

Polytechnic of Porto
School of Engineering (ISEP)
BSc in Telecommunications and Informatics
Engineering

LETI-FSOFT

TECHNICAL REPORT - 2º ITERATION
Bank Account Management System

André Moreira (1240567)
Vasco Magolo (1231562)
Francisco Silva (1230985)
Bernardo Meireles (1232024)

1. Introduction	3
1.1 Context/Problem	4
2. Glossary	5
3. Domain Model	6
4. User Interface	7
5. Class Diagrams	8
5.1 Model Class Diagram	9
5.2 View Class Diagram	9
5.3 Controllers Class Diagram	10
5.4 Exceptions Class Diagram	10
6. Requirements	11
6.1 Non-Functional Requirements	11
6.2 Functional Requirements	12
6.2.1 Specification Use-Case Diagram	12
6.2.2 Specifications	13
6.2.3 System Sequence Diagram (SSD)	23
6.2.4 UC Sequence Diagram (SD)	24
6.2.4.1 UC-1 Create Account	24
6.2.4.2 UC-2 Authenticate User	25
6.2.4.3 UC-3 Deposit Money	25
6.2.4.4 UC-4 Withdraw Money	26
6.2.4.5 UC-5 Display Account Details	26
6.2.4.6 UC-6 Update Account Information	27
6.2.4.7 UC-7 Close Account	27
6.2.4.8 UC-8 List All Accounts (Admin Only)	28
6.2.4.9 UC-9 Request Balance Receipt	28
6.2.4.10 UC-10 Manage User Accounts (Admin)	29
7. References	30

1. Introduction

This project focuses on the development of a banking application designed to provide users with an intuitive and secure platform for managing their financial transactions and account information. Core functionalities include account creation, user authentication, fund deposits and withdrawals, account detail viewing, and saving account balances to a file.

The application emphasizes key principles of banking systems, such as data validation, user security, and robust interaction with account features. With a focus on reliability, efficiency, and scalability, the system delivers a user-friendly experience while ensuring strong safeguards for data integrity and privacy.

1.1 Context/Problem

- **Purpose:**
 - Software application for a bank with JSON-based data storage.
- **Features:**
 - **Bank Account:**
 - Has an account number, balance, age, name, address, and password.
 - Supports deposit and withdrawal operations.
 - Can update personal information (name, address, age, password).
 - Can request account balance receipts.
 - Can view all their account information.
 - Can delete their own account.
 - Account data is saved and loaded using JSON.
 - Each account has a unique account number.
 - **Admin:**
 - Has name and password attributes.
 - Can view all accounts in the system.
 - Can delete any customer account.
 - Can update information for any account.
 - Has elevated privileges for system management.
- **User Stories:**
 - **Bank Account Users:**
 - Customers can create a bank account with personal details.
 - Customers can login using their account number and password.
 - Customers can deposit and withdraw money from their accounts.
 - Customers can update their personal information (name, address, age, password).
 - Customers can request and view a receipt showing their current balance.
 - Customers can view all their account information in one place.
 - Customers can delete their own account when no longer needed.
 - **Admin Users:**
 - Admins can log in with special admin credentials.
 - Admins can view a list of all accounts in the system.
 - Admins can update information for any customer account.
 - Admins can delete any customer account when necessary.

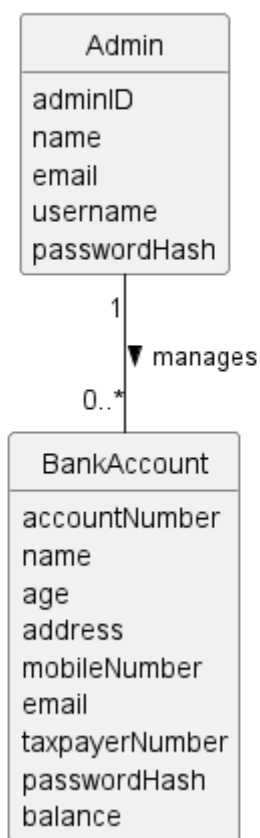
- **System:**
 - Account data is stored persistently in JSON format.
 - The system verifies password complexity and securely stores passwords.
- **Requirements:**
 - Data must be persistent using JSON file storage.
 - The system should properly update existing accounts rather than creating duplicates.
 - The user interface will be terminal-based (for now).
 - The application must follow an **MVC** (Model-View-Controller) architecture.
 - Account numbers must be unique and generated automatically.
 - Input validation must be implemented for numeric values.
 - Balance changes must update the correct account in storage.
 - Passwords must be stored using secure hashing algorithms rather than plaintext.
 - Password creation must enforce security rules: minimum 8 characters, containing at least one uppercase letter, one lowercase letter, one number, and one special character.

