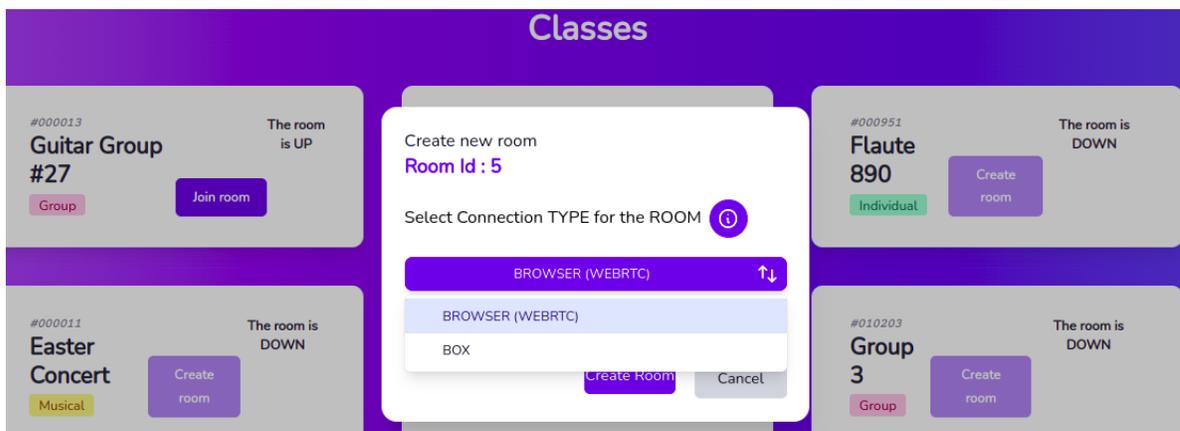


Figure 4.6: Create class: classes detail



Una volta che una classe é attiva, é possibile per gli studenti o gli insegnanti assegnati a questa classe creare una "Room". Una "Room" é un punto comune di accesso ad una comunicazione interattiva per gli utenti di quella classe. Tale comunicazione interattiva puó avvenire o mediante Browser utilizzando la tecnologia WEBRTC oppure utilizzando una Box Raspberry, quest'ultima opzione é disponibile solo previa selezione di una Box.

