**Software Requirements Specification**

**for**

**BANKING AND ACCOUNT**

**MANAGEMENT SYSTEM**

**Version 1.0 approved**

**BY:**

**TEAM – 3**

**19z226 – Manoj Kumar M**

**19z228 – Nandha Kishore V**

**19z234 – Rahul Raj D**

**19z242 – Sanjai S**

**19z261 – Vivekanandhan S**

**20z431 – Harish N R**

**Table of Contents**

**Revision History**

**1. Introduction**

1.1 Purpose

1.2 Product Scope

1.3 Overview

**2. Overall Description**

2.1 Product Perspective

2.2 Product Functions

2.3 User Classes and Characteristics

2.4 Software Interface

2.4 Operating Environment

2.6 User Documentation

**3. Functional Specifications**

3.1 Login

3.2 Validation

3.3 Get balance information

3.4 Withdrawal of money

3.5 Transfer Money

3.6 Customer info

**4. External User Interface Requirements**

4.1 GUI

4.2 User Interface

4.3 Hardware Interface

4.4 Software Interface

**5. Performance Requirements**

**6. Constraints**

**7. Performance**

7.1 Security

7.2 Reliability

7.3 Availability

7.4 Maintainability

7.5 Portability

**8. References**

**Appendix A: Glossary**

**Appendix B: Analysis Models**

**Appendix C: To Be Determined List**

1. **Introduction**

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

* 1. **Purpose:**

Online banking system provides is specifically developed for internet banking for Balance Enquiry, Funds Transfer to another account in the same bank, Request for cheque book/change of address/stop payment of cheques, Mini statements (Viewing Monthly and annual statements).

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 Internet banking and activities performed by various roles in the supply chain. Here, we provide an automation for banking system through Internet. Internet 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 Internet banking and activities performed by various roles in the supply chain.

* 1. **Product Scope:**

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

Complete elimination of paperwork in banking management by enabling the customer apply for leave as well as check their status through the system.

• Project Management:

Assign tasks and projects to employees, assign a project team and keep track of the progress.

• Report generation:

The manager will be able to generate timely reports in order to monitor the performance appraisals. The reports will be have all the information of an account.

• New Registration Process:

The admin will add an account details and sent to the new customer email. The manager will then have the ability to add an information to the database.

* 1. **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.

1. **OVERALL DESCRIPTION**
   1. **Product Perspective:**

The client will have client interface in which he can interact with the banking sys- tem. It is a web based interface which will be the web page of the banking application. Starting a page is displayed asking the type of customer he is whether ordinary or a corporate 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 a home page where he has the entire transaction 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 perform 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 username, password & by creating account in the database. He can view the cheque book request & perform action to issue the cheque books to the clients.

* 1. **Product Functions:**

A use case defines a goal-oriented set of interactions between external users and the system under consideration or development. Thus, a Use Case Scenario is a description that illustrates, step by step, how a user is intending to use a system, essentially capturing the system behavior from the user's point of view.

In order to create relevant use cases for the system, the following users for the system have been identified:

* Admin
* Manager
* Employee( including technicians and accountants )
* Customer
  1. **User and Classes and Characteristics:**

|  |  |  |
| --- | --- | --- |
| **User** | **Use case** | **Description** |
| Customer | Edit profile | Customer will be able to edit personal details such as emergency contacts but it needs to be verified. |
| Customer | Login | Customer logins by entering customer name & a login pin. |
| Employee | Add new customer | Employee can request to add new account to the customer |
| Manager | Verification of new account | Manager will verify the details of the newly requested account and customer and will approve. |
| Customer | Balance Enquiry | Customer can check the account balance by entering specific details |
| Customer | Shopping with ATM cards | Customer can shop using the ATM or Credit card in the website. |
| Customer | Requesting Credit Cards | User can request credit cards by giving the required instruction. |
| Manager | Accepting credit  Card request | After the verification manager will approve the credit card request. |
| Employee | Balance update | Updating the balance of the customers after every withdrawal and deposit in banks. |
| Customer | Transfer Money | Customer can transfer the desired amount from the account. |

* 1. **Software Interface**

Front End Client:

The system is a web based application clients are requiring using modern web browser such as Mozilla Firefox 1.5, Google chrome.

\* Web Server:

The web application will be hosted on one of the apache server.

\* Back End:

We use backend as MongoDB.

