Software Requirements Specification

For

Online Banking Application

Prepared by:

Grosu Claudia Mihaela
Patru Diana-Cristina
Ionita Alexandru

Version 2.7 Approved

2024

Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Grosu Claudia Mihaela	28.02.2023	Introduction	1.0
Patru Diana-Cristina	28.02.2023	Introduction	1.1
Ionita Alexandru	28.02.2023	Introduction	1.2
Grosu Claudia Mihaela	01.03.2023	Overall Description	1.3
Patru Diana-Cristina	01.03.2023	Overall Description	1.4

Ionita Alexandru	01.03.2023	Overall Description	1.5
Grosu Claudia Mihaela	02.03.2023	External Interface Requirements	1.6
Patru Diana-Cristina	02.03.2023	External Interface Requirements	1.7
Ionita Alexandru	02.03.2023	External Interface Requirements	1.8
Grosu Claudia Mihaela	03.03.2023	System Features	1.9
Patru Diana-Cristina	03.03.2023	System Features	2.0
Ionita Alexandru	03.03.2023	System Features	2.1
Grosu Claudia Mihaela	04.03.2023	Other Nonfunctional Requirements	2.2
Patru Diana-Cristina	04.03.2023	Other Nonfunctional Requirements	2.3
Ionita Alexandru	04.03.2023	Other Nonfunctional Requirements	2.4
Grosu Claudia Mihaela	07.03.2023	Service Payment - Account and Transactions - Business Rules	2.5
Patru Diana-Cristina	07.03.2023	Service Payment - Account and Transactions - Business Rules	2.6
Ionita Alexandru	07.03.2023	Service Payment - Account and Transactions - Business Rules	2.7

• Introduction

• Purpose

Online banking is a convenient way to manage your finances from the comfort of your own home or on-the-go. With this application, you can view your account balances and transaction history, transfer funds between accounts, pay bills, checks information about account.

Banking Application MTN uses advanced security features to protect your sensitive financial information. This includes authentication, encryption. With these security measures in place, you can have peace of mind knowing that your data is safe and secure.

Banking Application also offers a user-friendly interface that makes managing your finances easy and intuitive.

Overall, *Banking Application MTN* is a powerful tool for managing your finances on-the-go. With its robust features and top-notch security, you can stay on top of your finances with ease and confidence.

• Document Conventions

We used to write this document in Times New Roman font. For each chapter of this document that includes subchapters we used a font size equal to 18 and bold. The subchapters have a font size equal to 14 and bold. The description for each chapter or subchapter has a font size equal to 12.

Term / Acronym	Definition
DESC	Description
NFR	Non-functional requirement
ID	Unique identifier
Admin	Administrator
DB	Database

• Intended Audience and Reading Suggestions

This document is intended for all groups of people who will have contact with this application. The types of audience for this document are the following: users, developers, and administrators.

• Product Scope

The purpose of this online banking application is to provide a convenient and secure way for users to:

- View account information.
- Change the client's address.
- Change the password.
- Make transfers to other accounts.

- Make payments to certain services (Electricity, Phone, etc.)
- View transactions.

The app aims to make financial management as easy and intuitive as possible for users, while also providing advanced security features to protect users' financial data.

• References

https://www.raiffeisen.ro/
https://app.diagrams.net/

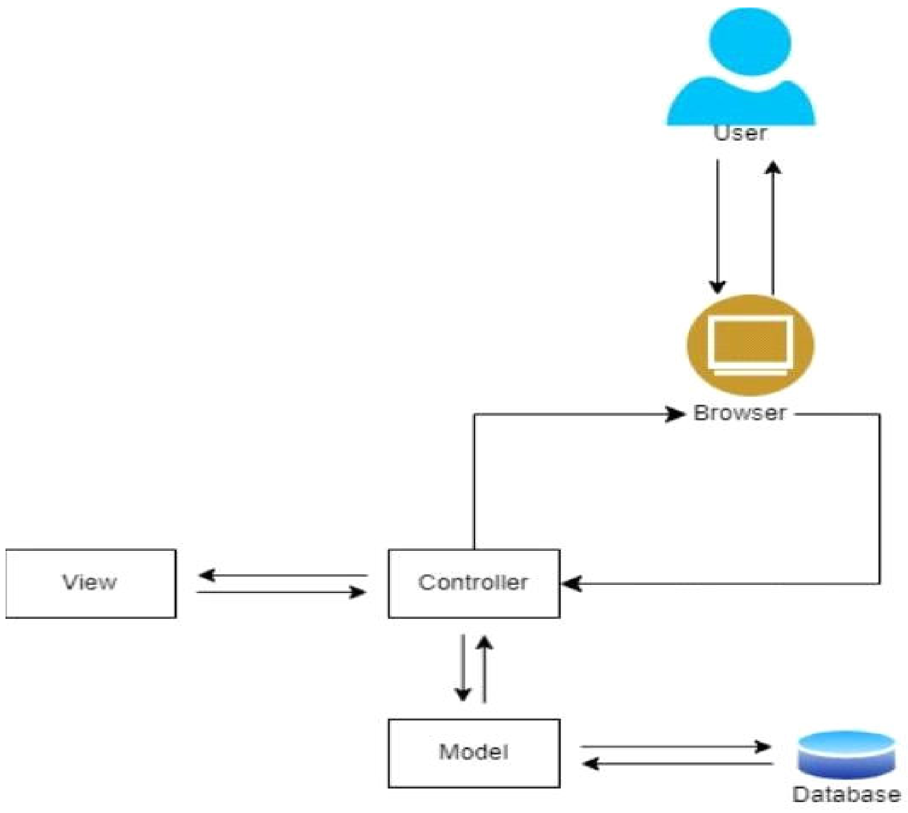
• Overall Description

• Product Perspective

DESC: This system consists of a single part, a web application, which will be used by admin, but also by users.

