Software Requirements Specification

For

Bank Management Application

Version 1.0

Prepared by Team CyberTech OOD

Syracuse University CSE687

Table of Contents

Table of Figures

1. Introduction:

1.1. Purpose:

The purpose of this Software Requirements Specification (SRS) is to specify the software requirements for the Bank Management Help Application for the end users. The SRS aims to serve as a roadmap for the developers to ensure that the required functionality is implemented, and for the test team to develop Verification and Validation (V&V) plans and procedures that will demonstrate to the customer that the system was built according to this specification.

1.2. Scope:

This document specifies the requirements for the following capabilities:

1. View Account Balance
2. Check Savings Account
3. Check Checking Account
4. Transaction
5. Transfers
6. Payment
7. Authorization
8. Cash Payment
9. Credit Payment

1.3. Definitions:

1. BDD: Block Definition Diagram.
2. IA: Information Assurance.
3. IP: Internet Protocol.
4. SRS: Software Requirements Specification.
5. STIGS: Security Technical Implementation Guides.
6. TCP: Transmit Control Protocol.
7. UDP: User Datagram Protocol.
8. UML: Unified Modeling Language.
9. V&V: Verification and Validation.

1.4. References:

1.5. Overview:

The format of this document adheres to the guidelines set out in IEEE Std 830-1998, which provides recommendations for software specifications. Specifically, for Section 3, template A.5 is used to structure the information according to the relevant features.

2. Overall Description:

2.1. Product Perspective:

Banking applications are crucial tools that facilitate essential banking operations, including but not limited to loan requests, money transfers, bill payments, and more. As a result, these applications are among the most significant and commonly used applications available. Figure 1 System Block Diagram shows the system overview, using a Unified Modeling Language (UML) Block Definition Diagram (BDD)

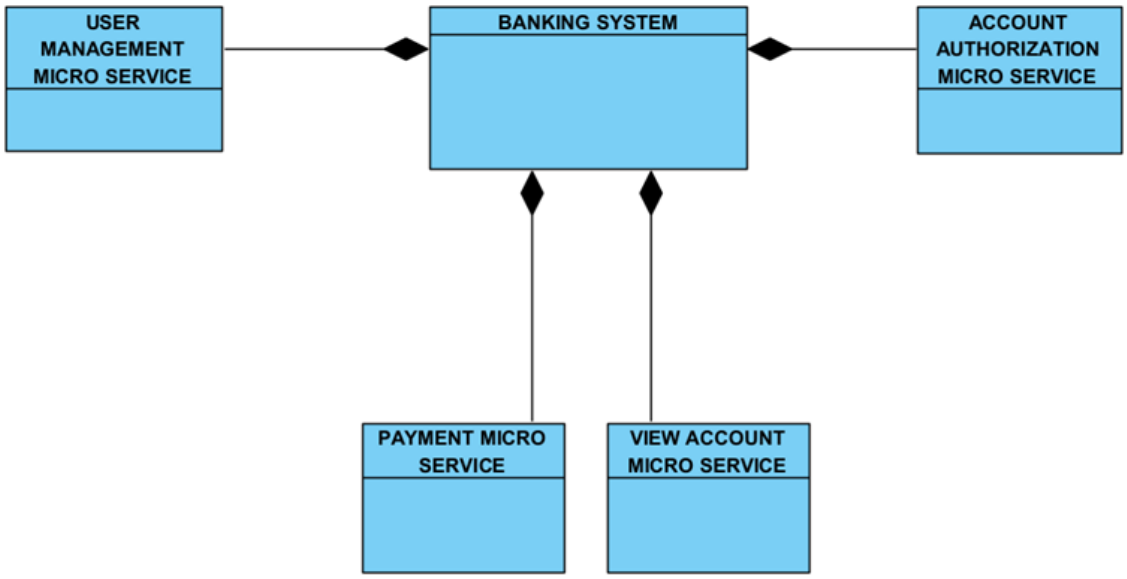

Figure 1: System Block Diagram

Account Management Microservice - This microservice would handle all the functionality related to managing bank accounts, such as depositing and withdrawing funds, transferring money between accounts, and checking account balances.

