Corso di Laurea in Ingegneria e Scienze informatiche

Università di Bologna – Sede di Cesena

Programmazione Orientata agli Oggetti – A.S: 2014/2015

RELAZIONE

HM – Hotel Manager

Mattia Oriani, Federico Vitali

Capitolo 1

Analisi

Abbiamo realizzato un'applicazione desktop utile alla gestione di un hotel. Si può utilizzare l'applicazione con due utenti, uno per gestire la reception e uno per gestire la parte amministrativa.

1.1 Requisiti

Nell'applicazione sarà possibile eseguire le seguenti funzioni:

Reception:

- •Inserire le prenotazioni, compilando i dati richiesti (utente, data, tipo di camera).
- •Modificare la camera scelta nella prenotazione.
- •Effettuare il check-out, anticiparlo, o eliminare la prenotazione se l'arrivo è futuro.
- •Simulare l'andamento del calendario, con aggiornamento automatico dei check-out.

Direction:

- •Modificare il costo fisso delle pulizie
- •Inserire una nuova bolletta da pagare
- •Visualizzare la situazione contabile (entrate/uscite avvenute)
- •Visualizzare lo storico dei clienti dell'albergo
- •Cercare fra i clienti, e visualizzare lo stato della sua prenotazione
- •Gestire gli account della reception, eliminandoli o aggiungendoli.

