



BANKING APPLICATION

**PROIECT – Administrarea si
dezvoltarea aplicatiilor cu baze de date**

Contribuitori:

Std. sg. Dicu Andrei

Std. sg. Enache Stefan

Grupa C113-D



CUPRINS:

1. Introducere
 - 1.1. Scopul proiectului
2. Descrierea generală a produsului software
 - 2.1. Descrierea produsului software
 - 2.2. Detalierea platformei HW/SW
 - 2.3. Constrângeri
3. Detalierea cerințelor software
 - 3.1. Cerințele funcționale
 - 3.2. Cerințele ne-funcționale



1.Introducere

1.1.Scopul proiectului

Proiectul implica dezvoltarea unei aplicatii de tip bancar care permite administrarea conturilor bancare ale clientilor inregistrati.Sistemul nostru este conceput sa permita crearea unui cont current pentru fiecare utilizator.

Scopul principal consta in facilitarea operatiunilor bancare: utilizatorii pot adauga bani intr-un depozit, pot efectua tranzactii, pot solicita imprumuturi.

Plata imprumuturilor este facilitata in aplicatie, utilizatorii pot plati ratele pentru fiecare dintre imprumuturi direct din aplicatie.

De asemenea, se pot vizualiza istoricul tranzactiei si un grafic de analiza al raportului incasari/cheltuieli pentru contul current.

2.Descrierea generală a produsului software

2.1. Descrierea produsului software

Arhitectura acestui proiect se bazeaza pe utilizarea sablonului Model-View-ViewModel (MVVM). MVVM este un sablon de proiectare care separă problemele datelor (model), prezentării (vizualizare) și interacțiunii cu utilizatorul (model de vizualizare) într-un mod care permite o separare clară a responsabilităților și promovează întreținerea și testabilitatea.

În aplicatia noastra, clasa **MainViewModel** reprezinta modelul de vizualizare în sablonul MVVM. Ea este responsabilă pentru încărcarea datelor utilizatorului curent, inițializarea comenzilor și setarea vizualizării implicite. Clasa **UserRepository** este modelul, responsabilă pentru interacțiunea cu baza de date și furnizarea datelor pentru modelul de vizualizare.

```
public MainViewModel()
{
    userRepository = new UserRepository();
    LoadCurrentUserData();

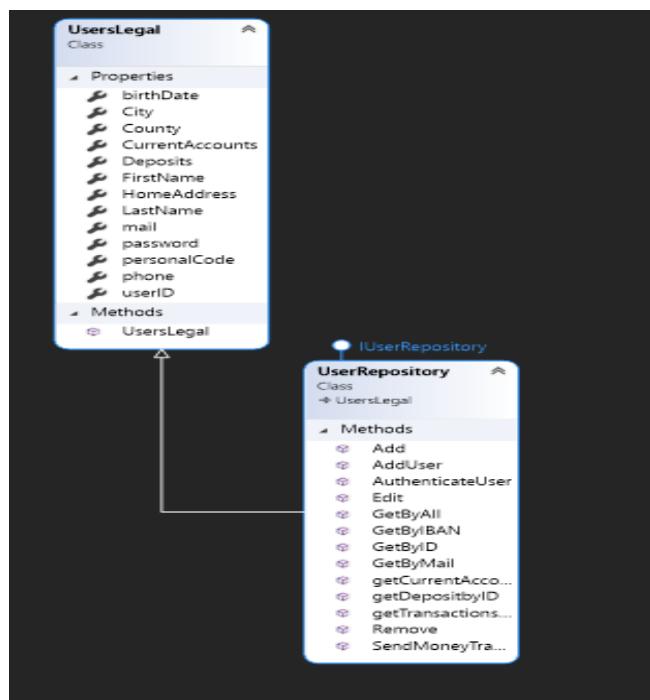
    //Initializez comenzile de mai sus
    ShowOverviewViewCommand = new ViewModelCommand(ExecuteShowOverviewViewCommand);
    ShowStatisticsViewCommand = new ViewModelCommand(ExecuteShowStatisticsViewCommand);
    ShowTransactionsViewCommand = new ViewModelCommand(ExecuteShowTransactionsViewCommand);
    ShowTransfersViewCommand = new ViewModelCommand(ExecuteShowTransfersViewCommand);

    //View DEFAULT!!!
    ExecuteShowOverviewViewCommand(null);
}
```

