

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA

Corso di Laurea in Informatica per il Management

CSDashboardTV: progettazione ed implementazione di  
un'applicazione Web per pubblicare contenuti su una  
rete di Smart TV

Relatore:

**Prof.**

**Angelo Di Iorio**

Presentata da:

**Riccardo Ardizzoni**

**Sessione I**

**Anno Accademico 2023/2024**



# Indice

Introduzione .....	5
1 Descrizione del contesto .....	7
2 Stato dell'arte: applicativi per Smart TV .....	12
2.1 React Tv Player.....	12
2.2 SmartTV Plus.....	14
2.3 Slider TV.....	16
2.4 Smart tv player.....	18
2.5 Google Maps Samsung tv .....	19
2.6 PhotoView Samsung Smart tv .....	20
3 Descrizione dell'applicazione.....	23
3.1 Dashboard Tv App.....	23
3.1.1 Pagina di Login .....	23
3.1.2 Pagina delle dashboard .....	24
3.1.3 Dashboard principale .....	26
3.2 Database Manager.....	29
4 Dettagli implementativi .....	34
4.1 Gestione della grafica .....	34
4.2 Funzioni JavaScript.....	35
4.3 Struttura del progetto .....	38
4.4 Descrizione del database.....	41
4.5 Punti di criticità.....	43
5 Conclusioni .....	45
5.1 Campi di applicazione .....	46
5.2 Feedback aziendali.....	46
Sitografia.....	48



## Introduzione

Questa tesi tratta la realizzazione di un'applicazione per smart tv, progettata al fine di poter visualizzare contenuti multimediali all'interno della sede principale di un'azienda e delle sue filiali. L'azienda in questione è Centro Software S.p.A. e verrà descritta nel prossimo capitolo.

Per garantire la fruizione su più televisioni, anche in diverse località, è stato necessario realizzare una dashboard per ogni tv adibita alla visualizzazione dei contenuti. L'applicazione da installare però rimane una, perciò ho dovuto realizzare un sistema che permetta all'utente di scegliere quale dashboard visualizzare.

I contenuti visualizzabili sono salvati all'interno di un dominio web, sono gestiti attraverso un database che permette il collegamento tra essi e le dashboard. Ogni dashboard ha specifici contenuti non esclusivi perciò possono essere visualizzati su più dashboard.

Ogni utente che utilizza questa applicazione dispone di una password, richiesta in fase di avvio. Ad ogni utente vengono associate determinate dashboard da poter visualizzare dopo aver inserito la password. L'utente può sfogliare quelle disponibili e attraverso il click del telecomando ha la possibilità di scegliere quella che desidera visualizzare.

Questa applicazione è pensata per essere utilizzata impiegando il minimo sforzo, perciò, sono presenti dei controlli che hanno il compito di eseguire diverse operazioni in automatico, al verificarsi di determinate condizioni. Tali funzioni verranno meglio approfondite nei capitoli successivi.

Struttura della tesi:

- Il primo capitolo verte sul contesto in cui è stata creata l'applicazione, in primo luogo viene descritta l'azienda per la quale è stata sviluppata e le sue esigenze. Successivamente viene approfondita la struttura dell'applicazione e per quali utenti è stata pensata.

- Il secondo capitolo mette in evidenza lo stato dell'arte relativo ad applicazioni già esistenti che ho analizzato nelle fasi iniziali dello sviluppo.
- Il terzo capitolo offre una visione completa del progetto spiegando le funzionalità e le modalità d'uso dell'applicazione.
- Il quarto capitolo descrive in maniera dettagliata il codice e la struttura del progetto, evidenziando le soluzioni ai problemi riscontrati durante lo sviluppo.
- Il quinto ed ultimo capitolo raggruppa le valutazioni finali ricevute e i test effettuati sui dispositivi reali.

## Capitolo 1

### Descrizione del contesto

La realizzazione di questo progetto è iniziata durante il tirocinio curricolare ed è proseguita individualmente una volta terminato. Il soggetto ospitante è stato Centro Software S.p.A. ([www.centrosoftware.com](http://www.centrosoftware.com)), azienda che si occupa di sviluppare soluzioni digitali per la gestione delle imprese, in particolare è specializzata nell'ambito ERP ed offre software tra i più avanzati nel mercato italiano.

Centro Software è in continua crescita da oltre 30 anni, il campus della sede si sviluppa su una superficie di 11.000 mq, in cui si trovano uffici ma anche aree verdi. Dispone di quattro settori, io sono stato inserito nel settore marketing.

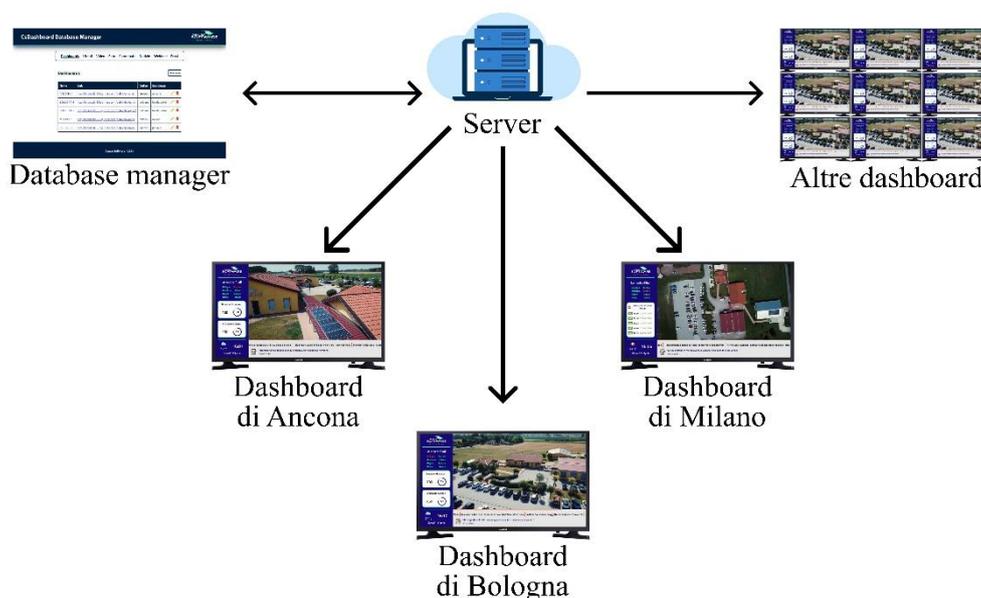
Quello che ho potuto notare è un ambiente fortemente collaborativo nel quale ognuno ha voglia di apprendere e potenziare nuove capacità quotidianamente.

Centro Software, oltre alla sede che si trova in provincia di Bologna, dispone di filiali in tutto il territorio italiano: Milano, Mantova, Trento, Padova, Ancona, Roma, Napoli, ma anche di una fitta rete di partner e centri di assistenza e consulenza a cui si possono affidare i clienti.

L'applicazione che ho realizzato si colloca perfettamente in questo contesto, infatti, non deve essere utilizzata unicamente nella televisione della sede principale ma anche nelle televisioni delle filiali. Proprio per questo motivo ho potuto realizzare un sistema di gestione per più dashboard con contenuti dedicati ad ognuna di esse e utenti che hanno la possibilità di visualizzarle. Inoltre, l'applicazione dispone di una piattaforma che permette di inserire e modificare i dati riguardanti le dashboard che può essere utilizzata dal computer, da qui in avanti tale piattaforma verrà chiamata "database manager".

Le dashboard e tutti i contenuti a loro dedicati sono salvati all'interno di un server principale verso il quale vengono effettuate le richieste per visualizzare la dashboard scelta e per modificare i dati attraverso il database manager. Ogni filiale che utilizza l'applicazione sulla smart tv rappresenta un client che si connette al server e ottiene i dati necessari alla visualizzazione corretta della dashboard. Allo stesso modo anche l'utente amministratore rappresenta un client nel momento in cui effettua modifiche al database attraverso il database manager, in questo caso, oltre ad ottenere i dati dal server, può anche inviarli.

La figura seguente rappresenta la realtà appena descritta, nella quale si trova un server principale e tutti i client che si possono connettere.



*Figura 1.1: struttura dell'applicazione*

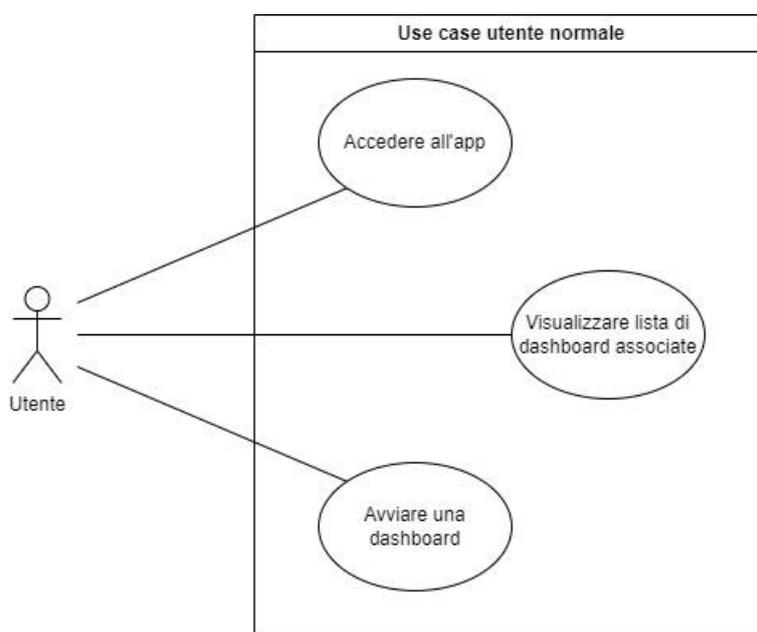
Come accennato in precedenza, questa applicazione prevede la presenza di due tipologie di utenti:

- Utenti normali
- Utenti amministratori

Gli utenti normali sono rappresentati da coloro che utilizzano l'applicazione sulla smart tv e che quindi hanno la sola possibilità di visualizzare passivamente le dashboard a loro dedicate e avviarle. Ad ogni utente normale viene associato un codice che, in fase di login, permette a questa tipologia di utenti di poter visualizzare solamente le dashboard che gli amministratori hanno associato loro.

Gli utenti amministratori sono coloro che hanno la possibilità di accedere al database manager tramite un computer e modificare i dati relativi a tutte le dashboard, infatti, non possiedono codici personali come nel caso degli utenti normali.