Also, the application will constantly communicate with its database. The application will be accessed using a computer that has at least the minimum usage specifications application.

Such a structure is shown graphically in the following figure:



Product Functions

- *Viewing account balances:* Users could see their account balances in real-time.
- *Money transfers:* Users could transfer money between their own accounts or to other bank accounts.
- *Bill payments:* Users could pay bills directly from the app, such as utility bills, mobile phone bills, etc.
- *Currency exchange:* Users could exchange currencies in the app.
- *Transaction checking:* Users could check their transaction history.
- *Card management:* Users could manage their bank cards, such as blocking or unblocking cards.

User Classes and Characteristics

In our application, a **User** can have multiple roles, and we have created a **Role** class for this purpose.

A User with the **Administrator** role:

- has an ID.
- can verify employees.
- can modify the role of each user.

A User with the **Employee** role:

- has an ID.
- verifies customer information.

- modifies certain customer information.
- approves or rejects transactions that involve a large amount of money.
- responds to help requests.

A User with the **Client** role:

- has an ID.
- can make transactions.
- can pay bills.
- can view their accounts information.

• Operating Environment

The application will work on PCs that have a Windows operating system 10.

To be able to use this application, users must have access to the Internet connection and Chrome Web Browser version 110.0.5481.178, Microsoft Edge version 99.0.1150.36 and Brave version 99.1.36.111 depending on the device (PC or Laptop).

• Design and Implementation Constraints

- The application will be implemented using Java 17 as programming language and Spring Boot as framework.
- The code will be written in IntelliJ IDEA Community Edition version 2022.2.3.
- Will use MySQL 8.0.32 as the database.
- The application will be tested using JUnit tests and Integration tests.
- Information about the users and each account is stored in a database that constantly communicates with the web application.

• User Documentation

All details about this application will be presented in this document.

• Assumptions and Dependencies

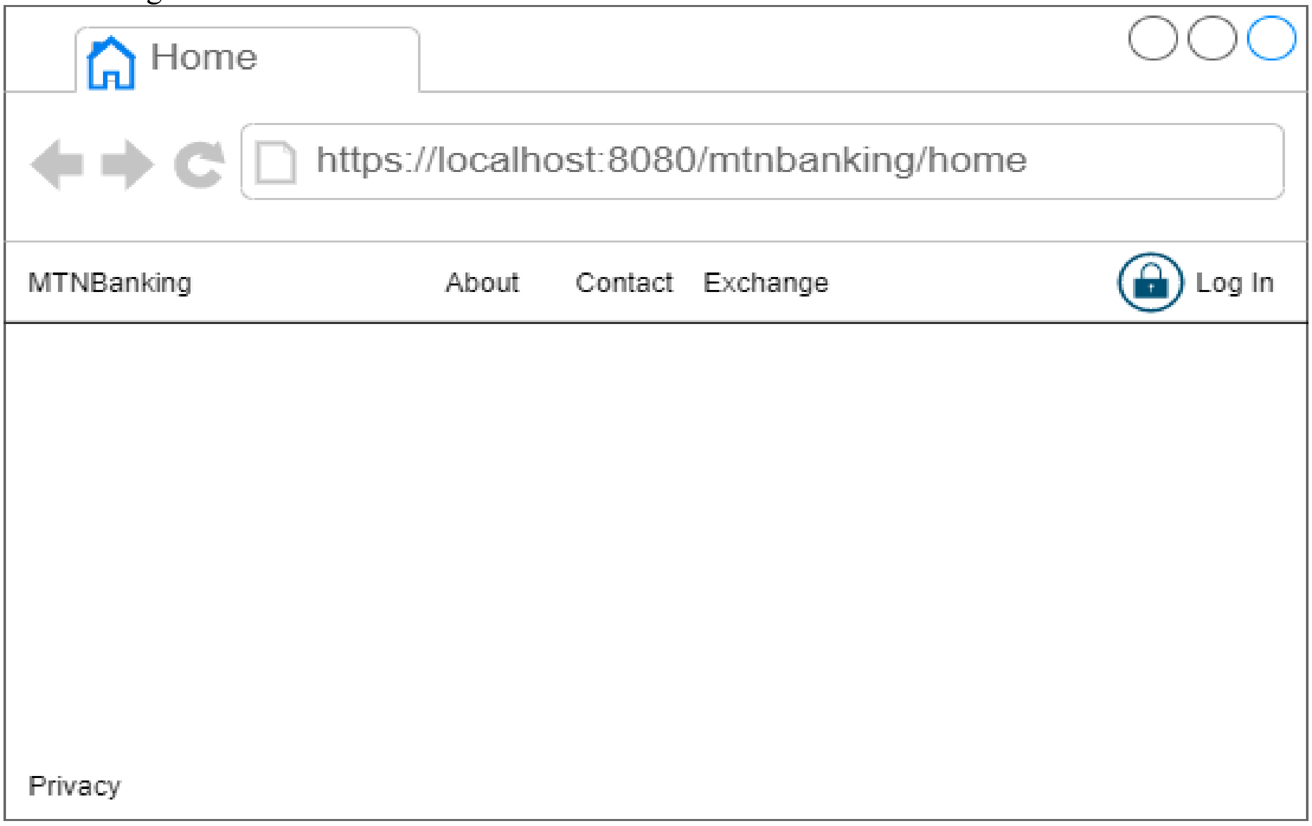
To carry out this project, we would need the following dependencies:

- spring-boot-starter-web
- spring-boot-starter-data-jpa
- mysql-connector-java
- spring-boot-starter-validation
- modelmapper
- flyway-core
- flyway-mysql
- spring-boot-starter-test
- junit-jupiter-engine
- h2

