

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

Progettazione e sviluppo di un applicativo java
per la realizzazione di un sistema di gestione finanziaria



Autori:

Antonio Bacio Terracino

matricola N86004952

`an.bacioterracino@studenti.unina.it`

Roberto Amendola

matricola N86004815

`robert.amendola@studenti.unina.it`

Docenti:

Prof. Sergio Di Martino

Indice

1	Analisi Della Traccia	3
1.1	Descrizione Del Dominio	3
1.2	Interpretazione Del Dominio e Classi Individuate	3
1.2.1	User	3
1.2.2	PaymentCard	3
1.2.3	Transaction	3
1.2.4	Category	4
1.2.5	Report	4
1.2.6	Family	4
1.3	Classi Ausiliarie	4
1.3.1	TransactionFilter	4
1.3.2	CurrencyConverter	4
1.3.3	MonthlyBalance	4
1.3.4	IncomeDetails / ExpenseDetails	5
2	Class Diagrams	6
2.1	Model	7
2.2	DAO	8
2.3	View	9
2.3.1	UiUtil	9
2.4	Controller	11
3	Sequence Diagrams	12
3.1	Execute Transaction	12
3.2	Login Attempt	13
4	GitHub Repository	14
4.1	Link	14

1 Analisi Della Traccia

1.1 Descrizione Del Dominio

SavingMoneyUnina è un sistema che permette di tenere sotto controllo le finanze personali o familiari. Permette di collegare più carte di pagamento gestendo le transazioni in entrata ed in uscita. Il sistema permette di suddividere le transazioni in gruppi (portafogli) appartenenti a diverse categorie (es. svago, spese mediche, stipendio, bollette ecc.). L'utente avrà la possibilità di guardare le sue transazioni e filtrarle secondo:

- 1) data iniziale
- 2) data finale
- 3) direzione
- 4) categoria

È inoltre possibile effettuare nuove transazioni scegliendo la desiderata valuta fra euro e dollari , con un rateo di conversione EUR -> USD pari a 1,06.

1.2 Interpretazione Del Dominio e Classi Individuate

1.2.1 User

È un profilo creato a seguito della registrazione di un cliente, a cui si può accedere utilizzando username e password collegati.

1.2.2 PaymentCard

Rappresenta una carta di pagamento posseduta dall'utente. Con essa si può visualizzare uno storico delle transazioni ed eseguirne di nuove. Sarà inoltre possibile visualizzare report mensile relativo alla carta.

1.2.3 Transaction

Una transazione rappresenta un movimento di denaro che interessa un utente. Le informazioni archiviate relative a ciascuna transazione includono l'importo, la causale, la data e il metodo di pagamento. Le transazioni effettuate saranno automaticamente smistate nelle relative categorie. Le transazioni effettuate in dollari saranno convertite in euro secondo il rateo di conversione di 1,06.

1.2.4 Category

Quando una transazione viene registrata, la causale è confrontata con le parole chiave associate alle categorie dell'utente. Se una o più parole chiave vengono trovate nella causale, la transazione sarà inserita nella categoria. Se una transazione non risulta collegata a nessuna categoria, il sistema la inserisce nella categoria "other".

1.2.5 Report

Consiste in un recap mensile di una precisa carta di un utente, il report contiene:

- 1) entrata massima,minima e media
- 2) spesa massima, minima e media
- 3) saldo iniziale e finale della carta.

1.2.6 Family

Consiste in un gruppo di utenti. i componenti di una famiglia hanno la possibilità di accedere ad un report mensile della intera famiglia contenente:

- 1) saldo iniziale e finale della famiglia
- 2) spesa totale della famiglia
- 3) guadagno totale della famiglia.

Le informazioni sopra descritte sono presenti anche relative ai singoli componenti.

1.3 Classi Ausiliarie

Per l'implementazione delle classi di dominio, sono state introdotte una serie di classi interne al fine di migliorare l'organizzazione e la manutenibilità del codice. Nel paragrafo seguente verrà fornita una descrizione dettagliata del loro funzionamento e del ruolo che svolgono all'interno del sistema.

1.3.1 TransactionFilter

Partendo una lista di transazioni relative ad una carta ritorna quelle che rispettano i filtri(elencati in precedenza) passati in input.

1.3.2 CurrencyConverter

Si occupa della conversione delle transazioni effettuate in dollari statunitensi (USD) in euro (EUR), applicando un tasso di cambio fisso pari a 1,06.

