

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 TRIENNALE IN INFORMATICA

Progettazione ed implementazione di un applicativo in Java per la gestione delle finanze personali

Docente

Prof. Sergio Di Martino

Autori

GIANCARLO BRANDI
N86004541

SALVATORE CARLEO
N86004897

Anno accademico 2023/2024

Indice

Indice	2
1. Introduzione	3
1.1 Traccia	3
1.2 SavingMoneyUnina: un applicativo per la gestione delle finanze personali	3
2. Progettazione	4
2.1 Architettura.....	4
2.2 CRC Cards.....	4
2.2.1 Entities	4
2.2.2 Entities.DAO	5
2.2.3 Controller	7
2.3 Class diagram	8
2.3.1 Controller	9
2.3.2 Classi DAO	9
3. Sequence Diagram e Mockup.....	10
3.1 Sequence Diagram.....	10
3.2 Mockup e navigazione delle interfacce	12

1. Introduzione

1.1 Traccia

SavingMoneyUnina è un sistema che permette di tenere sotto controllo le finanze personali o familiari. Permette di collegare più carte di credito o debito, proprie o di un altro membro della famiglia 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.). È possibile sincronizzare automaticamente le transazioni effettuate da una carta assegnandole ad uno specifico gruppo, oppure registrare una transazione manualmente. Si utilizzino le proprie conoscenze del dominio per definire dettagli non specificati nella traccia.

1.2 SavingMoneyUnina: un applicativo per la gestione delle finanze personali

SavingMoneyUnina è un applicativo per la gestione delle finanze personali, sviluppato con la libreria Swing di Java per la gestione delle interfacce e basato su PostgreSQL. Esso permette di gestire i propri conti corrente o quelli di altre persone con le relative carte di pagamento associate, visualizzare report mensili sulle spese, creare dei gruppi di spesa (“Portafogli”) e in generale di tenere conto delle proprie spese.

2. Progettazione

2.1 Architettura

Per lo sviluppo di SavingMoneyUnina è stato utilizzato il modello architetturale EBC (Entity-Boundary-Control). La scelta si riflette sul codice Java finale dell'applicativo: le classi sono state suddivise in package in base alle loro funzioni e responsabilità interne.

I package sono:

1. Entities
 - a. Dao
2. Boundaries
3. Control

2.2 CRC Cards

2.2.1 Entities

Nome della classe	Persona
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Persona fisica
Collaborazioni	Utente

Nome della classe	Utente
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Utente registrato all'applicativo, associato ad una Persona fisica
Collaborazioni	Persona

Nome della classe	ContoCorrente
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Conto Corrente gestito da un utente
Collaborazioni	Utente Carta

Nome della classe	Carta
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Carta di pagamento associata ad un Conto Corrente
Collaborazioni	ContoCorrente

Nome della classe	Transazione
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Transazione relativa ad un Conto Corrente
Collaborazioni	ContoCorrente Portafogli

Nome della classe	Portafogli
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Gruppo di transazioni
Collaborazioni	Persona Transazione

Nome della classe	Categoria
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Categoria di un Portafogli
Collaborazioni	Portafogli

2.2.2 Entities.DAO

Nome della classe	CartaDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative alle Carte
Collaborazioni	Carta UtenteDao ConnectionDatabase

Nome della classe	CategoriaDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative alle Categorie
Collaborazioni	Categoria ConnectionDatabase

Nome della classe	ContoCorrenteDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative ai Conti corrente
Collaborazioni	ContoCorrente Carta Transazione ConnectionDatabase

Nome della classe	PersonaDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative alle Persone
Collaborazioni	Persona Utente ConnectionDatabase

Nome della classe	PortafogliDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative ai Portafogli
Collaborazioni	Portafogli Transazione Categoria ConnectionDatabase

Nome della classe	TransazioneDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative alle Transazioni
Collaborazioni	Transazione ContoCorrente Portafogli ConnectionDatabase

Nome della classe	UtenteDao
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe DAO per le operazioni sul database relative alle Carte
Collaborazioni	Utente Persona ContoCorrente Carta Portafogli ConnectionDatabase