Livrabilul 1 - descrierea cerințelor software



Clasa UserRepository care mosteneste clasa numita **UsersLegal** si implementeaza interfata IUserRepository. Aceasta clasa este utilizata pentru a *accesa si gestiona datele* din entitatea asociata din baza de date, un tabel specific in baza de date. Proprietatile din UsersLegal sunt utilizate pentru a defini coloanele din tabel si IUserRepository defines metodele de operare pentru acea tabel. Prin urmare, clasa UserRepository poate fi utilizata pentru a realiza operatii precum adaugarea, actualizarea si selectarea datelor din tabelul asociat din baza de date.



MainViewModel este punctul central pentru interfața utilizatorului aplicației. Acesta conține comenzi care permit utilizatorului să navigheze între diferite vizualizări sau pagini din aplicație. Fiecare comandă este asociată cu un ViewModel copil specific. De exemplu, comanda *ShowOverviewViewCommand* este asociată cu *OverviewViewModel*, iar comanda *ShowStatisticsViewCommand* este asociată cu *StatisticsViewModel*.

Când o comandă este executată, se creează o nouă instanță a ViewModel-ului copil corespunzător și este setat ca CurrentChildView. Acest lucru permite MainViewModel-ului să mențină controlul asupra fluxului de navigare al aplicației, în timp ce permite fiecărui ViewModel copil să gestioneze propriul său stadiu și logica.

MainViewModel-ul expune, de asemenea, proprietăți și metode care sunt utilizate pentru a comunica cu ViewModels-ul copil, cum ar fi transmiterea datelor sau declanșarea actualizărilor. Acest lucru permite MainViewModel-ului să coordoneze comportamentul diferitelor ViewModels copil și să asigure că interfața utilizatorului rămâne consistentă și receptivă.



2.2. Detalierea platformei HW/SW

Produsul este alcatuit dintr o singura aplicatie software, ce va fi conectata cu o baza de date in care se stocheaza si se administreaza datele folosite in aplicatie. solicitantii de servicii. Ca instrument de lucru vom folosi : Windows Presentation Foundation (WPF) utilizând C# și Framework-ul .NET. Aplicația folosește Entity Framework pentru interacțiunea cu baza de date . combinația WPF, C#, Framework-ul .NET și Entity Framework creează un mediu robust și eficient pentru dezvoltarea aplicațiilor desktop Windows.

2.3. Constrângeri

- Specific pentru platformă: WPF este un cadru pentru dezvoltarea aplicațiilor desktop Windows, ceea ce înseamnă că aplicația poate fi executată numai pe sistemele de operare Windows.
- Cerințe hardware: Aplicațiile WPF pot necesita mai multe resurse decât aplicațiile Windows tradiționale, cum ar fi memorie suplimentară și putere de procesare. Acest lucru ar putea fi o constrângere pentru utilizatorii cu computere mai vechi sau mai puțin puternice.



3. Detalierea cerințelor software

3.1. Cerințele funcționale

- Autentificarea in contul de utilizator in baza unor credentiale

```
private void ExecuteLoginCommand(object obj)
{
    var isValidUser = userRepository.AuthenticateUser(new NetworkCredential(Username, Password));
    if (isValidUser)
    {
        Thread.CurrentPrincipal = new GenericPrincipal(new GenericIdentity(Username), null);
        IsViewVisible = false;
    }
    else
    {
        ErrorMessage = "* invalid mail or password";
    }
}
```

Proprietatea `Thread.CurrentPrincipal` permite să se definească contextul de securitate al unui fir de execuție. Este utilizat pentru a stoca identitatea și rolurile utilizatorului, poate fi utilizat pentru a autoriza operațiile pe care utilizatorul încearcă să le efectueze.

