

IONEL ROMÂNIA

- Document de proiectare arhitecturală-

Criciu Cosmin – 331CC

Dumitriu Stefan-Alexandru – 331CC

Ichim Dan-Gabriel – 331CC

Poalelungi Gabriel – 331CC

Cuprins

1.INTRODUCERE.....	3
1.1. Scopul sistemului.....	3
1.2. Definiții și acronime.....	3
2. OBIECTIVE DE PROIECTARE.....	4
3. ARHITECTURA PROPUȘĂ.....	5
3.1. Prezentarea generală a arhitecturii sistemului.....	5
3.2. Decompoziția în subsisteme și responsabilitățile fiecărui subsistem.....	6
3.3. Distribuția subsistemelor pe platforme hardware/software.....	8
3.4. Managementul datelor persistente.....	9
3.5. Controlul accesului utilizatorilor la sistem.....	11
3.6 Fluxul global al controlului.....	11
3.7 Condițiile limită.....	12
4. PLANIFICAREA IMPLEMENTĂRII PROIECTULUI.....	13

1.1 Scopul sistemului

Aplicația de online-banking “IONEL ROMANIA” aduce la un loc mai multe tipuri de operații financiare, ceea ce reprezintă un avantaj fata de aplicațiile bancare concurente. Pe lângă operațiunile bancare de bază, precum informațiile despre conturi (sold, istoric tranzacții), aprobare tranzacții sau transferuri bancare și interbancare, sunt implementate următoarele soluții: măsuri de securitate transparente și care nu necesită efort din partea utilizatorului (băncile obișnuiesc să își cheme utilizatorii la sediul acestora pentru a-și confirma contul pe aplicație, dar aplicația noastră folosește multi-factor authentication, după ce identitatea utilizatorului a fost confirmată), creare instantă de conturi în valută, schimb valutar actualizat zilnic, analiză transparentă a veniturilor și a cheltuielilor, opțiunea de a programa plăți recurente, transfer rapid de bani sau valuta, fără comisioane, doar pe baza numelui de utilizator sau prin tehnologia NFC pentru dispozitivele mobile, dar și posibilitatea de a plăti la terminalele comercianților prin NFC. De asemenea, există opțiuni de economisire sau de investire a banilor utilizatorului, precum opțiunea “seif” pentru a-și depozita banii sau “broker” pentru a-și investi banii în acțiuni.

1.2. Lista de definiții si acronime

Acronime:

- CVV - Card Verification Value
- NFC - Near Field Communication
- IBAN – International Bank Account Number
- API – Application programming interface
- Throughput – Număr de cereri/unitate de timp
- HTTPS - HyperText Transfer Protocol/Secure

2. Obiective de proiectare

- a. Îndeplinirea cerințelor de performanță**
- b. Îndeplinirea cerințelor de calitate**
- c. Îndeplinirea cerințelor de portabilitate**

Criterii de performanță

1. Timpul maxim de răspuns al sistemului (încărcare pagini + actualizare date) să fie mai mic de 5s
2. Throughput-ul pentru ciclul de request/response al API-ului va fi cel puțin de 100/s
3. Datele provenite din surse externe vor fi actualizate o dată la 1h (curs valutar + valoare acțiuni)

Criterii de calitate

1. Uptime-ul componentelor interne ale aplicației (ce nu depind de serviciile externe) va fi minim 95%.
2. Securizare backend site prin utilizarea unei semnături unice criptate.
3. Securizarea împotriva atacurilor de tip MITM, CSRF, XSS, precum și redirectarea request-urilor HTTP către HTTPS.
4. Log-out automat al unui utilizator după o perioadă de inactivitate.
5. Aplicația va cere reintroducerea unei parole pentru fiecare tranzacție.

Criterii de portabilitate

1. Codul sursă va fi clar și modular și va fi documentat, pentru o mentenanță mai ușoară dar și pentru o adăugare ulterioară a altor funcționalități.
2. Vor exista elemente ce pot fi reutilizate între aplicația mobilă și cea web.

