UNIVERSITÀ DEGLI STUDI DI PALERMO

Metodi Avanzati per la Programmazione

Gestionale cinema

Studenti:

Giorgio Prestigiacomo
Francesco Ippolito

Docente:

Prof. Gabriele Fici

Abstract

Il seguente progetto delineato mira allo sviluppo di un software per la gestione delle tipiche operazioni relative al funzionamento generale di un cinema: visualizzazione dei film disponibili da parte degli utenti con relativa disponibilità di prenotazione; gestione di prenotazioni o convalida di biglietti da parte dei dipendenti; inserimento di nuove proposte di film per i clienti o gestione dei dipendenti da parte degli amministratori.

Il sistema in oggetto è indirizzato al supporto gestionale di un cinema. Questo ha come scopo quello di agevolare gli utenti nella scelta e nella prenotazione di proiezioni dei film al cinema.
Tale sistema potrà essere utilizzato dagli utenti, dagli amministratori e dai dipendenti.

Le fasi affrontate per lo sviluppo del software sono le seguenti:
comprensione del dominio applicativo;
analisi dei requisiti funzionali e non funzionali;
implementazione dei diagrammi casi d'uso;
delineazione dei design pattern utili ai fini implementativi e diacussione su eventuali refactoring funzionali;
implementazione di diagrammi di classe;
inizio della fase di implementazione del codice;
inizio della fase refactoring funzionale;

Il software è stato pensato per essere utilizzato anche da chi possiede poca dimestichezza con l'uso di un computer.

Strumenti utilizzati per lo sviluppo

- Tramite vsCode è stata utilizzata l'estensione draw.io per disegnare i vari diagrammi.
- È stato utilizzato IntelliJ per lo sviluppo e per generare i diagramma delle classi.
- È stato utilizzato Github per poter condividere il codice e agevolare lo sviluppo in coppia.

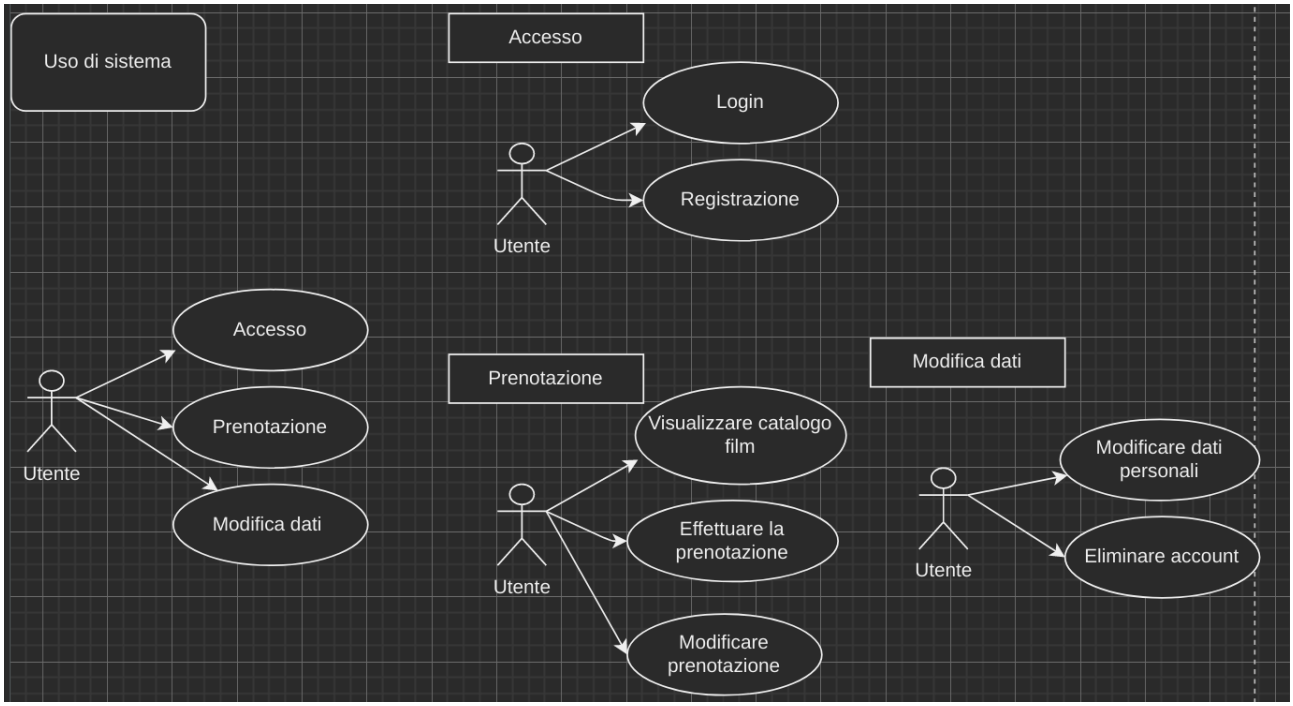
Diagrammi dei Casi d’uso

Casi d’uso utente

Nome caso d’uso	Accesso
ID	1
Attori	utente
Descrizione	Il seguente caso d’uso tratta la funzionalità di accesso al sistema. L’utente tramite credenziali, email e password, potrà accedere al sistema o non avendo a disposizione delle credenziali, potrà registrarsi.

Nome caso d’uso	Prenotazione
ID	2
Attori	utente
Descrizione	Il seguente caso d’uso tratta la funzionalità della prenotazione di un film da parte di un utente, visualizzare il catalogo dei film o modificare la propria prenotazione.

Nome caso d’uso	Modifica dati
ID	3
Attori	utente
Descrizione	Il seguente caso d’uso tratta la funzionalità di modifica dati. L’utente potrà modificare i propri dati personali, la password o eliminare il proprio account.

