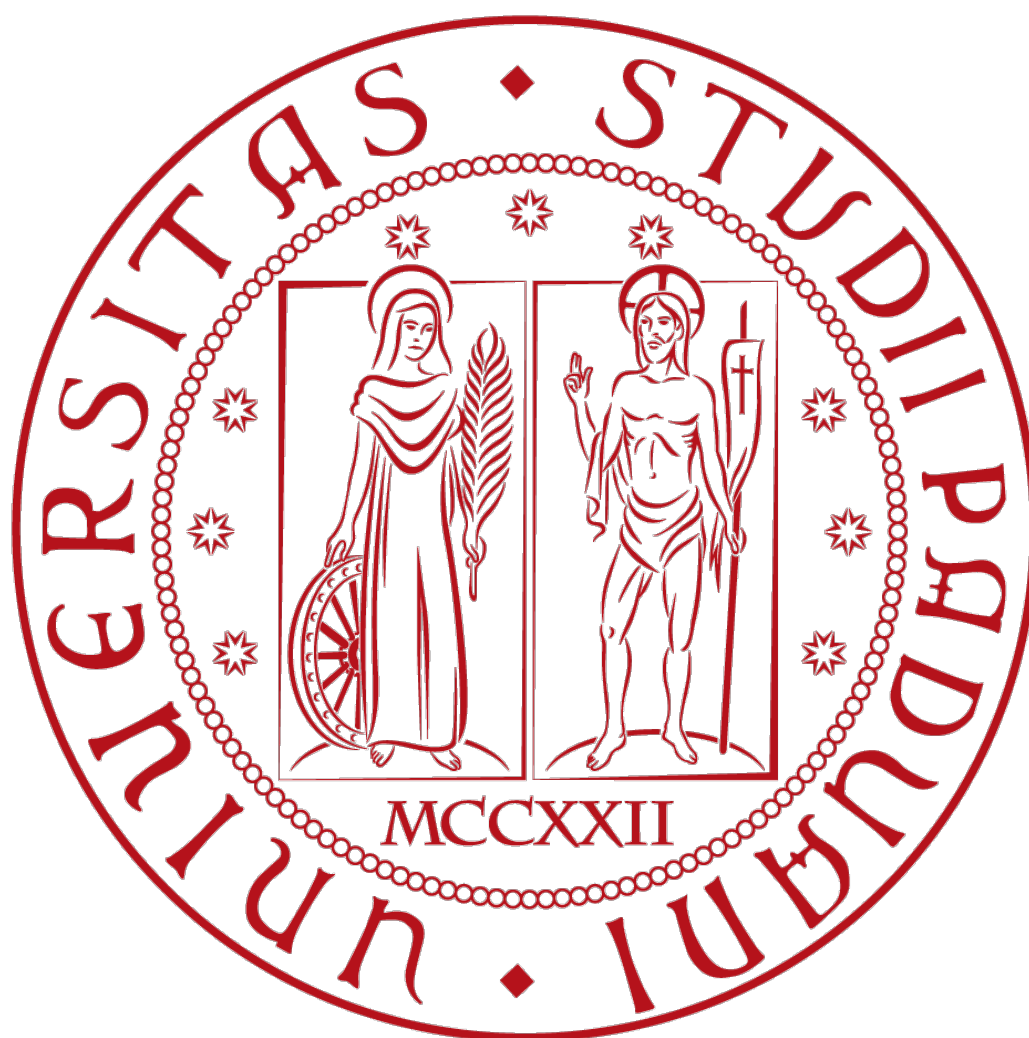


Relazione Progetto Programmazione ad Oggetti 2014/2015

Walter Sandon 1009138

Agosto 25, 2015



Ambiente di sviluppo

- Compilatore GCC 4.2.1, 64 bit
- Versione Qt Creator: 3.2.2
- Versione Qt: 5.3.2
- Sistema Operativo: Yosemite 10.10.5

Materiale consegnato

La cartella consegnata nominata Walter_Sandon contiene i seguenti elementi:

- LinQedIn: la cartella che contiene tutti i file .h, .cpp e un file LinQedIn.pro necessario per la compilazione del progetto in quanto quello originale è stato modificato.
- utentiLinQedIn.xml: file contenente un piccolo database di prova nel linguaggio xml.
- passwordLinQedIn.xml: che contiene la password dell'amministratore.
- relazione.pdf: che è la presente relazione del progetto.