3. Arhitectura propusă

3.1 Prezentare generală a arhitecturii sistemului

Web Application – Containerul prin care user-ul interacționează cu serviciile aplicației de mobile banking. Este reprezentat de gestionarea contului utilizatorului, vizualizarea conturilor, realizarea tranzacțiilor și cumpărarea și vânzarea acțiunilor.

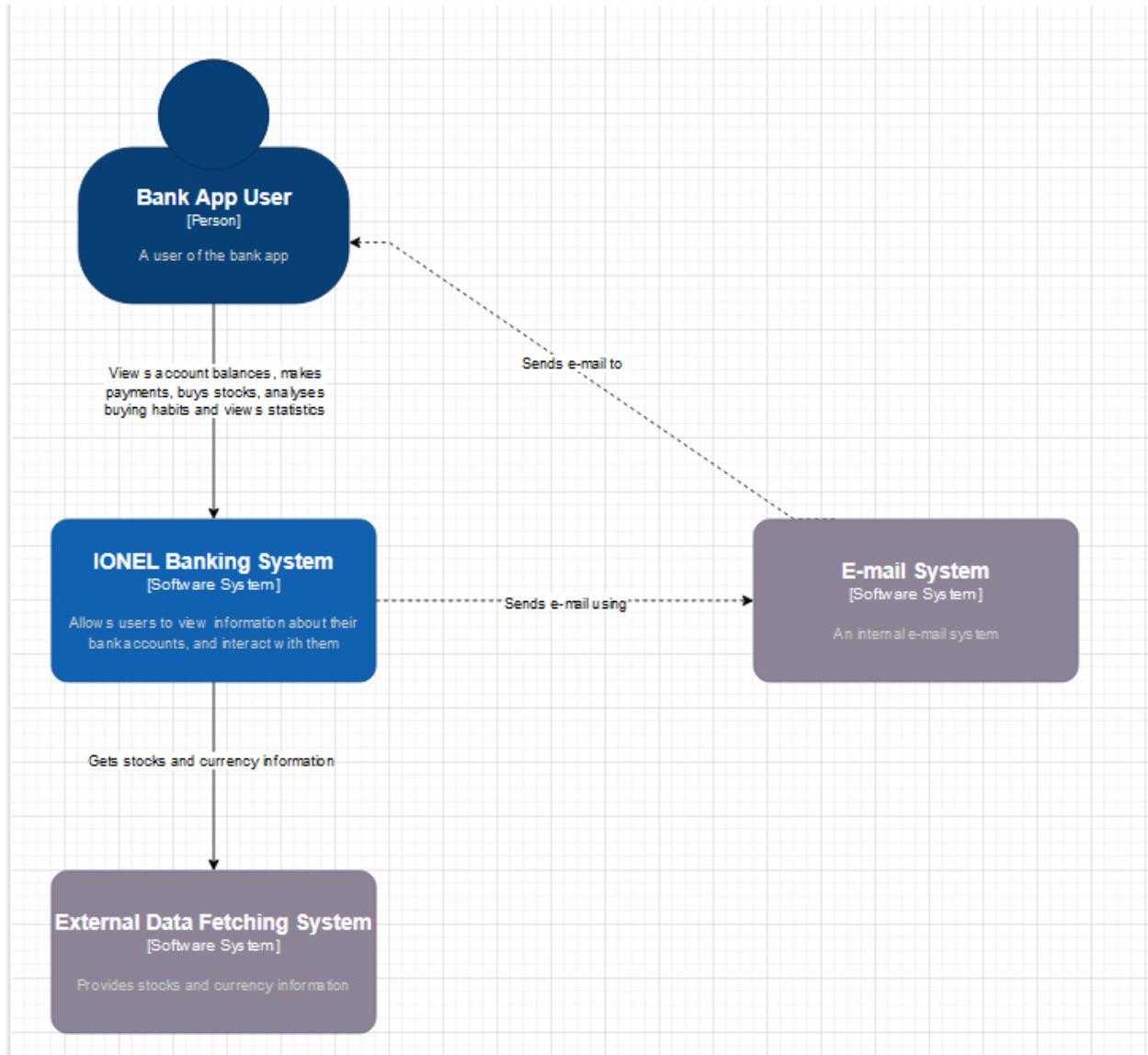
Single-Page Application – Oferă funcționalitate site-ului web, pentru că utilizatorul să poată interacționa prin intermediul unui motor de căutare.