* 1. **Operating Environment**

MONGODB:

* MongoDB **stores data in flexible, JSON-like documents**, meaning fields can vary from document to document and data structure can be changed over time
* The document model **maps to the objects in your application code**, making data easy to work with
* MongoDB is free to use. Versions released prior to October 16, 2018 are published under the AGPL. All versions released after October 16, 2018, including patch fixes for prior versions, are published under the [Server Side Public License (SSPL) v1](https://www.mongodb.com/licensing/server-side-public-license).
  1. **User Documentation**

The software give access to four kind of users:

* 1. Administrator : The management will have the administration access to add , delete, and modify information stored in the database
  2. Customer : Customer will have access to only view the data stored in the database.
  3. Authorized User: Bank employee can add new customer details and new account registration. And updating the transactions of the customer.
  4. Manager: Manager of the bank can verify and approve the important things of the customer.

1. **Functional Specification**

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

1. Login

2. Validation

3. Get balance information

4. Withdrawal of money

5. Transfer Money

6. Customer info.

* 1. **Login**

Customer logins by entering customer name & a login pin.

* 1. **Validation**

When a customer enters the ATM card, its validity must be ensured. Then customer is allowed to enter the valid PIN. The validation can be for following conditions

Validation for lost or stolen card

When card is already reported as lost or stolen

then the message “Lost/Stolen card!!!”.

Validation for card’s expiry date

If the card inserted by the customer has crossed the expiry date then the system will prompt

“Expired Card”.

Validation for PIN

After validating the card, the validity of PIN must be ensured. If he/she fails to enter valid code for three times then the card will not be returned to him. That means the account can be locked. The counter for number of logins must be maintained

Get balance information:

This system must be networked to the bank’s computer. The updated

database of every customer is maintained with bank. Hence the balance information of every account is available in the database and can be displayed to the customer.

* 1. **Payment of Money**

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.

* 1. **Transfer of Money**

The customer can deposit or transfer the desired amount of money.

* 1. **Transaction Report**

The bank statement showing credit and debit information of corresponding account must be printed by the machine.

* 1. **Technical Issues**

This product will work on client-server architecture. It will require an internet server and which will be able to run PHP applications. The product should support some commonly used browsers such as Internet Explorer, Mozilla Firefox.

1. **External Interface Requirements**
   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.

* 1. **User Interfaces**

The user interface for the system will be a web page on the Internet. The user interface will be limited to the types of controls that can be generated using HTML, Java script.

* 1. **Hardware Interfaces**

EMS should be able to work on a computer with the following minimum hardware specifications:

* OS: Windows 7/8/10 and Linux
* CPU: Pentium III (700MHz) and above
* Memory: 128 MB and above
* Capacity: 4GB of hard drive
* Others: Network interface card, mouse, keyboard, and monitor.
  1. **Software Interfaces**

1. Any windows operating system.

2. The Visual Studio Code must be installed. For the database handling Postman and MongoDb must be installed. These products are open source products.

3. The final application must be packaged in a set up program, so that the products can be easily installed on machines. This application must be networked to corresponding banks.

1. **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.

1. **Constraints**

The information of all the users must be stored in a database that is accessible by the online Banking System.

\* The Online Banking System is connected to the computer and is running all 24hours a day.

\* The users access the Online Banking System from any computer that has Internet browsing capabilities and an Internet connection.

\*The users must have their correct usernames and passwords to enter into the Online Banking System.

Design Constraints:

\* Software Language Used

The languages that shall be used for coding Online Banking System are HTML ,CSS , JAVASCRIPT, NodeJS, ExpressJS, ReactJS and MongoDB. For working on the coding phase of the Online job portal System Web Sphere Application Server/WebSphere Application Server CE Server needs to be installed.

\*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.

1. **Performance**
   1. **Security**

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

It should require pin for entry to a new environment.

* 1. **Reliability**

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

* 1. **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.

* 1. **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.

* 1. **Portability**

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

1. **REFERENCES**

www.w3schools.com

www.roseindia.net

www.dbforums.com

www.ibm.com

<http://tomcat.apache.org/>

**Appendix A**: Glossary

**AIMS** - ATM Information Management System.

**ATM** - An unattended electronic machine in a public place, connected

to a data system and related equipment and activated by a bank

customer to obtain cash withdrawals and other banking services

**CDMA** - Code Division Multiple Access, a reliable data

communication protocol.

**CMS** - Card Management Software developed by KPM Bank.

**Dial-Up POS** - A message format for low cost communications.

Internet An interconnected system of networks that connects computers

around the world via the TCP/IP protocol.

Smart Card Card without hardware which stores the user’s private keys

within a tamper proof software guard.

**TCP/IP** -Transmission Control Protocol/Internet Protocol.