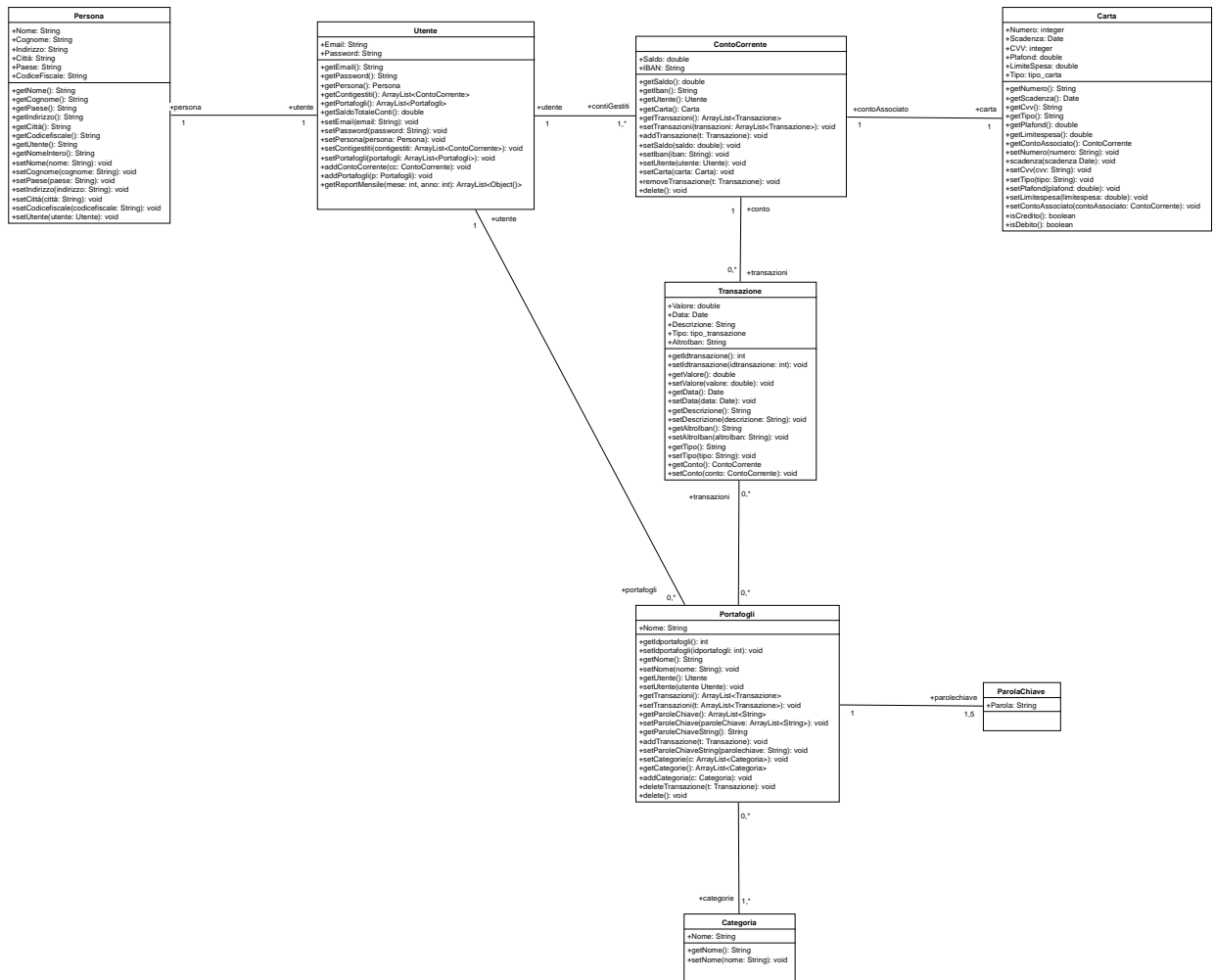
2.2.3 Controller

Nome della classe	NavigationController
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe controller per la gestione della navigazione tra finestre e per la visualizzazione degli avvisi
Collaborazioni	Utente ContoCorrente Portafogli

Nome della classe	LoginController
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe controller per la gestione della funzione di log-in
Collaborazioni	Utente

Nome della classe	DataController
Superclassi	Nessuna
Sottoclassi	Nessuna
Responsabilità	Classe controller per la gestione e la conversione dei dati
Collaborazioni	Categoria