User Management Microservice - This microservice would handle all the functionality related to managing users, such as creating new users, updating user information, and deleting user accounts.

View Account Microservice - This microservice would be responsible for handling requests related to viewing information about a bank account. This microservice could include functionality such as retrieving account information, generating account statements, and displaying account information.

Payment microservice - This microservice is a small, independent service that handles payment-related operations within a larger system or application. It is designed to handle all aspects of payment processing, transaction management, process payment, and updating records.

2.2. Product Functions:

The following use case diagram depicts the users of the system and the intended way in which they will interact with the system.

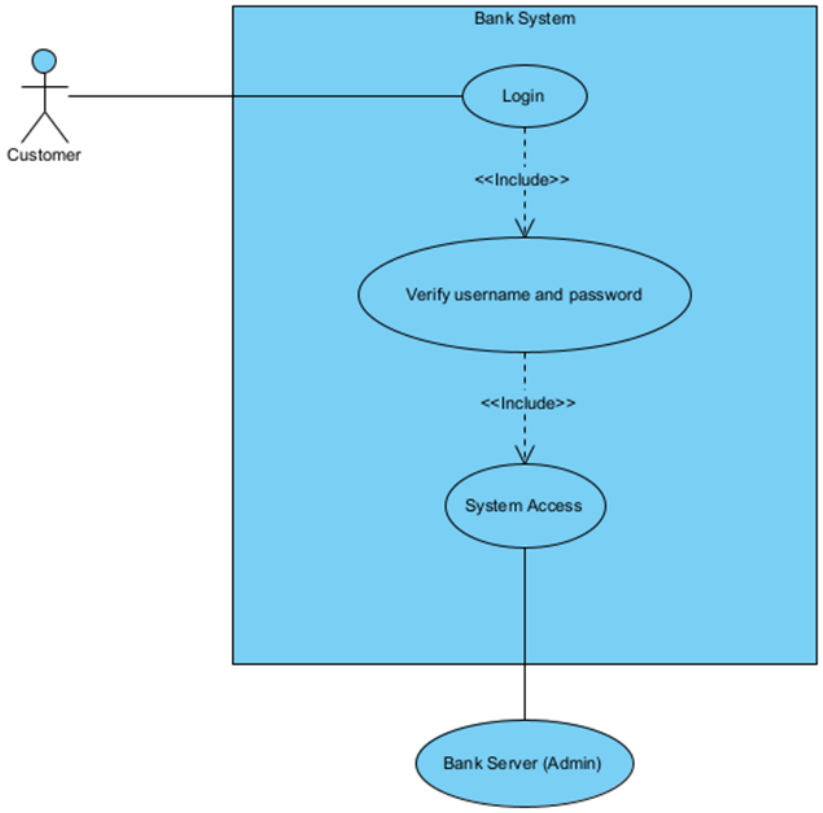

Figure 2: Bank Management Use Case Diagram-Login Page

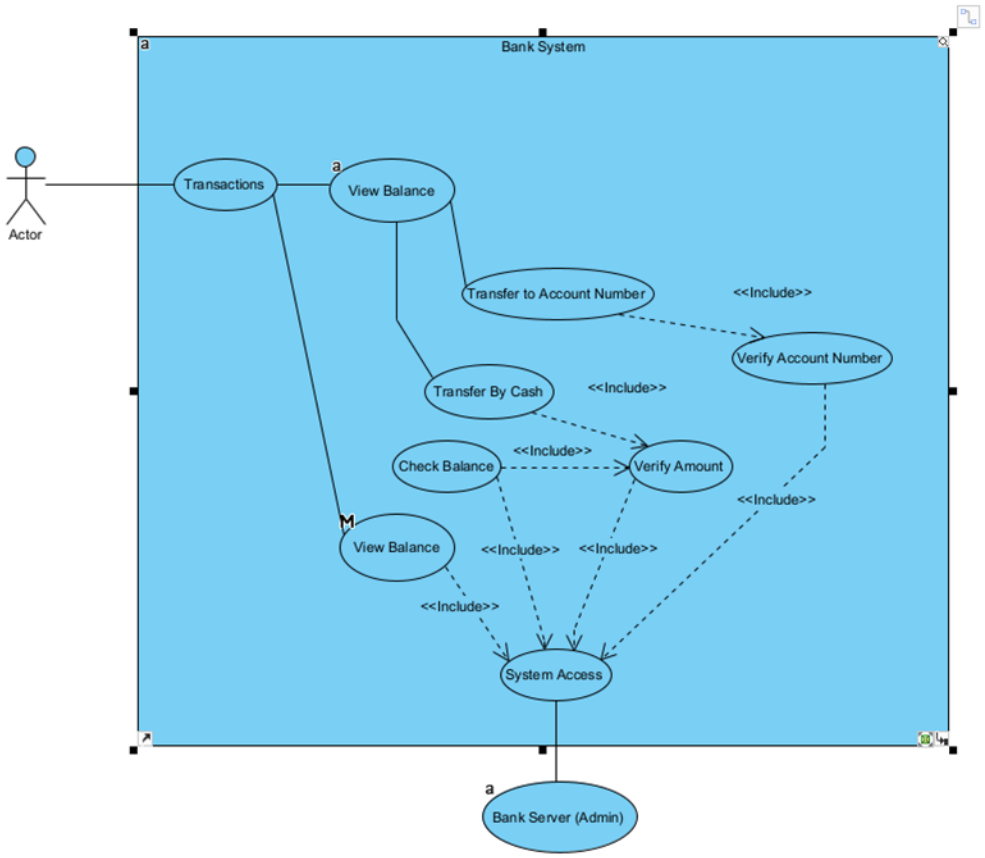

Figure 3: Bank Management Use Case Diagram-Transaction Page

2.3 Use Case Descriptions:

2.3.1: Create an Account

General Characteristics	
Intent	To allow a user to create a new account.
Scope	Account creation feature.
Primary Actor	User.
Secondary Actor	None.
Preconditions	The user does not have an existing account.
Trigger	The user selects the "Create Account" option from the login page.
Success Post Condition	The system creates a new account and logs the user in.
