



Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea in Informatica

CINEMA MULTISALA

Realizzazione di una web app per la gestione di un cinema

Relatore: *Dott.ssa Daniela Micucci*

Co-relatore: *Dott.ssa Annalisa Marra (Sync Lab S.r.l)*

Relazione della prova finale di:

Daniele Scalco

Matricola 844762

Anno Accademico 2020-2021

INDICE

Introduzione

1. Sync Lab
2. Descrizione del progetto
 - 2.1. Requisiti funzionali
 - 2.2. Diagramma dei casi d'uso
 - 2.3. Descrizione dettagliata dei casi d'uso
 - 2.4. Diagrammi di attività
3. Framework e tecnologie utilizzate per lo sviluppo
 - 3.1. Spring
 - 3.1.1. Spring MVC
 - 3.1.2. Spring Boot
 - 3.1.3. Spring Security
 - 3.2. Hibernate
 - 3.3. Thymeleaf
 - 3.4. Bootstrap
4. Base dati
 - 4.1. Modello Entity-Relationship
 - 4.2. Database MySql
5. Struttura del progetto java
6. Dettagli dell'interfaccia utente
7. Conclusioni e possibilità di sviluppo futuri

INTRODUZIONE

Lo scopo di questa relazione è trattare il contenuto pratico del tirocinio che ho svolto per tre mesi presso l'azienda Sync Lab. In questa durata ho effettuato studi e in parallelo ho sviluppato un software, seguendo le specifiche che mi sono state fornite, per la realizzazione di un'applicazione web per la gestione di un ipotetico cinema. Per la realizzazione sono stati utilizzati Eclipse e MySql Workbench e il codice del back-end è stato scritto in java e MySql, mentre il front-end in html, css, javascript e thymeleaf. Di seguito si ha la descrizione dei contenuti dei vari capitoli della relazione.

Si parte nel primo capitolo con una breve introduzione descrivendo brevemente l'azienda in cui ho effettuato lo stage. Nel secondo capitolo viene invece descritto il progetto e analizzato nel dettaglio tutte le funzionalità che l'applicazione in questione offre. Nel terzo capitolo vengono descritte, una per una, tutte le tecnologie che sono state usate nella realizzazione della webapp. Nel quarto capitolo viene affrontata la persistenza dei dati mostrando il processo che ha portato alla realizzazione del database. Nel quinto capitolo viene analizzata nel dettaglio la struttura di come è organizzato il codice java nei vari packages. Nel penultimo capitolo vengono mostrate le immagini dell'applicazione facendo vedere l'interfaccia grafica realizzata per ogni funzionalità. Infine, nell'ultimo capitolo si tirano le conclusioni su questo progetto e vengono proposte delle idee su possibili miglioramenti e/o aggiunte che è possibile realizzare in futuro.

1) SYNC LAB

Sync Lab è un'azienda nata come Software house e, nel corso degli anni dopo aver acquisito più competenze tecnologiche, si è tramutata in System Integrator e attualmente possiede cinque sedi in tutta Italia.

Grazie agli ottimi prodotti realizzati nel loro laboratorio di ricerca e sviluppo sono riusciti a conquistare fette di mercato nei settori mobile, videosorveglianza e sicurezza informatica delle infrastrutture informatiche aziendali e fra i loro clienti possono vantare alcuni colossi come Unicredit, H&M, Sky, Fastweb e Eni.

2) DESCRIZIONE DEL PROGETTO

Il progetto assegnatomi consiste nella realizzazione di un'applicazione web per la gestione di un ipotetico cinema realizzata utilizzando il framework di Spring e Hibernate (questi elementi verranno spiegati meglio nel capitolo successivo).

Le persone che interagiscono con il sistema possono avere i seguenti ruoli:

- Utente
- Dipendente
- Amministratore

L'utente è il cliente interessato al cinema in questione. Può visitare il sito liberamente e vedere i vari film in programmazione disponibili e se desidera effettuare una prenotazione è obbligato a dover effettuare prima la registrazione.

Il dipendente è colui che lavora per il cinema e si occupa della gestione della programmazione dei film.

L'amministratore può fare le stesse cose di un dipendente ed inoltre si occupa anche della gestione degli account registrati.

2.1) Requisiti funzionali

Di seguito vengono mostrare i requisiti funzionali, ovvero i servizi che il sistema deve (o non deve) offrire, suddivisi per ognuno dei ruoli menzionati precedentemente.

Utente:

- Può registrarsi nel sistema ed in seguito effettuare login/logout;
- Può visualizzare la programmazione dei film disponibili nell'arco di una settimana;
- Per ogni film in programmazione può visualizzare la relativa scheda con tutti i dati in dettaglio (titolo, immagine locandina, regista, cast, sala, descrizione, orari);
- Può effettuare una prenotazione per uno dei film disponibili se ci sono posti disponibili in sala.

Dipendente:

- Può effettuare il login/logout;
- Può aggiungere un nuovo film nel database specificando tutti i dati necessari;
- Può eliminare un film dal database (sia un singolo orario di riproduzione che tutti quelli con un determinato titolo);
- Può modificare i dati di dettaglio di un film e aggiungere nuovi orari.

Amministratore:

L'amministratore ha a disposizione le stesse funzionalità di un dipendente ed inoltre:

- Può visualizzare la lista degli account registrati nel sistema;
- Può modificare i ruoli assegnati ad un account;
- Può eliminare un account;
- Può creare un nuovo account fornendogli un qualunque ruolo.

2.2) Diagramma dei casi d'uso

Di seguito, nella figura 2.1, viene mostrato il diagramma dei casi d'uso che mostra i vari attori con i casi d'uso associati e come questi interagiscono con il sistema in questione.

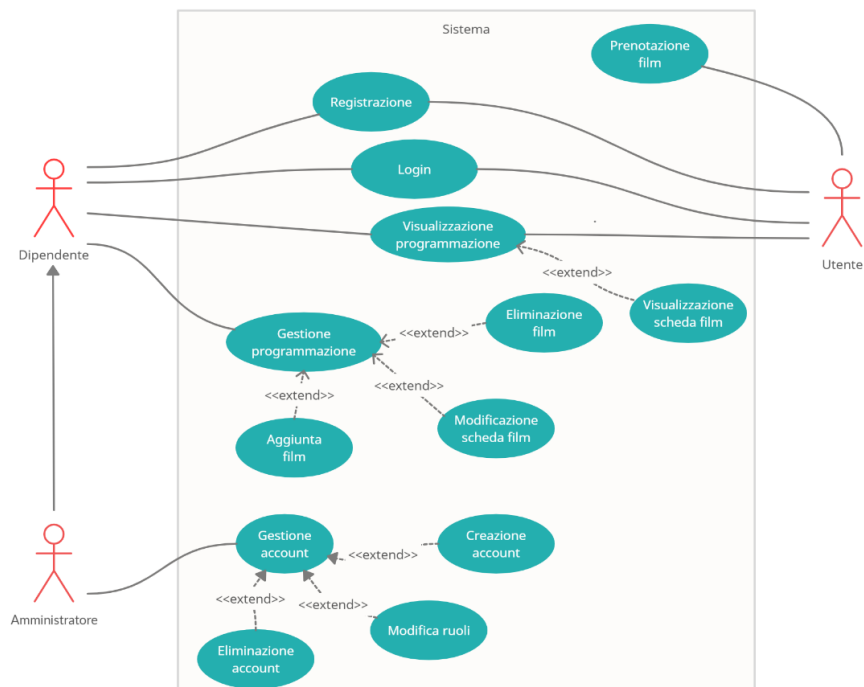


Figura 2.1: Diagramma dei casi d'uso

2.3) Descrizione dettagliata dei casi d'uso

In questa sezione vengono riportate le descrizioni dettagliate dei casi d'uso previsti, in cui vengono mostrati gli scenari di successo e anche i possibili errori che l'utilizzatore in questione potrebbe effettuare.

REGISTRAZIONE

Livello: sottofunzione

Attori Primari: utente

Parti interessate e interessi: l'utente vuole registrare il proprio account per poter poi usare l'applicazione sfruttando le sue funzionalità al massimo.