Tramite le seguenti immagini vengono mostrati i diagrammi dei casi d'uso degli utenti normali e degli utenti amministratori, in modo tale da evidenziare le operazioni principali che queste tipologie di utenti possono effettuare.



*Figura 1.2: diagramma dei casi d'uso dell'utente normale*



Figura 1.3: diagramma dei casi d'uso dell'utente amministratore

All'interno dell'azienda è già presente una televisione che visualizza alcuni dati riguardanti la sede, ad esempio dipendenti presenti e chiamate effettuate. Tale televisione, però è collegata ad un computer ed ogni giorno per visualizzare i contenuti risulta necessario avviare il computer e l'applicazione che li mostra. Il mio compito è stato quello di creare un'app appositamente per smart tv, che permetta la visualizzazione senza necessariamente collegare un pc alla tv. Un ulteriore obiettivo che questo progetto vuole realizzare è quello di sviluppare un ambiente di facile utilizzo per l'utente che ogni giorno dovrà avviare l'app ma anche

per chi si occuperà delle modifiche al database per mostrare contenuti diversi o per creare nuove dashboard.

Inizialmente ho intrapreso una fase di ricerca al fine di capire come sviluppare applicazioni per smart tv, non avendone mai creata una. Ho capito sin dai primi momenti che le tecnologie principali da utilizzare sono HTML, CSS e JavaScript. Nella scrittura del codice è possibile utilizzare qualsiasi IDE e io ho scelto Visual Studio Code ma per testare il funzionamento dell'app sulla televisione reale o anche su un simulatore è necessario utilizzare Tizen Studio.

Una volta installato Tizen Studio ho iniziato a sviluppare il progetto, inizialmente ho creato una dashboard da utilizzare come base per le successive. Una volta terminata la visualizzazione della prima dashboard mi sono concentrato sullo sviluppo del backend.

Lo sviluppo di applicazioni per smart tv ha sollevato diverse problematiche che verranno approfondite successivamente. A causa di tali difficoltà ho impiegato gran parte del mio tempo a cercare soluzioni online che risolvessero i problemi che ho riscontrato durante il processo di sviluppo.

Il risultato delle mie ricerche mi ha permesso di risolvere le criticità incontrate ma la maggior parte delle volte portava a soluzioni errate. Essendo lo sviluppo per televisioni non molto popolare, spesso, nei forum online o nei siti adibiti alla scrittura di codice software, ho trovato domande senza risposta oppure, nei casi in cui era presente, non è risultata utile al conseguimento del mio scopo.

Tuttavia, ho potuto visualizzare e comprendere alcune applicazioni già esistenti che verranno approfondite di seguito.

## Capitolo 2

### Stato dell'arte: applicativi per Smart TV

Durante la fase di ricerca iniziale ed intermedia ho trovato sul web diversi progetti realizzati da altre persone attraverso i quali ho potuto ottenere informazioni importanti per lo sviluppo dell'applicazione. Tali progetti mi sono risultati utili al fine di capire come sono stati strutturati e quali sono le metodologie di implementazione per applicazioni del genere. Tuttavia, essendo lo sviluppo di applicazioni per televisioni poco popolare, non ho trovato parti di codice realmente riutilizzabili in quanto tali progetti non risolvevano completamente ciò che la mia applicazione avrebbe dovuto compiere.

Di seguito sono elencati e descritti i progetti che ho individuato e analizzato durante la ricerca.

#### 2.1 React Tv Player

Questa applicazione permette di visualizzare contenuti multimediali sulla smart tv. Dispone di un'interfaccia utente moderna che favorisce l'interazione con l'utente, infatti, è possibile avviare e fermare il video, abilitare la riproduzione automatica, mettere "mi piace" al video, modificare il livello del volume e modificare la modalità di visualizzazione.

I video possono essere visualizzati in tre modi:

- partendo da un percorso locale
- tramite formati di streaming HLS e Dash
- tramite collegamenti alle principali piattaforme di visualizzazione di video presenti online (YouTube, Vimeo, SoundCloud, Twitch)

Altri punti di forza di questa applicazione sono la versatilità e la navigazione tra i bottoni. I pulsanti dell'interfaccia sono personalizzabili, così come i metadati del video (titolo e immagini di anteprima) in modo tale da soddisfare qualsiasi esigenza. La navigazione prevede l'utilizzo del telecomando, quindi, gestisce il click dei tasti freccia rendendo l'esperienza utente fluida e intuitiva.

Questo progetto è sicuramente innovativo e moderno, presenta molti punti di forza che lo rendono ottimo ma tali punti non sarebbero stati utili al raggiungimento dei miei obiettivi, primo fra tutti l'interazione con l'utente attraverso i bottoni: nella mia applicazione non devono essere presenti bottoni per interagire con i video ma devono avviarsi in maniera automatica. Inoltre, i video devono essere visualizzabili da un url ad una risorsa online mentre in questo progetto sono visualizzabili anche da locale e in streaming, sicuramente è una metodologia innovativa ma inutile al mio scopo.

Per quanto riguarda la compatibilità, questa applicazione presenta un ampio supporto di dispositivi: Amazon Fire TV, LG webOS, Samsung Tizen Smart TV ma anche console Xbox e browser Web desktop. Questa applicazione è visibile nella maggior parte di dispositivi a grande schermo che eseguono web app con browser Chromium successivi al 2017.

Il progetto è visibile al seguente link:

<https://github.com/lewhunt/react-tv-player>

Di seguito viene mostrata un'immagine dell'applicazione descritta.

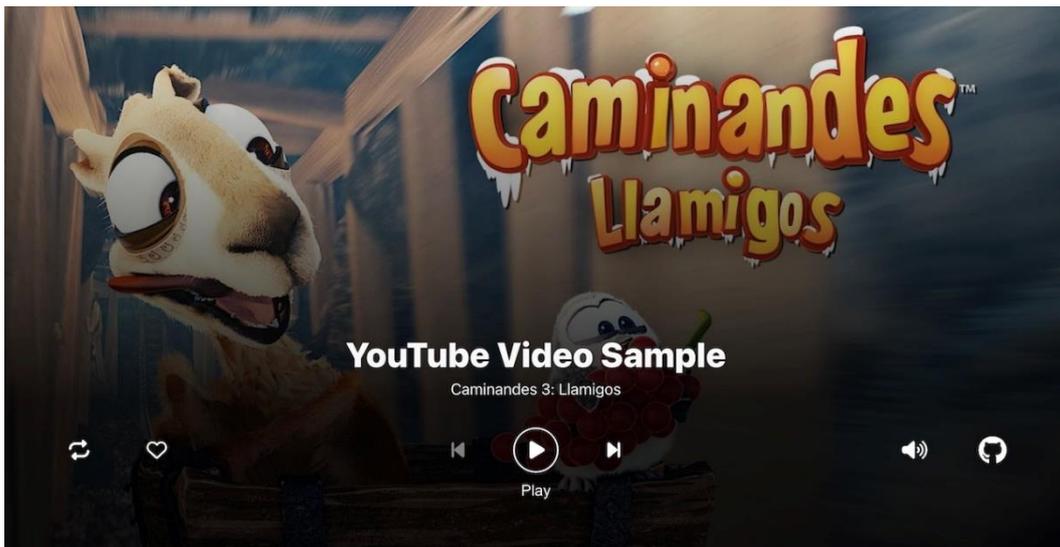


Figura 2.1: demo React Tv Player

## 2.2 SmartTV Plus

Applicazione che mostra un salvaschermo per televisioni di facciata (per esempio nelle camere degli hotel) con tre elementi principali:

- sfondo
- widget della data e dell'orario
- widget del meteo

Lo sfondo mostra l'immagine presente nei computer con sistema operativo MacOS Catalina e cambia il colore in base all'orario corrente.

Il widget della data viene aggiornato ogni minuto, mostra la data con il giorno della settimana e il mese scritti in lettere mentre il numero del giorno e l'orario sono scritti in cifre.

Il widget del meteo mostra il nome della città nella quale viene calcolato, un'icona che descriva la situazione metereologica corrente, i gradi della temperatura esatti e

percepiti e il grado di umidità presente. Per calcolare i valori del meteo, questo progetto utilizza la libreria Open Weather Map e aggiorna i dati ogni dieci minuti.

Il punto di forza principale di questa applicazione è la somiglianza con i widget che avrei dovuto implementare, infatti, la data e il meteo sono molto simili a quelli della mia applicazione e questo mi ha permesso di trarne ispirazione.

Questo progetto non è stato riutilizzato interamente perché presenta uno sfondo che non avrei dovuto implementare, inoltre alcuni dati dell'orario e del meteo non sono rilevanti al raggiungimento dei miei scopi come l'anno corrente e l'indice di umidità.

Questa applicazione è compatibile con la maggior parte dei dispositivi, non presenta particolari requisiti per la sua visualizzazione.

Il progetto è visibile al seguente link:

<https://github.com/Raavtube/SmartTV>

Di seguito viene mostrata un'immagine dell'applicazione descritta.



*Figura 2.2: demo SmartTV Plus*

## 2.3 Slider TV

Questo progetto si occupa di sviluppare un plugin di facile utilizzo per implementare un carosello che controlla qualsiasi contenuto basato su HTML in ordine orizzontale o verticale.

Attraverso questo plugin è possibile navigare tra i contenuti dello slider con le frecce del telecomando e la messa a fuoco del contenuto selezionato è automatica.

Il punto di forza di tale progetto è la facilità di utilizzo e di personalizzazione, infatti, è possibile modificare a proprio piacimento l'intero plugin o solamente una delle parti che lo compongono. Tra i componenti personalizzabili troviamo:

- la direzione dello slider: verticale o orizzontale
- la velocità e il tipo di animazione
- gli indicatori che mostrano quanti elementi sono presenti nello slider e quale è attualmente selezionato

Il punto di debolezza principale riguarda lo scopo del progetto, è utile per realizzare applicazioni inerenti al cinema che necessitano di un carosello per poter scegliere il film da guardare ma non è utile al mio scopo. All'interno del mio progetto non è necessario inserire uno slider di scelta di contenuti, perciò, ho deciso di non riutilizzare questo plugin.

Per quanto riguarda la compatibilità è possibile utilizzare questo plugin sulla maggior parte dei dispositivi disponibili, inoltre il supporto per le frecce di navigazione del telecomando è compatibile anche con LG Magic Control.

