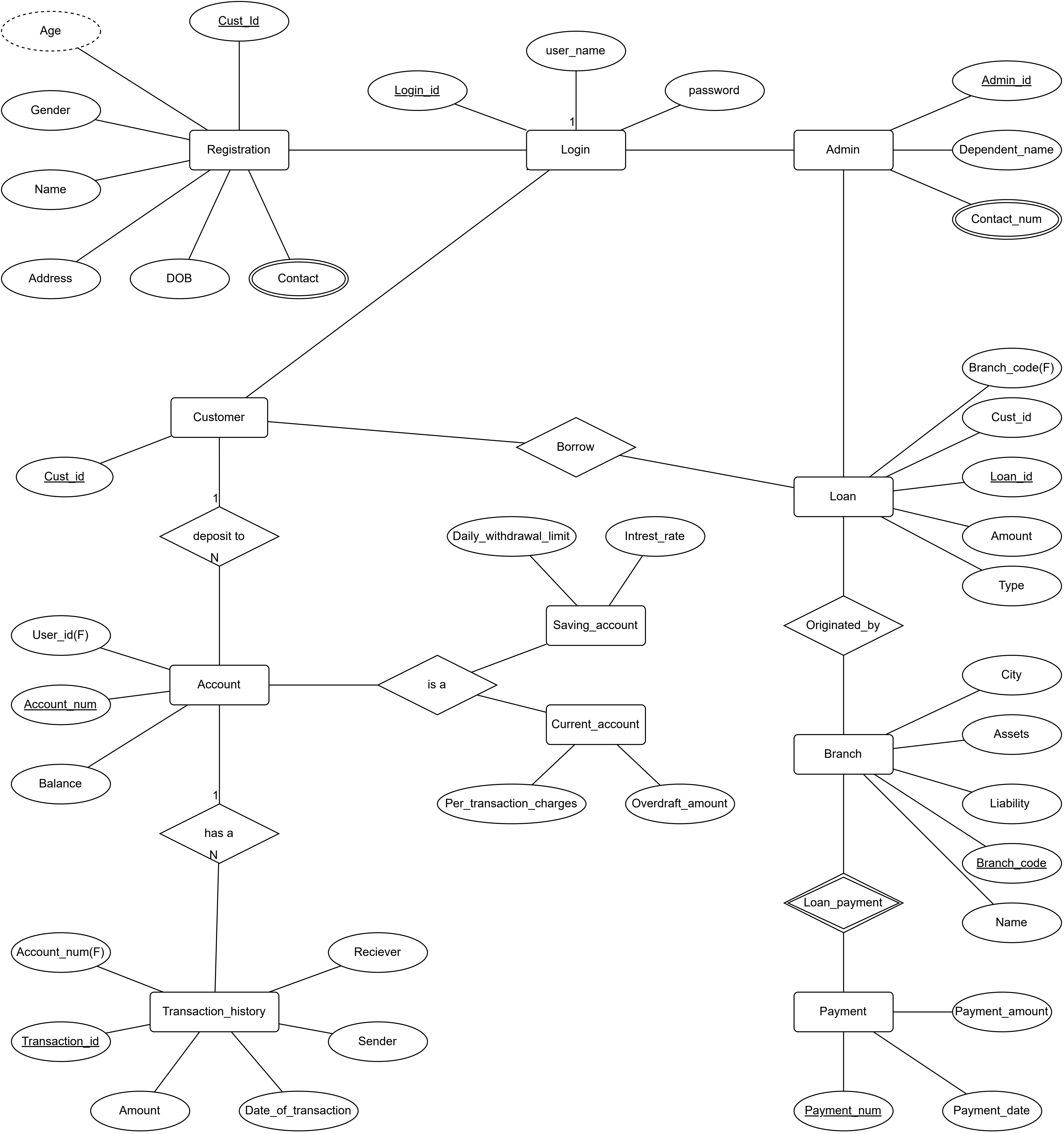

Age
Cust_Id
user_name
Login_id
password
Admin_id
Gender
Registration
1
Login
Admin
Dependent_name
Name
Contact_num
Address
DOB
Contact
Branch_code(F)
Customer
Cust_id
Borrow
Cust_id
Loan_id
Loan
1
deposit to
N
Amount
Daily_withdrawal_limit
Intrest_rate
Type
Saving_account
Originated_by
User_id(F)
City
Account
is a
Account_num
Assets
Current_account
Branch
Balance
Liability
1
Per_transaction_charges
Overdraft_amount
has a
N
Branch_code
Loan_payment
Name
Account_num(F)
Reciever
Transaction_history
Payment
Payment_amount
Transaction_id
Sender
Amount
Date_of_transaction
Payment_num
Payment_date

