

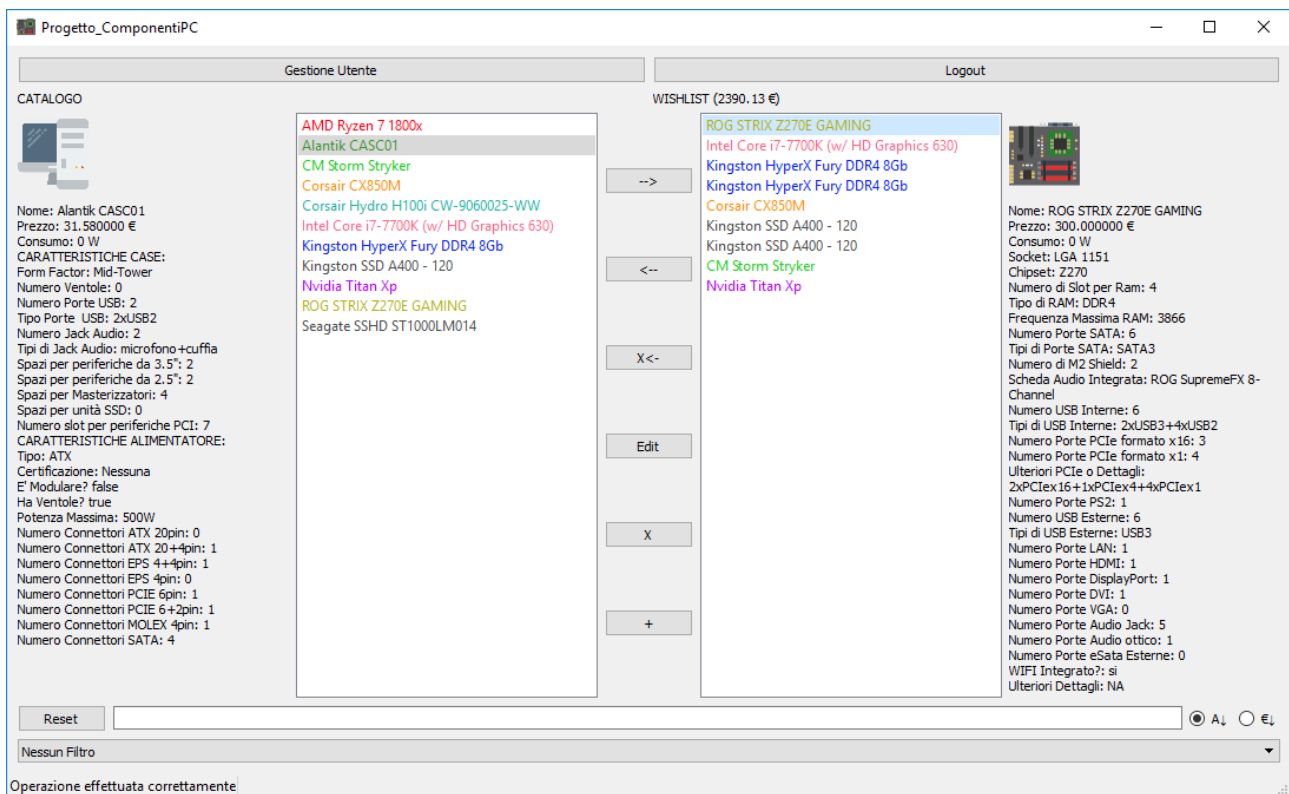
Programmazione ad Oggetti

Relazione di Cavazzana Damiano

Matricola 1122246

AA 2016/2017

Progetto: "Componenti PC"



(Screen della finestra principale)

SCOPO

Il progetto vuole andare a creare un programma con il quale l'utente finale possa visionare e scegliere componenti di pc desktop (fissi) da assemblare.

Le componenti presenti sono le principali parti che costituiscono un pc fisso: processore, scheda madre, memorie ram, scheda video, dissipatore, hard disk, alimentatore.

Ogni utente si può registrare autonomamente dalla finestra di login come utente comune senza particolari privilegi, una volta loggato ha una propria wishlist (letteralmente "lista dei desideri") dove può annotare i componenti scelti da un catalogo completo mentre ne cerca altri da confrontare o da aggiungere; questa lista rimane salvata anche a chiusura del programma.

Per semplificare la ricerca delle componenti è possibile filtrare la lista per tipo di componente, cercare nomi specifici e ordinarla per nome o per prezzo.