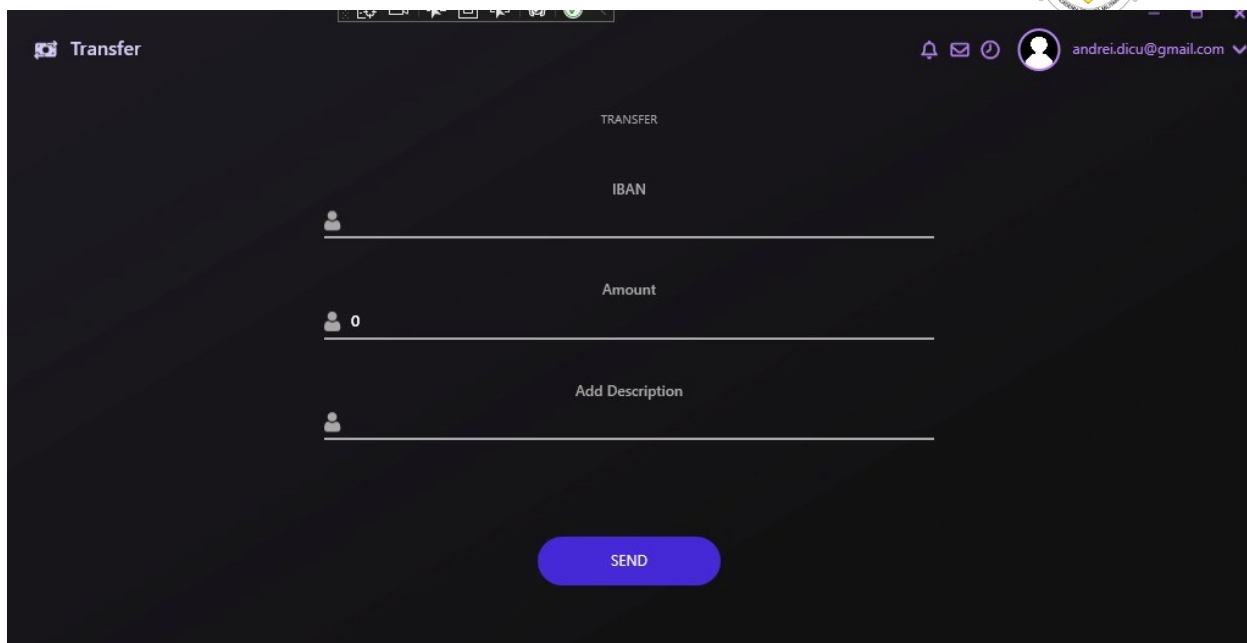
- Efectuarea de tranzactii din contul curent

```
private void ExecuteSendCommand(object obj)
{
    CurrentUserAccount = new UserAccountModel();
    var user = userRepository.GetById(Thread.CurrentPrincipal.Identity.Name);
    var current_account = userRepository.GetCurrentAccountById(user.userID);
    var receiver_accountID = userRepository.GetByIBAN(Iban);
    var receiver_account = userRepository.GetById(receiver_accountID);

    var transactions = userRepository.GetTransactionsListByIBAN(current_account.IBAN);
    CurrentUserAccount.Lista_tranzactii = transactions;
    for (int i = 0; i < transactions.Count; i++)
    {
        if (CurrentUserAccount.Lista_tranzactii[i].srcIBAN == current_account.IBAN)
        {
            CurrentUserAccount.CurrentAccount_Balance -= CurrentUserAccount.Lista_tranzactii[i].amount;
        }
        else
        {
            CurrentUserAccount.CurrentAccount_Balance += CurrentUserAccount.Lista_tranzactii[i].amount;
        }
    }
    var ctx = new BigBankDBEntities();
    var idCount = ctx.Transactions.Count();

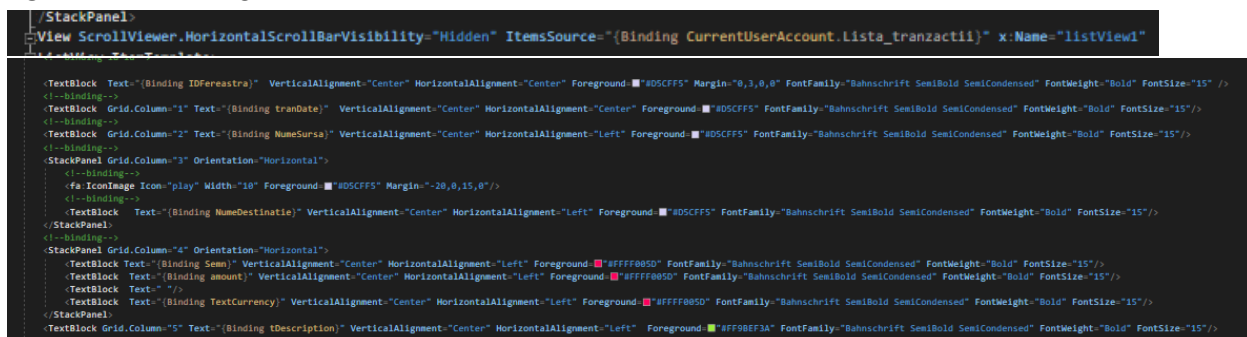
    if (receiver_account != null && Amount <= CurrentUserAccount.CurrentAccount_Balance)
    {
        idCount++;
        userRepository.SendMoneyTransfer(idCount, current_account.IBAN, Iban, current_account.currency, Amount, Description); //trebuie modificat astfel incat sa pot obtine iban current si currency
        ErrorMessage = "";
        SuccessMessage = "The transfer was executed successfully!";
    }
    else if (receiver_account == null)
    {
        ErrorMessage = "* invalid IBAN or doesn't exist";
    }
    else if (Amount > CurrentUserAccount.CurrentAccount_Balance)
    {
        ErrorMessage = "* insufficient funds";
    }
}
```

