UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

Dipartimento di Ingegneria
Corso di Laurea Triennale in Informatica

Progettazione e sviluppo della base di dati SavingMoneyUnina

Docente:
Prof. Mara Sangiovanni

Autori:
Francesco Donnarumma
N86004658
Arturo Donnarumma
N86004837

Anno Accademico 2023/2024

Indice

Capitolo 1

Introduzione

Benvenuti nella documentazione dettagliata relativa alla struttura del database di SavingMoneyUnina. Questo documento fornisce una panoramica completa degli elementi chiave che costituiscono la base di dati, offrendo informazioni essenziali sulla progettazione e organizzazione necessarie per una gestione efficiente delle transazioni finanziarie.

Il database di SavingMoneyUnina è stato progettato per facilitare la registrazione, il recupero e l'analisi efficiente delle informazioni finanziarie personali e familiari. Attraverso una struttura intuitiva, consentiamo agli utenti di tracciare e gestire le transazioni provenienti da diverse fonti finanziarie.

La documentazione dettaglierà le tabelle principali, le relazioni chiave e gli schemi di collegamento tra i dati, fornendo una visione chiara sulla gestione automatica e manuale delle transazioni.

Questa guida è essenziale per coloro che necessitano di una visione approfondita sulla progettazione del database, utile sia nello sviluppo che nella manutenzione del sistema nell'ecosistema finanziario.

