



RISTOAPP

Preparato per il corso di "Programmazione Ad Oggetti" dal professore Mirko Viroli

Preparato da: Melnychuk Oleksandr

4 maggio 2017

SOMMARIO

Questo documento è una relazione del progetto svolto da studenti Oleksandr Melnychuk per il corso di “Programmazione Ad Oggetti” dal professore Mirko Viroli. Lo scopo di questo documento è quello di evidenziare il lavoro svolto dal team, sottolineando le metodiche adottate, i problemi riscontrati durante lo sviluppo e le soluzioni scelte.

INDICE

1. Capitolo 1 : Analisi

- 1.1. Introduzione
- 1.2. Dominio
- 1.3. Requisiti Applicativi
- 1.4. Considerazioni (basate sulle esperienze)

2. Capitolo 2 : Architettura

- 2.1. Model
- 2.2. View
- 2.3. Controller
- 2.4. Utilities

3. Capitolo 3: Testing

4. Capitolo 4: Considerazioni finali

Capitolo 1 : Analisi

INTRODUZIONE:

Il progetto vede come dominio il mondo della ristorazione. E' per tanto necessaria una descrizione di quest'ultimo poiché andremo su una parte particolare di questo dominio.

DOMINIO:

Il mondo della ristorazione si basa sul volume d'affari che un ristorante riesce a concretizzare in un determinato periodo ma in particolare si basa sul volume della clientela e perciò sulla grandezza del ristorante. Siccome sto parlando di un ristorante immaginario diciamo, perciò presumo che la clientela non ha un limite di età e il ristorante ha un target di persone in particolare. Un ristorante avrà un tot di posti e tot di tavoli e in un giorno potrà accogliere tot clienti in assodi di tempo presunto di 2 ore ogni due ore si potrà effettuare una nuova prenotazione sullo stesso tavolo. Il target andrà a prendere i ristoranti che sono di grandezza medio-piccola e che hanno o vorranno sviluppare un sito apposito della ristorazione. L'analisi del personale andrà ignorata perché non è necessaria per il progetto.

REQUISITI:

L'applicazione si pone come obiettivo di gestire le prenotazioni per una azienda di ristorazione nel migliore dei modi per migliorare il servizio offerto ai clienti.

I requisiti base sono:

- Creazione di nuovo prenotazione.
- Modifica di una nuovo prenotazione.

-
- Cancellazione di una prenotazione.
 - Visualizzazione di una prenotazione nello specifico.
 - Visualizzazione di tutte le prenotazioni (Una panoramica).
 - Controllo sul inserimento dati.
 - Gestione orari.

I requisiti opzionali (che saranno sviluppati in seguito):

Questi requisiti saranno concordati con il gestore del ristorante una volta che ci sarà un server web attivo che potrà aggiungere o modificarli a piacere.