Pre-condizioni: nessuna

Garanzia di successo: l'utente viene registrato correttamente nel sistema

Scenario principale di successo:

1. L'utente seleziona "REGISTRAZIONE" nel menù in alto a destra dell'homepage
2. Il sistema mostra la pagina contenente la form per registrarsi
3. L'utente compila tutti i campi ("e-mail", "password" e "conferma password") e seleziona il bottone "Registrati"
4. Se i campi sono vuoti o non rispettano le condizioni di validità il sistema mostra i relativi errori e si torna al punto 2, altrimenti il sistema registra l'utente e ritorna nella Homepage mostrando un messaggio di conferma avvenuta registrazione

LOGIN

Livello: sottofunzione

Attori Primari: utente/dipendente/amministratore

Parti interessate e interessi: l'utilizzatore vuole effettuare l'accesso al proprio account

Pre-condizioni: l'utilizzatore dev'essersi precedentemente registrato nel sistema

Garanzia di successo: l'utilizzatore effettua il login

Scenario principale di successo:

1. L'utente seleziona "LOGIN" nel menù in alto a destra dell'homepage
2. Il sistema mostra la pagina contenente la form per effettuare il login
3. L'utente compila tutti i campi ("e-mail" e "password") e seleziona il bottone "Login"
4. Se i campi sono vuoti o non rispettano le condizioni di validità oppure se l'e-mail risulta già associata ad un account registrato il sistema mostra i relativi errori e si torna al punto 2, altrimenti l'utente viene loggato correttamente e ritorna nella Homepage.

VISUALIZZAZIONE PROGRAMMAZIONE

Livello: obiettivo utente

Attori Primari: utente

Parti interessate e interessi: l'utente vuole vedere i film che saranno presenti in sala nei prossimi sette giorni

Pre-condizioni: nessuna

Garanzia di successo: l'utente visualizza la lista di film in programmazione

Scenario principale di successo:

1. L'utente seleziona "PROGRAMMAZIONE" nel menù in alto a sinistra dell'homepage
2. Il sistema mostra la pagina contenente tutti i film in programmazione nei prossimi sette giorni partendo dalla data attuale

PRENOTAZIONE

Livello: obiettivo utente

Attori Primari: utente

Parti interessate e interessi: l'utente vuole prenotare dei posti per il film, la data e l'orario da lui scelti

Pre-condizioni: l'utente dev'essersi loggato nel sistema

Garanzia di successo: l'utente effettua la prenotazione con successo

Scenario principale di successo:

1. L'utente seleziona "PROGRAMMAZIONE" nel menù in alto a sinistra dell'homepage
2. Il sistema mostra la pagina con i film presenti in programmazione
3. L'utente seleziona la data e l'orario che preferisce
4. Il sistema mostra la pagina con lo stato attuale delle prenotazioni per il film scelto dall'utente
5. Se l'utente prova a selezionare un posto occupato compare un alert con un messaggio d'errore altrimenti l'utente seleziona i posti che desidera fra quelli segnalati come liberi e clicca sul bottone "Prenota"
6. Il sistema registra la prenotazione nel database e mostra una pagina di riepilogo con tutti i dati della prenotazione

GESTIONE PROGRAMMAZIONE

Livello: obiettivo utente

Attori Primari: dipendente/amministratore

Parti interessate e interessi: l'utilizzatore vuole gestire la programmazione aggiungendo, rimuovendo o modificando i film

Pre-condizioni: l'utilizzatore dev'essersi loggato nel sistema

Garanzia di successo: l'utilizzatore effettua i cambiamenti che desidera alla programmazione

Scenario principale di successo:

Se si vuole aggiungere un film:

1. Il dipendente seleziona "AGGIUNGI FILM" nel menù in alto a sinistra dell'homepage
2. Il sistema mostra la pagina con una form in cui inserire tutti i dati riguardanti il film da aggiungere
3. Il dipendente immette i dati del film (titolo, regista, cast, descrizione, immagine locandina, data, orario, sala) e preme su "Aggiungi"
4. Se i campi sono vuoti oppure se si prova ad inserire un film in una sala in un orario destinato ad un altro film già presente il sistema mostra i relativi errori e si torna al punto 2, altrimenti il sistema torna alla pagina di aggiunta film mostrando un messaggio di avvenuta aggiunta

Se si vuole modificare/eliminare un film:

1. Il dipendente seleziona "PROGRAMMAZIONE" nel menù in alto a sinistra dell'homepage
2. Il sistema mostra i film in programmazione

Se si vuole eliminare un orario di un film:

- 2.1. Il dipendente seleziona il bottone relativo all'orario che vuole eliminare
- 2.2. Il sistema elimina il film con l'orario selezionato e ritorna nella schermata della programmazione mostrando un messaggio di conferma

Se si vuole eliminare un film completamente con tutti gli orari:

- 2.1. Il dipendente seleziona il bottone "Elimina film" relativo ad un titolo presente
- 2.2. Il sistema elimina ogni film con quel titolo e ritorna alla schermata della programmazione mostrando un messaggio di conferma

Se si vuole aggiungere un orario per un film:

- 2.1. Il dipendente seleziona il bottone "Aggiungi orario" relativo ad un titolo presente
- 2.2. Il sistema mostra all'utilizzatore una nuova pagina in cui è possibile immettere data, ora e numero sala
- 2.3. Il dipendente compila tutti i campi e seleziona "Aggiungi orario"
- 2.4. Se l'orario selezionato si sovrappone con quello di un altro film nella stessa sala compare un messaggio d'errore e si torna al punto 2.2, altrimenti il sistema registra il nuovo orario e, rimanendo sulla schermata di aggiunta orario per permettere eventualmente di inserirne un altro, mostra un messaggio di conferma

Se si vuole modificare i dati di dettaglio di un film:

- 2.1. Il dipendente seleziona il bottone "Modifica film" relativo ad un titolo presente
- 2.2. Il sistema mostra una schermata con i dati di dettaglio attuali del film, ovvero titolo, regista, cast, descrizione e immagine locandina
- 2.3. Il dipendente modifica tutti i campi che desidera cambiare e seleziona "MODIFICA"
- 2.4. Se i campi sono vuoti viene mostrato un messaggio d'errore e si torna al punto 2.2, altrimenti il sistema ritorna alla schermata di programmazione mostrando un messaggio di conferma avvenuta modifiche

GESTIONE ACCOUNT

Livello: obiettivo utente

Attori Primari: amministratore

Parti interessate e interessi: l'amministratore vuole visualizzare e/o modificare gli account registrati nel sistema

Pre-condizioni: l'amministratore dev'essere loggato nel sistema

Garanzia di successo: l'amministratore porta a compimento tutte le azioni da lui richieste

Scenario principale di successo:

1. L'amministratore clicca in alto a sinistra nel menu "GESTIONE UTENTI"
2. Il sistema mostra una pagina con una lista contenente tutti gli account registrati in ordine alfabetico per indirizzo e-mail

Se si vuole aggiungere un nuovo account:

- 2.1. L'amministratore seleziona il bottone "Aggiungi utente"
- 2.2. Il sistema mostra una nuova pagina con una form per aggiungere i dati relativi al nuovo account
- 2.3. L'amministratore compila i campi "e-mail", "password" e "conferma password" e seleziona i ruoli e seleziona "Crea account"
- 2.4. Se i campi sono vuoti o non rispettano le condizioni di validità oppure se l'e-mail risulta già associata ad un account registrato il sistema mostra i relativi errori e si torna al punto 2.2, altrimenti il sistema ritorna alla schermata con la lista degli utenti aggiornata e mostra un messaggio di avvenuta registrazione

Se si vuole eliminare un account:

- 2.1. L'amministrazione seleziona il bottone "Elimina" relativo alla riga dell'account che vuole eliminare
- 2.2. Il sistema mostra un alert per confermare l'eliminazione dell'account selezionato
- 2.3. L'amministratore conferma la scelta
- 2.4. Il sistema elimina l'account e mostra un messaggio di conferma eliminazione

Se si vuole modificare i ruoli che possiede un account:

- 2.1. L'amministrazione seleziona il bottone "Aggiorna" relativo alla riga dell'account che vuole modificare
- 2.2. Il sistema mostra una nuova pagina in cui si vede la mail dell'account selezionato e si possono scegliere i ruoli

2.3. L'amministratore aggiunge e/o elimina i ruoli che desidera e seleziona "Applica modifiche"

2.4. Se non è stato selezionato alcun ruolo viene mostrato un messaggio d'errore, altrimenti il sistema ritorna alla schermata con la lista degli utenti aggiornata e mostra un messaggio di conferma delle modifiche

2.4) DIAGRAMMI DI ATTIVITA'

In questo paragrafo verranno mostrati i diagrammi di attività che modellano gli aspetti dinamici e sono molto utili a capire il flusso delle azioni che eseguono i vari attori che interagiscono con il sistema.

Di seguito vengono mostrati tre diagrammi, uno per ruolo, in cui vengono mostrate le attività che ciascuno può eseguire. Nella figura 2.2 si vedono le attività di un utente, nella 2.3 quelle di un dipendente e nella 2.4 quelle di un amministratore.

Convenzionalmente la "fork" è una barra di sincronizzazione che divide il flusso in due parti che avvengono in parallelo, qui si è deciso che invece rappresenta le possibili azioni differenti (esclusive, o una o l'altra) che l'attore coinvolto può decidere di eseguire.