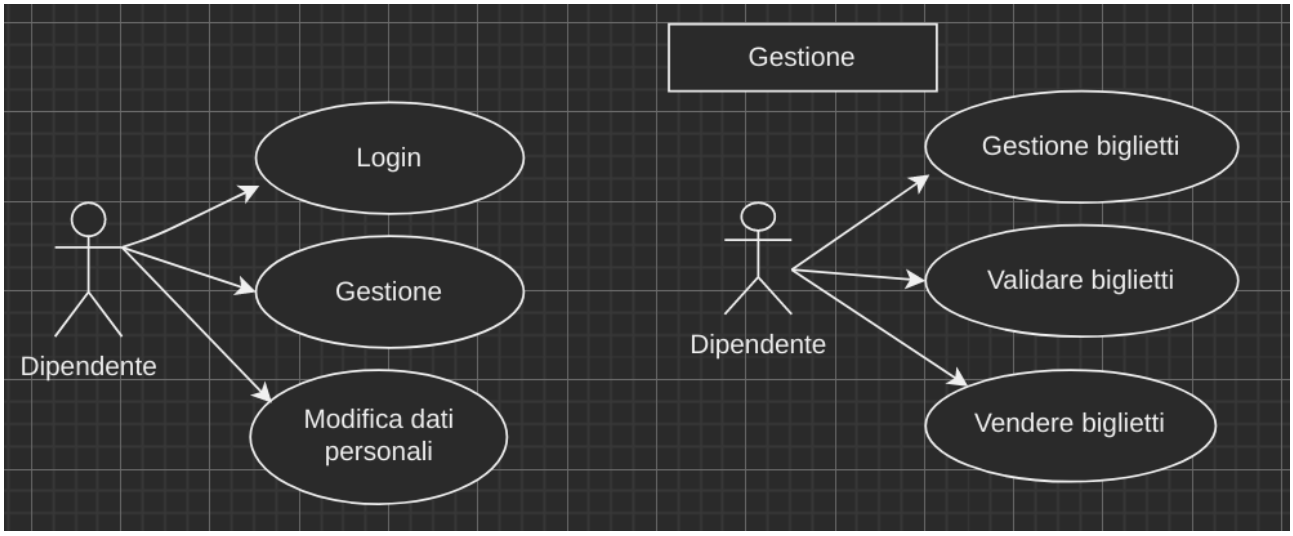


Casi d’uso dipendente

Nome caso d’uso	Login
ID	1
Attori	dipendente
Descrizione	Il seguente caso d’uso tratta la funzionalità di accesso al sistema. Il dipendente tramite credenziali fornite dall’amministratore, potrà accedere al sistema.

Nome caso d’uso	Gestione
ID	2
Attori	dipendente
Descrizione	Il seguente caso d’uso descrive quali funzionalità dipendono dal dipendente. Il dipendente potrà validare biglietti e vendere i biglietti

Nome caso d’uso	Modifica dati personali
ID	3
Attori	dipendente
Descrizione	Il seguente caso d’uso tratta la funzionalità di modificare i propri dati. Il dipendente potrà modificare i propri dati personali.

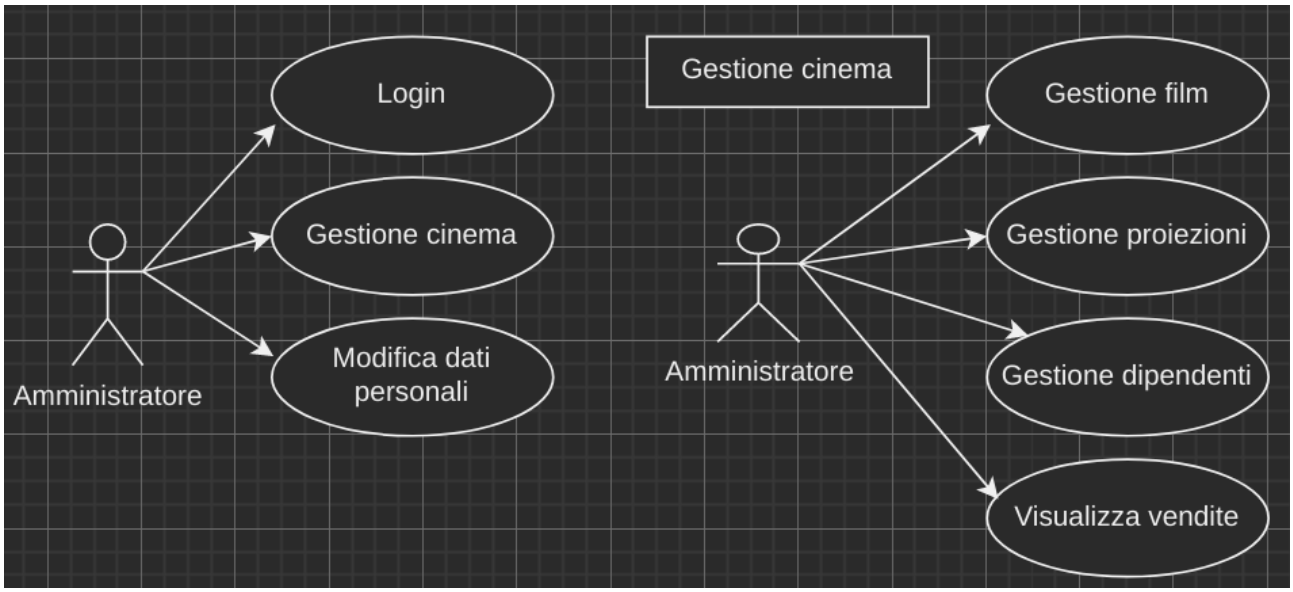


Casi d’uso amministratore

Nome caso d’uso	Login
ID	1
Attori	amministratore
Descrizione	Il seguente caso d’uso tratta la funzionalità di accesso. L’amministratore tramite credenziali, potrà accedere al sistema.