Capitolo 2

Design

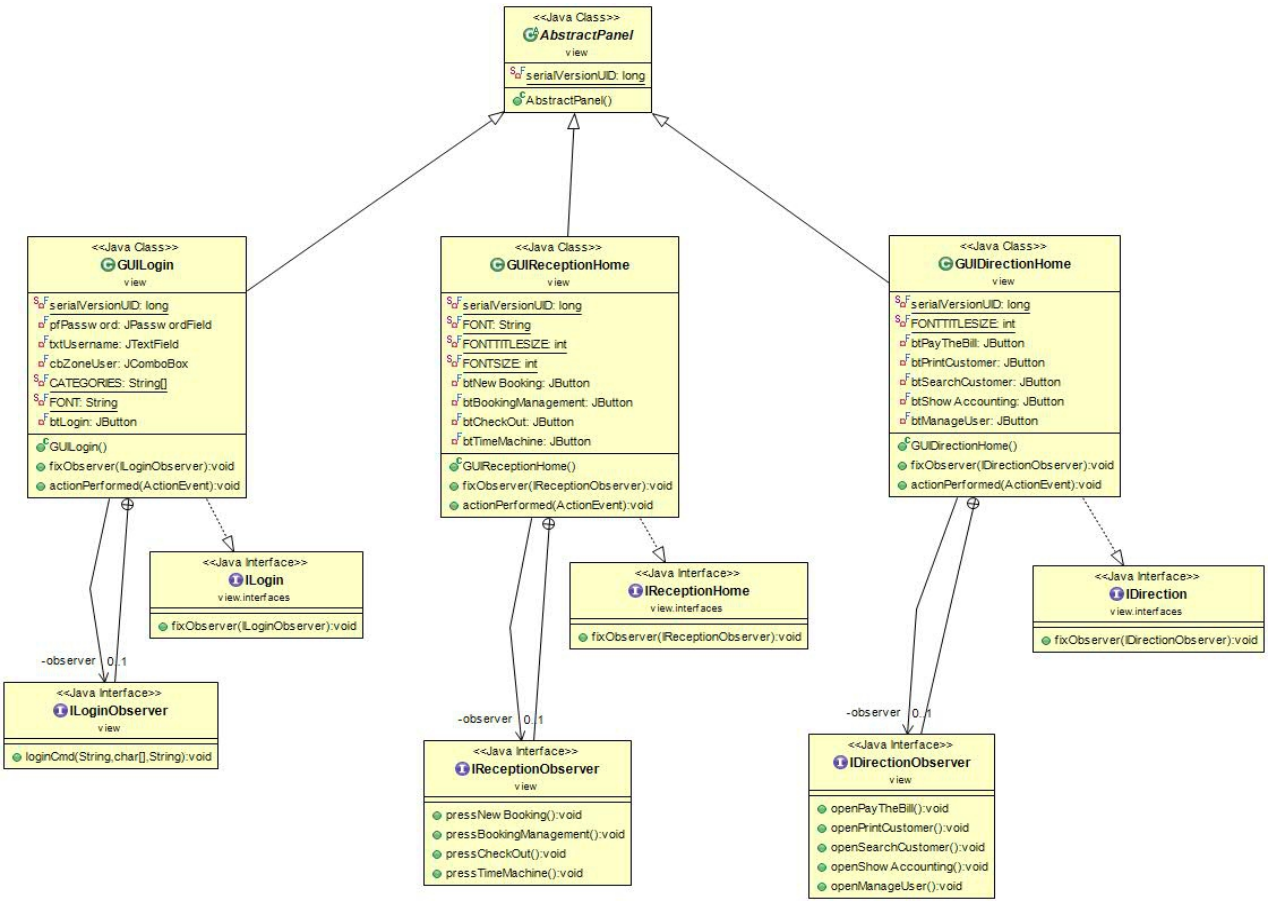
2.1 Architettura

Nella fase di design, è stato fatto e progettato lo scheletro del programma, capendo quali entità fossero da creare e in che relazione dovessero essere fra loro.

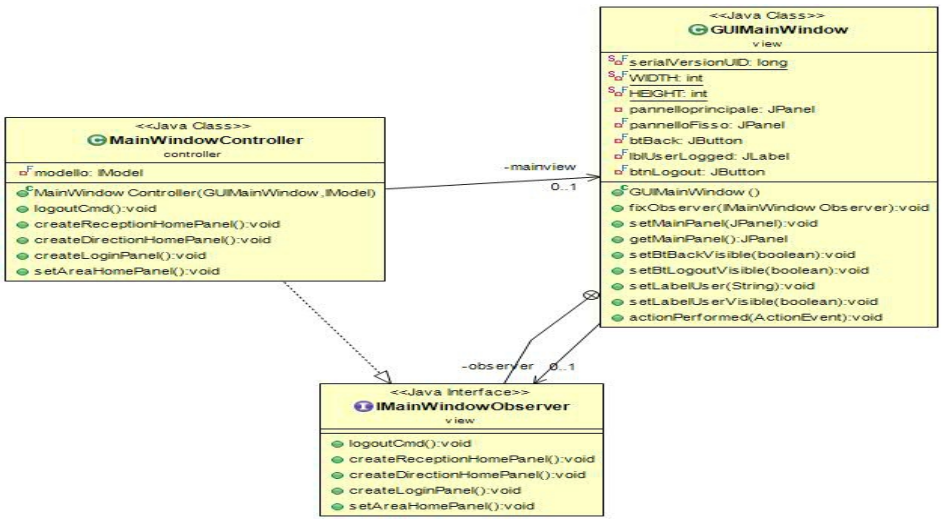
La progettazione si è basata sull'utilizzo del pattern MVC (Model – View – Controller). Questo consente di non rendere dipendenti le parti tra loro, quindi le parti di visualizzazione grafica (view) e di gestione dei dati (model) sono indipendenti, così come la parte di logica del programma (controller). Ciò consente in futuro il riutilizzo di una o più parti in applicazioni diverse, e consente facili modifiche a una parte specifica del codice per una facile manutenzione e manutenibilità.

Tramite diagrammi UML sono visualizzate le principali caratteristiche di quanto appena descritto.

In questo diagramma UML sono visualizzate le principali classi di view, quali Login, e i due pannelli di accesso come ReceptionHome e DirectionHome, ogni classe estende dalla propria interfaccia e ha il proprio observer.
Ogni classe della view è rappresentata in questo modo, estendendo da un pannello astratto che ne imposta caratteristiche comuni a tutte, e ogni classe della view ha il proprio observer.