Failed Post Condition	The system displays an error message and does not create an account.

Main Flow

Steps	Action
1	The user selects the "Create Account" option from the login page.
2	The system displays a form for the user to enter their personal information (name, address, etc.).
3	The user enters their personal information and submits the form.
4	The system creates a new account for the user.
5	The system logs the user in and displays their account information.

Alternate Flow

Step	Action
2a	The user attempts to create an account with an email address that is already in use. 1. The system displays an error message to the user. 2. The user selects the "Try Again" option. 3. The system returns to step 2.

2.3.2: View Balance

General Characteristics	
Intent	To allow the user to view their account balance.
Scope	Account balance feature.
Primary Actor	User.
Secondary Actor	None.
Preconditions	The user must be logged in to their account.
Assumptions	None
Trigger	The user selects the "View Balance" option from the menu.
Success Post Condition	The system displays the user's account balance.
Failed Post Condition	None

Main Flow

Steps	Action
1	The user selects the "View Balance" option from the menu
2	The system retrieves the user's account balance.
3	The system displays the user's account balance.

Alternate Flow

Step	Action
2a	The system is unable to retrieve the user's account balance. The system displays an error message to the user.
2b	The user selects the "Try Again" option.
3b	The system retries to retrieve the user's account balance.
3c	The system displays the user's account balance.