Nome caso d’uso	Gestione cinema
ID	2
Attori	amministratore
Descrizione	Il seguente caso d’uso tratta le funzionalità relative alla gestione del cinema da parte dell’amministratore. L’amministratore potrà gestire i film, gestire la proiezione, gestire i dipendenti e visualizzare le vendite

Nome caso d’uso	Modifica dati personali
ID	3
Attori	amministratore
Descrizione	Il seguente caso d’uso tratta la funzionalità di modificare i propri dati. L’amministratore potrà modificare i propri dati personali.



Analisi dei requisiti

Sono stati delineati i seguenti requisiti funzionali e non funzionali del sistema:

I requisiti funzionali sono:

L'utente potrà effettuare le seguenti operazioni:
- la registrazione e il login;
- la prenotazione della proiezione del film:
  - scegliere posti;
- visualizzare o modificare le prenotazioni effettuate;
- visualizzare la capienza delle sale;
- visualizzare il catalogo delle proiezioni (attuali e future);
- gestire il proprio account:
  - eliminare l'account,
  - modificare le credenziali.

L'amministratore potrà effettuare le seguenti operazioni:
- login;
- gestire il proprio account;
- visualizzare e modificare i cataloghi dei film;
- visualizzare il ricavato totale di un film in un determinato periodo di tempo;
- gestire i dipendenti: registrare, cancellare, promuovere o demansionare i dipendenti.

Il dipendente potrà effettuare le seguenti operazioni:
- login;
- visualizzare i prezzi dei biglietti delle proiezioni;
- visualizzare i posti disponibili;
- visualizzare l'elenco dei prenotati tramite piattaforma;
- gestire la vendita di biglietti ad utenti fisici:
  - selezionare la proiezione,
  - assegnare i posti;
- validare tramite codice il biglietto elettronico;
- gestire eventuali errori:
  - modificare i posti assegnati,
  - annullare i biglietti generati erroneamente;
- gestire il proprio account;

I requisiti non funzionali sono:

l'utente non potrà effettuare più registrazioni con la medesima email;
l'utente non potrà prenotare un biglietto per la proiezione in una sala piena;
l'utente non potrà eliminare il proprio account se ha una prenotazione attiva;
il dipendente non potrà modificare i posti prenotati dagli utenti;
il dipendente non potrà vendere un biglietto per una proiezione se la sala è piena;
ogni dipendente verrà registrato sulla piattaforma dell'amministratore;
effettuando una prenotazione, l'utente riceverà un codice da mostrare ai dipendenti per poter accedere alla proiezione del film scelto;
Solamente un amministratore potrà eliminare dipendenti o altri amministratori.
Solamente un amministrare potrà cambiare il ruolo di dipendenti o altri amministratori.

Diagramma delle classi

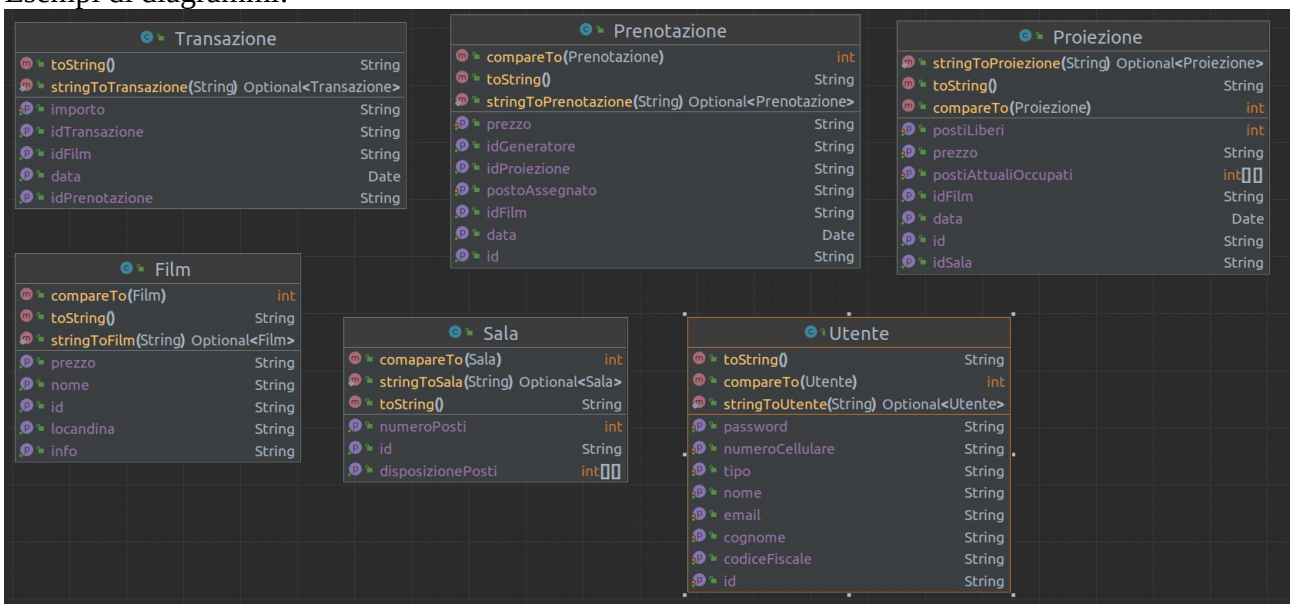
Il diagramma delle classi è stato generato tramite l'ambiente di sviluppo IntelliJ ultimate.