UTENTE

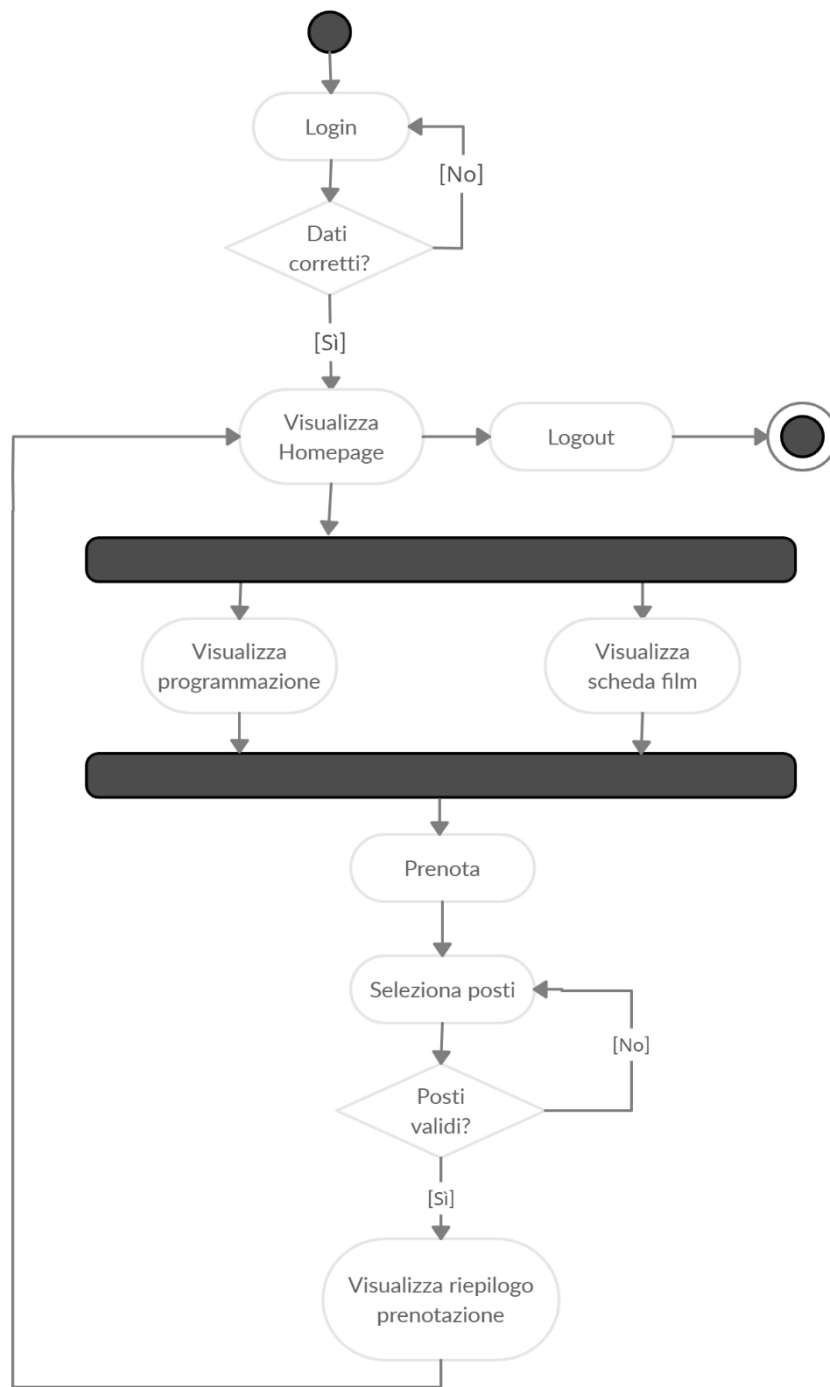


Figura 2.2: Diagramma di attività - utente

DIPENDENTE

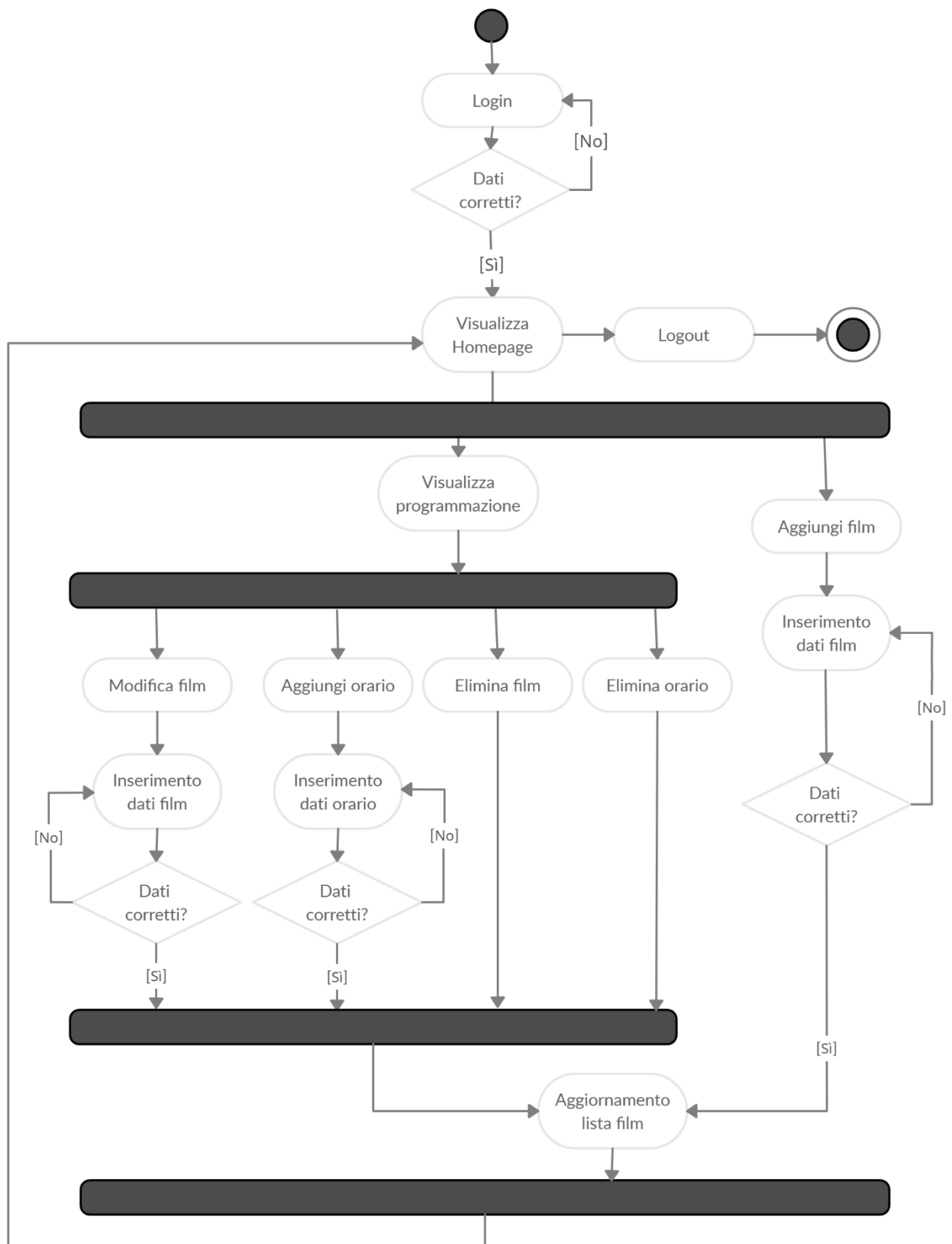


Figura 2.3: Diagramma di attività - dipendente

AMMINISTRATORE

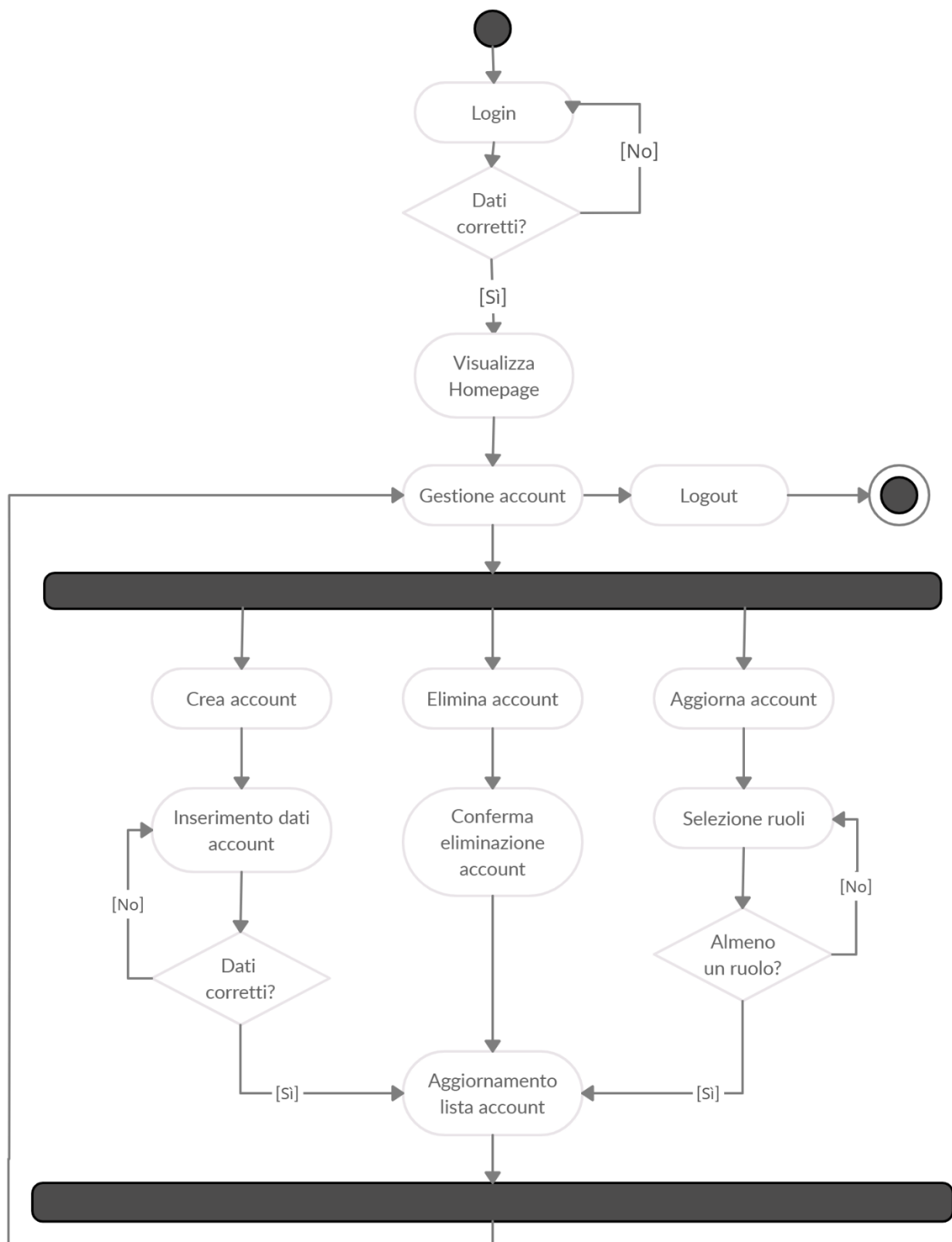


Figura 2.4: Diagramma di attività – amministratore

3) FRAMEWORK E TECNOLOGIE UTILIZZATE PER LO SVILUPPO

Per lo sviluppo dell'applicativo sono stati utilizzati diversi framework e piattaforme, ognuno con le sue peculiarità e funzionalità che verranno descritte in questo capitolo.

3.1) Spring

Spring è un framework opensource utile per lo sviluppo di applicazioni enterprise scritte in java. È nato come alternativa a *J2EE* (Java 2 Enterprise Edition) proponendo un modello più semplice e leggero per lo sviluppo di entità di business dato che la sua architettura è modulare permettendone così l'utilizzo sia nella sua interezza che solo in parte.

Spring si basa sui concetti di *Inversion of Control* (IoC) e *Dependency Injection* (DI):

IoC:

nella logica tradizionale è lo sviluppatore che si occupa della creazione, inizializzazione ed invocazione dei metodi di un oggetto, con questo principio architetturale invece è il framework che si preoccupa di gestire tutto ciò.

DI:

è un pattern per implementare l'IoC che permette di collegare gli oggetti tra loro attraverso un assembler che si occupa di iniettare le dipendenze. Le dipendenze possono essere iniettate in tre modi:

1. Constructor-based DI
2. Setter-based DI
3. Field-based DI (utilizzata in questo progetto)

3.1.1) Spring MVC

Spring MVC è un web framework di Spring basato sul paradigma Model-View-Controller (MVC), un pattern architetturale che separa la logica di business dalla logica di presentazione, mostrato nella figura 3.1. Ogni strato ha la sua funzione:

- Model: contiene i metodi di accesso ai dati
