

CORSO DI LAUREA IN INGEGNERIA E SCIENZE

INFORMATICHE – UNIVERSITA' DI BOLOGNA

CORSO DI PROGRAMMAZIONE A OGGETTI

A. A. 2016-2017

**RELAZIONE SULLO SVILUPPO DI UNA
APPLICAZIONE SIMIL – GESTIONALE**

TEATRO

Carloni Ilaria 0000653038

Indice

1	Analisi	
1.1	Requisiti	3
1.2	Analisi e modello del dominio	4
2	Design	
2.1	Architettura	5
2.2	Design dettagliato	7
3	Commenti finali	
3.1	Autovalutazione e lavori futuri	13
	Appendice A	
	Guida utente	14
	Note	16

Capitolo 1

Analisi

1.1 Requisiti

Lo scopo di questo progetto è di gestire l'assegnazione dei posti di una sala teatrale a seconda dei vari spettacoli.

Le funzionalità da realizzare sono le seguenti:

-Lato cliente:

-Selezionare spettacolo

-Prenotare posto/i

-Lato operatore:

-Aggiungere spettacolo

-Visualizzare spettacoli correnti

-Visualizzare storico spettacoli

-Prenota posto/i

Il sistema prevede la suddivisione tra cliente e operatore tramite registrazione e login.

I vari clienti/operatori/spettacoli/prenotazioni vengono memorizzati in modo persistente sul sistema attraverso il file system, in modo trasparente all'utente.

1.2 Analisi e modello del dominio

L'applicazione dovrà fornire un servizio di prenotazione riguardante il numero di posti dello spettacolo scelto, nella sala del teatro.

I soggetti coinvolti sono i clienti e gli operatori, che hanno accesso al sistema tramite login. Il sistema di login prevede che il cliente e l'operatore possano avere lo stesso nome e la stessa password, ma due clienti, come due operatori, non li possono avere uguali tra loro.

Prima di accedere un cliente deve essere registrato correttamente. In seguito potrà prenotare uno spettacolo aprendo l'apposita finestra "Visualizza", eventualmente cercando il nome dello spettacolo desiderato oppure selezionandolo dalla lista (inserita in precedenza dall'operatore) ed inserendo il numero di posti desiderato, che dovrà essere un intero positivo, altrimenti verrà generata l'eccezione: "Inserire un numero positivo". In caso di mancanza di posti disponibili residui, invece, si genera l'eccezione: "Non ci sono abbastanza posti". Il cliente ha inoltre la possibilità di vedere gli spettacoli da lui prenotati, aprendo l'apposita finestra "I miei spettacoli".

Anche l'operatore deve essere registrato correttamente per poter accedere. Una volta effettuato il login, potrà aggiungere uno o più spettacoli, a cui viene attribuito automaticamente un codice, visualizzare gli spettacoli già rappresentati e visualizzare gli spettacoli che non sono ancora stati messi in scena. L'operatore può anche prenotare gli spettacoli per conto del cliente, selezionando il nome del cliente e lo spettacolo da esso desiderato dalla finestra denominata "Spettacoli attuali" ed eventualmente cercandone

il nome.

Capitolo 2

Design

Il pattern principale che è stato utilizzato è il model-view-controller. A questo si uniscono i pattern strategy e singleton.

Le classi che ereditano dalla classe astratta Gestione seguono il pattern singleton. Esse permettono infatti di istanziare un'unica istanza. Lo stesso discorso vale per il frame che gestisce la registrazione.

Per semplicità è stato creato un nuovo frame per ogni finestra della GUI.

2.1 Architettura

I package sono stati divisi nei seguenti gruppi: Control, Exception, Model, View.

La View a sua volta si suddivide in View, View.cliente, View.operatore.

- Control: le classi che sono contenute qui gestiscono i cambiamenti nell'interfaccia grafica e i salvataggi dei vari oggetti.

-Gestione

-GestioneCliente

-GestioneOperatore

-GestionePrenotazione

-GestioneSpettacolo

-IGestione

- Exception: le classi in questo package gestiscono le varie eccezioni lanciate all'interno del progetto.

-ExceptionNomeEsistente

-ExceptionNumNegativo

-ExceptionPostiNonDisponibili

- Model: in questo package ci sono le classi che modellano gli oggetti principali del progetto.

