

Documentatie Aplicatie Bancara

Proiect de semestru la Inginerie Software

HÎRȚESCU CIPRIAN GABRIEL

February 21, 2023

1 Abstractizare

Aplicatia propusa descrie un sistem care simuleaza o aplicatie bancara cat mai apropiata de cele de zi cu zi. Ca si principala functionalitate aplicatia va putea sa permita logarea utilizatorilor pe baza unui id si a unei parole asociate. In cazul in care utilizatorul nu se afla in baza de date a bancii poate alege sa completeze un formular nou prin care sa isi creeze un cont in acest sens.

O data ce utilizatorul are posibilitatea de a se loga in aplicatie, se pot distinge doua cazuri pe baza credentialelor introduse in main si anume credentialele sunt asociate unui admin, caz in care se va deschide o pagina asociata adminului, altfel pentru un client obisnuit se va afisa pagina suport a acestuia, iar in final pentru credentiale gresite, sau negasite in aplicatie se va afisa o informare in acest sens.

Revenind la paginile asociate clientului si adminului, acestea sunt diferite mai ales datorita obiectivelor fiecaruia.

Se presupune de la inceput ca un client va putea sa execute urmatoarele operatiuni:

1. Sa se logheze sau sa isi creeze un cont nou,
2. Sa initieze un transfer spre un alt cont ce poate fi interpretat ca o plata a unei facturi sau orice alta tranzactie de acest fel.
3. Din pagina sa sa isi poata vedea soldul pe care il detine
4. Se poate reveni la pagina anterioara din orice alta pagina in log-out

Pe partea de admin se schimba destul de mult functionalitatea, exemplificarea operatiunilor posibile este desfasurata mai jos:

1. Sa se logheze,
2. Sa poata vedea toti clientii din baza de date sub forma unui tabel
3. Sa poata vedea toate tranzactiile atat cele aprobate cat si cele care urmeaza sa fie aprobate din baza de date sub forma unui tabel
4. Sa poata vedea toate tranzactiile care au fost primite de la un client, astfel incat sarcina de aprobare sau declin ii apartine, o astfel de reprezentare va fi asemanatoare cu cele de mai sus, diferenta ar fi ca in dreptul fiecarei tranzactii vor fi doua butoane de accept sau declin care conduc operatiunile specifice.
5. Se poate reveni la pagina anterioara din orice alta pagina in log-out

2 Descriere

2.1 Limbajul folosit, framework-uri

Limbajul folosit este java care aduce in prim-plan principiile de oop prin delimitarea clara intre clase, si a imparti functionalitatea pe mai multe capitole.

Clase:

1. SecurityConfiguration: verifica datele introduse ca si credentiale si cripteaza parola
2. Cont: contul clientului
3. Role: rolul utilizatorului
4. Transactions: tranzactiile asociate contului unui client
5. User: utilizatorul general
6. ContServiceImpl: contine metodele de CRUD de la nivelul tabeli cont
7. TransactionServiceImpl: contine metodele de CRUD de la nivelul tabeli transactions
8. UserServiceImpl: contine metodele de CRUD de la nivelul tabeli user
9. TransactionDto: creeaza un obiect nou de tipul tranzactie pe baza datelor primite
10. UserRegistrationDto: creeaza un user pe baza datelor primite
11. ClientController: contine metodele de handle pentru request-urile primite de la pagina asociata clientului
12. MainController: contine metodele de handle pentru requesturile generale
13. UserController: contine metodele de handle pentru requesturile asociate adminului
14. UserRegistrationController: contine metodele de handle pentru request-ul de creare a unui nou user
15. Main: RegistrationLoginSpringBootSecurityThymeleafApplication, aplicatia de baza unde se va rula programul

Interfete

1. ContRepository: repository-ul asociat tabeli cont
2. TransactionsRepository: repository-ul asociat tabeli transactions
3. UserRepository: repository-ul asociat tabeli user
4. ContService
5. TransactionsService
6. UserService

Framework-uri si dependinte:

1. SpringBoot
2. Spring Data Jpa
3. Spring Web
4. Spring Security
5. MySQL Driver

2.2 Diagrama bazei de date cu tabelele asociate

Diagrama bazei de date exemplificata [3](#)

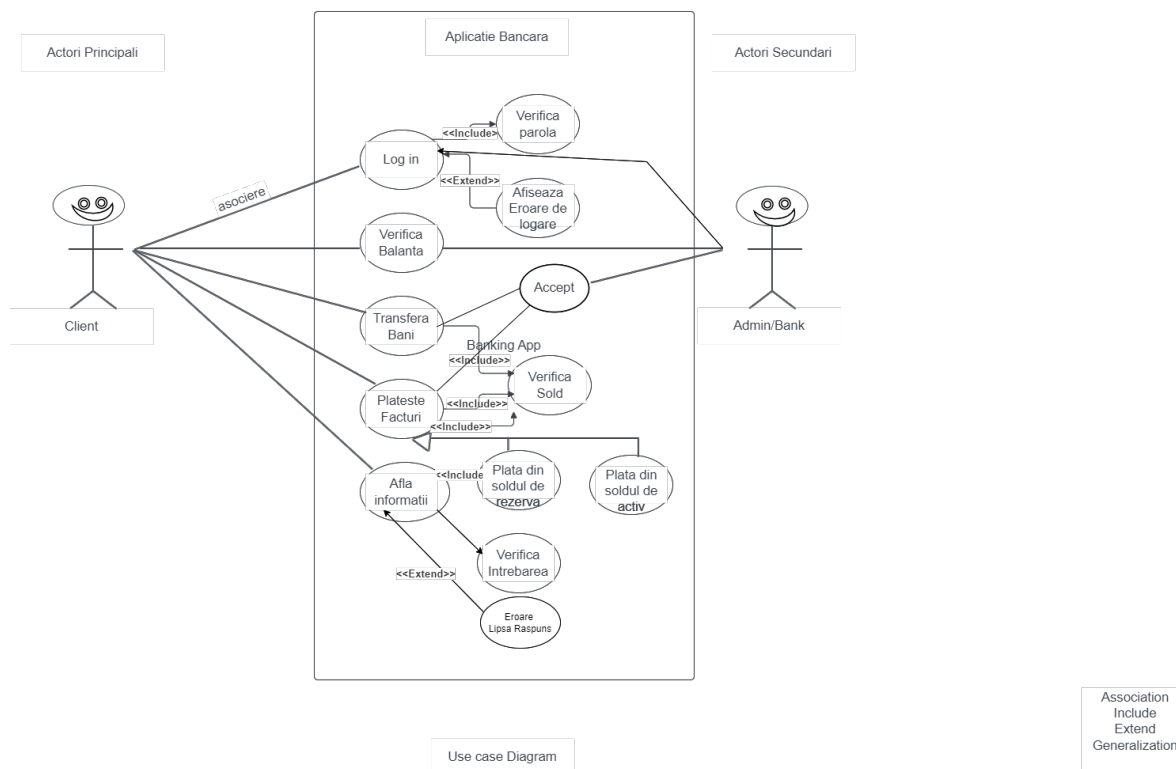


Figure 1: Diagrama de use case-uri.

2.3 Diagrama de use-case-uri

Diagrama de use case-uri prezentata mai jos exemplifica foarte bine intr-un mod usor de inteles ce va putea face aplicatia si cum va rula aceasta in spate.

1

Dupa cum se poate observa din diagrama exista doi actori posibili: principali(clientii) si secundari(adminii). Conform diagramei reies multimea de functionalitati precizate anterior.

3 Manual de utilizare:

In vederea folosirii aplicatiei sunt necesare trei tool-uri: intelij pentru java , mySql workbench pentru baza de date si un motor de cautare de preferat google chrome. Dupa descarcarea aplicatiei se deschid toate cele trei tool-uri mentionate anterior impreuna cu proiectul si se ruleaza.