BankWiz

Karuna Sharma - 2001099
Harsh Gaur - 2001074
Aditya Ratnaker - 2001013

ER Diagram

1st February 2023

Entity

1. Registration
2. Login
3. Customer
4. Admin
5. Account
6. Saving account
7. Current account
8. Transaction_history
9. Loan
10. Branch
11. Payment

Attributes

1. Registration
    - Cust_id (Primary key)
    - Gender
    - User_name
    - password
    - Name
    - Address
    - DOB
    - Age (Derived Attribute)
    - Contact (Multivalued Attribute)

2. Login
   - Login_id (Primary key)
   - User_name
   - password
3. Admin
   - Admin_id (Primary key)
   - Dependent_name
   - Contact_num (Multivalued attribute)
4. Customer
   - Cust_id (Primary key)
5. Loan
   - Branch_code (foreign key)
   - Cust_id (foreign key)
   - Loan_id (Primary key)
   - Amount
   - Type
6. Branch
   - City
   - Assets
   - Liability
   - Branch_code (Primary key)
   - Name
7. Account
   - User_id (foreign key)
   - Account_num (Primary key)
   - Balance
8. Transaction_history
   - Account_num (foreign key)
   - Transaction_id
   - Amount
   - Date_of_transaction
   - Sender
   - Receiver
9. Payment (Weak entity)
   - Payment_amount
   - Payment_num (Primary key)
   - Payment_date

10. Saving account
    - Daily_withdrawal_limit
    - Interest_rate
11. Current_account
    - Per_transaction_charges
    - Overdraft_amount

-

BankWiz

Bringing Banking Solutions to your Fingertips........

20.01.2023

Team Members -

Karuna Sharma	- 2001099
Aditya Ratnaker	- 2001013
Harsh Gaur	- 2001074

Introduction

1.1) Purpose of this Document

The purpose of this document is to present a detailed description of the **BankWiz Online Banking Application**. This document will explain everything about this application, including what its purpose and core features will be, how the interface of the application will look, what and how the application will work under different circumstances, how to use it, and most importantly, how it will react to external stimuli. Moreover, this document will act as a user manual and guiding manual for all parties concerned. This document is intended for both the developers of the system and the concerned instructors of the course under whom we are developing our application and will be held liable for the approval or disapproval of the project by the respected teaching assistants.

1.2) Product Scope

An online banking application will be applicable everywhere, where banking exists and the need for banking exists everywhere. This application will try to achieve a more efficient and easier way to have a record of data through which everyone can access it according to their rights as compared to the traditional banking system. Every bank now uses an online banking system as it contains many features which can be accessed from anywhere in the world through the internet and ultimately provides ease of access to all its customers.

1.3) Overview

This online banking system demonstrates the ease of access and use among users. The main objective of the application is to develop an application that literally changes the way our traditional banking systems work. This application will be different from its competitors in the market because of its ease of availability, access, and use.

The SRS - Software Requirement Specification will describe some of the core features of this application, its interconnections, and the external interface of the system.