-Cliente

-ILogIn

-Operatore

-Persona

-Prenotazione

-Spettacolo

- View: qui sono contenute le classi che permettono l'interazione con la login.

-MainFrame

-RegistrazioneFrame

- View.cliente: qui sono contenute le classi che gestiscono l'interazione tra il cliente loggato e il sistema

-MainFrameCliente

-VisualizzaPrenotazioni

- View.operatore: qui sono contenute le classi che gestiscono principalmente l'interazione tra un operatore e il sistema

- AggiuntaSpettacolo
- FrameNumBiglietti
- FramePrenotazione
- MainFrameOperatore
- VistaSpettCorrente
- VistaStoricoSpett

2.2 Design dettagliato

- Gestione: Classe astratta che implementa l'interfaccia IGestione.

Permette di gestire le mappe, attraverso cui vengono rappresentati i vari concetti di spettacolo, prenotazione, cliente e operatore e di salvarli su file e caricarli al momento opportuno.

- GestioneCliente: Classe che estende Gestione, fornendo il percorso specifico in cui salvare la mappa relativa ai clienti. Permette di instanziare una singola istanza. Contiene inoltre un metodo per effettuare i controlli sugli username inseriti durante la login dai clienti e dei metodi per aggiungerli e cercarli in base al loro nome.

- GestioneOperatore: Classe che estende Gestione, fornendo il percorso specifico in cui salvare la mappa relativa agli operatori. Permette di instanziare una singola istanza. Contiene inoltre un metodo per effettuare i controlli sugli username inseriti durante la login dagli operatori e un metodo per aggiungerli.

- **GestionePrenotazione**: Classe che estende **Gestione**, fornendo il percorso specifico in cui salvare la mappa relativa alle prenotazioni. Permette di instanziare una singola istanza. Contiene dei metodi per gestire le prenotazioni. Nello specifico ci sono un metodo per aggiungerle e un metodo per selezionarle in base al cliente che le ha effettuate.
- **GestioneSpettacolo**: Classe che estende **Gestione**, fornendo il percorso specifico in cui salvare la mappa relativa agli spettacoli. Permette di instanziare una singola istanza. Contiene inoltre dei metodi per aggiungere uno spettacolo, cercarlo in base al nome e suddividerlo in base alla data di esecuzione. Solo quelli da portare ancora in scena, infatti, saranno prenotabili, mentre gli altri finiscono nello storico.
- **IGestione**: Interfaccia che contiene i metodi per gestire il salvataggio sui file e quelli necessari a gestire le mappe.
- **ExceptionNomeEsistente**: Classe che estende **Exception** e comporta il lancio di un'eccezione nel momento in cui un utente tenta di registrarsi con uno username già presente nel sistema.
- **ExceptionNumNegativo**: Classe che estende **Exception** e comporta il lancio di un'eccezione nel momento in cui un utente tenta di prenotare un numero negativo di posti.
- **ExceptionPostiNonDisponibili**: Classe che estende **Exception** e comporta il lancio di un'eccezione nel momento in cui un utente tenta di prenotare un numero di posti maggiore di quelli disponibili rispetto alla capacità della sala.

- Cliente: Classe serializzabile che eredita da Persona e modella un generico cliente.
- Operatore: Classe serializzabile che eredita da Persona e modella un generico operatore.
- Spettacolo: Classe serializzabile che modella un generico spettacolo. Uno spettacolo è dotato di un nome, una compagnia di attori che vi recita, una data di esibizione, un prezzo e un codice univoco.
- Prenotazione: Classe serializzabile che modella una generica prenotazione. Una prenotazione è composta da un codice cliente, un codice operatore (-1 se il cliente prenota in autonomia), un codice spettacolo e il numero dei posti che il cliente vuole prenotare.
- Persona: Classe serializzabile che implementa l'interfaccia ILogin e modella un generico utente, dotato di nome, cognome, username e password.
- ILogin: Interfaccia che contiene i metodi che ritornano username e password, necessari alla login.
- MainFrame: Classe principale da cui viene avviato il programma e permette l'accesso al sistema tramite login.
- RegistrazioneFrame: Classe che permette di istanziare una singola istanza e permette la registrazione di un nuovo utente al sistema.
- MainFrameCliente: Classe principale a cui accede il cliente. In alto sono

visibili il nome e il cognome dell'utente loggato. Attraverso il JButton "visualizza spettacoli" all'utente vengono mostrati tutti gli spettacoli che ancora devono essere messi in scena e sono quindi prenotabili, mentre attraverso il JButton "le mie prenotazioni" egli può vedere quelli per cui ha già prenotato un determinato numero di posti. Attraverso il JButton "esci" il cliente effettua la logout e si torna al frame di inizio del programma.