In chrome se poate scrie portul local: <http://localhost:8080/> care va conduce la pagina principala de logare in mod automat.

De aici ii revinte utilizatorului sa aleaga ce are de facut, daca vrea sa se autentifice sau sa isi creeze un cont nou.

In functie de alegere va aparea o fila noua asociata operatiunii efectuate anterior.

Pe langa detaliile enuntate anterior se prezinta mai jos o seria de imagini destul de sugestive pentru aplicatie:

1. Pagina de login: 4
2. Pagina de inregistrare ca si client nou: 5
3. Pagina de asociata clientului 6
4. Pagina de asociata adminului 7

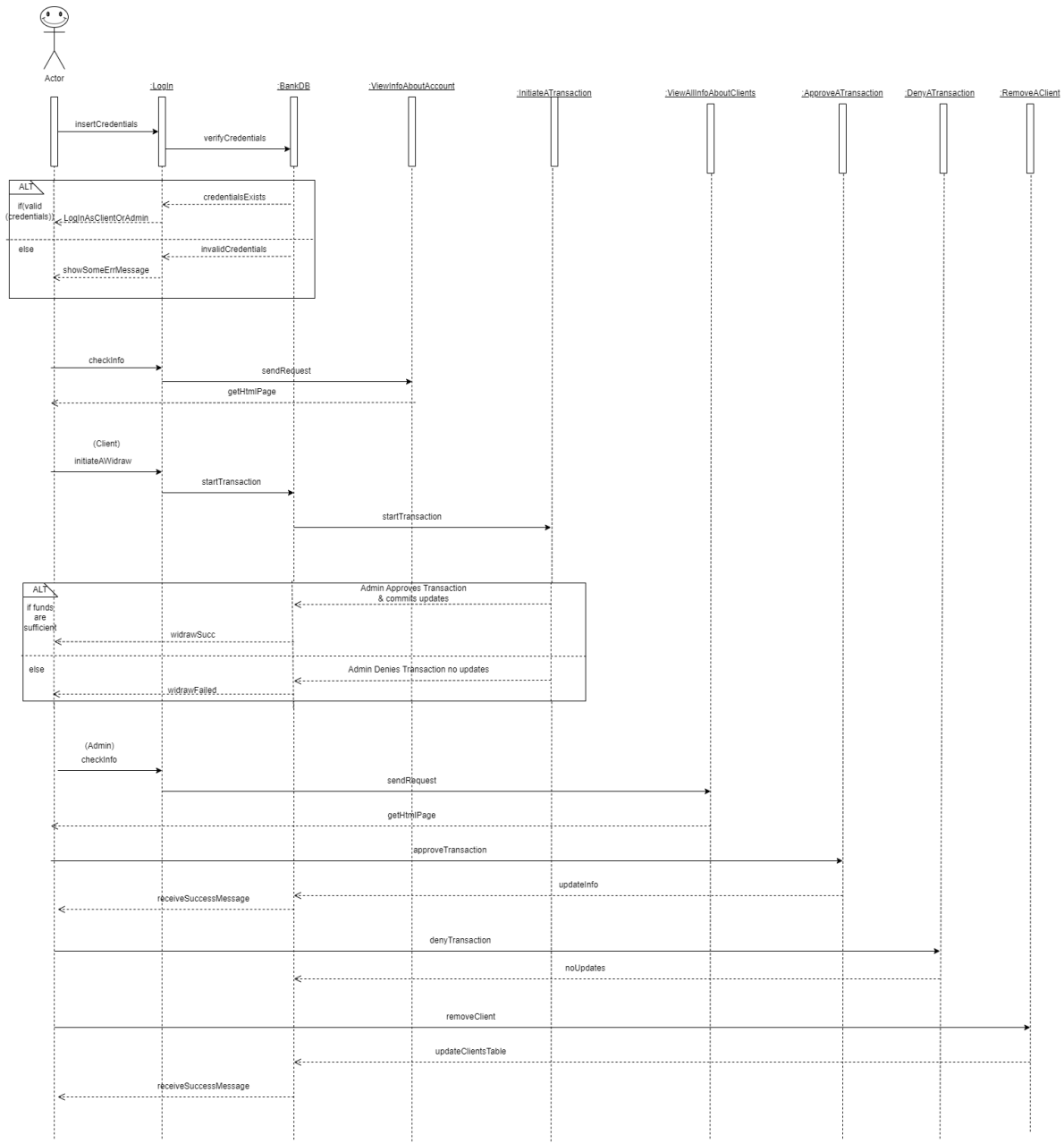


Figure 2: Diagrama de secvente.