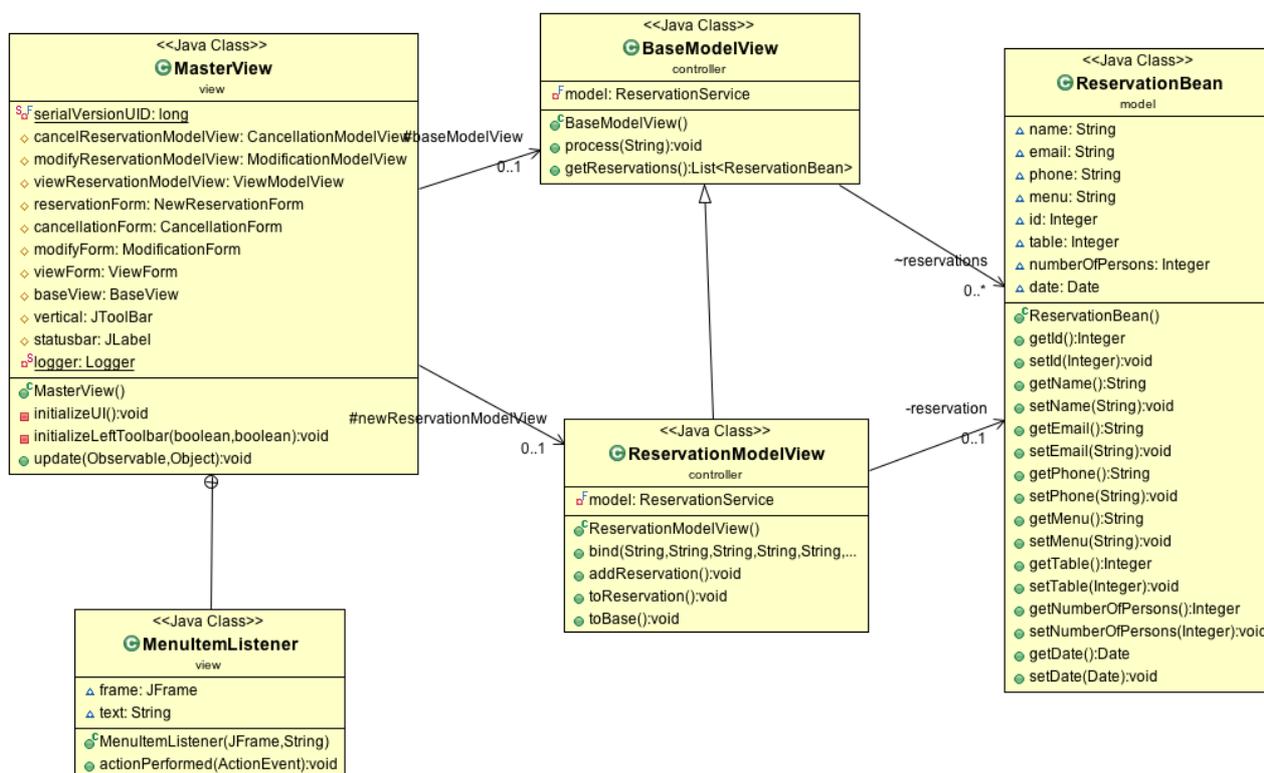
- Connessione al server web (in particolare caricamento del file xml aggiornato).
- Possibilità di osservare una storico dei clienti per premiare i clienti di fiducia con sconti vari o promozioni.
- Creazione di un'altra interfaccia apposita per i cuochi se si vorrà installare uno schermo in cucina per gestire gli ordini direttamente della cucina.
- Creazione menu online.

CONSIDERAZIONI:

Io Melnychuk Oleksandr da mesi lavoro in un locale che si occupa di ristorazione in particolare si occupa di consegne a domicilio e asporto. Ce un sistema di prenotazione che si basa sulle telefonate o messaggi via WhatsApp. Io a mio parere lo trovo scomodo e molto inefficace e incasinato perché richiede del tempo e comunque ha bisogno sempre di una persona che si occupi solamente di questo per l'intero servizio. E comunque qualche volta possono capire delle complicazioni come code di attesa o comunque incomprensione via telefono o via messaggio che potrà via del tempo oppure errori nella trascrizione dei ordini. A mio parere questo sistema potrà eliminare o comunque ridurre queste problematiche. Certo che questo sistema potrà

portare ad un impatto economico sul azienda siccome sara necessaria una manutenzione e comunque backup dei dati (settimanale), ma comunque credo che è ragionevole nel fine di migliorare il servizio offerto.