Mobile Application - Oferă funcționalitate aplicației de mobil, pentru că utilizatorul să poată interacționa prin intermediul unei aplicații.

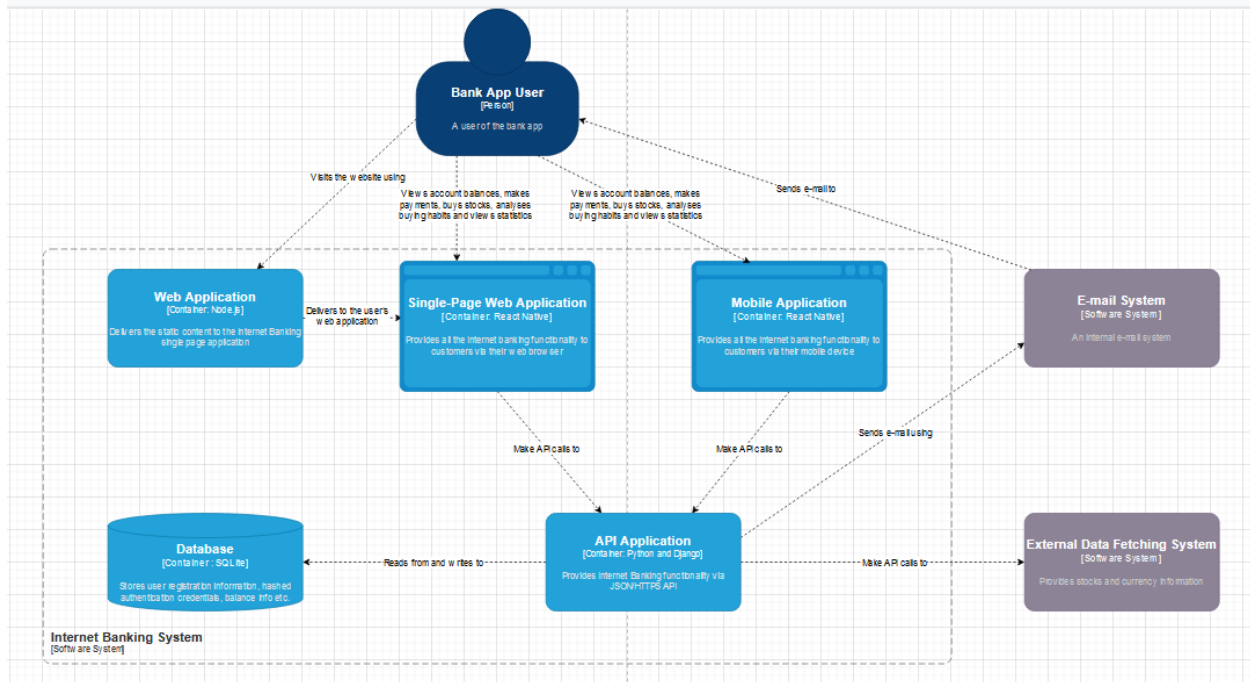
API Application – Containerul care oferă funcționalitate aplicației web, prin verificarea bazei de date și transmiterea informațiilor către utilizator.

Database – Stochează datele necesare rulării aplicației, precum datele utilizatorilor, tranzacțiile efectuate, acțiunile cumpărate sau detaliile conturilor.