- View: permette di visualizzare i dati ed è il mezzo con il quale l'utente interagisce col sistema
- Controller: si occupa della gestione delle altre due componenti. Riceve gli input dall'utente per mezzo della View, elabora i dati eseguendo operazioni sul Model e porta ad un cambiamento di stato della View.

Spring MVC è dotata di una classe *DispatcherServlet* (Front Controller) che riceve gli URI in arrivo e li inoltra alle classi Controller corrispondenti.

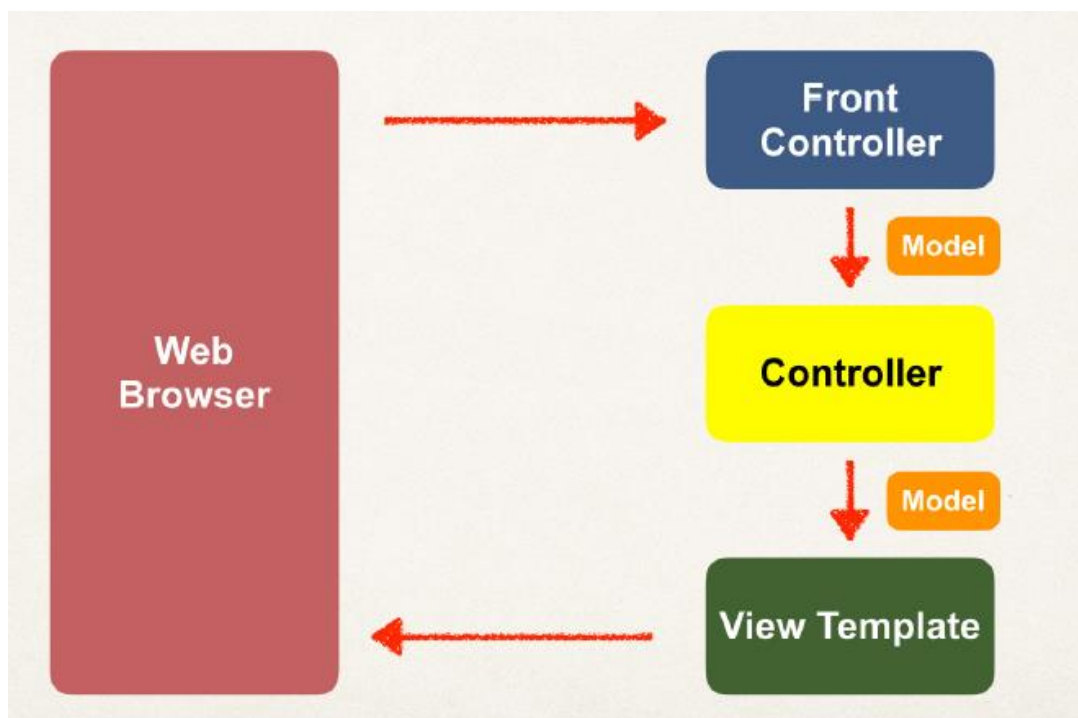


Figura 3.1: Spring MVC dietro le quinte

3.1.2) Spring Boot

Spring Boot è un framework che permette la semplificazione dello sviluppo di un'applicazione Spring standalone, riducendo le configurazioni manuali che deve effettuare

lo sviluppatore e il codice boilerplate. Spring Boot, infatti, configura in automatico l'applicazione basandosi sulle dipendenze che sono state aggiunte e fornisce inoltre anche un server Tomcat incorporato.

La creazione di un progetto può essere facilmente eseguita utilizzando Spring Initializr in cui bisogna solamente selezionare il tipo di progetto (Maven o Gradle), il linguaggio utilizzato e le dipendenze che servono.

3.1.3) Spring Security

Spring Security è un framework di supporto di Spring per le funzionalità di autenticazione, autorizzazione e controllo degli accessi. In questo modo è possibile salvare nella base dati gli account che vengono registrati e far accedere alle pagine web solo gli utenti con determinati ruoli per garantire la sicurezza dell'applicativo. Per il salvataggio delle password nel progetto si è fatto uso della classe *BCryptPasswordEncoder* che fanno uso di bcrypt, una funzione di hashing di password, in modo da salvare sul database i dati in modo sicuro. Una funzione di hashing è una funzione *one-way*, ovvero non è possibile invertirla e in questo modo i dati sensibili non possono essere decrittati.