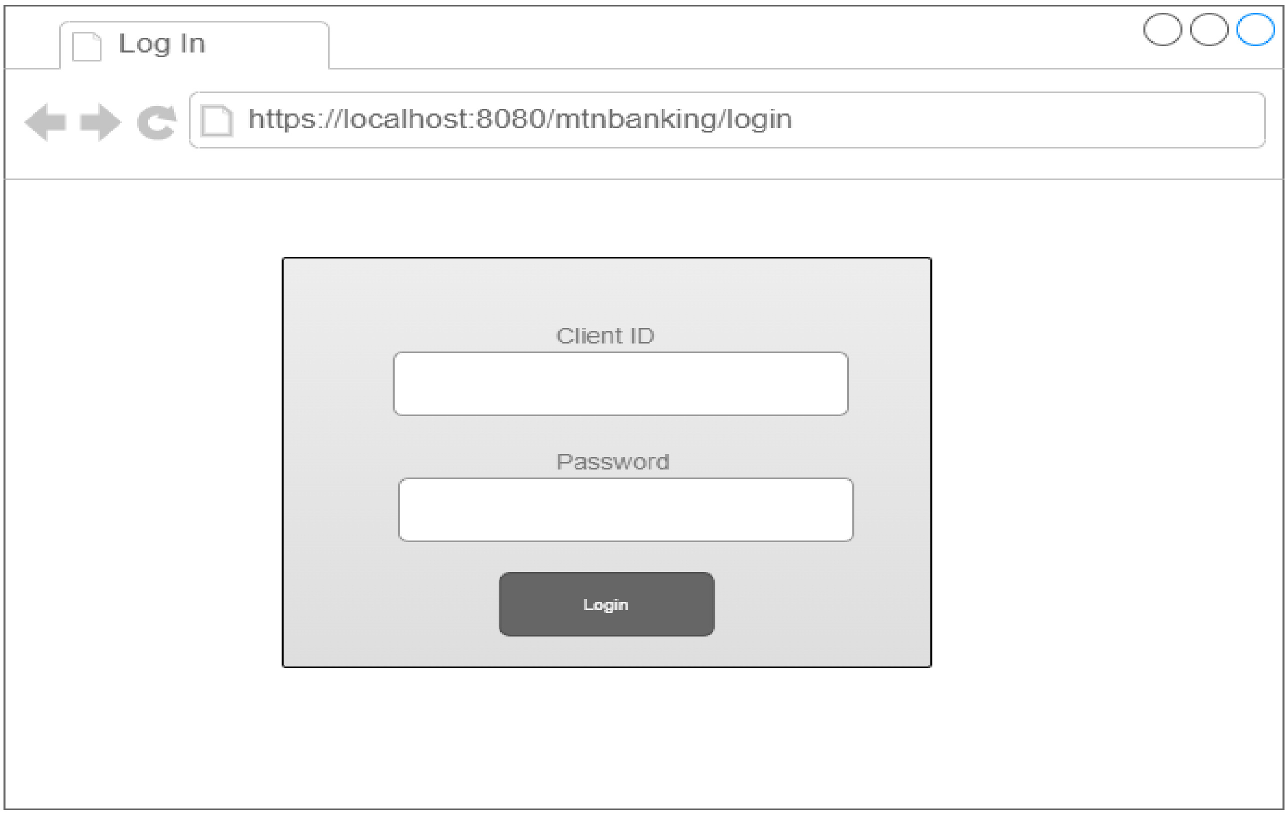
- ## External Interface Requirements

- ### User Interfaces

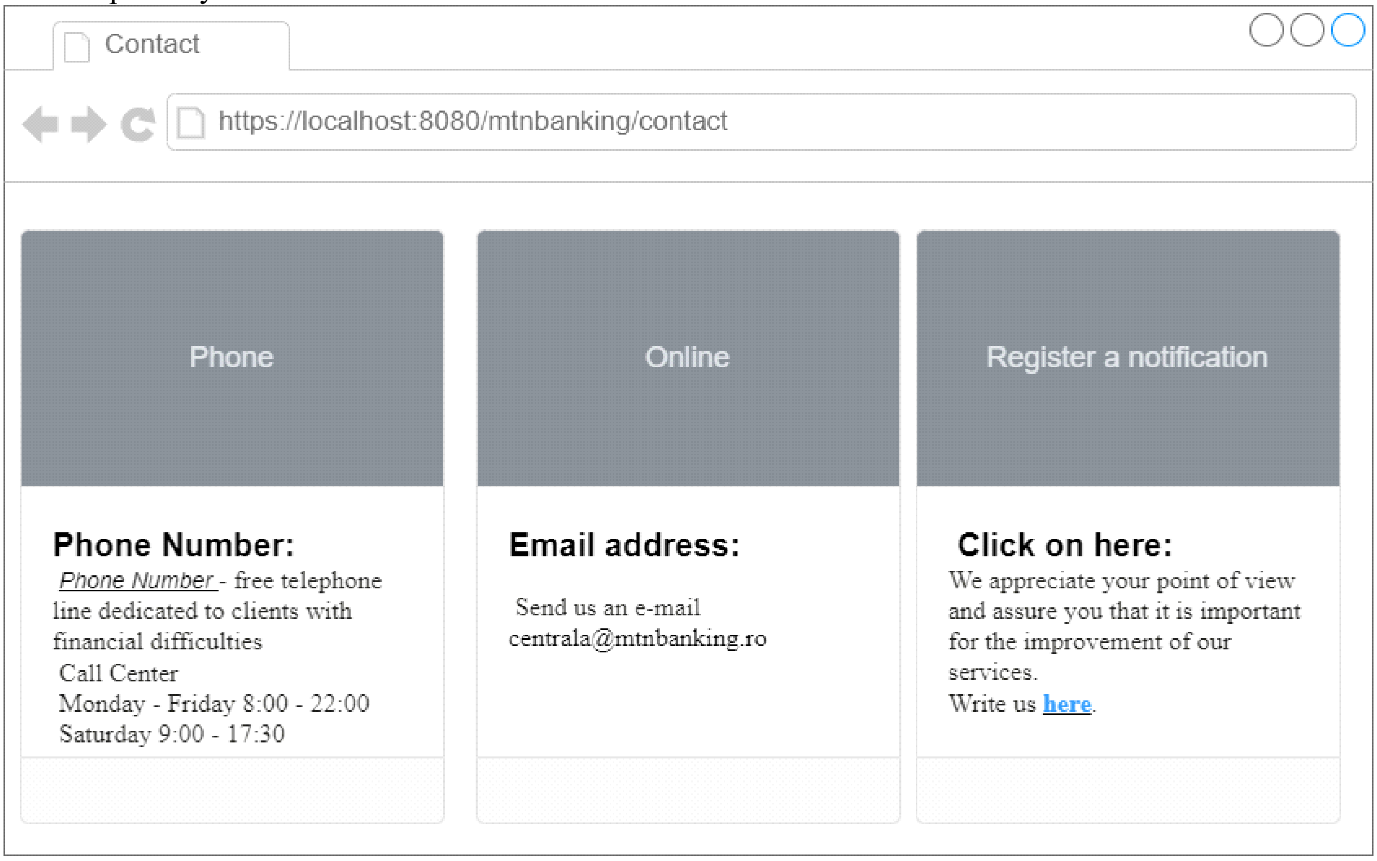
Once starting the web application, the user is welcomed with the **Home** page. This page contains the essential information about the product that the web application is delivering.