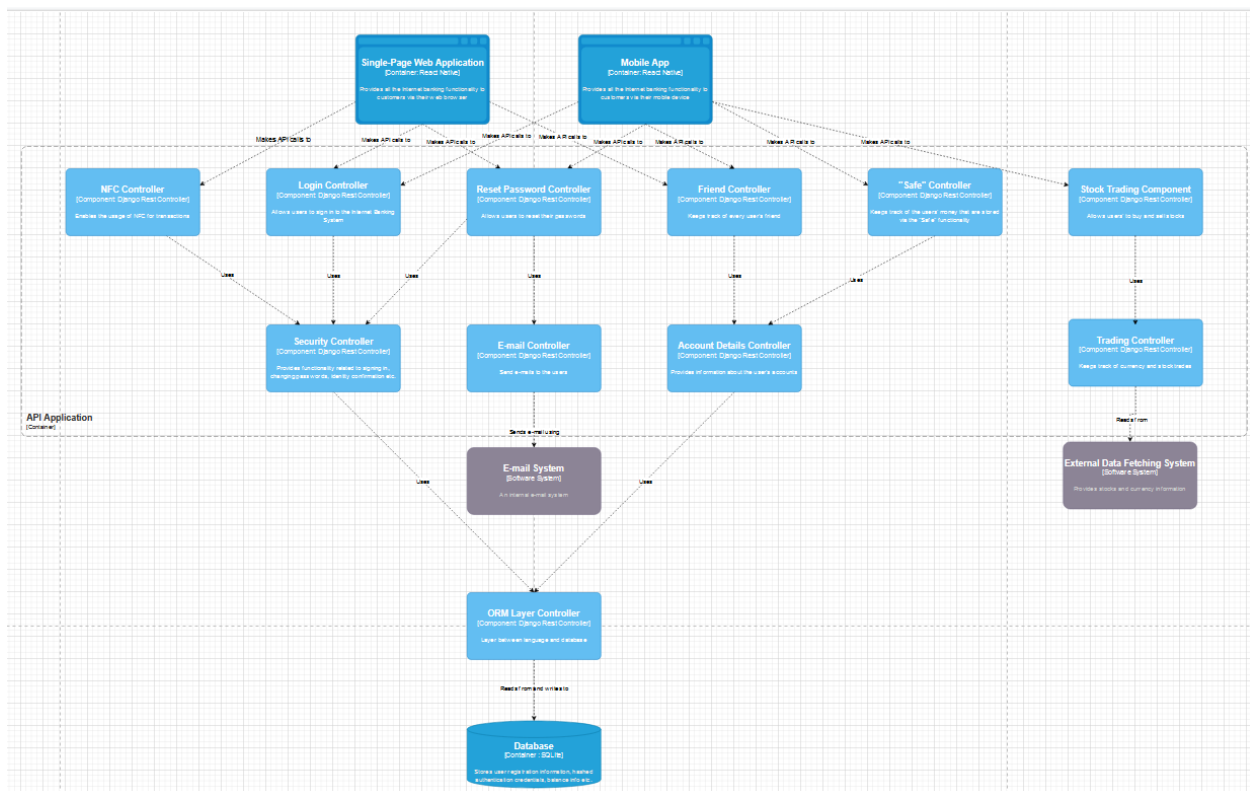
3.2 Decompoziția în subsisteme și responsabilitățile fiecărui subsistem



Contextul sistemului



Descrierea containerelor



Descrierea componentelor

3.3 Distribuția subsistemelor pe platforme hardware/software (diagrama de distribuție)

Împărțirea întregului sistem este făcută pe trei platforme hardware separate:

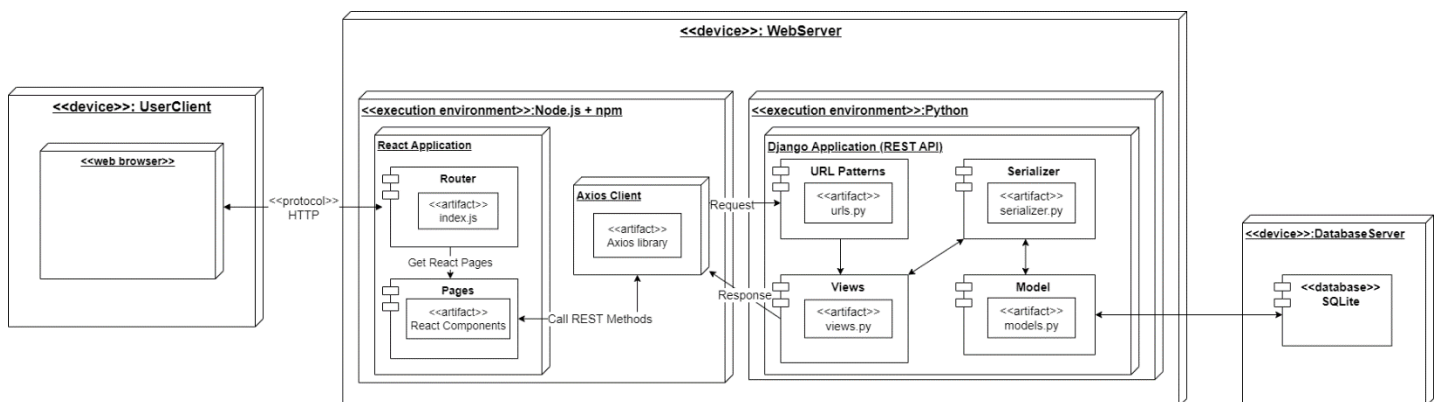
- client-side - user-ul va interacționa cu aplicația noastră fie prin browser, fie printr-o aplicație mobilă Android .

- server-ul web va hosta două componente:

1. partea de frontend ce constă într-o aplicație React (.js/Native) ce are ca environment un engine de JavaScript. Aplicația frontend va încărca în DOM diferite pagini în funcție de path-ul request-ului HTTP primit de la utilizator. Pentru datele dinamice, se va folosi modulul Axios, ce face cereri HTTP către API-ul nostru.

2. partea de backend este un REST API construit cu Django, ce va gestiona cererile primite de la clientul de Axios, întorcând răspunsul în format JSON, după serializarea și comunicarea cu DB.

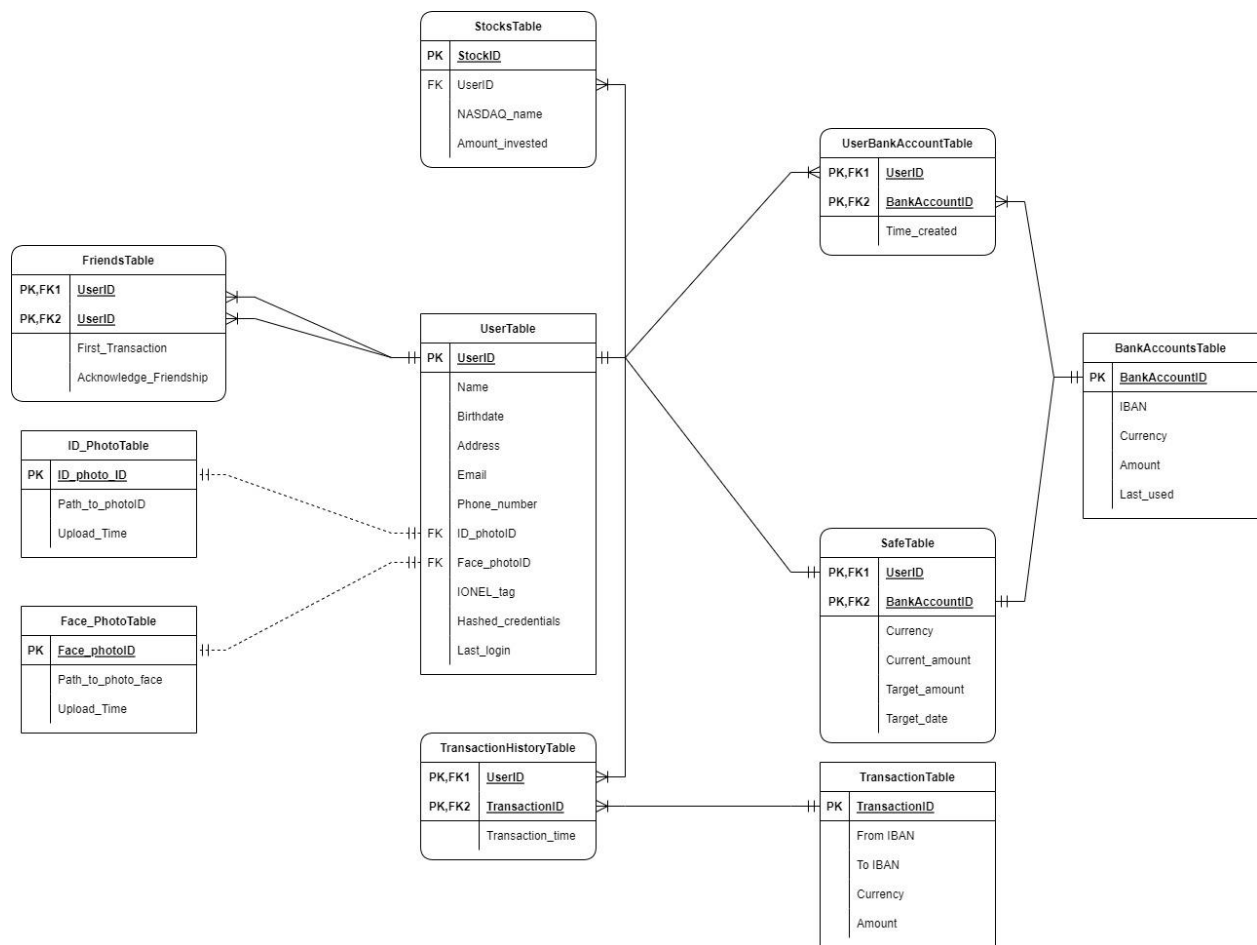
- pentru persistența datelor avem un server ce hostează o instanță a SQLite.