Figure 4.7: Create room

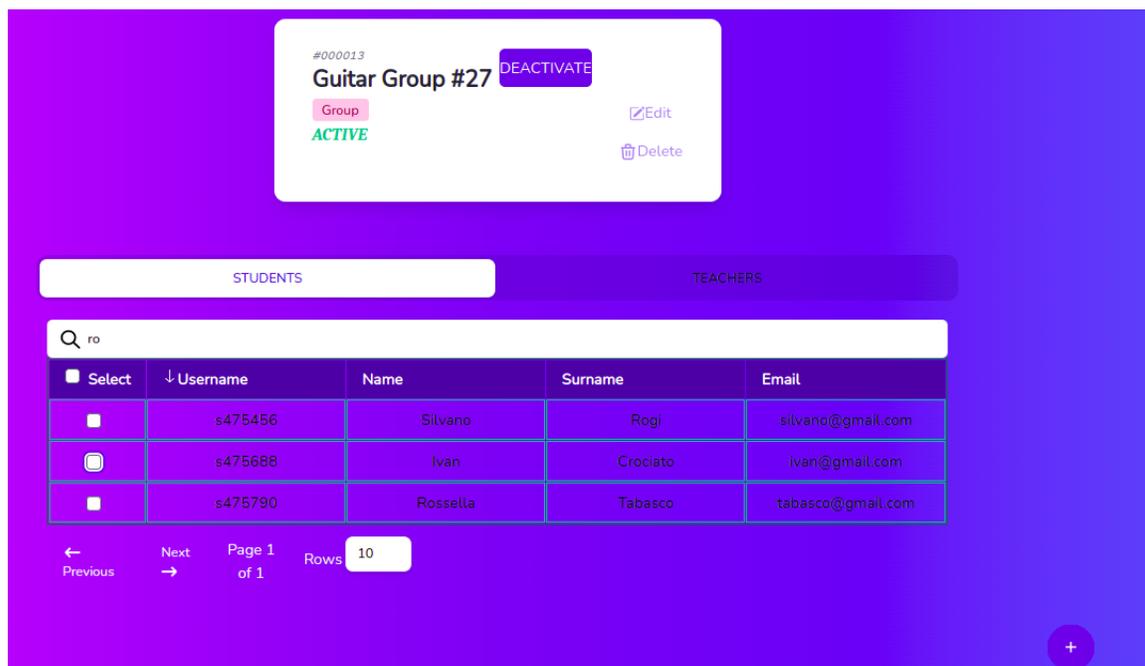


**Componente ClassDetail.vue** Associato alla URL "user/classes/:id" Premendo sul singolo class item della lista, sará possibile accedere a questo componente

Il Componente consente l'accesso ai dettagli della singola classe identificata dall'id presente come parametro all'interno della URL.

Tutti gli utenti possono visualizzare i dettagli delle classi a loro visibili. Solamente gli utenti dell'amministrazione possono però apportare modifiche. Inoltre qualsiasi tipo di modifica può essere apportata solo se la classe non è attiva.

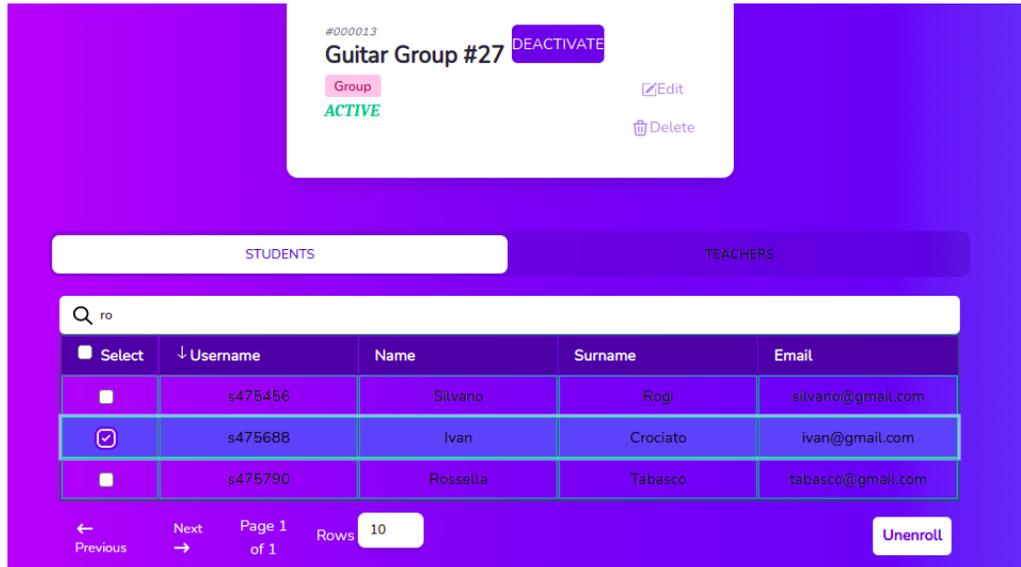
Figure 4.8: Class page: utenti iscritti



Una lista di due Tabs permette di visualizzare alternativamente gli studenti e gli insegnanti iscritti alla classe mediante una tabella. La tabella è dotata di una barra di ricerca per interrogare la lista di utenti che mostrerà solo le entry che soddisfano le parole chiave inserite nella barra.

Gli utenti dell'amministrazione hanno la possibilità di rimuovere studenti e insegnanti alle classi, selezionandoli all'interno delle tabelle appena descritte. Selezionando gli utenti in tabella usando le checkbox, diventa possibile rimuoverli dalla classe.

