Software Requirements Specification

**for**

**<BankPro: Integrated Banking Management System>**

Version 1.0 approved

Prepared by: Muntazer Mehdi

**The Bongcloud**

<23-07-2021>

**Table of Contents**

[1. Introduction 1](#_Toc170419251)

[1.1 Purpose 1](#_Toc170419252)

[1.2 Document Conventions 1](#_Toc170419253)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc170419254)

[1.4 Product Scope 1](#_Toc170419255)

[2. Overall Description 2](#_Toc170419256)

[2.1 Product Perspective 2](#_Toc170419257)

[2.2 Product Functions 2](#_Toc170419258)

[2.3 User Classes and Characteristics 2](#_Toc170419259)

[2.4 Operating Environment 3](#_Toc170419260)

[2.4.1 System requirement: 3](#_Toc170419261)

[1. Lowest specification: 3](#_Toc170419262)

[2. Recommended specification: 3](#_Toc170419263)

[2.4.2 Software requirement: 4](#_Toc170419264)

[2.5 User Documentation 4](#_Toc170419265)

[3. External Interface Requirements 4](#_Toc170419266)

[3.1 User Interfaces 4](#_Toc170419267)

[4. System Features 6](#_Toc170419268)

[4.1 System Feature 1 6](#_Toc170419269)

[4.2 System Feature 2 6](#_Toc170419270)

[4.3 System Feature 3 7](#_Toc170419271)

[4.4 System Feature 4 7](#_Toc170419272)

[4.5 System Feature 5 7](#_Toc170419273)

[4.6 System Feature 6 8](#_Toc170419274)

[4.7 System Feature 7 8](#_Toc170419275)

[5. Other Nonfunctional Requirements 8](#_Toc170419276)

[5.1 Performance Requirements 8](#_Toc170419277)

[5.2 Safety Requirements 9](#_Toc170419278)

[5.3 Security Requirements 9](#_Toc170419279)

[5.4 Software Quality Attributes 9](#_Toc170419280)

[6. System Design 9](#_Toc170419281)

[6.1 Class diagram: 9](#_Toc170419282)

[6.2 Activity diagram: 10](#_Toc170419283)

[6.3 Domain model: 11](#_Toc170419284)

[6.4 System sequence diagram: 11](#_Toc170419285)

[7. System Architecture 12](#_Toc170419286)

[8. System Testing 12](#_Toc170419287)

[9. References 12](#_Toc170419288)

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| SRS Document 1 | 21-06-21 | The stakeholder can not like interface and they does not give us a full requirements in final deliverable there are many requirement which does not add in it. | 1 |
| SRS Document 2 | 05-07-21 | In this SRS document stakeholder does not like ican frame work and they also include a transaction section which was completey online. | 2 |

# Introduction

## Purpose

Final documentation of the Banking Management System. This document covers the whole management system. It is officially the 3rd version of the SRS document. A Bank management system is going to be developed. This management system will be more efficient, and it the functions included in it will be of great help to faster the banking process. It will cover everything from customer history to money transfer to keeping everything up to date.

## Document Conventions

There is no specific document conversion needed for this documentation. So, no special conventions are used. However, this document is being made in standard document writing style using standard font families and standard font size.

## Intended Audience and Reading Suggestions

This document is intended for project team, that includes, developers, testers, coders, project manager. Other than team, this document is for marketing staff and customer/stakeholders. Purpose of showing this to project manager and customer is to check whether the software being developed is what user/customer wants of there’s something off the track or the project is on right path. Ideal sequence for reading this document for the team is to read introduction first, then go to project description, after that read non-functional requirements and then functional requirements. More to interface section after that and then continues rest of its flow in sequential way, like design, architecture, and testing at the end. For Marketing team, it’s better to read requirements i.e. both functional and non-functional and interface rather than going into other complications.

This is a highly **CONFIDENTIAL** document and available to those who are included in project. Any breach of confidential information will result in legal.

## Product Scope

Purpose of this management system is to make banking easy. Objective is to obtain an efficient and compatible software which work fine on low specification, Goal of it is to provide all the asked functionalities as soon as possible. Major benefit that it will give to the customer is that it will save a lot of both, the bank, and the client. More work can be performed in less time in this way.

# Overall Description

## Product Perspective

This system is the replacement of the older system. Due to the workload and so chances in basic requirements, this system is in making to replace that old function. The sole purpose of designing this software is to bring a revolution in the field of banking. Making banking easier and feasible for both public and bank users. Today, there are different bank management systems providing different functionalities, this one is made by taking the best features of most of the management systems and making lack of other management systems better.