Login page has text boxes for the client ID and password. After the person enters their credentials, the application will recognize if the person is an employee, a simple user or administrator.
Therefore, if a user with client role logs in, he will be redirected to the account list.
If an employee or administrator logs in, he will also be redirected to the specific manage page.



The **Contact** page lets any employee/client send a ticket to the user support regarding any bugs/issues encountered while in the web application. He can see details about contacts such as phone number, email address, and we have a separate page where he can report any issues he encounters.

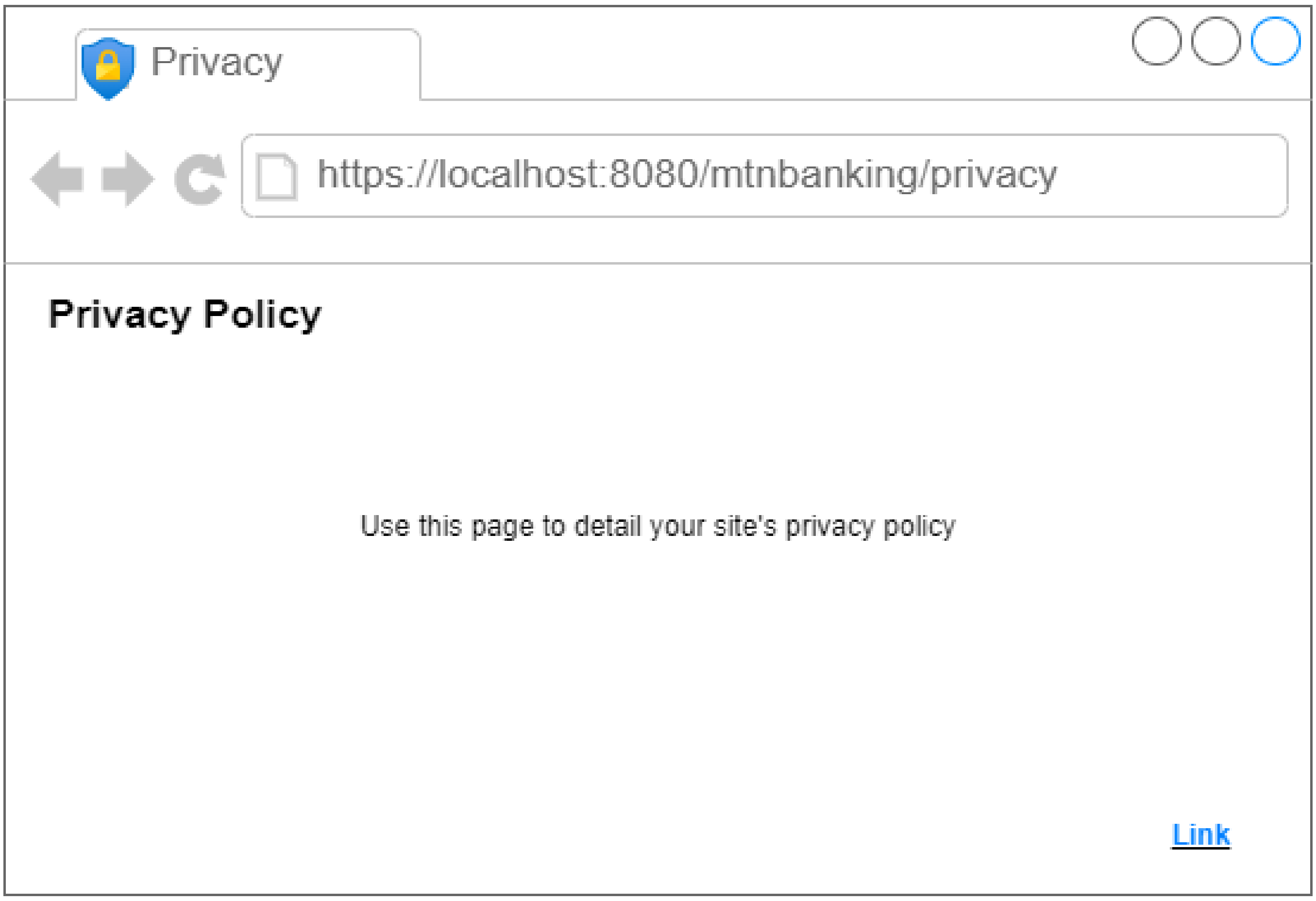


Phone Number:
Phone Number - free telephone line dedicated to clients with financial difficulties
Call Center
Monday - Friday 8:00 - 22:00
Saturday 9:00 - 17:30