Figure 4.9: Rimozione utenti



L'amministrazione può inoltre aggiungere studenti e insegnanti, utilizzando il floating button in basso. Questa azione apre un modale per selezionare gli utenti (studenti o insegnanti) tra tutti quelli presenti nel sistema

Figure 4.10: Iscrivere utenti a classi



**Componente Users.vue** Il personale amministrativo deve avere la possibilità di accedere in lettura, scrittura e cancellazione agli utenti presenti nel sistema. A questo scopo, per gli utenti il cui ruolo ricade in questa categoria, è presente una vista per la visualizzazione degli utenti. In base al ruolo e alla sua posizione nella

gerarchia amministrativa, sarà possibile intervenire su un certo sottoinsieme di utenti. Un amministratore avrà chiaramente il controllo su tutti gli utenti.

Figure 4.11: Vista amministratore degli utenti

Select	↓ Username	Name	Surname	Email	Details
<input type="checkbox"/>	s275766	Dario	Montella	montellato@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275790	Davide	Marino	davidema@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275781	Lorenzo	Rungi	loruzzo@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275798	Angelo	Rosso	anro@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275666	Giovanni	Piodino	giopio@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275456	Pasquale	Mecca	pasmec@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>

Un segretario invece avrà controllo unicamente sul personale studentesco e corpo docenti.

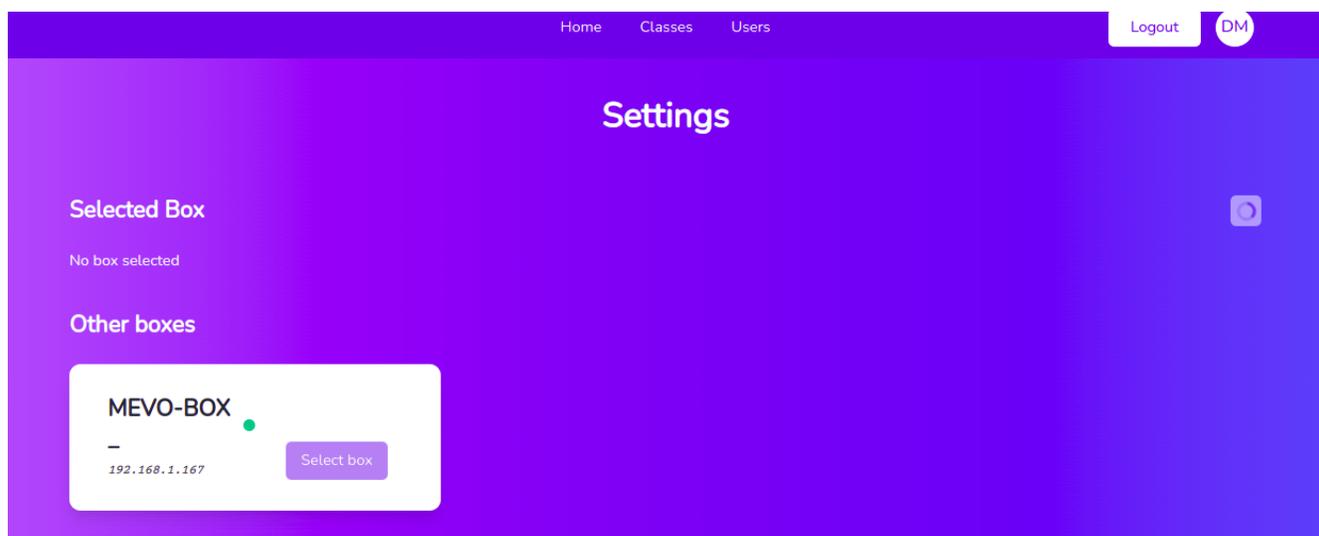
Figure 4.12: Vista segretario degli utenti

Select	↓ Username	Name	Surname	Email	Details
<input type="checkbox"/>	s275766	Dario	Montella	montellato@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275790	Davide	Marino	davidema@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275781	Lorenzo	Rungi	loruzzo@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275798	Angelo	Rosso	anro@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>
<input type="checkbox"/>	s275666	Giovanni	Piodino	giopio@gmail.com	<a href="#">Edit</a> <a href="#">Delete</a>

Uno studente o un insegnante invece non avrà alcuna possibilità di accedere a questo componente in quanto non è tra i suoi diritti la modifica del personale.