General Description:

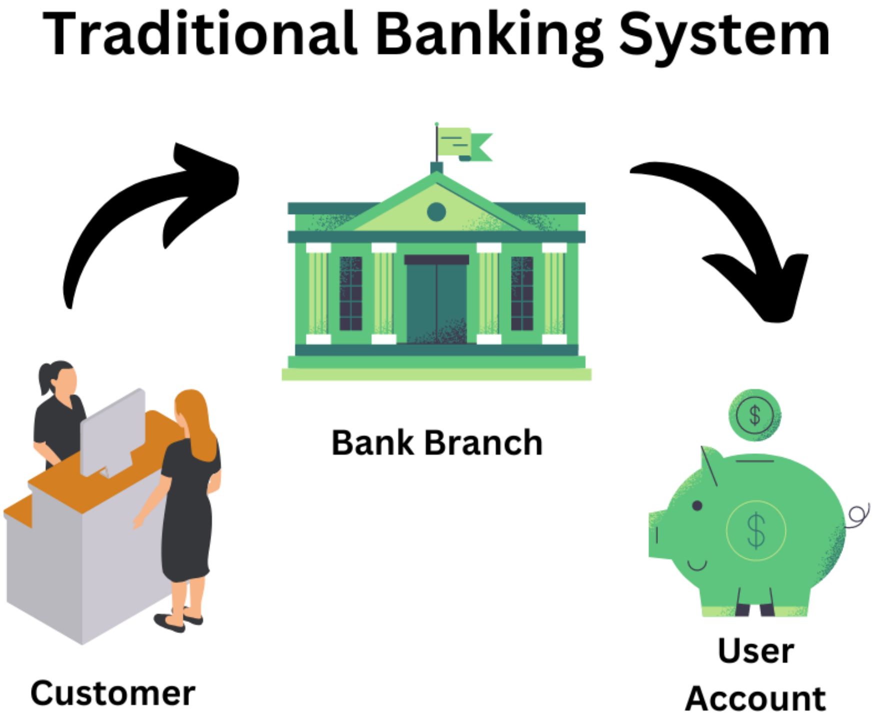
2.1) Objective of the User -

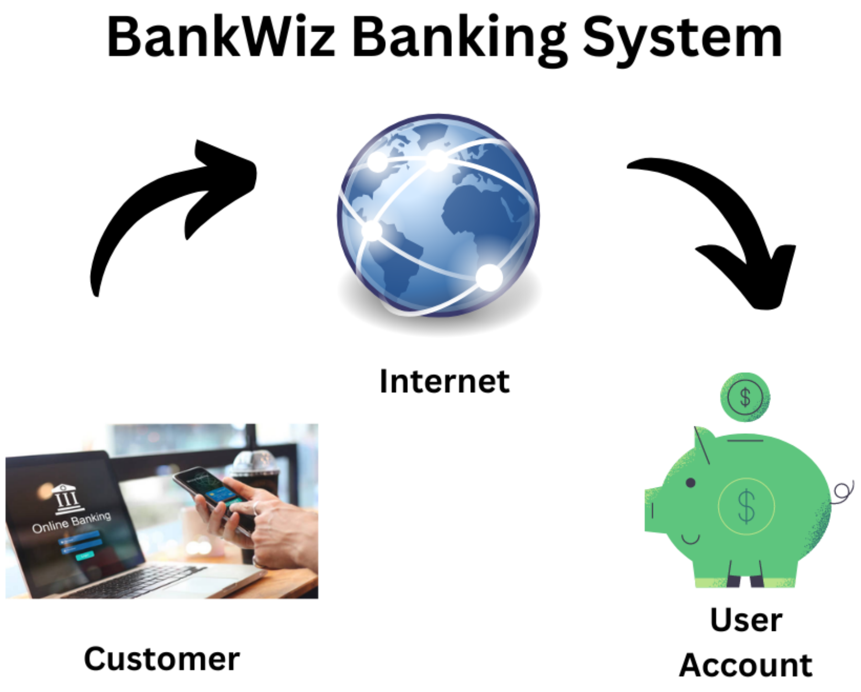
This application will try to achieve the following objectives -

1. To develop an application that is user-friendly.
2. To develop an application that gives you the benefit to access banking facilities from the comfort of your home.
3. To save fuel and paper as with this application their usage will be substantially reduced.

2.2) Application Perspective

The following figure illustrates the present scenario of the traditional banking applications that exist in the market.





In the traditional banking system, most of the time customers have to visit the bank in order to do some transactions, have some paperwork, and much more.

But now with this application, we are now trying to implement an altogether different type of application that is not only easy to use but also provides banking solutions to people in the comfort of their homes.

With this application, time usage will be reduced and tasks will be done fast instead of waiting for other humans and departments to complete their tasks and assign permission.

