Software Engineering

Mini Project Stage-II

Project Report

*Submitted by*

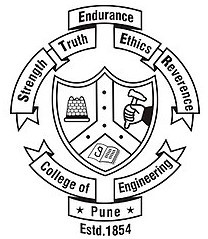
# Ankit Nehul Harshal Chavan

111903018 111903036

*Submitted to*

# Mrs. Tanuja Pattanshetti

Assistant Professor



Department of Computer Engineering,

College of Engineering, Pune

WEB APPLICATION OF BANK MANAGEMENT SYSTEM

## Problem statement

## Banking Management System

## 1.2 Objectives

## The main aim of designing and developing this Internet banking System primarily based Engineering project is to provide secure and efficient net banking facilities to the banking customers over the internet.

## JavaScript, MYSQL database used to develop this bank application where all banking customers can login through the secured web page by their account login id and password.

## Users will have all options and features in that application like get money from banks, money transfer to others, and send cash or money to inter banking as well as other banking customers by simply adding them as payees.

## 

* 1. **Motivation**

The main motivation of the project is to develop online Banking system for banks. In present system all banking work is done manually.

User have to visit bank to Withdraw or Deposit amount. In present bank system it is also difficult to find account information of account holder

1.4 TECHNOLOGY

Front End Client:

The system is a web based application clients are requiring

using modern web browser such as Google Chrome 97.0 or Mozilla

Firefox 1.5.

Back End:

We use backend as MY SQL.

VS code or any other JavaScript IDE.

Software Requirements Specification (SRS)

* 1. **External Interface Requirements**

## ■ User Interfaces

- New user:

He is a normal user which will have account in the bank. He can check bank balance and insert or withdraw money any time or transfer the funds from one account to another account.

- Admin:

Admin maintain the database and the permissions to users. Admin is master user of system.

## ■ Hardware Interfaces

The Hardware requirements are very minimal and the program can be run on most of the machines.

Processor - Intel 486/Pentium processor or better

Processor Speed - 500 MHz or above

Hard Disk- 20GB(approx.)

RAM - 64MB or above

Storage Space - Approx. 2MB

## ■ Software Interfaces

## 1. Language Used : JavaScript

## 2. Database : My SQL

## 3. User Interface Design : HTML

## 4. Web Browser : Google Chrome or any other web-browser

1.6 Non-functional Requirements

## Safety Requirements

## Secure databases.

## Data should be backup.

## There should be power backup for server.

## Security Requirements

Security:

The website does not allow access to any functionality by directly jumping to any particular link to that function’s page. Additionally, anything that is needed to be done can only be done by first logging in.

Data Integrity:

The project does not allow entry of data in case data is invalid. This is very important as if invalid data is added, then it can cause large problems.

Automatic data processing:

A lot of information is processed by the project instead of relying on the user to add perfect information and perform numerous functions each time.

## ■ Software Quality Attributes

Usability:

The user are assumed to have basic knowledge of the computer and internet browsing. The proper user interface, online help and guide to use and maintain the system must be sufficient to educate the user on how to use the system without any problems.

Reliability:

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do. Availability: Banking system shall not experience any delay in the service response to their commands.

Security:

All users of the system shall not be able to perform actions which will cause harm to any person or damage of the system.

Maintainability:

There shall be an access on the server for the purpose of upgrading the software .

## ■ Advantages of Bank Management System

It is based on a Client-Server System, meaning multiple Admins can be supported.

It operates on a minimalistic User Interface so that any user of the system can do what they need to do with almost no training and extreme ease.

Erroneous data is not entered into the system and rendered invalid

Admin can automatically remove the registered users.

## ■ References

1. HTML - https://developer.mozilla.org/en-US/docs/Web/HTML
2. CSS - https://developer.mozilla.org/en-US/docs/Web/CSS
3. JavaScript - https://developer.mozilla.org/en-US/docs/Web/JavaScript
4. React Js - <https://reactjs.org/docs/getting-started.html>
5. Other resources from YouTube.

## ■ SUMMARY

We have developed a highly comprehensive and easy to use system for banking .It is easy to implement and requires no training to use. It provides options for Admins . It is error-proof and does large amount of work in the background. Bank management system has the responsibility to transfer funds from one user to another.

## ■ ERD

Diagram

Description automatically generated

## ■ UML and Explanation

1. **Use Case Diagram**

Use case diagram describes the scenarios in which your system interacts with user.

If user is new, then he has to register and registration has include relationship with form filling, i.e. User must fill the form to register.

Existing user can login, system must verify details before log in and login has extend relationship with display error, i.e. system needs to display error only in case of invalid credentials

When user choose check balance option, system must display the balance.

In case of fund transfer system compulsorily check for enough balance and then update balance or display error according to balance.

User can see the transaction history, system has to search in database to display transactions.

Diagram

Description automatically generated

1. **Class Diagram**

The purpose of class diagram is to model the static view of an application.

Diagram shows the different classes of system with their attributes. Plus means public, minus represents private and hashtag means protected.

Diagram

Description automatically generated

1. **Activity Diagram**

Activity diagram is used to demonstrate flow of control within system.

After start, if the user is existing then he/she has to login if no the he/she has to register, then system will verify the details.

System will display list of services. In case of transfer fund if balance is not enough it will display error and if sufficient then it will update the database. For transaction history, system will search in database and display history. If user don't want to continue then he/she can logout.

Diagram, schematic

Description automatically generated

1. **Component Diagram**

Component diagram does not describe functionalities of system but it describes the components used to make those functionalities.

There is one login management subsystem which has authentication component to verify details.

Then there is admin panel and customer panel providing functionalities to admin and customer respectively.

**Diagram

Description automatically generated**

1. **Sequence Diagram**

Sequence diagram is used to describe interaction among active objects within the system.

New user will register. Then there is some processing and then system will acknowledge user that his/her details are approved.

There is loop for login. User has to login, then details are verified in database, if details are invalid, user has to login again and if details are valid loop will break.

User will request for fund transfer and after fund transfer system will acknowledge the user. In case of transaction history first system will interact with database and then interact with user by displaying transactions

Calendar

Description automatically generated

* **OUTPUT SCREENSHOT**

Graphical user interface

Description automatically generated

Graphical user interface, website

Description automatically generated

Graphical user interface

Description automatically generated

A picture containing text

Description automatically generated

* **BUGS**

Could not connect to database.

* **GitHub Link**

<https://github.com/ankitnehul/SE-2>