

SISTEM DE GESTIUNEA BAZEI DE DATE BANCARE

Studenti: Husar Andrei Dumitru

Harsan Simon Filip

Pantiru Cosmin Narcis

Cuprins

- I. Introducere**
- II. Importanta temei in contextul actual si obiectivele proiectului**
- III. Cercetare bibliografica 4.1 Sisteme informatice bancare 4.2 Tehnologii
 utilizate**
- IV. Arhitectura sistemului 5.1 Module principale 5.2 Diagrama ER 5.3
 Diagrame UML**
- V. Rezultate obtinute**
- VI. Concluzii**

I. Introducere

În era digitală, sistemele informatice au devenit coloana vertebrală a oricărei organizații, iar domeniul bancar nu face excepție. Necesitatea automatizării proceselor financiare, creșterea cerințelor în materie de securitate și accesibilitate, precum și evoluția comportamentului utilizatorilor impun dezvoltarea unor aplicații moderne, rapide și intuitive.

Proiectul de față își propune să creeze un sistem bancar simplificat, dar complet, care să reproducă funcționalități uzuale dintr-o bancă reală: crearea și gestionarea conturilor, înregistrarea și plata împrumuturilor, transferul de fonduri, autentificarea utilizatorilor și administrarea acestora. Alegerea limbajului Python și a platformei SQL Server se justifică prin accesibilitate, suport extins și capacitatea de integrare rapidă între interfața grafică și baza de date.

Această documentație descrie etapele realizării proiectului, de la justificarea alegerii temei și obiective, la implementare și rezultate obținute.

II. Importanța temei în contextul actual

În contextul transformării digitale accelerate și al nevoii crescânde de automatizare în instituțiile financiare, sistemele informatice bancare joacă un rol esențial în gestionarea eficientă a datelor clienților și a operațiunilor financiare. Digitalizarea serviciilor bancare nu este doar o tendință, ci o necesitate care răspunde cerințelor utilizatorilor moderni: acces rapid, securitate sporită, transparență și personalizare. Proiectul de față propune dezvoltarea unui sistem software care reproduce funcționalitățile esențiale ale unui sistem bancar, utilizând limbajul de programare

Python, SQL Server pentru gestiunea bazei de date si o interfata grafica intuitiva bazata pe Tkinter.

Acest proiect raspunde necesitatii de a invata, experimenta si simula un mediu bancar complet, oferind utilizatorilor posibilitatea de a vizualiza conturi, efectua tranzactii, solicita si rambursa imprumuturi, toate acestea printr-o aplicatie accesibila si prietenoasa.

Obiectivele proiectului

Proiectul urmareste dezvoltarea unei aplicatii modulare si functionale, care sa permita:

- Interactiunea utilizatorilor cu sistemul printr-o interfata grafica intuitiva.
- Stocarea, accesarea si manipularea datelor bancare printr-o baza de date relationala SQL Server.
- Gestionarea conturilor clientilor, a imprumuturilor, transferurilor si sesiunilor de autentificare.
- Separarea clara a responsabilitatilor prin module distincte pentru interogari, inserari, actualizari si stingeri de date.

III. Cercetare bibliografica

In aceasta sectiune sunt analizate atat sisteme informatice bancare consacrate, cat si tehnologiile utilizate in cadrul proiectului.

Sisteme informatice bancare

In literatura de specialitate, sistemele informatice bancare sunt descrise ca fiind coloana vertebrala a oricarei institutii financiare moderne. Acestea trebuie sa ofere securitate, fiabilitate, scalabilitate si usurinta in utilizare. Exemple de astfel de sisteme includ Finacle (de la Infosys), T24 (Temenos) sau Oracle FLEXCUBE, care sunt implementate pe scara larga in banci comerciale. Articolele din reviste de specialitate, precum *Journal of Financial Services Technology* sau *ACM Transactions on Information Systems*, subliniaza importanta modularitatii si a auditabilitatii in proiectarea acestor sisteme.