Livrabilul 1 - descrierea cerințelor software



- Vizualizare istoric tranzactii

FORMAT TRANZACTIE:

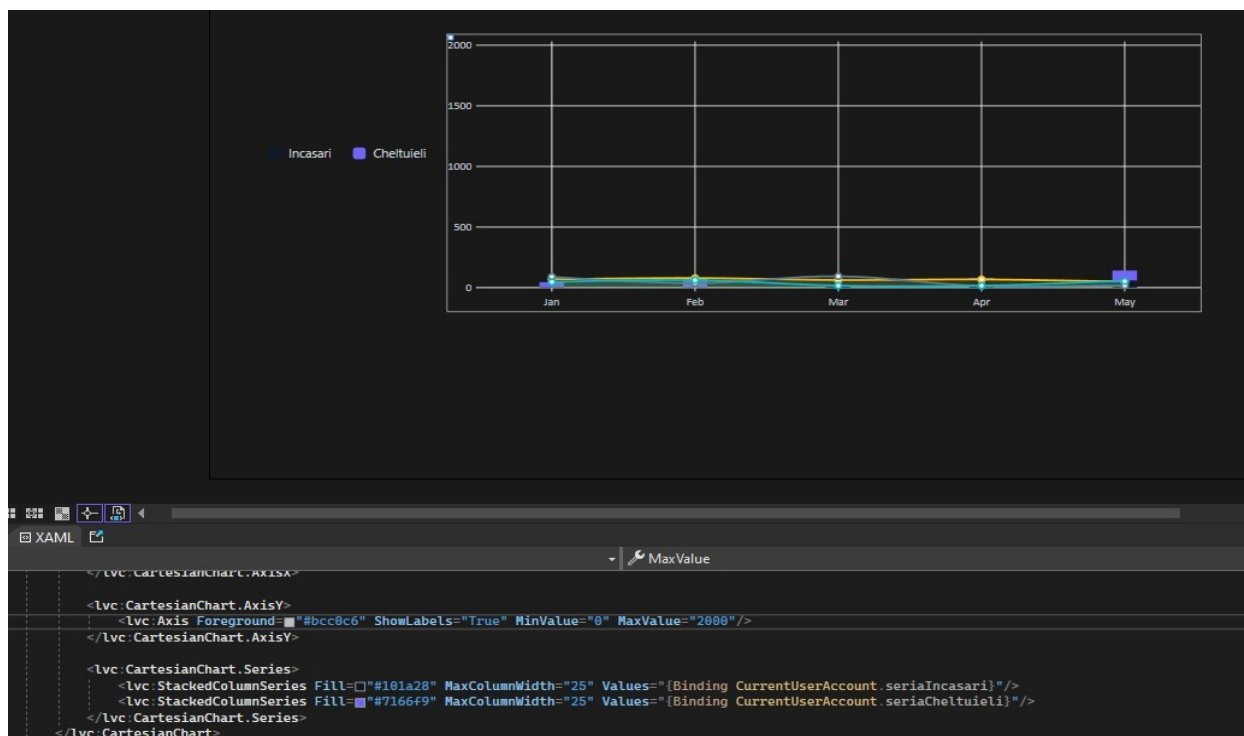


Livrabilul 1 - descrierea cerințelor software



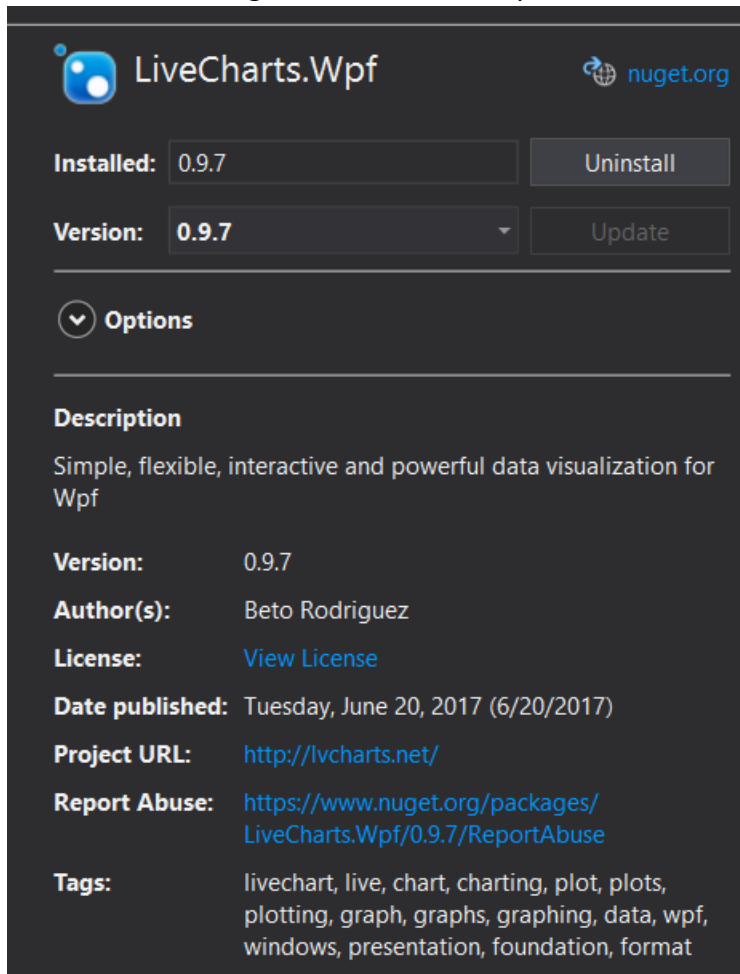
Transactions						
ID	TRANSACTION DATE	SOURCE	DESTINATION	AMOUNT	DESCRIPTION	
1	1/17/2023 12:00:00 AM	Andrei Dicu ▶	Vlad Roman	-300.0000 RON	Verificare statistica	
2	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
3	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
4	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
5	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
6	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
7	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
8	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	
9	1/17/2023 12:00:00 AM	Andrei Dicu ▶	BIG BANK	-127.5000 RON	BigBank - The installment pay	