Il progetto è visibile al seguente link:

<https://github.com/gibbok/jquery-slidertv>

Di seguito vengono mostrate due immagini dell'applicazione descritta.

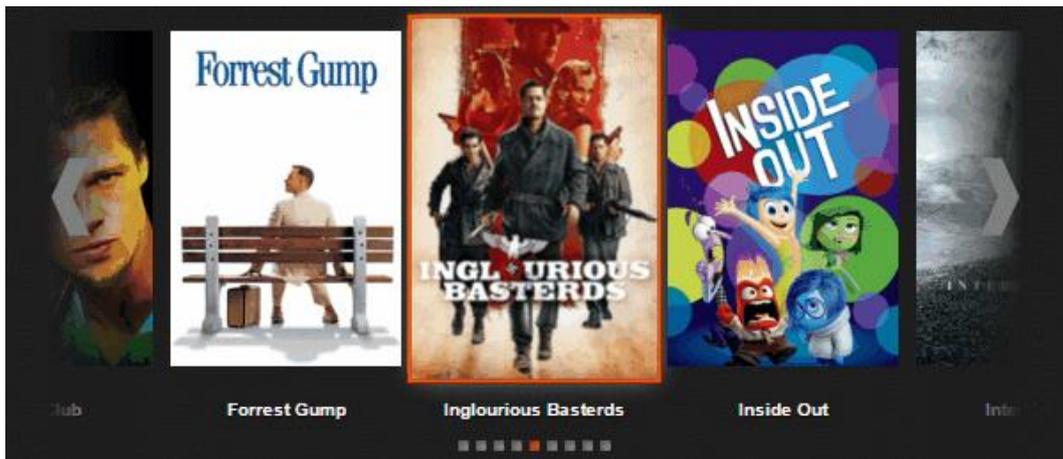


Figura 2.3: demo Slider TV orizzontale



Figura 2.4: demo Slider TV verticale

## 2.4 Smart tv player

Questa applicazione permette di visualizzare un video direttamente sulla televisione tramite url, è possibile dunque visualizzare un video salvato nel pc ma anche presente sul web.

In particolare, è utile per la visione di film, infatti, i dati da mostrare vengono ottenuti tramite lettura di un file json nel quale devono essere inserite tutte le informazioni relative al film. Tali informazioni possono essere visualizzate dall'utente nella pagina principale.

Il punto di forza di questo progetto è la possibilità di visualizzare qualsiasi video, sia locale sia dal web.

Tuttavia, questo progetto presenta diversi punti di debolezza, tra questi il fatto che tale applicazione è stata sviluppata una decina di anni fa e le metodologie di visualizzazione dei contenuti, ad oggi, sono cambiate; inoltre è pensata per visualizzare film e presenta molte informazioni che non risultano utili al raggiungimento degli obiettivi. In ultimo, il codice relativo alla visualizzazione dei video risulta poco chiaro e molto rigido, dato che per utilizzarlo nella mia applicazione avrei dovuto modificarlo, non sarebbe stato un vantaggio bensì un ostacolo.

Questa applicazione è compatibile con tutti i dispositivi, non presenta requisiti particolari. Tuttavia, non è possibile visualizzare contenuti salvati in locale attraverso il percorso del file system sui dispositivi di marca Samsung, in questi casi è necessario utilizzare un url http (inserendo il contenuto all'interno di un web server).

Il progetto è visibile al seguente link:

<https://github.com/giang-nghg/smart-tv-player>

Di seguito viene mostrata un'immagine dell'applicazione descritta.

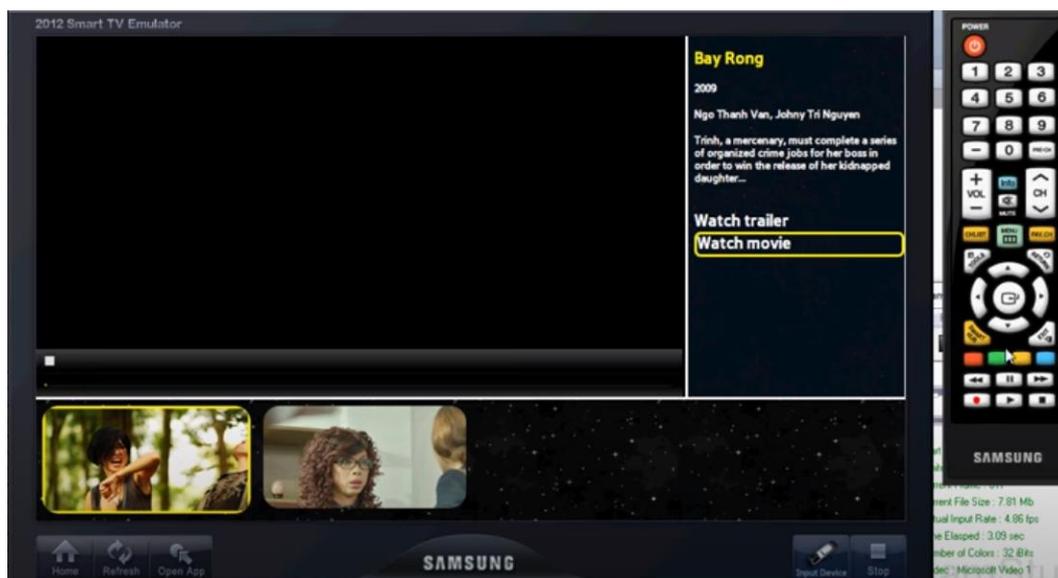


Figura 2.5: demo Smart tv Player

## 2.5 Google Maps Samsung tv

Applicazione che permette all'utente di visualizzare ed interagire con la mappa di Google. Presenta i comandi di zoom e alcuni bottoni per mostrare la mappa in versione normale o in versione satellite. Indica anche le coordinate del punto centrale che si sta guardando.

Tra i punti di forza di questo progetto troviamo la possibilità di visualizzare la mappa in maniera molto semplice, inoltre il codice è ben strutturato ed è possibile modificarlo in maniera adeguata.

Il punto di debolezza principale è l'utilizzo della mappa di Google. Proprio come il progetto sopra citato, anche in questo caso risulta essere stato sviluppato più di dieci anni fa e nel frattempo l'utilizzo della mappa fornita da Google è cambiato e questo porta a non poter visualizzare correttamente la mappa su un apparecchio televisivo.

Per questo motivo ho optato di non utilizzare questo codice, inoltre sono presenti vari comandi che nella mia applicazione non sarebbero serviti. Sarebbero stati utili

invece dei movimenti automatici nella mappa che in questa applicazione non sono presenti.

Per quanto riguarda la compatibilità, è possibile utilizzare questa applicazione solamente su dispositivi Samsung. Infatti, è stata creata per approfondire le conoscenze riguardanti il mondo dello sviluppo di applicazioni per smart tv Samsung. Inoltre, per installarla è necessario aver scaricato nel proprio computer il Software Development Kit (SDK) di Samsung.

Il progetto è visibile al seguente link:

[https://github.com/amberbit/google\\_maps\\_samsung\\_tv](https://github.com/amberbit/google_maps_samsung_tv)

Di seguito viene mostrata un'immagine dell'applicazione descritta.

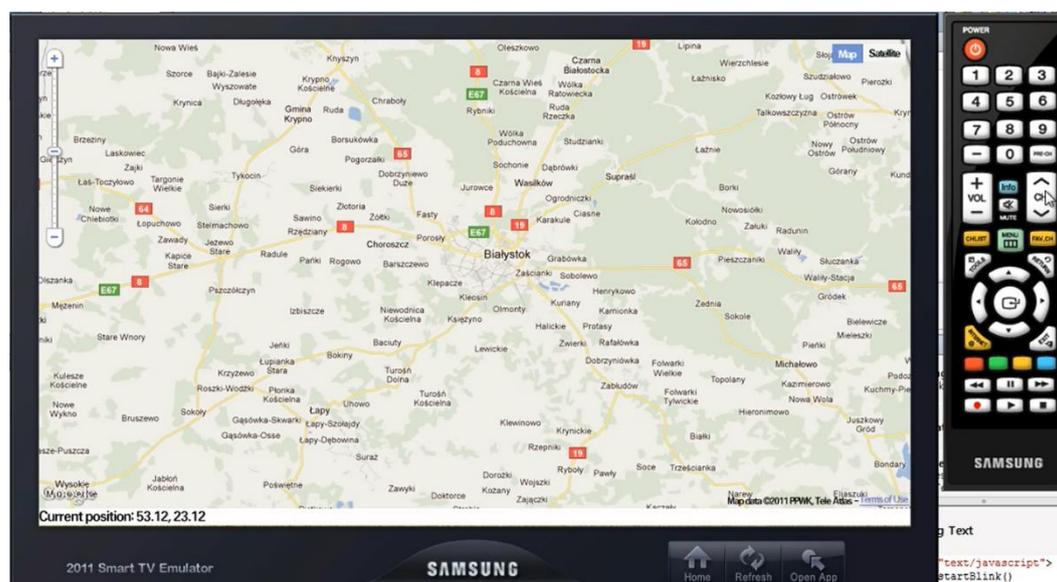


Figura 2.6: demo Google Maps Samsung tv

## 2.6 PhotoView Samsung Smart tv

Applicazione che mostra due immagini e permette all'utente di navigare tra queste. Come punto di forza vi è il fatto di poter cambiare immagine e quindi non ne mostra soltanto una, tuttavia, presenta troppi punti di debolezza che mi hanno

portato ad escludere questa soluzione. In primo luogo, mostra solo due immagini e quindi sarebbe stata necessaria una modifica, inoltre l'immagine viene cambiata solo dopo il click del bottone e nella mia applicazione il cambio sarebbe dovuto essere automatico, dopo un certo periodo di tempo.

Questa applicazione è compatibile con qualsiasi dispositivo, essendo molto semplice non presenta punti critici che potrebbero bloccare la visualizzazione e per questo non presenta alcun tipo di requisito per poter funzionare.

Il progetto è visibile al seguente link:

[https://github.com/hoomi/PhotoView\\_SamsungSmartTv](https://github.com/hoomi/PhotoView_SamsungSmartTv)

Alla luce di quanto ho potuto osservare questi erano i progetti che più si avvicinavano alle mie esigenze, pur essendo molto lontani da ciò che dovevo realizzare. Per questo motivo ho deciso di sviluppare interamente un'applicazione nuova che potesse soddisfare a pieno ciò che doveva essere creato ovvero un'applicazione nella quale si potessero visualizzare tutti i contenuti visti finora nei progetti precedenti ma in maniera automatica, senza la necessità, per l'utente, di interagire con il dispositivo se non per effettuare il login.

