*Sistem informatic de tip online banking*

Grupa C-112D

Std. Cap. Negoescu Elena - Camelia

Std. Cap. Badea Alexandru - Gabriel

**Cuprins**

[Capitolul 1: Introducere 3](#_wzfgdic4mcc8)

[1.1 Scopul proiectului 3](#_ej1g81xz6osp)

[1.2 Lista definitiilor 3](#_7edrtk8fh5tv)

[1.3 Modificări 4](#_fe5uoz4hwp14)

[1.4 Structura DCS 4](#_weksi8wpse3m)

[Capitolul 2: Descrierea generală a produsului software 4](#_bi8jl8syctzx)

[2.1 Descrierea produsului software 4](#_eq9qvd8xqscr)

[2.2 Detalierea platformei HW/SW 5](#_km0ig6k1l46d)

[Capitolul 3: Detalierea cerințelor specifice 6](#_emwni0niia8g)

[1. Crearea contului de utilizator 6](#_3cn8yqx9wwo1)

[2. Autentificare 8](#_w66zxobxu3xk)

[a) Autentificarea utilizatorilor cu nume de utilizator și parola 8](#_4834ckk0xzrn)

[b) Implementarea opțiunii de „Parola uitată” 8](#_xlfmzjbl78o5)

[3. Gestionarea conturilor 9](#_b3kqtla9ivh3)

[a) Inchiderea conturilor 9](#_1n09a7eec6yv)

[b) Modificarea detaliilor contului 9](#_kfejeqstniq)

[4. Tranzactii 10](#_jq8hh1e0p3z1)

[a) Transferuri de bani între cardurile proprii 10](#_b4qq503k8un2)

[b) Transferuri de bani către alți beneficiari(interbancare) 11](#_1wtchoc2t72v)

[c) Conversie valutara 11](#_mgfa3muy8yb)

[5. Carduri bancare 12](#_bb8p8wq6f9bz)

[a) Vizualizarea și gestionarea cardurilor bancare asociate contului 12](#_3i9j26ri1old)

[b) Crearea unui nou card 12](#_je6aoz7vmuji)

[c) Activarea/dezactivarea cardurilor 12](#_sm2kcik7iegy)

[d) Blocarea cardurilor în caz de pierdere sau furt 12](#_j4u7ykc97585)

[6. Istoric tranzactii 13](#_pb0vzwxlu8ws)

[a) Vizualizarea detaliilor tranzacțiilor efectuate 13](#_63m7oadzm10l)

[b) Filtrarea tranzacțiilor după anumite criterii(perioada, tip de tranzactie, etc) 13](#_6zmetga84stg)

# **Capitolul 1:** Introducere

## 1.1 Scopul proiectului

Proiectul nostru are ca scop dezvoltarea unui sistem de online banking care să ofere clienților o modalitate convenabilă și eficientă de a-și gestiona finanțele online. Sistemul va permite clienților să efectueze diverse operațiuni bancare, cum ar fi transferuri de bani, conversii valutare, plata facturilor și vizualizarea istoricului tranzacțiilor, de oriunde și la orice oră, fără a fi nevoiți să se deplaseze fizic la o sucursală bancară.

## 1.2 Lista definitiilor

| **ABREVIERE** | **TERMEN** |
| --- | --- |
| DCS | Document cu cerințe software |
| HW | Hardware |
| SW | Software |
| JSON | JavaScript Object Notation |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| API | Application Programming Interface |
| GUI | Graphical User Interface |
| IBAN | International Bank Account Number |

## 1.3 Modificări

| **Numărul modificării** | **Data** | **Observații** |
| --- | --- | --- |
| 1. | 29.03.2024 | Varianta inițială a proiectului |
|  |  |  |
|  |  |  |

## 1.4 Structura DCS

În documentul cu cerințe software se va prezenta în capitol 2 descrierea generală a sistemului informatic de tip online banking ca mai apoi, în capitolul 3, să fie detaliate toate cerințele funcționale. De asemenea, tot în capitolul 2, se vor prezenta elementele de hardware și software pentru care produsul este dezvoltat.

