LinQedIn

di Antonio Iacobucci

Progetto di Programmazione ad Oggetti 2014-2015

1 Introduzione

1.1 Scopo del documento

Questo documento descrive sinteticamente la struttura e le scelte progettuali ritenute più significative per lo sviluppo del progetto didattico LinQedIn, per il corso di Programmazione ad Oggetti aa. 2014-2015.

1.2 Scopo del progetto

Lo scopo del progetto è lo sviluppo in C++/Qt di un sistema minimale per l'amministrazione ed utilizzo tramite interfaccia utente grafica di un database di contatti professionali ispirato a LinkedIn.

1.3 Ambiente di Sviluppo

- Sistema operativo di sviluppo: Windows 8.1
- Versione del compilatore: Qt Creator 3.0.1
- Versione della libreria Qt: 4.8.4

2 Scelte Progettuali

Il pattern architetturale di base è un MVC. L'unica eccezione al pattern è la classe Widget che oltre alla View presenta anche alcune funzioni di Controller all'interno di widget.cpp, per la gestione del cambio pagina o alcune operazioni sui dati inseriti.

2.1 Modello

LinQedIn prevede tre tipologie di account, sviluppate attraverso una classe base astratta Utente dalla quale sono definite tre tipologie di utente:

- UtenteBasic;
- UtenteBusiness;
- UtenteExecutive.

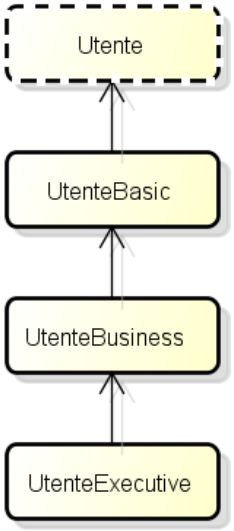

Figure 1: Gerarchia classe Utente

Esse sono strutturate mediante una gerarchia e presentano funzionalità crescenti, prevedendo quindi anche un'eventuale estensione del codice per aggiungere funzionalità alle classi più specializzate.

Ogni utente è caratterizzato da Username, Profilo e Rete. Username è a sua volta una classe che racchiude all'interno la coppia username-password necessaria per l'autenticazione. Profilo ha al suo interno un oggetto Info che contiene tutte le informazioni dell'utente: nome, cognome, email, città, sesso, data di nascita, qualifiche(lista di stringhe) e ovviamente tipologia di account. Infine Rete è la classe in cui è presente la lista dei contatti di un utente, immagazzinati all'interno di una lista di Username.

Tutte le informazioni sono memorizzate nella classe Database attraverso una lista di puntatori smart polimorfi alla classe base Utente. Il database si occupa inoltre del salvataggio e caricamento di tutte le informazioni tramite un file XML utilizzando i metodi save() e load(string). Essi sono gestiti utilizzando le classi fornite da Qt, rispettivamente QDomDocument per il caricamento e QXmlStreamWriter per il salvataggio.

L'applicazione è sviluppata sia lato Admin che Client. Il lato admin è ovviamente destinato all'amministratore del sistema. Egli avrà la possibilità di inserire, ricercare o rimuovere un utente dal sistema, oppure cambiare il tipo di account di un utente.

Dal lato client saranno invece disponibili i servizi destinati all'utente iscritto e autenticato nell'applicazione. Le funzioni principali sono la visualizzazione e modifica del proprio profilo utente, la visualizzazione e modifica della propria lista di contatti e la ricerca di un utente per poterne visualizzare le informazioni.

La GUI dell'applicazione è sviluppata attraverso un unico Widget che cambia layout dinamicamente utilizzando le connect sui bottoni all'interno delle pagine. Essa verrà discussa più approfonditamente nella prossima sotto-sezione.

2.2 GUI

Avviando l'applicazione il sistema creerà un nuovo database, settandone il path e caricando le informazioni dal file XML. Successivamente verrà creata la barra del menù e il form di autenticazione per permettere all'utente di autenticarsi. Qualora non disponesse ancora di un account personale l'utente potrà registrarne uno nuovo.

Dalla barra del menù saranno disponibili le funzioni login/registrazione qualora l'utente non sia autenticato; in caso contrario sarà mostrato il bottone di logout. Le altre funzioni disponibili dal menù sono esci dall'applicazione, fullscreen/normal view e about. La ricerca viene invece resa disponibile solo per l'utente autenticato.

L'admin, una volta autenticato avrà a disposizione le operazioni elencate nella sezione precedente, potrà quindi inserire, cercare, rimuovere o cambiare il tipo di un account di un utente.
Dal lato client, effettuando l'autenticazione l'utente potrà visualizzare il proprio profilo. Da qui potrà cliccare sul bottone relativo alla modifica del profilo personale oppure sul bottone per la visualizzazione della propria rete di contatti.

Dalla schermata di modifica potrà editare tutte le informazioni relative al proprio account, esclusi lo username e il tipo dell'account, oppure aggiungere/rimuovere una qualifica. Inoltre cliccando sol bottone di cambio password verrà indirizzato ad una schermata in cui poter modificare la propria password.
Dalla schermata di visualizzazione della propria rete verranno mostrati tutti i propri contatti, sarà possibile aggiungerne altri, rimuoverli o visualizzare il loro profilo.

Per tutte le operazioni effettuate verrà mostrato, in fondo alla GUI all'interno di una statusBar, un messaggio che confermerà l'esito dell'operazione o eventuali errori, informazioni e suggerimenti, mostrando inoltre all'utente l'eventuale campo errato tramite un contorno rosso sulla casella di testo.
Queste funzioni sono state sviluppate nella maggior parte dei casi utilizzando lo slot error(int) che a seconda del valore intero ricevuto segnala il messaggio corrispondente ed effettua le modifiche alle caselle di testo.

Ritengo opportuno descrivere brevemente anche lo slot redirect() all'interno della classe Widget. Questo slot raggruppa numerosi slot, che agiscono coerentemente a seconda del sender(), ovvero l'oggetto che ha chiamato la SIGNAL. In questo modo molti bottoni all'iterno dell'applicazione chiamano lo stesso slot redirect() che si occupa di eseguire la funzione corretta.

2.2.1 Ricerca

La funzionalità di ricerca di un utente viene utilizzata per visualizzarne le informazioni personali dell'utente cercato. La ricerca sarà disponibile solo ad un utente autenticato e le informazioni mostrate dipenderanno da diversi fattori.

La ricerca dal lato amministratore fornisce tutte le informazioni disponibili dell'utente ricercato, compresa la rete di contatti.

Dal lato client invece l'output può variare a seconda delle seguenti variabili.

Se l'utente ricercato non appartiene alla propria rete di contatti, verranno mostrati solamente username, nome e cognome. Se l'utente dovesse cercare il proprio username sarà semplicemente reindirizzato al proprio profilo.

Se invece l'utente cercato è uno dei propri contatti, l'output dipenderà dalla tipologia del proprio account. Ad un utente Basic verranno mostrate le informazioni escludendo qualifiche e contatti. L'utente Business potrà invece visualizzare anche le qualifiche dell'utente ricercato. Infine l'utente Executive, come l'amministratore, potrà visualizzare le informazioni complete, includendo quindi anche la rete di contatti.