3.2) Hibernate

Hibernate è una piattaforma middleware open source per lo sviluppo di applicazioni java che offre numerosi servizi. Il servizio più usato è quello di *Object-Relational Mapping* (ORM) che permette di mappare degli oggetti java su un database relazione in modo da gestire correttamente la persistenza e riducendo il codice *Java Database Connectivity* (JDBC) da dover inserire. Hibernate implementa le specifiche della *Java Persistence Api* (JPA) per la persistenza dei dati. Un altro servizio che fornisce che è stato utilizzato nel progetto è Hibernate Validator, utilizzato per la validazione dei dati degli account degli utenti che si vogliono registrare.

3.3) Thymeleaf

Thymeleaf è un *template engine* opensource che si può utilizzare sia in ambienti web che non web. È pensato per essere utilizzato nello strato di *View* in un paradigma *Model-View-Controller* (MVC) mirando ad essere un sostituto delle *Java Server Pages* (JSP). Offre una completa integrazione con Spring.

3.4) Bootstrap

Bootstrap è un framework per la creazione di siti e applicazioni web. È nato con lo scopo di uniformare e accelerare lo sviluppo delle interfacce web e dalla versione 2.0 permette di rendere i siti web responsivi, permettendo così di adattarsi alla miriade di schermi di misure differenti presenti oggi. È una raccolta di strumenti grafici, stilistici e di impaginazione che permettono di avere a disposizione una gran quantità di funzionalità ed è compatibile con tutte le ultime versioni dei principali browser.

4) BASE DATI

La base dati è una componente necessaria per ogni progetto che ha bisogno di far persistere dei dati e in questo capitolo verranno mostrati i passi che hanno portato alla sua creazione.

4.1) Modello Entity-Relationship

La prima fase nella progettazione di una base dati è la fase di progettazione logica in cui, dopo un attento studio del modello di dominio dell'applicativo, si analizza “che cosa” dev'essere rappresentato. L'output di questa fase, visibile nella figura 4.1, è il modello *Entity-Relationship* (ER), uno schema concettuale, composto principalmente da entità, relazioni e attributi:

- Entità: elementi che rappresentano gli oggetti del mondo reale che hanno esistenza autonoma
- Relazione: elemento che rappresenta come le entità sono collegate logicamente tra loro
- Attributi: rappresentano delle qualità che possono avere le entità o le relazioni

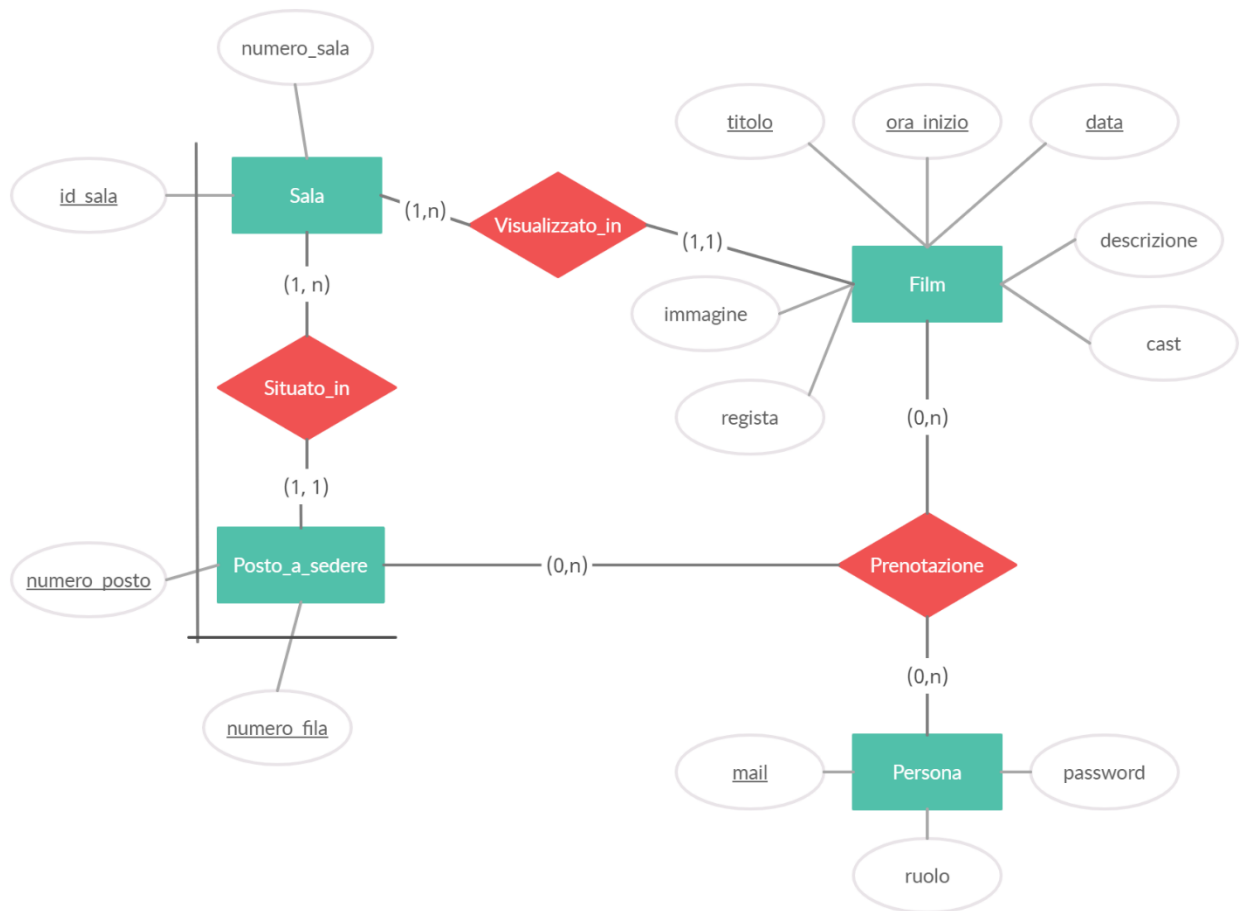


Figura 4.1: Modello ER

4.2) Database MySQL

In seguito alla creazione del modello ER si è passati alla creazione del database vero e proprio utilizzando MySQL, un sistema open source di gestione di database relazionali SQL (Structured Query Language).

Un database relazionale è composto da un insieme di relazioni, ovvero delle tabelle, che contengono tutti i dati. Queste tabelle rappresentano le entità o le relazioni e ognuna di esse ha delle chiavi primarie, ovvero un insieme di uno o più attributi che permettono di identificare univocamente una riga di una tabella. I collegamenti tra le tabelle avvengono per mezzo di queste chiavi primarie per evitare la ripetizione dei dati all'interno del database.

Nella figura 4.2 si può vedere il database realizzato per il progetto.