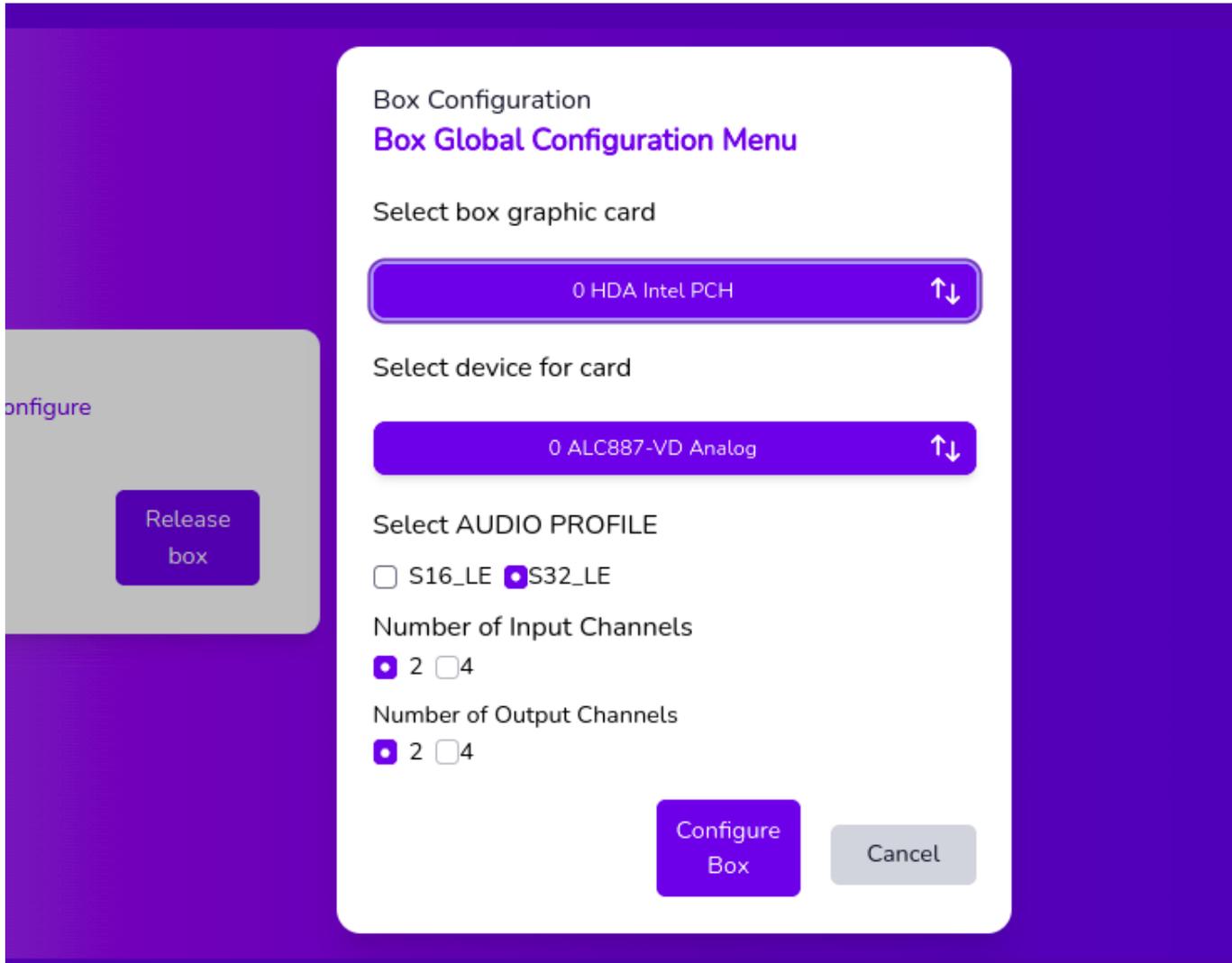
**Componente Settings.vue** Associato all'URL "/user/settings". L'applicazione esegue una ricerca periodica delle Box Raspberry connesse alla rete. Esse vengono identificate automaticamente nel momento in cui l'applicazione a bordo del dispositivo viene lanciata e il server http implementato dall'applicazione diventa raggiungibile. Da questo momento in poi l'interfaccia web mostra la possibilità di selezionare le Box.