2. Glossary

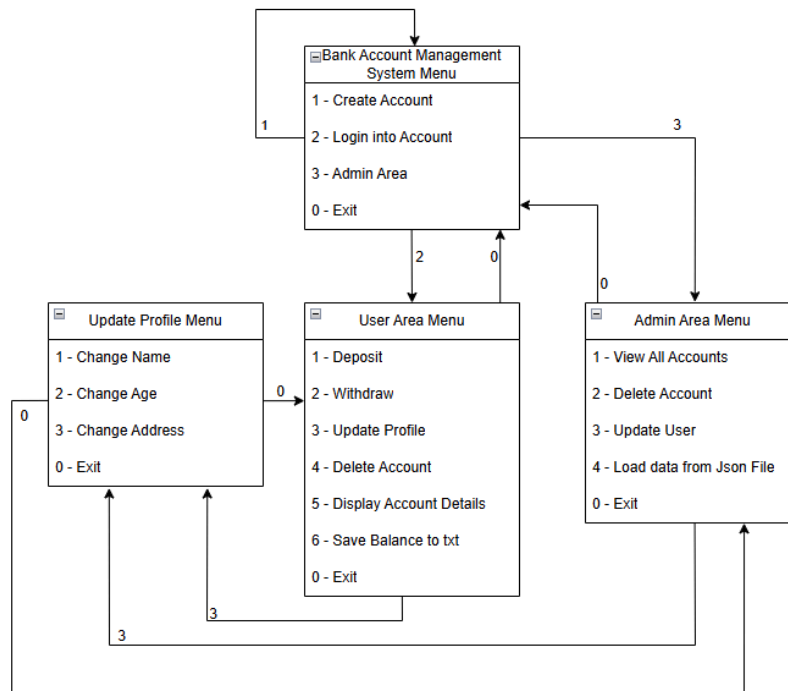
Term	Definition/Description
User	The one who uses the bank's features.
Admin	The one with extra permissions that allow him to access and modify bank accounts even if they don't belong to him.
Json file	File that stores the user bank account data.
Bank	The entity that provides services such as creating a bank account, deposits, and withdrawals.
Bank Account	Directly used by the customer through the bank's interface.