3.4 Managementul datelor persistente

Baza de date este alcătuită din 10 tabele: UserTable, StocksTable, FriendsTable, ID_PhotoTable, Face_PhotoTable, TransactionHistoryTable, TransactionTable, SafeTable, UserBankAccountTable, BankAccountTable.

- 1) UserTable: tabelă ce stochează datele personale ale unui utilizator IONEL Romania, dar și date precum ultima logare.
- 2) ID_PhotoTable: tabelă ce stochează poza cărții de identitate a unui utilizator.
- 3) Face_PhotoTable: tabelă ce stochează selfie-ul unui utilizator.
- 4) FriendsTable: o tabelă de legătură ce stochează prietenii dintre utilizatori.
- 5) StocksTable: o tabelă ce reține informații despre o acțiune a utilizatorului. Aceste informații sunt transmise mai departe unui API specializat în tranzacții de acțiuni.
- 6) BankAccountTable: o tabelă ce reține informații despre un cont bancar al unui utilizator.
- 7) UserBankAccountTable: o tabelă de legătură între utilizatori și conturi. Astfel, fiecare utilizator poate avea mai multe conturi.
- 8) SafeTable: o tabelă ce reține date despre seiful unui utilizator;
- 9) TransactionTable: o tabelă ce reține informații despre o tranzacție făcută.
- 10) TransactionHistoryTable: o tabelă de legătura ce reține informații despre istoricul tranzacțiilor al unui utilizator.