2.3.3: Payment

General Characteristics	
Intent	To allow the user to make a payment from their account.
Scope	Payment feature.
Primary Actor	User.
Secondary Actor	None.
Preconditions	The user must be logged in to their account and have sufficient funds to make a payment.
Assumptions	None
Trigger	The user selects the "Make Payment" option from the menu.
Success Post Condition	The system processes the payment and updates the user's account balance.
Failed Post Condition	The system displays an error message and does not process the payment.

Main Flow

Steps	Action
1	The user selects the "Make Payment" option from the menu.
2	The system displays a form for the user to enter the payment details (recipient, amount, etc.).
3	The user enters the payment details and submits the form.
4	The system processes the payment and updates the user's account balance.
5	The system displays a confirmation message to the user.

Alternate Flow

Step	Action
2a	The user attempts to make a payment for an amount greater than their account balance. The system displays an error message to the user. The user selects the "Try Again" option.

	The system returns to step 2.

2.3.4: Transfer

General Characteristics	
Intent	To allow the user to transfer funds from their account to another account.
Scope	Transfer feature.
Primary Actor	User.
Secondary Actor	None.
Preconditions	The user must be logged in to their account and have sufficient funds to make the transfer.
Assumptions	None
Trigger	The user selects the "Transfer Funds" option from the menu.
Success Post Condition	The system processes the transfer and updates the user's account balance and the recipient's account balance.
Failed Post Condition	The system displays an error message and does not process the transfer.

Main Flow

Steps	Action
1	The user selects the "Transfer Funds" option from the menu.
2	The system displays a form for the user to enter the transfer details (recipient, amount, etc.).
3	The user enters the transfer details and submits the form.
4	The system processes the transfer and updates the user's account balance and the recipient's account balance.
5	The system displays a confirmation message to the user.

Alternate Flow

Step	Action
2a	The user attempts to make a transfer for an amount greater than their account balance. The system displays an error message to th e user. The user selects the "Try Again" option. The system returns to step 2.
2b	The user enters an invalid account number for the recipient. The system displays an error message to the user. The user selects the "Try Again" option. The system returns to step 2.

2.3.5: Authorization

General Characteristics	
Intent	To verify the user's identity and authorize access to their account.
Scope	Authorization feature.
Primary Actor	User.
Secondary Actor	None.
Preconditions	The user must have a valid username and password.
Assumptions	None
Trigger	The user attempts to log in to their account.
Success Post Condition	The system verifies the user's identity and grants access to their account.
Failed Post Condition	The system displays an error message and does not grant access to the user's account.

Main Flow

Steps	Action
1	The user enters their username and password on the login screen
2	The system verifies the user's identity.

3	The system grants access to the user's account.
4	The system displays the user's account information.

Alternate Flow

Step	Action
2a	The user enters an incorrect username or password. The system displays an error message to the user. The user selects the "Try Again" option. The system returns to step 1.
2b	The user attempts to log in to a disabled account. The system displays an error message to the user.

2.3.6: Apply for a Cheque Book

General Characteristics	
Intent	To allow the user to apply for a checkbook.
Scope	Chequebook application feature.
Primary Actor	None.
Secondary Actor	None.
Preconditions	The user must have an existing account with the bank.
Assumptions	None
Trigger	The user selects the "Apply for a Cheque Book" option from the menu.
Success Post Condition	The system approves the user's request for a checkbook and mails the checkbook to the user's registered address.
Failed Post Condition	The system displays an error message and does not approve the user's request for a checkbook.

Main Flow

Steps	Action
1	The user selects the "Apply for a Cheque Book" option from the menu.
2	The system presents the checkbook application form to the user.

3	The user fills in the checkbook application form with their personal information, account number, and other relevant details.
4	The user submits the checkbook application form.
5	The system reviews the checkbook application and checks the user's account balance and transaction history.
6	The system approves or denies the user's request for a checkbook and informs the user of the decision.
7	If approved, the system mails the checkbook to the user's registered address.

