



# **Banking System Software Requirement Specification**

Project Code: 08

Document Code:

**HCM City , Saturday April 17 2021**



## RECORD OF CHANGE

\*A - Added M - Modified D - Deleted

Effective Date	Changed Items	A* M, D	Change Description	New Version
15/04/2021	Initial	A	Add project over view	
16/04/2021	Add Use case diagram	A	Add actorr and use case diagram to the srs	
20/04/2021	Add functional requirement	A	Add the Login, Transfer and deposit functional requirements	
27/04/2021	Add non-functional requirement	A		

## **SIGNATURE PAGE**

**ORIGINATOR:** Nguyễn Hoàng Hải 15/04/2021  
Leader

**REVIEWERS:** Phan Tiến Lào\_\_27/04/2021  
Supporter

**APPROVAL:** Phan Tiến Lào\_\_27/04/2021  
Supporter

## TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION.....</b>	<b>6</b>
1.1	Purpose.....	6
1.2	Scope.....	6
1.3	Definitions, Acronyms, and Abbreviations.....	<b>Error! Bookmark not defined.</b>
1.4	References.....	<b>Error! Bookmark not defined.</b>
1.5	Overview.....	<b>Error! Bookmark not defined.</b>
<b>2</b>	<b>OVERALL DESCRIPTION.....</b>	<b>7</b>
<b>3</b>	<b>FUNCTIONAL REQUIREMENTS.....</b>	<b>8</b>
3.1	<The actors>.....	<b>Error! Bookmark not defined.</b>
3.2	<The use cases>.....	<b>Error! Bookmark not defined.</b>
3.3	<Use Cases Diagram>.....	<b>Error! Bookmark not defined.</b>
3.3.1	<Functional Requirement one>.....	11
<b>4</b>	<b>NON-FUNCTIONAL REQUIREMENTS.....</b>	<b>12</b>
4.1	Usability.....	<b>Error! Bookmark not defined.</b>
4.2	Reliability.....	14
4.3	Performance.....	14
4.4	Supportability.....	14
4.5	Design Constraints.....	15
4.6	On-line User Documentation and Help System Requirements.....	<b>Error! Bookmark not defined.</b>
4.7	Purchased Components.....	<b>Error! Bookmark not defined.</b>
4.8	Interfaces.....	19
4.9	Licensing Requirements.....	<b>Error! Bookmark not defined.</b>
4.10	Legal, Copyright, and Other Notices.....	<b>Error! Bookmark not defined.</b>
4.11	Applicable Standards.....	<b>Error! Bookmark not defined.</b>
<b>5</b>	<b>SUPPORTING INFORMATION.....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>

## **1 INTRODUCTION**

This document, Software Requirements Specification (SRS), is created to document the software requirements for the Banking System. A bank has many customer accounts and that is really hard in manage these accounts so every bank will have a Banking management software. With this application, bank's staff can create, modify the balance, update, transfer money with the request of customer. A transaction is initiated when customer transfer money, deposit and withdraw their balance. Each customer has a id number when they register in bank and bank's staff will according this to find out their account and perform transaction. To login to system, each staff will have a account. When find out the customer's account, bank's staff can help customer to deposit, transfer or withdraw. When a transaction is created, it will be stored in the database of customer. By input customer's number, bank's staff can see the transaction history. To update an account, bank's staff input customer's account and find, then they can edit customer's update request and the new information will be stored in the database

### **1.1 Purpose**

There are 10 activity in this application, there are Update account, Account details, Transaction history, Search Account, Customer List, Create Account, Deposit, Withdraw, Transfer, Check Balance and some little activity of banking like calculator, notepad, currency converter and admin setting.