3. Domain Model

Domain Model - Bank Account Managment System

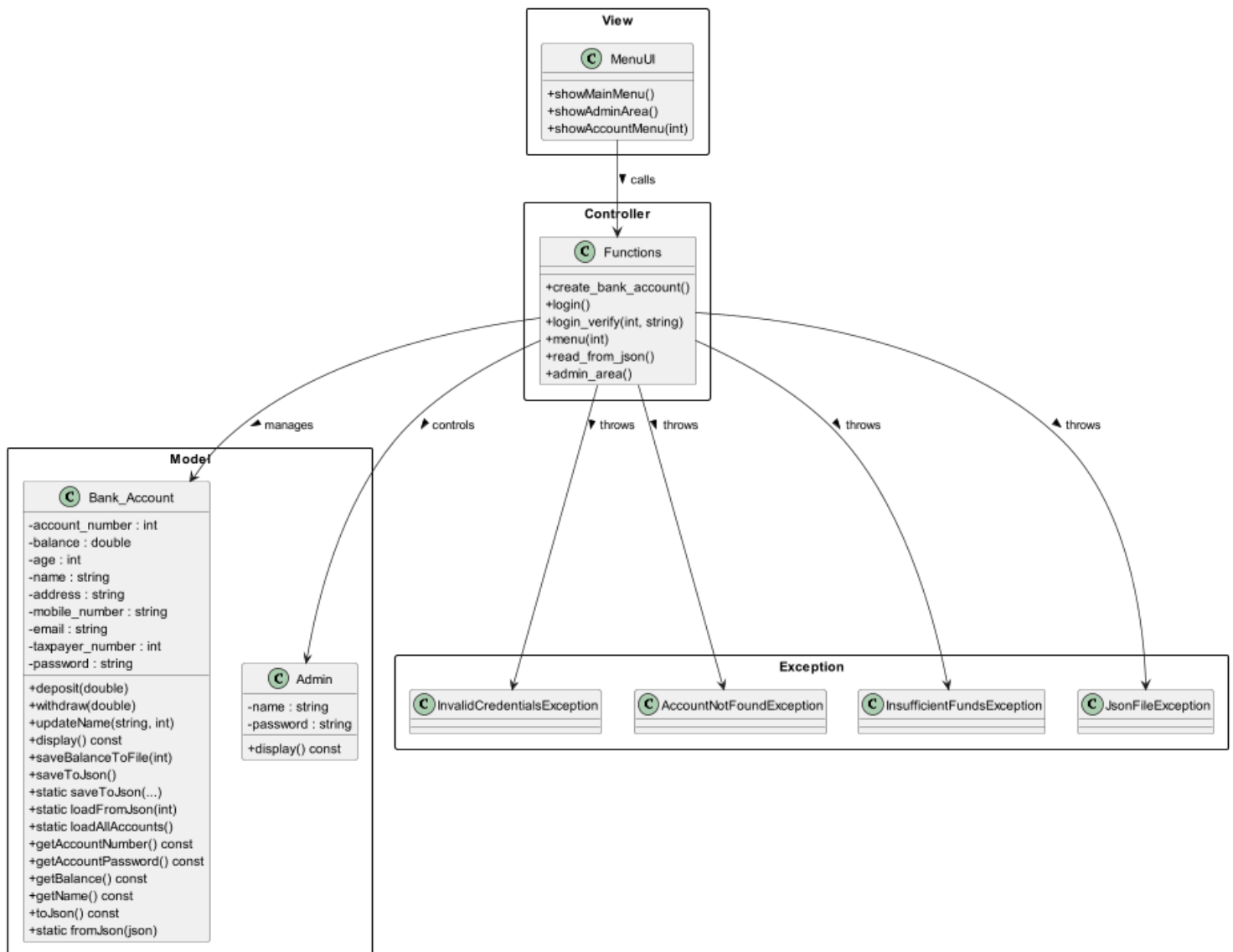


4. User Interface

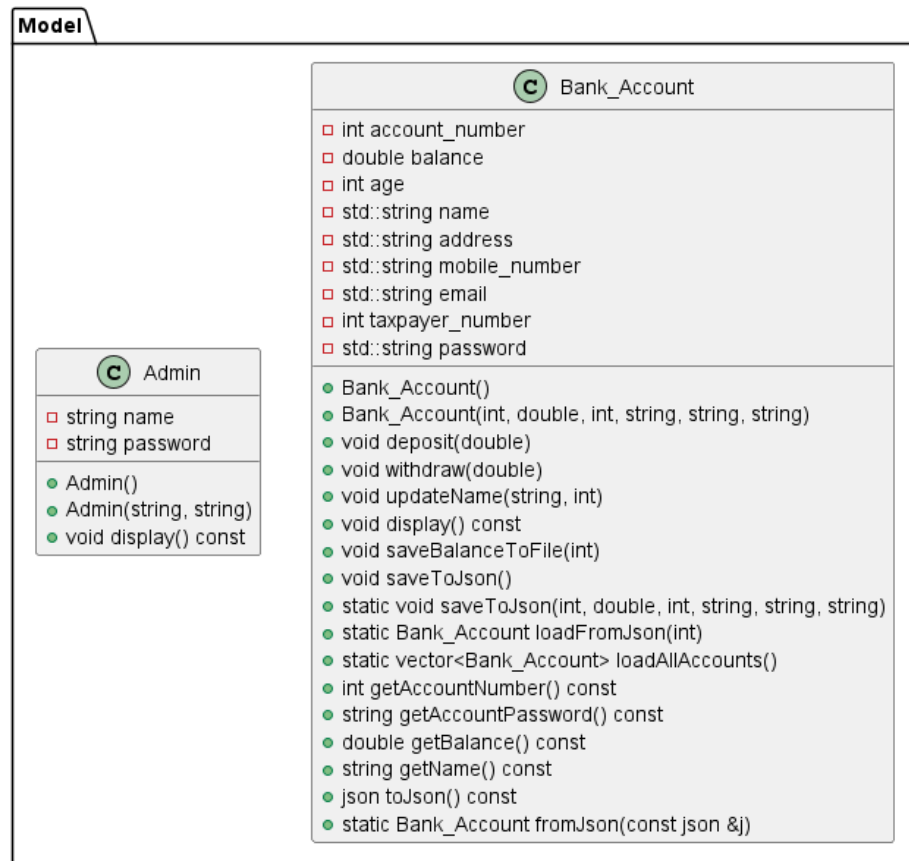


5. Class Diagrams

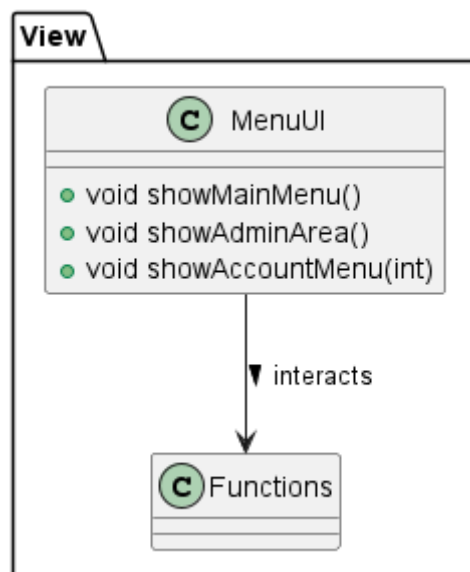
Class Diagram - Bank Account Management System



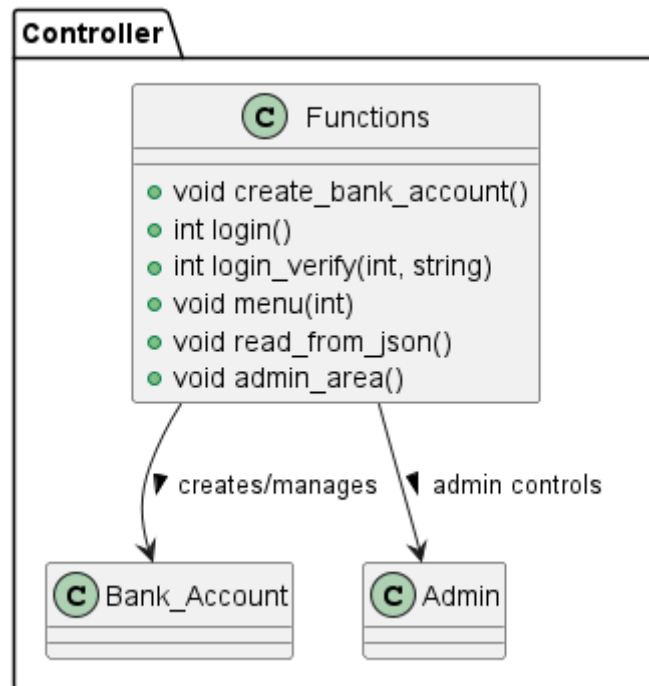
5.1 Model Class Diagram



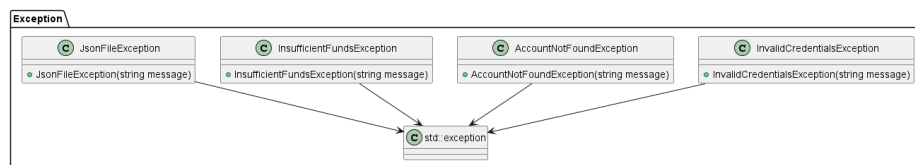
5.2 View Class Diagram



5.3 Controllers Class Diagram



5.4 Exceptions Class Diagram



6. Requirements

6.1 Non-Functional Requirements

Usability

- The program runs in the terminal/console. It's kept simple and easy to navigate for both users and admins.

Reliability

- The app is expected to work properly during normal use.

Performance

- No special performance needed.

Support / Maintainability

- A good number of tests will be made to help catch bugs and make updates easier later on.

Design Constraints

- All account and transaction data will be saved using JSON files to keep things organized and easy to manage.