1.3.3 MonthlyBalance

Se le informazioni relative a spese e guadagni mensili di una carta, possono essere ottenute tramite query SQL, il bilancio iniziale e finale devono necessariamente essere salvate per ottenere i Report mensili.

1.3.4 IncomeDetails / ExpenseDetails

Per questioni di miglior efficienza del codice le singole informazioni del report sono state divise nelle tre classi interne MonthlyBalance e due classi parallele income ed expense Details dediche alla gestione delle rispettive entrate/uscite massime minime e medie.

2 Class Diagrams

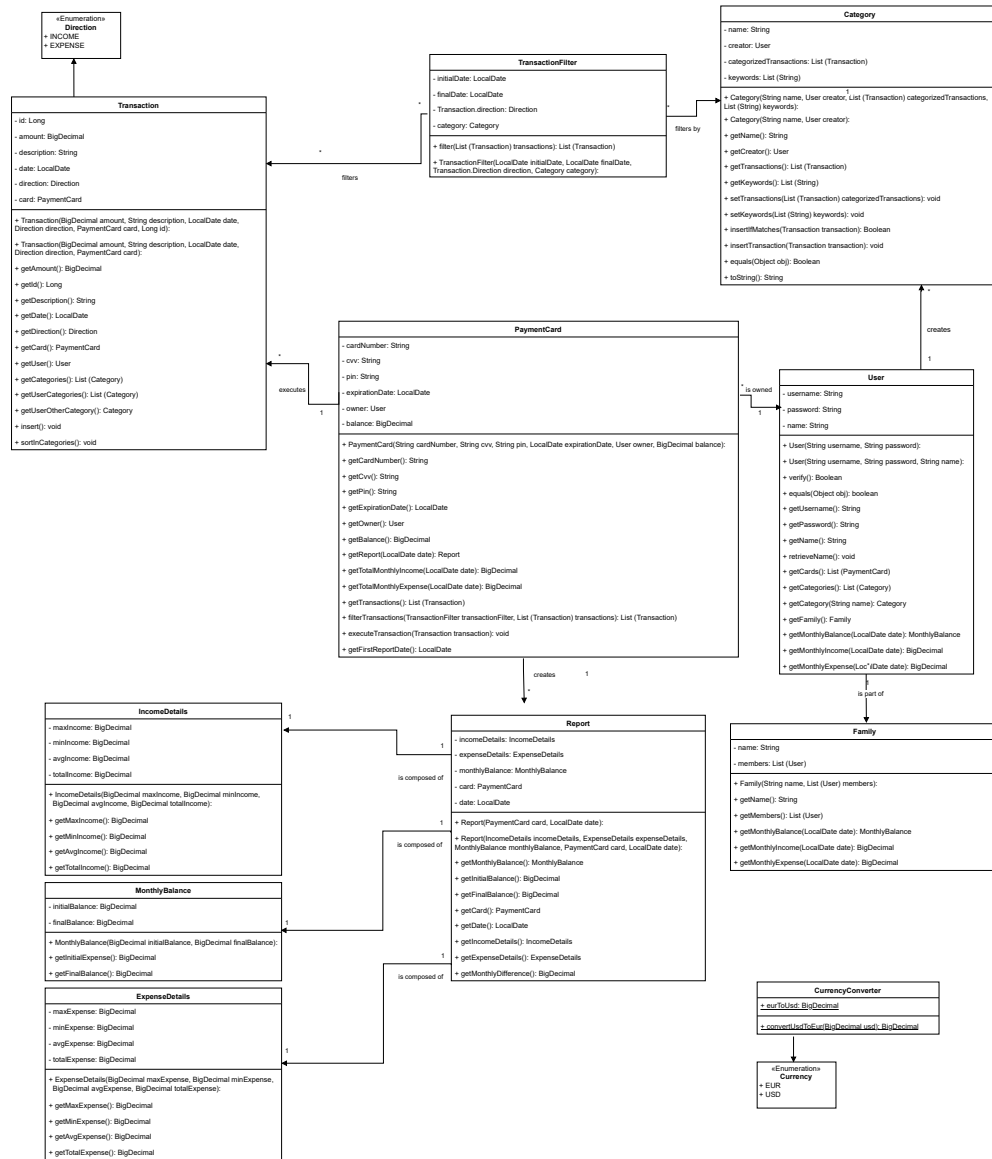
Per migliorarne la leggibilità, il diagramma è stato suddiviso nei seguenti blocchi logici:

- 1) Model
- 2) DAO
- 3) View
- 4) Controller.