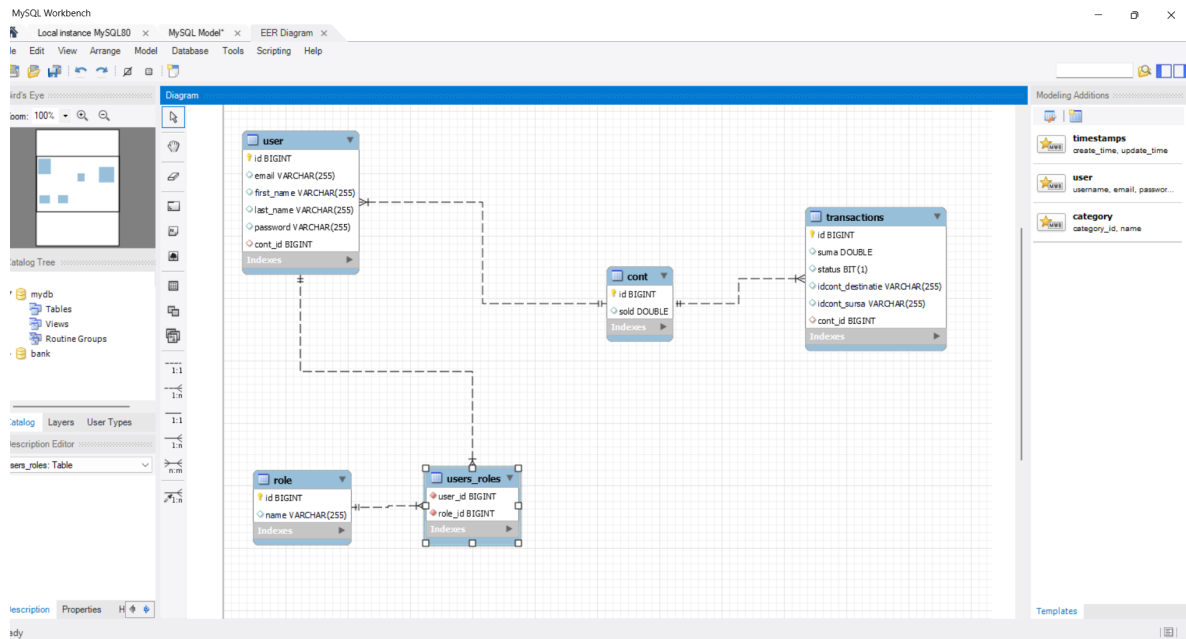


Figure 3: Diagrama MySQL cu tabele si asocierile respective.

The screenshot shows a web browser window with the URL `localhost:8080/login`. The page title is "Registration and Login App". The main content area displays the "User Login Page" with the following elements:

- Username :** A text input field with the placeholder text "Enter Email ID".
- Password:** A text input field with the placeholder text "Enter Password".
- Log In**: A blue button.
- New user? Register here**: A link.

Figure 4: Pagina Principala

Registration

First Name

Last Name

Email

Password Completează acest câmp...

