Proiect Programare și Proiecate Orientată Obiect

Facultatea de Cibernetică, Statistică și Informatică Economică

ASE București



Profesor: Conf. Univ. Dr. Cristian Ciurea

Student: Țucă Mădălin-Gabriel

Temă: Aplicație e-Banking

Contents

[Descrierea Aplicației 3](#_Toc86971220)

[**Problema întâmpinată** 3](#_Toc86971221)

[**Soluția Software** 3](#_Toc86971222)

[Tehnologii Utilizate 4](#_Toc86971223)

[**Limbaj de programare** 4](#_Toc86971224)

[**Biblioteci și programe externe** 4](#_Toc86971225)

[Implementarea Aplicației 5](#_Toc86971226)

[**Structura claselor** 5](#_Toc86971227)

[**Fluxul logic al aplicației** 6](#_Toc86971228)

[Concluzie 7](#_Toc86971229)

## Descrierea Aplicației

### **Problema întâmpinată**

Presupunem existența a unei bănci fictive. Banca respectivă dorește eficientizarea proceselor financiar-bancare pentru o colaborare mai bună cu clienții săi.

În prezent, clienții băncii sunt nevoiți să se prezinte la bancă sau la un dispozitiv ATM pentru a efectua diferite tranzacții (Încărcare sold, Depunere sold, Creare cont, etc.)

Banca dorește o soluție software care să permită desfășurarea activităților bancare cu clientul de la distanță, în special în contextul actual pandemic.

### **Soluția Software**

Soluția software dezvoltată oferă o interfață intuitivă pentru orice utilizator, astfel încât aceștia să își poată gestiona conturile bancare, să poată efectua tranzacții sau alte operațiuni specifice cu banca la care sunt abonați.

Aplicația oferă un meniu ce pune la dispoziție o suită de funcționalități. Utilizatorii băncii își pot crea un cont pentru aplicație, iar în acest cont își pot integra conturile deschise la bancă.

Odată înregistrați si autentificați, aceștia pot efectua tranzacții precum:

* Interogare sold
* Depunere numerar
* Adăugare Numerar
* Adăugare/Anulare cont bancar

Prin funcționalitățile puse la dispoziție, prezența fizică la bancă este redusă aproape la minimum, acest lucru oferind o lejeritate atât pentru clienți, cât și pentru personalul bancar. Productivitatea personalului bancar este de asemenea îmbunătățită, aceștia având mai mult timp pentru a se canaliza asupra lucrurilor importante în cadrul băncii.

## Tehnologii Utilizate

### **Limbaj de programare**

Aplicația software a fost dezvoltată folosind limbajul Java, versiunea 16, ce include funcționalități actualizate, în pas cu restul limbajelor și al practicilor folosite în zilele de astăzi.

Java este un limbaj cunoscut pentru securitatea oferită aplicațiilor, reducând posibilitatea atacatorilor.

Spre deosebire de alte limbaje de programare și sisteme, unde securitatea sistemelor informatice era implementată după dezvoltare sau deja după anumite atacuri, mecanismele de securitate fac deja parte din tehnologia limbajului Java (Cay S. Horstmann, 2006)

### **Biblioteci și programe externe**

Pe lângă bibliotecile specifice limbajului, a fost folosită biblioteca Gson, bibliotecă Java

dezvoltată de Google, ce oferă metode predefinite de lucru cu fișierele de tip JSON, precum citire fișier JSON, scriere JSON, parsare, formatare etc.

Această bibliotecă a fost dezvoltată pentru uz intern în compania Google, urmând să fie lansată public în 2008 în versiunea 1.0, actualizată până în 2021 la versiunea 2.8.8 (Wikipedia, 2021)

Ca și mediu de dezvoltare, a fost folosit IDE-ul Intellij IDEA ce pune la dispoziție o gamă largă de utilități menite să ușureze procesul de dezvoltare al aplicației.

Printre aceste utilități putem enumera:

* Auto-formatarea codului sursă
* Text sugestiv
* Completare automată
* Analizare și îmbunătățire cod sursă
* Interfață de versionare a codului sursă

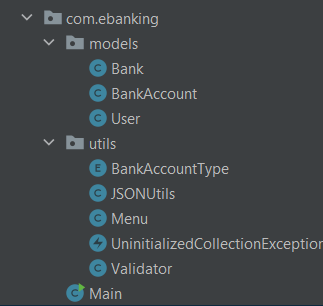
Ca metodă de versionare a codului sursă, a fost folosit Git și Github. Versionarea codului sursă permite dezvoltarea sistemică a unei soluții software, oferind atât posibilitatea de salvare/restaurare a codului sursă, cât și dezvoltarea colaborativă a unei aplicații, în branșe, fără a afecta cod sursă scris de către alt programator.

## Implementarea Aplicației

### **Structura claselor**

Referitor la proiectarea aplicației, aceasta a fost împărțită în trei clase concrete, trei clase utilitare, o clasă de excepție custom și un Enum.

Structura fișierelor se poate observa în firgura de mai jos.



**Clasa Bank:**

Este clasa ce are ca proprietăți datele referitoare la banca ce deține aplicația de e-banking. Având în vedere că este aplicația unei singure bănci, este permisă deci o unică instanță a clasei Bank.

Pentru a ne asigura de acest lucru, a fost folosit design pattern-ul creațional Singleton.