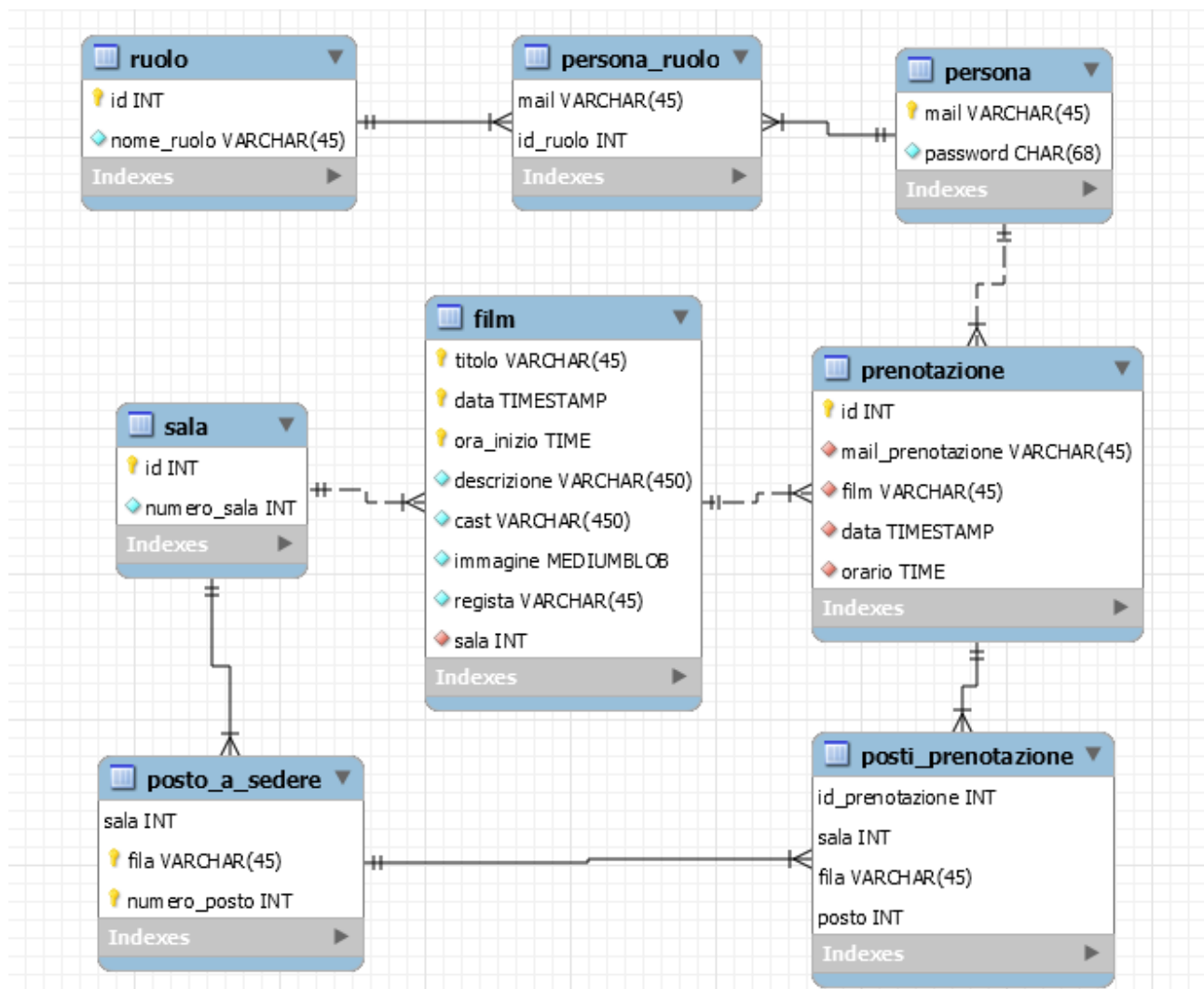


Figura 4.2: Database MySql

5) STRUTTURA DEL PROGETTO JAVA

Di seguito vengono analizzati i packages in cui è stato diviso il codice java, come è mostrato nella figura 5.1.

Tutti i packages sono contenuti nel package principale *com.synclab.cinemamultisala* che contiene anche la classe *CinemaMultisalaApplication* che contiene il metodo main per far partire l'applicazione.

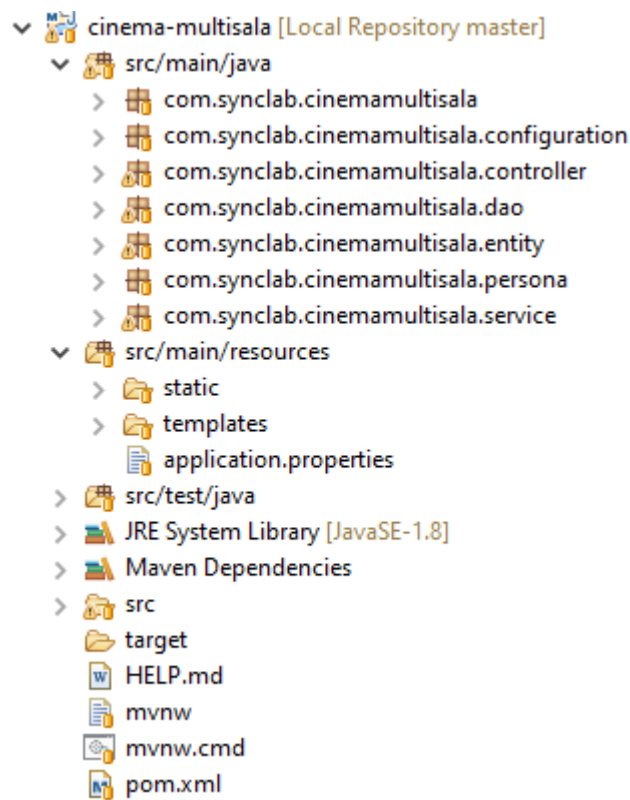


Figura 5.1: Struttura progetto

- *configuration*: contiene solamente due classi, *SecurityConfig* e *SecurityWebApplicationInitializr*, e si occupa di tutta la gestione della sicurezza impostando l'*authentication provider* e il password encoder di bcrypt e configurando i permessi in base al ruolo dell'account per i vari url dell'applicazione
- *controller*: questo package contiene tutti i controller del modello MVC. Sono presenti due classi *AmministratoreController* e *DipendenteController* che si occupano della gestione delle azioni che possono eseguire i rispettivi ruoli e gli altri controller si occupano ciascuno di una sezione dell'applicazione

- *dao*: dao significa Data Access Object, quindi questo package contiene tutte le interfacce che servono per interagire con il database. Ognuna di esse estende l'interfaccia *JpaRepository<entity, primaryKey>* che supporta le operazioni CRUD (Create, Read, Update e Delete) e contiene eventuali query aggiuntive. Di seguito nella figura 5.2 viene mostrato un esempio.

```
public interface FilmRepository extends JpaRepository<Film, FilmId> {

    // Trova tutti i film differenti in programmazione questa settimana
    @Query(value = "SELECT * FROM Film f WHERE f.data >= ?1 AND f.data <= ?2 GROUP BY f.titolo", nativeQuery = true)
    public List<Film> getFilmsFromDayToDay(LocalDate daData, LocalDate aData);
}
```

Figura 5.2: query per ricercare i film in programmazione

- *entity*: in questo package si trovano le classi che rappresentano gli elementi del modello di dominio che sono salvati nel database. Ogni classe possiede i metodi getter e setter e per mezzo delle annotazioni sono mappate con le rispettive tabelle di MySql
- *persona*: questo package contiene un'unica classe, *CrmPersona*, che è una classe di supporto per la validazione della registrazione degli account. Infatti, una volta validati tutti i campi della registrazione (e-mail, password e matching password) un oggetto di tipo *CrmPersona* viene convertito in un oggetto della classe *Persona* per poi venire salvato nel database.
- *service*: contiene tutti i file che forniscono i vari servizi per l'applicazione, sia le interfacce che le relative classi concrete che le implementano. Ogni classe nel package service sfrutta uno o più classi del package *repository* per accedere ai dati persistenti.

6) DETTAGLI DELL'INTERFACCIA UTENTE

In questa sezione vengono mostrate delle schermate, relative all'interfaccia, per ogni funzionalità che l'applicazione offre.

HOMEPAGE

Appena si accede al sito ci si trova sull'Homepage, visibile nella figura 6.1, in cui è possibile vedere un'anteprima dei film disponibili in programmazione nell'arco di una settimana. Nel menù di navigazione in alto sono mostrate le varie opzioni che il visitatore può selezionare e nel caso si effettui un'autenticazione come dipendente comparirà la funzionalità "AGGIUNGI FILM" e nel caso sia invece un amministratore sarà visibile pure "GESTIONE ACCOUNT". In tutte le pagine del sito è possibile tornare all'Homepage cliccando sulla scritta in alto "CINEMAMULTISALA".

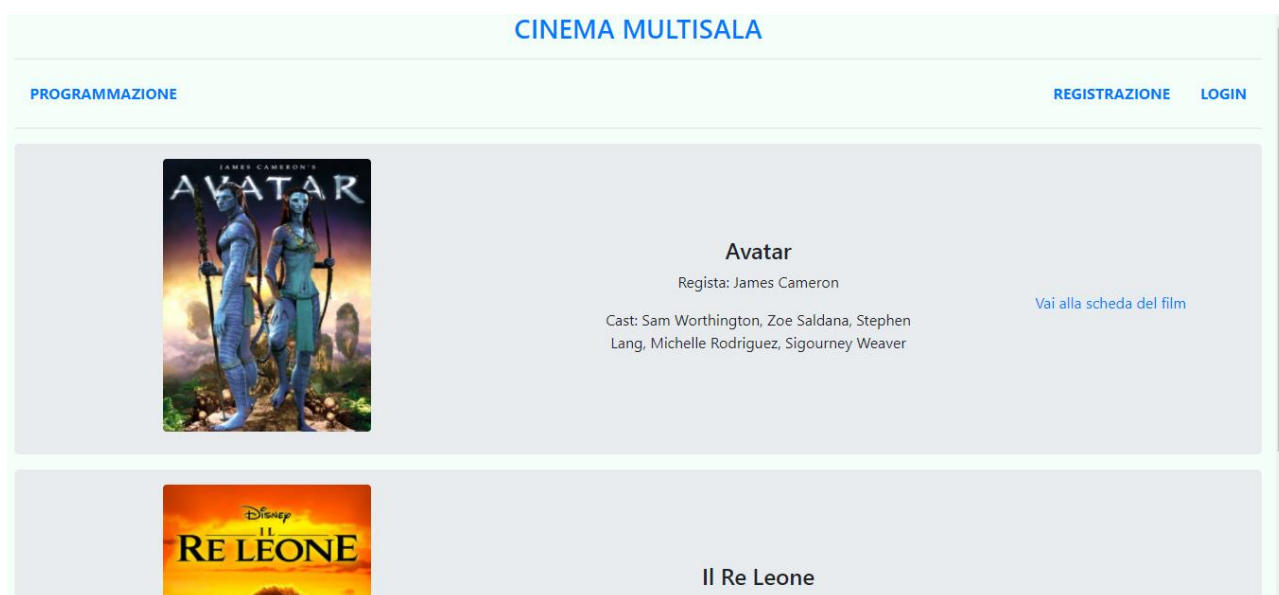


Figura 6.1: schermata Homepage appena si accede al sito

REGISTRAZIONE

In questa pagina, come mostrato nella figura 6.2, è possibile registrare il proprio account decidendo e-mail e password. Di default all'account viene associato il ruolo di utente. Se si è già registrati c'è un link per passare invece alla schermata di login.