Nella seguente tabella vengono mostrate le principali funzionalità di ogni applicazione descritta in precedenza, confrontate con quelle dell'applicazione da me realizzata. In particolare, è possibile notare come nessuna di esse svolga in maniera automatica le operazioni, requisito fondamentale di quanto richiestomi. Inoltre, nessuna delle applicazioni presenta la possibilità di visualizzare tutti i contenuti richiesti e alcune di esse non sono neanche compatibili con i dispositivi, essendo state realizzate circa dieci anni fa.

Legenda della tabella:

- **Video, foto, mappa, slider contenuti, meteo, data e ora:** la 'X' indica la possibilità di visualizzare il contenuto o meno.

- **Interazione:** indica la possibilità per l'utente di interagire con l'applicazione, in particolare, l'applicazione presenta dei comandi che permettono all'utente di utilizzarla più facilmente.
- **Modernità:** indica se l'applicazione presenta un'interfaccia utente moderna e se è stata realizzata negli ultimi cinque anni.
- **Personalizzazione:** indica la possibilità, per gli sviluppatori, di apportare modifiche all'applicazione in modo da aggiungere o rimuovere funzionalità
- **Compatibilità:** indica se l'applicazione è compatibile con i principali sistemi operativi delle moderne smart tv.
- **Autonomia:** indica se l'applicazione è in grado di funzionare senza la necessità da parte dell'utente di interagire attraverso i comandi. In questo caso nessuna delle applicazioni presenta questa caratteristica, tutte necessitano dei comandi dell'utente che la utilizza.

Funzionalità	React Tv Player	Smart Tv Plus	Slider TV	Smart tv player	Google Maps Samsung tv	PhotoView Samsung Smart tv
Video	X			X		
Foto						X
Mappa					X	
Slider contenuti			X			X
Meteo		X				
Data e ora		X				
Interazione	X		X		X	X
Modernità	X					
Personalizzazione	X		X			
Compatibilità	X	X	X			X
Autonomia						

Tabella 2.1: funzionalità delle applicazioni a confronto

## Capitolo 3

### Descrizione dell'applicazione

L'obiettivo principale di questa applicazione è quello di mostrare diversi contenuti multimediali in maniera automatica e continua nel tempo infatti, è stata progettata per essere avviata la mattina, all'apertura dell'azienda, ed essere spenta nel tardo pomeriggio durante la chiusura degli uffici.

L'applicazione è composta da due parti principali: la prima comprende le dashboard che possono essere visualizzate sulle televisioni, la seconda consiste in un'applicazione web che permetta agli amministratori di modificare i contenuti da mostrare e tutti i restanti dati relativi al sistema di dashboard.

#### 3.1 Dashboard Tv App

Il sistema delle dashboard comprende una pagina di login al fine di accedere alla piattaforma ed una pagina che mostra, in base al codice utente, le dashboard che possono essere avviate e visualizzate.

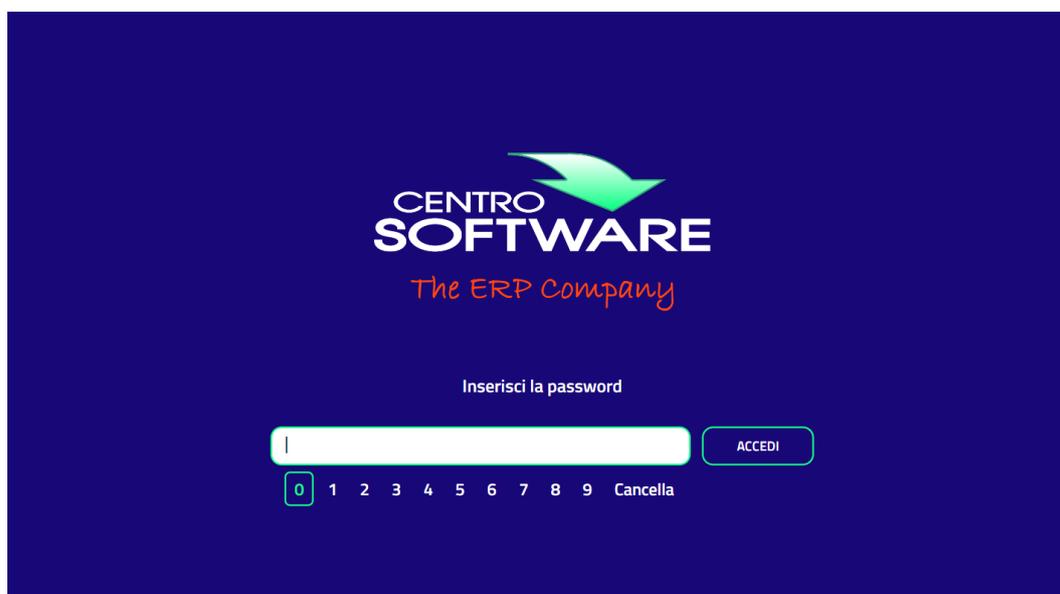
Nei successivi paragrafi verranno analizzati tutti i componenti dell'applicazione.

##### *3.1.1 Pagina di Login*

Al primo avvio viene visualizzata la pagina di login, è possibile interagire con tale pagina attraverso il click del telecomando. Nel caso in cui venga cliccato il tasto "freccia giù" oppure il tasto "ok" viene mostrata la tastiera virtuale attraverso la quale è possibile inserire la password nel campo di testo.

Nel caso in cui la password inserita sia giusta, viene visualizzata la lista di dashboard che è possibile visualizzare e il login viene salvato nella memoria della televisione in modo tale che non sia necessario inserire nuovamente la password per utilizzi futuri. Per tale motivo la pagina di login viene mostrata solamente durante il primo avvio dell'applicazione ma anche nel caso in cui l'utente decida di effettuare il logout per inserire una password diversa e poter visualizzare altre dashboard.

Di seguito viene mostrata la schermata descritta in precedenza: la pagina di login dopo aver aperto la tastiera virtuale.



*Figura 3.1: pagina di login con tastiera virtuale*

### ***3.1.2 Pagina delle dashboard***

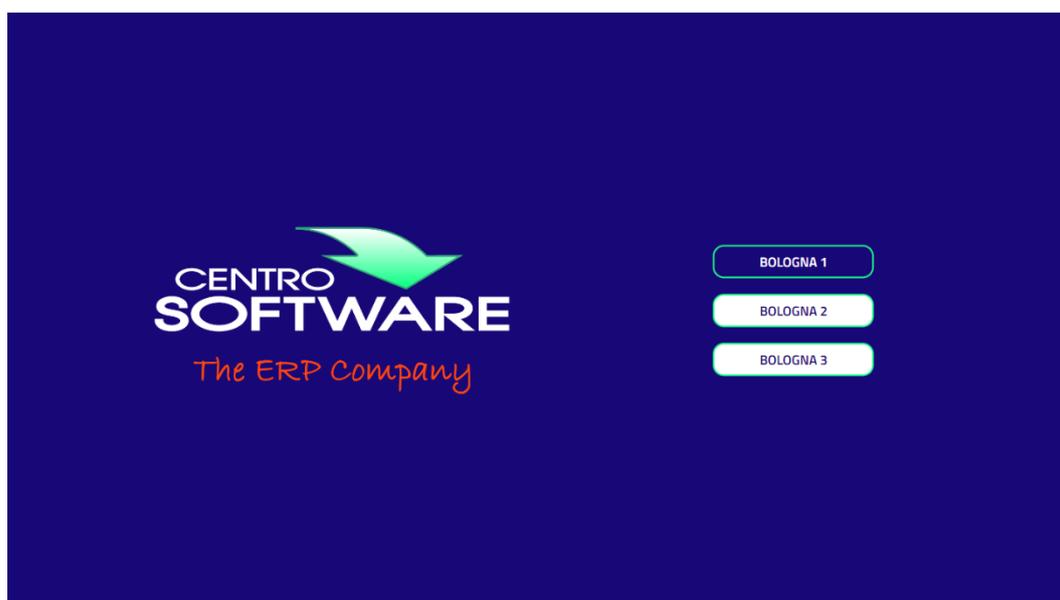
Successivamente all'inserimento della password corretta, l'utente viene reindirizzato verso la pagina delle dashboard. Tale pagina contiene una lista di tutte le dashboard che possono essere avviate dall'utente.

Come accennato in precedenza ogni filiale dell'azienda potrebbe voler mostrare contenuti diversi da quelli delle altre, per questo motivo alle password sono associate determinate dashboard che variano rispetto a quelle disponibili per altri utenti. Inoltre, ogni dashboard potrebbe avere alcune varianti, potrebbe, per esempio, essere creata una variante che mostra contenuti relativi alle festività.

L'utente può navigare tra le dashboard disponibili e avviare la dashboard selezionata. Nel momento in cui l'utente avvia una dashboard, tale scelta viene salvata nella memoria del dispositivo, in modo tale che venga avviata direttamente durante gli avvii futuri dell'applicazione, senza necessità di inserire nuovamente la password e scegliere la dashboard.

Nel caso in cui l'utente voglia aprire una dashboard diversa da quella scelta può ritornare alla lista di dashboard tramite il tasto "back" del telecomando e sceglierne un'altra. Se ne viene chiusa una viene avviato un timer di 30 secondi che, al termine, riapre l'ultima dashboard aperta se non ne viene scelta una diversa, questo per garantire che la schermata non rimanga ferma sulla pagina della lista per un periodo di tempo prolungato se l'utente non riapre nessuna dashboard.

Di seguito viene mostrata la pagina della lista di dashboard.



*Figura 3.2: pagina della lista di dashboard*

### ***3.1.3 Dashboard principale***

Nel momento in cui viene scelta la dashboard da visualizzare l'applicazione avvia la schermata relativa ad essa. Ogni dashboard può essere diversa dalle altre ed ogni filiale può mostrare contenuti diversi, in questo capitolo, però, verrà descritta la dashboard generale ovvero quella utilizzata come base per creare le altre.

La dashboard è l'elemento principale dell'intera applicazione, tutto il resto è stato costruito intorno ad essa. Contiene numerosi elementi dinamici che variano durante la visualizzazione, l'unica interazione prevista per l'utente è la possibilità di ritornare alla pagina della lista delle dashboard tramite il tasto "back", in questo modo è possibile lasciare spazio alle automazioni che permettono all'applicazione di funzionare autonomamente in maniera continuativa.