Le omissioni derivanti da questa suddivisione sono riassunte nei seguenti criteri:

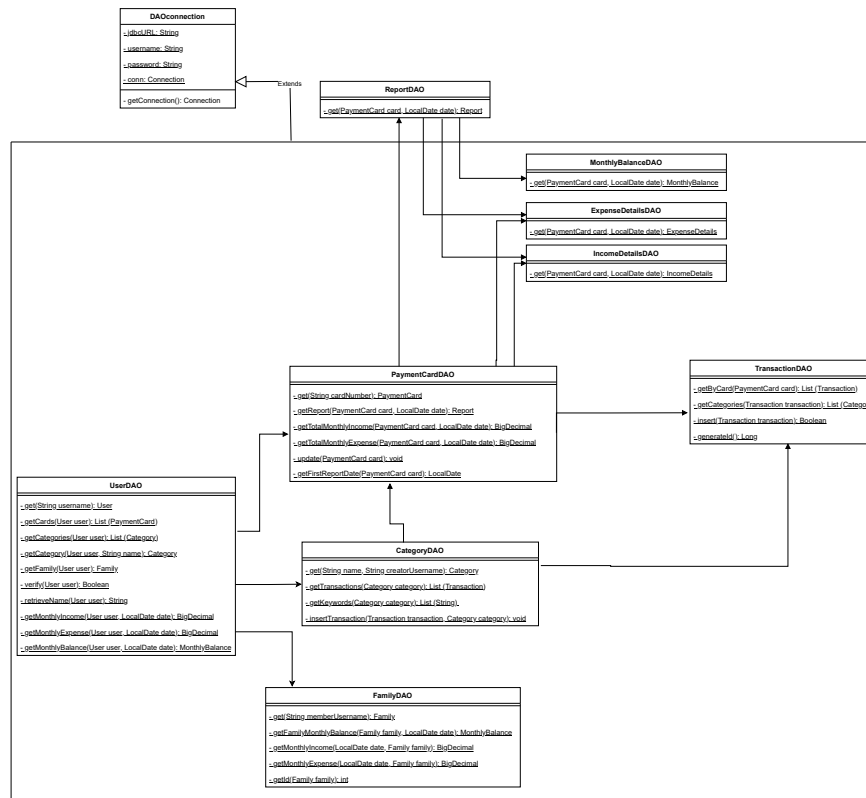
- 1) **Ogni classe Model interagisce con le classi DAO esclusivamente attraverso la rispettiva classe DAO.**
- 2) **Ogni classe View interagisce con le classi Controller esclusivamente attraverso la rispettiva classe Controller.**

2.1 Model



2.2 DAO

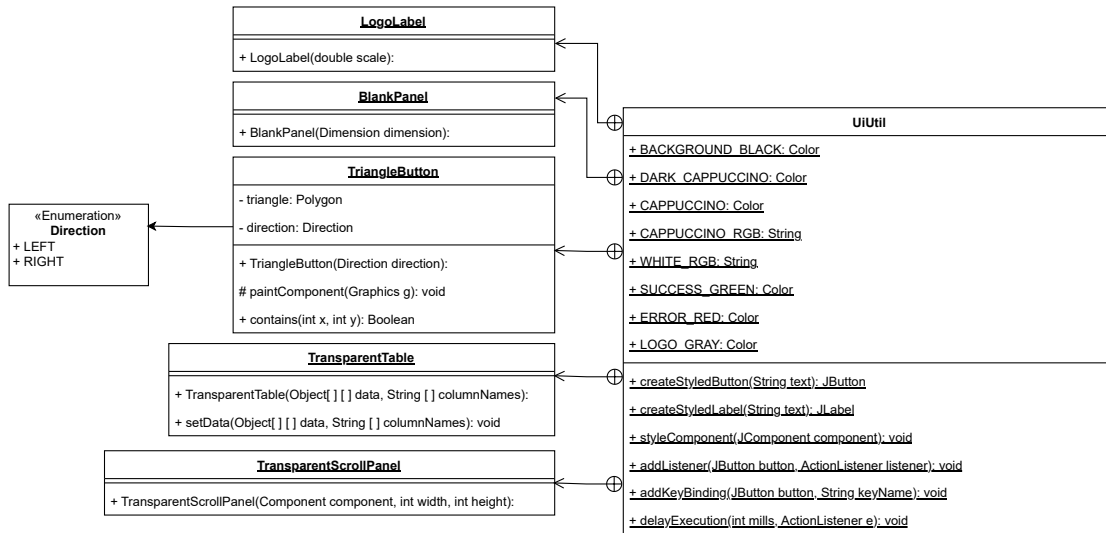
Nota: Tutte le classi DAO, tranne ReportDAO, estendono DAOconnection. Per semplicità è stato inserito un singolo collegamento "extends" per tutte le classi.