2.3 Class diagram



2.3.1 Controller

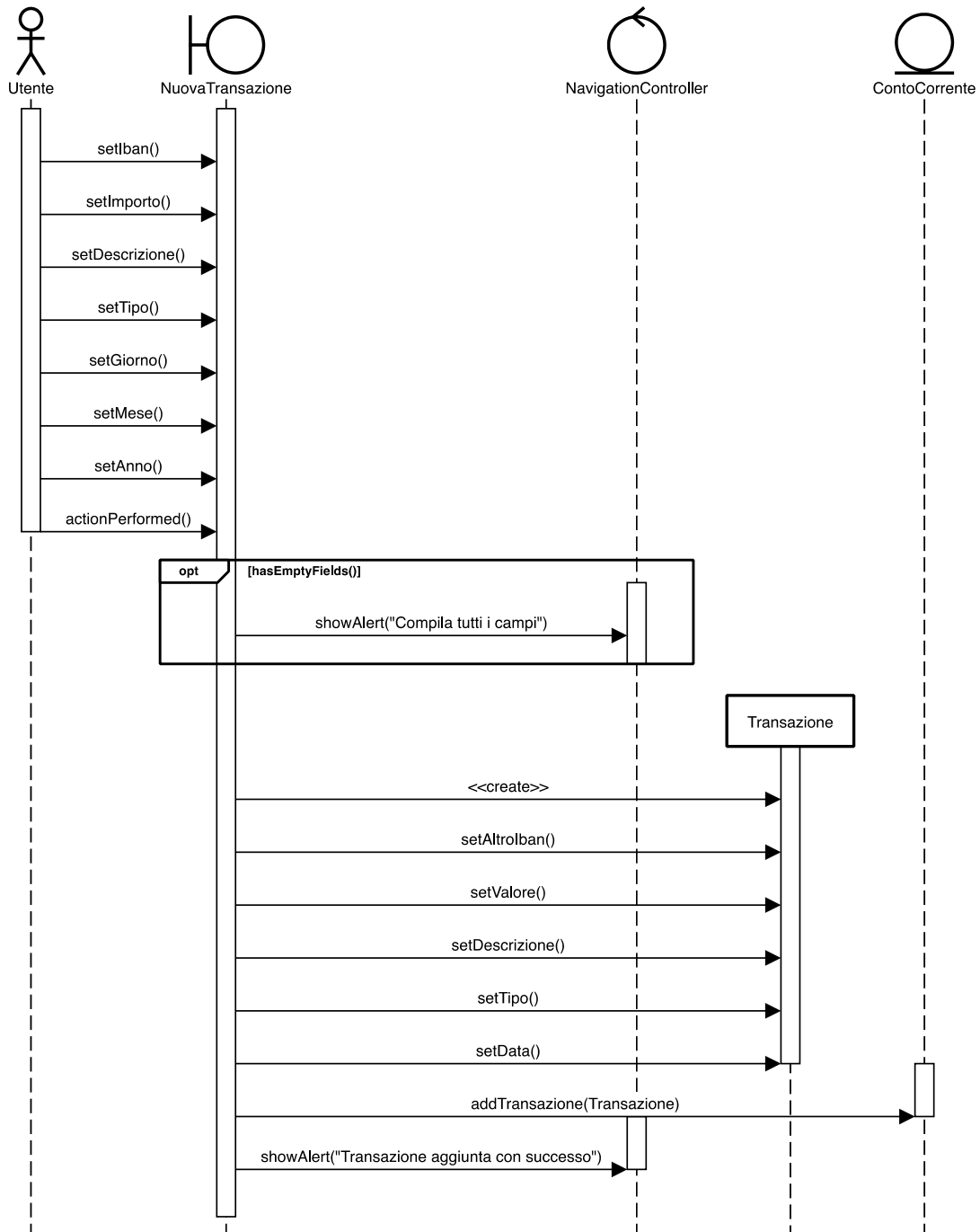
NavigationController	LoginController
+showHomePage(u: Utente): void +showNuovoConto(caller: JFrame, u: Utente): void +showInformazioniConto(caller: JFrame, cc: ContoCorrente): void +showNuovaTransazione(caller: JFrame, cc: ContoCorrente): void +showInformazioniPortafogli(u: Utente, p: Portafogli): void +showSceltaManuale(caller: JFrame, u: Utente, p: Portafogli): void +showNuovoPortafogli(caller: JFrame, u: Utente): void +showRicercaAvanzata(u: Utente): void +showReportMensile(u: Utente): void +showAlert(text: String): void	+checkLoginCredentials(email: String, password: String): Utente
	DataController
	+getCategorie(): ArrayList<Categoria> +getDate(anno: int, mese: int, giorno: int): Date +dateToString(date: Date): String +formatMoney(money: double): String