## Product Functions

Some of the major functions that this software is going to perform are:

* Cash Deposit
* Pay Drafts
* Cash Withdrawal
* Check to cash (vice versa)
* Trans-section history
* Different Loan policies
* Locker system
* Available cash in branch
* Live transactional details sharing and others.

Diagram

Description automatically generated

## User Classes and Characteristics

* Person:

this class can store general information about person like person name, person id, person email and others. This is a abstract base class.

* Customer:

This class can manage all the information of customer like customer name, customer id, customer email and others. It is a derived class of a person class.

* Employee:

this class can store general information about Employee like Employee name, Employee id, Employee email and others. This is derived class from person class.

* Admin

This class will store all the general information about the admins of the bank. This class will access confidential data of bank which cannot even be shared with the employees of the bank. Admins may include branch manager or security in charge or some other staff that can monitor data.

* Accounts:

This class can manage all operations on accounts like cash deposit, cash withdraw, add account and others.

* Transaction:

This class can manage all operations related to transaction like transaction history, live transaction, detail sharing and others.

* Locker:

This class can manage all the information of customer how can get a locker facility to individual for annual charges.

* Loan:

This class will contain all the information about people who took loan or took different loan offers from the bank.

## Operating Environment

### System requirement:

#### Lowest specification:

* CPU: intel core 2 Duo 1.8Ghz, AMD Athlon X2 64 2.4GHz.
* CPU speed: 1.8Ghz.
* RAM: 1GB.
* OS: window 7.
* Video card: NO.
* Free Disk space: 4 GB.
* Dedicated video RAM: 256 MB.

#### Recommended specification:

* CPU: intel core 2 Quad 2.4Ghz, AMD Phenom X3 2.1GHz.
* CPU speed: 2.4Ghz.
* RAM: 2GB.
* OS: window 7.
* Video card: 512 MB NVIDIA 8600+.
* Free Disk space: 6 GB.
* Dedicated video RAM: 512 MB.

### 2.4.2 Software requirement:

* DirectX Version: DirectX 9.0c Compliant Card.
* Microsoft's .NET Framework

## User Documentation

User manual was the deliverable 1 we have sent before. After that there was a feasibility study document. These two are the only documents that are to be delivered along the software.

# External Interface Requirements

## User Interfaces

* Clarity is the most important element of user interface design.
* Second this is familiar. If you are familiar with your system than you know, what is it behaves.
* Third is responsive it also means interface provides some form of feedback.
* Graphical user interface, application

  Description automatically generatedFourth thing is attractive. Attractive in a sense that it make the use of interface enjoyable.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generatedIn their interfaces we did changings on different points our logo and buttons are totally different it just a sample output how our project lokes like. We can use at least 16px font size and the font style we use is classic-roman and arial.

As our project was a data entry module than this types of error occurs when a user enters a wrong input.

# System Features

## System Feature 1

Creating New account, another important feature.

1. **Description**

This feature is the most Important feature of the program. New account is to be created in the program. Benefits will be rated as 9, cost will somewhere between 6-7 due to the data, which is to be stored, there are some risks like having a previously created accounts and so on, so it will be rated as 5.

1. **Functional Requirements**

Functional requirements of creating an account includes many things. First of all, all the general information of the person is required. The person will be given a form to fill, in which he is required to give information like, name, address, phone no, and other stuff like that. Then this information is to be transferred on computer. Then will show their incomes. Signatures will be required at the bottom of the form, which will then be scanned to the computer. Thumb impressions will be required. Cnic of the account owner, both original and copies are required. Other official information like properties may or may not be included.

REQ-1: General information

REQ-2: Thumb Impression

REQ-3: Signatures

REQ-4: Cnic (both copies and original)

REQ-5: Other official documents (original scanned)

## System Feature 2

*Cash Deposit is our system feature number 1.*

1. **Description**

This feature will allow employees to register bank deposit in 2-3 clicks. This feature will be of high priority. Benefit will be rated as 9, cost will be hardly 4-5 and risks will be low so 3, I guess.

1. **Functional Requirements**

First the system should take account number of the account to which money is to be deposited. For that purpose, system should be connected to network to verify the account number and Account name is also required. If someone is there to send money to someone else’s account that for that purpose system should ask for his identity (i.e., cnic copy.) Cnic number and account number should be in numeric form. If account number did not match the account name an error should be generated, and user should be notified. A scanned copy of cnic wi

ll be uploaded.

REQ-1: Scanned copy of cnic to be attached with.

REQ-2: User Account Number User Account Name

REQ-3: Cnic Number

## System Feature 3

Cash Withdrawal is our Feature Number two.

1. **Description**

This feature will allow employees to withdraw money in 2-3 clicks. It is equally important as cash deposit. This feature will be of high priority. Benefit will be rated as 9, cost will be hardly 4-5 and risks will be low so 3.