Gli elementi sono disposti nella schermata utilizzando tre contenitori principali: sidebar, main e footer. Tutti gli elementi della sidebar e del footer sono sempre visibili, mentre quelli del main sono visibili in maniera alternata.

Gli elementi dinamici della sidebar sono:

- Widget dipendenti presenti
- Widget chiamate odierne
- Widget meteo
- Orario e giorno correnti

I widget dei dipendenti presenti e delle chiamate odierne mostrano rispettivamente quanti dipendenti sono presenti in azienda in quel momento e quante chiamate sono state effettuate fino a quel momento.

Il widget meteo mostra quanti sono i gradi di temperatura esterna e attraverso un'icona viene indicata la condizione atmosferica (come: nuvoloso, soleggiato,

nuvoloso con pioggia, etc...). Tale widget indica le condizioni metereologiche della zona in cui si trova l'apparecchio televisivo.

Gli elementi del main sono:

- Slider di video
- Slider di foto
- Mappa degli interventi sul campo
- Pagina prossimi webinar aziendali

All'avvio di ogni contenuto presente nel main viene visualizzata una barra di avanzamento che percorre tutta la schermata ad una velocità variabile, che si modifica in base al periodo di tempo che il contenuto occupa nella schermata.

Lo slider di video è il primo elemento che viene visualizzato e mostra, uno alla volta, i video previsti nella dashboard selezionata. Al termine dei video presenti vengono mostrate le immagini, ogni immagine rimane visibile per un periodo di tempo stabilito di default pari a dieci secondi ma che può essere modificato.

Al termine della visualizzazione delle foto viene mostrata la mappa contenente gli interventi sul campo effettuati, ogni intervento è mostrato nella mappa sotto forma di marker blu. In un primo momento viene visualizzata la mappa con uno zoom elevato, in modo tale da poter vedere la maggior parte dei marker presenti, successivamente la mappa si sposta verso cinque marker scelti in modo casuale, uno alla volta, ed effettua un'animazione che permette di visualizzarli più in dettaglio aumentando il livello di zoom. Anche in questo caso è possibile modificare il periodo di tempo che intercorre tra le visualizzazioni dei marker che di default è sempre pari a quindici secondi.

In seguito, viene visualizzata la pagina che mostra il prossimo webinar aziendale, ne viene indicata la data, il titolo e un codice QR che, se inquadrato, indirizza alla pagina web aziendale relativa al webinar con la possibilità di iscriversi.

Una volta terminata la visualizzazione, i contenuti vengono riavviati dall'inizio in maniera automatica, in questo modo la schermata non rimane bloccata in attesa di un comando da parte dell'utente ma prosegue senza interruzioni.

Gli elementi del footer sono:

- News feed
- Frase del giorno

Le notizie sono ricavate a partire da un link web attraverso il quale si ottiene un feed di notizie aggiornate per quanto riguarda la tecnologia e sono visualizzate mediante uno slider simile a quello dei principali telegiornali nazionali.

La frase del giorno mostra una frase motivazionale ed il suo autore, ad ogni giornata corrisponde una frase diversa che può essere modificata attraverso il database manager.

Di seguito viene mostrata la schermata descritta in precedenza: la dashboard principale.



Figura 3.3: dashboard principale

Per quanto riguarda le dashboard realizzate per la filiale di Milano è necessaria la descrizione di una particolare modifica: all'interno della sidebar non vengono visualizzati i widget dei dipendenti presenti e delle chiamate odierne, al loro posto viene visualizzato un widget che indica gli orari delle prossime partenze della linea metropolitana più vicina alla filiale. Tale widget può risultare utile per i dipendenti oppure per coloro che visitano la filiale e devono usufruire della metropolitana all'uscita, in questo modo possono visualizzare velocemente gli orari sulla televisione e dirigersi alla stazione. Inoltre, è stato inserito solo nelle dashboard della filiale di Milano, in quanto è l'unica con una stazione della metropolitana nelle vicinanze.

Di seguito viene mostrato un esempio del widget appena descritto.



*Figura 3.4: widget della metropolitana*

### 3.2 Database Manager

Come descritto in precedenza, oltre al sistema delle dashboard, ho realizzato anche una piattaforma che permetta agli utenti amministratori di poter modificare i dati relativi alle dashboard in maniera facile e veloce, senza dover necessariamente accedere al database.

Tale piattaforma viene chiamata “Database Manager” e corrisponde ad un'applicazione web alla quale è possibile accedere tramite credenziali anche da un personal computer.

All'interno della piattaforma è presente un menu attraverso il quale è possibile cambiare pagina e modificare altri contenuti. Nello stesso vi sono le seguenti voci:

- Dashboards
- Utenti
- Video
- Foto
- Coordinate
- Notizie
- Webinar
- Frasi

Per ognuna di queste pagine troviamo una tabella all'interno della quale è possibile visualizzare i dati già presenti e, attraverso specifici bottoni, è possibile modificare o eliminare la riga selezionata. Inoltre, è possibile aggiungere nuovi dati attraverso un form specifico.

Per quanto riguarda la pagina “Dashboards” è possibile, oltre alle operazioni descritte in precedenza, visualizzare per ogni dashboard le foto e i video ad essa associati in modo tale da poter aggiungere o eliminare determinati collegamenti.

Infatti, ogni volta che si aggiunge una nuova foto o un nuovo video sulla piattaforma, tale contenuto non è associato a nessuna dashboard, pertanto, non sarà visualizzato finché non viene collegato ad almeno una di esse. Perciò, fino a quel momento sarà solamente presente sulla piattaforma e potrà essere visualizzato in futuro. In questo modo, lo stesso contenuto potrà essere visualizzato su più di una dashboard e un'eventuale modifica del contenuto verrà aggiornata in automatico in

tutte le dashboard al quale è collegato, senza la necessità di modificarle una ad una. Le foto e i video aggiunti in seguito devono essere inseriti nelle pagine corrispondenti mentre i collegamenti tra i contenuti e le dashboard possono essere inseriti e modificati nella pagina “Dashboards”.

La pagina “Utenti” si riferisce agli utenti normali a cui è consentito l’accesso all’applicazione sulla televisione. Il codice corrisponde al codice presente anche nella tabella delle dashboard e indica, per ogni utente, quali sono le dashboard ad esso dedicate. La password corrisponde alla password che l’utente deve inserire all’avvio dell’applicazione e, attraverso essa, il sistema permette all’utente di poter visualizzare solo le dashboard corrispondenti.

Le coordinate si riferiscono alle coordinate utilizzate per ottenere i dati riguardanti il meteo. Ad ogni dashboard è associato lo stesso codice che indica le coordinate corrispondenti. Per facilitare l’inserimento e la modifica delle coordinate è presente una mappa che permette di inserire un segnaposto nella posizione scelta e ottenere le coordinate automaticamente.

Per quanto riguarda la pagina delle notizie è presente solo una voce nella tabella, quella relativa al link del feed rss dal quale ottenere le news. In questa pagina vengono mostrate alcune notizie relative al link per capire immediatamente se le notizie sono aggiornate senza aprire il link.

La pagina relativa al webinar mostra un link, che permette la realizzazione del qr code da visualizzare nell’applicazione, e una lista di webinar nella quale sono presenti il titolo e la data. In base alla data viene mostrato nell’applicazione il prossimo webinar in programma.

Per quanto riguarda la pagina “Frase” nella tabella è presente, oltre alla frase e all’autore, un identificativo numerico che corrisponde alla data del mese nella quale verrà visualizzata.

Di seguito vengono mostrate alcune immagini della piattaforma appena descritta.