- Visualizza Prenotazioni: Classe che permette al cliente di visualizzare le proprie prenotazioni. Attraverso il JButton "indietro" torna alla home page a lui dedicata.

- AggiuntaSpettacolo: Classe che permette all'operatore di inserire un nuovo spettacolo nel sistema.

- MainFrameOperatore: Classe principale a cui accede l'operatore. In alto sono visibili il nome e il cognome dell'utente loggato. Attraverso il JButton "aggiungi spettacolo" viene aperto il frame per aggiungere un nuovo spettacolo al sistema. "visualizza storico spettacoli" permette di visualizzare gli spettacoli già messi in scena, con il rispettivo numero di biglietti venduti. "visualizza spettacoli correnti", invece, apre il frame attraverso cui vengono visualizzati gli spettacoli che ancora devono essere eseguiti. Attraverso il JButton "esci" il cliente effettua la logout e si torna al frame di inizio del programma.

- VistaSpettCorrente: Classe che permette di visualizzare gli spettacoli che devono ancora essere rappresentati. Attraverso il pulsante cerca questi vengono filtrati in base al nome inserito nella JTextField in alto. Attraverso

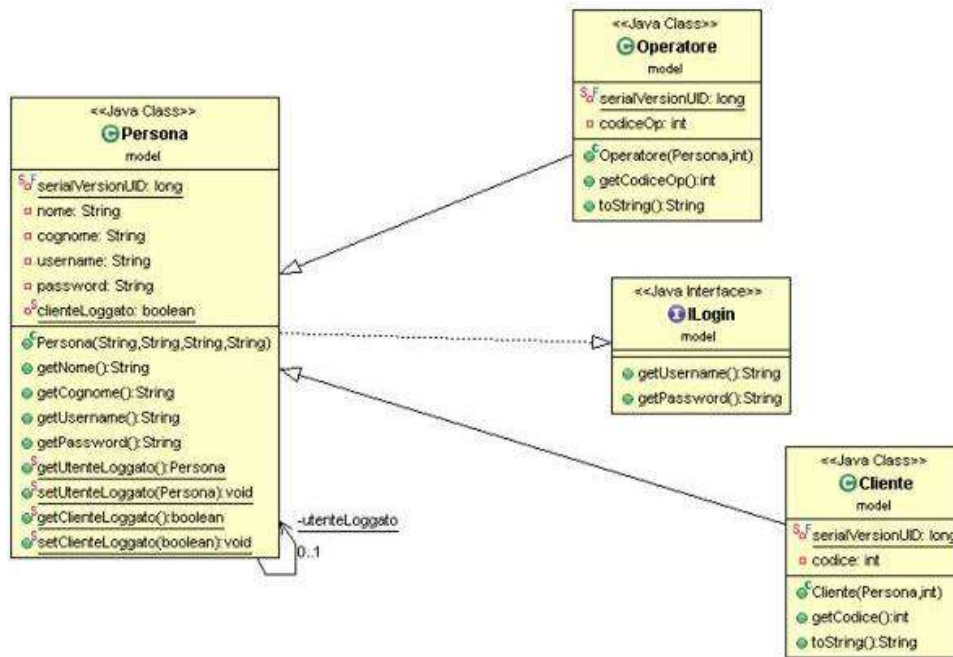
il pulsante prenota viene aperto direttamente lo spinner per selezionare il numero dei posti da prenotare se l'utente loggato è un cliente o si viene rimandati all'apposito Frame se l'utente loggato è un operatore. Dopo aver effettuato una prenotazione per visualizzare il corretto numero di posti residui è necessario premere il JButton "aggiorna".

- VistaStoricoSpettacoli: Classe che permette semplicemente di visualizzare gli spettacoli già rappresentati.

