

Relazione per “JEstates”

Corso di Programmazione ad Oggetti

Calabrese Christian
Chiarello Emanuele

24 Maggio 2015

Indice

1 Analisi	2
1.1 Requisiti	3
1.2 Problema	3
2 Design	5
2.1 Architettura	5
2.2 Design dettagliato	5
3 Sviluppo	7
3.1 Testing automatizzato	7
3.2 Divisione dei compiti e metodologia di lavoro	7
3.3 Note di sviluppo	9
4 Commenti finali	8
4.1 Conclusioni e lavori futuri	8
A Guida Utente	9

Capitolo 1

Analisi

1.1 Requisiti

Il programma, JEstates, permette:

- La gestione della situazione di un insieme di proprietà (Appartamenti, ville e case indipendenti);
- Immissione di un nuovo immobile e le relative foto;
- Immissione di un nuovo cliente;
- La ricerca di un sottoinsieme di proprietà con specifiche caratteristiche;
- Overview completa delle proprietà e dei clienti;
- Creazione della locandina (formato PDF) contenente le proprietà disponibili all' affitto.

1.2 Problema

JEstates ha presentato come primo problema la memorizzazione e caricamento dei dati dovuto alla grande mole di informazioni necessarie relative agli immobili. Per ovviare il problema, si è ricorso all' uso di particolari strutture dati.

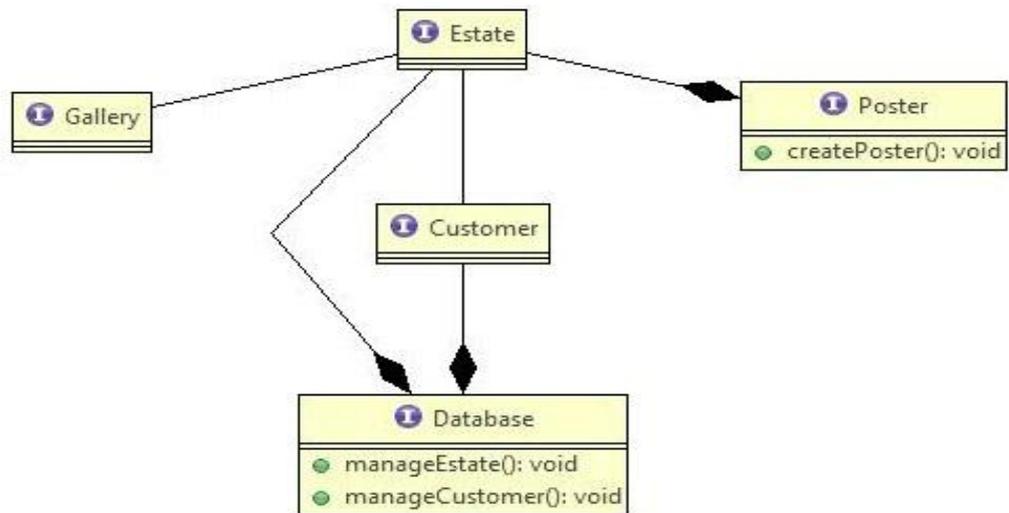


Figura 1.1: Schema UML della descrizione del problema. Ogni abitazione (Estate) e cliente (Customer) sono memorizzati in un database. Inoltre di ogni proprietà si potrà definire il proprietario tra i diversi clienti registrati il quale potrà essere detentore di più immobili. Un'altra funzionalità è la creazione di un volantino (Poster) il quale conterrà alcune informazioni delle abitazioni disponibili all' affitto ed un immagine selezionata tra quelle della galleria di ogni estate.

Capitolo 2

Design

2.1 Architettura

Per organizzare al meglio le classi, si è utilizzato il pattern MVC. Questo è stato di aiuto nell' organizzazione generale delle operazioni di inserimento e caricamento dei dati e delle relative interfacce grafiche, in quanto questo pattern permette la condivisione dei dati tra classi in modo ben strutturato.

2.2 Design dettagliato

Un design pattern che è stato implementato in diverse classi di JEstates è il SINGLETON, questo è stato utile per avere una sola istanza di specifiche classi. Il caso principale in cui è stato impiegato questo approccio è quello dei controller relativi all' inserimento di un nuovo immobile perché è stato fondamentale non creare più istanze di una classe in quanto si sarebbero state perse le strutture dati popolate in precedenza.