Figure 4.13: Selezione Box



Inoltre da questo stesso componente è possibile effettuare la configurazione della Box Raspberry se ad essa è connessa una o più schede audio. La configurazione permette di scegliere la scheda audio che la Box dovrà prediligere per la connessione, il profilo audio da impostare e il numero di canali input - output.

Figure 4.14: Configurare Box



**Componente BoxRoom** Associato alla URL `"/user/box/:roomId"`. Questo é il componente che ospita l'interfaccia per l'interazione remota tra utenti connessi alla stessa Room mediante Box Raspberry.

Questo componente esegue la ricerca periodica dei Peer connessi alla Box. Da questa schermata inoltre l'utente puó eseguire il mixing della strumentazione connessa alla scheda audio della Box Raspberry.

Figure 4.15: Mixing View



In particolare gli aspetti configurabili sono i seguenti che vengono descritti in riferimento alla figura:

1. Master Gain / Master Volume: consente la modulazione del volume complessivo della Box, questo volume condiziona il volume di tutti i Peer con cui l'utente è connesso
2. Peer Gain / Peer Volume : consente la modulazione del volume dell'audio proveniente dal singolo Peer.
3. Input - Output Modulator: consente la modulazione degli input di ogni Peer, per indicare alla Box come l'utente desidera ascoltare il flusso audio proveniente da quell'input, vale a dire se si desidera un ascolto uniforme che divida l'input stereo in maniera paritaria tra i due canali di output (cuffia destra e cuffia sinistra) oppure se si desidera una distribuzione differente.
4. Input Gain : consente di modulare il Volume del singolo canale di input

## CHAPTER V

### Conclusioni

Le problematiche sorte negli ultimi anni, hanno portato ad una estensione dei servizi di interazione remota che sarebbe stata impensabile fino a poco tempo fa. La comunicazione mediante strumenti online che si é imposta come soluzione alla necessità di mantenere le distanze tra gli uomini ha permesso all'umanità di conoscere la flessibilità di una comunicazione efficiente a distanza. Ovunque sorgono problematiche sorgono infatti anche stimoli e opportunità di miglioramento.

Lo sviluppo di questo progetto ha rappresentato una grossa opportunità per poter apprezzare quanto il Web possa essere uno strumento potente non solo per distribuire dei servizi ma per incoraggiare realmente le persone alla crescita personale. L'implementazione di un sistema di comunicazione innovativo tra studenti di musica su un'applicazione online ha come sfondo la volontà di dimostrare come la didattica affermata tramite strumenti di remotizzazione rappresenti non un limite, ma un'opportunità per abbattere ulteriormente le barriere di apprendimento che si pongono tra gli studenti di tutto il mondo e l'accesso ad un'educazione di prima qualità.

La potenza del web ci costringe inoltre ad altre osservazioni pratiche per quel che riguarda la disponibilità di uno strumento così pervasivo e rivoluzionario. Il mondo dell'utenza dei servizi web é illimitato in varietà e caratteristiche. Sulla base di questa consapevolezza, gli sviluppatori devono concentrarsi sul rendere i servizi implementati accessibili alla più ampia varietà di utenti, sia quelli avvezzi all'uso di questo tipo di tecnologia, sia quelli meno pratici, sia quelli che interagiscono mediante

gli strumenti piú diffusi sia i meno avvantaggiati costretti ad utilizzare tecnologie alternative. Sviluppare un'applicazione web mantenendo il focus sulla sua usabilitá e accessibilitá non é stata un'impresa scontata, anzi rappresenta tuttora una sfida. É necessario fornire un'interfaccia che risulti comprensibile anche dagli utenti che non sono esperti del contesto applicativo e che nello stesso tempo sia di rapido approccio per i piú navigati. Per garantirne l'accessibilitá é necessario mettersi nei panni degli utenti che interagiscono mediante strumenti alternativi quali Screen Reader, tastiere o altre tecnologie, ciò comporta l'assunzione di pratiche di programmazione del tutto nuove rispetto a quelle a cui lo sviluppatore comune é abituato, quali l'utilizzo degli attributi ARIA e una seria riconsiderazione di tutta la struttura della pagina web per facilitarne la lettura e navigabilitá.