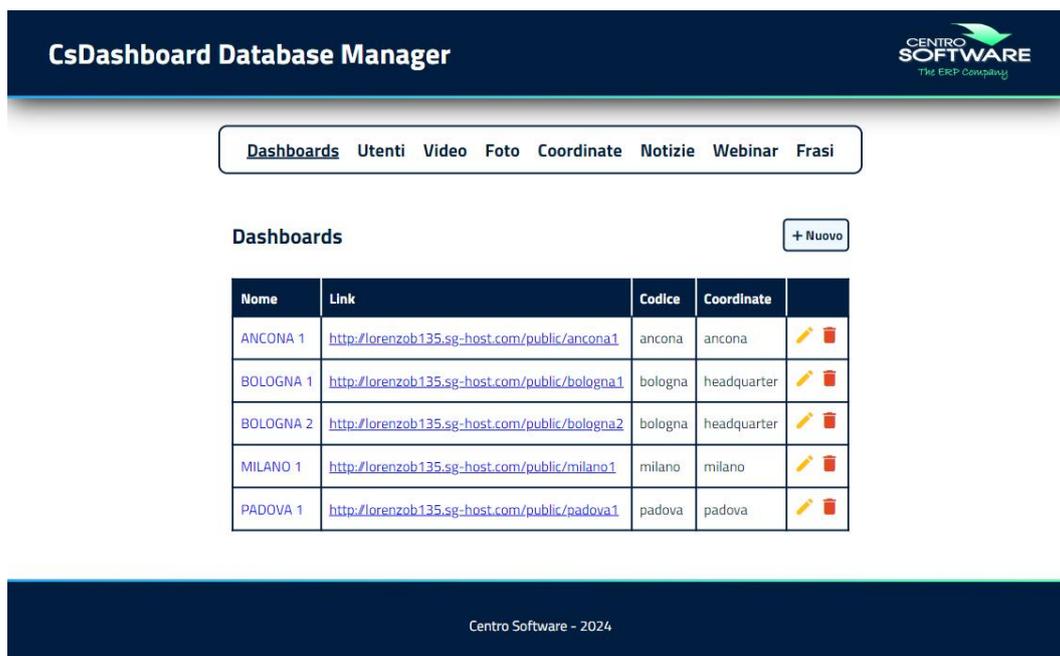


Figura 3.5: Dashboard Manager

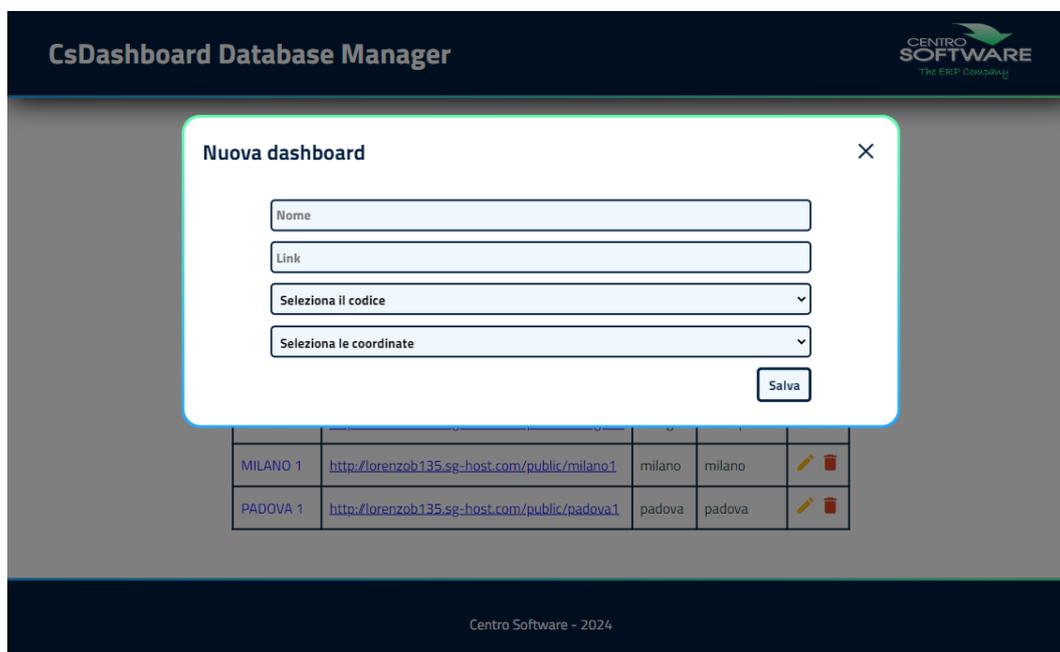


Figura 3.6: inserimento nuova dashboard

La medesima piattaforma è utilizzabile anche da dispositivi mobile, pertanto è stata realizzata in maniera “responsive” al fine di garantire una corretta visualizzazione di tutti i dati in base alla dimensione del dispositivo.

Di seguito viene mostrato un esempio di visualizzazione da un dispositivo mobile.



Figura 3.7: Database Manager mobile

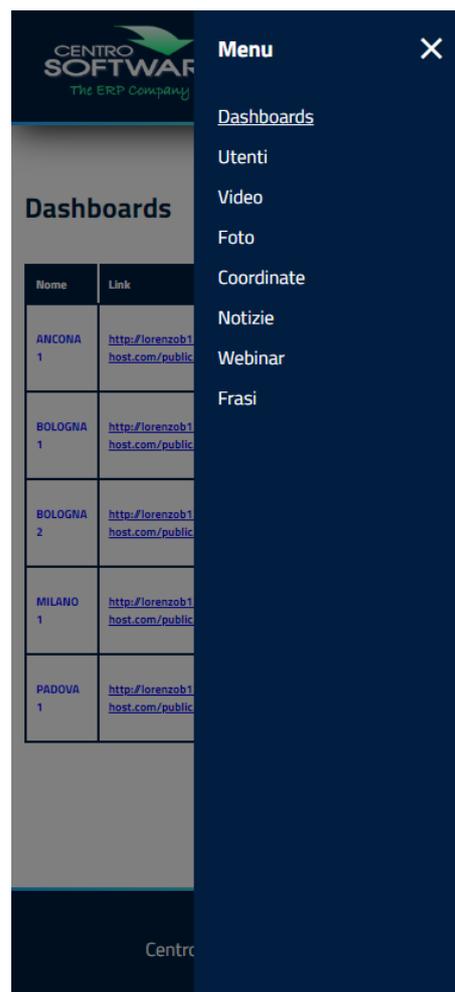


Figura 3.8: menu mobile

## Capitolo 4

### Dettagli implementativi

In questo capitolo viene descritto il processo d'implementazione che ha portato alla realizzazione dell'applicazione dalle fasi iniziali e successivamente verrà fornita una descrizione più dettagliata delle singole componenti a basso livello.

Durante la prima fase ho deciso quale IDE utilizzare per lo sviluppo, la mia scelta è ricaduta su Tizen Studio, che mette a disposizione, oltre all'editor per scrivere il codice, anche un simulatore di dispositivi che nel mio caso è risultato utile al fine di simulare una televisione. Inoltre, permette di collegarsi ad una televisione reale, accedendo alla developer mode della tv tramite indirizzo ip e mandando in esecuzione l'app sulla tv.

I linguaggi di programmazione utilizzati sono:

- HTML, CSS e JavaScript per quanto riguarda la visualizzazione dell'app sulla televisione.
- PHP e SQL per quanto riguarda la piattaforma di gestione del database e al fine di realizzare le api che vengono utilizzate dall'app per ottenere i dati da mostrare.

#### 4.1 Gestione della grafica

Nella fase iniziale mi sono occupato dello sviluppo della dashboard di base e del suo funzionamento su un dispositivo reale e, in particolare, della visualizzazione statica della schermata così da avere un'idea chiara di come sarebbe stata. Il body dell'app si divide in due div principali:

- sidebar, all'interno del quale troviamo altri tre div, uno contenente il logo, uno contenente i widget centrali e un ultimo contenente la data, l'orario e il meteo.
- right-content, diviso a sua volta in:
  - main, al cui interno troviamo un div per ogni contenuto (video, foto, mappa e webinar), la visualizzazione di questi elementi si andrà ad alternare dopo l'avvio.
  - footer, contenente un div per le notizie e un div per la frase del giorno.

Le dimensioni occupate da questi elementi e il loro posizionamento sono gestiti tramite il codice CSS, in particolare tramite l'utilizzo dell'istruzione "display: flex", che permette all'elemento HTML di ridimensionarsi in maniera flessibile in base alle istruzioni date. Grazie a questa operazione è possibile visualizzare l'app in maniera ottimale su dispositivi di dimensioni diverse.

In particolare, al right-content è stata assegnata una larghezza pari all'80% del totale, in questo modo la sidebar si ridimensiona al 20% del totale in maniera automatica.

## 4.2 Funzioni JavaScript

Per garantire che l'applicazione funzioni in maniera automatica senza la necessità di interagire con l'utente ho realizzato delle funzioni in JavaScript che permettono di animare l'app.

Queste funzioni ottengono i dati dal database e, in base al risultato della richiesta, mostrano i contenuti adeguati. Alcune funzioni vengono richiamate dopo un certo periodo di tempo, ad esempio la funzione che mostra il meteo viene richiamata ogni

dieci minuti per mantenere il meteo aggiornato e non sovraccaricare il processore. Allo stesso modo la funzione che aggiorna l'orario viene richiamata ogni minuto.

Per quanto riguarda i contenuti centrali, quindi video, foto, mappa e webinar, sono presenti funzioni che ottengono i dati e aggiungono gli elementi alla pagina HTML ma non vengono visualizzati all'istante in quanto tramite la modifica del parametro "display" del CSS la loro visualizzazione viene rimossa e inserita in base alla precedenza del contenuto.

In particolare, se sono presenti dei video vengono inseriti nel documento e visualizzati subito uno di seguito all'altro. La funzione che li mostra richiama sé stessa fino a quando raggiunge l'ultimo video, in questo caso rimuove la visualizzazione dei video e mostra le foto.

Il medesimo funzionamento possiamo riscontrarlo nella visualizzazione delle foto, in questo caso le foto vengono mostrate una alla volta e ognuna ha un periodo di tempo prestabilito, una volta raggiunta la fine della lista contenente le foto viene visualizzata la mappa.

Per quanto riguarda la mappa, la funzione calcola quanto tempo occuperà la sua visualizzazione in base a quanti marker sono presenti:

- se i marker sono più di cinque ne verranno scelti cinque in maniera casuale quindi il periodo di tempo totale sarà pari a novanta secondi, quindici per la visualizzazione iniziale e quindici per ogni marker.
- se i marker sono meno di cinque viene assegnato un periodo di tempo pari a quindici aggiunto a quindici moltiplicato per il numero di marker presenti.

Al termine del periodo calcolato verrà visualizzata la pagina del webinar che rimarrà per trenta secondi, al termine dei quali verranno visualizzati nuovamente i video e il ciclo ripartirà dall'inizio. Tale sistema permette all'app di restare in funzione per un periodo di tempo illimitato e allo stesso tempo risulta sempre aggiornata in base ai contenuti che ottiene dal database. Inoltre, se alcuni contenuti non risultano presenti, l'app non considera quel determinato elemento per

visualizzare il successivo. Per esempio, non si blocca se non sono presenti delle foto ma procede direttamente alla visualizzazione della mappa.

Anche la visualizzazione della barra di avanzamento viene gestita tramite JavaScript: infatti, ogni volta che si avvia un contenuto viene richiamata la funzione che modifica il valore di larghezza della barra ogni secondo. Con il fine ultimo di permettere alla barra di raggiungere il termine solo quando la visualizzazione del contenuto è conclusa viene calcolata la distanza che deve percorrere ad ogni giro in base alla durata del contenuto. Per esempio, se un video ha una durata di un minuto, la barra avanzerà in maniera più lenta (e quindi percorrerà meno distanza ogni secondo) rispetto alla visualizzazione di una foto che dura dieci secondi.

Per quanto riguarda la visualizzazione delle notizie, ho dovuto realizzare un algoritmo che calcolasse, in base alla larghezza occupata dalle notizie inserite nel documento HTML, quanto tempo dovesse durare l'animazione che le scorre, al fine di ottenere una visualizzazione simile a quella dei telegiornali. Tale algoritmo consente di ottenere un'animazione con una velocità costante qualsiasi sia la quantità di notizie ottenute, pertanto, la visualizzazione sarà la stessa sia per cinque notizie sia per cinquanta e per qualsiasi numero, in quanto i valori sono calcolati in maniera automatica e non sono impostati a priori. Le stesse notizie vengono visualizzate cinque volte in sequenza e al termine del quinto giro vengono aggiornate e la visualizzazione ricomincia.

Per ottenere i dati dal database vengono effettuate delle chiamate AJAX tramite jQuery, specificando l'url attraverso il quale è possibile usufruire del controller PHP in grado di inviare i dati richiesti. Per esempio, per quanto riguarda i video, è possibile ottenere un file json contenente gli url dei video da mostrare in base alla dashboard collegandosi al seguente link:

`http://lorenzob135.sg-host.com/public/show-video/" + APP_NAME`

dove "APP\_NAME" è la costante che indica il nome della dashboard all'interno del file JavaScript. Tale link corrisponde al controller PHP che effettua una chiamata al database, ottiene i risultati e li spedisce tramite un file json.