# **Capitolul 2:** Descrierea generală a produsului software

## 2.1 Descrierea produsului software

Produsul software constă într-un sistem informatic de tip online banking cu arhitectura client-server, dezvoltat în limbajul de programare C++, folosind, de asemenea, baze de date MySql, sockets și Qt. Acest sistem va include patru componente principale: aplicația client cu interfață grafică (GUI), server de frontend(proxy) care servește ca intermediar între client și serverul de backend, server de backend care gestionează operațiunile, tranzacțiile și lucrul cu baza de date și baza de date care stochează datele utilizatorilor, tranzacțiile, informațiile despre conturi.

Aplicația client va oferi o interfață grafică intuitivă și ușor de utilizat pentru clienți, permițându-le să efectueze diverse operațiuni bancare precum:

* Autentificare: Clienții vor putea să se autentifice în sistem folosind un nume de utilizator și o parolă;
* Gestionare conturi: Utilizatorii vor putea să vizualizeze și să gestioneze conturile lor bancare, inclusiv să consulte soldurile, să efectueze depuneri sau retrageri de fonduri și să acceseze detalii despre tranzacții;
* Transferuri de bani: Clienții vor putea să efectueze transferuri de bani între conturile lor sau către alți utilizatori ai sistemului;
* Conversii valutare: Sistemul va oferi suport pentru conversii valutare la rate de schimb actualizate, permițând clienților să efectueze tranzacții în diferite monede;
* Istoric tranzacții: Utilizatorii vor putea să acceseze un istoric detaliat al tranzacțiilor lor, pentru a monitoriza și verifica toate operațiunile efectuate în conturile lor.

Serverul de frontend(proxy) va primi cererile de la client și le va redirecționa către serverul de backend. Astfel, serverul proxy va funcționa ca un intermediar între client și server. Acesta va ascunde adresa IP a serverului de backend de client, sporind astfel securitatea.

Serverul de backend va gestiona toate operațiunile bancare și datele asociate, inclusiv:

* Autentificare și autorizare: Serverul va gestiona procesul de autentificare și autorizare a utilizatorilor, asigurându-se că doar utilizatorii autorizați au acces la funcționalitățile sistemului;
* Gestionare conturi: Serverul va gestiona toate detaliile legate de conturile bancare ale clienților, inclusiv informații despre solduri, tranzacții și alte detalii relevante;
* Procesarea tranzacțiilor: Serverul va fi responsabil pentru procesarea și executarea tranzacțiilor financiare, asigurându-se că acestea sunt realizate în mod sigur și eficient.

Toate aceste operațiuni sunt detaliate în [capitolul 3](#_emwni0niia8g).

## 2.2 Detalierea platformei HW/SW

Cerințe minime pentru ca aplicația să poată fi rulată:

* Sistem de operare: Windows 7 sau mai recent
* Procesor:
  + Intel Core i5 sau mai recent
  + AMD Ryzen 3 sau mai recent
* Memorie: minimum 2 GB RAM (recomandat 4 GB RAM)
* Spațiu de stocare: 3 GB disponibil

Alte cerințe:

* Conexiune la internet stabilă

# **Capitolul 3:** Detalierea cerințelor specifice

### Crearea contului de utilizator

Interfată client: Clientul accesează aplicația și alege opțiunea „Creare cont” din meniul principal. Acesta completează formularul de înregistrare cu datele personale(nume, adresă, CNP, număr de telefon) și datele de autentificare(username și parolă), apoi confirmă crearea contului prin apăsarea butonului ”Creare cont”.

Există anumite constrângeri privind crearea contului și anume:

* username-ul poate conține maximum 30 de caractere(litere si cifre, \_, .), este unic și nu poate fi schimbat;
* parola trebuie să conțină minimum 8 caractere și maximum 16, dintre care primul caracter nu poate fi o cifră sau caracter special. De asemenea, trebuie să conțină obligatoriu cel puțin un caracter special, o literă mare și o cifră;

La crearea contului, clientul trebuie să răspundă și la o serie de întrebări de securitate care îl vom ajuta în cazul blocării contului. Aceste întrebări sunt:

- Care este numele de fată al mamei tale?