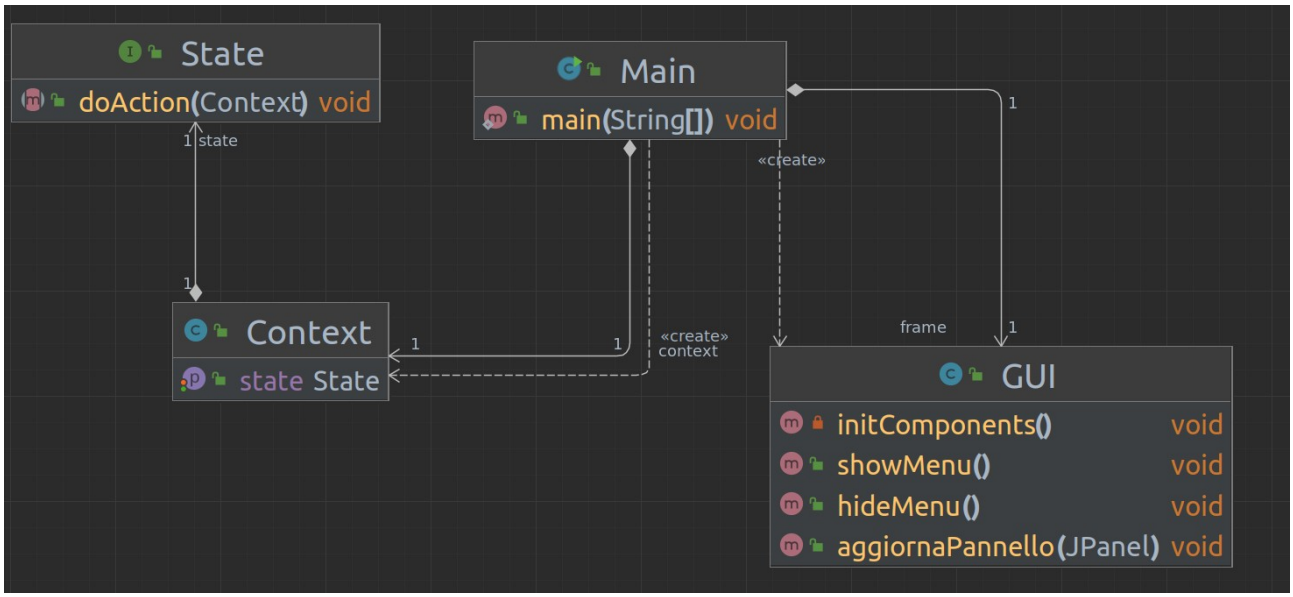
All'interno della cartella UML, nel file UML.drawio, viene delineato il diagramma delle classi realizzato prima della fase implementativa, utilizzato come input per aver un'idea sulla struttura del progetto.

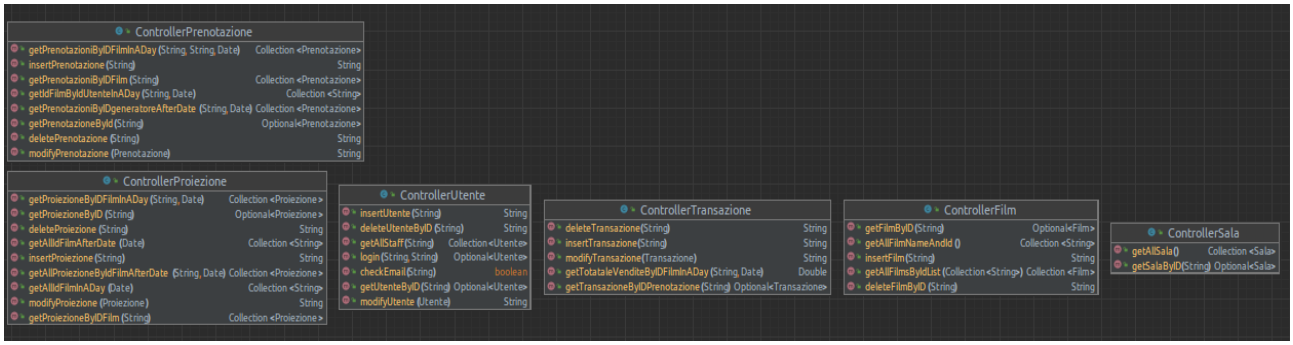
Per analizzare i diagrammi delle classi del progetto è possibile prenderne visione tramite il generatore automatico di IntelliJ (versione Ultimate).

- In allegato un link tutorial: https://www.youtube.com/watch?v=66ArYQUFLdM

Esempi di diagrammi:






ControllerPrenotazione
getPrenotazioniByIDFilmInADay (String, String, Date) Collection <Prenotazione>
insertPrenotazione (String) String
getPrenotazioniByIDFilm (String) Collection <Prenotazione>
getIdFilmByIdUtenteInADay (String, Date) Collection <String>
getPrenotazioniByIDgeneratoreAfterDate (String, Date) Collection <Prenotazione>
getPrenotazioneById (String) Optional<Prenotazione>
deletePrenotazione (String) String
modifyPrenotazione (Prenotazione) String
ControllerProiezione
getProiezioneByIDFilmInADay (String, Date) Collection <Proiezione>
getProiezioneByID (String) Optional<Proiezione>
deleteProiezione (String) String
getAlIIdFilmAfterDate (Date) Collection <String>
insertProiezione (String) String
getAllProiezioneByIdFilmAfterDate (String, Date) Collection <Proiezione>
getAlIIdFilmInADay (Date) Collection <String>
modifyProiezione (Proiezione) String
getProiezioneByIDFilm (String) Collection <Proiezione>
ControllerUtente
insertUtente (String) String
deleteUtenteByID (String) String
getAllStaff (String) Collection <Utente>
login (String, String) Optional<Utente>
checkEmail (String) boolean
getUtenteByID (String) Optional<Utente>
modifyUtente (Utente) String
ControllerTransazione
deleteTransazione (String) String
insertTransazione (String) String
modifyTransazione (Transazione) String
getTotataleVenditeByIDFilmInADay (String, Date) Double
getTransazioneByIDPrenotazione (String) Optional<Transazione>
ControllerFilm
getFilmByID (String) Optional<Film>
getAllFilmNameAndId () Collection <String>
insertFilm (String) String
getAllFilmsByIdList (Collection <String>) Collection <Film>
deleteFilmByID (String) String
ControllerSala
getAllSala() Collection <Sala>
getSalaByID (String) Optional<Sala>

Descrizione del Software

Sulla base dei requisiti analizzati e dal soggetto che accede al sistema, il software assume comportamenti differenti.
I soggetti delineati sono tre: utente, dipendente ed amministratore.