### 4.3 Struttura del progetto

La struttura del progetto si divide in due parti principali: l'applicazione da scaricare sulla smart tv e una pagina web contenente tutti i file e i dati necessari.

L'applicazione comprende principalmente due pagine: la pagina di login e la pagina che mostra le dashboard che l'utente può visualizzare. Nel momento in cui l'utente avvia una dashboard il programma ottiene, tramite una chiamata al database, il link corrispondente al codice del file relativo alla dashboard selezionata e lo inserisce all'interno dell'app tramite un elemento iframe. Se l'utente chiude la dashboard il codice inserito viene eliminato per fare spazio al codice di un'eventuale nuova dashboard.

La pagina web comprende tutto ciò che può essere modificato nel tempo, al fine di rendere l'applicazione più flessibile. Tale pagina è stata gestita tramite Laravel e l'host utilizzato è Site Ground. I file principali presenti all'interno di tale pagina sono: i file che rappresentano le singole dashboard, la classe che definisce le route, la classe che descrive i controller, la cartella che contiene foto e video e la cartella che contiene le pagine del database manager.

Di seguito viene mostrato uno schema delle directory principali.

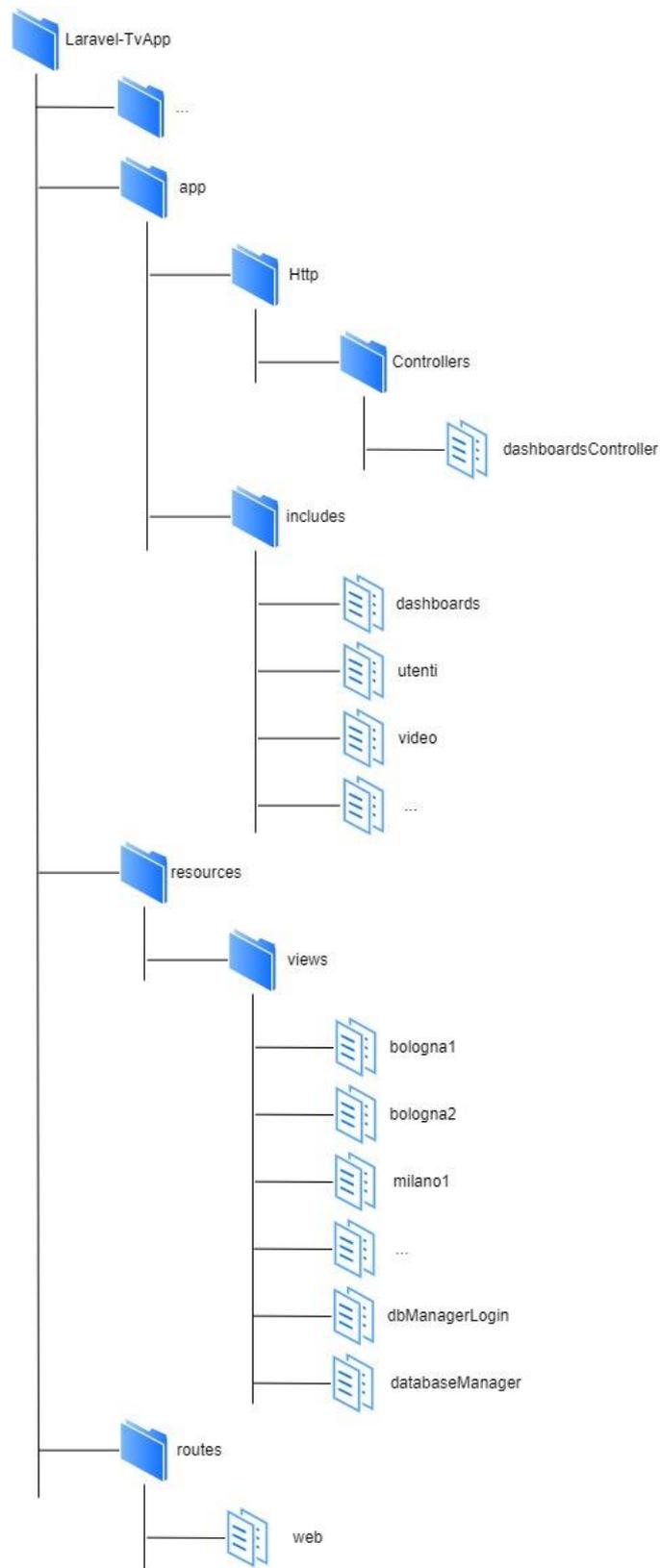


Figura 4.1: schema delle directory

Tutti i file relativi alle dashboard sono salvati all'interno del seguente percorso: "TVproject2.0/Laravel-TvApp/resources/views".

Per quanto riguarda gli url da utilizzare per ottenere i dati, essi sono specificati all'interno della classe "web.php" della cartella "TVproject2.0/Laravel-TvApp/routes". All'interno di tale file è possibile specificare i percorsi e ciò che il programma deve eseguire. Considerando l'esempio dei video, questa è la route:

```
Route::get('/show-video/{appName}',[dashboardsController::class,
'showVideoByAppName']);
```

Attraverso questa funzione è possibile, pertanto, specificare il link da cui sarà possibile ottenere le informazioni, la classe PHP "dashboardsController", all'interno della quale è presente il metodo che esegue le operazioni e il nome del metodo da utilizzare, "showVideoByAppName".

Le classi corrispondenti ai controller sono salvate all'interno della seguente cartella: "TVproject2.0/Laravel-TvApp/app/Http/Controllers".

Per quanto riguarda i file relativi al database manager, essi sono stati realizzati in PHP e sono salvati all'interno della cartella: "TVproject2.0/Laravel-TvApp/resources/views". All'interno di essa sono due i file principali:

- dbManagerLogin: file relativo alla pagina di login della piattaforma
- databaseManager: file relativo alla home page del database manager, all'interno di esso sono definiti header e footer del programma, mentre il body è vuoto e viene sostituito con il file corrispondente alla voce del menu che l'amministratore sceglie. Questi file sono presenti all'interno della cartella "TVproject2.0/Laravel-TvApp/app/includes", anch'essi sviluppati in PHP.

Tali file permettono di visualizzare i dati presenti nel database, aggiungerne di nuovi e modificare o eliminare quelli già presenti. Infatti, è comune a tutte le pagine una tabella che mostra i dati, per ogni riga della tabella sono presenti due bottoni, uno a forma di matita che permette di modificare la riga scelta e uno a forma di

cestino che permette di eliminare la riga. Inoltre, sopra la tabella, è presente un pulsante che, se cliccato, mostra un form attraverso il quale è possibile inserire i dati nuovi.

La gestione del database è affidata a dei controller che vengono utilizzati ogni volta che viene inviato un form dai file descritti in precedenza.

Le librerie esterne che ho utilizzato sono:

- jQuery: per poter effettuare le chiamate AJAX al database
- QrCodeGenerator: per creare in tempo reale il qr code della pagina internet dei webinar aziendali
- Leaflet Map: per poter utilizzare le funzioni JavaScript della mappa

Le API che ho utilizzato sono:

- Open Weather Map: per ottenere i dati relativi al meteo in base alla località
- Open Street Map: per visualizzare la mappa
- Here API: per ottenere i dati relativi alla metropolitana

### **4.4 Descrizione del database**

Per gestire i dati del programma è stato necessario progettare ed implementare un database nel quale l'amministratore potesse salvare i dati che verranno richiesti dall'applicazione presente sulla smart tv.

Di seguito viene mostrato il diagramma entità-relazione del database.

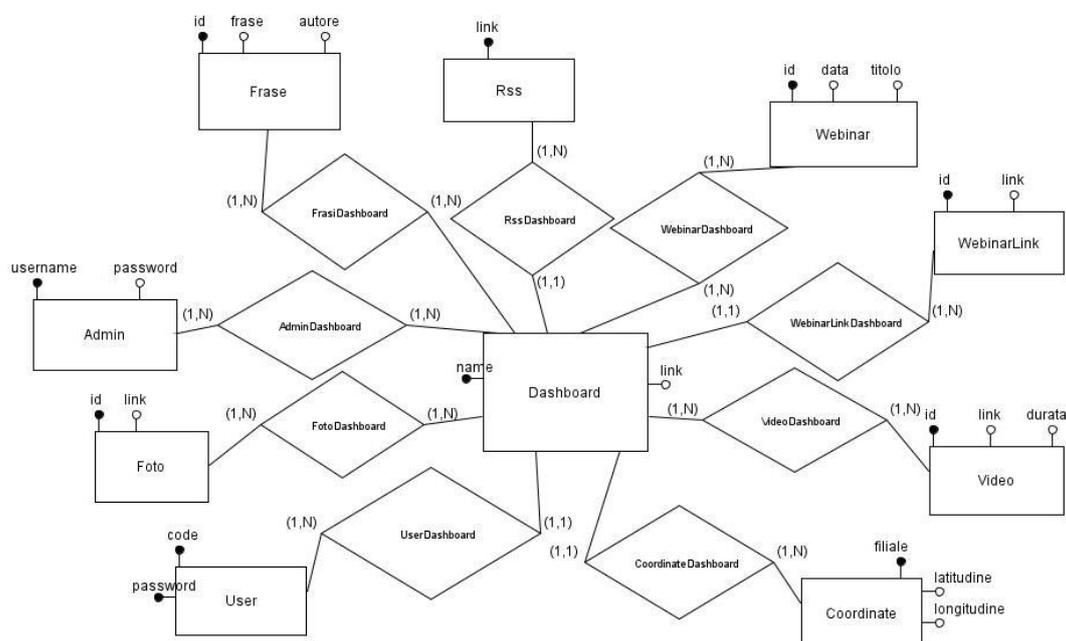


Figura 4.2: diagramma entità-relazione

La parte centrale del database riguarda l'entità delle dashboard e tutte le restanti entità sono collegate tramite una relazione ad essa.

L'entità Dashboard corrisponde a tutte le dashboard presenti nella piattaforma, è collegata tramite una relazione uno a molti all'entità User, che corrisponde agli utenti che possono utilizzare l'app sulla smart tv, e tramite una relazione molti a molti all'entità Admin, che corrisponde agli utenti amministratori che possono modificare i dati di tutte le dashboard.

