Polytechnic of Porto

School of Engineering (ISEP)

BSc in Telecommunications and Informatics

Engineering

LETI-FSOFT

TECNICAL REPORT - 1º ITERATION

Software Application for a Bank

André Moreira (1240567)

Vasco Magolo (1231562)

Francisco Silva (1230985)

Bernardo Meireles (1232024)

April 2025

Contents

Abstract

This project involves the development of a banking application aimed at providing users with a simple way to manage their financial transactions and account details. The application enables users to perform a range of essential banking functions such as creating a new account, logging into an existing account, depositing and withdrawing money, displaying account details, and saving account balances to a file.

This project utilizes core banking system principles, focusing on user security, data validation, and seamless interaction with account-related functions. It aims to offer an efficient, reliable, and user-friendly application for managing finances, with a robust design that ensures scalability and security.

Context/Problem

- **Purpose:**
  - Software application for a bank.
- **Features:**
  - A bank account has an account number, balance, age, name, address, and associated password.
  - The bank has a name.
  - The customer has a name, age, and address.
- **User Stories:**
  - The customer can create a bank account or log in using their details to access a bank account.
  - The file is responsible for storing the data of a bank account in a file.
- **Requirements:**
  - Data must be persistent.
  - The user interface will be terminal-based.

Domain Model

- **Entities**

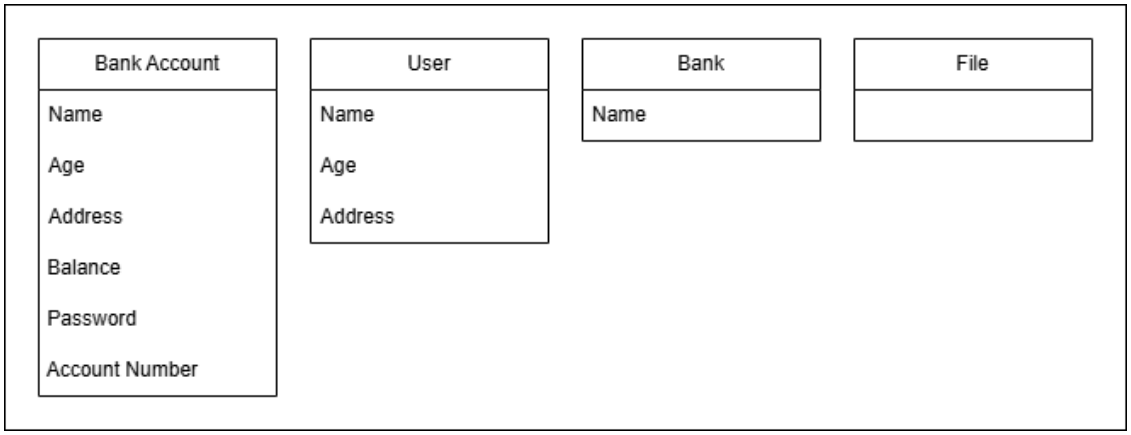

Figure 1 - Entities

Glossary

Term	Definition/Description
Customer	The one who uses the bank's features.
Bank	The entity that provides services such as creating a bank account, deposits, and withdrawals.
File	Responsible for saving the bank account data in a file.
Bank account	Directly used by the customer through the bank's interface.

Table 1 - Glossary

Non-Functional Requirements

- **Usability**
  - The user interface will be terminal-based.
- **Reliability**
  - No
- **Performance**
  - No
- **Support Capacity**
  - Huge amount of tests.
- **Design Constraints**
  - Data persistent in binary files.
- **Implementation Constraints**
  - C++ language
- **Interface Constraints**
  - No