Nel sistema un amministratore è registrato di default, il quale avrà il compito di registrare i dipendenti o promuovere amministratore.

- L'amministratore, accedendo con le proprie credenziali, potrà navigare sui seguenti pannelli:
Film;
Visualizza vendite;
Visualizza dipendenti;
Inserimento Film;
Profilo.

Nel pannello "Film" si avrà la possibilità di visualizzare tutti i film disponibili, i quali potranno essere eliminati o si potrà visualizzare per film le relative proiezioni.
Cliccando su "Proiezioni" l'amministratore visualizzerà le proiezioni ed eventualmente potrà modificare il prezzo della singola proiezione, la seguente modifica del prezzo non farà variare il prezzo delle prenotazioni già effettuate.

Nel pannello "Visualizza vendite" si avrà la possibilità di verificare la vendita complessiva per film selezionato, specificando il periodo di interesse.

Nel pannello "Visualizza dipendenti", l'amministratore potrà visualizzare tutti i dipendenti assunti e avrà la possibilità di promuoverli, licenziarli o demansionare eventuali amministratori.
Cliccando su inserisci dipendente, l'amministratore potrà inserire un nuovo dipendente, riempiendo i campi obbligatori.

Nel pannello "Inserimento Film" l'amministratore potrà inserire un nuovo film compilando i campi obbligatori.

Nel pannello "Profilo" si potranno visualizzare le proprie credenziali ed eventualmente modificare password e dati personali.

L'amministratore cliccando su "Logout" verrà disconnesso dal sistema e reindirizzato nel pannello "Login"

- Un nuovo dipendente non potrà personalmente effettuare la registrazione. La registrazione sarà effettuata dall'amministratore.

Un dipendente, accedendo con le proprie credenziali potrà navigare sui seguenti pannelli:
Film;
Gestione Biglietti;
Convalida biglietti;
Profilo.

Nel pannello "Film" si avrà la possibilità di visualizzare tutti i film disponibili in data odierna.
Per ogni singolo Film, cliccando su Proiezioni, il dipendente visualizzerà le proiezioni e potrà effettuare la prenotazione scegliendo tra i posti disponibili.

Nel pannello “Gestione Biglietti” il dipendente potrà, visualizzare i biglietti per film e potrà modificare o cancellare i biglietti.

Nel pannello “Convalida biglietti”, il dipendente potrà convalidare i biglietti degli utenti inserendo il codice del biglietto fornito durante la prenotazione.

Nel pannello “Profilo” il dipendente potrà visualizzare i propri dati ed eventualmente modificare dati e password personali.

Il dipendente cliccando su “Logout” verrà disconnesso dal sistema e reindirizzato nel pannello “Login”.

- L’utente all’avvio del software avrà la possibilità di poter accedere al sistema, inserendo le proprie credenziali, email e password, o non avendo a disposizione le credenziali potrà registrarsi cliccando sul bottone Registrazione.

Avendo cliccato il tasto Registrazione l’utente dovrà inserire le seguenti credenziali:
nome, cognome, e-mail, codice fiscale, numero di cellulare, password.

L’utente dovrà rispettare delle regole per poter concludere con successo la registrazione.
Una volta validati i campi da parte del sistema, l’utente potrà concludere la registrazione con successo.

Un utente, accedendo con le proprie credenziali potrà navigare sui seguenti pannelli:
Film;
Prenotazioni;
Profilo.

Nel pannello “Film” si avrà la possibilità di visualizzare tutti i film disponibili.
Per ogni singolo film, si potranno visualizzare le proiezioni cliccando il tasto Proiezioni.
Cliccando su Proiezioni l’utente verrà reindirizzato su un nuovo pannello che mostra tutte le proiezioni disponibili.
Ogni proiezione mostrerà la data con l’ora di inizio, il prezzo, la sala e i posti disponibili.

Cliccando sul tasto “Prenota” l’utente verrà reindirizzato su un nuovo pannello mostrando i vari posti disponibili o occupali. L’utente stesso potrà selezionare il posto che desidera tra i posti disponibili.
Effettuata la prenotazione l’utente entrerà in possesso di un codice identificativo del biglietto da mostrare ad un dipendente nel momento in cui si recherà al cinema.

Nel pannello “Prenotazioni” si potranno visualizzare tutte le prenotazioni effettuate.
Una volta effettuata la prenotazione si potrà modificare o cancellare.

Nel pannello “Profilo” l’utente potrà visualizzare le relative informazioni personali e se si desidera potranno essere modificate.
Potranno essere modificati i dati personali o la password e si potrà eliminare l’account.

Gestione del Database

I dati vengono memorizzati in file “.csv”. Sono stati delineati i seguenti file:

Nome file: film.csv
Header del csv: IDFilm, nome, locandina, info, prezzo

Nome file: prenotazione.csv
Header del csv: IDPrenotazione, IDGeneratore, IDProiezione, IDFilm, data, prezzo, postoAssegnato

Nome file: proiezione.csv
Header del csv: IDProiezione, IDFilm, IDsala, prezzo, data, postiLiberi, disposizionePosti

Nome file: sala.csv
Header del csv: IDsala, numeroPosti, disposizionePosti

Nome file: transazione.csv
Header del csv: IDtransazione, IDPrenotazione, IDFilm, data, importo

Nome file: utente.csv
Header del csv: IDUtente, tipo, nome, cognome, e-mail, numeroCellulare, codiceFiscale, passwordUtente