### **1.2 Scope**

Bank's manager and banking system

## 2 OVERALL DESCRIPTION

- Product perspective:
  - Standard PC
  - 2gb or more RAM
  - At least 500MB Hard disk space
- Product functions:
  - + )Create Account
  - + )Update account
  - + )Account details
  - + )Search Account
  - + )Customer List
  - + )Transaction history
  - + )Deposit
  - + )Check Balance
  - + )Transfer
  - + )Withdraw
- User characteristics
  - User can use computer
  - Bank's manager
  - User can read and understand the system
- Constraints:
  - This system will support for banking system in account management
  - This system use SQL sever to stored data
  - This system will run under Windows
  - All codes was written in C#

### 3 FUNCTIONAL REQUIREMENTS

#### 3.1 The Actor

*[An Actor is someone outside the system that communicates directly with the system]*

*[List the actors of the system with their permissions, privileges...]*

#	Main functions	Customer	Bank's manager
1	Create Account		x
2	Update Account		x
3	Account details		x
4	Search Account		x
5	Customer List		x
6	Transaction history		x
7	Deposit		x
8	Check Balance		x
9	Transfer		x
10	Withdraw		x
11	Login		x

#### 3.2 Use Case

*[A Use Case is a description of a complete flow of events that results in something of value to an Actor.]*

System Functions	Main Use Cases	Use Case #
Functional Requirement One		
	Create Account	UC_CreateAccount
	Update Account	UC_UpdateAccount
	Account details	UC_AccountDetails



	Search Account	UC_SearchAccount
	Customer List	UC_CustomerList
	Transaction history	UC_Transaction history
	Deposit	UC_Deposit
	Check Balance	UC_CheckBalance
	Transfer	UC_Transfer
	Withdraw	UC_Withdraw
	Login	UC_Login

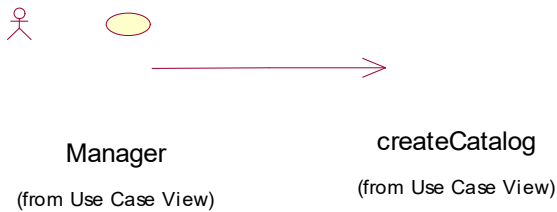
### **3.3 Use case diagram**

*[The main Use Case Diagrams of the system]*



### 3.3.1 <Functional Requirement one>

#### 3.3.1.1 Login to the system



User Case ID	UC_Login	
Name	Login to the system	
Goal	Login and use the system devices	
Actors	Customer/Manager	
Pre-conditions	If there is customer just login with the account registered in the bank. If there is manager, he/she have to login with the admin account	
Post-conditions		
Main Flow	1. Insert UserName and Password The manager have to check to the admin box  2. Press the login button	
Exception	1A: If the password or username was incorrect, user have to insert again. If the user check the admin box, reveal the message box "You are not admin, please try again"  1.B: If the manager insert wrong username or password, reveal message box "Wrong username or password, please try again"	
Open Issues	N/A	

#### 3.3.1.2 Deposit

User Case ID	UC_Deposit
Name	Login to the system
Goal	Login and use the system devices

Actors	Customer/Manager	
Pre-conditions	If there is customer just login with the account registered in the bank. If there is manager, he/she have to login with the admin account	
Post-conditions		
Main Flow	1. Insert UserName and Password The manager have to check to the admin box  2. Press the login button	
Exception	1A: If the password or username was incorrect, user have to insert again. If the user check the admin box, reveal the message box "You are not admin, please try again"  1.B: If the manager insert wrong username or password, reveal message box "Wrong username or password, please try again"	
Open Issues	N/A	

### 3.3.1.3 Tranfer Money

User Case ID	UC_Transfer
Name	Tranfer Money
Goal	To tranfer money to somebody
Actors	Customer/Manager
Pre-conditions	PRE.1: Login to the system PRE.2: Sufficient funds
Post-conditions	POST.1: The customer account subtracts the amount transferred  POST.2: The transaction is saved in the database of the customer

Main Flow	<ol style="list-style-type: none"><li>1. Insert the account number of the receiver</li><li>2. Insert the amount of money</li><li>3. The system will check if the customer have sufficient funds</li><li>4. Insert the Title of the transfer</li><li>5. Press the Accept button to perform</li></ol>	
Exception	<p>1A: If the account number of the receiver is invalid, the customer have to re-enter the account number.</p> <p>1.B: If the customer's balance is smaller than the amount of money want to transfer, reveal message box to notify" You don't have enough money, please try again"</p>	
Open Issues	N/A	

## **4 NON-FUNCTIONAL REQUIREMENTS**

### **4.1 Reliability**

Measure if product is reliable enough to sustain in any condition. Should give consistently correct results. Product reliability is measured in terms of working of project under different working environment and different conditions.