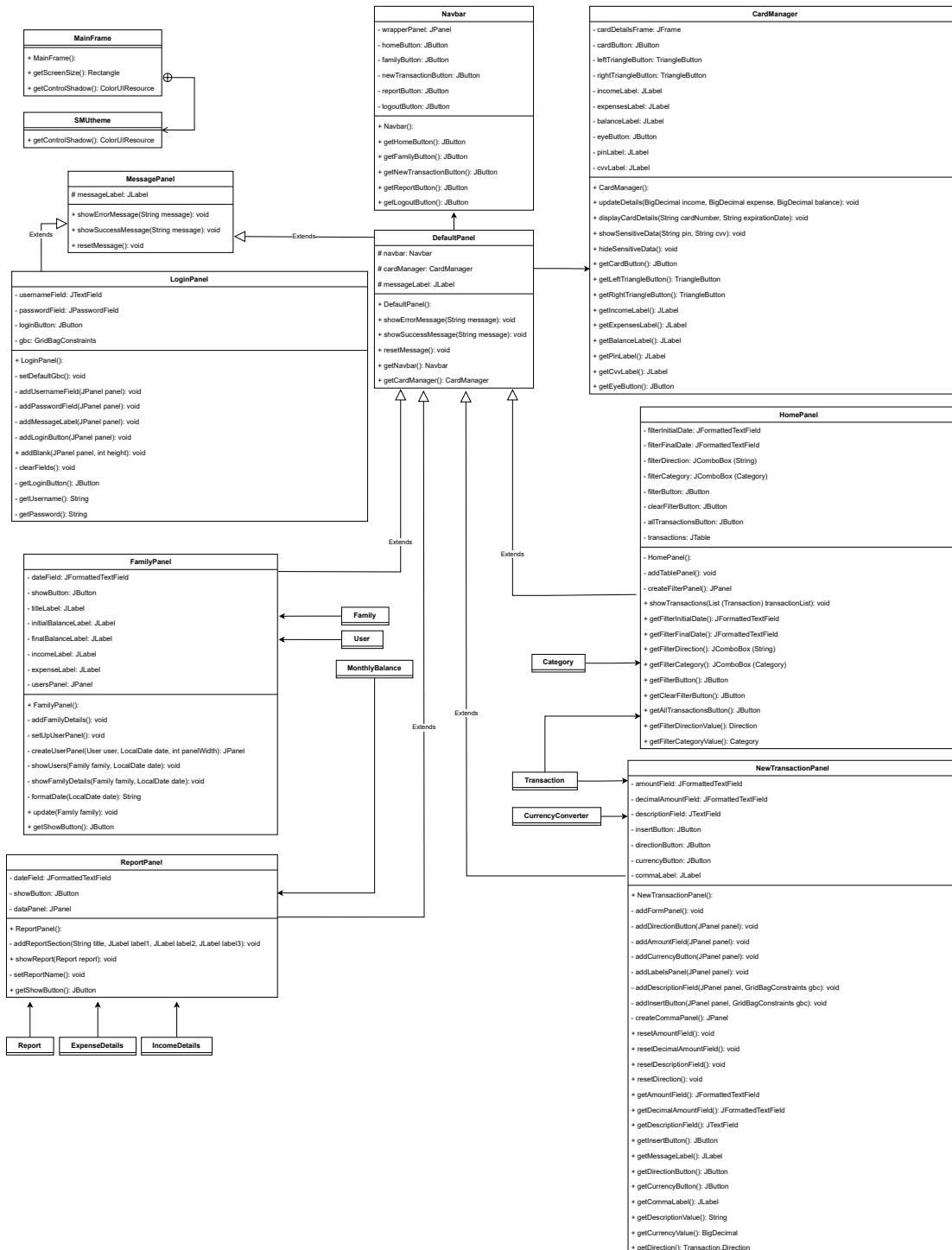
2.3 View

2.3.1 UiUtil

Nota: La seguente classe UiUtil, importata da tutte le altre classi View, è stata riportata separatamente per semplicità.