2.3.2 Classi DAO

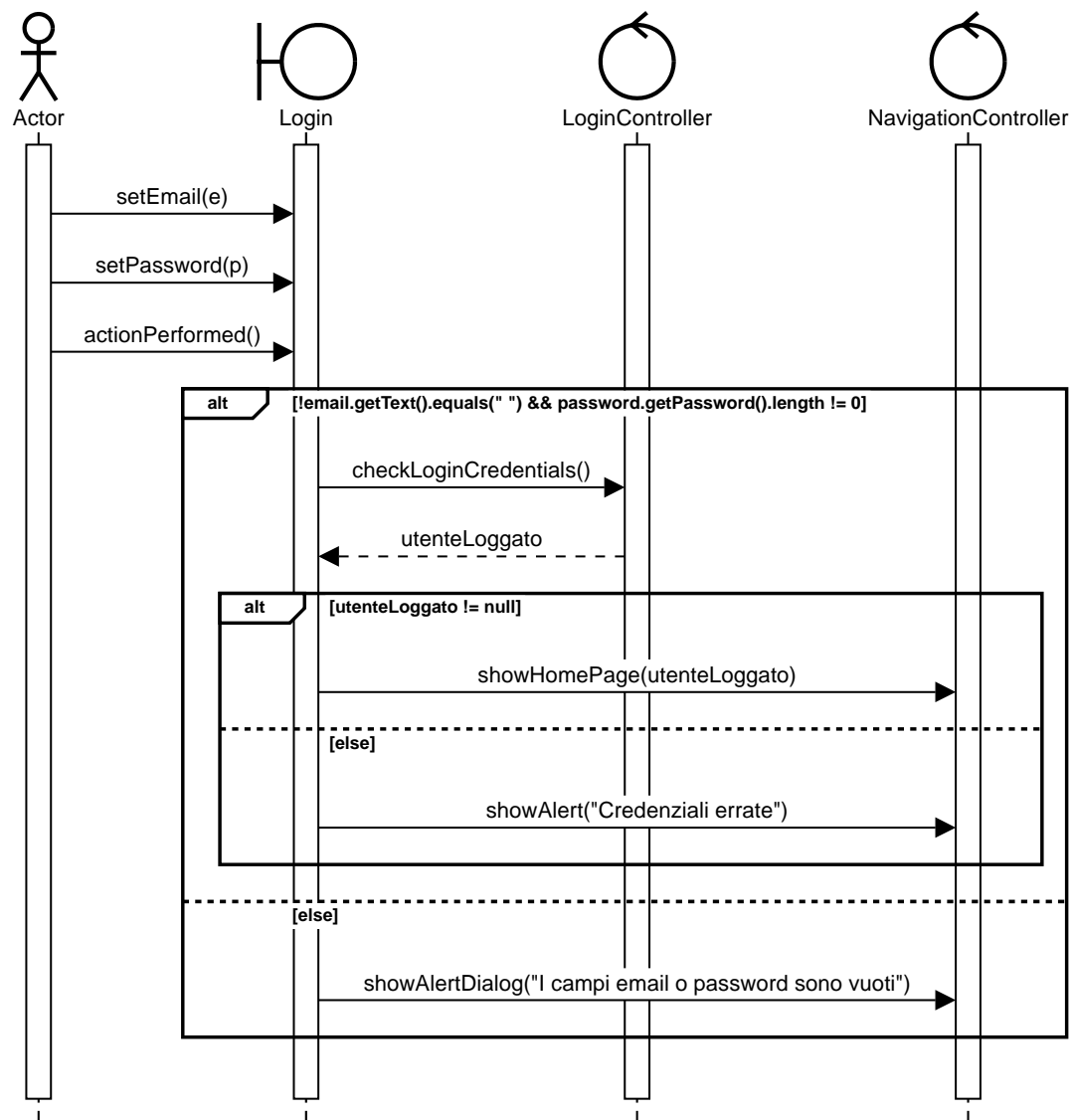
CartaDao	CategoriaDao	UtenteDao
+getContoCorrente(carta: Carta): ContoCorrente	+getCategoria(): ArrayList<Categoria>	+getContiGestiti(utente: Utente): ArrayList<ContoCorrente> +getUtenteByEmail(email: String): Utente +getPersona(utente: Utente): Persona +getPortafogli(utente: Utente): ArrayList<Portafogli> +getSaldoTotaleConti(utente: Utente): double +inserisciNuovoConto(u: Utente, cc: ContoCorrente, c: Carta): void +inserisciNuovoPortafogli(u: Utente, p: Portafogli): int +getReportMensile(u: Utente, mese: int, anno: int): ArrayList<Object>
ContoCorrente	PersonaDao	
+getCartaAssociata(cc: ContoCorrente): Carta +getTransazione(cc: ContoCorrente): ArrayList<Transazione> +aggiungiTransazione(cc: ContoCorrente, t: Transazione): void +getConto(cc: ContoCorrente): ContoCorrente +getSaldo(cc: ContoCorrente): double +deleteTransazione(cc: ContoCorrente, t: Transazione): void +deleteConto(cc: ContoCorrente): void	+getUtente(persona: Persona): Utente	
PortafogliDao	TransazioneDao	
+getTransazioni(portafogli: Portafogli): ArrayList<Transazione> +getParoleChiave(portafogli: Portafogli): ArrayList<String> +getParoleChiaveString(portafogli: Portafogli): String +addTransazione(portafogli: Portafogli, transazione: Transazione): void +addParoleChiaveMultiple(p: Portafogli, parolechiave: String): void +addCategoria(p: Portafogli, c: Categoria): void +getCategoria(p: Portafogli): ArrayList<Categoria> +deleteTransazione(p: Portafogli, t: Transazione): void +deletePortafogli(p: Portafogli): void	+cercaTransazione(from: Date, to: Date): ArrayList<Transazione> +cercaTransazione(cc: ContoCorrente, from: Date, to: Date): ArrayList<Transazione> +cercaTransazione(p: Portafogli, from: Date, to: Date): ArrayList<Transazione> +cercaTransazione(cc: ContoCorrente, p: Portafogli, from: Date, to: Date): ArrayList<Transazione>	

