Banking-App

Proiectare Software

Autor: Nemtanu Alexandru-Vasile

**Cuprins**

1. **Obiectiv**
2. **Tehnologii folosite**
3. **Structura proiectului**
4. **Endpoint-uri**
5. **Diagrame**
6. **Concluzii si dezvoltari ulterioare**
7. **Obiectiv**

Proiectul constă în dezvoltarea unei aplicații bancare simplificate care permite utilizatorilor să efectueze operațiuni bancare de bază, cum ar fi crearea de conturi, efectuarea de depozite, retrageri și transferuri de bani între conturi.

1. **Tehnologii utilizate**

Java: Limbajul de programare principal pentru dezvoltarea aplicației.

Spring Boot: Framework-ul pentru dezvoltarea aplicațiilor Java, folosit pentru crearea serviciilor REST.

Spring Data JPA: Abstracție a interacțiunilor cu baza de date, utilizând tehnologia Java Persistence API (JPA).

Mockito: Framework pentru teste unitare și simularea obiectelor în testare.

JUnit: Framework pentru teste unitare în Java.

Lombok: Bibliotecă pentru generarea automată a codului boilerplate, cum ar fi getteri, setteri, constructori etc.

1. **Structura Proiectului**

Proiectul este împărțit în mai multe pachete pentru a organiza codul într-un mod clar și ușor de gestionat:

* **Controller**: Conține clasele care expun endpoint-uri HTTP și gestionează cererile primite de la client.
* **Service**: Conține logica de business a aplicației, care este izolată de straturile de interfețe și acces la date.
* **Repository**: Oferă acces la date prin interacțiunea cu baza de date. Utilizează Spring Data JPA pentru a facilita operațiunile CRUD (create, read, update, delete).
* **Model**: Definirea modelelor de date care sunt utilizate în aplicație.
* **DTO (Data Transfer Object)**: Obiecte utilizate pentru a transfera date între straturile de interfețe ale aplicației și pentru a izola entitățile JPA de straturile superioare.

1. **Endpoint-uri disponibile**

* POST /api/accounts: Crează un nou cont bancar.
* GET /api/accounts/{id}: Returnează informații despre un cont bancar specificat de ID-ul său.
* PUT /api/accounts/{id}/deposit: Realizează un depozit într-un cont bancar specificat de ID-ul său.
* PUT /api/accounts/{id}/withdraw: Realizează o retragere dintr-un cont bancar specificat de ID-ul său.
* GET /api/accounts: Returnează o listă cu toate conturile bancare existente.
* DELETE /api/accounts/{id}: Șterge un cont bancar specificat de ID-ul său.
* PUT /api/accounts/{fromAccountId}/transfer/{toAccountId}: Transferă bani între două conturi bancare.