1. Introduzione

Lo scopo di questo progetto è lo sviluppo in C++/Qt di un sistema minimale per l'amministrazione ed utilizzo tramite interfaccia grafica di un (piccolo) database ispirato a LinkedIn, il social network professionale più diffuso al mondo. Un potente strumento che permette di allacciare rapporti lavorativi condividendo le proprie esperienze con altri colleghi presenti nel social, che contribuiranno a far crescere la propria rete di contatti, aumentando di fatto la propria visibilità.

L'iscrizione a LinkedIn è gratuita ma nel contempo offre dei servizi a pagamento che permettono all'utente che sta cercando un nuovo partner per un progetto, di poter ottenere dei risultati di ricerca via via sempre più dettagliati sulla persona che sta cercando.

In particolare si vuole modellare i principali servizi che LinkedIn mette a disposizione ai propri utenti, mediante l'implementazione di tre diverse tipologie d'utenza: Basic, Business, Executive, rendendo disponibili le principali funzionalità sia dal lato amministratore che dal lato cliente.

Un amministratore potrà inserire un nuovo utente, rimuoverlo, visualizzarne il profilo completo nonché modificarne la tipologia d'utenza. Un utente potrà invece visualizzare il proprio profilo, aggiornarlo o modificarlo ove fosse necessario, aggiungere/rimuovere contatti alla propria rete infine ricercare un utente all'interno del database e conoscerne il profilo in base alla propria tipologia d'utenza.

Nello specifico se sono un utente Basic potrò solo visualizzare soltanto lo username, il nome e il cognome dell'utente cercato. Se Business oltre a queste informazioni potrò visualizzare le esperienze lavorative, le competenze, le lingue conosciute e i vari titoli di studio. Infine se sono Executive oltre a questo appena descritto posso visualizzare anche la rete di contatti.

2. Parte logica

Di seguito verranno elencate tutte le classi che fanno parte della logica del progetto e le relative funzioni principali:

I. Data

Cercando di mantenere il progetto meno vincolato possibile alla piattaforma Qt se non per la parte grafica, ho implementato la classe Data che rappresenta appunto una qualsiasi data ed è formata da tre campi interi, giorno, mese, anno. La classe prevede i metodi di visualizzazione e modifica dei tre campi appena menzionati, l'overloading di vari operatori di confronto e un metodo che controlla se la data appena inserita è effettivamente un data esistente.

II. Competenza

Questa classe raffigura la competenza che un utente ha acquisito finora ed è rappresentata dal nome della competenza e dalla valutazione espressa in un intervallo compreso tra 1 e 10. Prevede soltanto i metodi di visualizzazione e modifica dei campi.

III. Lavoro

La classe Lavoro raffigura una singola esperienza nel campo lavorativo dell'utente ed è rappresentata dal nome dell'azienda in cui ha lavorato e il relativo periodo. Anche in questo caso sono presenti solo metodi per la visualizzazione e modifica.

IV. Istruzione

La classe Istruzione raffigura un titolo di studio acquisito in seguito a un percorso di studio. La classe è rappresentata dal nome del titolo di studio, da due interi che rappresentano l'anno d'inizio e la fine di tale percorso di studi.

V. Lingua

La classe Lingua rappresenta le lingue conosciute dall'utente. E' composta dal nome della lingua seguita da tre campi stringa che rappresentano il livello di conoscenza nell'ambito della comprensione, del parlato e dello scritto. Il livello di ogni lingua può assumere solo i seguenti valori: A1, A2, B1, B2, C1, C2.