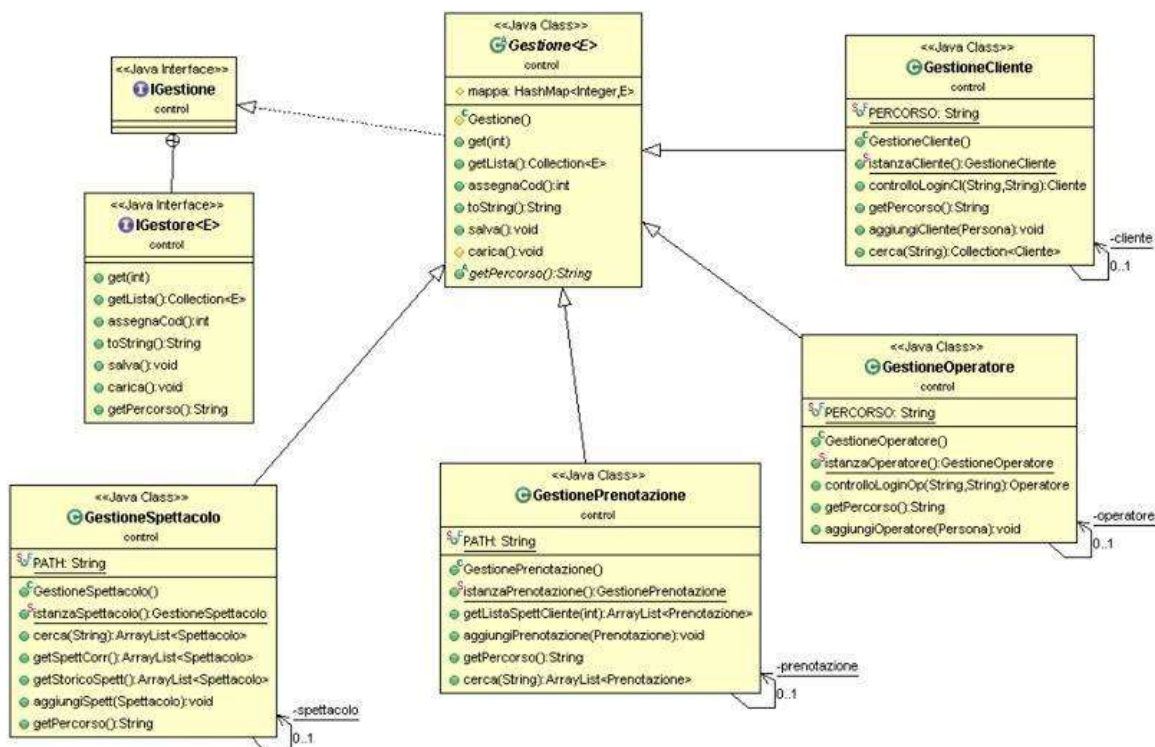
- FrameNumBiglietti: Classe che viene richiamata al momento di una prenotazione per selezionare i posti desiderati e salvare le modifiche.

- FramePrenotazione: Classe che permette di visualizzare il frame attraverso cui è un operatore ad effettuare una prenotazione. "Cerca" permette di filtrare gli spettacoli e i clienti in base alle stringhe inserite nelle apposite JTextField. "Prenota" richiama la classe FrameNumBiglietti, mentre "Aggiorna" permette di visualizzare l'esatto numero di posti disponibili nei vari spettacoli dopo una qualche prenotazione. "Indietro" permette di tornare al frame generato dalla classe VistaSpettCorrente.

Il model, il controller e la view sono stati progettati separatamente, secondo il pattern model-view-controller, per cui in caso di modifica della view non dovrebbero esserci ripercussioni sul controller e sul model.



La prima figura rappresenta il Model, che parte dalla radice dell'interfaccia ILogin collegata alla classe Persona la quale si suddivide in due classi: la classe Operatore e nella classe Cliente.



Nella seconda figura viene rappresentato il Control, che ha come radice l'interfaccia IGestione, la quale è collegata alla classe Gestione. Da questa si diramano le classi: GestioneCliente, GestionOperatore, GestionePrenotazione e GestioneSpettacolo.