Tehnologii utilizate

- **Python:** Python este un limbaj de programare de nivel înalt, interpretat, cu o sintaxă clară și concisă, ceea ce îl face ideal pentru începători dar și pentru dezvoltatori avansați. Este un limbaj multiparadigma, susținând programarea orientată pe obiect, funcțională și imperativă. În contextul acestui proiect, Python este utilizat pentru dezvoltarea interfeței grafice, conectarea la baza de date și gestionarea logică a aplicației.
- **SQL Server (SSMS):** Microsoft SQL Server este un sistem de gestiune a bazelor de date relationale dezvoltat de Microsoft. Este destinat aplicațiilor de business care necesită stocare sigură și scalabilă a datelor. SQL Server Management Studio (SSMS) este un instrument puternic care permite dezvoltatorilor și administratorilor de baze de date să creeze, să gestioneze și să interogheze baze de date. În proiectul de față, SSMS este folosit pentru definirea tabelor, a relațiilor și pentru gestionarea directă a conținutului bazei de date. Reprezintă o soluție profesională pentru gestiunea datelor relationale, oferind securitate, integritate și performanță. Este ideal pentru dezvoltare în medii Windows, având o interfață de administrare prietenoasă.

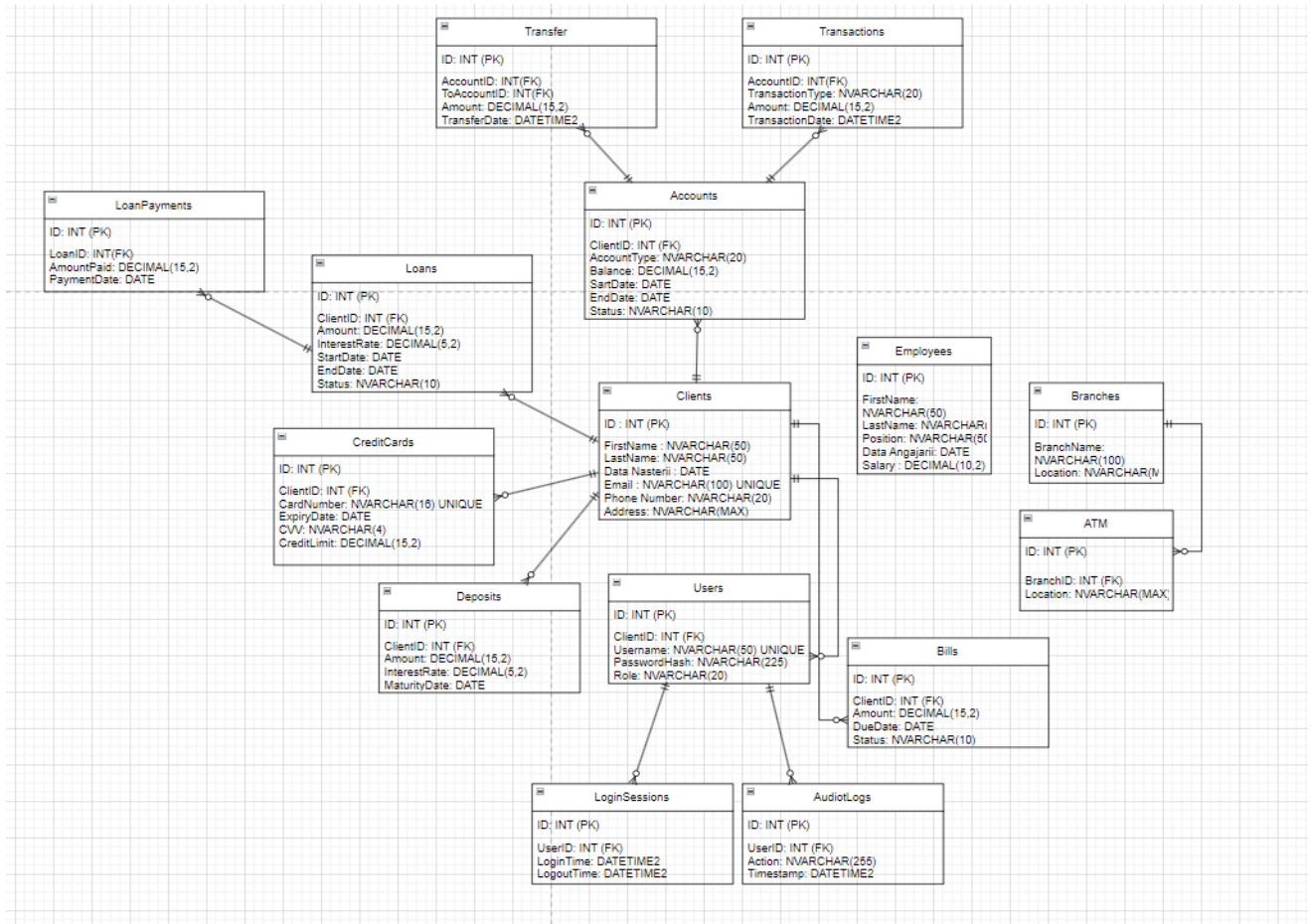
IV. Arhitectura sistemului

Sistemul este compus din mai multe module funcționale:

- **gestiune.py:** Contine logica de conectare la baza de date și definirea structurii pentru entități precum clienți, împrumuturi, conturi, utilizatori, sesiuni de login etc.
- **insert.py:** Contine funcții dedicate pentru introducerea de date în toate tabelele relevante (clienți, conturi, împrumuturi, plăți etc.).
- **get.py:** Oferă metode pentru extragerea datelor din baza de date, utilizate în interfața grafică.
- **delete.py:** Permite ștergerea datelor din tabele, după identificator.
- **update.py:** Contine metode generice pentru actualizarea valorilor din înregistrări existente.
- **bankloginUI.py:** Interfața simplificată de login pentru testarea funcționalității de autentificare, cu validare hardcodată.