### **4.2 Performance**

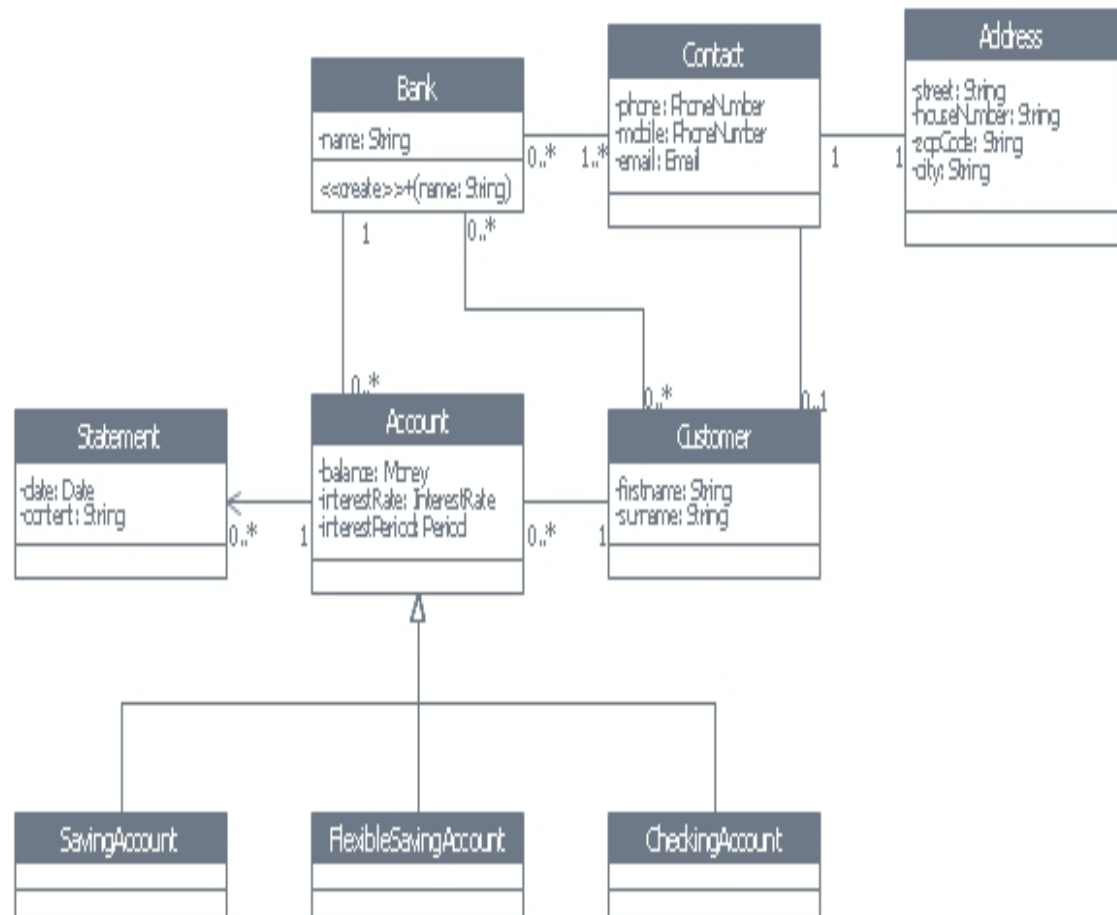
To Major system quality attribute. Measured in terms of time required to complete any task given to the system. For example system should utilize processor capacity, disk space and memory efficiently. If system is using all the available resources then user will get degraded performance failing the system for efficiency. If system is not efficient then it can not be used in real time applications.