Mentre ogni utente può aggiungere e rimuovere componenti dalla propria wishlist come vuole, per gestire il catalogo completo è necessario avere dei permessi.

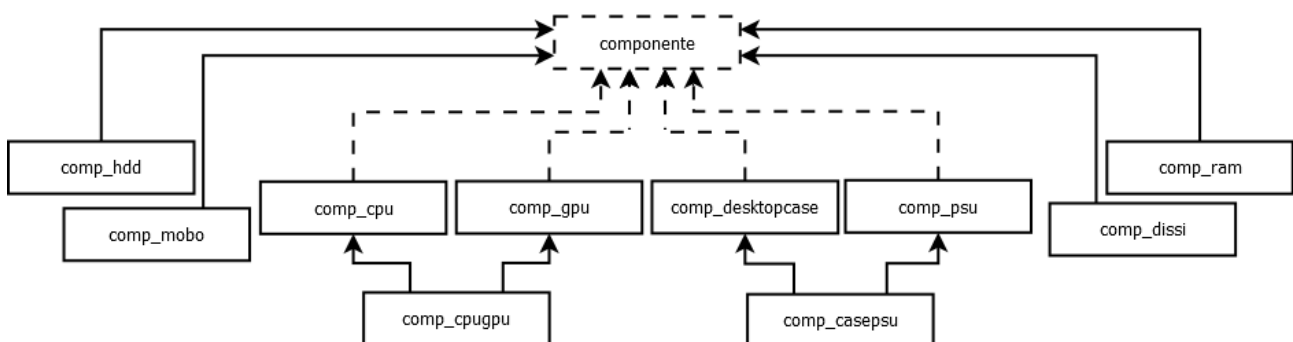
I permessi che distinguono gli utenti comuni dagli amministratori sono due:

- La possibilità di gestire il catalogo, posseduta da amministratori e tecnici, con la quale è possibile rimuovere e modificare componenti dal catalogo completo.
- La possibilità di gestire gli altri utenti, posseduta solo dagli amministratori, con la quale è possibile promuovere, declassare o eliminare gli altri utenti concedendo o togliendo loro i permessi (non è possibile però modificare i propri permessi o eliminare sé stessi); gli amministratori possono anche eventualmente modificare il loro profilo (ID, Nome, Cognome, Password - quest'ultima rimane comunque non visibile a nessuno -) qualora fosse necessario.

GERARCHIE DI TIPI USATE

Il progetto contiene 3 gerarchie di tipo principali: 2 appartenenti alla parte logica e una alla parte grafica.

GERARCHIA COMPONENTI



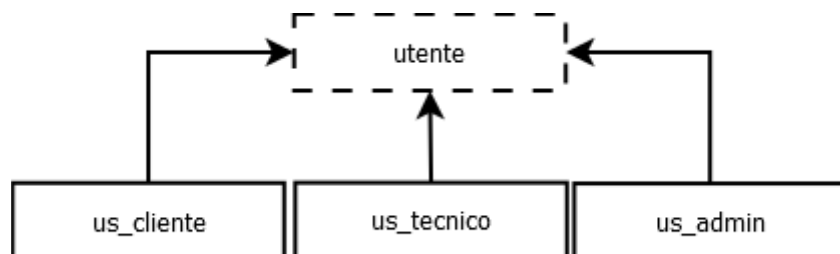
La prima gerarchia nonché la principale rappresenta l'insieme di tutte le componenti del pc e corrisponde al contenitore C richiesto nei vincoli del progetto.

La classe madre astratta "componente" è un componente generico contenente solo le informazioni in comune a tutte le componenti: nome, prezzo e consumo, per poi diramarsi in varie altre classi che contengono ciascuna i dettagli in più necessari a descrivere le caratteristiche del componente specifico.

Inoltre, sono considerati i casi particolari di cpu con gpu integrata (particolarmente diffuso per la maggior parte dei processori Intel) e case con alimentatore integrato (generalmente presente nei case dalle dimensioni molto piccole o particolarmente economici), che costituiscono le due classi diamante di questa prima gerarchia.

In generale ogni classe possiede metodi per esportare e importare facilmente il contenuto per le varie applicazioni solitamente richieste: oltre ai semplici set e get sono presenti metodi per stampare il contenuto anche come stringa - *toString()* e *stampaTutto()* - o per importare/esportare su un file xml: *stampaElementoXml()* ereditato dalla classe madre, *stampaContenutoXml()* e *stampaNomeXml()* per quanto riguarda l'esportazione e *importaDettagliXml()* per quanto riguarda l'importazione, che sono praticamente gli unici metodi di queste classi che utilizzano codice non appartenente alla libreria stl ma alla libreria qt.