3.5 Controlul accesului utilizatorilor la sistem

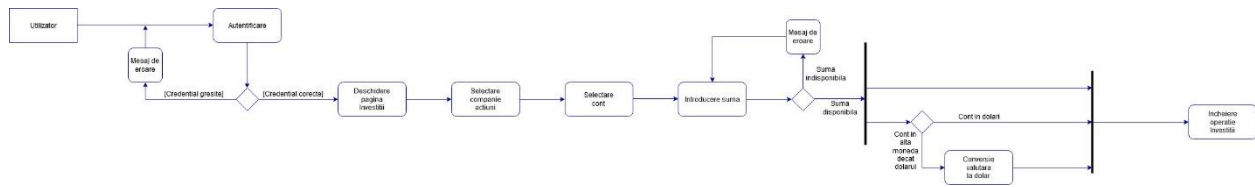
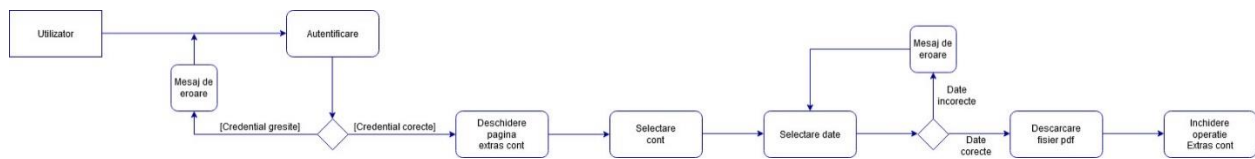
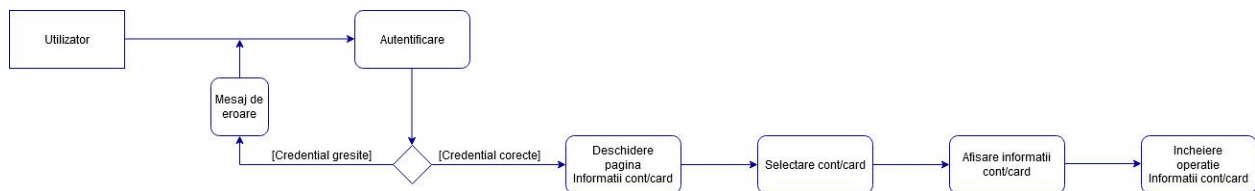
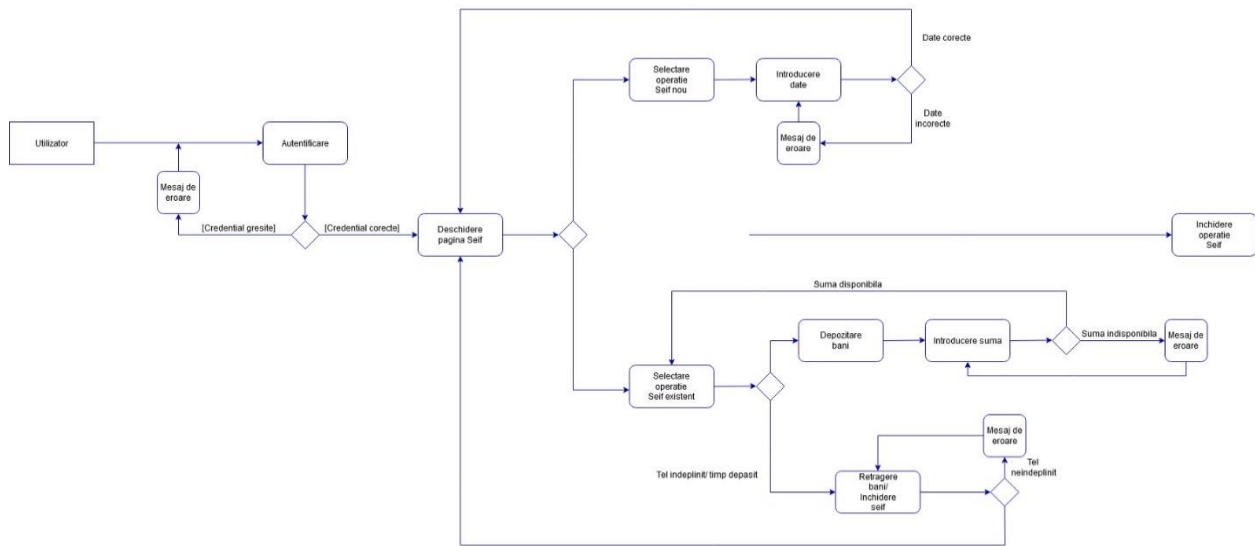
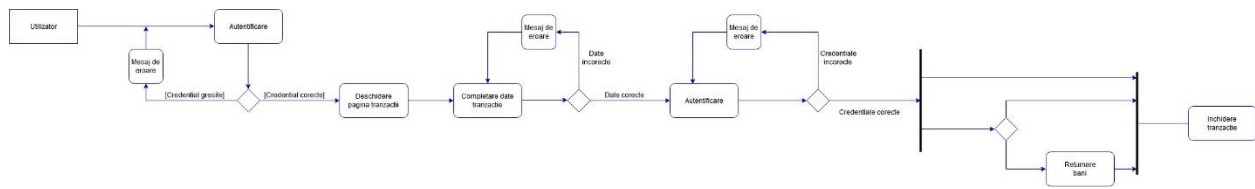
Autentificarea utilizatorilor în sistem se va face prin introducerea credențialelor, precum și introducerea unui cod primit pe telefon.