VI. Profilo

La classe Profilo rappresenta il profilo di un utente ed è stato scelto di rappresentarla con dei vettori di puntatori a Competenze, Lavoro, Istruzione, Lingua. In questo caso oltre la visualizzazione e la modifica dei campi è stato ridefinito il distruttore profondo per non lasciare garbage, in quanto i campi dati sono puntatori ad oggetti allocati nello heap e non oggetti veri e propri.

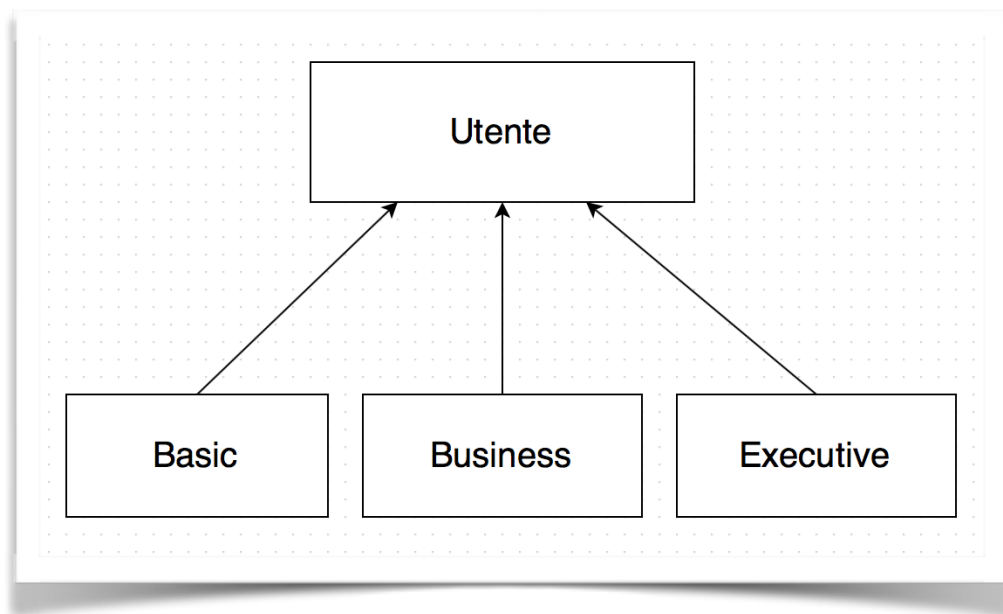
VII. InfoPers

Questa classe rappresenta le informazioni generali dell'utente ovvero nome, cognome, data di nascita e una descrizione e sono presenti solo metodi per la visualizzazione e la modifica

VIII. Rete

Rappresenta un singolo contatto di un utente ed è rappresentato tramite una stringa che rappresenta lo username del contatto.

IX. Utente



La classe utente è una classe base polimorfa e astratta. Polimorfa perché contiene un distruttore virtuale e astratta in quanto contiene un paio di metodi virtuali puri come la ricerca di un utente e la visualizzazione del costo mensile del servizio a seconda della tipologia d'utenza. Le classi *Basic*, *Business*, *Executive* sono le classi concrete derivate direttamente da Utente le quali differiscono sostanzialmente per la diversa implementazione di questi due metodi. La classe utente contiene una campo dati stringa per lo username univoco, un puntatore a Profilo, un puntatore ad InfoPers e un vector di puntatori a rete e di conseguenza è stato ridefinito il distruttore per lo stesso motivo che è stato ridefinito nella classe Profilo.

Oltre a qualche metodo di servizio, contiene la classe annidata Functor, che grazie a questo particolare oggetto le classi derivate effettueranno un diverso overriding del metodo di ricerca trova_tipo_utente. Di conseguenza se l'utente che effettua la ricerca è Basic, il metodo istanzierà un funtore di tipo 1, restituendo una lista che è stata passata per riferimento contenente solo i campi nome, cognome e data di nascita. Mentre per gli altri tipi di utente rispettivamente con un funtore di tipo 2 e 3, la ricerca sarà incrementale.