GERARCHIA UTENTI



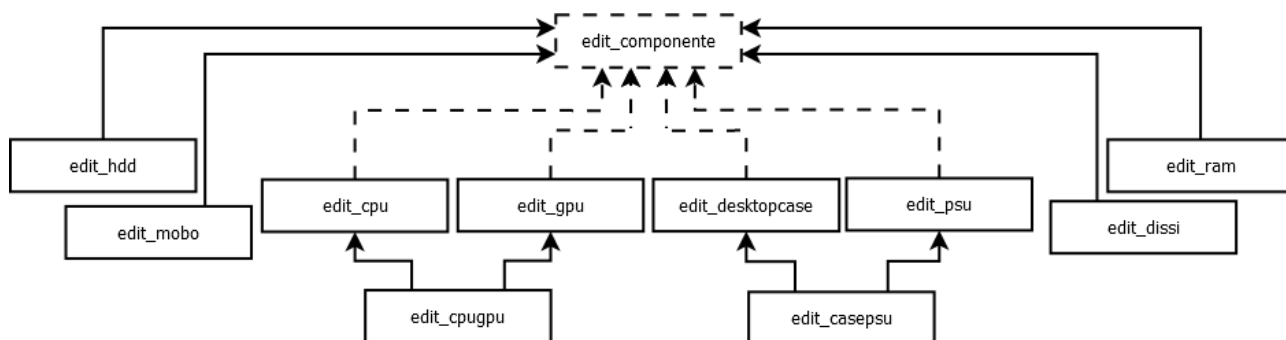
Questa piccola gerarchia della parte logica si occupa di rappresentare i vari gradi degli utenti del programma.

La classe madre astratta "utente" contiene di per sé già tutte le informazioni richieste per rappresentare un utente: ID, nome, cognome e password, oltre ad un contenitore di tipo "listcomp", contenente a sua volta puntatori agli oggetti della gerarchia componenti, che rappresenta la wishlist dell'utente.

Le sottoclassi si occupano quindi di suddividere i permessi concessi ai vari gradi di utente, dal cliente senza permessi fino all'amministratore con tutti i permessi. Questi permessi sono dettati dalla ridefinizione dei metodi virtuali che li richiedono e possono essere modificati quasi istantaneamente cambiando semplicemente il valore booleano ritornato in caso di future modifiche.

Anche questa gerarchia come la precedente possiede tutti i metodi necessari per essere esportata e importata su file xml.

VIEW PER AGGIUNTA E MODIFICA



L'ultima gerarchia si sviluppa sulla falsa riga di quella dei componenti, ma appartiene invece alla gui in quanto serve appunto a generare l'interfaccia per inserire e modificare i vari componenti nel catalogo.

Questa gerarchia si occupa in particolare di convertire valori inseriti tramite checkbox, spinbox e altri tipi di input in tipi comprensibili alle classi componente, dalle quali sia possibile ridisegnare poi nuovamente una finestra di modifica; inoltre si assicura che i valori inseriti siano validi.

CLASSI CONTENITORI

Le classi contenitore sono due: **listacomp** e **listautenti**.

listacomp contiene al suo interno un vettore di puntatori a componenti, il quale può contenere solo puntatori ad oggetti con nome diverso tra loro oppure accettarne qualsiasi quantità in base a come viene dichiarata la lista (cambiando il valore bool al richiamo del costruttore); questo permette di poterla utilizzare sia come catalogo, dove gli oggetti devono avere tutti nome diverso in quanto non ha senso che un catalogo mostri 2 volte lo stesso componente, sia come wishlist degli utenti, che può invece contenere più volte lo stesso componente (basti pensare che in un pc è spesso possibile trovare più ram o più hard disk, ma anche più schede video), il nome è quindi trattato come "chiave primaria" facendo un paragone coi database.

I metodi forniti soddisfano l'aggiunta, rimozione e ricerca di componenti (sempre sotto forma di puntatori, in quanto è presumibile che lo stesso oggetto appaia in più "listacomp"), l'ordinamento per nome e per prezzo, il calcolo del costo totale, l'esportazione e importazione da file xml e l'accesso agli iteratori `begin()` ed `end()` del vettore.