Design Pattern utilizzati

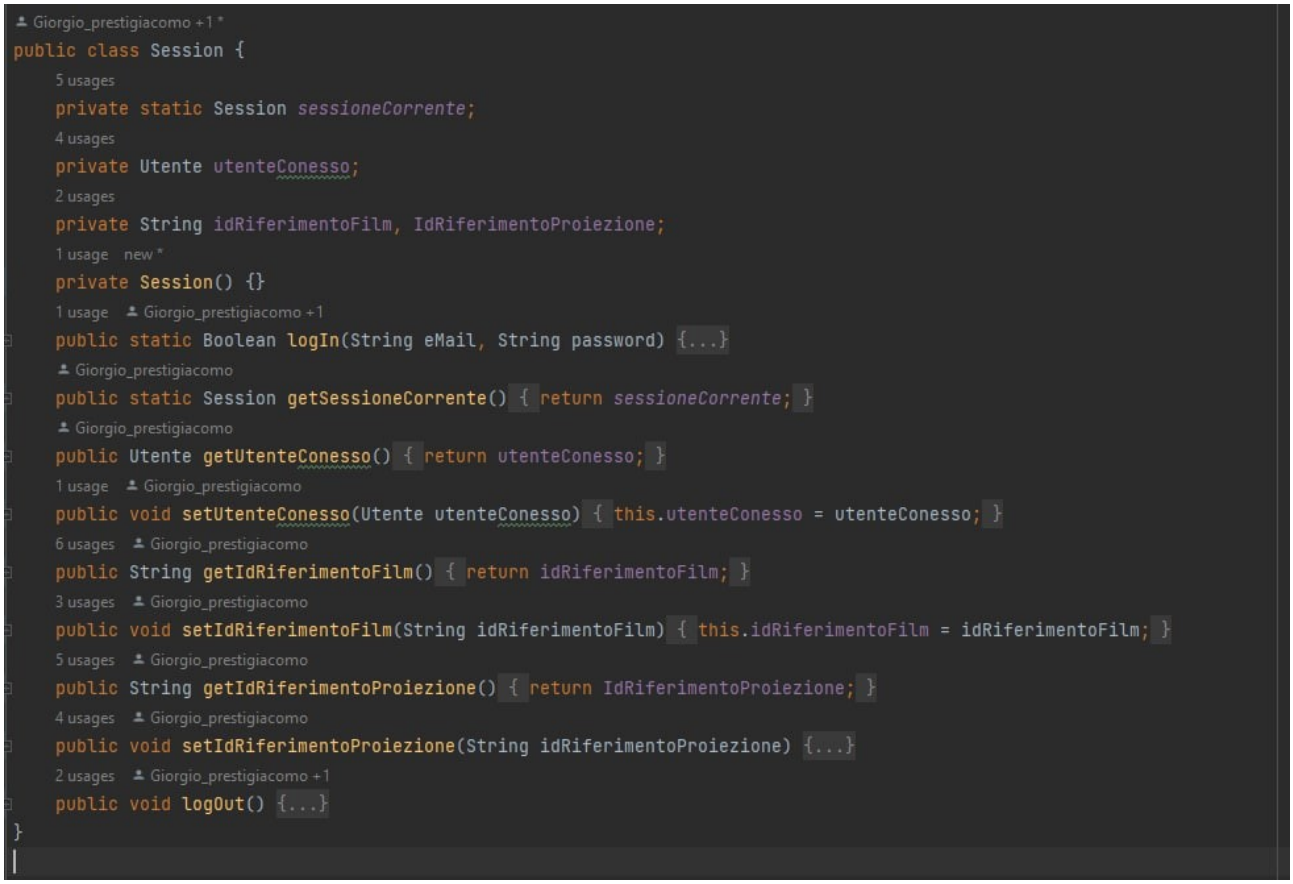
Alla base del progetto è stato delineato il design pattern Model View Controller.

Il model viene utilizzato per gestire il nostro oggetto all'interno del programma.
Il controller viene utilizzato per poter interagire con il database, estraendo le informazioni che il database rilascia, condividendole alla view.
La view permette di utilizzare le funzionalità delle interfacce grafiche per utilizzare le funzionalità dei vari controller.