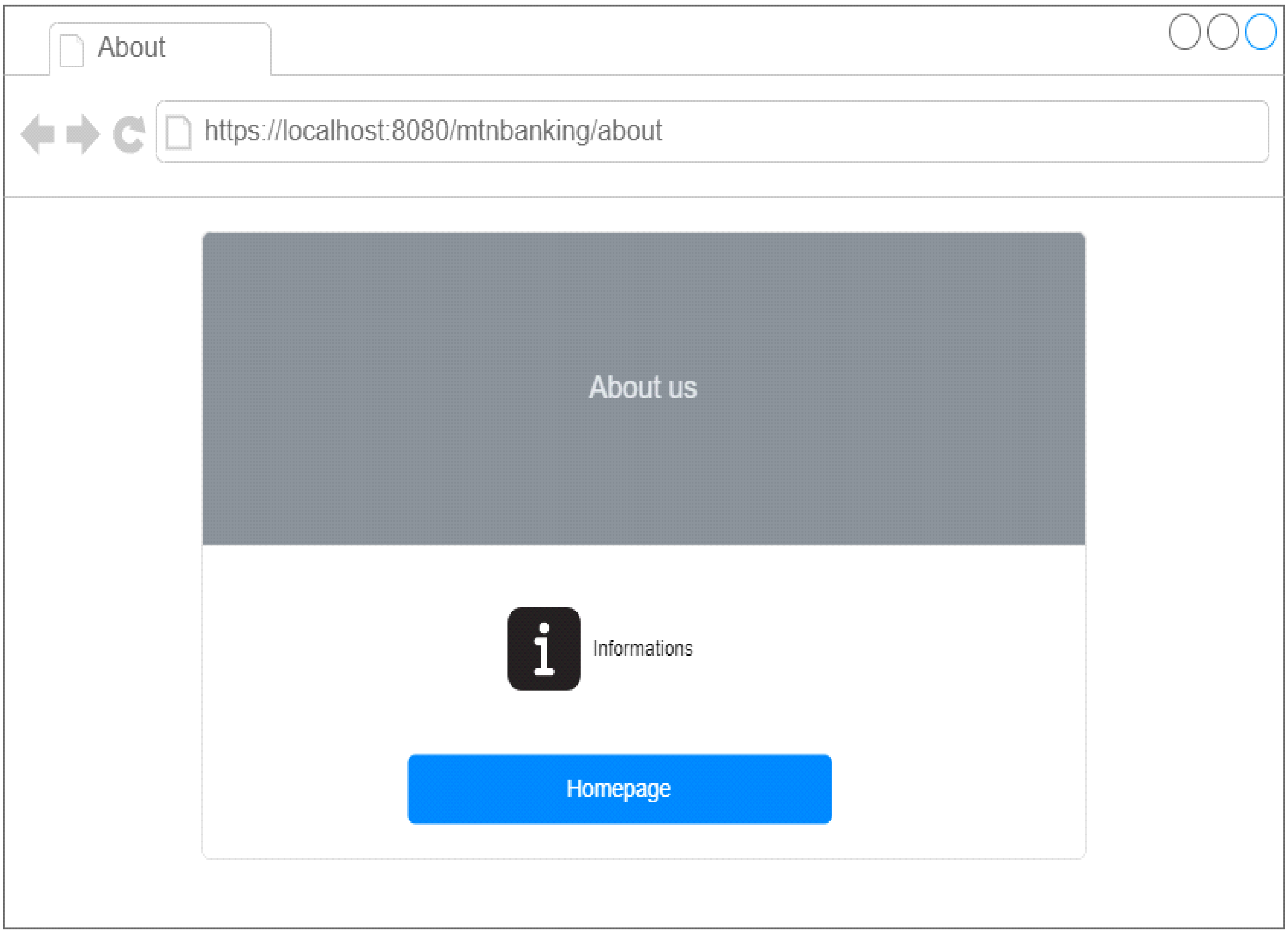
Email address:
Send us an e-mail
centrala@mtnbanking.ro

Click on here:
We appreciate your point of view and assure you that it is important for the improvement of our services.
Write us **here**.

