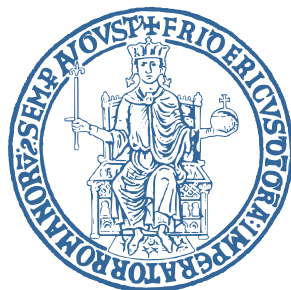


UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE



CORSO DI LAUREA IN INFORMATICA

INSEGNAMENTO DI OBJECT ORIENTATION

ANNO ACCADEMICO 2023/2024

SavingMoneyUnina

Autori:

Luigi CESARO

N86004727

luigi.cesaro@studenti.unina.it

Federico DE NICOLA

N86004507

federico.denicola@studenti.unina.it

Docente:

Prof. Sergio DI MARTINO

Indice

Descrizione del progetto.....	3
Filosofia di progetto.....	3
Struttura del progetto.....	4
Class Diagram.....	5
Entity.....	6
DAO.....	6
Controller.....	6
Database.....	7
GUI.....	7
Sequence Diagrams.....	8
Funzione di Login.....	8
Funzione di aggiunta transazione manuale.....	9

Descrizione del progetto

SavingMoneyUnina è un sistema completo per il monitoraggio delle finanze personali o familiari. Consente di gestire più carte di credito o debito, sia proprie che di altri membri della famiglia, tenendo traccia delle transazioni in entrata e in uscita.

Grazie alla sua funzionalità di suddivisione delle transazioni in diversi portafogli, gli utenti possono categorizzare le spese in modo efficace, ad esempio distinguendo tra svago, spese mediche, stipendio e bollette.

Il sistema offre la flessibilità di sincronizzare automaticamente le transazioni effettuate con una carta specifica con un gruppo predefinito oppure di inserire manualmente le transazioni. Questo strumento aiuta gli utenti a mantenere il controllo delle loro finanze, facilitando la gestione e l'analisi dei flussi di denaro in modo organizzato e intuitivo.

Link Repository Github:

<https://github.com/n00w4/SavingMoneyUNINA>

Filosofia di progetto

L'utente (*User*) può possedere uno o più conti bancari (*BankAccount*), al quale potranno essere associate da zero o più carte (*Card*). Le carte potranno essere fondamentalmente di debito (*DebitCard*) o di credito (*CreditCard*).

Le transazioni (*Transaction*) possono essere effettuate da una carta e l'utente potrà effettuare zero o più transazioni che potranno essere raggruppate in appositi portafogli (*Portfolio*).

Il portafoglio può contenere più transazioni e saranno categorizzati attraverso le *Category*.

Una transazione si divide a sua volta in transazione d'uscita (*Expense*) e transazione d'entrata (*Income*).

Più utenti possono far parte di una famiglia (*Family*).