3. Sequence Diagram e Mockup

3.1 Sequence Diagram

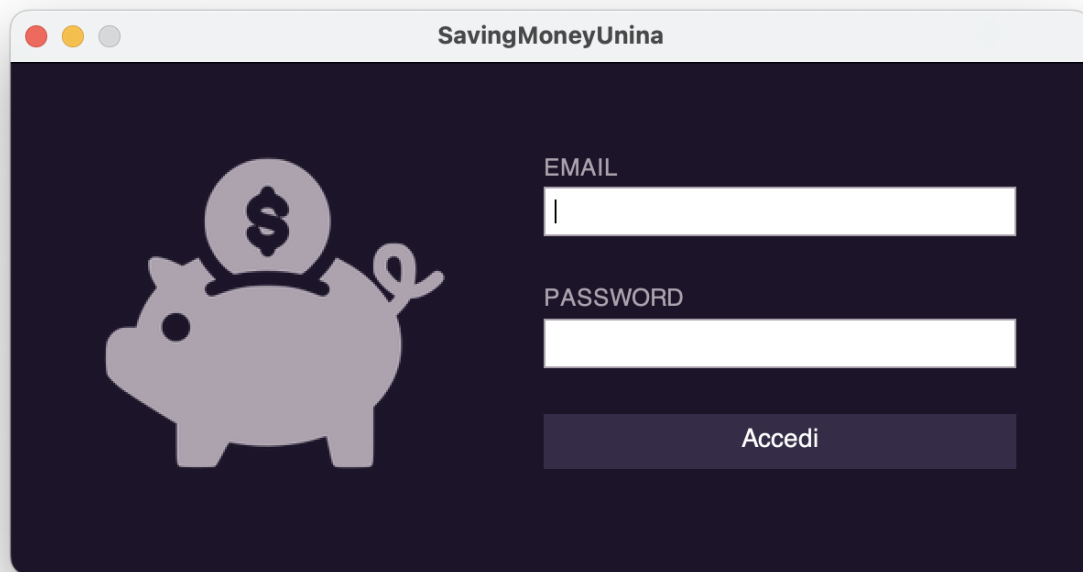


3.1.1: Inserimento di una Transazione