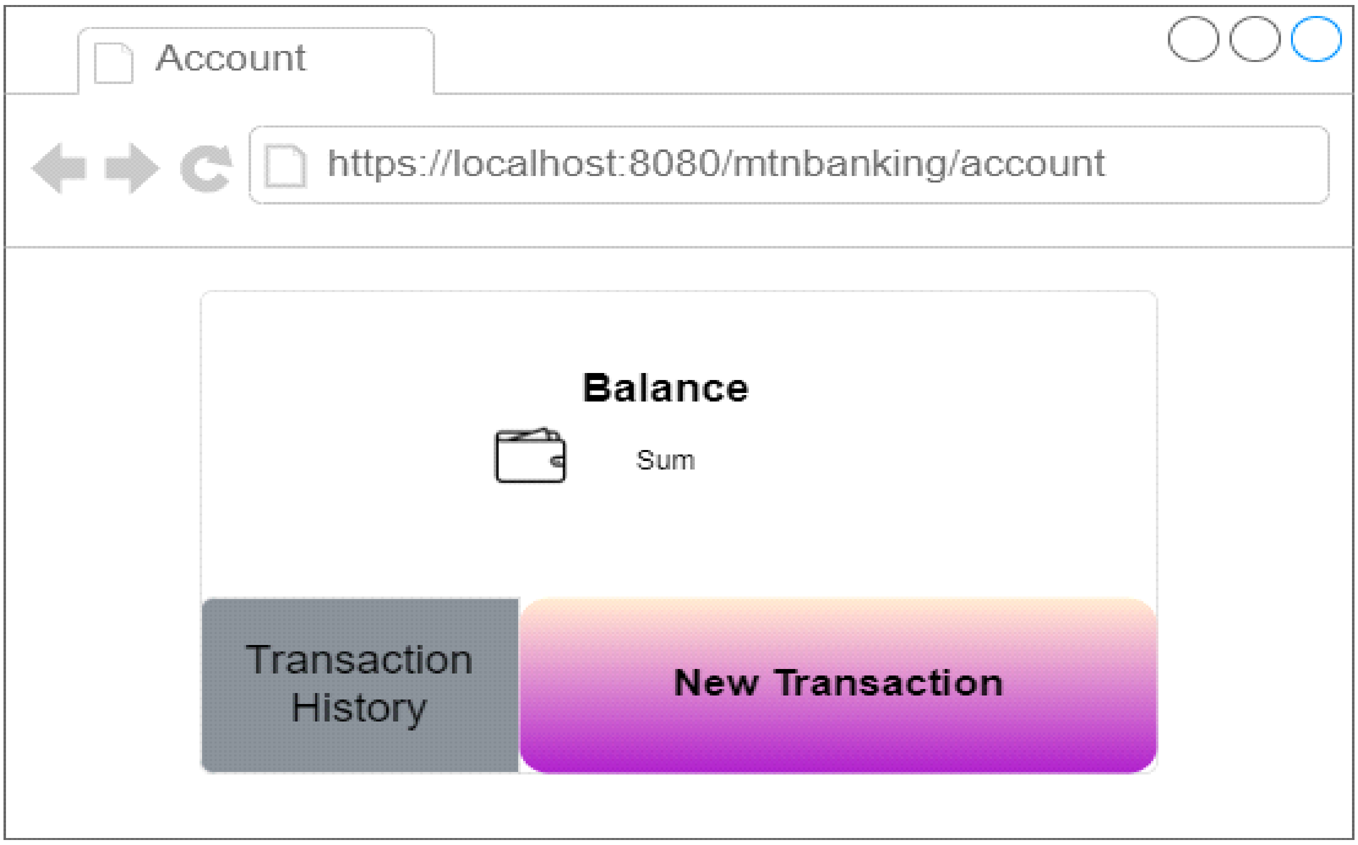
The **Privacy** page contains the privacy and policy of the web application according to the regulations and data stored within the database. The user presses the link button and lands on the external privacy data.


Privacy
https://localhost:8080/mtnbanking/privacy
Privacy Policy
Use this page to detail your site's privacy policy
Link

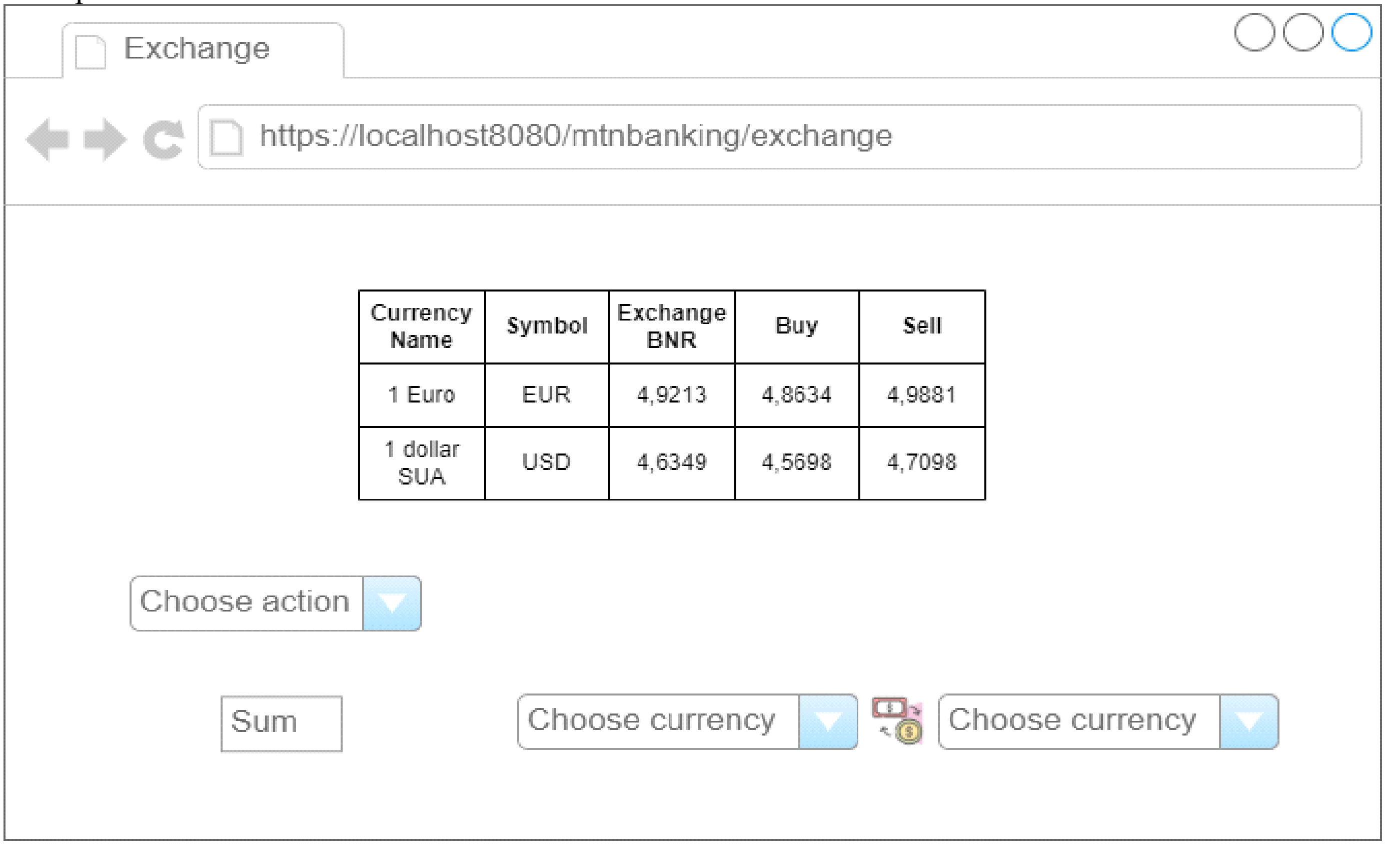
The **About** page is an essential source of information for all who want to know more about that application. More information about the bank can be found here. The user presses the ‘Homepage’ button, and he lands on the Home page.