X. DataBase

E' la classe collezione di tutti gli utenti presenti e si è scelto come contenitore STL un vector di puntatori ad Utente in quanto il vector, generalizzando un pò l'idea di un array, ha la possibilità di poter accedere in maniera casuale agli elementi in tempo $O(1)$, quindi sembra essere il contenitore migliore in questo particolare contesto di ricerca, modifica e rimozione di un utente iesimo.

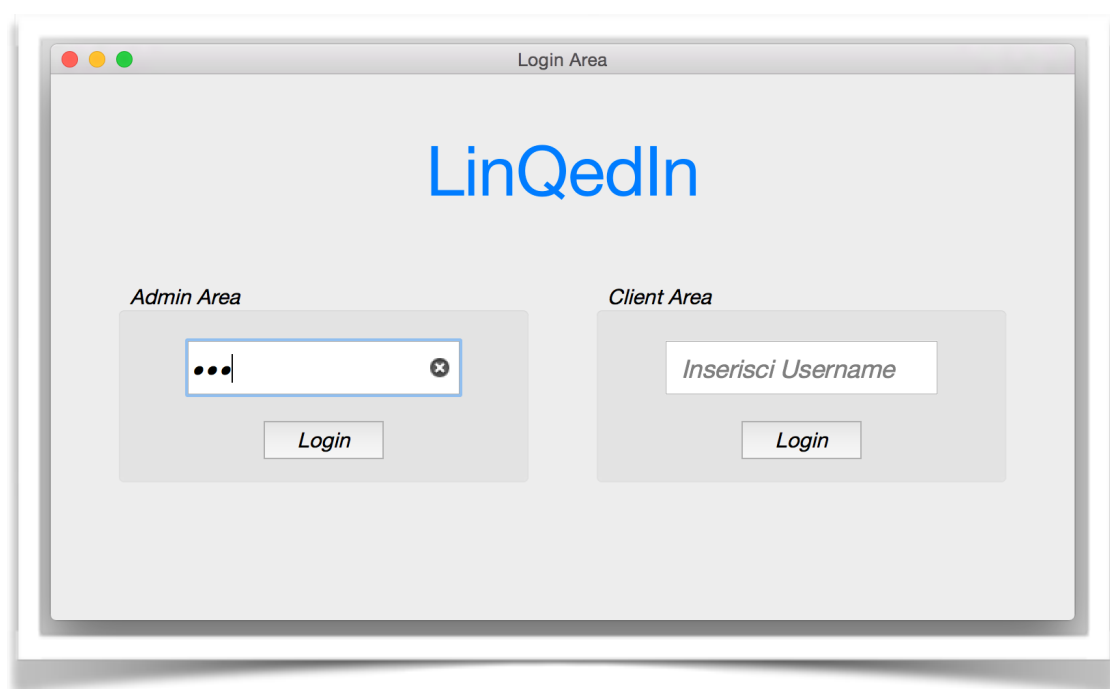
La classe offre, oltre ai metodi di inserimento e rimozione di un utente, anche metodi d'utilità per la parte grafica come il metodo che restituisce l'intero array in modo da poter visualizzare e stampare tutte le informazioni dell'utente e il metodo che cambia la tipologia d'utenza utile per la parte amministratore. Per quanto riguarda invece il salvataggio e il caricamento da file sono state usate due librerie offerte da Qt ovvero la QXmlStreamReader per lettura mentre QXmlStreamWriter usata per scrivere su file xml.

Nei metodi per la costruzione e lettura dei file .xml ho inserito il path assoluto quindi non necessita di nessuna modifica di path nei quattro metodi nel momento in qui si proverà il progetto su altri computer.

3. Interfaccia Grafica

Il tool Qt Designer non è stato utilizzato nello sviluppo delle interfacce grafiche ma solo per avere un riferimento grafico. L'interfaccia grafica si divide in tre classi ovvero in lato admin e lato client, e la finestra principale, dalla quale si può accedere ad uno dei due lati. Di seguito verranno descritti i principali servizi offerti dalle classi

I. Main Window



La main window è la finestra principale del programma dalla quale si può accedere al lato Admin inserendo una password o al lato Client tramite l'inserimento del proprio username.

