Software Requirements Specification for ATM Simulation System

Contents

1 Introduction

1.1 Purpose

This document explains the requirements for a simple ATM Simulation System. It is designed for a student project to mimic basic ATM functions like withdrawing money, checking balance, and changing PIN. It helps developers and testers understand what the system should do.

1.2 Scope

The ATM Simulation System is a simple software program that lets users pretend to use an ATM. It includes features like logging in, withdrawing money, depositing money, checking account balance, and changing PIN. The system uses a fake database to store user and account details. It is meant for learning purposes and does not connect to real banks.

1.3 Definitions

- **ATM**: Automated Teller Machine (a pretend version for this project).
- **PIN**: Personal Identification Number (a 4-digit code to log in).
- **GUI**: Graphical User Interface (the screen users interact with).
- **SRS**: Software Requirements Specification (this document).
- **UML**: Unified Modeling Language (used for diagrams to show system interactions).

1.4 Overview

Section 1 explains the purpose and scope. Section 2 describes the system and its features. Section 3 lists the interface requirements. Section 4 details the main features. Section 5 covers other requirements. Section 6 includes a UML diagram.

2 Overall Description

2.1 Product Perspective

The ATM Simulation System is a standalone software program with a simple user interface that looks like an ATM screen. It uses a fake database to store user details like account number, PIN, and balance.

2.2 Product Functions

The system supports these main functions:

- **Login**: Users enter their account number and PIN to access the system.
- **Cash Withdrawal**: Users can withdraw a fixed amount of money (e.g., $20, $50).
- **Deposit**: Users can add money to their account.
- **Balance Inquiry**: Users can check their account balance.
- **PIN Change**: Users can update their PIN.
- **Logout**: Users can exit the system safely.

2.3 User Characteristics

- **Users**: Students or beginners who act as bank customers, using the system to learn how an ATM works.
- **Admin**: A teacher or developer who can reset the system or add fake user data.

2.4 Constraints

- Only one user can use the system at a time.
- Users get three tries to enter the correct PIN before the system locks them out.
- Minimum withdrawal: $10; maximum withdrawal: $500 per transaction.
- Maximum deposit: $1000 per transaction.

2.5 Assumptions

- The system uses a simple fake database (e.g., a text file or list) to store data.
- All users have a valid account number and PIN in the fake database.

3 External Interface Requirements

3.1 User Interfaces

- **Screen**: A simple window (GUI) showing options like ”Withdraw,” ”Deposit,” ”Check Balance,” and ”Change PIN.”
- **Input**: Users type their account number, PIN, and amounts using a keyboard or buttons on the screen.
- **Output**: The screen shows messages like ”Login Successful,” ”Balance: $100,” or ”Error: Wrong PIN.”

3.2 Software Interfaces

- **Database**: A simple file or in-memory list to store account numbers, PINs, and balances.
- **Program**: The system is built using a simple programming language like Python, Java, or C++.

4 System Features

4.1 User Login

- **Description**: Users enter their account number and PIN to log in.
- **Priority**: High
- **Input**: Account number (e.g., 123456), 4-digit PIN (e.g., 1234).
- **Process**: Checks if the account number and PIN match the fake database. Locks account after three wrong PIN attempts.
- **Output**: ”Login Successful” or ”Wrong PIN” message.

4.2 Cash Withdrawal

- **Description**: Users withdraw money from their account.
- **Priority**: High
- **Input**: Amount to withdraw (e.g., $20, $50).
- **Process**: Checks if the account has enough money. Updates balance in the fake database.
- **Output**: ”Withdrawal Successful, New Balance: $X” or ”Insufficient Funds” message.

4.3 Deposit

- **Description**: Users add money to their account.
- **Priority**: Medium
- **Input**: Amount to deposit (e.g., $100).
- **Process**: Adds the amount to the account balance in the fake database.
- **Output**: ”Deposit Successful, New Balance: $X” message.

4.4 Balance Inquiry

- **Description**: Users check their account balance.
- **Priority**: Medium
- **Input**: None (after login).
- **Process**: Retrieves balance from the fake database.
- **Output**: ”Your Balance: $X” message.

4.5 PIN Change

- **Description**: Users change their PIN.
- **Priority**: Medium
- **Input**: Old PIN, new PIN, confirm new PIN.
- **Process**: Verifies old PIN and updates to new PIN in the fake database if the new PIN matches the confirmation.
- **Output**: ”PIN Changed Successfully” or ”Error: PIN Mismatch” message.

5 Other Requirements

5.1 Performance

- The system responds to user actions (e.g., login, withdraw) within 2 seconds.
- It handles one user at a time.

5.2 Security

- PINs are stored securely in the fake database (e.g., not shown in plain text).
- The system logs out users after 1 minute of no activity.

5.3 Usability

- The interface is simple with clear buttons and messages.
- Error messages are easy to understand (e.g., ”Wrong PIN, try again”).

5.4 Documentation

- A short user guide explains how to use the system.

- A developer guide explains how to set up the fake database and run the program.

6 UML Diagram

6.1 Use Case Diagram

The following UML Use Case Diagram shows how users (Customer and Admin) interact with the ATM Simulation System. It includes the main functions described in Section 4.

```
graph TD
    A[Customer] -->|Login| B[ATM Simulation System]
    A -->|Cash Withdrawal| B
    A -->|Deposit| B
    A -->|Balance Inquiry| B
    A -->|PIN Change| B
    A -->|Logout| B
    C[Admin] -->|Manage Database| B
    C -->|Reset System| B
```

Explanation:

- **Actors**:
  - **Customer**: Uses the ATM to perform transactions like login, withdraw, deposit, check balance, change PIN, and logout.
  - **Admin**: Manages the fake database (e.g., adds user data) and resets the system.
- **Use Cases**: Represent the system's main functions (Login, Cash Withdrawal, Deposit, Balance Inquiry, PIN Change, Logout, Manage Database, Reset System).
- **Relationships**: Arrows show which actor interacts with each function.

7 Appendix: Glossary

- **ATM**: A pretend machine for banking tasks.
- **Fake Database**: A simple file or list storing user account details.
- **PIN**: A 4-digit code to log in.
- **UML**: A way to draw diagrams to show how the system works.