[Register](#) [Already registered? Login here](#)

Figure 5: Pagina de inregistrare

Welcome user: as@gmail.com

Display Sum:

1555.5500000000002

Tranzactii Efectuate:

Destination	Amount	Status
asd@gmail.com	20.0	true
asd@gmail.com	220.0	true
asd@gmail.com	320.0	true

New Transaction :

0.0

Destination

[Send](#)

Figure 6: Pagina asociata clientului

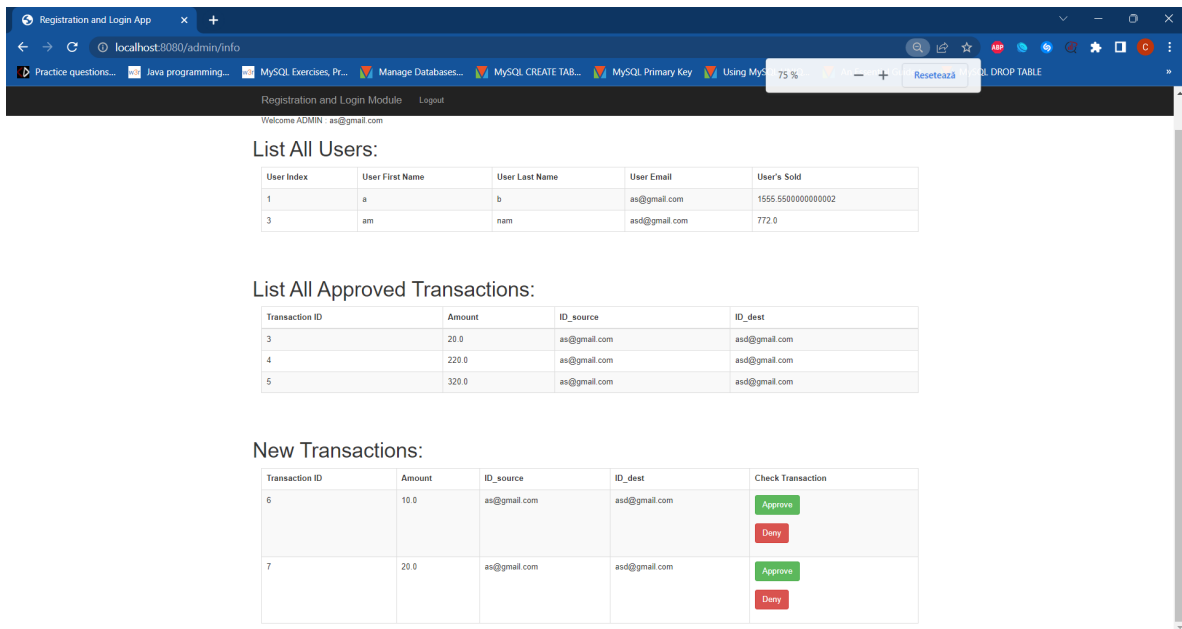


Figure 7: Pagina asociata adminului

3.1 Tutorial de utilizare

Pentru inceput se ruleaza aplicatia in ide-ul: IntelliJ in paralel si MySql Workbench trebuie sa fie deschis, iau user-ul root sa fie logat.

Urmatorul pas este de a deschide un browser de preferat chrome si sa se scrie adresa: <http://localhost:8080/> care va redirectiona spre pagina de login asociata. Mai apoi user-ul poate alege sa se logheze sau sa isi creeze un cont nou, astfel, creindu-se o noua intrare in baza de date asociata unui utilizator nou, alaturi de toate detaliile urmatoare: cont nou cu sold-ul de 0.0 si o lista de tranzactii vida.

Ajungand la pagina clientului, daca se cunoaste despre acesta explicit ca este admin, atunci el are posibilitatea de a trece la o alta pagina asociata acestuia, unde poate vedea toate conturile existente, tranzactiile aprobate si tranzactiile noi aparute.

Tranzactiile noi aparute ii revin responsabilitatea adminului sa le verifice, iar apoi sa le aprobe sau sa le respinga, ceea ce va conduce la trecerea amount-ului declarat in tranzactie de la contul sursa la contul destinatie, actualizand in prealabil toate informatiile din baza de date.

References

- [1] Spring details. [\[resource\]](#)
- [2] Spring tutorial Baeldung. [\[resource\]](#)