

my Magazine

PROGRAMMAZIONE AD OGGETTI

Relazione del progetto

A.A 2018/2019

Fontolan Carlo

1099082

Indice

1 Scopo del progetto

2 Funzionalità

3 Schema logico

4 GUI

5 Polimorfismo

6 Informazioni per l'utilizzo

Scopo del progetto

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

Funzionalità

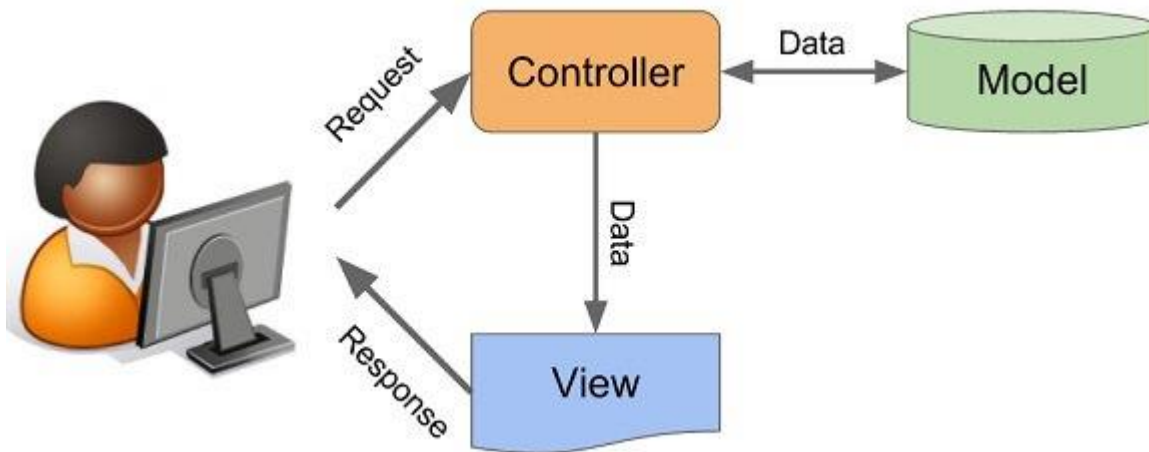
L'applicativo intende rappresentare la gestione di un magazzino di prodotti informatici composto da Smartphone, Notebook e Smartwatch.

L'applicazione offre un pannello di controllo gestito dall'amministratore che consente di inserire, modificare e rimuovere utenti e oggetti presenti nel magazzino.

myMagazine offre anche l'accesso al singolo utente che può venir registrato esclusivamente dall'amministratore. Essi potranno aggiungere, modificare e rimuovere i propri oggetti e cercarli in base a diversi criteri.

Schema logico

È stato utilizzato un design pattern Model-View-Controller che separa il modello dalla view grazie a 3 classi che formano il controller.



3.1 Gerarchia Polimorfa

La gerarchia che forma il modello è composta dalle seguenti classi:

Item: classe base astratta della gerarchia contenente i campi che rappresentano le informazioni “di base” di un prodotto informatico.

Al suo interno sono dichiarati i seguenti metodi virtuali puri:

- virtual void saveItem(QXmlStreamWriter& xmlWriter) const=0;

per eseguire il salvataggio degli oggetti all'interno del database.

- Virtual QString Type() const=0;

ritorno il tipo polimorfo dell'oggetto.

Notebook: classe concreta derivata pubblicamente dalla base.

Estende la classe la classe base con i campi privati GPU che mostra il tipo di scheda grafica presente, bool fanless che è true se il notebook è fanless e bool backlight che è true se è presente la tastiera retrioluminata.

Smartphone: classe concreta derivata pubblicamente dalla base.

Estende la classe base con i campi privati Cam1 e Cam2 che indicano i megapixel della fotocamera anteriore e posteriore; inoltre è presente un bool fingerprint che indica se la smartphone è disposto di sensore biometrico.

Smartwatch: classe concreta derivata pubblicamente dalla base.

Estende la classe base con i campi privati Sensor che indica il tipo di sensore presente nel dispositivo, Certificate che indica i certificati di impermeabilità ed infine il materiale del cinturino.

La seconda gerarchia polimorfa presente in myMagazine è composta dalle classi User, Admin e Manager. La classe User è definita astratta.

Queste rappresentano i due tipi di accessi che si possono eseguire sull'applicazione.

3.2 Contenitore

Container è una classe templetizzata esterna alla gerarchia sopracitata che contiene una completa definizione di un opportuno contenitore, con relativi iteratori che permettono inserimenti, rimozioni e modifiche di oggetti.

Il contenitore templetizzato è formato da due classi Nodo e Smartp annidate con un unico campo dati SmartP first.

Nella parte pubblica del contenitore è definita una classe Iterator con un campo dati SmartP che verrà utilizzato per iterare su tutti gli oggetti presenti nel contenitore.

3.3 Database

Classe concreta che rappresenta il database degli utenti ed oggetti. Database presenta due campi dati contenente i contenitori della due classi polimorfe. Questi contenitori vengono riempiti con le entità ricavate da userDatabase.xml e itemDatabase.xml tramite delle funzioni load; mentre, alla chiusura delle view (manager o admin) vengono richiamate le funzioni di save saveltem(QXmlStreamWriter&) e writeUser(QXmlStreamWriter&) chiamando la funzione adatta al tipo dinamico del puntatore dell'oggetto nel container.

GUI

Tutte le view sono state create tramite lo strumento di Qt Designer. La gestione dei widget è stata impostata prevalentemente tramite la gestione dei signals e degli slot.

Polimorfismo

Il codice prodotto risulta estendibile. È disponibile l'aggiunta di altre classi derivate dalla base astratta (per esempio Stampante oppure Tablet) sia per quanto riguarda gli utenti.

Questo estende le funzionalità per tipo di utente e oggetto evolvendo il codice rendendolo polimorfo in alcune parti, come per esempio il salvataggio degli oggetti ma anche degli utenti.

Informazione per l'utilizzo

Il software dispone di un account amministratore il quale può nominare a sua volta nuovi amministratori

Compilando con la build Qt sarà necessario copiare all'interno della cartella build appena creata la cartella “db” contenente i file itemDatabase.xml e userDatabase.xml;

altrimenti non sarà possibile caricare i database.

Credenziali per l'accesso come amministratore

- Username: admin
- Password: password

Credenziali per l'accesso utente

- Username: cfontola
- Password: password

- Username: mario
- Password: password

Ambiente di sviluppo

Qt Creator 4.4.1(opensource)

Based on Qt 5.9.2 (MinGW, 32 bit)

Ore impiegate:55