Implementation Constraints

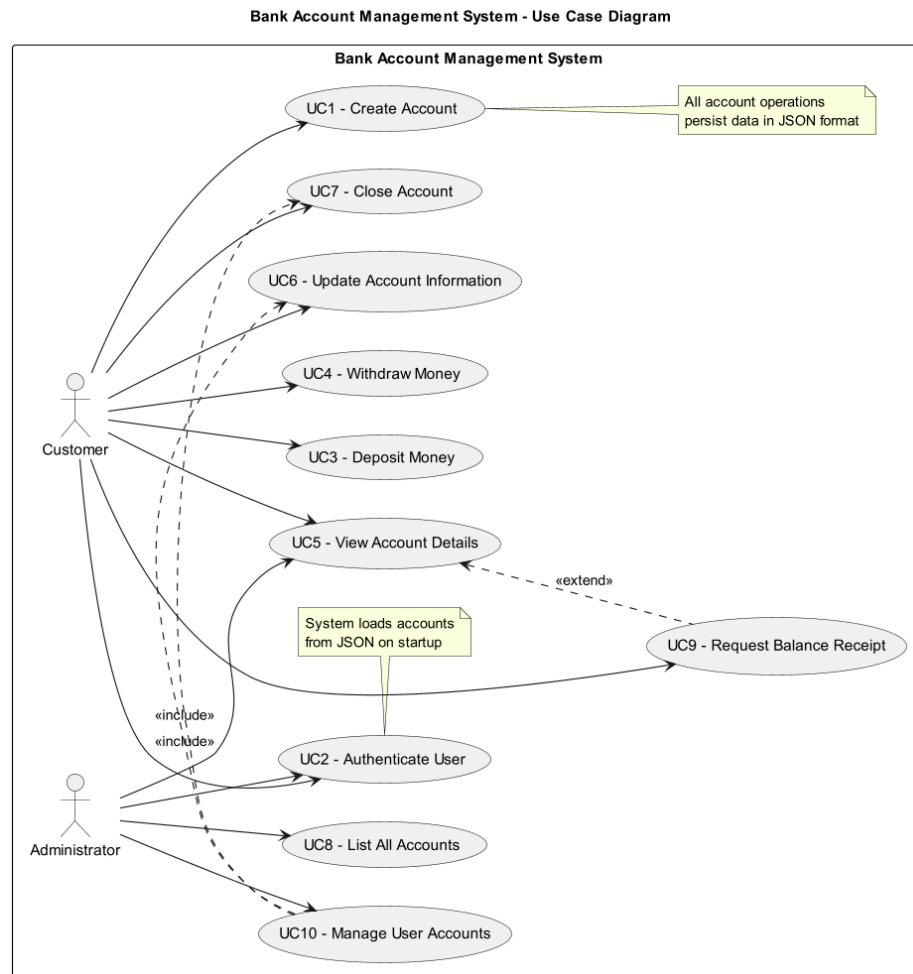
- The whole project will be built in C++, and the [nlohmann/json] library is used for working with JSON.

Interface Constraints

- The program doesn't connect to other systems or need any special external interfaces.

6.2 Functional Requirements

6.2.1 Specification Use-Case Diagram



6.2.2 Specifications

Specification: UC 1 – Create Account

Description	Create a new bank account in the system
Pre-condition	There are no pre-condition associated with this use case
Post-condition	A new bank account is created with a unique number
Basic path	<ul style="list-style-type: none">· User selects “Create Account” option· User inputs name, age, address and password· The System validates the user data· The System generates a unique account number· The System stores the user data· The System returns success with account details
Alternative Path	<ul style="list-style-type: none">· Invalid age (under 18)<ul style="list-style-type: none">o The System displays error messageo The System prompts user to enter valid data

Specification: UC 2 – Authenticate User

Description	User logs into an existing account
Pre-condition	Account exists in system
Post-condition	User gains access to account options
Basic path	<ul style="list-style-type: none">· User selects “Login into account”· System prompts for account number and password· User enters details· System checks if account exists and credentials match account details· System displays available account options
Alternative Path	<ul style="list-style-type: none">· Invalid account number or password not matching<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

Specification: UC 3 – Deposit Money

Description	Adds funds to an existing account
Pre-condition	Account exists in system
Post-condition	Account balance is increase The transaction is recorded
Basic path	<ul style="list-style-type: none">· User selects “Deposit” option· System prompts for deposit amount· User enters deposit amount· System checks if deposit amount is positive· System updates account balance· System records transaction· System displays new account balance
Alternative Path	<ul style="list-style-type: none">· Invalid deposit amount<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

Specification: UC 4 – Withdraw Money

Description	Withdraw funds from an existing account
Pre-condition	Account has sufficient balance
Post-condition	Account balance is decreased by withdraw amount Transaction is recorded
Basic path	<ul style="list-style-type: none">· User selects “Withdraw” option· System prompts for withdrawal amount· User inputs withdrawal amount· System validates if account has sufficient funds· System updates account balance· System records transaction· System displays new account balance
Alternative Path	<ul style="list-style-type: none">· Invalid withdrawal amount<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

Specification: UC 5 – Display Account Details

Description	Displays details from an existing account
Pre-condition	Account exists in the system
Post-condition	Account information is displayed
Basic path	<ul style="list-style-type: none">· User selects “Display Account” option· System prompts for account number· User enters account number· System validates if account exists· System retrieves and displays account details
Alternative Path	<ul style="list-style-type: none">· Invalid account number<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

Specification: UC 6 – Update Account Information (Admin and User)

Description	Users can edit their personal information
Pre-condition	Account exists in the system and user has access to the account
Post-condition	Account details are updated
Basic path	<ul style="list-style-type: none">· User selects “Update Account Information” option· User inputs the new details· System validates new details· System saves new details· System returns success
Alternative Path	<ul style="list-style-type: none">· Invalid account details<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

· **Specification: UC 7 – Close Account**

Description	Users can close their account
Pre-condition	Account exists in the system and user has access to the account
Post-condition	Account will be removed or marked as inactive in the system
Basic path	<ul style="list-style-type: none">· User selects “Close Account” option· System checks if account can be closed (has no funds in it)· System shows confirmation message· User confirms· System closes account
Alternative Path	<ul style="list-style-type: none">· Account can’t be closed<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