Nel caso della rimozione in particolare è possibile specificare se i puntatori vanno solamente rimossi dalla lista o se l'oggetto puntato vada distrutto (generalmente va distrutto definitivamente se viene rimosso dal catalogo e non solo da una wishlist di un utente).

listautenti contiene al suo interno un vettore di puntatori a utenti; assomiglia molto a **listacomp**, con la differenza che mentre **listacomp** utilizza i nomi dei componenti come "chiave primaria" solo in certe situazioni, **listautenti** invece usa sempre l'ID degli utenti come "chiave primaria", assicurandosi che non ci siano mai due utenti con lo stesso ID, pertanto quando il puntatore ad un utente viene rimosso dalla lista, viene sempre distrutto anche l'oggetto puntato (l'utente vero e proprio), cosa che non accade sempre in **listacomp**.

Anche in questo caso i metodi spaziano tra inserimento, rimozioni ricerca, ordinamento alfabetico, stampa in xml e accesso agli iteratori `begin()` e `end()` del vettore contenuto.

USO DI CODICE POLIMORFO

Tutte le gerarchie fanno uso di funzioni polimorfe e possiedono un distruttore virtuale in modo da richiamare sempre il distruttore corretto.

NELLA GERARCHIA UTENTI

I metodi polimorfi della gerarchia utenti sono bool *gestisceCatalogo()* e bool *gestisceUtenti()*, che sono essenzialmente i metodi che restituiscono informazioni sui permessi dell'utente.

Inoltre sono polimorfi i metodi *importadettagliXml()* e *etichettaXml()* che servono rispettivamente a riempire i campi nel modo corretto quando vengono letti dall'xml e a restituire il tag giusto da scrivere nel file xml al momento del salvataggio.

NELLA GERARCHIA COMPONENTI

I metodi polimorfi della gerarchia componenti sono la maggior parte di quelli per stampare il contenuto come stringa o xml, più precisamente sono *toString()*, *etichettaXml()*, *stampaContenutoXml()* e *importaDettagliXml()*

NELLA GERARCHIA DELLE VISTE

La gerarchia delle viste possiede due metodi virtuali: *aggiornaPezzo()* e *campiCorretti()*. Il primo si occupa di riempire i campi con i valori attuali quando si genera la finestra per creare o modificare un componente, il secondo si occupa di verificare per ogni tipo di oggetto se i campi inseriti sono corretti o meno, in base ai quali si deciderà se confermare o scartare le modifiche.

MANUALE UTENTE

L'applicazione salva e ricarica le informazioni sui file "catalogo.xml" e "registrati.xml" presenti nella stessa cartella della build, se all'avvio non dovesse trovarli verranno generati automaticamente e verrà creato automaticamente un utente con ID e Password "admin" con tutti i permessi. E' sconsigliato modificare questi file manualmente in quanto una scorretta formattazione può causare problemi all'applicazione.

(tra i file forniti alla consegna, nella cartella "XML" è presente un file "catalogo.xml" già riempito di una dozzina di componenti con dettagli realistici, notare comunque che una volta cambiato manualmente il file è consigliabile eliminare il file "registrati.xml" in quanto le wishlist degli utenti potrebbero essere state riempite di oggetti non presenti nel catalogo salvato su "catalogo.xml" durante altri test se sono stati effettuati prima della sostituzione del catalogo, causando problemi all'applicazione che è sviluppata sul presupposto che i file non vengano manomessi)

La prima finestra che si apre è la finestra di login, nella quale è possibile effettuare il login inserendo ID e password e cliccando il bottone "Login" oppure registrarsi compilando anche i campi Nome e Cognome e cliccando il bottone "Registrati".

Una volta effettuato login o registrazione si apre la finestra principale del programma, la quale presenta 2 liste: il catalogo completo (a sinistra) e la propria wishlist (a destra), gli elementi delle quali sono colorati diversamente in base al loro tipo; queste liste possono essere gestite con i bottoni presenti tra esse.

BOTTONI DISPONIBILI A TUTTI GLI UTENTI

 Bottone "Aggiungi alla Wishlist"

Aggiunge alla propria wishlist il componente selezionato dal catalogo.

 Bottone "Rimuovi dalla Wishlist"

Rimuove dalla propria wishlist il componente selezionato nella wishlist.

 Bottone "Svuota Wishlist"

Rimuove tutti i componenti dalla propria wishlist.

BOTTONI DISPONIBILI A CHI HA I PERMESSI DI GESTIONE CATALOGO

 Bottone "Modifica Componente"

Apri la finestra di modifica del componente selezionato dal catalogo.