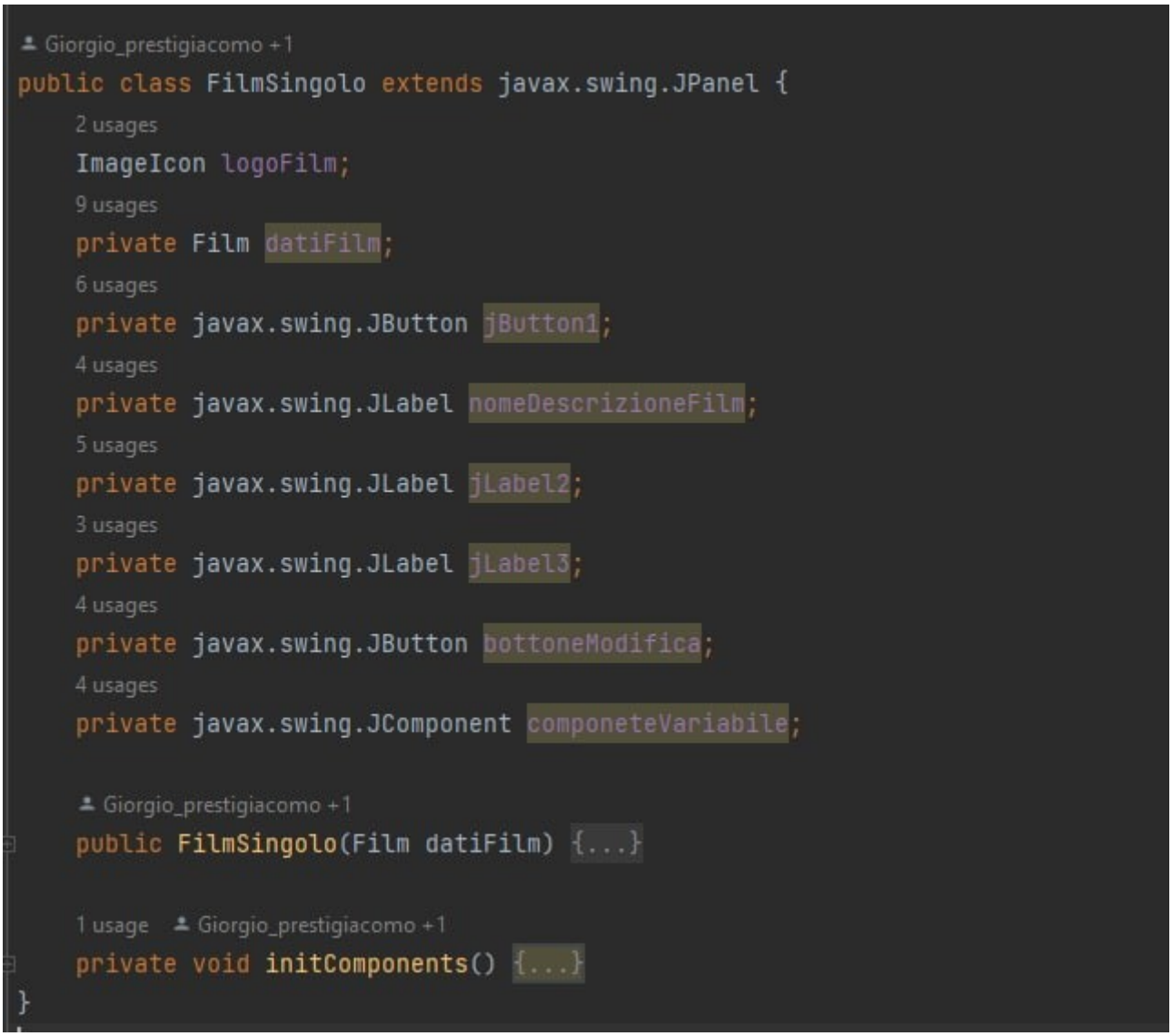
Il vantaggio dell'utilizzo di questo pattern sta nella separazione dei ruoli agevolando la leggibilità del codice, migliorando la gestione delle classi.

- design pattern Creazionali

Singleton
utilizzato per non permettere la creazione di oggetti multipli della stessa classe perché questo causerebbe problemi. Viene utilizzato nella gestione della sessione corrente.

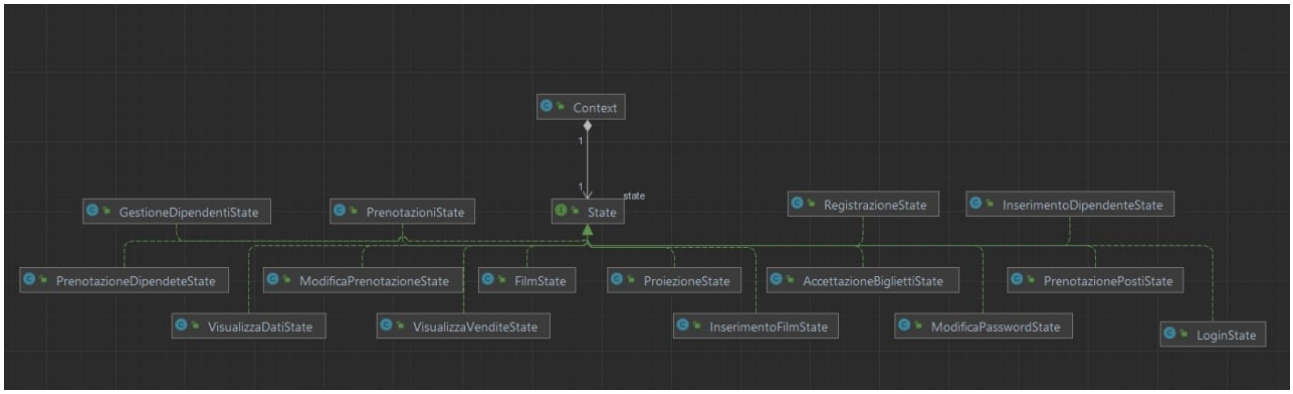


Factory
Il seguente pattern consente di delegare la costruzione di un oggetto ad una classe che racchiude tutta la logica complessa della costruzione.
E’ stato utilizzato per gestire gli elementi grafici ossia FilmSingolo, ProiezioneSingola ecc...



- design pattern Comportamentali

State
Il seguente pattern è stato utilizzato per la gestione delle interfacce delineando un modo standardizzato di passare da una interfaccia all'altra.