Capitolo 2 : Architettura



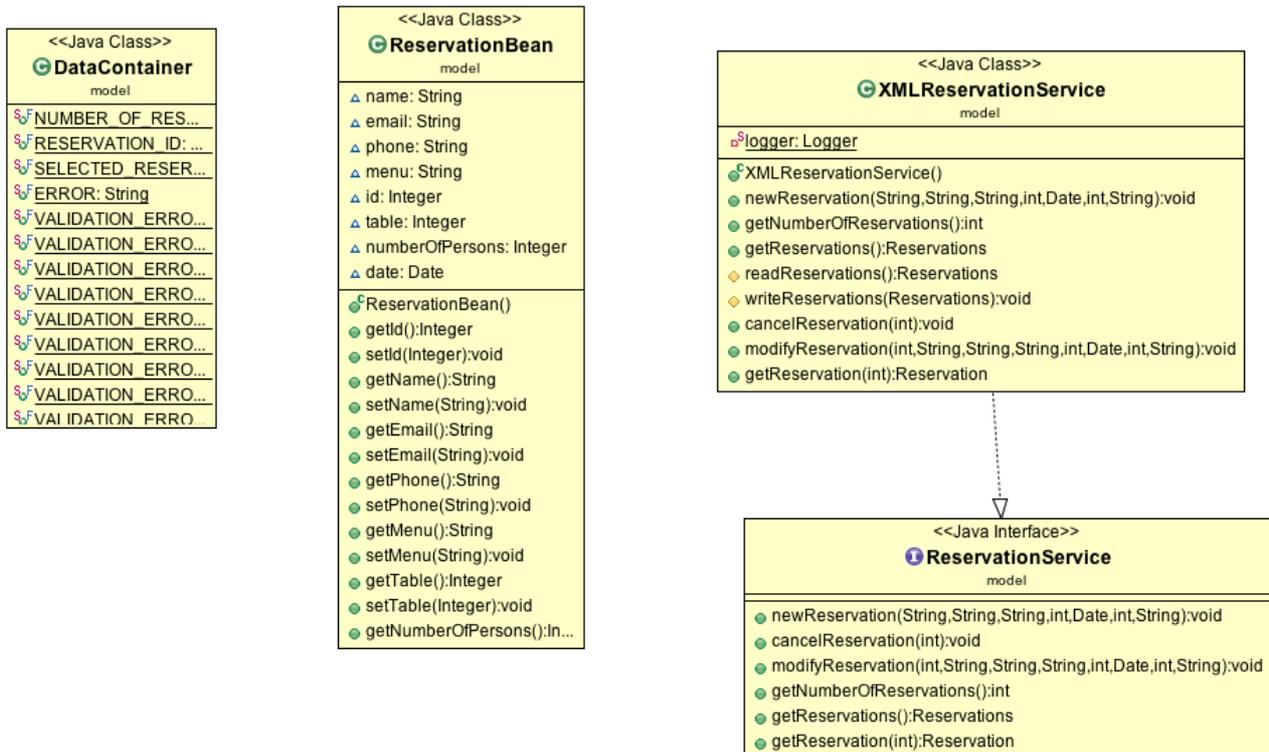
In questo capitolo verrà spigato il design progettuale e i pattern scelti. Verranno inoltre elencati vantaggi e svantaggi del lavoro e problemi riscontrati. Verranno spiegate inoltre le parti più complesse del progetto.

Oleksandr Melnychuk ha lavorato seguendo il pattern progettuale MCV ovvero Model Control View. Il model ha avuto il compito di descrivere gli elementi in causa valutando anche quali comportamenti avrebbe dovuto implementare e quali invece avrebbe dovuto lasciar implementare al controller. Gli elementi sono numerosi e vari. Il controller si occupa delle interazioni tra gli elementi e

della gestione di essi. Mentre la view fornisce un'interfaccia grafica chiara e semplice da utilizzare. Ogni parte ha lavorato nella maniera più indipendente possibile. Nella organizzazione il controller ha fatto da ponte tra Model e Grafica rendendo i due elementi completamente autonomi tra loro.

MODEL

Il model si è posto come obiettivo la realizzazione delle classi base e dei relativi metodi di interazione da mettere a disposizione per il Controller e per la View implementando alcuni controlli sulla correttezza delle operazioni effettuate e dei valori inseriti.



Obiettivi Model:

- descrizione del dominio analizzato
- corrette relazioni tra le classi

Sviluppo:

Si è scelto di effettuare la persistenza dei dati attraverso il file xml, per una comodità architetturale perciò nel model è stata sviluppata la classe con la sua interfaccia ReservationService che si occupa della gestione del file xml. Poi ce la classe DataContainer che in pratica definisce le strutture dei dati che si andranno ad utilizzare. Invece la classe ReservationBean riguarda la parte della reservation e aiuta allo scambio di dati tra il model e view layer.