Alternate Flow 1: Insufficient Account Balance

Step	Action
5a	The system determines that the user has an insufficient account balance to order a checkbook.
5b	The system informs the user that they do not meet the minimum account balance requirements to order a checkbook.

Alternate Flow 2: Invalid Address

Step	Action
6a	The system determines that the user's registered address is invalid or incomplete.
6b	The system informs the user that they need to update their registered address before the checkbook can be mailed.

2.4. User Characteristics

User characteristics for a banking system can include:

- Accessibility requirements: Accessibility requirements are important for users with disabilities, such as visual or hearing impairments. Banks may need to consider how they can make their products and services accessible to these users.
- Location: A user's location can impact their access to physical bank branches and ATMs, and may also influence their banking needs and preferences.

2.5 Constraints

These are defined specifically in Section 3.

3. Specific Requirements

3.1 Overview

Bank management applications are software systems designed to help financial institutions, such as banks, credit unions, and other financial institutions, to manage their day-to-day operations effectively. These applications can help automate various banking processes, such as account management, loan processing, transaction processing, customer relationship management, and risk management.
Bank management applications often consist of multiple modules that can be customized according to the specific needs of the financial institution.
Overall, bank management applications are designed to help financial institutions streamline their operations, improve efficiency, and enhance customer satisfaction by providing an integrated, user-friendly system that simplifies day-to-day banking activities.

3.2 Interface Requirements

3.2.1.1. User Interfaces:

- A banking user interface shall allow customers to interact with the interface via a display.
- The screen display shall be a minimum of a 10” diagonal screen.

- The banking system user interface shall include a keypad with the following buttons:

3.2.1.2. WELCOME TO THE MAIN MENU:

- View Customer Accounts
- Customer Account Registration
- Edit Customer Account
- Delete Customer Account
- Search Customer Account
- Transaction
- Log Out
- About US

3.2.1.3. Account Creation Menu:

- Enter Your Name
- Enter Your Account Number
- Enter Your Phone Number
- Enter Your Address
- Enter Your Email

3.2.1.4. Account Searching

- Enter Your Account Number To Search
- Account Number Matched
- Detail Information of "ACCOUNT NUMBER" Account Number
- First Name
- Last Name
- UserID

3.2.1.5. AUTHORIZATION

- Username
- Password
- Admin Username
- Admin Password

3.2.1.6. TRANSACTION MENU

- Balance Inquiry
- Cash Deposit
- Cash Withdrawal
- Fund Transfer
- Main Menu

3.2.2. Hardware Interfaces:

- **The screen of the device on which the banking system is compiled and is running.**
- The banking system shall require a 32-bit processor.
- The banking system shall require a processor requirement: Intel Core i3-3240 (2 * 3400); AMD FX-4300 (4 * 3800).
- The banking system shall have a memory requirement of 4 GB RAM.
- The banking system shall have a Storage requirement < 100 MB available space.

3.2.3. Software Interfaces:

- The banking system shall require an operating system OS: Windows XP or later.
- The software shall support TCP and UDP IP over IPV6 protocols.

3.3. System Features

- The system software supports the Use Cases described in Figure 2 Banking management system Use Cases.

3.3.1.1 Perform Diagnostics

- Introduction/Purpose of Feature: This feature will provide the maintenance operator with the ability to perform diagnostics and troubleshooting on the bank management help application as well as the console which provides the clerk with monitoring capabilities.

- Stimulus/Response Sequence

- Associated Functional Requirements

- The bank management help application shall contain embedded diagnostic and maintenance tools to monitor the application's operation and troubleshoot issues.

- The software shall provide alerts or notifications when maintenance is required.
- The software shall include self-diagnostic and self-repair features.
- The software shall support on-the-air software updates.

3.3.1.2 New Account

- Introduction/Purpose of Feature: This feature will allow the Customer to bank the application, following the Bank Management Hrlp Applications Use Case above.