- Care este orașul natal al tatălui tău?

- Care este prima școală la care ai mers?

- Care a fost prima ta mașină?

- Care este numele animalului tău de companie preferat?

- Alegeți o frază de securitate.

Clientul trimite o cerere către serverul proxy astfel:

* Clientul utilizează o bibliotecă de rețea pentru a crea un socket TCP/IP pentru a stabili o conexiune cu serverul proxy. Acest socket este folosit pentru a trimite cererea către serverul proxy și pentru a primi răspunsul de la acesta.
* Clientul trimite o solicitare către proxy.
* Solicitarea conține datele introduse de client (nume, adresă de email, număr de telefon, username, parolă, răspunsurile la întrebările de securitate) și informații despre conexiunea clientului (adresa IP). Toate acestea vor fi criptate folosind SHA-256, iar apoi vor fi serializate in JSON si transmise mai departe. În structura JSON vom avea și tipul de cerere.
* Serverul proxy ascultă și acceptă conexiuni de la clienți pe un anumit port. Odată ce se stabilește o conexiune cu un client, serverul proxy creează un nou socket pentru a comunica cu clientul respectiv și pentru a primi și a trimite date către și de la acesta.
* Serverul proxy creează un alt socket TCP/IP pentru a stabili o conexiune cu serverul backend. In acest fel, trimite cererea mai departe serverului backend.
* Similar cu serverul proxy, serverul backend ascultă și acceptă conexiuni de la serverul proxy pe un anumit port. Odată ce se stabilește o conexiune, serverul backend creează un socket separat pentru a comunica cu serverul proxy și pentru a primi și a trimite date către și de la acesta.

Server backend: Serverul procesează solicitarea astfel:

* Datele trimise sub forma de JSON sunt deserializate.
* Serverul backend verifică dacă username-ul ales de client este disponibil în baza de date, (prin conectare la baza de date și căutarea username-ului prin interogări). În cazul în care nu este disponibil, va trimite înapoi la client o înștiințare.
* Altfel, serverul stochează datele clientului în baza de date.
* În momentul creării unui cont, serverul va crea un card de debit care va primi un IBAN, un număr de card(un set unic de 16 cifre, ID-ul cardului), CVC-ul și un PIN generat automat. Aceste date vor fi introduse în baza de date. În tabela specifica cardurilor va exista și un câmp prin care se va indica starea contului(activ/inactiv). Starea contului va fi setată ca activ.
* Serverul trimite un mesaj de confirmare către proxy, iar server-ul proxy trimite această confirmare clientului, împreună cu PIN-ul.
* Clientul este informat dacă contul a fost creat cu succes sau dacă au apărut erori printr-un mesaj afișat pe ecran.
* Clientul are posibilitatea de a-și schimba PIN-ul, după autentificarea în aplicație. În acest sens, după apăsarea acestei opțiuni în interfață grafică, utilizatorului i se va deschide o fereastră în care își va putea introduce noul PIN. După, clientul va trimite o cerere către server prin conexiune TCP/IP ce va contine: tipul de cerere(„Schimbare PIN”), IBAN-ul cardului pentru care se schimbă PIN-ul, username-ul și noul PIN criptat SHA-256.
* Serverul va primi această cerere, va interoga baza de date pentru a găsi username-ul și cardul specific după IBAN, apoi va actualiza PIN-ul.
* La final, va trimite un mesaj prin care îl informează pe utilizator că PIN-ul a fost schimbat.

După crearea contului, utilizatorul se va putea autentifica în aplicație, unde va avea acces la cardul de debit, acesta fiind vizibil în meniul aplicației. În meniul principal pentru fiecare card vor fi vizibile: soldul cardului, opțiunea de a accesa istoricul tranzacțiilor, opțiunea de a face o plata nouă sau opțiunea de curs valutar.