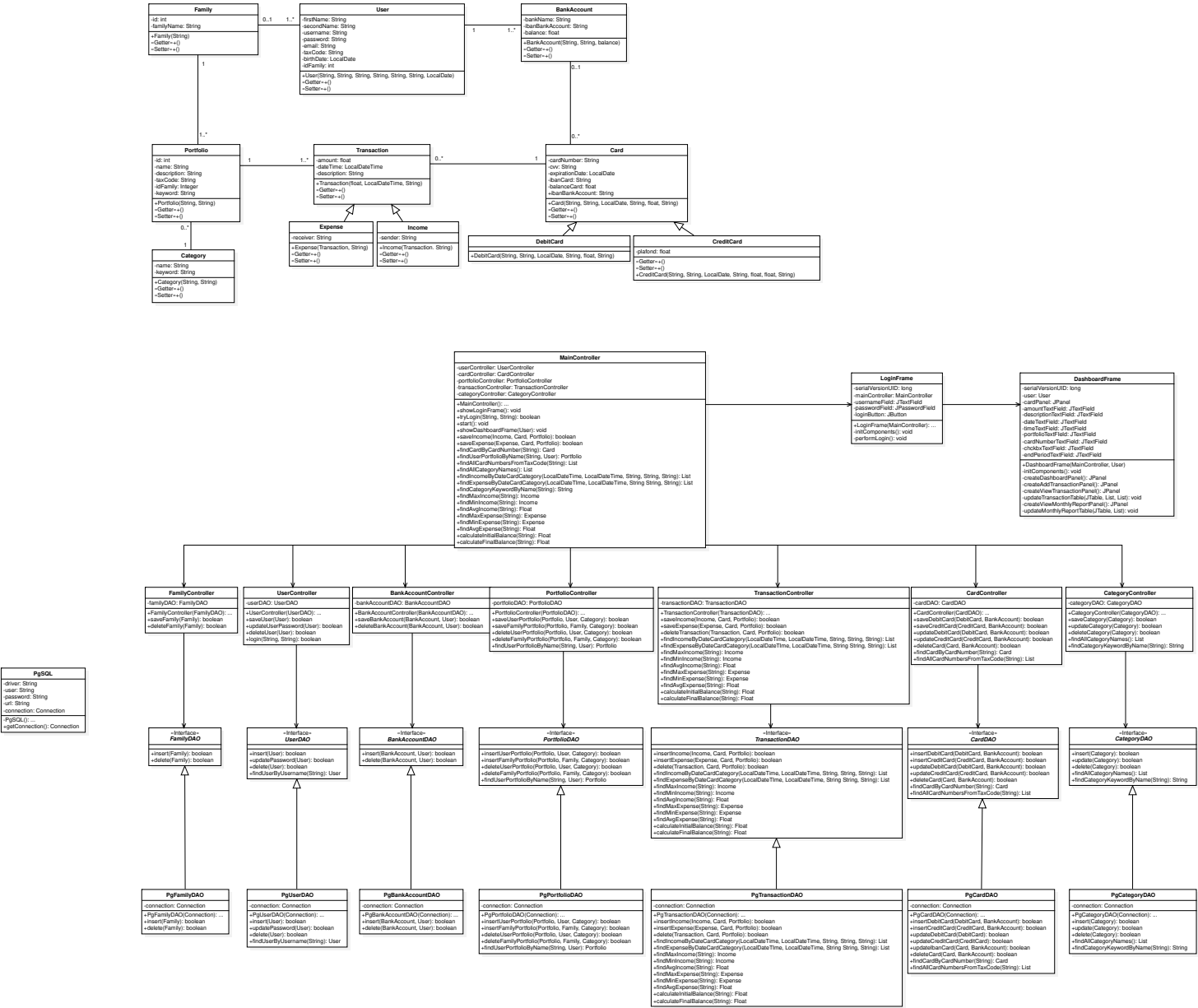
Struttura del progetto

Il progetto è stato ideato secondo la filosofia **EBC (Entity-Boundary-Controller)** ed utilizza puramente la libreria **Java Swing** per la realizzazione della GUI.

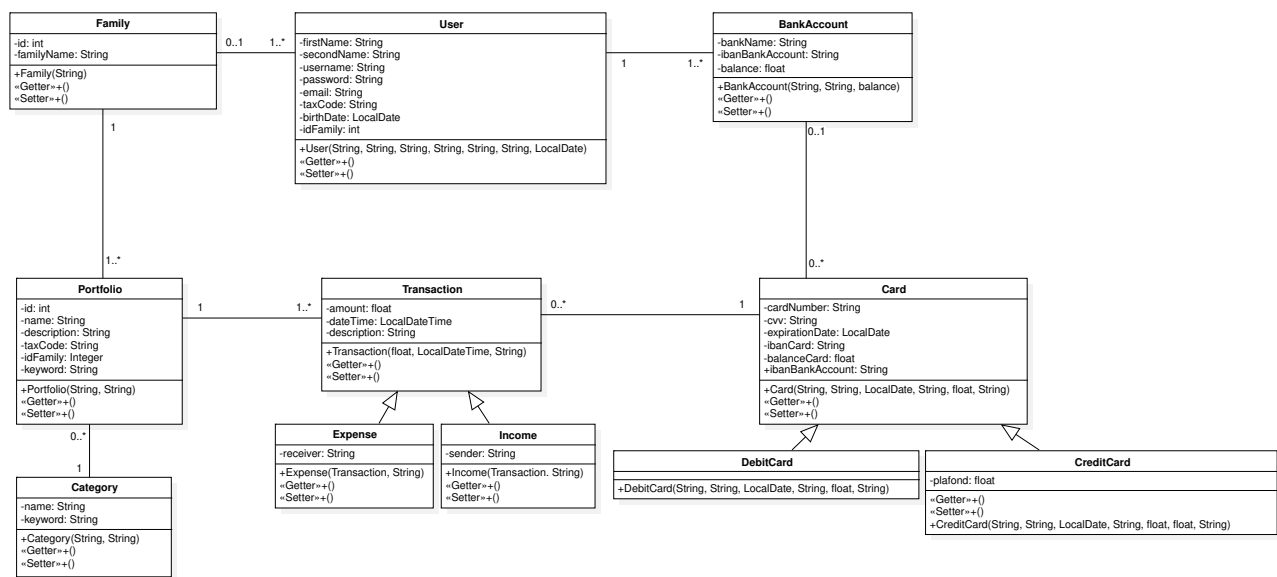
Ecco com'è stato strutturato:

- **Controller:** qui sono contenute tutte le **classi di controllo**, ovvero i controller per ogni entità ed il MainController che orchestra dietro le quinte tutte le operazioni di SavingMoneyUNINA;
- **DAO:** contiene le **interfacce DAO** con i metodi per richiedere e manipolare i dati necessari all'app, utili per future implementazioni con altri DBMS ed anche per dividere le implementazioni sottostanti dai controller;
 - **PgSQLImp:** contiene tutte le implementazioni delle interfacce DAO per lavorare con PostgreSQL;
- **Database:** qui sono contenuti tutti i dettagli che implementano le **connessioni** con PostgreSQL e potrebbe anche essere utilizzato per future implementazioni con altri DBMS.
- **Entity:** qui sono contenute le **classi di entità** che rappresentano il mondo dentro cui opera SavingMoneyUNINA, come l'utente o una transazione;
- **GUI:** contiene tutti i **frame** necessari alla grafica dell'app;
- **App.java:** file main che si occuperà di **avviare** SavingMoneyUNINA.

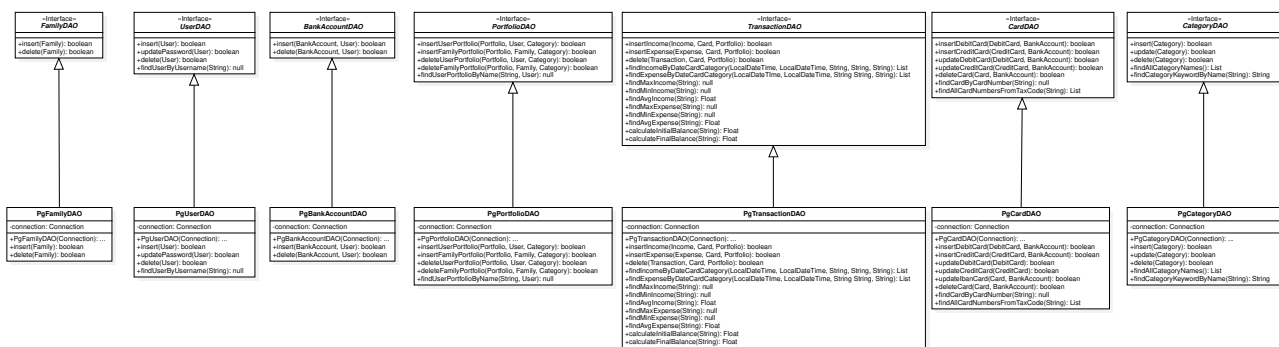
Class Diagram



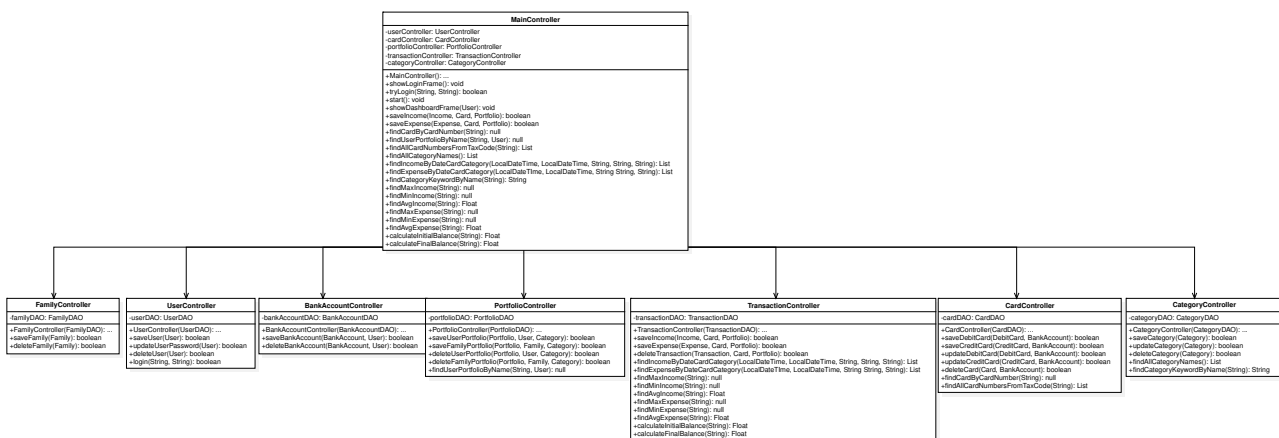
Entity



DAO



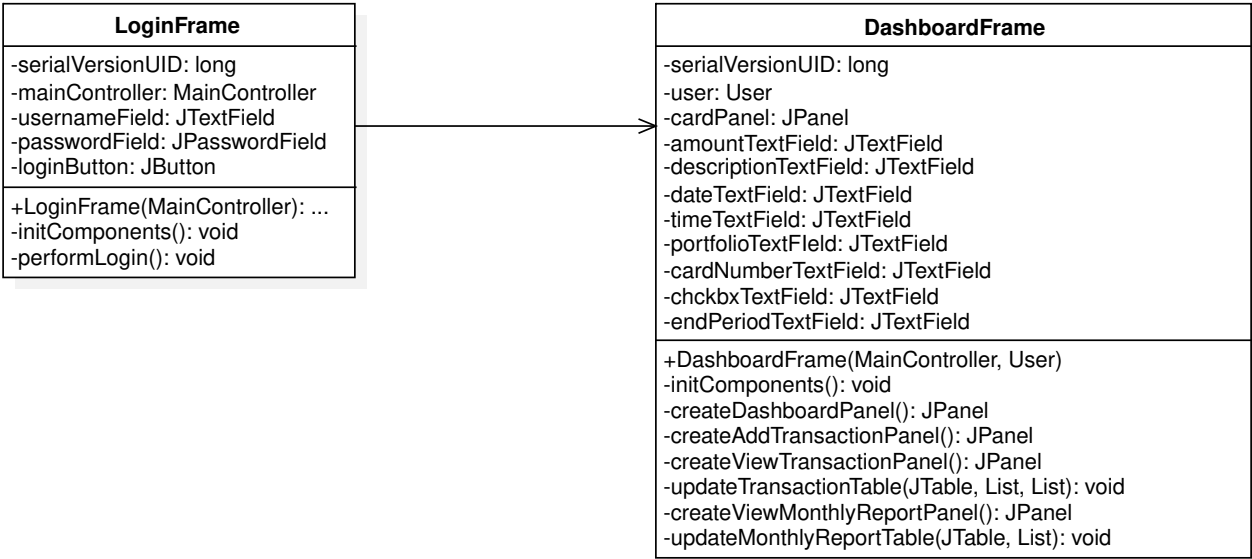
Controller



Database

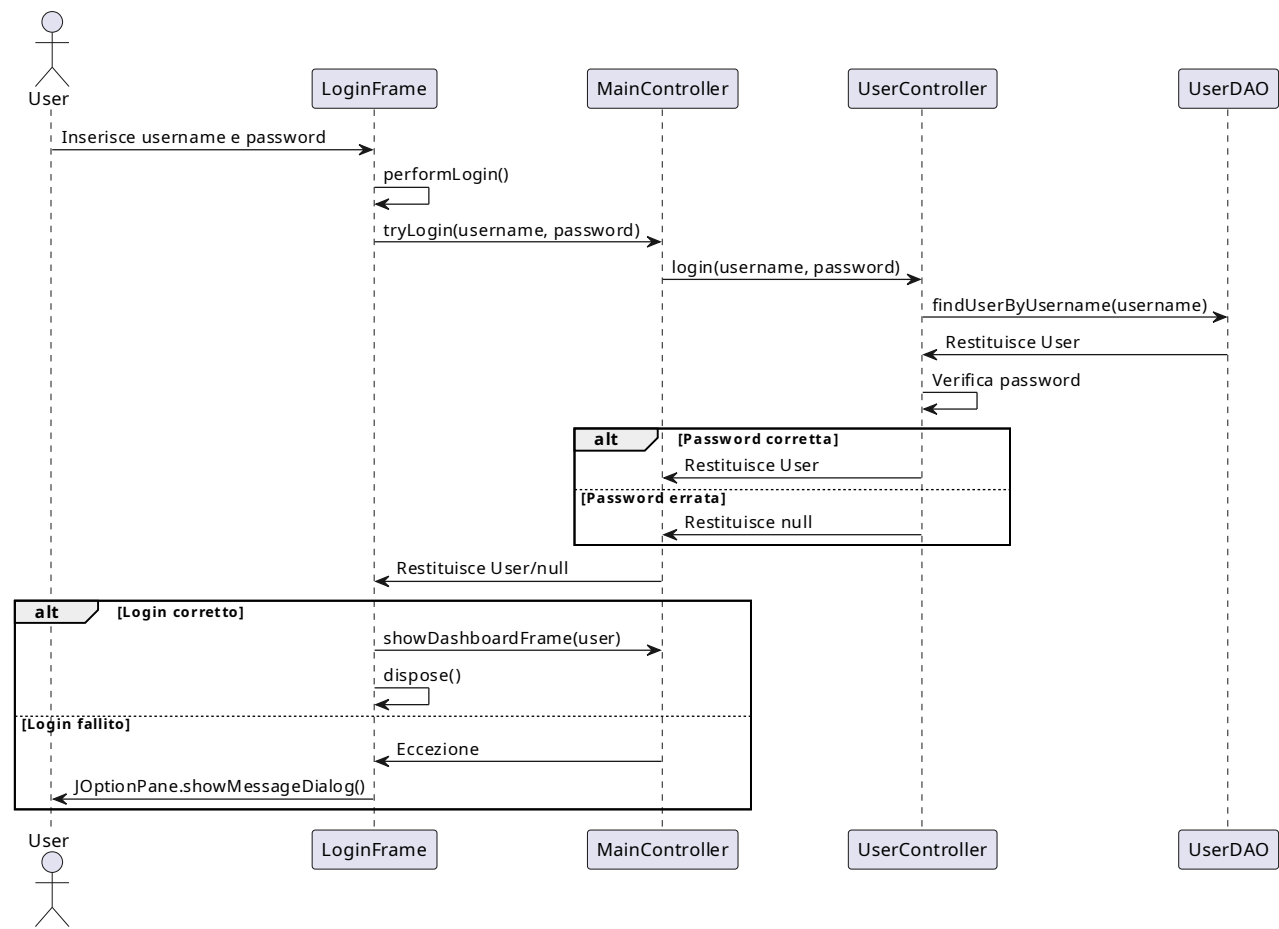
PgSQL
-driver: String -user: String -password: String -url: String -connection: Connection
-PgSQL(): ... +getConnection(): Connection

GUI



Sequence Diagrams

Funzione di Login



Funzione di aggiunta transazione manuale