CINEMA MULTISALA

Registrazione

Registrati

Se hai già un account effettua il [login](#)

Figura 6.2: schermata di registrazione

LOGIN

In questa pagina, visibile nella figura 6.3, è possibile effettuare il login col proprio account inserendo e-mail e password. Se non si è ancora registrati c'è un link per passare invece alla schermata di registrazione.

CINEMA MULTISALA

Login

Login

Non hai un account? [Registrati](#)

Figura 6.3: schermata di Login

SCHEDA FILM

A questa schermata si accede dall'homepage tramite il link "Vai alla scheda del film" del relativo film di cui si vogliono vedere maggiori informazioni e da qui, come si può notare nella figura 6.4, per un utente è possibile effettuare una prenotazione selezionando il bottone verde relativo all'orario preferito.

The screenshot shows the 'Avatar' movie details page. At the top, there are links for 'PROGRAMMAZIONE' and 'LOGOUT'. The movie poster for 'Avatar' is displayed on the left. To the right, the title 'Avatar' is shown, followed by the director 'Regista: James Cameron' and the cast 'Cast: Sam Worthington, Zoe Saldana, Stephen Lang, Michelle Rodriguez, Sigourney Weaver'. A synopsis follows: 'L'ex marine Jake Sully è stato reclutato per una missione sul pianeta Pandora con lo scopo di recuperare risorse naturali in esaurimento sulla Terra. Inaspettatamente si ritrova a voler proteggere il mondo magico al quale si sente stranamente legato.' Below this, a table lists the showtimes and the booking status for each.

Data	10/6	10/6	10/6	10/6	10/6	10/6	10/6	10/6
Ora	19:00	19:30	20:00	21:00	21:30	22:00	23:00	23:30
Sala	1	3	2	1	3	2	1	3
Prenota	<input type="button" value="x"/>	<input type="button" value="x"/>	<input type="button" value="x"/>	<input type="button" value="x"/>	<input type="button" value="x"/>	<input type="button" value="x"/>	<input type="button" value="x"/>	<input type="button" value="x"/>

Figura 6.4: scheda con il dettaglio di un film

PRENOTAZIONE

In questa schermata, mostrata nella figura 6.5, vengono mostrati i dati del film che si ha intenzione di vedere ed è possibile vedere tutti i posti in sala e il loro stato (occupato, libero o selezionato) per poter scegliere quali prenotare.

The screenshot shows the 'Sala 1 - Avatar' booking page. At the top, there are links for 'PROGRAMMAZIONE' and 'LOGOUT'. The title 'Sala 1 - Avatar' is displayed, followed by the date and time 'Data: 2021-06-10, ora: 19:00'. A legend indicates the status of the seats: 'Occupato' (red square), 'Libero' (green square), and 'Selezionato' (yellow square). Below the legend, the seating chart is shown with two rows, A and B, each with two seats, 1 and 2. All seats are currently green, indicating they are free. A 'Prenota' button is located at the bottom.

Legenda: Occupato ■ Libero ■ Selezionato ■

A 1 2

B 1 2

Figura 6.5: schermata di visualizzazione dei posti liberi in sala per la prenotazione

CONFERMA PRENOTAZIONE

In questa pagina, visibile nella figura 6.6, sono visualizzati tutti i dati riepilogativi della prenotazione che è stata correttamente registrata nel sistema.

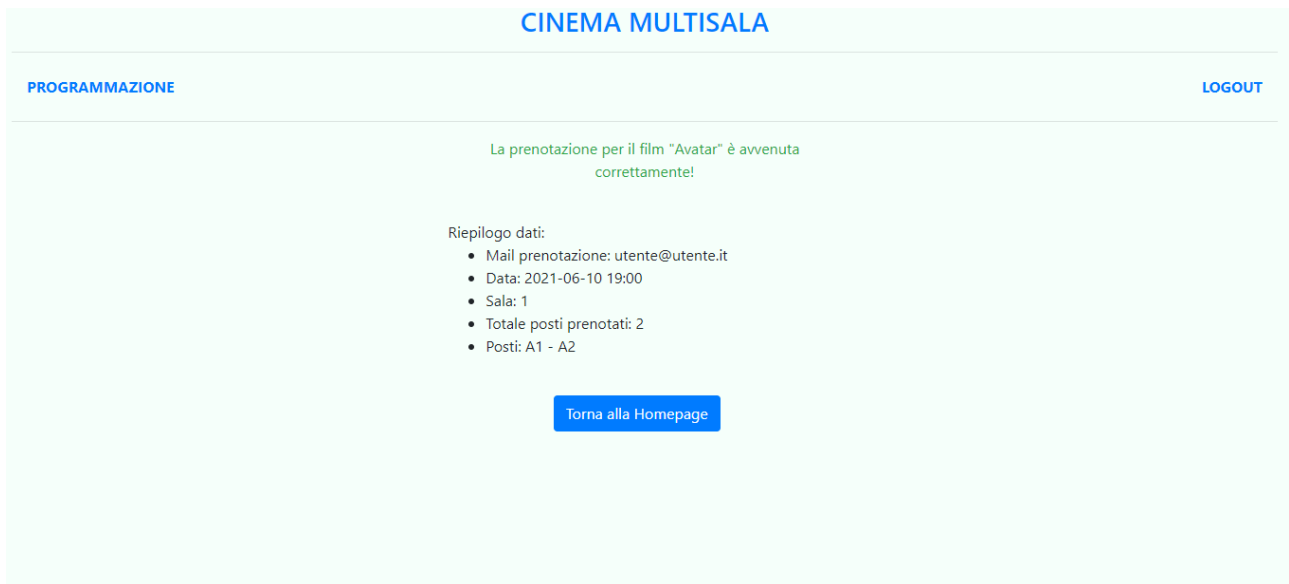


Figura 6.6: schermata di conferma della prenotazione

PROGRAMMAZIONE

In questa pagina è visibile tutta la programmazione nell'arco di una settimana con i dati in dettaglio. Per un utente saranno visibili i pulsanti per effettuare una prenotazione, per un dipendente invece ci sono i comandi per la gestione dei film in programmazione come si può notare dalla figura 6.7.

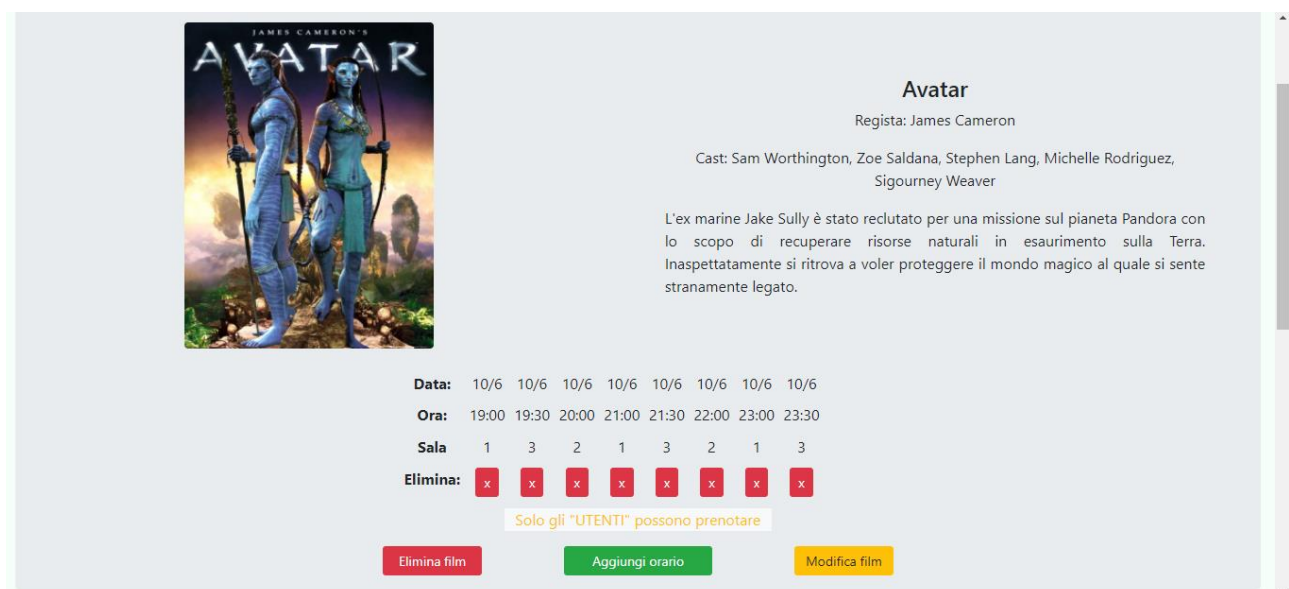


Figura 6.7: schermata della programmazione in dettaglio

AGGIUNTA ORARIO

Da qui è possibile, selezionando una data, un orario e una sala, aggiungere un orario per la visualizzazione di un film (se in tale momento non è già presente un film) come mostrato sotto nella figura 6.8.