3.3.1.2.2 Stimulus/Response Sequence

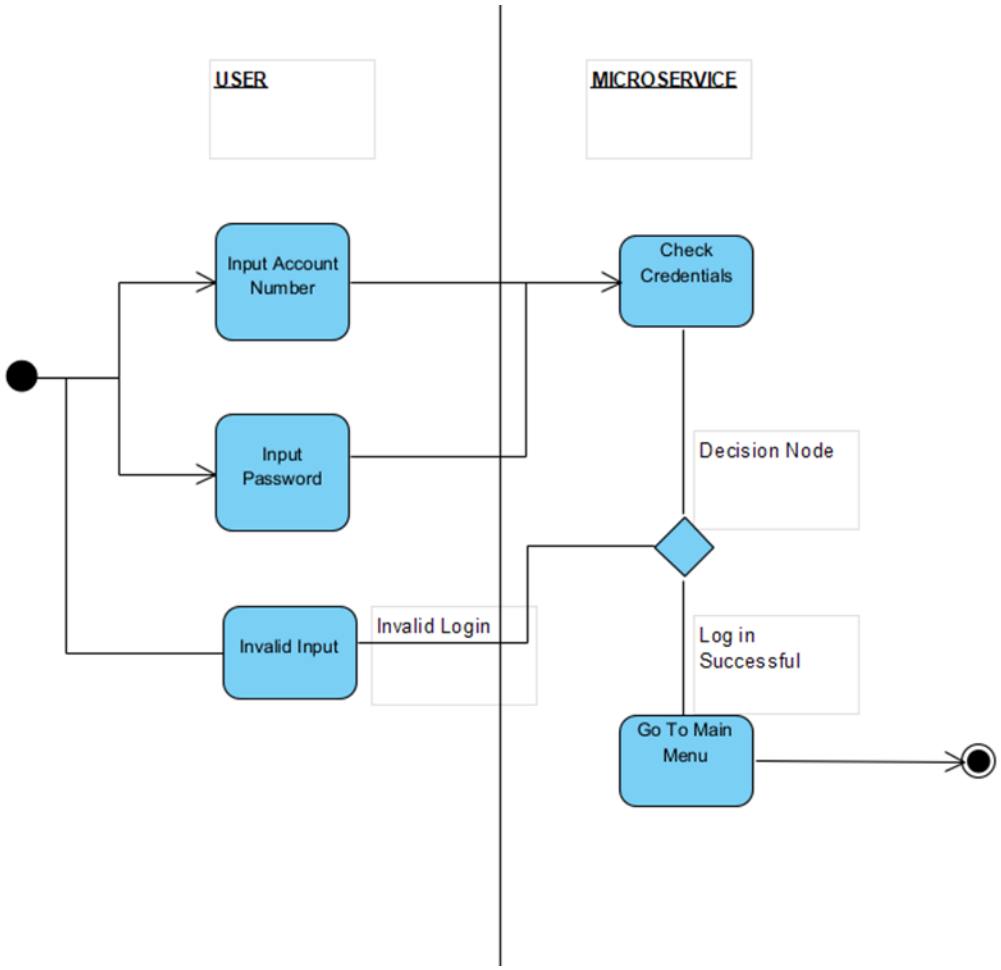

Figure 4: Bank Management Application Diagram-Login Page

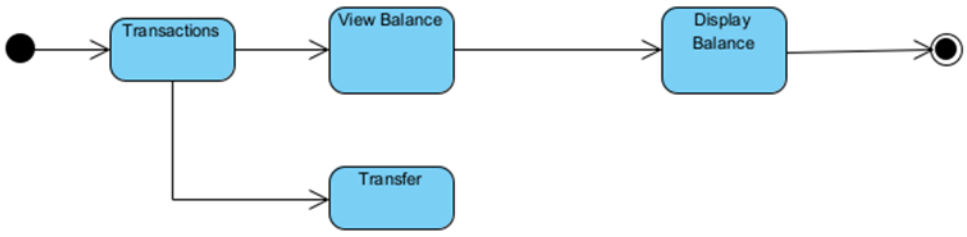