Functional Requirements

Functionalities

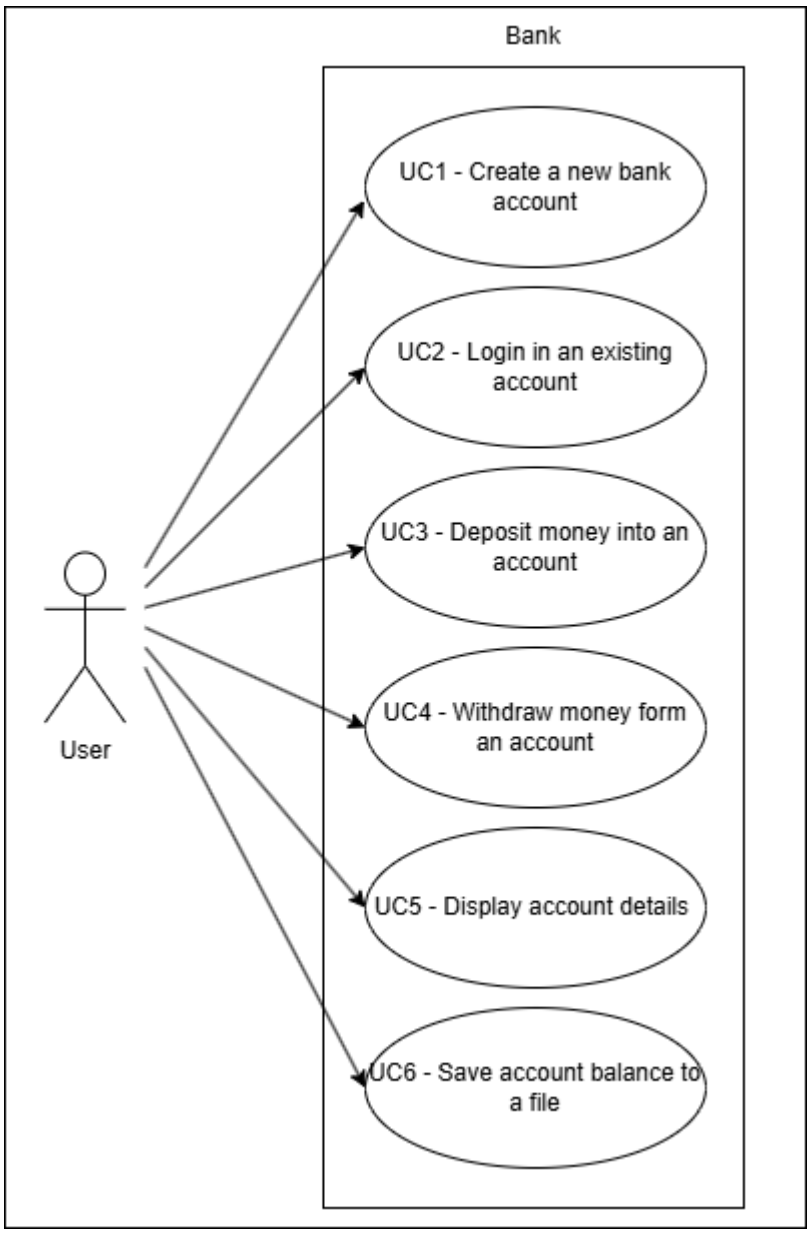

Figure 2 - Functionalities

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	• User selects “Create Account” option • User inputs the required details • 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	• Invalid data
	  ○ The System displays error message
	  ○ The System prompts user to enter valid data