II. Lato Amministratore

La login dell'amministratore provoca l'apertura della finestra AdminWindow. Nella parte superiore troviamo il box *utenti iscritti* nel quale troviamo da un lato la lista completa di tutti gli utenti e dall'altra tutte le loro informazioni compreso anche la loro tipologia d'utenza. Inoltre al centro di questo box troviamo anche uno spazio di ricerca per username di un utente iscritto.

Nella parte inferiore troviamo il box dell'inserimento di un nuovo utente da parte dell'amministratore che potrà solo inserire username, nome, cognome, data di nascita e ovviamente il tipo. In questo modo l'utente sarà registrato nel database così potrà loggarsi nella parte client per poter completare il proprio profilo.

Infine, in caso di necessità è stato implementato il box del cambio password con un controllo sulla vecchia password. Questa password inizialmente verrà settata con Admin in un file `passwordLinQedIn.xml`.

Le modifiche definitive al Database vengono fatte all'atto dell'inserimento di un nuovo utente o eventualmente nel cambio della password. Il pulsante ESCI serve soltanto per uscire dal lato admin.

III. Lato Cliente

Cliente

Informazioni Personali

Tipo: Basic

Username: fmin

Nome: Fabio

Cognome: Minichino

Data di Nascita: 05.09.1983

Descrizione: amichevole

Tariffa

Il Costo Mensile è di: 0 €

Competenze

	Argomento	Livello
1	telefonia	9

Esperienze Lavorative

	Esperienza	Inizio	Fine
1	ikea	01.12.2000	01.06.2001
2	tre	01.07.2010	01.07.2010

Titoli Di Studio

	Percorso	Inizio	Fine
1	scientifico	1995	2000
2	ingegneria	2000	2010

Lingue

	Lingua	Compr	Parlato	Scritto
--	--------	-------	---------	---------

Contatti

wsandon

Contatto LinQedin

Ricerca Contatti

Username:

Cerca

Username :

Aggiungi Contatto

Salva E Esci

Rimuovi

Dopo aver digitato il proprio username nella MainWindow un utente può accedere al proprio profilo nella finestra chiamata ClientWindow.

Questa finestra è stata suddivisa in due gruppi, quello delle informazioni personali dove un utente può aggiungere o eventualmente modificare il proprio profilo, come ad esempio nome, cognome, data di nascita e tutte le altre componenti del profilo come le proprie competenze, esperienze etc. Gli unici due campi che un utente non può modificare sono il tipo d'utenza in quanto lo può fare solo l'amministratore e il proprio username. Troviamo inoltre il pulsante Tariffa che permette di visualizzare il costo mensile del servizio a seconda del tipo d'utente in questione.

In ogni sezione del profilo ci sono i pulsanti aggiungi e rimuovi che consentono aggiornare il proprio curriculum. Grazie al tasto *aggiungi* viene inserito nella sezione dell'utente loggiato quella parte del profilo che vuole aggiornare invocando un costruttore di default. Il pulsane *rimuovi* lavora in direzione opposta ovvero distrugge quella parte di profilo appena selezionata. Le modifiche definitive al proprio profilo nel file xml saranno effettuate tramite il pulsante Salva E Esci.

Nella parte destra troviamo la parte della ricerca di contatti che potranno fare parte della nostra rete e grazie al pulsante aggiungi contatto si potrà aggiungere tale contatto alla propria rete.

4. Compilazione

Il progetto che è stato testato anche nei computer del laboratorio del Paolotti, compila ed esegue correttamente. Come accennato nella parte iniziale della relazione per un corretto funzionamento del progetto è necessario includere anche il file LinQedin.pro altrimenti non compila. Nello specifico con la shell di comandi posizionarsi nella cartella LinQedin ed eseguire i seguenti comandi: qmake, make e alla fine eseguire il programma.