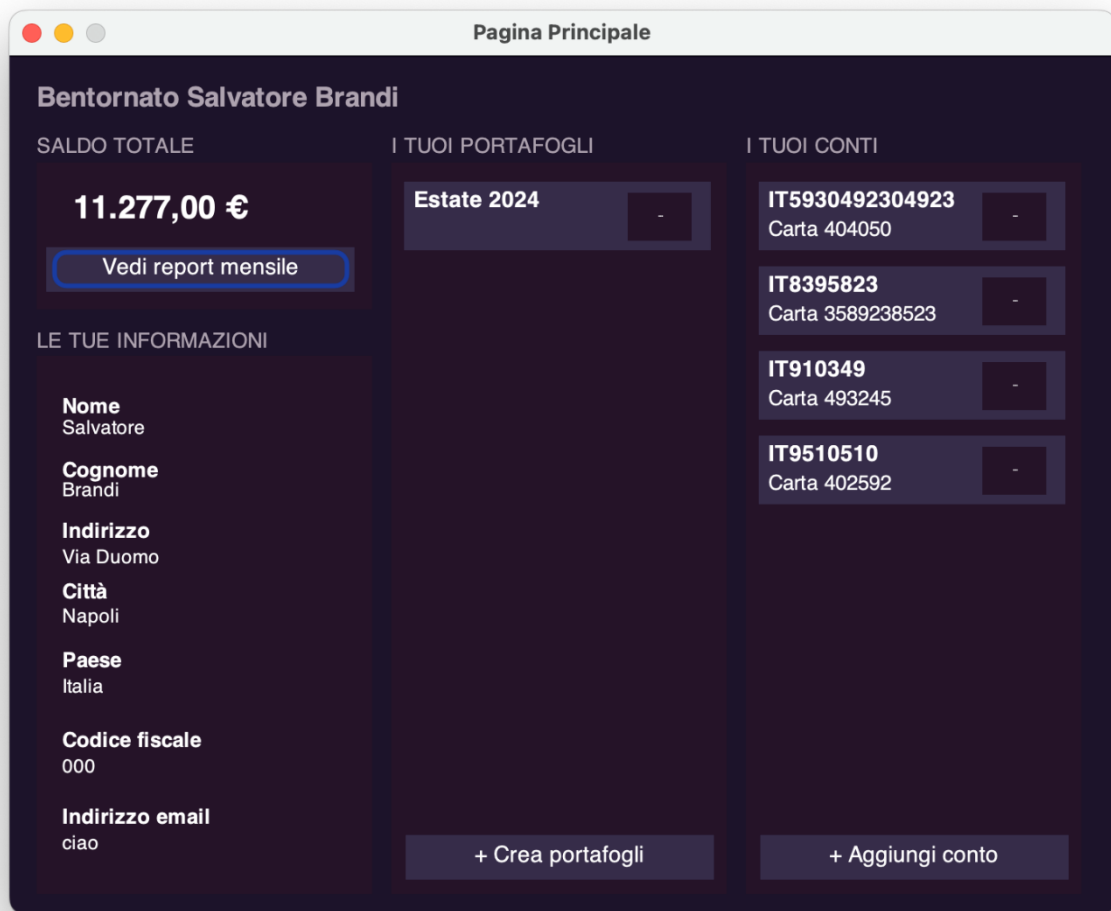


3.1.1: *Funzione di login*

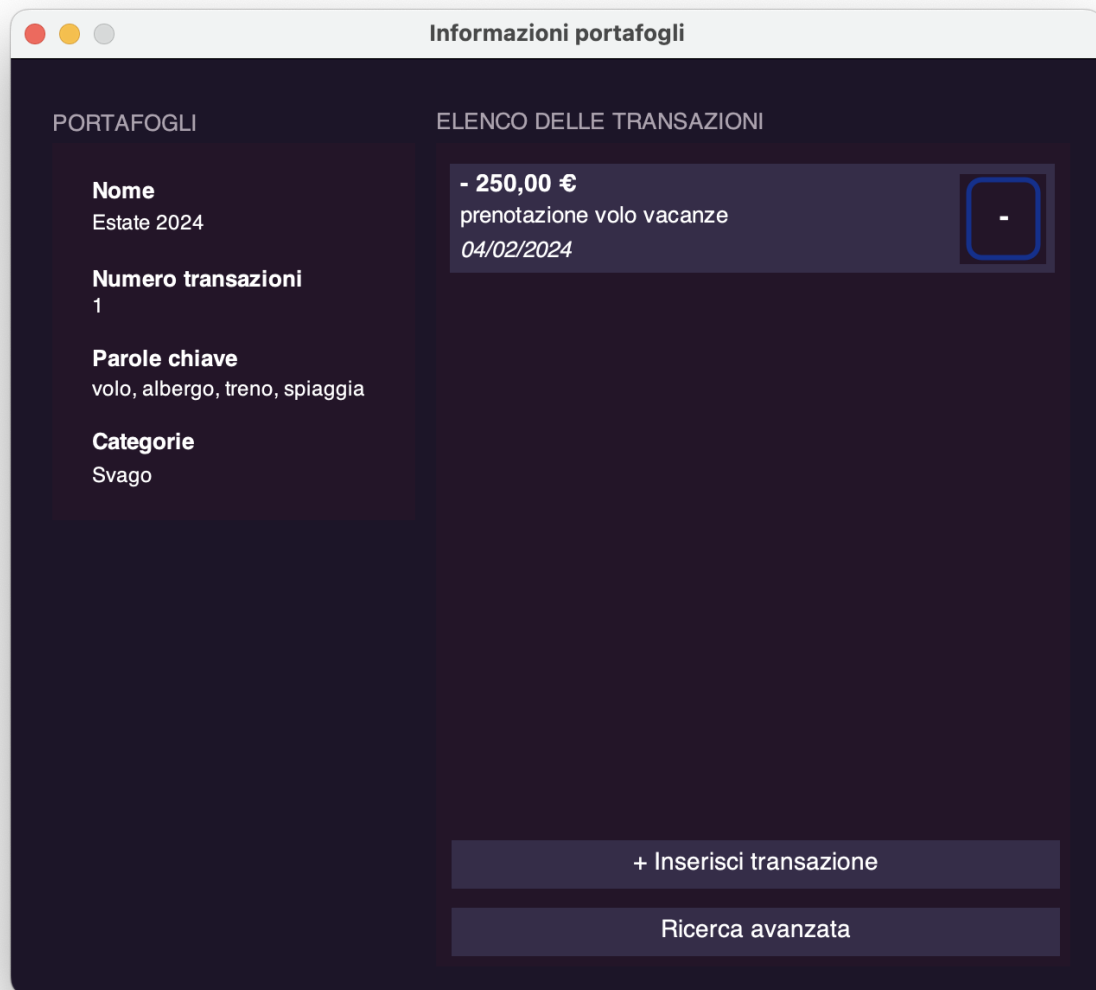
3.2 Mockup e navigazione delle interfacce