### **4.3 Supportability**

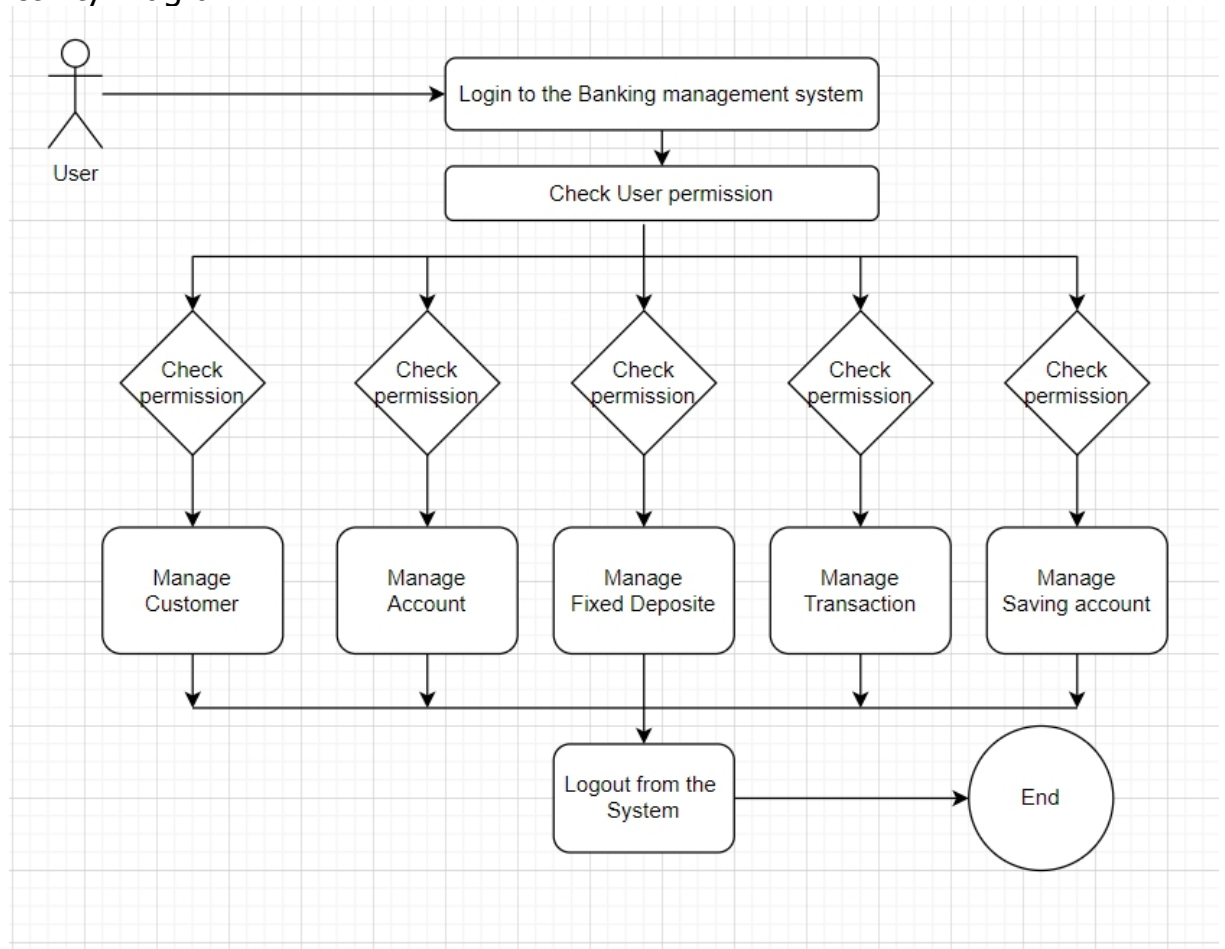
Different versions of the product should be easy to maintain. For development, its should be easy to add code to existing system, should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy. System be easy to maintain and correcting defects or making a change in the software.