The screenshot shows a web interface titled "CINEMA MULTISALA". Below the title, there is a link "Aggiungi un orario per Avatar". The form contains three input fields: a date field with the placeholder "gg/mm/aaaa" and a calendar icon, a time field with the placeholder "--:--" and a clock icon, and a dropdown menu labeled "Seleziona la sala" with the value "1" selected. Below these fields is a blue button labeled "Aggiungi orario". At the bottom of the form is a link "Ritorna alla programmazione".

Figura 6.8: schermata di aggiunta di un nuovo orario per un film

MODIFICA FILM

In questa pagina, mostrata nella figura 6.9, si possono modificare i dati di dettaglio di un film quali titolo, regista, cast, descrizione e l'immagine della locandina.

The screenshot shows a web interface titled "CINEMA MULTISALA". At the top, there are two navigation links: "PROGRAMMAZIONE" and "AGGIUNGI FILM", and a "LOGOUT" link on the right. Below the navigation bar, there is a section titled "Modifica Film". On the left side of this section, there is a placeholder for the movie poster labeled "Immagine locandina" with the text "Scegli file" and "Nessun file selezionato". On the right side, there are four form fields: "Titolo" (Avatar), "Regista" (James Cameron), "Cast" (Sam Worthington, Zoe Saldana, Stephen Lang, Michelle Rodriguez, Sigourney Weaver), and "Descrizione" (L'ex marine Jake Sully è stato reclutato per una missione sul pianeta Pandora con lo scopo di recuperare risorse naturali in esaurimento sulla Terra. Inaspettatamente si ritrova a voler proteggere il mondo magico al quale si sente stranamente legato.). Below the form fields is a blue button labeled "MODIFICA".

Figura 6.9: schermata per la modifica dei dati di dettaglio relativi ad un film

NUOVO FILM

Da questa schermata, visibile nella figura 6.10, è possibile aggiungere un nuovo film, inserendo tutti i dati necessari, al database.

The screenshot shows a web form titled "Aggiungi un film alla collezione". It contains several input fields: "Titolo", "Regista", "Cast", and a larger text area for "Aggiungi una descrizione". Below these is a section for "Carica l'immagine della locandina" with a "Scegli file" button and the text "Nessun file selezionato". There are also date pickers for "gg/mm/aaaa" and "--:--". At the bottom, there is a "Seleziona la sala" dropdown menu showing "1" and a large blue "Aggiungi" button.

Figura 6.10: schermata per la modifica dei dati di dettaglio relativi ad un film

GESTIONE ACCOUNT

In questa pagina un amministratore può visualizzare la lista di tutti gli account registrati nel sistema e, per ognuno di essi, può eliminarli oppure modificare i ruoli associati a ciascuno, come mostrato nella figura 6.11. Da qui è possibile pure accedere ad una schermata per la creazione di un account potendo selezionare anche i ruoli.

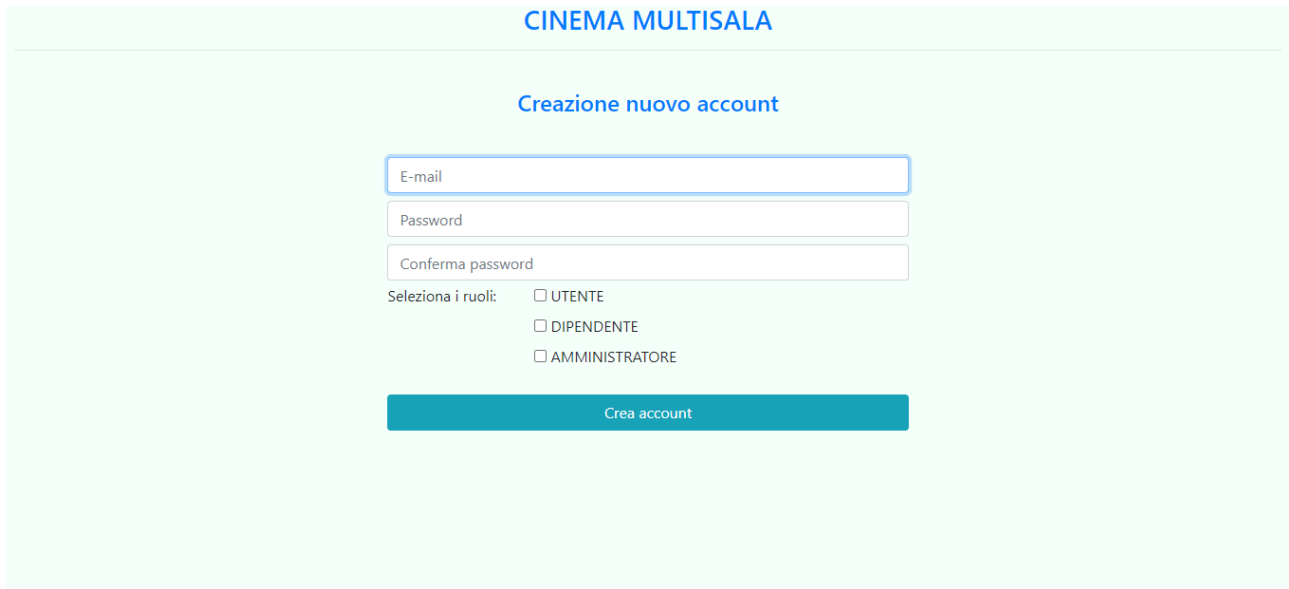
The screenshot shows a page titled "CINEMA MULTISALA" with a sub-header "Account registrati". There is a blue button labeled "Aggiungi Utente". Below it is a table with three columns: "#", "Mail", "Ruoli", and "Azioni". The table lists three accounts: an administrator, a dependent, and a user, each with "Aggiorna" and "Elimina" action buttons.

#	Mail	Ruoli	Azioni
1	danielescalco@hotmail.it	AMMINISTRATORE	<button>Aggiorna</button> <button>Elimina</button>
2	dipendente@dipendente.it	DIPENDENTE	<button>Aggiorna</button> <button>Elimina</button>
3	utente@utente.it	UTENTE	<button>Aggiorna</button> <button>Elimina</button>

Figura 6.11: schermata di visualizzazione degli account registrati nel sistema

AGGIUNTA UTENTE

In questa schermata un amministratore può creare un nuovo account e, a differenza della pagina di registrazione, qua è possibile selezionare i vari ruoli voluti, come si può vedere dalla figura 6.12.



CINEMA MULTISALA

Creazione nuovo account

E-mail

Password

Conferma password

Seleziona i ruoli:

- ☐ UTENTE
- ☐ DIPENDENTE
- ☐ AMMINISTRATORE

Crea account

Figura 6.12: schermata di creazione di un nuovo account

7) CONCLUSIONI E POSSIBILITA' DI SVILUPPI FUTURI

In conclusione, il progetto soddisfa tutte le specifiche richieste, che sono state modificate nel tempo laddove sono sorti dubbi o domande, ma è comunque possibile effettuare molti aggiornamenti e miglioramenti. Tra le varie modifiche ipotizzabili è possibile aggiungere la possibilità di recuperare la password, la conferma di registrazione del proprio account tramite mail, inserire la logica per effettuare i pagamenti quando si prenota. È sicuramente possibile migliorare la grafica dell'applicazione e sarebbe carino introdurre una pagina in cui è possibile visualizzare le informazioni riguardanti il cinema, quali orari di apertura, indirizzo, informazioni di contatto e delle immagini del posto. Se il cinema dovesse diventare una catena si potrebbe anche inserire una pagina in cui sono visualizzate tutte le sedi da cui si potrebbe accedere ad ognuna di esse.

Infine, sarebbe molto utile sviluppare la relativa applicazione mobile dato la larga diffusione dei dispositivi smartphone.