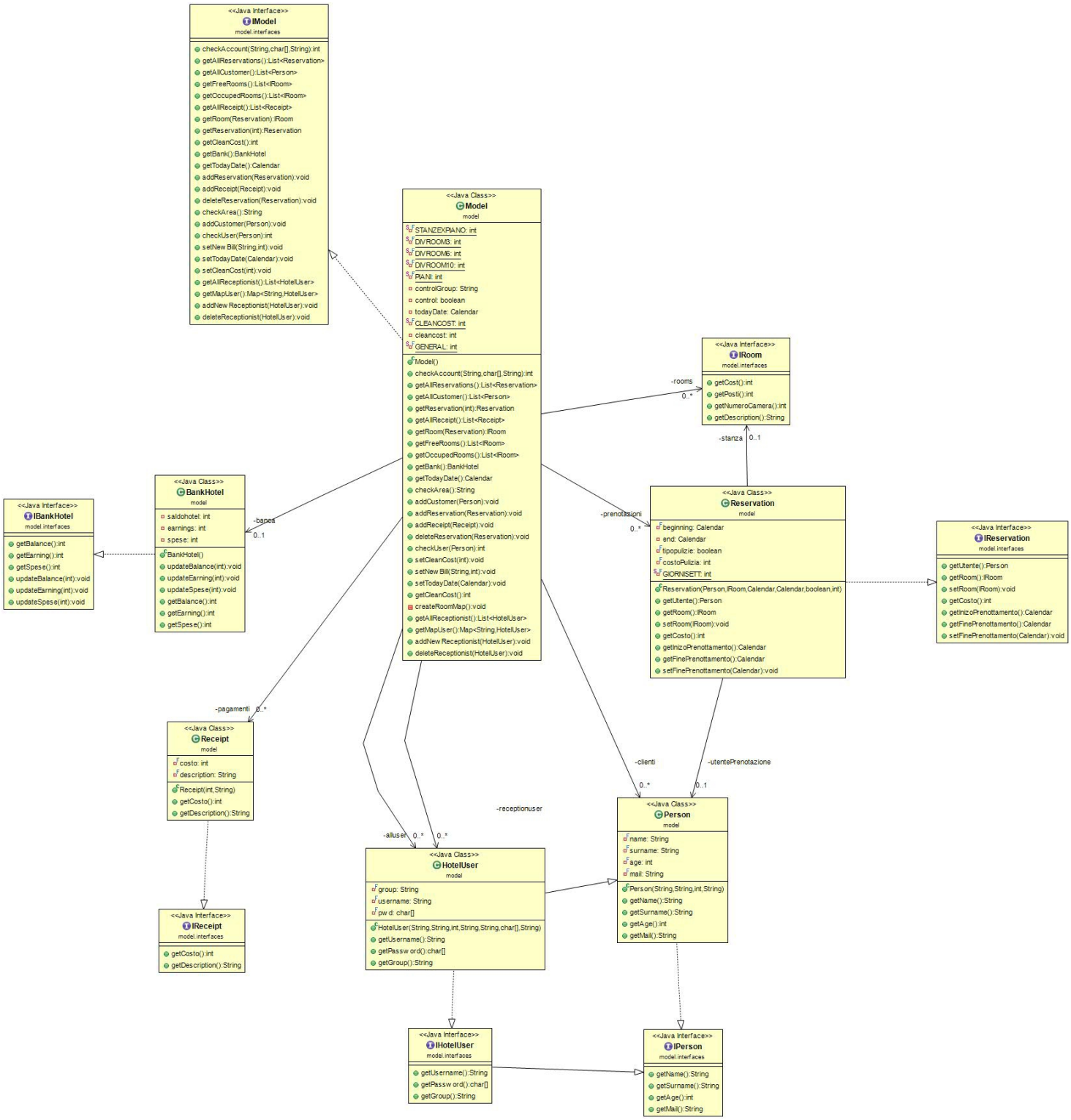


Ogni pannello viene inserito quando opportuno e coerente nel MainWindow, che estende il frame, quindi questa è una parte della gui che non cambia mai.