Once logged in, the user can view their account balance by accessing the "Transaction History" button to view their transaction history or by making a new transaction by pressing the "New Transaction" button.



On this page, you can see the real-time exchange rate, and in addition to that, it has the functionality to convert an amount from one currency to another, using the dropdown menus.



Currency Name	Symbol	Exchange BNR	Buy	Sell
1 Euro	EUR	4,9213	4,8634	4,9881
1 dollar SUA	USD	4,6349	4,5698	4,7098

- **Hardware Interfaces**

Not applicable.

- **Software Interfaces**

The connection between the database and the server will always be active. This application will contain various functions with different functionalities (to read from the DB, to write to the DB, or delete from the DB, there will also be a function that will view the data in the DB).

Function prototypes are:

- private Object insert (Object obj)
- private void update (Object obj, String uuid)
- private void delete (Object obj, String uuid))
- private Object getData (String uuid)

- **Communications Interfaces**

The communication between the website and database will be done using the HTTP communication protocol.

• System Features

In this paragraph are noted and described the functional requirements for the application.

• User with Administrator Role - login and sign-up

- **Description and Priority**

The administrator can create a new account if he does not already have an account. The adm

Priority: **High**

- **Stimulus/Response Sequences**
  - **Input:** The administrator enters the username and password.
  - **Output:** After logging in to the account, the application will send a response as to whether the administrator has logged in successfully.

- **Functional Requirements**

REQ-1: There must be a text field in which the administrator enters the username.

REQ-2: There must be a text field in which the administrator enters the password.

REQ-3: There must be a button that the administrator must press after entering the data.

REQ-4: If the username is wrong, the administrator cannot login with success.

REQ-5: If the password does not correspond to the user, then the administrator will not be able to login successfully.

REQ-6: For accessing the account, the administrator needs to enter the username and the password.

REQ-7: For registering an administrator, we will follow a special procedure, and they will be added to the database as such to enhance the application's security. There is no possibility for a user to create an administrator account.

REQ-8: The password must contain at least 8 characters.

REQ-9: The username needs to be unique.

REQ-10: After the login was successful, a pop-up message will be displayed to confirm that.

REQ-11: The password will be encrypted.

REQ-12: The administrator has a special function through which they can add employees to the database.

• User with Employee Role – login and sign-up

- **Description and Priority**

  The employee can create a new account if he does not already have an account. The employee will be able to login.
  Priority: **High**

- **Stimulus/Response Sequences**

  - **Input:** The employee enters the username and password.
  - **Output:** After logging in to the account, the application will send a response as to whether the employee has logged in successfully.

- **Functional Requirements**

REQ-13: There must be a text field in which the employee enters the username.

REQ-14: There must be a text field in which the employee enters the password.

REQ-15: There must be a button that the employee must press after entering the data.

REQ-16: If the username is wrong, the employee cannot login with success.

REQ-17: If the password does not correspond to the user, then the employee will not be able to login successfully.

REQ-18: For accessing the account, the employee needs to enter the username and the password.

REQ-19: For registering an employee, we will follow a special procedure, and they will be added to the database with the help of an administrator to increase the application's security. There is no possibility for a user to create an employee account.

REQ-20: The password must contain at least 8 characters.

REQ-21: The username needs to be unique.

REQ-22: After the login was successful, a pop-up message will be displayed to confirm that.

REQ-23: The password will be encrypted.

REQ-24: The employee has a special function through which they can add clients to the database.

• User with Client Role – login and sign-up

- **Description and Priority**

  The client can create a new account if he does not already have an account. The client will be able to login.
  Priority: **High**

- **Stimulus/Response Sequences**

  - **Input:** The client enters the username and password.

- **Output:** After logging in to the account, the application will send a response as to whether the client has logged in successfully.

- **Functional Requirements**

REQ-25: There must be a text field in which the client enters the

username.

REQ-26: There must be a text field in which the client enters the

password.

REQ-27: There must be a button that the client must press after

entering the data.

REQ-28: If the username is wrong, the client cannot login with

success.

REQ-29: If the password does not correspond to the user, then the client will not be able to login successfully.

REQ-30: For accessing the account, the client needs to enter the username and the password.

REQ-31: For registering a client, we will follow a special procedure, and they will be added to the database with the help of an employee to increase the application's security. There is no possibility for a user to create a client account.

REQ-32: The password must contain at least 8 characters.

REQ-33: The username needs to be unique.

REQ-34: After the login was successful, a pop-up message will be displayed to confirm that.

REQ-35: The password will be encrypted.

REQ-36: The client can update their address and change their password.

• Account and Transactions:

- **Description and Priority**

The customer can select the account they want to access, and then they can perform certain transactions and view account details.

Priority: **High**

- **Stimulus/Response Sequences**

- **Input**: If the client chooses to make a transfer to another account, they must enter information about destination account.

- **Output**: The transfer is done by transferring the money to the destination account and withdrawing it from the source account.

- **Functional Requirements**

REQ-37: When the user presses the "New Transaction" button and selects to make a transfer to another account, they enter information such as the IBAN of the account they are sending to, the name of the beneficiary, the amount they are transferring, and optionally a description, and then press the "Send" button.

REQ-38: After pressing the "Send" button, a pop-up window will appear to confirm if the data is correct.