Specification: UC 8 – List All Accounts (Admin Only)

Description	Admin users can view all accounts registered in the system
Pre-condition	User must be logged in as an administrator
Post-Condition	The System displays a complete list of all registered accounts
Basic path	<ul style="list-style-type: none">· System verifies user permissions· Admin selects “List All Accounts” option· System shows a list of accounts
Alternative Path	<ul style="list-style-type: none">· User is not logged in as an administrator or there are no accounts<ul style="list-style-type: none">o The System displays error messageo The System prompts user to enter valid data

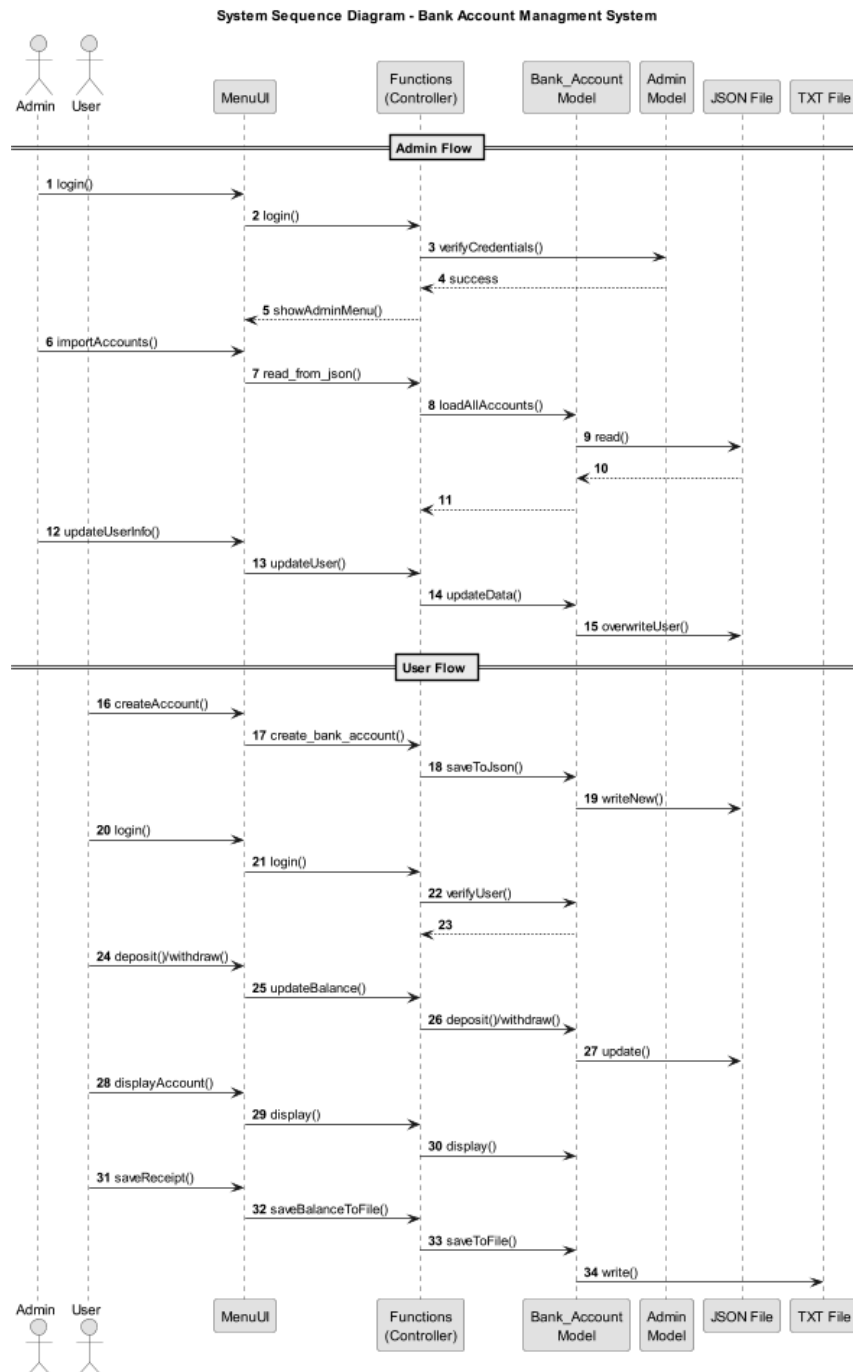
Specification: UC 9 – Request Balance Receipt

Description	Exports account balance into a txt file like a receipt
Pre-condition	Account exists in the system
Post-condition	None
Basic path	<ul style="list-style-type: none">· User selects “Request Balance Receipt” option· System exports balance into a txt file· System shows txt file
Alternative Path	None

Specification: UC 10 – Manage User Accounts (Admin)

