SRS FOR YOUR BANKING APPLICATION

Table of Contents

1. Introduction

This document gives detailed functional and nonfunctional requirements for the bank management system. This product will support easy banking modules for customer and employee. The purpose of this document is that the requirements mentioned in it should be utilized by software developer to implement the system.

1.1 Purpose:

This banking application provides is specifically developed for easy banking for Cash deposit/withdrawal, Balance Enquiry, Funds Transfer to another account in the same bank and other bank account, Making draft, Edit account details, Generating new password and Change of password, Printing passbook, Deleting account.

The Traditional way of maintaining details of a user in a bank was to enter the details and record them. Every time the user need to perform some transactions he has to go to bank and perform the necessary actions, which may not be so feasible all the time. It may be a hard-hitting task for the users and the bankers too. The project gives real life understanding of easy banking and activities performed by various roles in the supply chain. Here, we provide an automation for banking system through our C++ application. Easy banking system project captures activities performed by different roles in real life banking which provides enhanced techniques for maintaining the required in- formation up-to-date, which results in efficiency. The project gives real life understanding of easy banking and activities performed by various roles in the supply chain.

1.2 Scope:

This Product will automate of banking transaction process.This Project investigates the entry threshold for providing a new transactionservice channel via the real options approach, where the entry threshold is established by using an easy banking system designed for the use of normal users(individuals), Industrialists, Entrepreneurs, Educational Institutions(Financial sections), Organizations and Academicians under transaction rate uncertainty.

1.3 Overview

The system provides easy solution to banks.

Overview: The SRS will include two sections, namely

Overall Description: This section will describe major components of the system, interconnections, and external interfaces.

Specific Requirements: This section will describe the functions of actors, their roles in the system and the constraints faced by sys- tem.

2. General description

2.1 Product Perspective:

The client will have client interface in which he can interact with the banking system. It is a computer applicatiion based interface which will be the main page of the banking application. Starting a page is displayed asking the type of customer he is whether existing customer or a new customer. Then the page is redirected to login page where the user can enter the login details. If the login particulars are valid then the user is taken to main page where he has the entire tr list that he can perform with the bank. All the above activities come under the client interface.

The administrator will have an administrative in- terface which is a GUI so that he can view the entire system. He will also have a login page
where he can enter the login particulars so that he can perfor m all his actions. This administrative interface provides different environment such that he can maintain data- base &
provide backups for the information in the database. He can register the users by providing them with account
number, password & by creating account in the database. He can make draft & perform action to money transfer to same bank account and another bank account.

2.2 Software Interface:

Front End Client:

The system is a c++ based application clients are requiring using CODE BLOCKS IDE

3. Functional Specifications

This section provides the functional overview of the product. The project will require the PHP as a front end and at the back end the database MYSQL will be running. Various functional modules that can be implemented by the product will be

1.Choice for Employee/Customer

2. User details

3. Cash deposit

4. Cash Withdrawal

5. Balance Enquiry

6. Money Transffer

7. Print passbook

8. Edit Account

9. Delete Account

10.Exit

3.1 Login:

Customer logins by entering customer or employee account no. & a login password.

3.2 User Details:

This choice is for account holder’s details like account no,date of birth,address.

3.3 Cash Deposit:

A customer is allowed to enter the amount which he/she wishes to deposit.

3.4 Cash With drawal:

A customer is allowed to enter the amount which he/she wishes to withdraw. If the entered amount is less than the available balance and if after withdraw if the minimum required balance is maintained then allow the transaction.

3.5 Balance Enquiry:

A customer is allowed to check his/her current balance.

3.6 Money Transffer:

A customer can transfer the desired amount of money from his account to same bank account and other bank account.

3.7 Print Passbook:

A customer can update his/her passbook after transaction.

3.8 Edit Account:

A customer can edit his account details by this choice.

3.9 Delete Account:

An employee or staff of the bank management can delete account of a customer.A customer can’t do it.

3.10 Exit:

A customer can exit or quit from the system by this choice.

4. Interface Requirements

4.1 GUI

This is interface must be highly intuitive or interactive because there will not be an assistance for the user who is operating the System. At most of the places help desk should be provided for users convenience. The screens appearing should be designed in such a manner that it can draw User attaraction towards the new plans for the customers.

Also the pin and password confidentiality should be maintained,

This can be done by using asterisks at the password panel.

Proper security messages should be displayed at most of the places.

4.2 Hardware Interface

Various interfaces for the product could be

1. Pc or Laptop

2. Keypad

3. Minimum RAM-2 GB

4. Printer which can produce the hard copy.

5. Interface that connects the device to bank’s computer.

4.3 Software Interface

1. Any windows operating system.

2. The ‘Code Block Ide’ be installed.These products are open source products.

3. GNU GCC Compiler must be installed.

5. Performance Requirements

The system should be compatible enough to hold the general traffic. It should not get hang or show some other problems arising out due to large no of concurrent users . The system should be fast enough to meet the customer The high and low temperature should not affect the performance of the device. An uninterrupted transaction must be performed.

6.Constraints

* The information of all the users must be stored in a database that is accessible by the easy c++ application based

Banking System.

* The users access the Easy Banking System from any computer.

*The users must have their correct account number and passwords to enter into the Easy Banking System.

Design Constraints:

* Software Language Used

The languages that shall be used for coding Easy Banking System are c++ .

*Database design

In our database design, we give names to data flows, processes and data stores. Although the names are descriptive of data, they do not give details .So following DFD, our interest is to build some details of the contents of data flows, processes and data store. A data dictionary is a structured repository of data about data .It is a set of rigorous definitions of all DFD data elements and data structures .

7. Performance

7.1 Security

The banking system must be fully accessible to only authentic user.

It should require password for entry to a new environment.

7.2 Reliability

The application should be highly reliable and it should generate all the updated information in correct order.

7.3 Availability

Any information about the account should be quickly available from any computer to the authorized user. The previously visited customer’s data must not be cleared.

7.4 Maintainability

The application should be maintainable in such a manner that if any new requirement occurs then it should be easily incorporated in an individual module.

7.5 Portability

The application should be portable on any windows/Ubantu/Linux based system. It should not be machine specific.