Testare l'effetto dell'applicazione di queste pratiche é un compito ancora meno scontato, poiché comporta l'utilizzo delle tecnologie assistive in tutti gli scenari possibili all'interno dell'applicazione per verificare che le funzionalitá del sistema siano sempre facili da fruire e, nel caso di navigazione mediante Screen Reader, che la navigazione non sia mai confusionaria e i contenuti sempre correttamente definiti e annunciati.

Approcciarsi per la prima volta a queste pratiche é ostico ma permette di raggiungere una consapevolezza molto maggiore di quello che é il web, uno strumento che diamo ormai quasi per scontato grazie alla sua diffusione e semplicitá di accesso quando invece é proprio la sua universalitá che comporta l'esigenza di osservarlo usando la prospettiva di varie categorie di utenza.

Si tratta di uno sforzo non trascurabile che si aggiunge a quello giá considerevole di sviluppo dell'applicazione, ma che nessuno sviluppatore puó piú permettersi di trascurare in un'era in cui Internet pervade in tutti i servizi.

## References

MDN (2022a). ARIA - Accessibility | MDN. <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA>.

MDN (2022b). tabindex - HTML: HyperText Markup Language | MDN. [https://developer.mozilla.org/en-US/docs/Web/HTML/Global\\_attributes/tabindex](https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes/tabindex).

Tailwind (2022). Tailwind CSS. <https://v2.tailwindcss.com/docs/>.

Vue. Vuejs documentation. <https://vuejs.org/>.

VueRouter. Vue router documentation. <https://router.vuejs.org/guide/>.

Vuex. Vuex documentation. <https://vuex.vuejs.org/>.

W3C (2022). Accessible Rich Internet Applications (WAI-ARIA) 1.3. <https://w3c.github.io/aria/>.

## ACKNOWLEDGMENTS

E' un dovere e un onore per me dedicare una pagina di questo elaborato alle persone che hanno reso possibile questo percorso e senza il supporto delle quali non sarebbe stato possibile arrivarne alla conclusione.

Ringrazio innanzitutto il mio relatore, il professore Antonio Servetti per aver creduto in me nella realizzazione di questo progetto e per la sua disponibilità nel darmi le indicazioni necessarie nei momenti in cui ne avevo bisogno.

Ringrazio il mio collega, il dottor Matteo Sacchetto, principale responsabile dell'intero progetto, per avermi seguito costantemente nel corso del mio lavoro, con grande pazienza e disponibilità. Lavorare con te é stato come lavorare con un amico e molto di ciò che ho imparato da questo progetto lo devo a te e alle nozioni che mi hai trasferito.

Ringrazio la mia famiglia che mi ha sostenuto in ogni momento e che ha sofferto e esultato con me per i sacrifici e le gioie che questo viaggio ha portato nella mia vita. Senza di voi, tutto questo non sarebbe stato possibile, questo traguardo é mio quanto vostro, abbiate orgoglio come ne ho io.

Acknowledgments reflect the views of the author and are not endorsed by committee members or Politecnico di Torino.

VITA

DARIO MONTELLA

Candidato alla Laurea di

DOTTORE IN INGEGNERIA INFORMATICA

Thesis: IMPLEMENTAZIONE DI UN'INTERFACCIA WEB ACCESSIBILE PER  
UNA SCUOLA FORMATIVA MUSICALE ONLINE

Major Field: Computer Science

Biographical:

Education:

Consegue la Laurea Triennale in Ingegneria Informatica presso l'Università  
degli Studi del Sannio il 27 Giugno 2019

Experience:

Lavora presso Bi Elle azienda di consulenza informatica