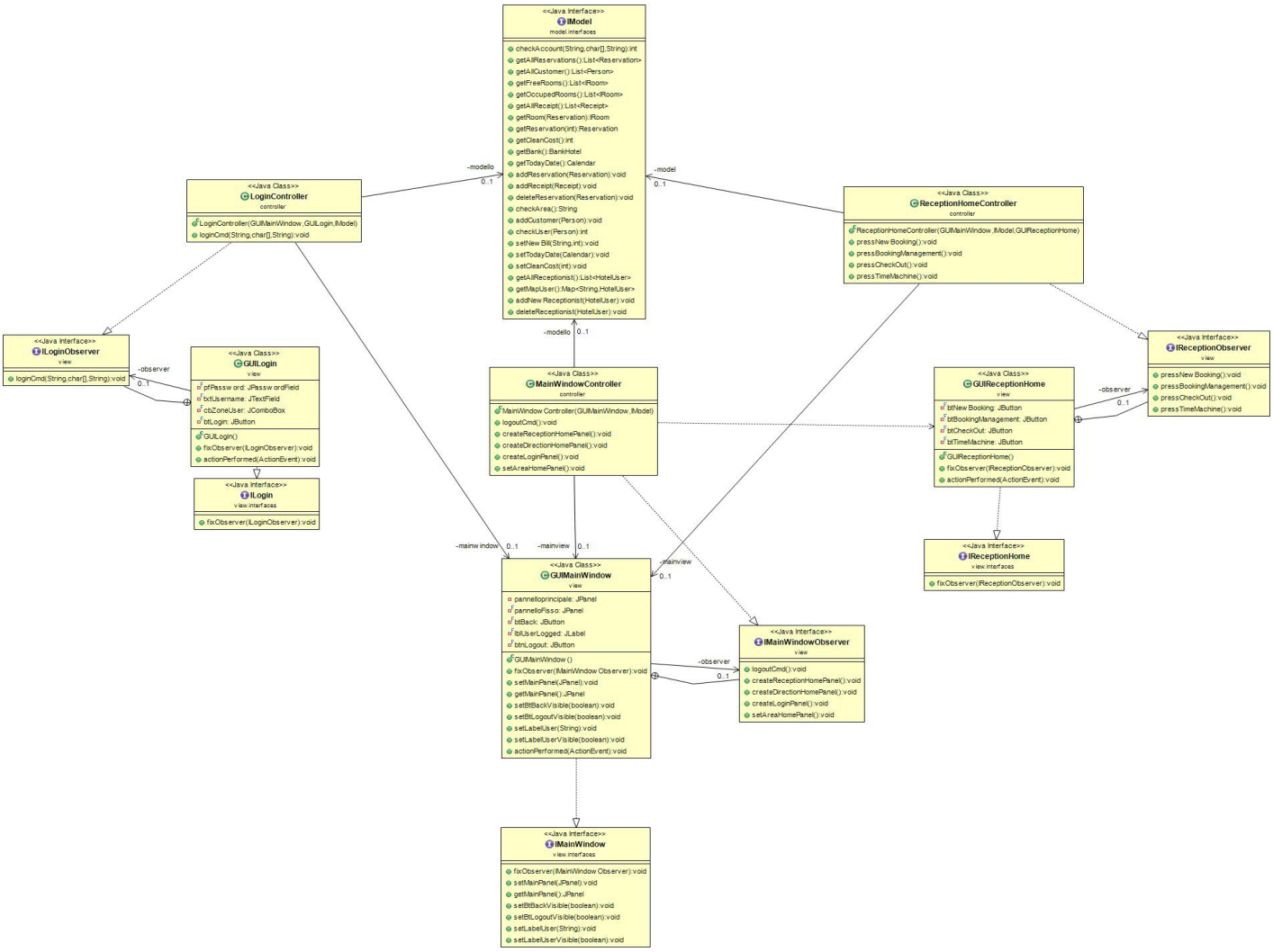
Nel seguente diagramma UML invece viene presentata la parte relativa al modello dove vengono processati i dati e realizzate le principali operazioni, e dove sono contenute le principali classi che realizzano tali oggetti.