- **SSD: UC 1 – Create Account**

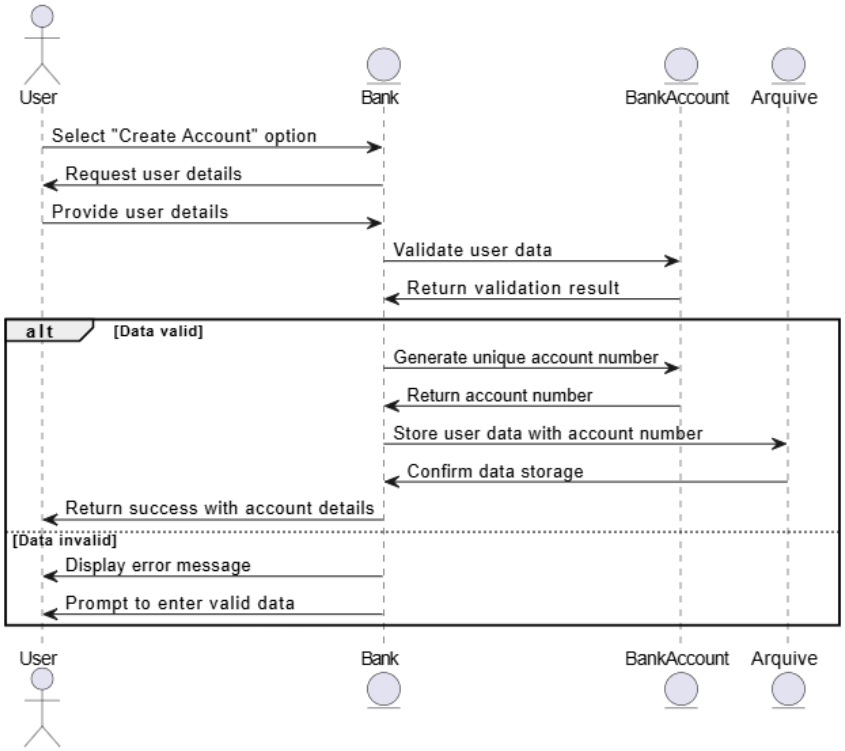

Figure 3 – SSD: Create Account

- **Specification: UC 2 – Login into Account**

Description	• User logins into an existing account
Pre-condition	• Account exists in system
Post-condition	• User gains access to account options
Basic path	• User selects “Login into account” • System prompts for: • Name • Address • Password • User enters details • System checks if • Account exists • Credentials match account details • System displays available account options

Alternative Path	• Invalid data
	  ○ The System displays error message
	  ○ The System prompts user to enter valid data

- **SSD: UC 2 – Login into Account**

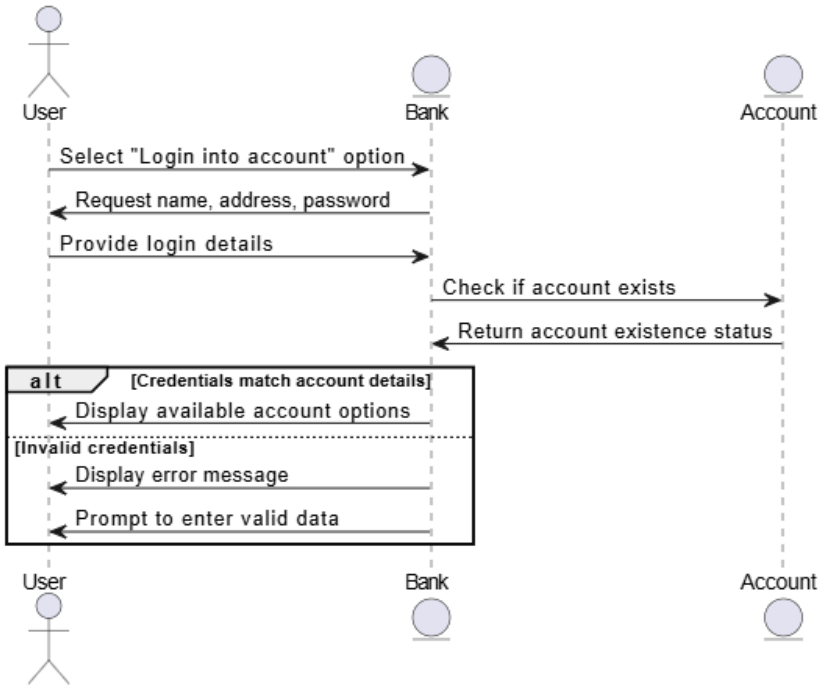

Figure 4 – SSD: Login into Account

- **Specification: UC 3 – Deposit Funds**

Description	• Adds funds to an existing account
Pre-condition	• Account exists in system
Post-condition	• Account balance is increased • The transaction is recorded
Basic path	• User selects “deposit money” • 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	• Invalid data
	  ○ The System displays error message
	  ○ The System prompts user to enter valid data