### Autentificare

#### Autentificarea utilizatorilor cu nume de utilizator și parola

* Clientul introduce numele de utilizator și parola în câmpurile corespunzătoare ale ferestrei de autentificare.
* Clientul utilizează o bibliotecă de rețea pentru a stabili o conexiune TCP/IP cu server-ul, procedeu similar cu cel de la crearea contului.
* Clientul trimite o solicitare către proxy care conține numele de utilizator și parola criptate folosind SHA-256, transformate în format JSON, și informații despre conexiunea clientului (adresa IP). În structura JSON se va specifica tipul de cerere, în acest caz, autentificare.
* Serverul proxy primește solicitarea și o distribuie mai departe către serverul backend.
* Serverul procesează solicitarea începând cu deserializarea datelor primite sub forma JSON
* Serverul backend caută în baza de date numele de utilizator și verifică dacă parola corespunde acestuia.
* Dacă autentificarea este reușită, serverul trimite un mesaj de confirmare către client, prin proxy.
* Dacă autentificarea eșuează, serverul trimite un mesaj de eroare către client, prin proxy.
* Clientul este informat dacă autentificarea a fost sau nu reușită printr-un mesaj afișat pe ecran și i se permite accesul în aplicație.

#### Implementarea opțiunii de „Parola uitată”

* Interfața clientului îi pune la dispoziție în pagina de autentificare și un buton cu funcționalitatea de „Parola uitată”.
* Utilizatorului îi sunt solicitate următoarele date pentru a se verifica identitatea acestuia: adresa de email folosită la crearea contului, și răspunsurile la întrebările de securitate oferite, de asemenea, la crearea contului.
* După completarea acestor date, se trimite o cerere către serverul proxy pentru a iniția procesul de resetare a parolei.
* Serverul proxy trimite mai departe această cerere către server-ul de backend.
* Serverul primește cererea de resetare a parolei și o procesează astfel: verifică informațiile furnizate cu cele din baza de date.
* Dacă identitatea utilizatorului este confirmată, serverul generează un răspuns către client prin care i se permite să își schimbe parola.
* Clientul primește acest răspuns prin proxy și i se deschide o nouă interfață prin care i se cere să introducă username-ul și noua parolă, iar apoi să o confirme.
* Clientul trimite noua parolă criptată și username-ul către server prin proxy pentru a înregistra schimbarea.
* După ce primește noua parolă, serverul se conectează la baza de date și în funcție de username, actualizează parola utilizatorului, iar apoi trimite un mesaj de confirmare către client pentru a indica reușita schimbării parolei.

### Gestionarea conturilor

#### Inchiderea conturilor

* Utilizatorul are posibilitatea în interfață de a-și închide contul.
* Clientul trimite o cerere către server, prin proxy, ce va conține, username-ul, parola criptată și tipul de cerere, în acest caz, „Închidere cont”.
* Serverul primește această cerere, verifică în baza de date identitatea utilizatorului și dacă este confirmată atunci toate datele corespunzătoare contului vor fi șterse din baza de date.
* Serverul trimite răspunsul către client prin care i se transmite reușita cererii(apare o fereastră în interfață prin care îi se va comunica „Contul a fost închis cu success!”)
* Clientului i se va deschide fereastra de autentificare.
* Dacă identitatea nu este confirmată sau închiderea contului nu este reușită, serverul va transmite către client un mesaj corespunzător și i se va da posibilitatea de a reîncerca.

#### Modificarea detaliilor contului

* În interfață, utilizatorului i se dă opțiunea de a-și modifica datele, precum adresa, numele, prenumele, numărul de telefon, adresa de e-mail.
* Clientul trimite o cerere de tipul ”Modificare date cont”, împreună cu datele pe care dorește să le modifice și username. Toate aceste date vor fi trimise sub formă JSON
* Proxy-ul primește această cerere și o trimite mai departe către server.
* Server-ul deserializează această cerere, se conectează la baza de date și actualizează datele corespunzătoare, apoi trimite un răspuns către client prin care îl informează că modificarea a avut loc.

