

RELAZIONE

Project REGISTRO

by Francesco Fasolato - 1069395

Versione Qt: 5.5.1

Ambiente di Sviluppo: Qt Creator 3.6.0

Compiler: Clang 7.0

Sistema Operativo della macchina utilizzata: OSX El Capitan 10.11.6

ABSTRACT

Questo progetto è stato realizzato per svolgere le funzioni di un registro elettronico digitale, simile a quelli che vengono dati in gestione alle scuole primarie e secondarie d'Italia.

Gli utenti ai quali è permesso l'accesso si dividono in 3 tipologie distinte:

- Membri del personale ATA
- Membri del corpo Docenti
- Dirigenti degli Istituti

Dal momento in cui vengono inseriti i dati personali di ogni utente, verranno forniti all'utente in questione i dati (USERNAME e PASSWORD), indispensabili per l'accesso alla propria area personale all'interno di Project REGISTRO.

Agli utenti non è permessa la registrazione, poiché questa verrà fatta automaticamente dal Dirigente del proprio Istituto, al momento della creazione virtuale dei membri del personale.

Gli utenti base, quali membri del corpo Docenti e del personale ATA, una volta effettuato l'accesso all'interno di Project REGISTRO, potranno avere accesso ai propri dati personali e ai dati del plesso a cui afferiscono.

I Dirigenti invece, godono delle funzionalità aggiuntive di inserimento e rimozione di un membro del personale, nonché di inserimento e rimozione dei dati concernenti uno dei plessi da loro gestiti.

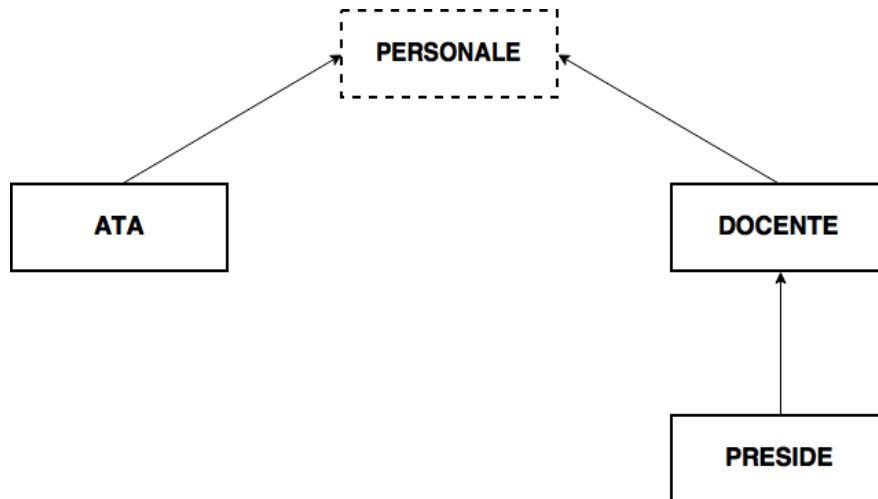
Per quanto riguarda la rimozione di un'Istituto dal database, Project REGISTRO controllerà automaticamente che non vi sia nemmeno un utente che afferisca al plesso in questione prima di eliminarlo. In caso contrario bisognerà prima provvedere a rimuovere gli utenti che afferiscono a quell'Istituto.

Sempre per i Dirigenti vengono anche messe a disposizione le liste di tutti i membri del personale e di tutti i plessi contenuti all'interno del database di Project REGISTRO. Per ogni membro di entrambe le liste, al Dirigente è permesso avere accesso a tutte le informazioni inserite nel database.

L'utilità di Project REGISTRO risiede nella possibilità da parte di tutti gli utenti di tenere sotto controllo i propri dati personali, di sapere il proprio orario lavorativo ed il proprio stipendio mensile. Per i Dirigenti, i quali svolgono le funzioni di amministratori, Project REGISTRO offre la possibilità di tenere sotto controllo i propri membri del personale, registrando nuove assunzioni, rimuovendo dal database chi è stato congedato e controllando tutti gli orari lavorativi del personale.

CARATTERISTICHE DEL PROGETTO

GERARCHIA:



La gerarchia raffigurata rappresenta la suddivisione delle classi principali della parte MODEL utilizzata in Project REGISTRO. Personale è una classe base astratta che contiene informazioni comuni a tutti i membri del personale. Inoltre contiene i metodi virtuali:

- stipendio() const
- writeTipo(QXmlStreamWriter&) const
- writeSpecifiche(QXmlStreamWriter&) const
- openRightView() const

L'implementazione di questi metodi, diversa per ogni classe derivata da “personale”, è stata pensata appositamente per facilitare l'esecuzione del programma in presenza di puntatori con tipoStatico “personale* ”, ma con tipoDinamico diverso, ovvero uno dei tipi delle classi derivate. In particolare “writeTipo(QXmlStreamWriter&) const” e “writeSpecifiche(QXmlStreamWriter&) const” sono utili per scrivere dati specifici delle classi derivate all'interno del database.