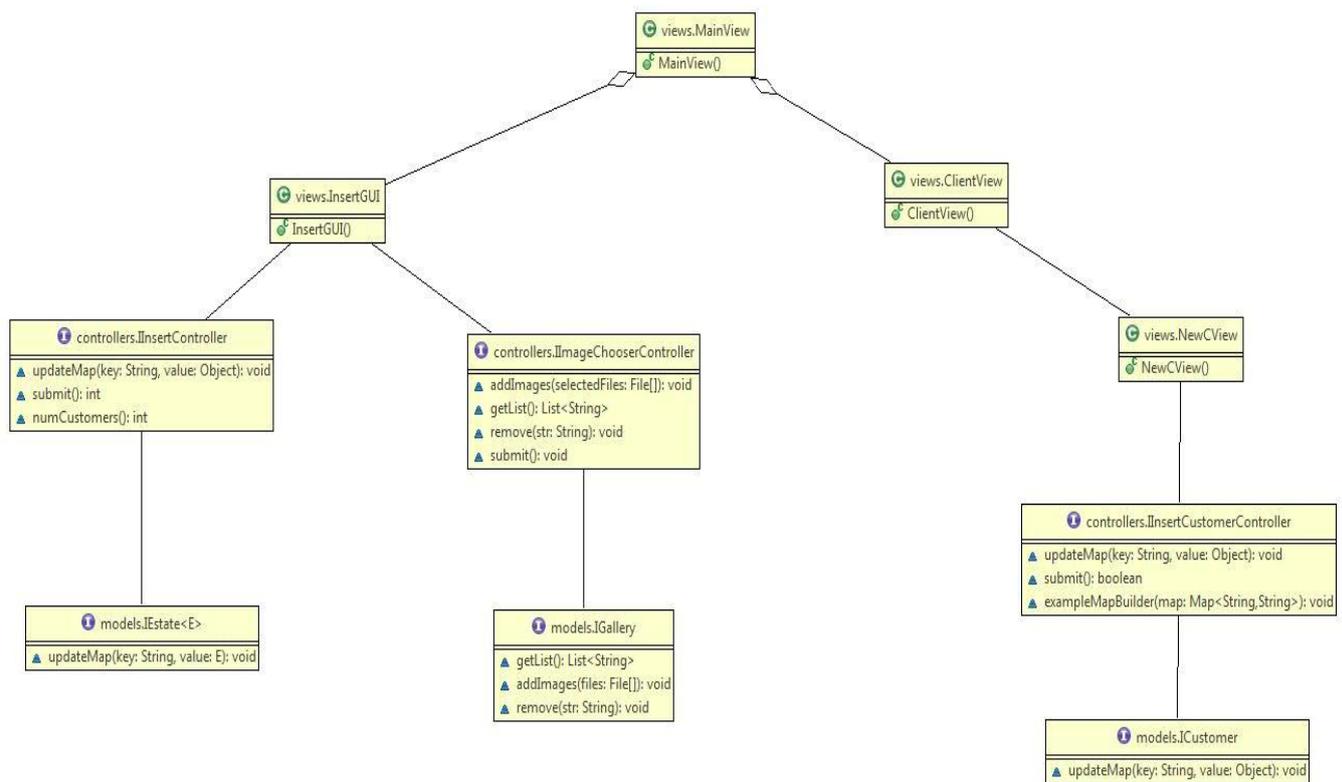


Figura 1.2: Schema UML rappresentante la metodologia di inserimento usata nel progetto. Per la realizzazione del processo di inserimento dei dati riguardanti le abitazioni e i clienti si è ricorso ad un pattern MVC, ad ogni view corrisponde un modello e ad ogni modello corrisponde un controller in modo da avere ben separate le strutture dati da chi le gestisce e da chi le propone all' utente.

Capitolo 3

Sviluppo

2.3 Testing automatizzato

Per verificare di volta in volta il corretto funzionamento di JEstates si è ricorso alla creazione di un sistema che crea automaticamente dei dati d' esempio personalizzabili.

Inoltre sono stati eseguiti diversi test manuali per verificare la correttezza delle funzionalità di JEstates su sistemi Windows 7 64 bit, Windows 10 64 bit insider preview.

2.4 Divisione dei compiti e metodologia di lavoro

Durante la produzione del progetto abbiamo deciso di modificare la divisione dei compiti rispetto a quella presentata precedentemente con la seguente:

Calabrese Christian: Parte grafica ed interfacciamento con il database per gli immobili, creazione della locandina;

Chiarello Emanuele: Parte grafica ed interfacciamento con il database per i clienti, gestione database;

Parte comune: Interfacciamento tra database e parte grafica.

2.5 Note di sviluppo

Per effettuare l'immagazzinamento dei dati abbiamo utilizzato sqlLite in modo da averli registrati in un database

(<https://bitbucket.org/xerial/sqlite-jdbc>). Inoltre per realizzare la creazione del volantino in formato pdf si è usata la libreria lowagie (<http://mvnrepository.com/artifact/com.lowagie/itext/2.1.7>).

Abbiamo utilizzato un jar contenente una componente che ci è servita nella ricerca e cioè uno slider con due cursori

(<http://www.java2s.com/Code/Jar/p/Downloadprefusecore021jar.htm>).

Capitolo 4

Commenti finali

3.1 Conclusioni e lavori futuri

Riteniamo di aver svolto un buon lavoro nella realizzazione del progetto JEstates in quanto potrebbe già essere utilizzato in una piccola agenzia per avere uno storico organizzato delle proprietà affittabili. Future modifiche potrebbero rendere la creazione del volantino più accattivante con un pannello dedicato alla sua formattazione. Un altro aspetto che riteniamo aver implementato con successo per rendere JEstates più versatile è la struttura di alcune enumerations, in quanto la semplice modifica di esse rende possibile la scelta delle informazioni da salvare per immobili e clienti.

Appendice A

Guida utente

Il programma accoglie l'utente nell' interfaccia principale dove sarà possibile scegliere l' azione che si desidera effettuare tra: inserimento di un immobile, la ricerca di uno specifico, accesso alla sezione riguardante i clienti dove si potrà inserire di nuovi o ricercare tra quelli già registrati ed infine creare una locandina in formato pdf nel percorso file che si desidera.

Inserimento: in questa sezione è possibile scegliere delle immagini da caricare e compilare i campi necessari per descrivere l' immobile da salvare. Eventuali errori di digitazione nei campi saranno segnalati.

Ricerca: in questa sezione si possono ricercare gli immobili secondo due diverse modalità: ricerca tramite codice immobile oppure ricerca avanzata. Facendo doppio click su una delle righe della tabella si aprirà un pannello con le specifiche dell' immobile.

Clienti: analogamente alla sezione degli immobili si può inserire o ricercare dei clienti, con l' unica variante che il doppio click su una riga della tabella di ricerca, stavolta, permette la visualizzazione delle proprietà possedute da quel cliente.