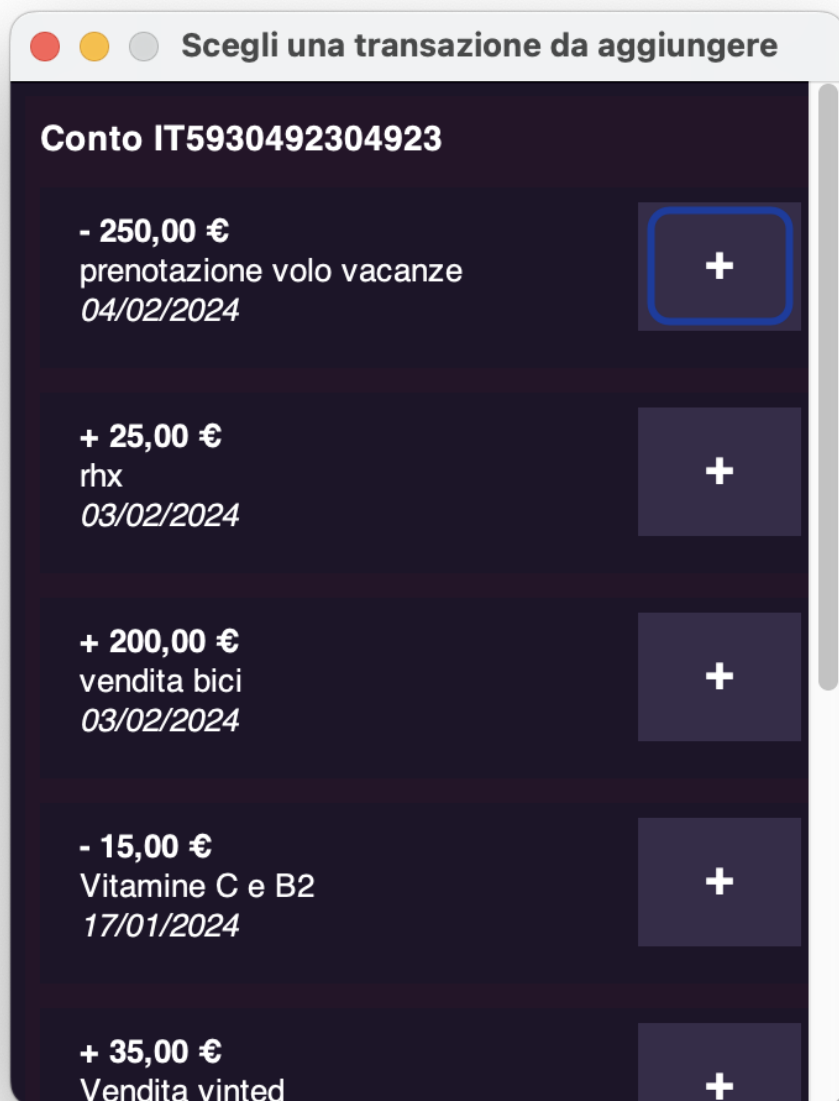
3.2.1: Schermata iniziale di accesso



3.2.2: Rendicontazione generale



3.2.3: Informazioni e gestione di un Portafogli



3.2.4: Inserimento manuale di una transazione in un Portafogli

RICERCA AVANZATA

CONTO CORRENTE

IT5930492304923 (Carta 4040...)