Figure 5: Bank Management Application Diagram-Transactions-View Balance Page

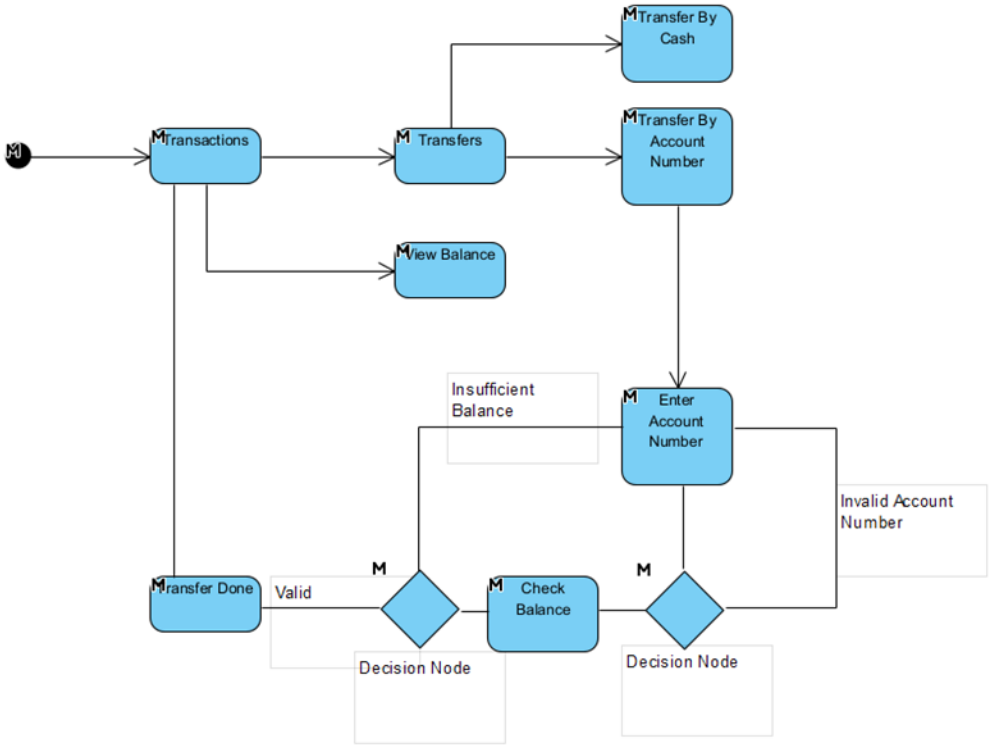

Figure 6: Bank Management Application Diagram-Transactions Page-Transfer by Account No.