Il metodo “stipendio() const” calcola lo stipendio mensile percepito da ogni singolo utente, utilizzando campi dati specifici per ogni classe.

Il metodo “openRightView() const” è utile per scegliere, in base ad un puntatore ritornato dal contenitore, con tipo statico classe base ma con tipo dinamico classe derivata, quale sia la corretta visualizzazione da fornire all'utente, evitando un'inutile ulteriore richiesta della tipologia specifica al momento del Log in.

Sono presenti altre due classi MODEL, esterne alla gerarchia rappresentata sopra: “plesso” e “orarioSettimanale”, che rappresentano rispettivamente i dati dei plessi ai quali afferiscono gli utenti presenti all'interno di Project REGISTRO e l'orario settimanale che questi devono rispettare.

CONTAINER:

Project REGISTRO utilizza due tipi di container simili per memorizzare oggetti di tipo “personale” (classe base della gerarchia), memorizzati in “ista” e “plesso”, memorizzati in “listaPlessi”. Il container per gli oggetti di tipo “personale”, come richiesto dalla definizione specifica di container, è stato realizzato come un template di classe.

La struttura dati rappresentata all'interno di entrambi i container è una lista singolarmente linkata di oggetti di tipo “nodo” (classe interna ai container realizzata appositamente).

I container comunicano direttamente con i due file che fungono da database database: “lista” legge e memorizza i dati in “DB.xml” e “listaPlessi” in “plessoDB.xml”

DATABASE:

I database utilizzati sono entrambi scritti in formato xml. Nel file “DB.xml” l'organizzazione degli elementi stampati cambia in base alla tipologia di utente memorizzata. All'interno del tag “info_generali” sono contenute le informazioni comuni a tutti gli utenti, mentre le informazioni contenute all'interno di “info_specifiche” cambiano a seconda della tipologia di utente. La funzione di lettura del container “lista<personale>” si occuperà di distinguere la tipologia di utente da creare grazie al tag “tipologia”, tag che viene utilizzato anche dalla funzione di scrittura.

GRAFICA:

Il progetto e' stato costruito seguendo il pattern MVC, separando la model dalla view con l'ausilio di una classe di controllo per quanto riguarda le classi più grandi, per facilitare la gestione di queste.

Il progetto si divide graficamente nelle seguenti parti:

1. Inizio:

comprende i seguenti file che sono comuni per tutti gli utenti:

- startview
- c_startview
- loginView

Come si potrà facilmente intuire il file “startview” contiene la schermata principale del programma, mentre “c_startview” è il suo file controller. Per poter permettere l'accesso agli utenti, viene subito invocata la classe “loginView”, che si preoccupa di effettuare il Login, controllando la corrispondenza fra username e password. Una volta confermata la correttezza dei campi compilati, “loginView” costruisce un puntatore ad oggetto di tipo “personale”, la classe base dell'intera gerarchia di utenti. Grazie a questo puntatore sarà possibile sfruttare un metodo polimorfo che garantirà l'apertura della corretta sezione, a seconda che abbia effettuato l'accesso un membro del personale ATA, del corpo Docenti o un Dirigente.

2. Sezione Dirigenti:

comprende i seguenti file:

- PresideView
- C_PresideView
- PresideContent
- C_PresideContent
- insertPlesso
- insertAta
- insertDocente
- insertPreside

La vista principale della sezione Dirigenti è contenuta all'interno del file “PresideView”, il cui controller è “C_PresideView”. La classe è supportata da “PresideContent”, che, sempre con il suo controller “C_PresideContent”, ha lo scopo di aggiungere widget e funzionalità a “PresideView”. Le classi “insertPlesso”, “insertAta”, “insertDocente” ed “insertPreside” sono widget realizzati appositamente per permettere l'inserimento di un nuovo Istituto, membro del personale ATA, membro del corpo Docenti e Dirigente rispettivamente.

3. Sezione membri del corpo Docenti:

comprende i seguenti file:

- DocenteView

Il file “DocenteView” fornisce al membro del corpo Docenti una visuale completa di tutte le informazioni che lo riguardano e delle informazioni che riguardano l'Istituto a cui afferisce.

4. Sezione membri del personale ATA:

comprende i seguenti file:

- AtaView

Il file “AtaView” fornisce al membro del personale ATA una visuale completa di tutte le informazioni che lo riguardano e delle informazioni che riguardano l'Istituto a cui afferisce.

INFORMAZIONI UTILI:

Credenziali per l'accesso a Project REGISTRO:

– MEMBRO DEL PERSONALE ATA:

- USERNAME: TLucci01
- PASSWORD: zlatan64

– MEMBRO DEL CORPO DOCENTI:

- USERNAME: PMerli01
- PASSWORD: ciao123

– DIRIGENTE:

- USERNAME: CNanni03
- PASSWORD: banjo93