**Clasa User:**

Este clasa ce are ca proprietăți datele unui client al bănci, această clasă este folosită în concordanță cu clasa Bank, deoarece una dintre proprietățile acesteia este o listă cu clienții băncii.

**Clasa BankAccount:**

Clasa BankAccount este responsabilă cu deținerea informațiilor referitoare la conturile bancare înregistrate la bancă. Fiecare client al băncii deține în contul său o listă cu maxim trei conturi(Credit, Debit, Economii).

**Clasele utilitare:**

Clasele utilitare dețin metode statice specifice ce vin în sprijinul aplicației și ajută la structurarea și ordonoarea codului sursă, respectând principiul de „Seggregation of Responsabilities”.

### **Fluxul logic al aplicației**

În momentul lansării în execuție al aplicației, aceasta oferă un meniu introductiv ce permite utilizatorului să se autentifice în contul său, sau să își creeze un nou cont în cadrul aplicației.

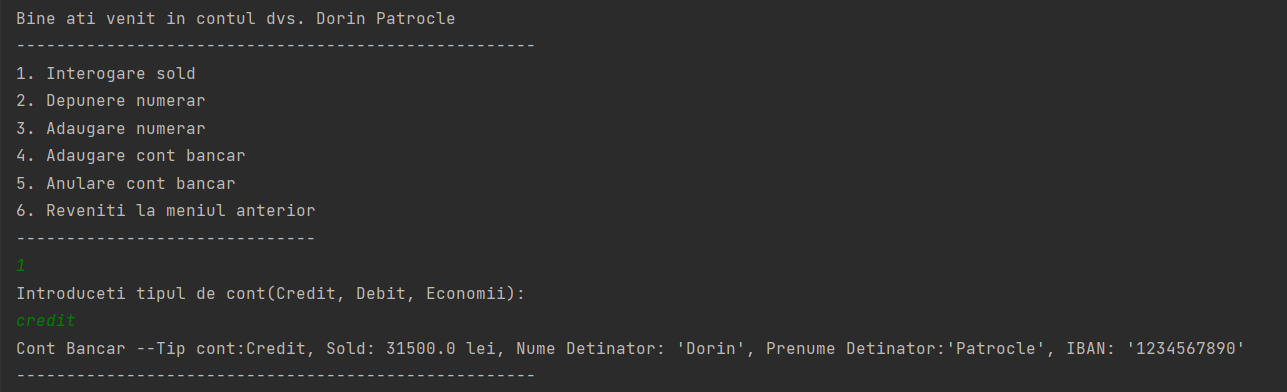
Utilizatorul va fi îndrumat spre a introduce datele necesare înregistrării/autentificării.

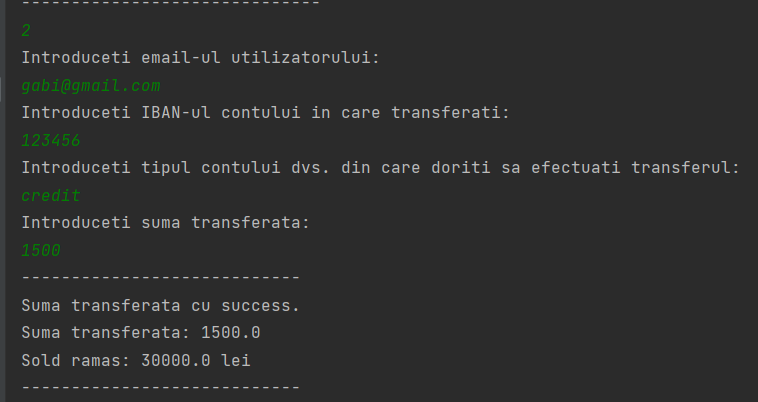
La momenul autentificării, dacă aceasta a fost îndeplinită cu succes, utilizatorului îi va fi prezentat un nou meniu, în care poate efectua diverse operațiuni bancare.

În funcție de alegerea făcută de utilizator, acesta va fi îndrumat prin intermediul interfeței din consolă spre finalizarea procesului bancar. Datele de intrare sunt validate, iar în cazul în care acestea nu sunt corecte, utilizatorului îi se va pune la dispoziție o nouă încercare până când datele sunt valide.

După cum a fost menționat mai sus, acesta își poate interoga soldul, depune numerar în alt cont bancar, adăugare sumă în cont, adăugare nou cont bancar și de asemenea nularea conturilor bancare la alegere.

Câteva dintre aceste funcționalități vor prin prezentate mai jos:





În prima figură este prezentată funcționalitatea de interogare sold. La alegerea tipului de cont de către utilizator, atât soldul, cât și datele utile ale contului vor fi afișate către utilizator.

În a doua figură este prezentată funcționalitatea de depunere numerar într-un alt cont. Utilizatorul va fi îndrumat să ofere toate datele necesare transferului, precum suma și IBAN-ul utilizatorului destinatar. Transferul va putea fi efectuat doar dacă soldul curent este mai mare sau egal decât suma transferată.

## Concluzie

În cele din urmă, aplicația dezvoltată oferă o soluție optimă pentru gestiunea operațiunilor bancare, punând la dispoziție facilitățile oferite de domeniul e-Banking oricărui tip de utilizator.

Prin funcționalitățile acesteia, clienții băncii își pot utiliza serviciile oferite de aceasta de la distanță.

De asemenea, implicarea prezenței fizice la bancă pentru a efectua aceste activități a fost redusă considerabil iar problema băncii a fost rezolvată.