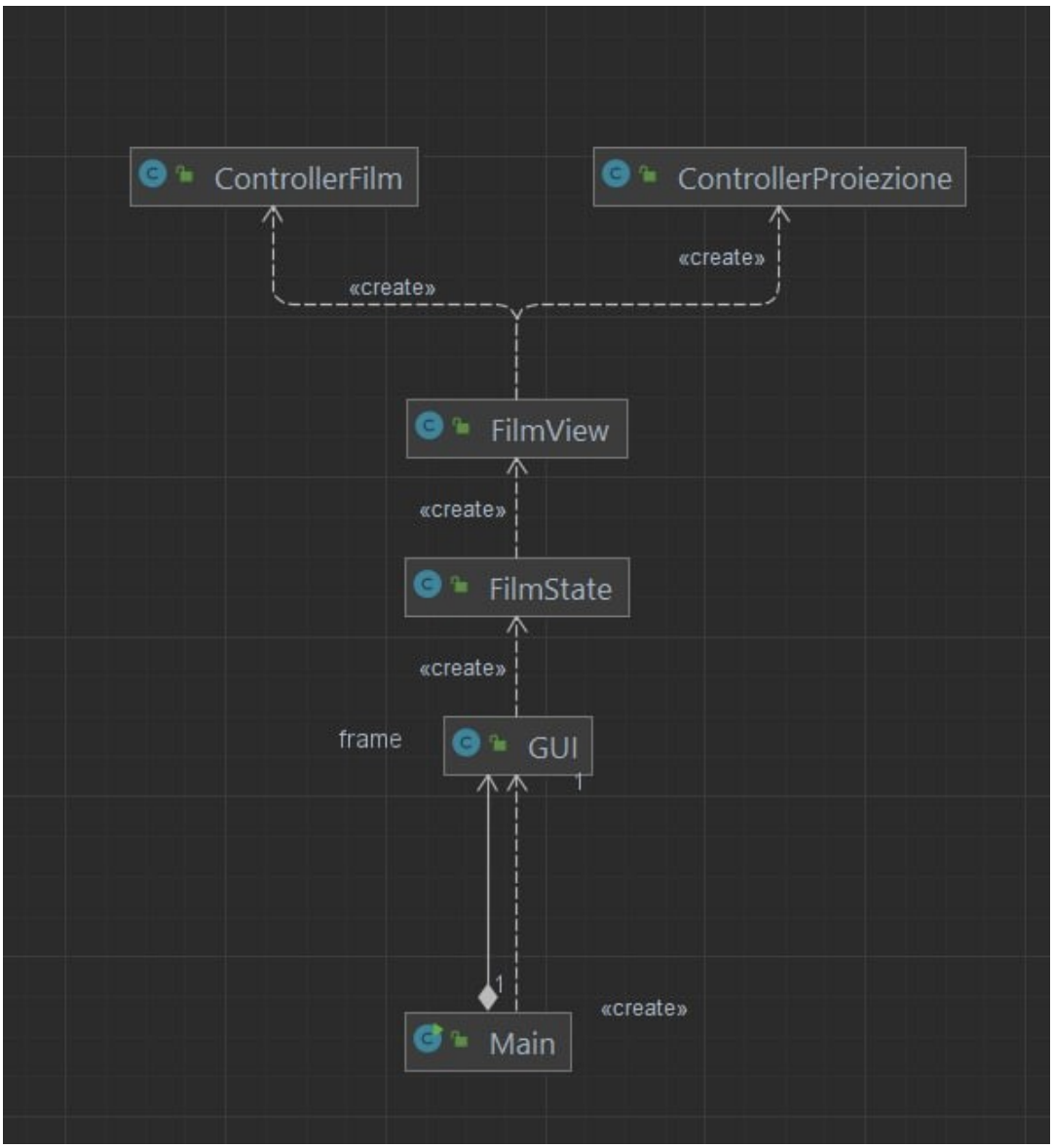

```
Franc3sco07 +2
public class FilmState implements State {
    Franc3sco07 +2
    @Override
    public void doAction(Context context) {
        context.setState(this);
        Session.getSessioneCorrente().setIdRiferimentoProiezione(null);
        Session.getSessioneCorrente().setIdRiferimentoFilm(null);
        Main.frame.aggiornaPannello(new FilmView());

    }
}
```

Observer
Analizzando gli aspetti base della gestione degli eventi di Java, abbiamo dedotto che tutti i vari ascoltatori rispettano la logica del design pattern observer.

- design pattern Strutturali

Façade
Il seguente pattern permette di togliere complessità all'utilizzo del programma, tale pattern viene usato quando viene implementata una GUI, poiché di sua natura semplifica l'accesso al programma.

In questo esempio FilmView funge da Façade per i controller come tutte le varie interfacce.



Decorator
Il seguente pattern consente di aggiungere nuove funzionalità ad oggetti già esistenti.
Utilizzato nelle interfacce view che gestiscono la visualizzazione delle diverse funzionalità dei diversi tipi di utente.

Esempi di refactoring funzionale

Dove è stato possibile è stato effettuato il refactoring funzionale, di seguito vengono riportati parti di codice:

```
3 usages  Giorgio_prestigiacomo +1
public Collection<Film> getAllFilmsByIdList(Collection<String> idFilms) {
    Optional<BufferedReader> optionalBufferedReader = Gestione_db.getTable(tableName);
    if (optionalBufferedReader.isPresent()) {
        BufferedReader in = optionalBufferedReader.get();
        return in.lines().parallel()  Stream<String>
                .map(s -> Film.stringToFilm(s))  Stream<Optional<Film>>
                .filter(s -> s.isPresent())
                .map(s -> s.get())  Stream<Film>
                .filter(x -> idFilms.contains(x.getId()))
                .toList();
    } else {
        return new ArrayList<>();
    }

}
```

```
filmDisponibili.stream()  Stream<Film>
        .sorted(Film::compareTo)
        .map(s -> generazioneFilm(s))  Stream<JPanel>
        .forEach(s -> infoPannello.add(s));
```

```
public static Boolean readExceptID(String ID, String path, Collection<String> dati) {
    try {
        BufferedReader file = GestioneFile.openFile(path);
        return dati.addAll( file.lines().parallel()
                .filter(s -> !s.split( regex: ",")[0].equals(ID))
                .toList()
        );
    } catch (FileNotFoundException e) {
    } catch (IOException e) {
    }
    return false;
}
```

Per il corretto funzionamento corretto degli stream, le nostre funzioni restituiscono oggetti Optional invece di aver la possibilità di restituire elementi nulli.

```java
public static Optional<Film> stringToFilm(String filmString) {
    String[] datiFilm = filmString.split( regex: ",");
    if (datiFilm.length > 2) {
        Film elemento = new Film(datiFilm[0], datiFilm[1], datiFilm[2], datiFilm[3], datiFilm[4]);
        return Optional.of(elemento);
    }
    return Optional.empty();
}
```