- **UI_simplu.py:** Module de interfață grafică, ce oferă o experiență completă atât pentru utilizatorii obișnuiți, cât și pentru administratori. Funcționalitățile includ gestionarea conturilor, împrumuturilor, transferurilor și vizualizarea rapoartelor.

Diagrams UML

- **Use Case Diagram:** Definirea actorilor principali (Client, Administrator) si a interactiunilor posibile (login, solicitare imprumut, vizualizare cont, transfer de fonduri, administrare utilizatori etc.).
- **Diagrama de activitate:** Fluxul unui client – de la autentificare pana la efectuarea unui transfer sau solicitarea unui imprumut.



V. Rezultate obtinute

La finalizarea proiectului, s-a obtinut o aplicatie functionala care permite:

- Autentificarea utilizatorilor si redirectionarea acestora catre un panou corespunzator (admin/client).
- Vizualizarea conturilor si soldurilor disponibile.
- Efectuarea de transferuri intre conturi.
- Solicitarea de imprumuturi si inregistrarea platilor aferente.
- Gestionarea datelor personale si a utilizatorilor (in cazul administratorului).
- Stergerea si actualizarea inregistrarilor prin modulele `delete.py` si `update.py`, ceea ce demonstreaza capacitatea completa CRUD a aplicatiei.

Interfata grafica ofera o experienta intuitiva, iar baza de date robusta asigura integritatea si consistenta datelor.

VI. Concluzii

Proiectul propus demonstreaza ca este posibila realizarea unui sistem bancar simplificat, complet functional, folosind tehnologii open-source si resurse software accesibile. Arhitectura modulara permite extinderea facila a aplicatiei, iar alegerea unei baze de date SQL Server garanteaza

integritatea datelor intr-un mediu controlat. Acest proiect poate fi extins pentru a include functionalitati aditionale precum notificari, generare de rapoarte sau integrare cu aplicatii mobile. De asemenea, reprezinta o baza solida pentru initierea in dezvoltarea de aplicatii enterprise, fiind util atat in context educational, cat si in prototiparea unor solutii reale.