Fiecare categorie de utilizatori va putea efectua anumite operații, ele fiind restricționate de către backend:

- Utilizator obișnuit: autentificare, tranzacții, sold, schimb valutar, cont de investiții, peer-to-peer transfer, plăți programate, seif, analiză venituri/cheltuieli;
- Administrator: autentificare, adăugare/eliminare client, modificare curs valutar, deschidere conturi, obținere statistici/raport sistem bancar;

3.6 Fluxul global al controlului

Sistemul nostru așteaptă întotdeauna o intrare a utilizatorului. De exemplu, întrucât proiectul nostru este unul bazat pe web, serverul web așteaptă solicitările de la browserul web. La primirea unei cereri, serverul web procesează și îl trimite către pagina web corespunzătoare rezultând un flux de control bazat pe evenimente.



3.7 Condițiile limită (cazurile de utilizare limită)

- Nesincronizarea dintre ceasul serverului și ceasul telefonului utilizatorului;
- Eșuarea repetată a autentificării unui utilizator va rezulta în blocarea contului respective;
- Defectarea dispozitivelor de stocare: datele vor fi stocate pe mai multe dispozitive pentru a asigura redundanță.

Planificarea implementării proiectului

<i>ID</i>	<i>Task</i>	<i>Asignat</i>	<i>Sprint</i>	<i>Estimare</i>
<i>G-1</i>	Setup environment	Toți membrii	1	4h
<i>F-1</i>	Pagină autentificare	GP + GI	1	6h
<i>F-2</i>	Pagină principală	GP + GI	1	12h
<i>F-3</i>	Pagină conturi/carduri	GP + GI	2	9h
<i>F-4</i>	Pagină schimb valutar	GP + GI	2	3h
<i>F-5</i>	Pagină tranzacții	GP + GI	2	6h
<i>F-6</i>	Pagină investiții	GP + GI	2	6h
<i>F-7</i>	Pagină analiză financiară	GP + GI	3	6h
<i>F-8</i>	Pagină programare plăți	GP + GI	3	6h
<i>F-9</i>	Pagină cont economii	GP + GI	3	6h
<i>M-1</i>	Rutare React + API calls	GP + GI	1	9h
<i>B-1</i>	Modelare intrări BD	CC + SD	1	6h
<i>B-2</i>	Serializare date	CC + SD	1	6h
<i>B-3</i>	Implementare REST API	CC + SD	1	12h
<i>B-4</i>	Autentificare	CC + SD	1	9h
<i>B-5</i>	Generare PDF	CC + SD	3	9h
<i>B-6</i>	Implementare P2P NFC	Toți membrii	3	12h
<i>B-7</i>	Rutare URL	CC + SD	1	6h
<i>B-8</i>	Securizare Aplicație	CC + SD	3	12h+
<i>B-9</i>	Fetch schimb valutar	CC + SD	2	6h
<i>B-10</i>	Fetch stock market info	CC + SD	2	9h
<i>G-2</i>	Docs + Coding style	Toți membrii	3	9h

Legendă:

CC – Cosmin Criciu

GI – Gabi Ichim

GP – Gabi Poalelungi

SD – Stefan Dumitriu