Software Requirements Specification

for

Banking Management System

Prepared by Team:

Srihari Adiga B S
Rapaka Vivek
Srikrshna P
Samyam N

Table of Contents

1. Introduction

This document outlines the bank management system's functional and nonfunctional requirements in depth. This product will support transactions via online banking. This document's goal is to have the software developer use the requirements listed in it to create the system.

1.1 Purpose

The online banking system offers features designed specifically for internet banking, such as checking your balance, transferring money to another account at the same financial institution, requesting a checkbook, changing your address, or stopping the payment of checks, and 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 needs to perform some transactions he has to go to the bank and perform the necessary actions, which may not be so feasible all the time. It may be a hard-hitting task for the bankers too.. 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 information 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.

1.2 Intended Audience

The product is designed for use by normal users(individuals), Industrialists, Entrepreneurs, Educational Institutions(Financial sections), Organizations to carry out their financial transactions.

1.3 Product Scope

This product will automate the banking transaction process.

1.4 References

https://www.pesuacademy.com/Academy/

https://www.w3schools.com/sql/sql_intro.asp#:~:text=What%20is%20SQL%3F,for%20Standardization%20(ISO)%20in%201987

https://docs.python.org/3/tutorial/

2. Overview

The system provides a method to implement Online Banking. 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 system.

3. Overall Description

3.1 Product Perspective

A client interface will be available for him to use to communicate with the banking system. It is a web-based interface where the user can submit their login information after being forwarded to a login page. If the login information is accurate, the user is directed to a home page with a list of every transaction he can do with the bank. The client interface includes all of the aforementioned activities.

To view the complete system, the administrator will have access to a GUI-based administrative interface. Additionally, he will have a login page where he can input his login information and carry out all of his actions. This administrative interface gives him access to a variety of environments that enable him to manage the database and offer backups of its contents. He can create accounts in the database and register people by giving them usernames and passwords.

3.2 Design and Implementation Constraints

The interface must be user-friendly. The languages that shall be used for coding the Banking System are:

Front-end : Streamlit / Tkinter

Back-end : Python,sql

Database : Realtional database

4. External Interface Requirements

4.1 User Interfaces

We provide different convenient and easy-to-use interfaces for each of the functionalities provided by the Banking System. Each interface includes buttons and links to navigate between the pages and to perform other required operations.

4.2 Hardware Interfaces

- *Standard PC*

- *128MB RAM or more (256 recommended)*

4.3 Software Interfaces

Front End Client:

The system is a web-based application , hence clients are required to use any web browser such as Google Chrome, Mozilla Firefox, Brave.

Requires Python and included Libraries to be installed.

Back End:

MySQL will be used for processing and retrieval data from the database

5. Functional Requirements

The product's functional overview is provided in this section. Python will be needed for the project, along with Streamlit for the front end and MySql for the back end. The numerous functional modules that the product will incorporate are

1. Login
2. Validation
3. Get balance information
4. Withdrawal and Deposit
5. User Information
6. PrintTransactions

5.1 Login

User can login by entering username and a login pin.

The Administrator has the option to create/register a new user account.

5.2 Validation

An authentication process to verify the username and login pin. If the credential matches, the user will be directed to the banking home page where he/she can carry out further processes, if not, a warning message will be displayed.

5.3 Get Balance Information

The updated database of every user is maintained with the bank. Hence the balance information of every account is available in the database and can be retrieved whenever required.

5.4 Withdrawal and Deposit

A user is allowed to enter the amount which he/she wishes to withdraw or deposit. If he/she wants to withdraw, then the minimum requirements to withdraw with respect to the bank is checked and only then a valid transaction can be made.

5.5 Print Transactions

This Function Display Recent transaction made by the user with appropriate details and allows to view transaction made during specific period

5.6 User Information

The bank statement showing credit and debit information of corresponding account will be displayed. (General account details of the user)

6. Non-functional Requirements

6.1 Performance Requirements

The system should be compatible enough to hold the general traffic and be fast enough to meet the customer requirements. Must support multiple uninterrupted transactions.

6.2 Safety or Security Requirements

The banking system must be fully accessible to only authentic or valid users and provide secure methods to carry out transactions.

6.3 Availability

The account information should always be obtainable to user without any occurrences of data loss.

6.4 Reliability

The application should be highly reliable and it should sustain in any condition and consistently give correct details.

6.5 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. Maintenance should be cost effective and easy.

.