### Tranzactii

#### Transferuri de bani între cardurile proprii

* În interfața utilizatorului, sunt afișate conturile utilizatorului, iar utilizatorul selectează unul dintre cardurile din listă.
* Utilizatorul introduce manual IBAN-ul cardului destinație într-un câmp text, apoi introduce suma pe care dorește să o transfere către cardul destinație într-un câmp de introducere a sumei. Apoi apasă butonul corespunzător pentru a finaliza tranzacția.
* Clientul trimite o cerere de tip „Tranzacție între cardurile aceluiași cont”, ce va contine IBAN-ul sursa, IBAN-ul destinație și suma ce se dorește a fi transferată, toate serializate în format JSON.
* Se realizează o conexiunea print-un socket TCP/IP cu serverul proxy, care la rândul lui va realiza o conexiunea prin socket TCP/IP cu server-ul backend, după ce primește cererea de la client.
* După ce clientul trimite detaliile tranzacției către server, acesta validează informațiile pentru a verifica atât dacă cardurile sursă și destinație sunt valide și aparțin aceluiași utilizator, cât și dacă suma ce urmează să fie transferată este mai mică decât soldul curent al cardului sursă, prin intermediul unor interogări SQL. Dacă cardurile sunt valide și soldul este suficient, serverul continuă procesarea tranzacției; în caz contrar, se returnează un mesaj de eroare către client.
* Procesarea tranzacției constă în actualizarea soldurilor curente ale celor doua carduri. Acest lucru se realizează prin intermediul unor interogari SQL in baza de date prin care din soldul cardului sursă se scade suma cerută, iar în soldul cardului destinație se va adaugă respectiva suma.
* După finalizarea cu succes a tranzacției, serverul va trimite prin proxy un răspuns către client pentru a-l notifica ca transferul de bani a fost efectuat cu succes.

#### Transferuri de bani către alți beneficiari(interbancare)

* Transferul de bani între doi utilizatori diferiți are un proces similar cu cel de mai sus, cu următoarele mențiuni:
* Utilizatorul care face tranzacția va trebui să introducă în formularul de efectuare a transferului, numele complet al clientului destinație, cât și IBAN-ul cardului în care vrea să transfere banii. Astfel, cererea pe care o trimite clientul către server va conține pe lângă username-ul clientului sursă și IBAN-ul cardul din care se face transferul și numele complet al clientului destinație, precum și IBAN-ul în care se dorește a se transfera banii. Toate aceste date vor fi serializate în format JSON și transmise prin sockets TCP/IP la proxy, iar apoi tot prin sockets TCP/IP către server.
* După aceea, serverul realizează confirmarea identității celor 2 utilizatori, prin conectarea la baze de date MySQL. Dacă ambele identități sunt confirmate, atunci serverul realizează tranzacția în mod similar pasului anterior.
* După finalizarea cu succes a tranzacției, serverul trimite 2 răspunsuri: unul către clientul sursă și unul pentru clientul destinație pentru a-i informa de efectuarea tranzacției, fapt ce se va regăsi atât în istoricul tranzacțiilor ambilor clienți, cât și în soldurile curente.

#### Conversie valutară

* Utilizatorul introduce suma de bani și valutele sursă și destinație în interfață, apoi apasă un buton pentru a iniția conversia
* Clientul trimite o cerere către serverul de backend care include informații despre suma de bani și valutele sursă și destinație, precum și tipul de cerere, în acest caz ”Conversie valutară”.
* Serverul de backend primește această cerere și apelează funcția de conversie.
* Va exista o funcție de conversie care primește suma, valutele sursă și destinație ca parametrii.
* Se utilizează o bibliotecă de rețea(Qt Network) pentru a stabili o conexiune TCP/IP cu serverul API(de exemplu, putem folosi Exchange Rates API)
* Se construiește adresa URL a API-ului și se utilizează o biblioteca HTTPS pentru a trimite o solicitare GET către API.
* Se trimite solicitarea GET construită la serverul API.
* Se primește răspunsul JSON de la API și se parsează folosind o bibliotecă JSON, iar apoi se extrage rata de schimb valutar. Se calculează suma convertită prin înmulțirea sumei inițiale cu rata de schimb.
* La final se afișează suma convertită în interfața utilizatorului.