- **SSD: UC 3 – Deposit Funds**

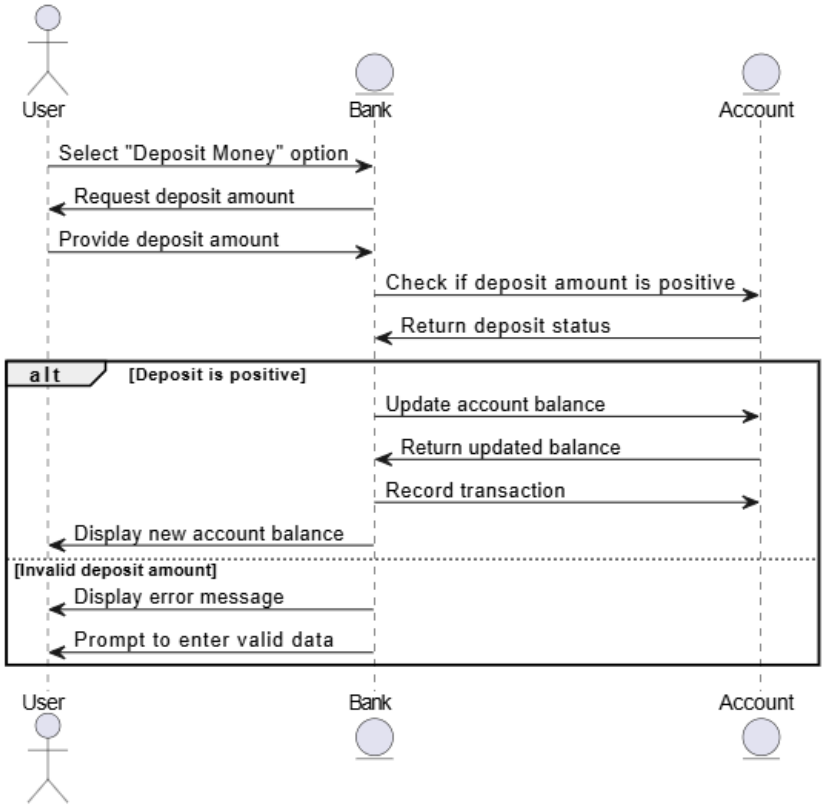

Figure 5 – SSD: Deposit Funds

- **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	• 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	• Invalid data
	  ○ The System displays error message
	  ○ The System prompts user to enter valid data

- **SSD: UC 4 – Withdraw Money**

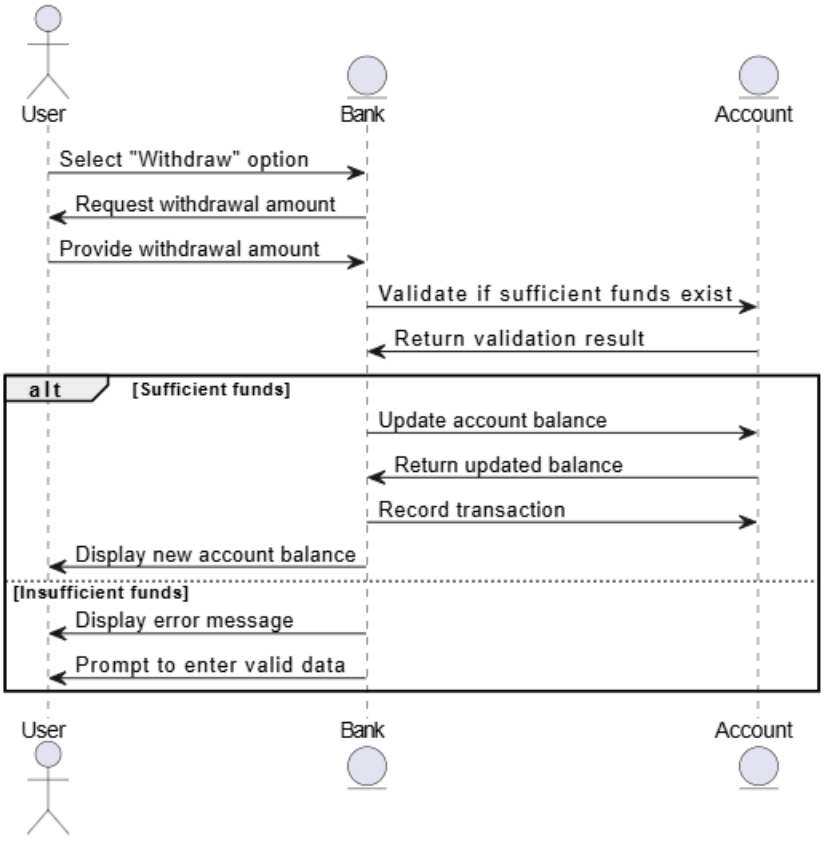

Figure 6 – SSD: Withdraw Money

- **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	• 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	• Invalid data
	  ○ The System displays error message
	  ○ The System prompts user to enter valid data