1. **Functional Requirements**

Cash withdrawal requires account from which money is to be withdrawal, its account name and thumb impression of the owner. In some special case, signatures could be used instead of impression. No one else can withdrawal cash other than account owner. Another important thing is, account should have some specific amount of money to get a withdrawal.

REQ-1: Right thumb impression.

REQ-2: Account number/Account name

REQ-3: Enough money to get a withdrawal.

REQ-4: Signatures (if needed)

## System Feature 4

Transection History i.e., is feature number 4

1. **Description**

It is very important feature of the program. Benefits will be 9 out of 9, cost will be around 8, risk is high like 7-8.

1. **Functional Requirements**

Transection history pre requires all the information of the account where transaction is made. To keep the track of the real time transections, system should be connected to network at the time of transection. One transection can change the transaction history in all main server of the bank, which will automatically change the transections details in all around the server.

REQ-1: Account trough which or to which transection is taking place.

REQ-2: Network connections.

## System Feature 5

Check to cash, conversion. Our feature number 5.

1. **Description**

It is an intermediate level of important feature. Benefit will be around 6-7, cost will be like 7-8 due to MICR, and risk will be 5.

1. **Functional Requirements**

First, we need to check whether it is a crossed check or not. If it a crossed check then we need a specific account to get it deposited in otherwise it will be of no use. Then we need to get all the same requirements we needed in money deposit. If it is not a crossed check then we still we need to verify the check through MICR and signature test.

REQ-1: Signatures

REQ-2: Account number/Account name (mentioned on the check)

REQ-3: MICR machine

## System Feature 6

Loan History and Loan details. Feature number 6.

1. **Description**

Not so very important feature of the program. Benefit is rated 4 or 5 maybe, cost is low as 3 and risk is like 4 or even less as 3.

1. **Functional Requirements**

Loan has different packages. Like loan for house, business, and others. We need to store the information of the person who is there to take the loan. He/She should have a bank account to take lone. Should have something for security purpose of the loan. A form should be filled with necessary information which is to be stored in database after that. Signature is very important in this also.

REQ-1: Signatures

REQ-2: Loan Package

REQ-3: loan taker information.

## System Feature 7

Feature number 7, Loacker in the branch.

1. **Description**

This feature is of low priority. Cost will be rated around 2, not so beneficial like it will be around 2, and risks will also not be high and will be somewhere between 1-2.

1. **Functional Requirements**

Lockers are not available in every branch of the bank, so this feature is not very important. Just the information of the locker owner is required for the assigning of the lockers.

REQ-1: Information of the locker taker.

# Other Nonfunctional Requirements

## Performance Requirements

***When Network goes offline***

If the network goes offline, system will store the data temporarily on the local system and then ill apply changes as soon as the network is again online. On branch functions are available full time even if the network goes offline. However, in each branch, one system will always remain on the network, no matter what, which will get real time data from local network.

***High Network Traffic***

Network will be built in such way that there is a very low possible chance of high traffic load. However, to cope up with such situation, all the computers will be disconnected from the server, how ever two systems from respected branch will be connected to the server, to continue the ongoing work, In the long run, the traffic of the nearer server will be diverted to another server and the traffic of this branch will be moved to nearer server and soon all the systems will be back online in no time. 

## Safety Requirements

There are two major international security standards which are predefined, first is*ISO 13849 (International Organization for Standardization.)*It will define all parts of a functional safety analysis. And the second is *IEC 61511 (International Electrotechnical Commission.)*Follow these both standards, it will be enough for the safety purpose of the system.

## Security Requirements

The first security requirement is that the bank management can give their customer card with specific pattern so if he/she can deposit or withdraw cash then they can do whatever they want by inserting the given pattern otherwise they don’t withdraw cash. They should also be able to protect customer lockers by giving specific key to their customers. At home, the customer should provide security that they can check their account by applying their username or password to check their account, if he/she insert wrong username or password thrice then the account will be blocked automatically. In this, it also verifies that the unauthorized person cannot access the bank account or lockers.

## Software Quality Attributes

In software quality attributes the system quality is developed or maintained in such a way that it can be user friendly to all the customers so that they do not need any help to understand the system.