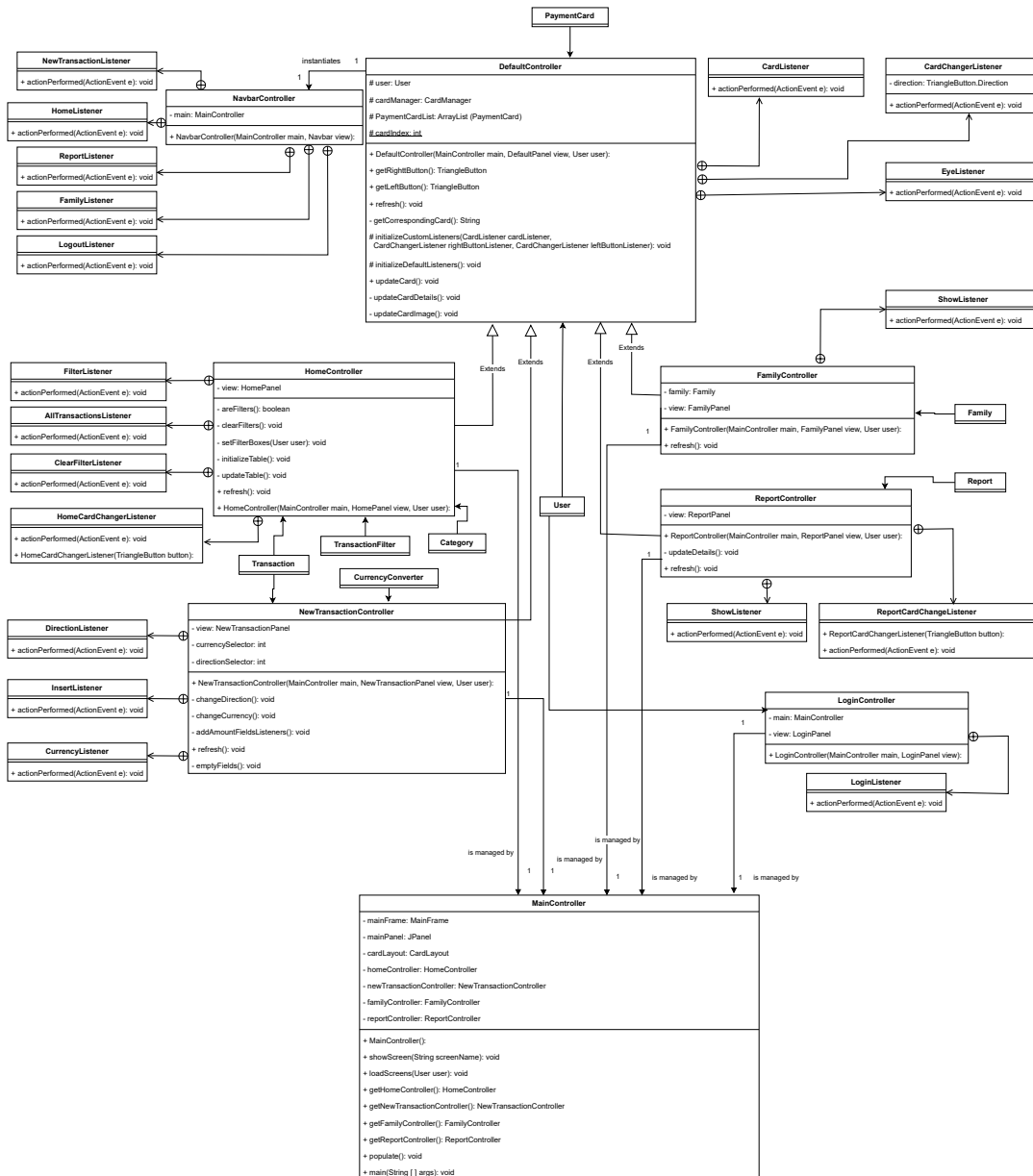


Nota: Le classi rappresentate esclusivamente dal loro nome sono classi Model, che mantengono attributi, metodi e collegamenti rappresentati nel grafico Model.