Capitolo 2

Progettazione Concettuale

2.1 Diagramma Delle Classi UML

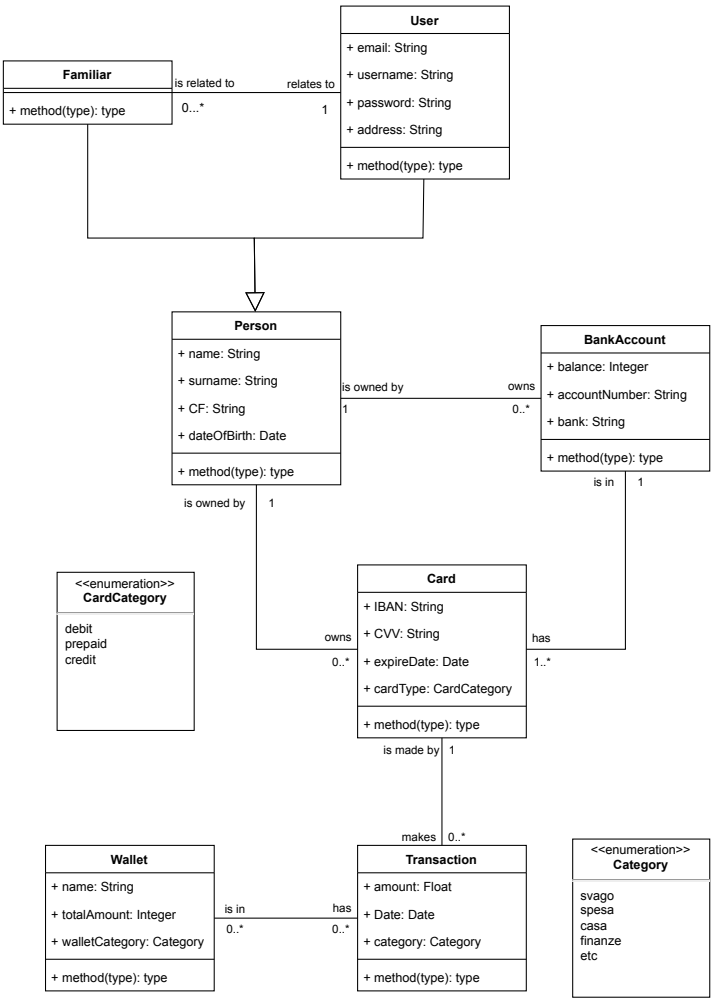

Figura 2.1: Diagramma UML

2.2 Diagramma ER (Entità Relazione)

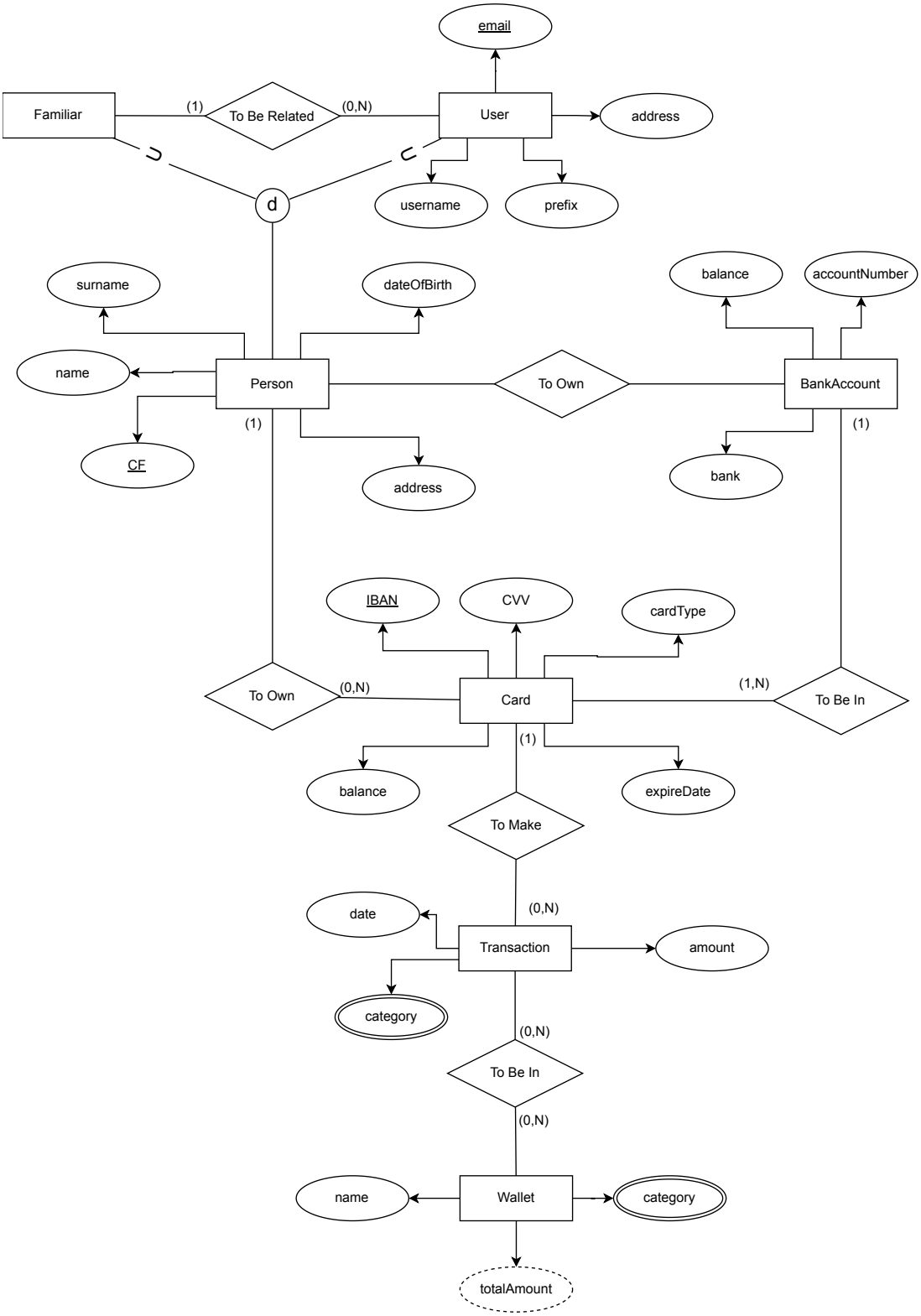

Figura 2.2: Diagramma ER

2.3 Ristrutturazione

2.3.1 Attributi multipli

Per quanto riguarda la gestione di attributi multipli, abbiamo deciso di gestire l'attributo *category* della tabella **Transaction**, originariamente definito come enumerazione, trasformandolo in una stringa, poiché non abbiamo bisogno di valori specifici, trattandosi di una categoria personalizzabile.

Invece, per l'attributo *cardType* della tabella **Card**, è stato deciso di non applicare lo stesso metodo, poiché le tipologie di carte sono ben definite e non possono essere modificate.

2.3.2 Generalizzazioni

Per la generalizzazione, essendo di tipologia totale e disgiunta, abbiamo optato per il metodo di eliminare la classe generale. Abbiamo trasferito tutti gli attributi di essa nelle classi specializzate, conservando le relative relazioni.

2.3.3 Analisi degli identificativi

Per la maggior parte delle classi, saranno utilizzati come identificativi attributi già presenti di natura nelle classi stesse, poiché risultano sufficienti e non richiedono l'uso di una chiave surrogata. Tuttavia, in alcune classi, sono presenti chiavi surrogate, identificate con il prefisso **ID_**.