Sono presenti le classi Reservation per indicare una prenotazione, Bank per gestire le finanze, Person e HotelUser per gestire rispettivamente clienti e utenti per la reception, quest'ultima classe estende la precedente in quanto aggiunge solo informazioni.



Nel successivo diagramma UML invece viene visualizzata la gestione del programma fra gui, model, e i vari controller nella fase di login per la reception, in relazione con quanto progettato secondo il pattern mvc.

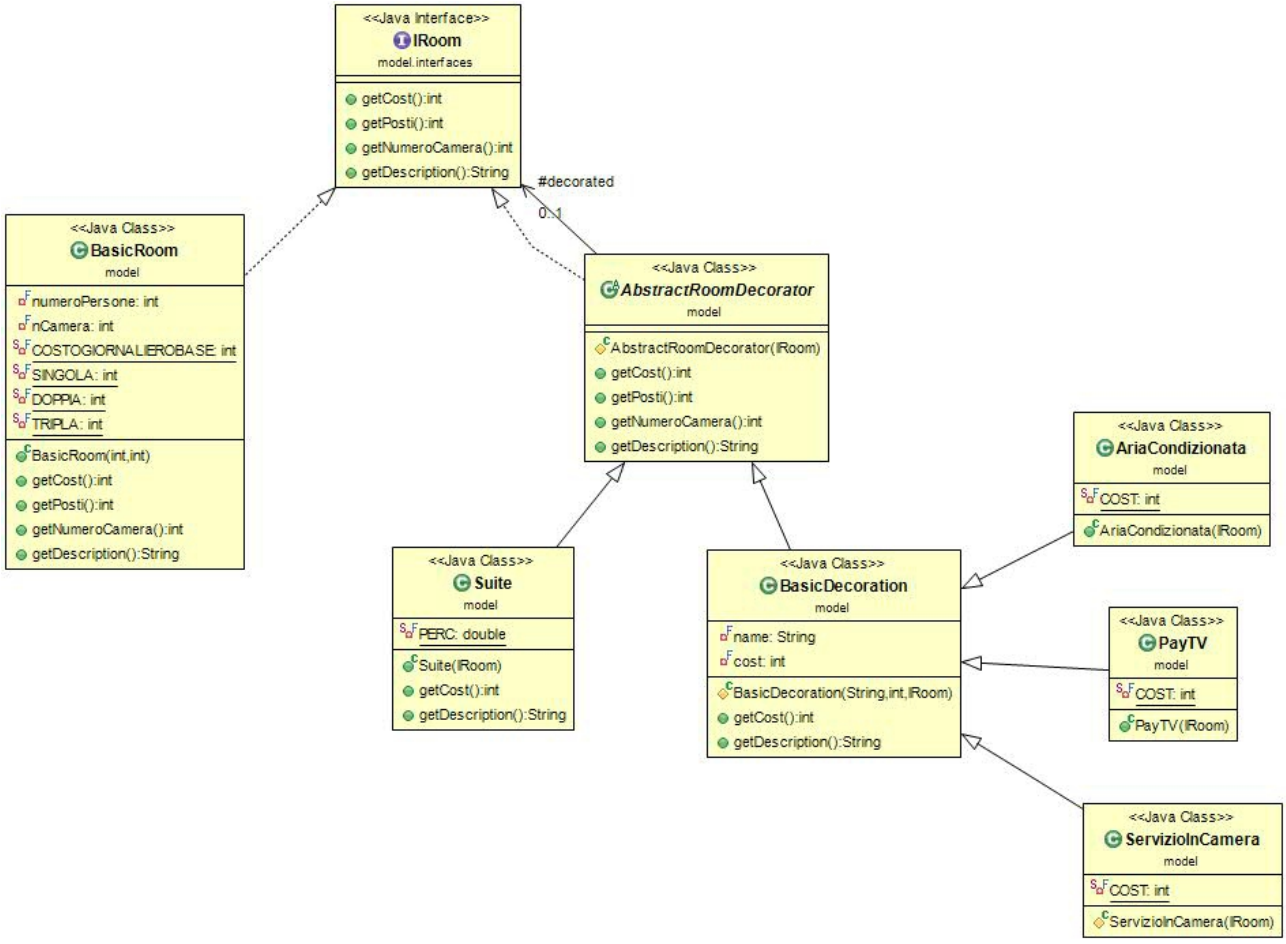
Quando si preme login il pannello cambierà e diventerà quello relativo alla reception.