-Vizualizare raport lunar(incasari/cheltuieli)





Pentru crearea diagramelor am utilizat pachetul LiveCharts.Wpf

The image shows the NuGet package details for LiveCharts.Wpf. At the top, there is a logo for LiveCharts.Wpf and a link to nuget.org. Below this, there are two rows: 'Installed: 0.9.7' with an 'Uninstall' button, and 'Version: 0.9.7' with an 'Update' button. A section titled 'Options' is collapsed. Below that, there is a 'Description' section with the text 'Simple, flexible, interactive and powerful data visualization for Wpf'. This is followed by a list of metadata: 'Version: 0.9.7', 'Author(s): Beto Rodriguez', 'License: View License', 'Date published: Tuesday, June 20, 2017 (6/20/2017)', 'Project URL: http://lvcharts.net/', 'Report Abuse: https://www.nuget.org/packages/LiveCharts.Wpf/0.9.7/ReportAbuse', and 'Tags: livechart, live, chart, charting, plot, plots, plotting, graph, graphs, graphing, data, wpf, windows, presentation, foundation, format'.

Livrabilul 1 - descrierea cerințelor software



- Selectarea unui tip de imprumut si solicitarea acestuia catre banca, banii vor fi furnizati in contul current

The screenshot shows a dark-themed interface for selecting a loan type. At the top, there is a hamburger menu icon followed by the text "LOANS:". Below this, there are two cards. The left card is titled "TYPE STANDART" in red, with "Number Months: 12" and "Interest: 2%" in red text. The right card is titled "TYPE EXTENSIVE" in yellow, with "Number Months: 24" and "Interest: 3%" in yellow text. Below the cards, there is a label "Choose the type of loan:" followed by a radio button and the text "TYPE STANDART". Below this, there is a label "Loan Amount:" followed by a text input field containing "0". At the bottom, there is a blue button labeled "CONFIRM".

- Plata de rate standard pentru un anumit imprumut

The screenshot shows a dark-themed interface for selecting a loan to pay. At the top, there is a hamburger menu icon followed by the text "LOANS:". Below this, there is a table with two rows of loan data. The first row has columns for ID (1005), Start Date (1/17/2023 12:00:00 AM), Amount (97.5000), Currency (RON), Term (12), and End Date (1/17/2023 12:00:00 AM). The second row has columns for ID (1004), Start Date (1/17/2023 12:00:00 AM), Amount (65.0000), Currency (RON), Term (12), and End Date (1/17/2023 12:00:00 AM). Below the table, there is a label "Insert the ID of the loan for which you want to pay:" followed by a text input field containing "0". At the bottom, there is a blue button labeled "PAY FOR THIS MONTH".

ID	Start Date	Amount	Currency	Term	End Date
1005	1/17/2023 12:00:00 AM	97.5000	RON	12	1/17/2023 12:00:00 AM
1004	1/17/2023 12:00:00 AM	65.0000	RON	12	1/17/2023 12:00:00 AM



3.2.Cerințele ne-funcționale

- Produsul poate functiona pe sisteme de operare Windows.
- Cerintele de performanta pentru functionarea optima sunt minime.
- Aplicatia va avea nevoie de o comunicatie la un server MS SQL