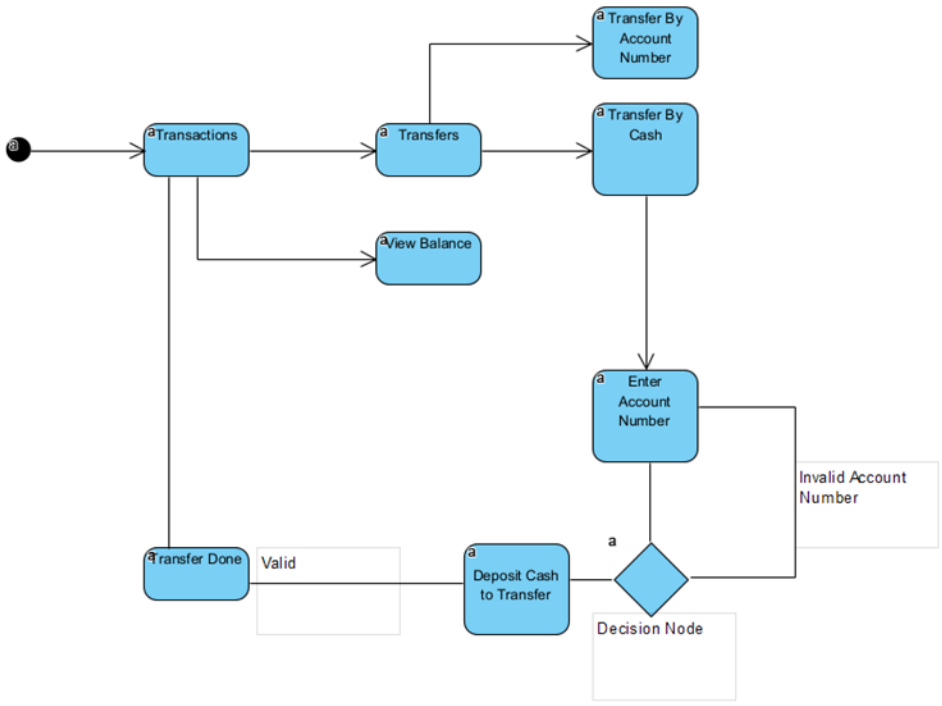

Figure 7: Bank Management Application Diagram-Transactions Page-Transfer by Cash.

3.4 Performance Requirements

- The software shall keep track of all the user data present along with

3.5 Design Constraints

- Software processing shall allow a spare capacity of 10% for memory, CPU utilization, and long-term storage (e.g. disk storage).

- The system shall be able to support multiple languages to cater to all users, including the following:

    1. English

- The software shall be developed and maintained in accordance with industry-standard ISO/IEC 5055:2021.

3.6 Software System Attributes

- The Software shall use data encryption across all interfaces.

- The software shall conform to commercial standard Information Assurance (IA) controls and Security Technical Implementation Guides (STIGS).

- The software shall allow only authorized users to make configuration changes.

- The software shall allow only authorized users to perform diagnostic operations.

3.7 Other Requirements

GENERAL CHARACTERISTICS	
Intent	Allow the Customer to make use of the banking system, securely and
Scope	Bank Management System
Primary Actor	Customer
Secondary Actors	Admin
Preconditions	The customer has an email. They are required during account creation.
Assumptions	The system is secure and the payments done are only monitored by the admin. All transaction requests are auto-approved by the admin.
Trigger	Opening the app which runs primarily on C++.
Success Post Condition	The transactions or queries made are updated to the database.

Failed Post Condition	The customer is unable to log in or transactions fail.

Sunny Day Scenario:

Step	Action
Start	This Scenario begins when the user launches the app to create a new bank user.
1	The Customer is instructed by the User Interface to add in their personal information.
2	The Customer is then guided to the user interface according to the choice made by them.
3	Once the account has been created the customer will then be led to the Main list of choices they want to make.
4	This Use Case ends when the user is done making their account in the user database.

Step	Action

Start	This Scenario begins when the user launches the app to login into their bank account to check their balance.
1	The Customer is instructed by the User Interface to add in their login details.
2	The Customer is then guided to the user interface and given multiple choices after a successful login.
3	The Customer will then choose to view their account balance or details.
4	This Use Case ends when the user is done with their queries from the banking system.

Step	Action
Start	This Scenario begins when the user launches the app to make a transaction.
1	The Customer is instructed by the User Interface to
2	The Customer is then guided to the user interface to login to their account.
3	The Customer is then given a choice to go through the transactions made or if they want to make another transaction.

4	The Customer will then be prompted to either log out or move on ahead with further queries/transactions. When they are done they can log out and close the app.
5	This Use Case ends when the user is done with their transactions from the banking system.

Rainy Day Scenario:

Step	Action
Start	This scenario begins with a customer accessing their account on multiple devices
1	The customers would make two different transactions on different devices.
2	The transaction will then go through depending on their timings and can throw off the actual amount in their account.
3	This Use Case ends when the user is done with their transactions made by the banking system.