2.2 Dettagli di progettazione

La realizzazione delle camere dell'hotel, è stata progettata seguendo il pattern decorator, così ogni camera sarà "decorata" in base a vari optional.

Segue lo schema UML in cui si può verificare il pattern, la camera base, e gli optional PayTV, A/C, Servizio Camera. La SUITE fa lievitare il costo della camera decorata.



Capitolo 3

Sviluppo

3.1 Testing dell'applicazione:

Ogni parte dell'applicazione, sia per la parte grafica che non, è stata testata subito dopo la sua realizzazione, con un uso del debug per trovare eventuali errori non segnalati.
Alla fine si è sistemato il codice con i Plugin suggeriti quali PMD, Checkstyle, Findbugs.

3.2 Suddivisione dei compiti:

Il lavoro è stato diviso equamente fra i componenti del gruppo, e integrato volta per volta con mercurial fra i due componenti grazie all'uso di bitbucket come repository sulla quale appoggiarsi. Ogni volta che veniva apportata una modifica sostanziale si aggiornava la repository.
Nel dettaglio i compiti sono stati suddivisi in:
Parte Grafica relativa ai vari ambiti: Mattia Oriani
Parte gestione dati reception: Federico Vitali
L'interfacciamento fra le due parti è stato implementato insieme, svolgendo ognuno la relativa parte, per collegare la grafica da un lato e la parte di gestione dall'altro.
Parte gestione direction: Mattia Oriani, Federico Vitali, in quanto dato il ritiro di un componente, ci siamo divisi la sua parte per completare il progetto secondo le funzioni prestabilite.

3.3 Note di sviluppo

Dopo aver consultato progetti di anni passati abbiamo trovato utile l'uso dei calendari, quindi è stata aggiunta una libreria esterna chiamata “jcalendar” (importata da com.toedter.calendar), per facilitare la scelta delle date durante le prenotazioni utilizzando un calendario apposito.

CAPITOLO 4

Commenti Finali

4.1 Conclusioni e lavori futuri

Il progetto soddisfa ciò che avevamo pensato e ideato inizialmente, quindi ci riteniamo soddisfatti del lavoro. Sicuramente la totale assenza di Casanova ci ha obbligati a "correre" per riuscire a concludere e realizzare tutti i punti definiti all'inizio, probabilmente compromettendo alcune parti del Model.

In futuro pensiamo sia utile aggiungere il salvataggio delle prenotazioni e di tutti gli altri dati su un database esterno, e aggiungere una parte per la gestione del ristorante, in modo da rendere univoco il software per qualsiasi funzione dell'hotel.

APPENDICE A

GUIDA UTENTE:

Nella finestra del login del programma si accede alla parte della reception con l'utente:
username: **mariorossi** password: **dompero**
e si accede alla parte direction con l'utente unico:
username: **direttore** password: **hotel**
Quando si viaggia con la time machine fino a una data specifica, verranno processate tutte le prenotazioni fino al giorno prima, in modo che quella della data stessa siano fattibili manualmente.
Nel check-out, si può, anticiparlo con conferma di richiesta, effettuarlo direttamente se si è nel giorno stesso, e anche eliminare una prenotazione se non si è ancora arrivati.