2.3.4 Diagramma UML ristrutturato

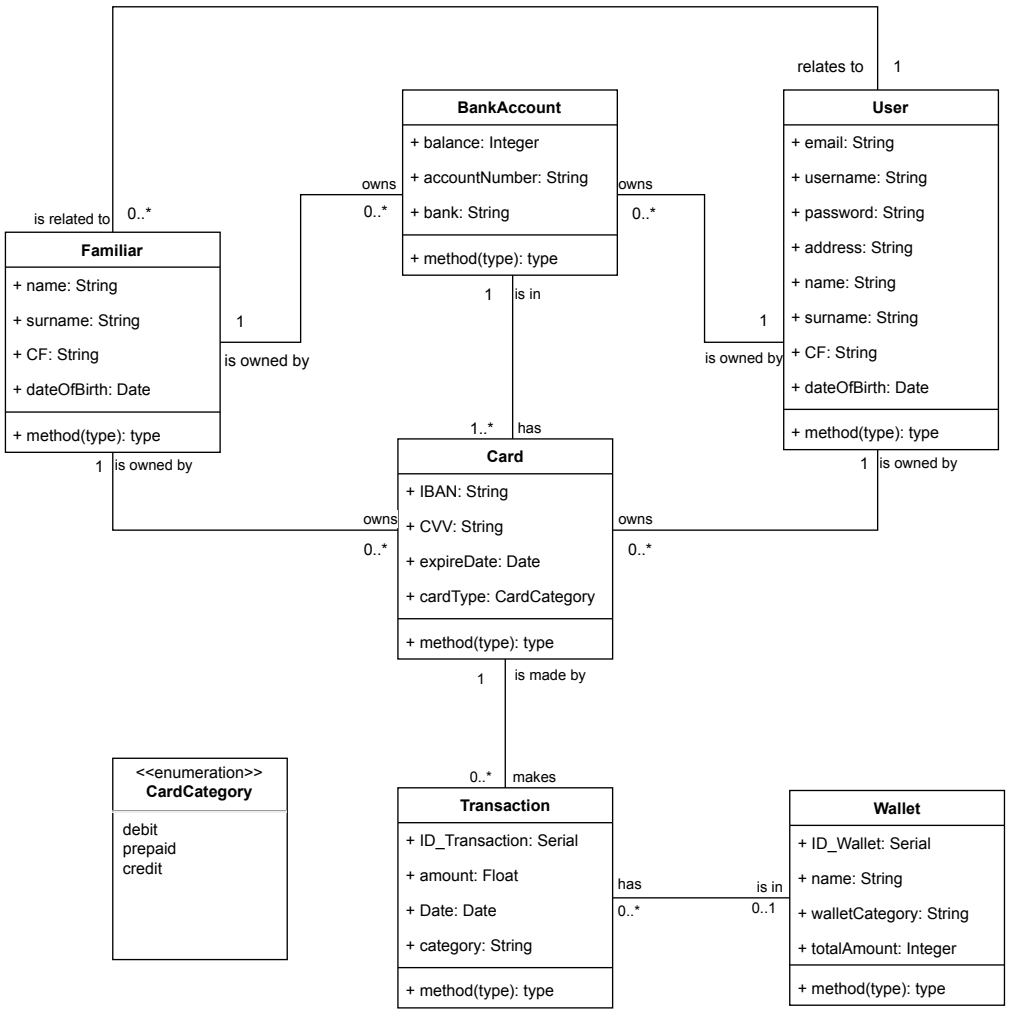

Figura 2.3: Diagramma UML Ristrutturato

2.4 Dizionari

Classe	Descrizione	Attributi
User	Classe utilizzata per identificare gli effettivi utenti che sono registrati alla piattaforma	**email** (*String*): email con la quale l'utente si è registrato **username** (*String*): chiave primaria, identificativa dell'utente. È anche il nome che viene mostrato per riconoscere lo stesso **password** (*String*): stringa atta alla convalidazione durante l'accesso all'account **address** (*String*): indirizzo del domicilio **name** (*String*): nome **surname** (*String*): cognome **CF** (*String*): codice fiscale **dateOfBirth** (*Date*): data di nascita
Familiar	Classe utilizzata per identificare i familiari, degli utenti, che sono presenti sul database	**name** (*String*): nome **surname** (*String*): cognome **CF** (*String*): codice fiscale **dateOfBirth** (*Date*): data di nascita

2.4.1 Dizionario delle classi

2.4.2 Dizionario delle associazioni

2.4.3 Dizionario dei vincoli

Capitolo 3

Progettazione Logica

3.1 Schema Logico

3.1.1 Traduzione delle classi

3.1.2 Traduzione delle associazioni

3.1.3 Schema logico definitivo

Capitolo 4

Schema Fisico

4.1 Definioni SQL delle tabelle