**Software Engineering Lab Experiment No. 3**

**Aim:** Create an SRS for the project

**Objectives:**

1. Capture and Document User Needs.
2. Enhance Project Transparency.
3. Use the SRS to develop use case.
4. Use the information gathered in the SRS to create UML diagrams, such as class diagrams.

**Requirements:**

1. Computer with internet access.
2. A Figma account.
3. Sample software project or problem statement for requirements analysis.
4. Word processing software for creating the lab report & SRS document.

**Tools:**

* **Figma:** Figma is a web-based design tool that allows users to create interactive user interface prototypes. It enables real-time collaboration and can be accessed from anywhere and on any device.
* **Word document** for creating the lab report & SRS document.

**Table of Contents**

1. **INTRODUCTION**

**1.1** PURPOSE

**1.2** SCOPE

**1.3** DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

**1.4** REFERENCES

**1.5** OVERVIEW

1. **GENERAL DESCRIPTION**

**2.1** PRODUCT PERSPECTIVE

* 1. ASSUMPTIONS AND DEPENDENCIES

1. **SPECIFIC REQUIREMENTS**

**3.1** FUNCTIONAL REQUIREMENTS

**3.1.1** Account Creation

**3.1.2** Deposit

* 1. USE CASES

**3.2.1** Use Case #1: Account Registration

**3.2.2** Use Case #2: Deposit Funds

**3.3** CLASSES / OBJECTS

**3.3.1** User

**3.3.2** Account

1. **ANALYSIS MODELS** 
   1. UNIFIED MODELING LANGUAGE (UML)
2. **USE CASE**

1. **Introduction**
   1. **Purpose**

The purpose of this Software Requirements Specification (SRS) is to provide a detailed description of the Account Transaction System, outlining its features, functionality, and requirements.

* 1. **Scope**

The Account Transaction System is designed to facilitate account management, allowing users to perform various banking transactions. This system will cover the core functionality of account creation, deposit, withdrawal, balance inquiry, and fund transfer.

* 1. **Definitions, Acronyms, and Abbreviations**

- SRS: Software Requirements Specification

- PIN: Personal Identification Number

* 1. **References**

**-** Used IEEE SRS template for creating SRS document.

* 1. **Overview**

The Account Transaction System is a web-based application that enables customers to manage their bank accounts efficiently. Users can perform basic banking transactions from the comfort of their homes or on the go, ensuring the security and reliability of their financial operations.

1. **General Description**
   1. **Product Perspective**

The Account Transaction System is a standalone application that interacts with a central banking database for account management. It is not dependent on any other external systems or applications.

**2.2 Assumptions and Dependencies**

- Users have a valid account with the bank.

- Secure authentication mechanisms are in place, including username and PIN.

- The system relies on a stable internet connection for real-time transaction processing.

1. **Specific Requirements**

**3.1 Functional Requirements**

**3.1.1 Account Creation**

- Users can create new bank accounts.

- Users must provide their personal information, including name, address, and contact details.

- The system generates a unique account number and associates it with the user.

**3.1.2 Deposit**

- Users can deposit money into their accounts.

- Deposits can be made in various forms, including cash and electronic transfers.

- The system updates the account balance accordingly.

**3.2 Use Cases**

**3.2.1 Use Case #1: Account Registration**

- User initiates the account creation process.

- User provides personal details.

- System generates an account number and assigns it to the user.

**3.2.2 Use Case #2: Deposit Funds**

- User selects the deposit option.

- User specifies the amount and deposit method.

- System updates the account balance.

**3.3 Classes / Objects**

**3.3.1 User**

- **Attributes**: Name, Address, Contact Details, Account Number

- **Methods**: CreateAccount(), DepositFunds(), WithdrawFunds()

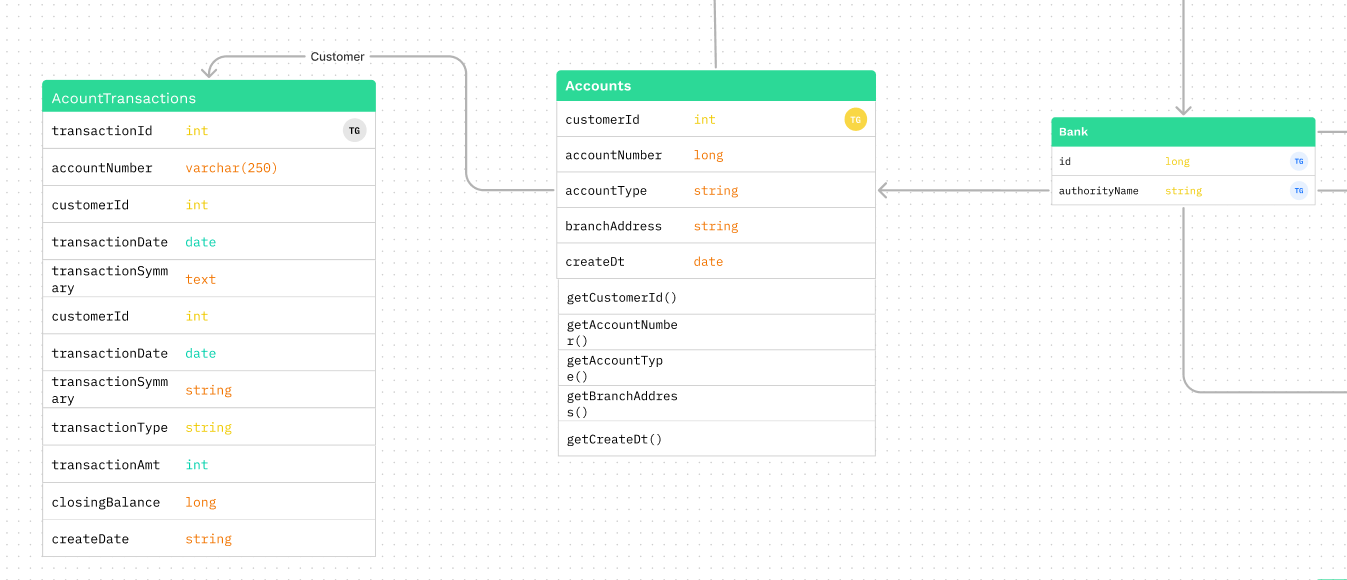
**3.3.2 Account**

- **Attributes**: Account Number, Balance

- **Methods**: GetBalance(), UpdateBalance()

**4. Analysis Models**

**4.1 Unified Modeling Language (UML)**



**5. Use Case**

**Use Case:** Account Transaction

**Primary Actor:** User

**Precondition:** The User is authenticated and has access to their account.

**Main Success Scenario:**

1. The User selects the "Account Transaction" option in their account settings.
2. The system presents options for different types of account transactions (e.g., transfer funds, make a payment, view transaction history).
3. The User selects the desired account transaction type.
4. The system guides the User through the specific steps for the chosen transaction, which may include entering transaction details, recipient information, and amount.
5. The User enters the required information.
6. If the validation is successful, the system processes the account transaction.
7. The system confirms the successful completion of the transaction to the User.

**Exception Scenarios:**

1. Invalid Transaction Details: If the User provides incomplete or incorrect transaction details.
2. Insufficient Funds: If the User's account balance is insufficient for the selected transaction.
3. Technical Failure (e.g., Network Error): If there is a technical issue that prevents the transaction from being completed.
4. Transaction Limit Exceeded: If the User exceeds their transaction limit.
5. Transaction Cancelled by User: If the User decides to cancel the transaction during the process.