2.4 Controller

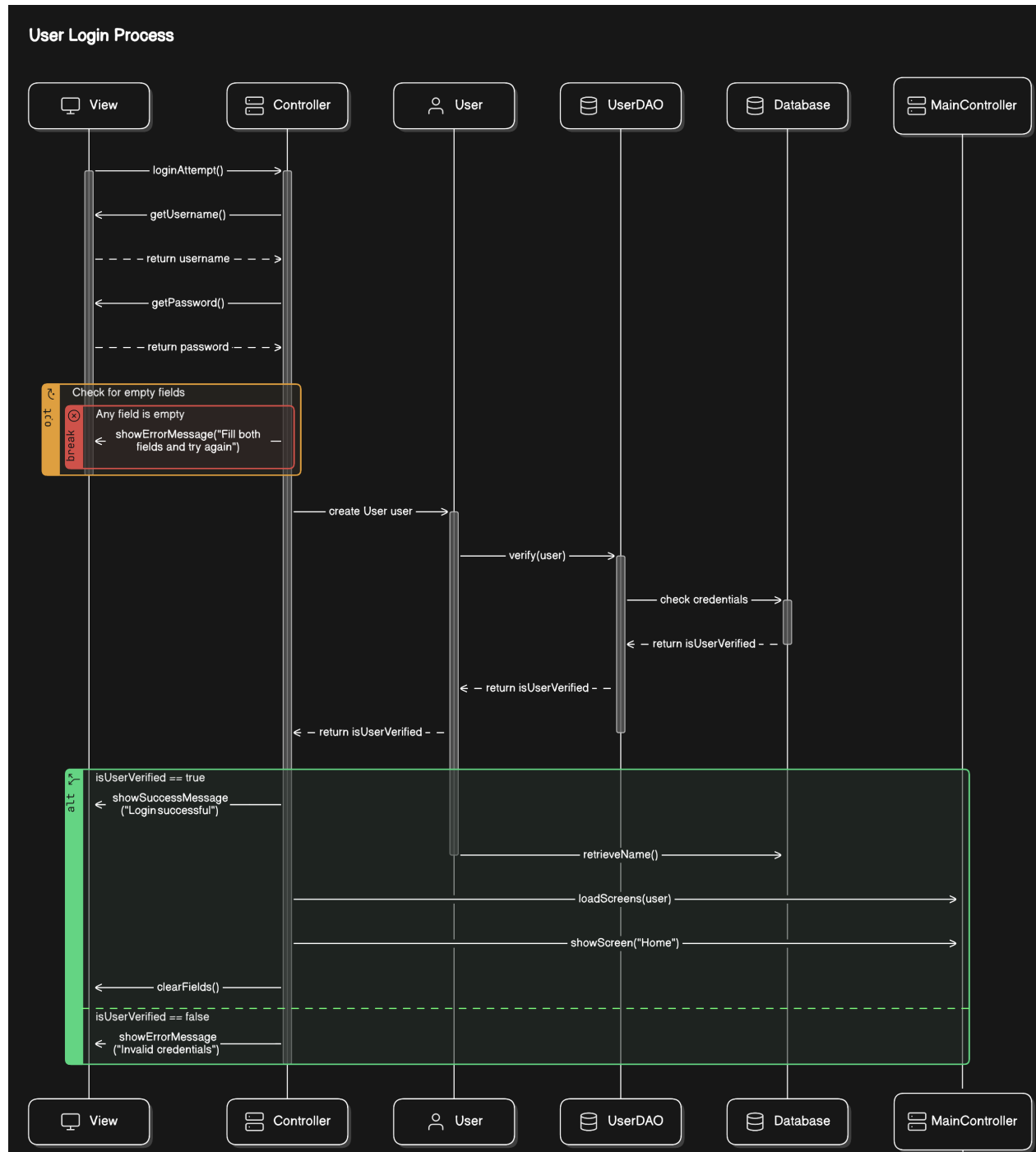
Nota: Le classi rappresentate esclusivamente dal loro nome sono classi Model, che mantengono attributi, metodi e collegamenti rappresentati nel grafico Model.



3.1 Execute Transaction



3.2 Login Attempt



4 GitHub Repository

4.1 Link

Link: <https://github.com/RobIsNotAvailable/SavingMoneyUnina>