 Bottone "Elimina Componente"

Elimina dal catalogo e da tutte le wishlist di tutti gli utenti il componente selezionato dal catalogo.

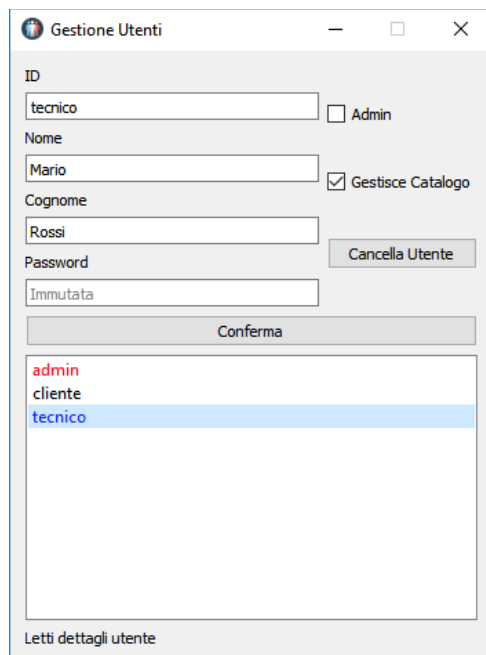
 Bottone "Aggiungi Componente"

Apri la finestra di creazione di un nuovo componente; per scegliere il tipo di componente occorre filtrare la lista per tipo tramite la combobox più in basso, in quanto il tipo di componente si basa su quel filtraggio, nel caso questo bottone venga premuto senza aver filtrato la lista la statusbar mostrerà un messaggio di errore che indica di filtrare la lista.

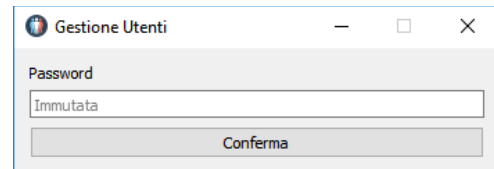
Oltre alle liste e ai bottoni, la finestra principale presenta in basso una barra di ricerca per il catalogo con la quale si possono cercare nomi specifici, una combobox per filtrare il catalogo per tipo di componente (la stessa usata dal pulsante "Aggiungi Componente") e dei radiobutton per decidere se ordinare il catalogo per nome o per prezzo.

IL BOTTONE GESTIONE UTENTI

La finestra aperta dal click questo bottone può avere due aspetti differenti in base ai permessi dell'utente loggato:



The screenshot shows a window titled "Gestione Utenti" with a standard Windows title bar. The window is divided into two main sections. The top section contains form fields for "ID" (with "tecnico" entered), "Nome" (with "Mario" entered), "Cognome" (with "Rossi" entered), and "Password" (with "Immutata" entered). To the right of the "ID" field is a checkbox labeled "Admin" which is unchecked. To the right of the "Password" field is a checkbox labeled "Gestisce Catalogo" which is checked. Below these fields are two buttons: "Cancella Utente" and "Conferma". The bottom section of the window is a list box containing three items: "admin" (in red), "cliente" (in black), and "tecnico" (in blue and highlighted with a blue selection bar). At the bottom left of the window, the text "Letti dettagli utente" is visible.



The screenshot shows a window titled "Gestione Utenti" with a standard Windows title bar. The window is divided into two main sections. The top section contains a "Password" label and a text field with "Immutata" entered. Below the text field is a "Conferma" button.

Nel caso in cui l'utente loggato sia un admin la finestra si presenta come un lista di utenti dove è possibile modificare qualsiasi dettaglio di ogni utente (meno che i propri permessi, in modo da evitare che il programma rimanga senza alcun admin), altrimenti la finestra offre semplicemente la possibilità di cambiare la propria password.

ORE IMPIEGATE

Progettazione: 5 ore

Progettazione GUI: 5 ore

Scrittura Model: 14 ore

Scrittura GUI: 38 ore

Testing e Debug: 6 ore

Il lungo tempo delle scritture di codice è dovuto in particolare al fatto di avere un grande quantità di campi da inserire in alcune classi (ad esempio nella classe `comp_mobo`), che ha di conseguenza allungato la scrittura della gui per modificare gli oggetti delle suddette classi.

AMBIENTE DI SVILUPPO

Sistema Operativo: Microsoft Windows 10 Pro N 64-bit 10.0.15063

Compilatore: g++ 4.9.2

QT: 5.9.1

IDE: Qt Creator 4.3.1

Qmake: 3.0