MEDIARY

PROGRAMMAZIONE AD OGGETTI

Relazione del progetto

Anno accademico 2015/2016

Zecchin Giacomo (1070122)

4 settembre 2016

Indice

1 Scopo del progetto **2**

2 Funzionalità offerte **2**
2.1 Utenti 2
2.2 Media 2

3 Schema Logico **2**
3.1 Model 3

4 View **4**

1 Scopo del progetto

Mediary è il risultato di questo progetto per il corso di programmazione ad oggetti che aveva come scopo quello di creare un applicativo sviluppato in C++/Qt.

L'applicazione intende rappresentare uno spazio personale nel quale annotare tutte le serieTv o i film già visti, preferiti oppure col desiderio di vedere in futuro.
Mediary consente quindi di accedere singolarmente come utente per aggiungere nuovi *media* al proprio diario (*diary*).
L'unione di queste due parole, **Media** e **Diary**, va difatti a creare il nome dell'applicazione.

Il progetto è semplice ma allo stesso tempo funzionale, diretto e comprensibile per l'utente; vediamo nel seguito tutte le funzionalità.

2 Funzionalità offerte

2.1 Utenti

Avviata, l'applicazione presenta subito una finestra centrale per l'inserimento diretto dei dati necessari ad effettuare il **login** ed accedere immediatamente all'area personale.
In alternativa, se non si possiedono già delle credenziali, è presente un bottone per la **registrazione** poco più in basso, che farà entrare nella finestra dedicata alla creazione di un nuovo utente. In entrambi questi due casi sono presenti dei controlli di consistenza dei dati descritti più approfonditamente nella sezione *sez.* .
E' comunque data la possibilità di **modificare i propri dati** in un secondo momento.

2.2 Media

Le funzionalità offerte per i media sono invece quelle di **creazione, modifica, visualizzazione per tipo ed eliminazione**. (In dettaglio qui *sez*)

3 Schema Logico

Il *Design Pattern* utilizzato per la progettazione è il classico pattern **MVC** separando quindi il *Model* dalla *View* con l'ausilio due classi che formano il *Controller*.

Vediamo nelle sottosezioni seguenti le tre componenti e le classi che li compongono.

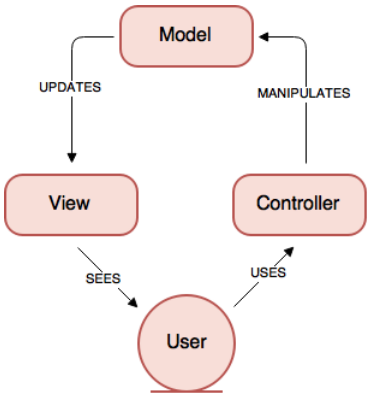

Figura 1: Interazione nel pattern MVC

3.1 Model

La **gerarchia G** che forma il modello dati del progetto è così rappresentata:

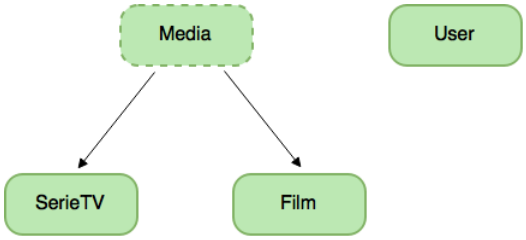

Figura 2: Gerarchia

4 View