VIEW

La View si è posta come obiettivo il fornire all'utente finale una GUI il più possibile piacevole da vedere e semplice da utilizzare, considerando sia gli aspetti prettamente stilistici, che quelli funzionali . La GUI è stata implementata cercando di rispettare a pieno il pattern MVC, richiamando il Controller per le varie operazioni.

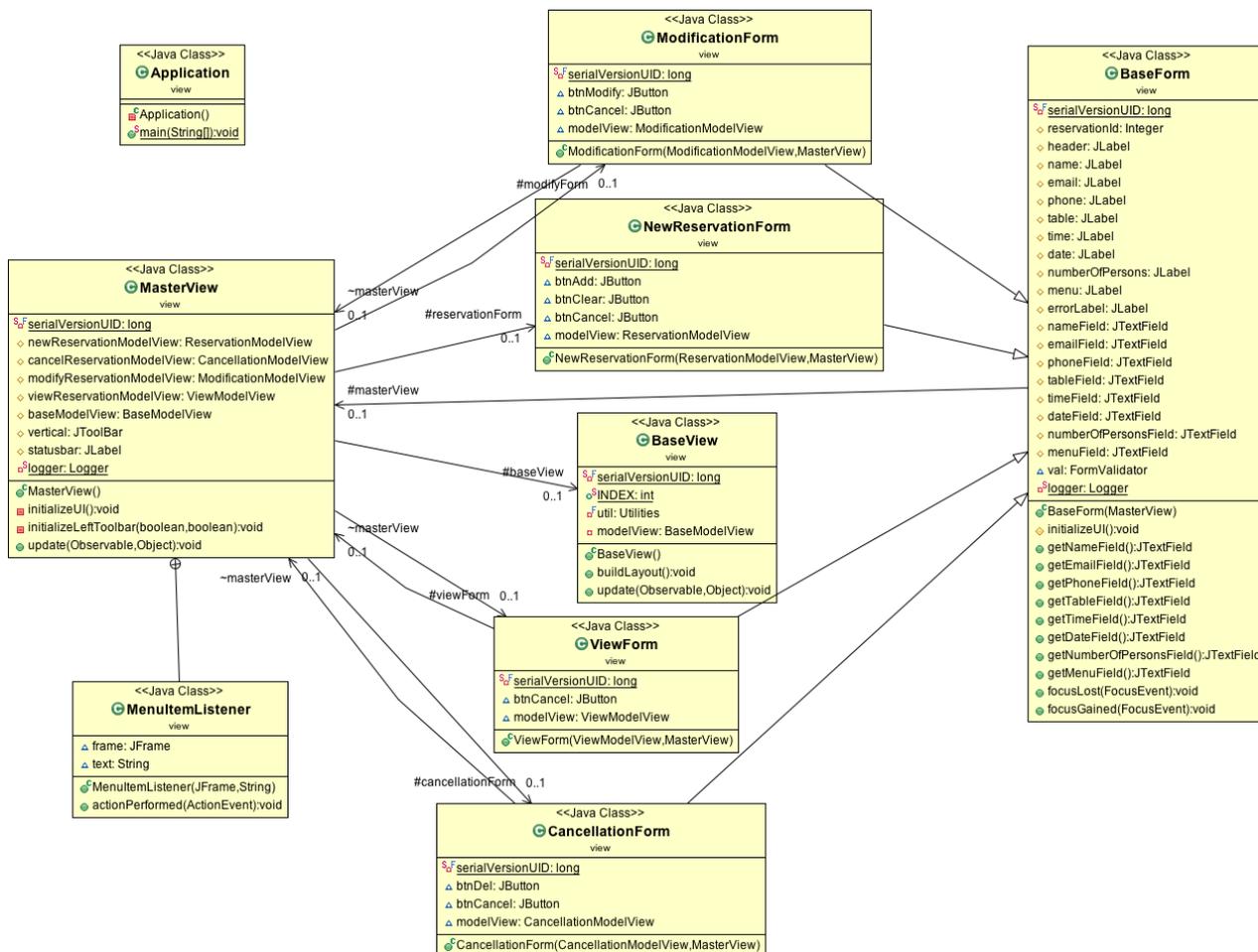
OBIETTIVI VIEW:

- Produrre una GUI piacevole alla vista
- Produrre una GUI completa e semplice da utilizzare

Sviluppo :

L'entry point è la classe Application delega il flusso di controllo dell'applicazione all'oscillazione dei Thread di inizializzazione che a suo tempo invoca la classe MasterView che rappresenta il quadro generale dell'applicazione questa classe si può dire che è il cuore della applicazione da dove si può arrivare a tutte le funzionalità dell'applicazione e offre una panoramica sulle prenotazioni. Per quanto riguarda BaseView è la classe che implementa la pagina di panoramica dalla quale vengono innescati i nuovi casi di utilizzo delle prenotazioni di modifica e di cancellazione che potranno essere raggiunti

tramite appositi buttons. Poi ci sono i vari layer della view che riguardano le form di creazione, modifica, visualizzazione e cancellazione.



CONTROLLER

Il controller ha avuto come obiettivo quello di fornire tutto ciò che la grafica necessitava, gestire i rapporti tra le varie entità, relazionare l'applicazione con il Sistema Operativo, creare gli oggetti e soprattutto rendere completamente autonomi tra loro view e model.

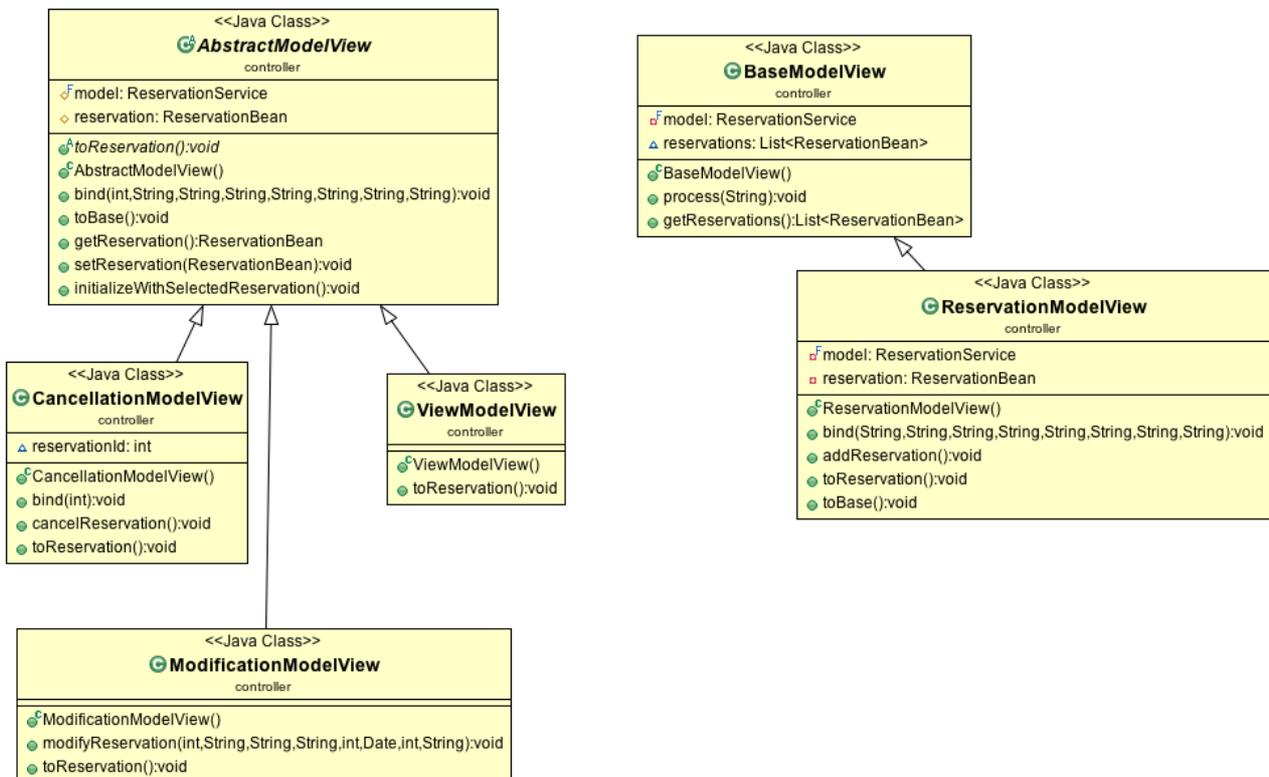
OBIETTIVI CONTROLLER:

- Indipendenza tra View e Model

- Coordinamento
- Relazione Applicazione e Sistema Operativo in particolare in fase di salvataggio

Sviluppo:

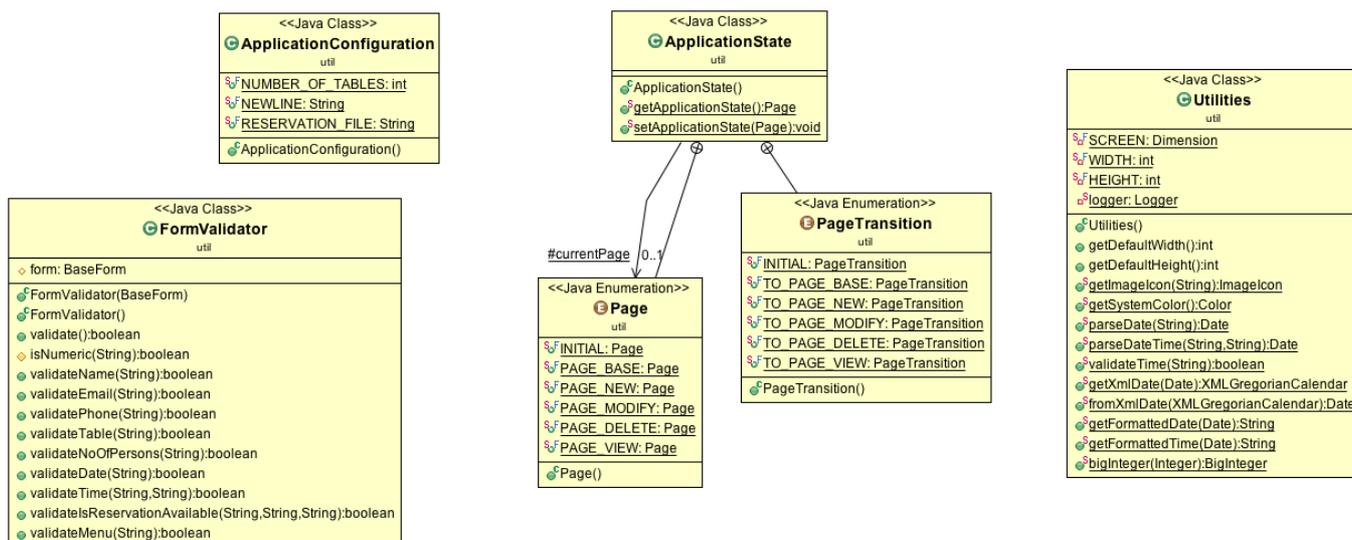
La classe BaseModelView controlla il flusso di pagine per la panoramica e la visualizzazione di avvio. Le altre classe riguardano creazione, modifica, visualizzazione e cancellazione.



UTILITIES

Utilities rappresenta un insieme delle classi che servono per migliorare la facilità d'uso come ApplicationState che gestisce lo stato dell'applicazione e le sue

transizioni , ogni pagina riflette il suo stato. Poi ce la ValidationForm che ha la responsabilità è quella di convalidare i campi di input nei moduli. Invece la classe Utilities che fornisce alcune funzioni di utilità per l'analisi e la formattazione di attività.



Testing

La parte più importante della fase di testing. Una volta terminato il programma è stato lanciato e testato in varie condizioni, testando gli inserimenti, cercando i bug o semplici errori grafici.

Considerazioni finali

A seguito delle conoscenze apprese e del risultato ottenuto ci riteniamo soddisfatti del progetto. Il programma funziona bene e si accosta molto all'idea iniziale. Molto probabilmente sarà proposto al locale dove lavoro affidandolo con il service web.