- **SSD: UC 5 – Display Accounts Details**

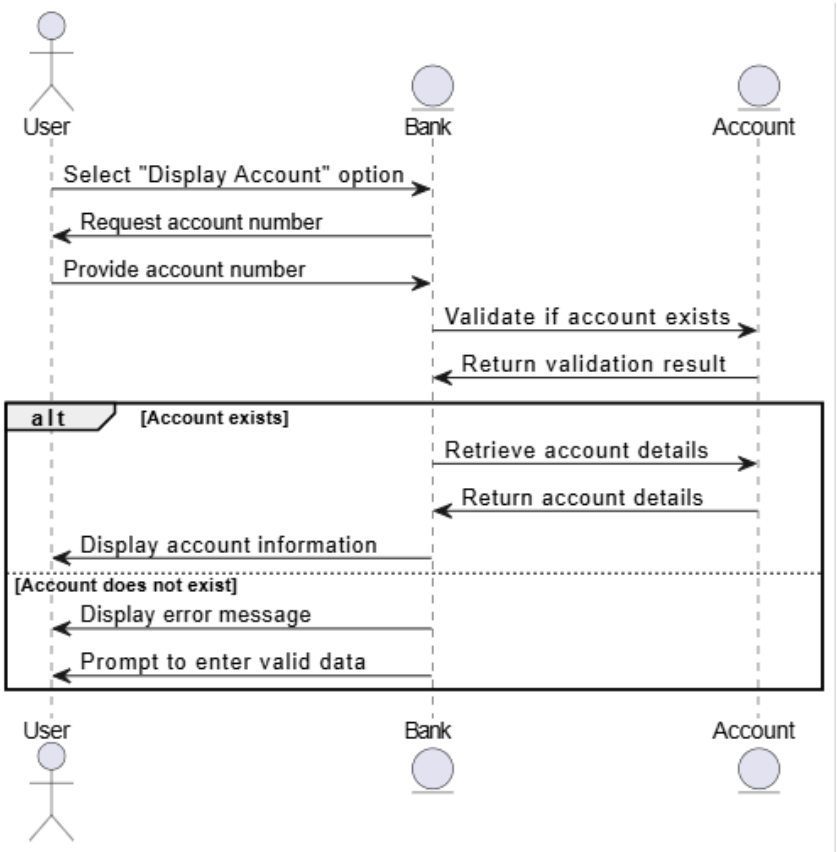

Figure 7 – SSD: Display Account Details

- **Specification: UC 6 – Save Account Details**

Description	Save account details to a file
Pre-condition	Account exists in the system
Post-condition	Account details are written to a file
Basic path	• User selects “Save to File” option • System creates file • System writes account details to file • System returns success

Alternative Path	• Invalid data
	  ○ The System displays error message
	  ○ The System prompts user to enter valid data

- **SSD: UC 6 – Save Account Details**

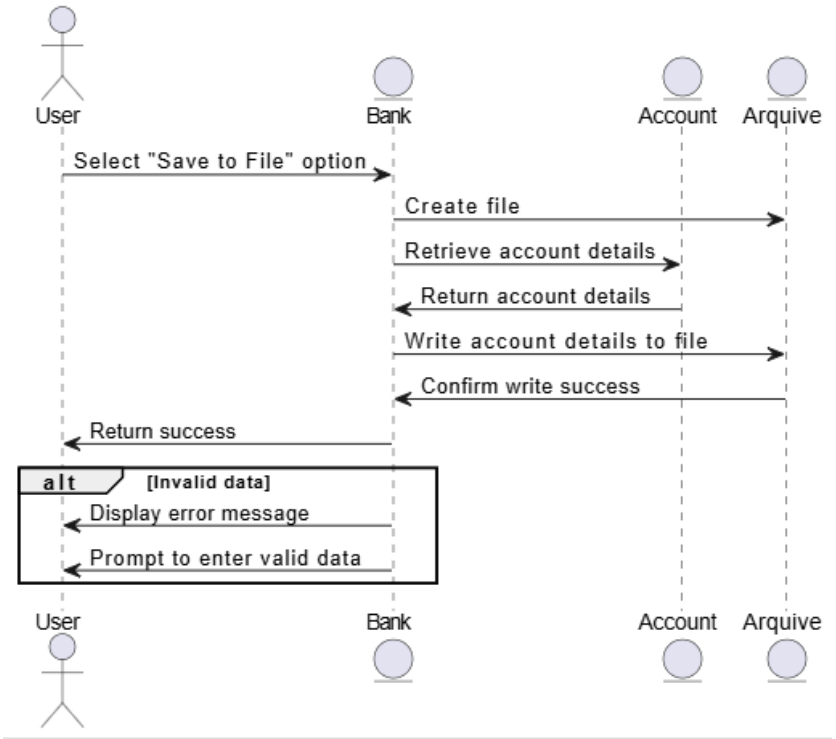

Figure 8 – SSD: Save Account Details