Tutte le altre entità corrispondono ai contenuti che possono essere visualizzati sulla televisione e per ognuna di esse sono presenti gli attributi che caratterizzano il contenuto corrispondente.

## 4.5 Punti di criticità

I problemi maggiori che ho riscontrato durante lo sviluppo di questo progetto sono principalmente due: uno riguardante la visualizzazione dei contenuti su un dispositivo reale, l'altro in merito alla gestione dell'interazione utente attraverso l'utilizzo del telecomando.

Per quanto riguarda la visualizzazione dei video, inizialmente la soluzione prevista è stata mostrarli tramite le API di YouTube però, diversamente da quanto mi aspettavo, su un dispositivo reale, l'applicazione non mostrava il video nell'app ma apriva il browser collegandosi alla pagina web di YouTube. Un'altra soluzione che ho considerato è stata l'inserimento del video nel file system della televisione e la sua visualizzazione tramite il percorso al file ma sarebbe stata una soluzione poco flessibile in quanto non avrebbe permesso il corretto utilizzo su televisioni differenti. Per tali motivi, la soluzione che ho scelto è stata inserire il video online e visualizzarlo tramite url, richiamando così la funzione "play()" del video ogni volta che deve essere avviato.

Oltre alla visualizzazione dei video, anche la visualizzazione della mappa è stata complicata. In particolare, la mappa deve mostrare gli interventi sul campo dei tecnici aziendali e in fase iniziale la soluzione adottata è stata utilizzare la mappa di Google ma in seguito ad alcuni accertamenti, ho potuto riscontrare che essa non era la soluzione migliore per diversi motivi. In primo luogo, Google mette a disposizione la mappa gratuitamente ma sono necessari altri controlli che possono compromettere la sua visualizzazione su più dispositivi. In secondo luogo, dopo molteplici tentativi effettuati, la mappa di Google veniva visualizzata sul simulatore ma non su un dispositivo reale, per tale motivo, inizialmente, ho provato ad aggiungere alcune impostazioni di privacy nel file di configurazione, successivamente ho provato ad inserire una mappa molto basilare ma non è stato sufficiente al fine di risolvere il problema. Per questi motivi ho deciso di utilizzare Open Street Map e attraverso questa libreria la mappa viene visualizzata correttamente.

Per quanto riguarda la gestione dell'interazione con l'utente, essa è stata necessaria per fare in modo che l'utente che utilizza l'app sulla televisione possa autenticarsi attraverso la password. Perciò ho realizzato una pagina di login che venga visualizzata all'avvio dell'applicazione. L'ostacolo che ho riscontrato è stato decidere come permettere all'utente di inserire la password, non disponendo né di tastiera né di mouse ma di un telecomando. Se prendiamo in considerazione le principali applicazioni per smart tv, per esempio Netflix o YouTube, viene spontaneo pensare che, inserendo nel file html un campo di testo, ciò renda automatica la possibilità per l'utente di scrivere i numeri attraverso il telecomando. In realtà, lo sviluppo di un programma di questa tipologia per televisioni è molto diverso rispetto allo sviluppo per pc o dispositivi mobili: nel caso delle televisioni, infatti, è necessario intercettare il click del telecomando e, in base al tasto cliccato, effettuare le operazioni richieste. In particolare, all'avvio dell'app la tastiera virtuale non è visibile e viene mostrata solo nel momento in cui l'utente clicca il tasto "OK" del telecomando, successivamente può navigare tra i numeri mostrati attraverso i tasti "freccia sinistra" e "freccia destra" e successivamente alla selezione del numero, cliccare nuovamente "OK". Questo procedimento è possibile grazie ad una serie di blocchi if che permettono al programma di capire in quale situazione si trova l'utente nel momento in cui clicca i bottoni, dato che ad ogni singolo bottone sono associate operazioni diverse, per esempio il bottone "OK".

## Capitolo 5

### Conclusioni

Lo sviluppo di questo progetto è stato interessante fin dalle fasi iniziali, progettare un'applicazione per smart tv non è molto comune e risolvere i problemi riscontrati durante la realizzazione è stata una bella sfida che mi ha permesso di approfondire temi già conosciuti e di scoprirne dei nuovi che non avevo mai affrontato.

I punti di forza di tale applicazione sono diversi, tra i più importanti troviamo la possibilità di utilizzare l'applicazione in filiali differenti e di poter realizzare delle dashboard ad hoc per ognuna di esse. Un altro punto di forza è la flessibilità del sistema distribuito, infatti, è possibile modificare i contenuti da mostrare utilizzando solamente il database manager, senza la necessità di modificare il codice sorgente, perciò, anche un dipendente che non è uno sviluppatore può amministrare le pagine. Inoltre, l'applicazione può funzionare correttamente per un periodo di tempo potenzialmente illimitato, infatti, può essere avviata al mattino e rimanere in funzione senza interagire con l'utente per tutta la giornata.

Tra i punti di debolezza troviamo il funzionamento dell'app sulla smart tv, in particolare, il processore di una televisione non è avanzato come quello dei computer e perciò a volte capita che l'app si veda a scatti per qualche istante. Inoltre, alcune parti del codice relativo alle dashboard sono duplicate tra i singoli file, perciò, un eventuale miglioramento può essere realizzare dei componenti HTML che vengano riutilizzati all'interno di ogni file in cui sono necessari. In ultimo, la visualizzazione dei contenuti centrali segue un ordine prestabilito: video, foto, mappa e webinar, un eventuale miglioramento può essere inserire per ogni contenuto una precedenza e mostrare i contenuti in base ad essa.

## 5.1 Campi di applicazione

L'applicazione descritta è stata progettata e sviluppata per l'azienda nella quale ho svolto il tirocinio, tuttavia, può essere utilizzata in contesti differenti. In particolare, non è specifica per l'azienda o il settore per la quale è stata realizzata perché può mostrare qualsiasi tipo di contenuto multimediale.

Per esempio, può essere utilizzata all'entrata di negozi di vendita al dettaglio per mostrare foto di prodotti in vendita, per visualizzare un qr code che rimanda alla pagina web del negozio oppure per indicare attraverso la mappa tutti i punti vendita presenti sul territorio.

Inoltre, può diventare utile per i ristoranti. Molti di essi presentano delle televisioni all'interno delle loro sale, tale applicazione può essere utilizzata per mostrare il menu del ristorante, foto di piatti oppure video di presentazione degli ingredienti utilizzati in cucina. Spesso mi è capitato di recarmi in alcuni ristoranti e di notare alcuni video in riproduzione nelle televisioni del locale, in questi casi però, nel momento in cui il video termina, il cameriere deve far ripartire il video a mano, sarebbe più opportuno utilizzare l'applicazione che ho realizzato per fare in modo che i contenuti scorrano in maniera automatica per un periodo di tempo indefinito.

In conclusione, penso che tale applicazione possa essere impiegata in qualsiasi tipo di azienda, non presenta limiti legati al settore o al tipo di impresa e può essere facilmente modificata in modo tale che possa adattarsi al meglio alle esigenze di chi la utilizza e tramite il database manager, la gestione dei contenuti risulta di facile utilizzo, per qualsiasi tipo di utente.

## 5.2 Feedback aziendali

Al termine della realizzazione del progetto, ho posto al referente aziendale alcune domande sul prodotto finito, di seguito vengono elencate le risposte.

### 1. Punti di forza:

Il punto di forza principale è poter modificare la visualizzazione di ogni TV decentralizzata in ogni filiale direttamente da un portale web ad uso interno del team marketing.

Questo permette di poter modificare completamente la visualizzazione sui dispositivi senza recarsi in ogni filiale e azzerare i tempi di distribuzione delle modifiche.

### 2. Punti di debolezza:

Eventuali aggiornamenti legati al funzionamento dell'applicazione ad oggi devono essere ancora installati manualmente.

Modifiche minime come l'aggiunta di un attributo di sicurezza per permettere all'applicazione di poter accedere a determinati siti o endpoint web non possono essere fatti in modo centralizzato ma è necessario ricompilare l'applicazione e procedere nuovamente con l'installazione manuale.

### 3. Eventuali miglioramenti:

L'integrazione e pubblicazione dell'applicazione sullo store ufficiale permetterebbe di poter distribuire l'applicazione ed evitare spostamenti per l'installazione degli aggiornamenti.

Ulteriori miglioramenti riguardano capire come pubblicare l'applicazione sullo store ufficiale ma allo stesso tempo renderla un'applicazione business privata, cioè ad uso e consumo solo dell'azienda che l'ha creata.

## Sitografia

- [1] <https://github.com/lewhunt/react-tv-player>
- [2] <https://github.com/Raavtube/SmartTV>
- [3] <https://github.com/gibbok/jquery-slidertv>
- [4] <https://github.com/giang-nghg/smart-tv-player>
- [5] [https://github.com/amberbit/google\\_maps\\_samsung\\_tv](https://github.com/amberbit/google_maps_samsung_tv)
- [6] [https://github.com/hoomi/PhotoView\\_SamsungSmartTv](https://github.com/hoomi/PhotoView_SamsungSmartTv)
- [7] <https://laravel.com/>
- [8] <https://it.siteground.com/>
- [9] <https://developer.tizen.org/development/tizen-studio>
- [10] <https://jquery.com/>
- [11] <https://davidshimjs.github.io/qrcodejs/>
- [12] <https://www.here.com/>
- [13] <https://www.openstreetmap.org/>
- [14] <https://leafletjs.com/>
- [15] <https://openweathermap.org/>



## **Ringraziamenti**

Ringrazio i miei genitori, Claudio e Daniela, per avermi sempre sostenuto durante questa esperienza, attraverso i loro sacrifici e i loro consigli sempre utili.

Ringrazio i miei nonni Clara e Paolo, i miei zii Antonella e Fabio e tutta la mia famiglia per aver reso le mie giornate più spensierate e per interessarsi sempre ai miei studi.

Ringrazio la mia fidanzata, Maya, per avermi sempre supportato con tanto affetto e per avermi rallegrato con il suo sorriso contagioso nelle giornate più complicate.

Ringrazio la Professoressa Guidetti, che non è stata solo un'insegnante di matematica ma anche un'insegnante di vita.

Ringrazio Centro Software e tutto il personale dell'ufficio marketing per la collaborazione durante lo sviluppo del progetto.

Infine, vorrei ringraziare il Professor Di Iorio per la sua disponibilità e per avermi aiutato nella stesura di questa tesi.