PORTAFOGLI

Estate 2024

DAL

GIORNO

1

MESE

1

ANNO

2021

AL

GIORNO

4

MESE

4

ANNO

2024

Cerca

RISULTATI

- 250,00 €

prenotazione volo vacanze

04/02/2024

3.2.5: Ricerca avanzata di transazioni

Informazioni conto

SALDO

10.010,00 €

DATI DELLA CARTA

Numero

404050

Scadenza

2025-10-10

Codice

335

Tipologia

debito

Limite spesa

1.000,00 €

DATI DEL CONTO

IBAN

IT5930492304923

ELENCO DELLE TRANSAZIONI

- 250,00 €

prenotazione volo vacanze

04/02/2024

-

+ 25,00 €

rhx

03/02/2024

-

+ 200,00 €

vendita bici

03/02/2024

-

- 15,00 €

Vitamine C e B2

17/01/2024

-

+ 35,00 €

Vendita vinted

13/01/2024

-

+ 250,00 €

-

+ Inserisci transazione

Ricerca avanzata

3.2.6: Informazioni e gestione di un Conto Corrente

17

Aggiungi transazione

IBAN ORDINANTE

IMPORTO

DESCRIZIONE

TIPOLOGIA

Entrata

DATA

GIORNO

4

MESE

2

ANNO

2024

+ Aggiungi

3.2.7: Inserimento di una transazione relativa ad un Conto Corrente

Report mensile

MESE

Gennaio

ANNO

2024

REPORT

Conto IT9510510

Entrata massima	Uscita massima
0,00 €	0,00 €
Entrata media	Uscita media
0,00 €	0,00 €
Entrata minima	Uscita minima
0,00 €	0,00 €
Saldo iniziale	Saldo finale
0,00 €	0,00 €

Conto IT910349

Entrata massima	Uscita massima
0,00 €	0,00 €
Entrata media	Uscita media
0,00 €	0,00 €
Entrata minima	Uscita minima
0,00 €	0,00 €

3.2.8: Visualizzazione del report mensile

Aggiungi conto

IBAN

SALDO

NUMERO DELLA CARTA

TIPO CARTA

Credito

PLAFOND

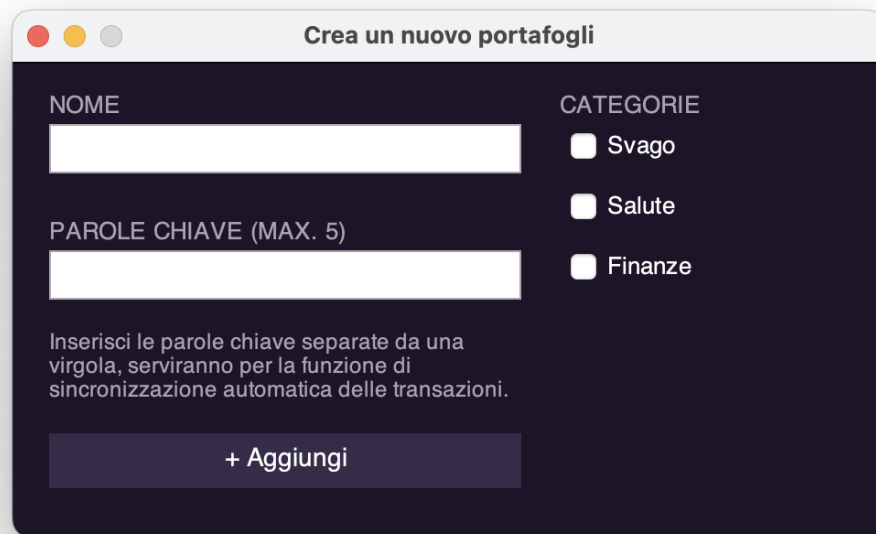
SCADENZA

GIORNO MESE ANNO

CODICE DI SICUREZZA

+ Aggiungi

3.2.9: Inserimento di un nuovo Conto Corrente



Crea un nuovo portafogli

NOME

PAROLE CHIAVE (MAX. 5)

CATEGORIE

- ☐ Svago
- ☐ Salute
- ☐ Finanze

Inserisci le parole chiave separate da una virgola, serviranno per la funzione di sincronizzazione automatica delle transazioni.

+ Aggiungi

3.2.10: Inserimento di un nuovo Portafogli