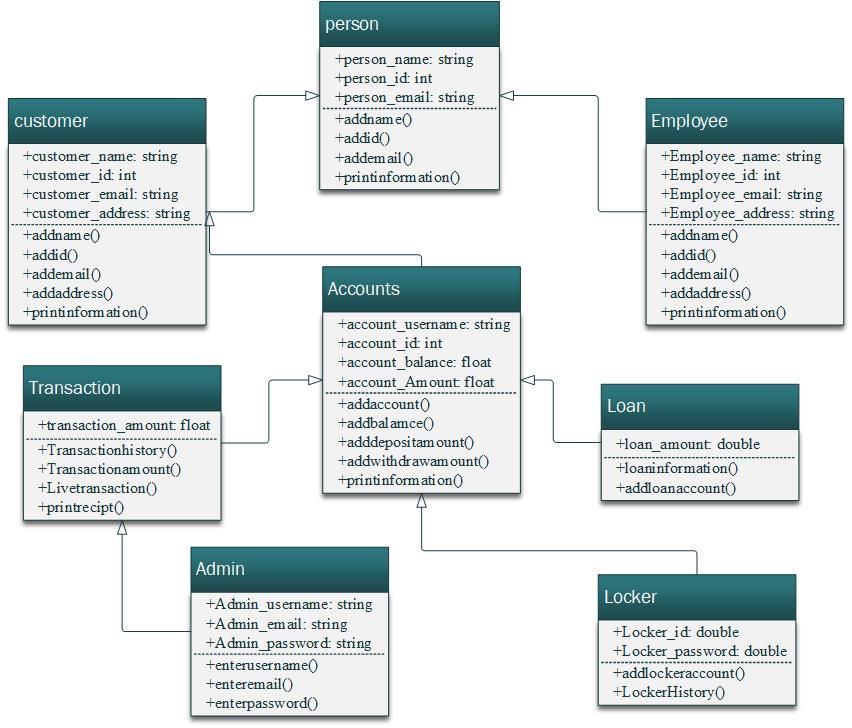
Here are some of the software quality attributes:

* 1. **Ease of use:** The software should be very easy to learn and use so that it can be made user friendly. In this way the user or customer can interact easily with the software.
  2. **Efficient:** The software should be efficient that means that the processing time will be reduced, and the completing time of any task will also be reduced, and it also include the minimum use of recourses.
  3. **Reusability:** In this process, the software being developed should be portable which means that a part of the software or whole software can be reused in other system again and again which is the best approach of any software.
  4. **Reliability:** The software is safety critical. Therefore, it must be more reliable to protect its information and it should also generate all the updated information in correct order.

# System Design

## Class diagram:

In class diagram first we include a person class which is base class which include person name, id, and email attributes. From this base class there are two classes inherit which are customer and employee it also include same attributes as person class. After than there is a account class which inherit form customer class which have attributes like username, balance and other which includes all the information of accounts. After that there are three more classes which are transaction loan and locker which are inherit from class which include transaction history, loan information and locker accounts.



## Activity diagram:

In activity diagram first admin was login and enter username and password after checking it there are two options of employee and customer which contains information of employee and customer. After than in customer section there are three more points of deposit, and with draw amount and add account option in add account option there are two more option of transaction check and locker check.

**Diagram

Description automatically generated**

## Domain model:

In domain model we include all the classes as we include in class diagram but there is a different er include only variables and the number of programs run and number of output.

**Diagram

Description automatically generated**

## System sequence diagram:

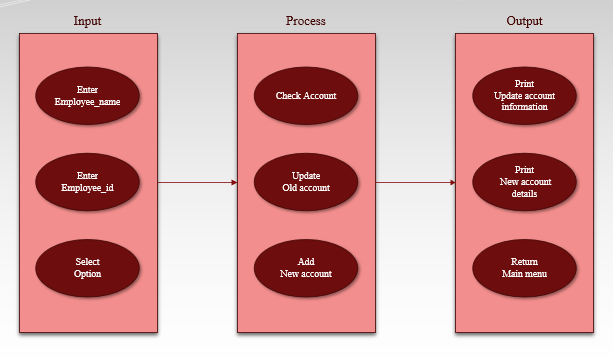
In system sequence diagram we add a admin attributes and all other classes as we use above in login class it can manage system and person class it manage person details like add or remove in employee, customer we can add, remove or update and in account we add account, deposit and withdraw money. And last locker it include locker history.

Graphical user interface, diagram

Description automatically generated

# System Architecture

The system architecture is a conceptual model that define the structure model and more view of a system. We used system architecture after the designing process and before the development stages it is a blueprint of over coding part. It define the structure of code.

It consists of three elements.

1. Processing.
2. Data.
3. Connection.

In bank management system we first do the processing in which we get the input form the user and return this into output like:

# System Testing

**Performance testing** will be done first, will plan a series of test according to the software. Initially all the tests will be done by development team to check all the bugs in the program. Then testing team will test the program by different inputs. **Alpha Testing** will be performed during the team’s testing. Then will apply this management system will be provided to user for further testing to ensure **Beta Testing. Acceptance testing**will be performed in the end to check the behavior of system in real world.

# References

* Real Life Banking management system