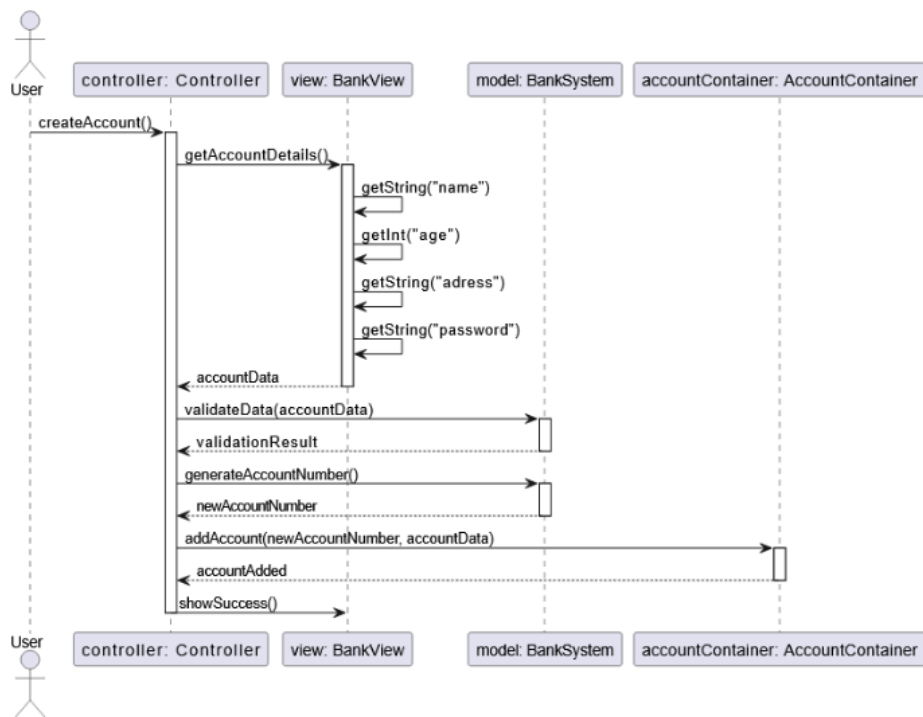
Description	Admins can update or delete accounts
Pre-condition	Account exists in the system
Post-condition	Account will be update or removed from the system
Basic path	<ul style="list-style-type: none">· Admins selects “Manage User Accounts” option· If Admin selects “Update Account”<ol style="list-style-type: none">1. Admin inputs new details2. System validates new details3. System saves new details· If Admin selects “Close account”<ol style="list-style-type: none">1. System shows confirmation message2. System confirms if account can be closed3. System closes accounts· System shows txt file
Alternative Path	<ul style="list-style-type: none">· Account can’t be closed or new details are invalid<ul style="list-style-type: none">○ The System displays error message○ The System prompts user to enter valid data