REQ-39: After confirming the transfer, the customer has the option to download a PDF proof of the transaction.

REQ-40: When accessing their account, the customer can view information about it, such as the amount of money they hold and the currency in which the funds are held.

REQ-41: When accessing the "Transaction History" button, the user will be able to view the history of transactions.

• Service Payment

- **Description and Priority**

The customer has the possibility to make payments to services such as phone, electricity, gas,
etc.
Priority: **Medium**

- **Stimulus/Response Sequences**

- **Input**: When selecting the option to make payments to services, the customer needs to enter the necessary information.

- **Output**: After entering the necessary data, proof of payment will be generated.

- **Functional Requirements**

REQ-42: When selecting the option to make payments to services, the customer needs to select the service and choose the provider from a list provided by the interface and enter the amount.

REQ-43: After entering the necessary data, proof of payment will be generated. In the case of a transfer, the client has the possibility to send money to other accounts.

• Administrator/Employee log out:

- **Description and Priority**

  The administrator/employee can log out by clicking on the logout button. Priority: **High**

- **Stimulus/Response Sequences**

- **Input**: There is a logout button on the administrator/employee's permissions page.

- **Output**: After selecting the button the administrator/employee will exit from this page.

- **Functional Requirements**

REQ-44: There must be a button for the administrator/employee to press when they want to log out.

REQ-45: After pressing the button, the administrator/employee is redirected to the home page.

REQ-46: After the logout was successful, a pop-up message will be displayed to confirm that.

• Client log out:

- **Description and Priority**

The client can log out by clicking on the logout button or automatically when the session expired.
Priority: **High**

- **Stimulus/Response Sequences**

- **Input**: There is a logout button on the client 's permissions page.
- **Output**: After selecting the button the client will exit from this page.

- **Functional Requirements**

REQ-47: There must be a button for the user to press when they want to log out.

REQ-48: After pressing the button, the client is redirected to the home page.

REQ-49: After the logout was successful, a pop-up message will be displayed to confirm that.

REQ-50: After 5 minutes of inactivity, the logout is done automatically.

• Contact:

- **Description and Priority**

If the user access this section, he can view the phone number and email address to access customer service. Also, user can send a ticket to the user support regarding any bugs/issues encountered while in the web application using “here” hyperlink from Register Notification.
Priority: **High**

- **Stimulus/Response Sequences**

- **Input**: The user enters the first name, last name, email, the subject regarding the problem and a send button.

- **Output**: This information will be first sent to the database where it will be stored and later displayed in the **Issues Page**.

- **Functional Requirements**

REQ-51: There must be a text field in which the user enters

the first name.

REQ-52: There must be a text field in which the user enters

the last name.

REQ-53: There must be a text field in which the user enters

the email.

REQ-54: There must be a text field in which the user enters the subject regarding the issues found.

REQ-55: There must be a Send button that the user must press after entering the data.

REQ-56: There must be a valid adress email.

REQ-57: After the ticket was successfully sent by the user, a pop-up message will be displayed to confirm that.

• About:

- **Description and Priority**

The user can see an essential source of information that can help him understand the functionality of the application.
Priority: **Low**

- **Stimulus/Response Sequences**

- **Input:** The user presses the ‘Homepage’ button.
- **Output:** The user lands on the home page.

- **Functional Requirements**

REQ-55: There is a ‘Homepage’ button that the user can press.

- ## Privacy:

- **Description and Priority**

The user can see the privacy and policy of the web application according to the regulations and data stored within the database.
Priority: **Low**

- **Stimulus/Response Sequences**
- **Input**: The user presses the “here” hyperlink.
- **Output**: The user lands on the external privacy data.

- **Functional Requirements**

REQ-57: There is a “here” hyperlink that the user can press.

- ## Home:

- **Description and Priority**

The user can see the essential details about the product that the web application is delivering. Priority: **Low**

- **Stimulus/Response Sequences**
- **Input:** The user can see/select any of the pages from the navigation bar.
- **Output:** The user reaches any of the pages that he chooses.

- **Functional Requirements**

REQ-58: If the user wants to go to any page, he can select the hyperlink to that page.

- ## Other Nonfunctional Requirements

Performance Requirements

ID Requirement	Requirement Specification
NFR-1	It is necessary for the application to display the requested data in 1-3 seconds.
NFR-2	The application should be able to handle a 10 of concurrent user requests without significant slowdown or downtime.
NFR-3	A transaction is processed within a maximum of 20 seconds.

Safety Requirements

Not applicable.

Security Requirements

- Users log-in using a unique account and password.
- The password will be encrypted in the database.
- The administrator can see the personal data of the user.
- Security: Being a banking application, security must be an absolute priority. We must consider using authentication, data encryption, and other security methods to protect users' data and prevent unauthorized access to accounts.

Software Quality Attributes

Coding rules:

- The increment (++) and decrement (--) operators should not be mixed with other operators in an expression.
- Identifiers shall be given for all of the parameters in a function prototype declaration.
- Each class must not have more than 1000 lines written.
- The public protected and private keywords must be used explicitly in the class declaration.
- Every class should have at least one constructor.
- Every 'if' must be followed by an 'else'.
- All statements must have open and closed accolades.
- Switch statements have the case at the same indentation as the switch.
- No identifier name should be reused.
- Sections of code should not be “commented out”.

Business Rules

- The bank may limit or restrict certain types of transactions or customers based on risk, compliance, or other factors.

- The bank may change the terms of service, fees, or other aspects of the application at any time, with appropriate notice to customers.
- Legal regulations: The application must comply with legal regulations and be compatible with laws and regulations regarding financial transactions, data protection, and others. For example, if the amount of this ttransaction is higher than 50.000 RON, it will be automatically reported to ONPCSB.

• Other Requirements

Not applicable.

Appendix A:

Glossary Appendix

B: Analysis Models

Appendix C: To Be Determined List