Capitolo 3

Commenti finali

3.1 Autovalutazione e lavori futuri

I punti di forza di questa applicazione consistono nel fatto che un cliente può prenotare dal computer gli spettacoli desiderati saltando la fila adibita all'acquisto ma facendone un'altra, immediata, che ha la precedenza e pagando subito all'operatore, senza che appunto l'operatore debba inserirli a mano; ma in caso di problemi può contattare telefonicamente un operatore, il quale chiederà il nome del cliente e lo spettacolo che vorrebbe prenotare e poi ci penserà l'operatore, semplicemente selezionando il nome del cliente, il quale avrà già associati tutti i suoi dati e selezionando lo spettacolo desiderato e premendo il tasto aggiungi.

In futuro si potrebbe aggiungere il JButton “rimuovi”, per rimuovere uno spettacolo e una prenotazione, il JButton “modifica” per modificarli in caso di cambiamenti o errori di inserimento, ed un ultimo JButton “stampa” per poter stampare direttamente i biglietti.

Appendice A

Guida utente

Il funzionamento di questo software è intuibile, ma per ogni evenienza si descrive il suo funzionamento. Se la persona che vuole accedere non è ancora iscritta, dovrà spingere il bottone “Registrati” in basso a sinistra nella finestra iniziale. Così facendo si aprirà una seconda finestra, dove verranno richieste delle informazioni che dovranno essere inserite obbligatoriamente, si seleziona se si tratta di un cliente o di un operatore e si preme il tasto “Ok” per confermare o “Annulla” per chiudere la finestra. Se le informazioni inserite corrispondono a quelle di un utente già registrato viene visualizzato un messaggio d’errore.

In caso si abbia confermato e la registrazione sia andata a buon fine, verrà visualizzata nuovamente la finestra di accesso, dove poter inserire le credenziali inserite nella registrazione e selezionare se si tratta di un cliente o di un operatore. Per accedere occorre spingere “Login”, mentre per uscire dall’applicazione “Esci”.

Se si è un cliente, dopo aver acceduto si aprirà una finestra dove compariranno nella parte alta il nome e il cognome, utilizzati durante la registrazione, e sotto di essi tre bottoni:

-Visualizza spettacoli: mostra gli spettacoli da prenotare. Qui vengono visualizzati solo quelli che devono ancora essere messi in scena e possono essere ricercati tramite il nome. Attraverso il pulsante “Cerca” la lista viene aggiornata, mostrando solo gli spettacoli che contengono nel nome

la stringa inserita.

-Le mie prenotazioni: mostra le prenotazioni effettuate in precedenza grazie a questa applicazione;

-Esci: per uscire dalla home del cliente.

Se si tratta di un operatore, dopo aver acceduto come tale, si aprirà una finestra dove compariranno in alto il nome e il cognome, utilizzati durante la registrazione, e sotto di essi quattro bottoni:

-Aggiungi spettacolo: l'operatore potrà aggiungere gli spettacoli che verranno messi in scena completando tutti i campi richiesti e premendo “Aggiungi” per confermare o “Indietro” per annullare;

-Visualizza storico spettacoli: l'operatore potrà visualizzare gli spettacoli la cui data è antecedente a quella attuale e potrà anche cercare un certo spettacolo in base al nome;

-Visualizza spettacoli correnti: la finestra che si aprirà mostrerà gli spettacoli disponibili per essere prenotati, che possono essere filtrati in base al nome attraverso il pulsante “Cerca”. Da qui è anche possibile aggiungere per conto del cliente lo spettacolo da lui desiderato. Per visualizzare il corretto numero di posti disponibili nei vari spettacoli dopo aver effettuato una prenotazione occorre premere il pulsante “Aggiorna”.

-Logout: per uscire dalla home dell'operatore.

Note

Nella classe “Gestione”, all’interno del metodo “Carica”, è stato inserito un `@SuppressWarnings`. A generarlo è stato il cast, necessario per l’inserimento dell’oggetto caricato dalla memoria nella mappa.

In particolare ci si riferisce alla linea di codice:

```
“this.mappa = (HashMap<Integer, E>) o;”
```

Per risolverlo si sarebbe dovuto inserire un’istruzione “instance of”, che però non è possibile utilizzare in modo generico.