6.2.3 System Sequence Diagram (SSD)

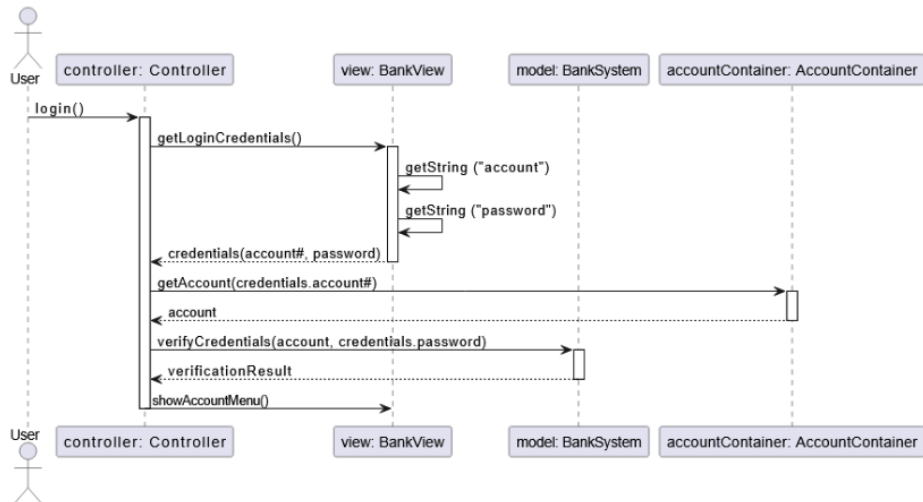


6.2.4 UC Sequence Diagram (SD)

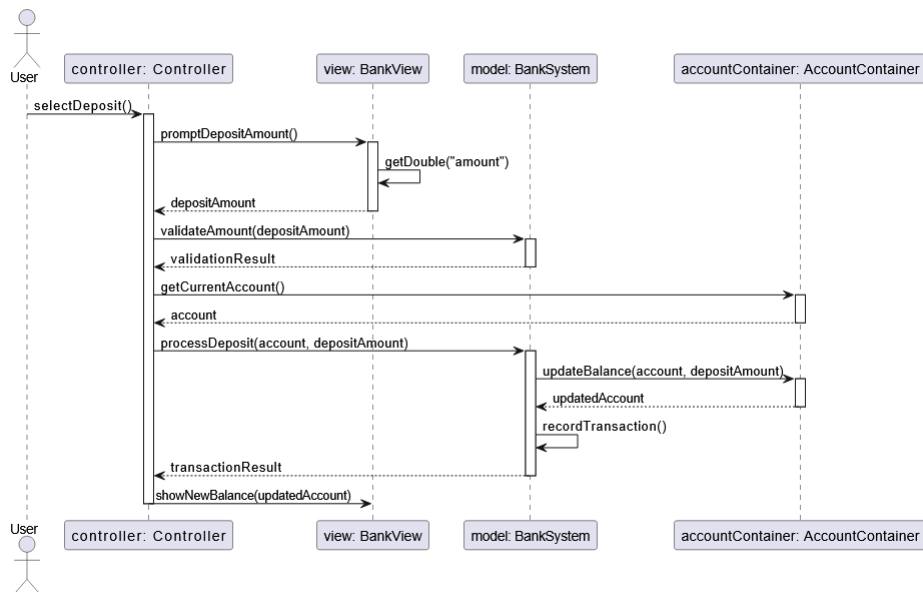
6.2.4.1 UC-1 Create Account



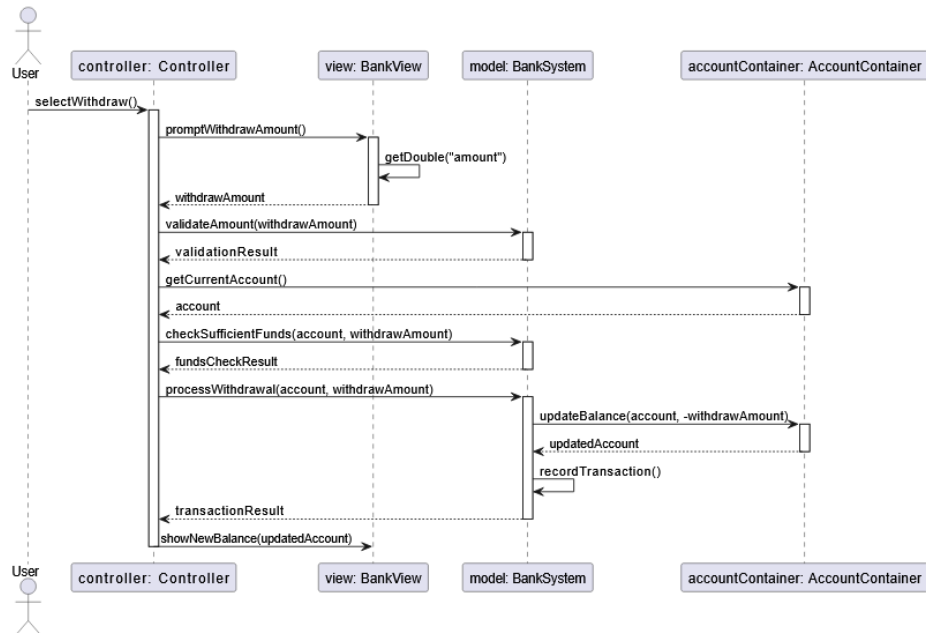
6.2.4.2 UC-2 Authenticate User



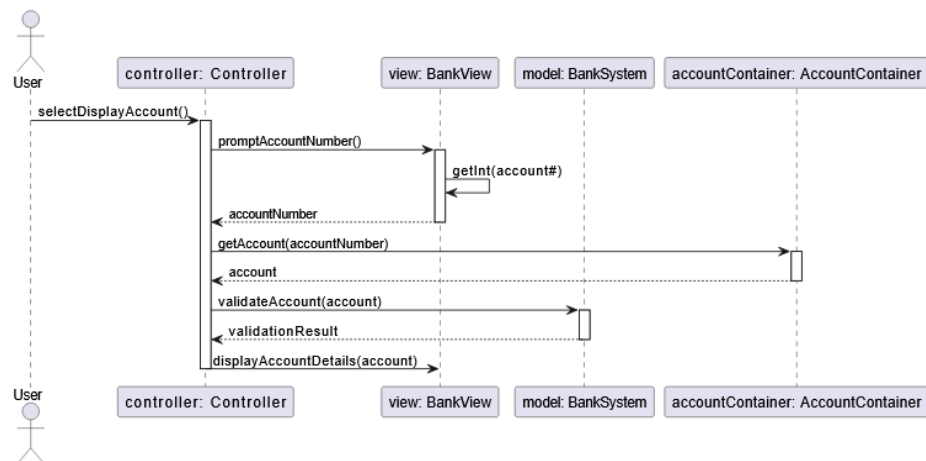
6.2.4.3 UC-3 Deposit Money



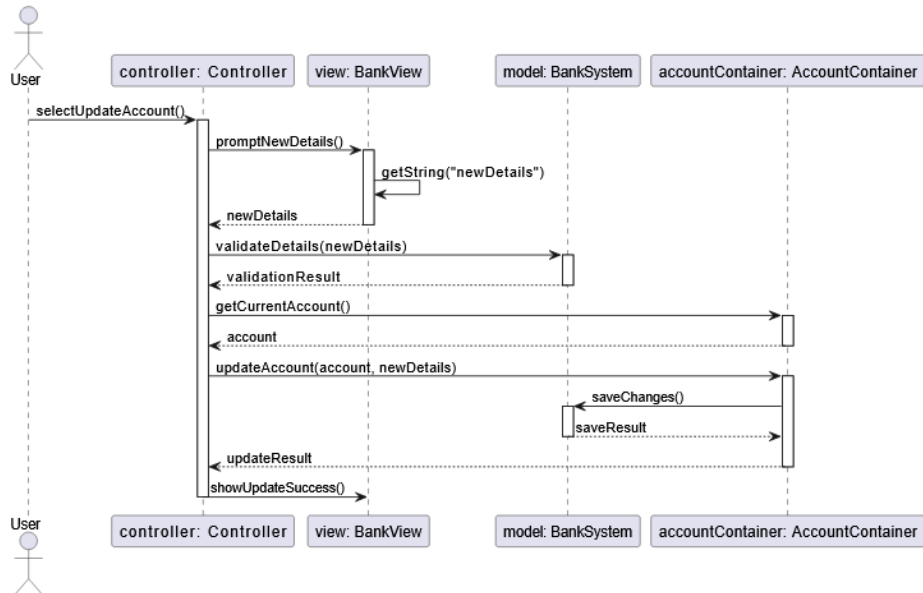
6.2.4.4 UC-4 Withdraw Money



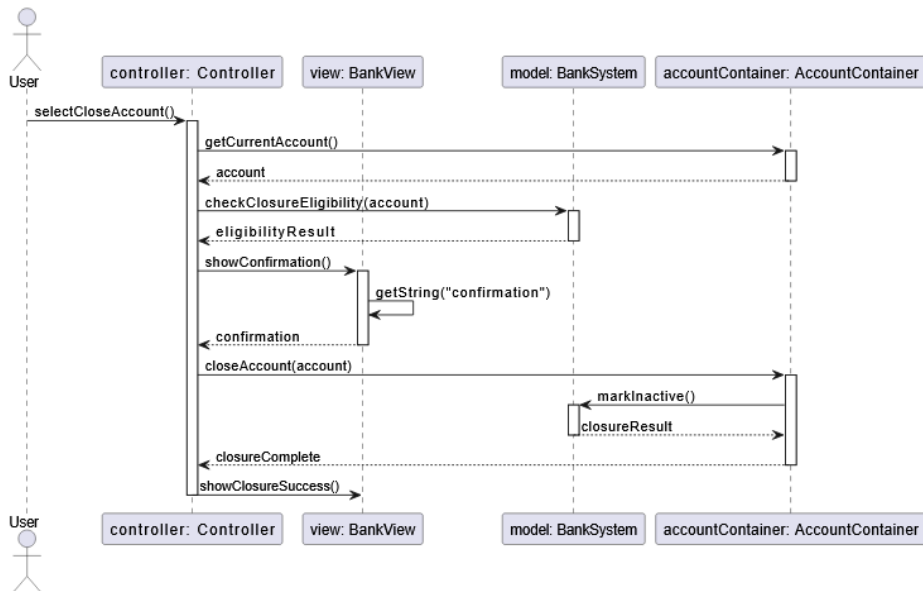
6.2.4.5 UC-5 Display Account Details



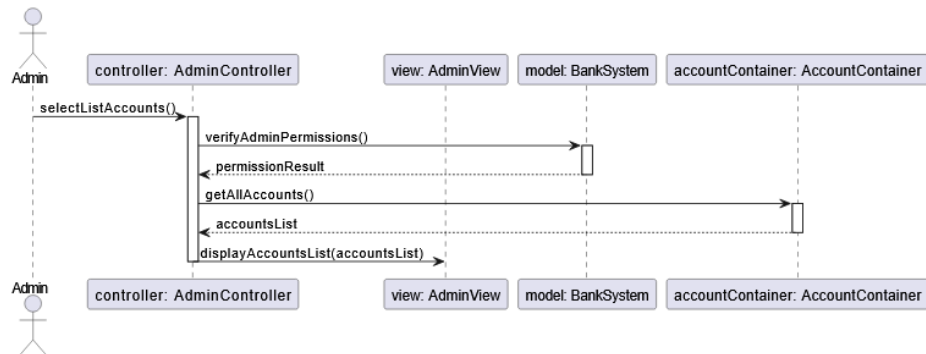