### Carduri bancare

#### Vizualizarea și gestionarea cardurilor bancare asociate contului

* În interfața utilizatorului îi sunt puse la dispoziție cardurile pe care le deține. Fiecare card va avea afișat soldul curent, IBAN-ul, și butoane pentru a vedea istoricul tranzacțiilor și pentru a realiza o plata nouă.
* La fiecare interacțiune cu vreun buton, clientul va trimite o cerere corespunzătoare către server, ce va trece prin proxy. Serverul, de asemenea, se va conecta la baza de date MySQL și va trimite înapoi către client un răspuns. Toate aceste lucruri se vor realiza prin sockets TCP/IP, iar cererea va trebui, obligatoriu, să conțină tipul de cerere, ip sursă, ip destinatie și datele specifice tipului de cerere.

#### Crearea unui nou card

* După ce se autentifică în aplicație utilizatorul are opțiunea de a-și crea/adăuga un alt card.
* Utilizatorul introduce informațiile necesare pentru noul card și anume PIN-ul și tipul de card(card de debit sau card de credit), apoi aplicația client trimite o cerere(ce contine tipul de cerere, tipul cardului, username și PIN-ul criptat SHA-256) printr-un socket TCP/IP către proxy. Serverul Proxy primește această cerere și după crează alt socket TCP/IP prin care trimite cererea către serverul backend.
* Serverul primește solicitarea de la proxy și generează un nou număr de card, un nou IBAN, data expirarii cardului, apoi adaugă toate aceste informații în baza de date, setând starea contului ca activ.
* Serverul trimite către client un mesaj de confirmare prin care îl informează de faptul că cererea s-a realizat cu succes, iar noul card va fi vizibil în pagina principală a aplicației

#### Activarea/dezactivarea cardurilor

#### Blocarea cardurilor în caz de pierdere sau furt

* Utilizatorul accesează aplicația de online banking și se autentifică, iar în secțiunea de gestionare a cardurilor, acesta poate selecta opțiunea pentru activarea, dezactivarea sau blocarea unui card.
* Utilizatorul selectează cardul pe care dorește să îl activeze/dezactiveze/blocheze și trimite solicitarea către proxy(printr-o conexiunea TCP/IP), solicitare care va conține username-ul, IBAN-ul și tipul de cerere.
* Serverul proxy primește solicitarea și o trimite mai departe(conexiunea TCP/IP) către serverul backend.
* Serverul primește solicitarea și în funcție de aceasta actualizează starea cardului asociat contului în baza de date.
* În cazul în care solicitarea este de tip „Blocare card”, cardul și datele aferente acestuia sunt șterse din baza de date.
* Serverul trimite un răspuns cu rezultatul solicitării către proxy, iar proxy-ul trimite în continuare acest răspuns către client. În interfața grafică va apărea un mesaj cu starea contului(activ/inactiv/blocat). În cazul ștergerii cardului, acesta nu se va mai regăsi în lista cardurilor asociate contului utilizatorului.

### Istoric tranzactii

#### Vizualizarea detaliilor tranzacțiilor efectuate

* Utilizatorul apasă butonul specific pentru a vizualiza istoricul tranzacțiilor. Astfel, acesta trimite o cerere către proxy, iar proxy-ul redirecționează această cerere către server prin sockets TCP/IP. Această cerere va avea specific tipul de cerere și anume „Istoric tranzacții”.
* După ce serverul a primit cererea, se conectează la baza de date pentru a realizat interogari SQL în vederea extragerii datelor corespunzătoare.
* Serverul împachetează datele despre tranzacții într-un mesaj, în format JSON, pe care îl trimite înapoi la client, prin server-ul intermediar proxy.
* Aplicația clientul primește răspunsul de la server prin conexiune TCP/IP, iar apoi datele sunt prelucrate și afișate în interfața de utilizator.

#### Filtrarea tranzacțiilor după anumite criterii(perioada, tip de tranzactie, etc)

* Realizarea filtrării istoricului tranzacțiilor după anumite criterii se realizează similar cu procedeul anterior cu următoarele mențiuni:
* Cererea trimisă de aplicația client către server va mai conține și criteriul după care se realizează filtrarea.
* Interogarea pe care serverul o face în baza de date se va face conform criteriului specificat, iar aceste date vor fi trimise înapoi către client.