## 4.4 Design Constraints

- Class Diagram:

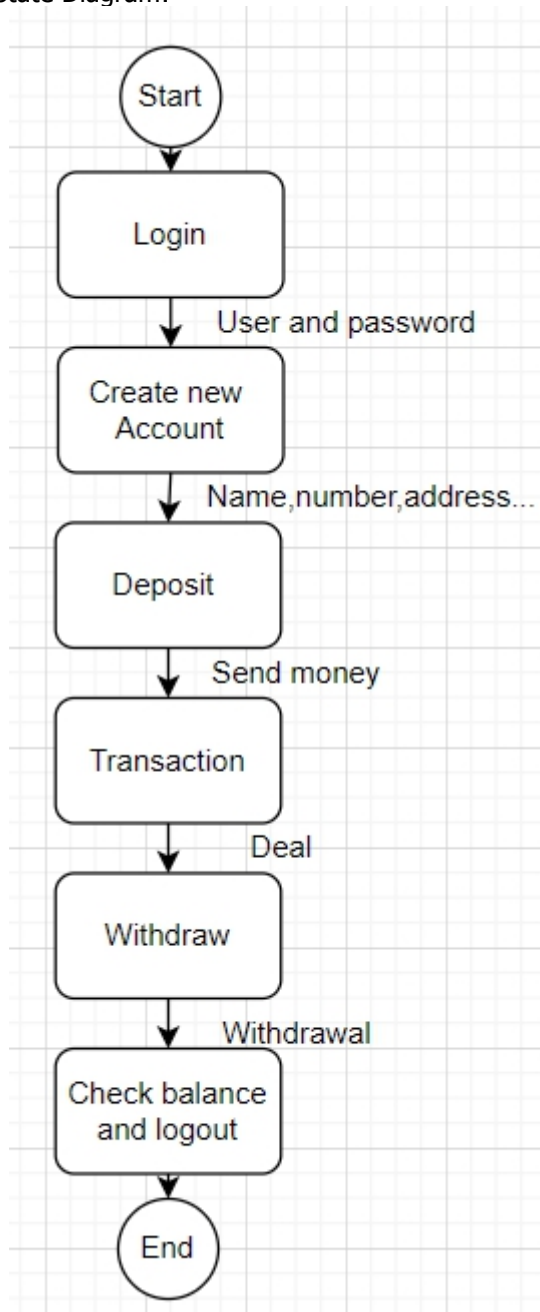


- Activity Diagram:





- State Diagram:

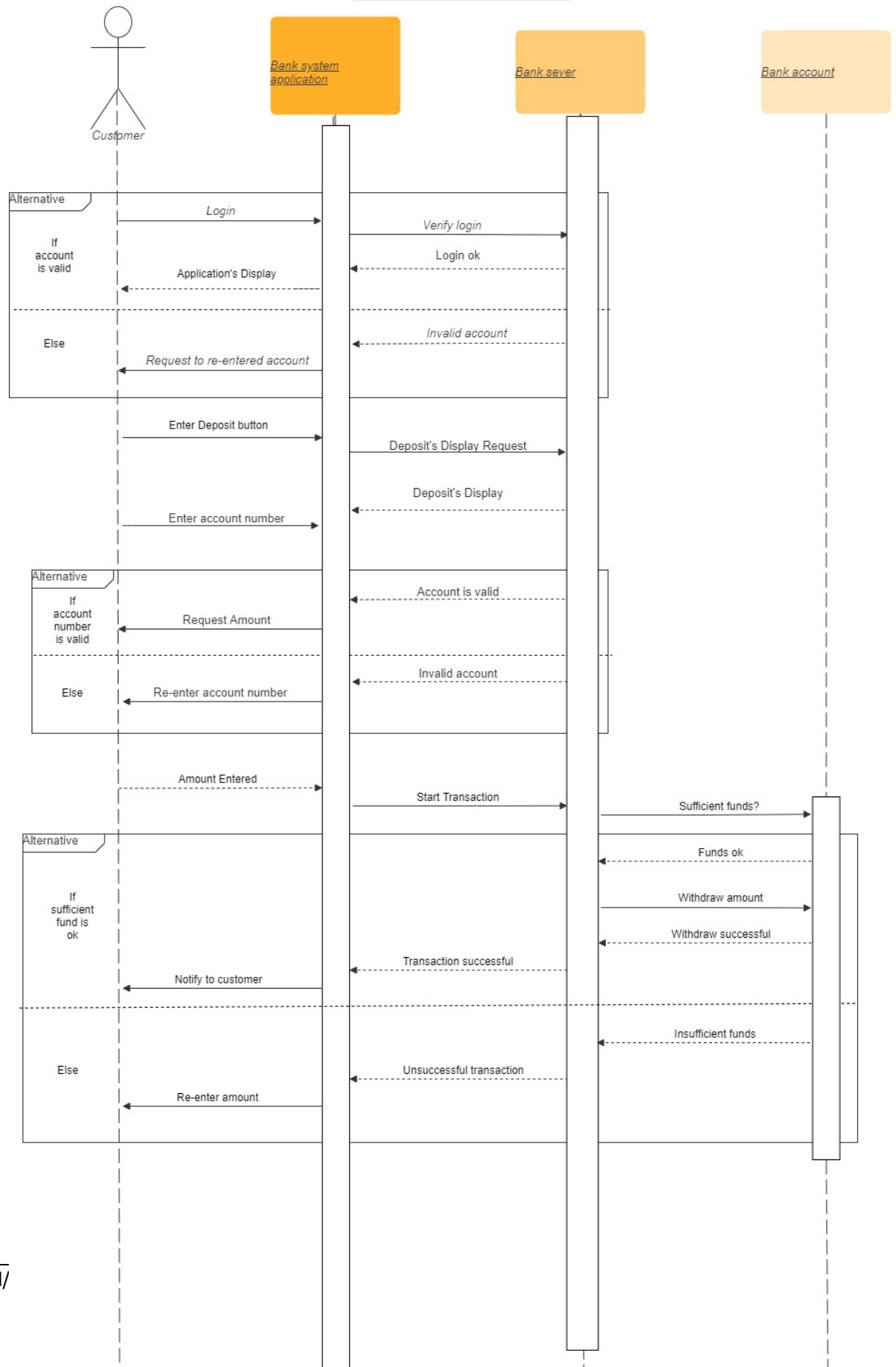


- Sequence Diagram:

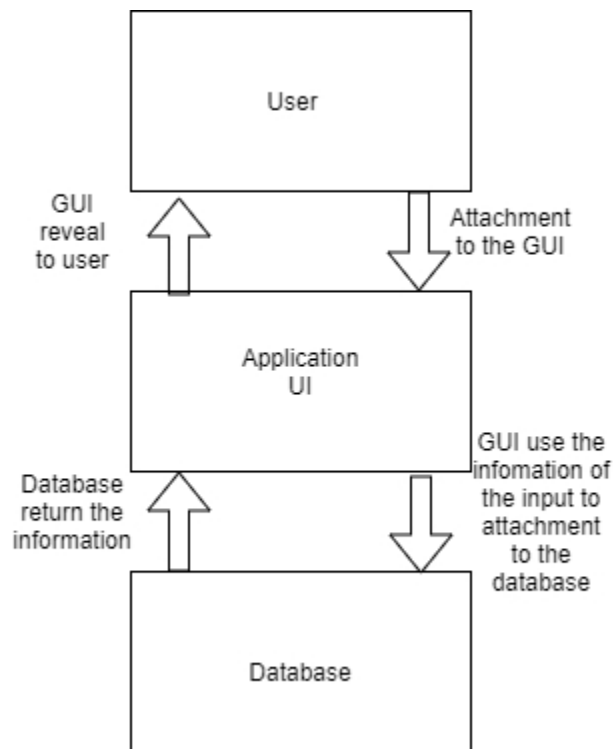
Sequence Diagram of Deposit function:

### Deposit Sequence diagram

%creatorName% | April 26, 2021



## 4.5 Interfaces



### Communications Interfaces

**User interface:** User attachment to the application UI, like input number, click in a button,... The system will receive the information and start run to connect to the database

**Database Interface:** Database will receive the information from system and find out the data. Then database return it to system and system use the data received to reveal to user by UI