6.2.4.6 UC-6 Update Account Information



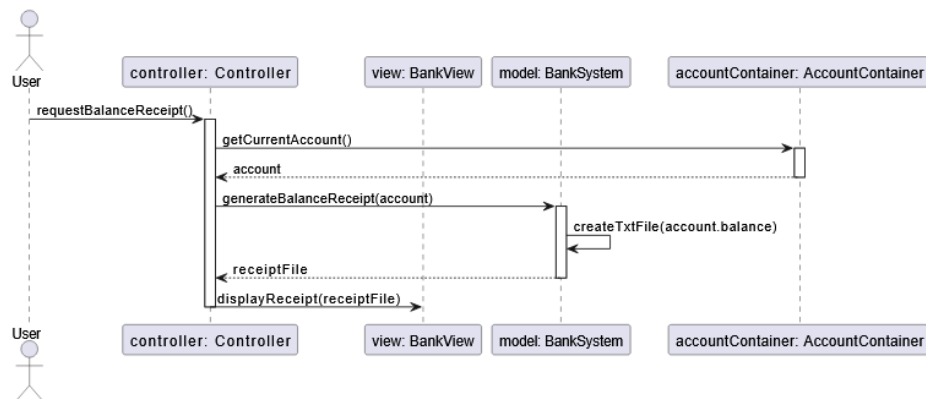
6.2.4.7 UC-7 Close Account



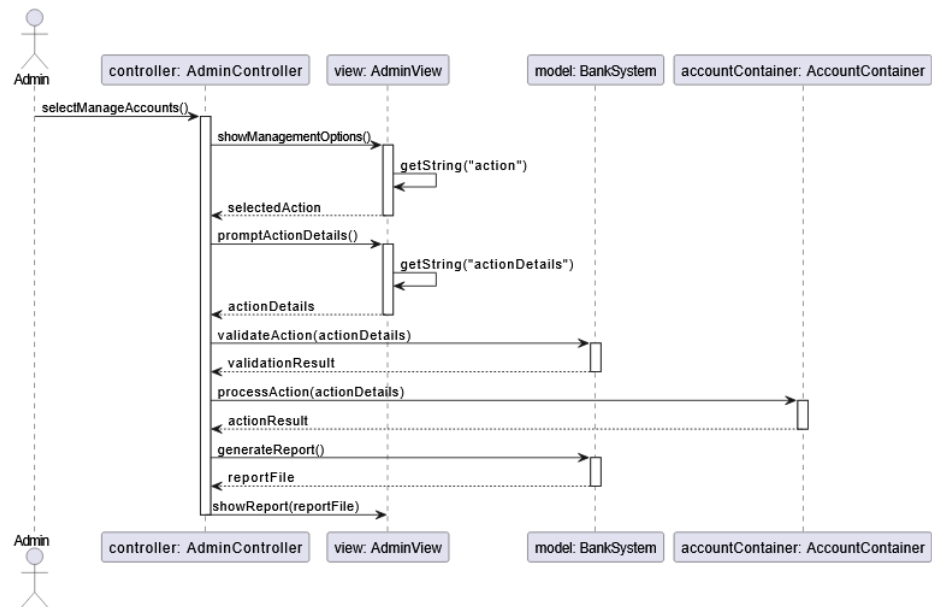
6.2.4.8 UC-8 List All Accounts (Admin Only)



6.2.4.9 UC-9 Request Balance Receipt



6.2.4.10 UC-10 Manage User Accounts (Admin)



7. References

1. Sousa, P. B. (2024/25). Lecture slides for FSOF – Foundations of Software Engineering. Bachelor's in Telecommunications and Informatics Engineering, Instituto Superior de Engenharia do Porto (ISEP).