2.3) Application Features

This software application will contain a great combination of some core banking features and some new and innovative features -

- **A Safe and Secure Login System**

  This application will have a safe and secure login-based system for customers to access their credentials and make transactions and avail the benefit of their core features.

- **Opening New Account**

  To open a bank account one does not have to go through the long and tiring physical process to create a bank account. With this application, they will be able to open not only one bank account but multiple accounts even with Rupees 0/- in the account.

- **Online Balance Check**

  Once the user is authenticated this application will allow him/her to check the balance of not only one account but all other accounts they have.

- **A Customized ChatBox**

  This personalized chatbox will not only help the customers to solve their frequently asked queries but also help them to register the grievances that they are facing while using the application. Further on which the developers can improve.

- **View Transaction History**

  It will be easy for the customer to view his/her transaction history up to the past year. It will enable the customer to keep track of their transactions and accordingly manage further spending.

- **Balance Transfer**

  This system will provide a path for the customers of the bank to transfer their balance to other accounts in very simple and easy steps. A very small minimal transaction fee will be charged to complete the transaction.

- **Loan Facilities**

  This online banking application will allow users to apply for loan facilities in a hassle-free manner.

2.4) User Characteristics

Now, there are various kinds of users for the product. Usually, web applications are visited by various users for various different reasons.

The users include -

- **Admins**

  Admin users will access the system as the controller and will have all the privileges of the administrator.

- **Users**

  All the persons who want to utilize different types of banking services.

General Constraints:

Some general constraints need to be defined which will have a great impact on the overall success of the online banking project.

3.1) Hardware Requirements

As this application is an online Web-based application so a client-server will be the most suitable Organizational style for this system. Necessary Computer Hardware will be needed to run this application error-free. Also, an internet connection must be available in order for the user to access the data.

So, concisely the following hardware will be needed -

1) Computer Hardware - Memory, Processors, and much more.
2) Internet Availability.
3) 24*7 - Electricity Requirement.
4) A Safe and Secure Server Room.

3.2) Safety and Security

This application must remain safe and secure from all types of dangerous and unauthorized access as this application contains the personal data of all the customers, not only their personal accounts along with money information need to be secured at all costs. This application needs to identify the valid customer according to his/her bank details and password. So, it will be the most difficult task to prevent the system from major disasters by preventing unauthorized access to the system.

Requirements

How the online bank will interact with the environment, and what will be the functional and non-functional requirements? These all requirements should be defined here for providing a powerful base for the design phase of the application.

The design of the project will be completely dependent on the functional and non-functional requirements.

So, they need to be clearly and accurately defined for effectiveness.

Functional Requirements

This section provides the functional description and overview of the product. The project will require HTML, CSS, and JavaScript as a front end, and the database MongoDB will be running. There will be numerous functional modules that can be implemented in the product will be -

The login and authentication will be matched by the data stored in MongoDB. And we can further add details of the new users by sign_up form.

There will be unique collections related to every account. By which you can access the transaction history of a particular account.

We can transfer money from one account to the other by using the ACID properties of the database.

Loans/payments will be stored in the database with each transaction having its unique id.

ER Diagram

Entity Relationship Diagram

https://drive.google.com/file/d/1z7OfqVZVheDSFRbjRRW7mCOogR8CteEp/view?usp=sharing

Process Specification

We need to implement the following processes to incorporate all the entities mentioned in our ER Diagram -

- **Login**

  Each customer will have its User_id, user name, and password assigned to them when they sign up for the first time. This page will require all of these attributes for them to access the account.

- **Features of Bank**

  We know this fact that every user who visits the site of this online application system isn't always the user so we need to make sure that our home page of the bank should be looking great and provide the users with all the necessary information they need. Also, the page should be well organized and creates a great first impression on the user.

- **Creation of the Bank Account**

  A new visitor to the Bank would be interested in opening a new account in the Bank. so he must be provided with an easy path to create a new account in the online banking application.

- **Fill out the Form**

  Newcomers should have to fill out the form to register him/herself with the bank. After filling out the form, if the values inputted by the user were logically correct, his account will be created and his account number will be the fixed number assigned by the bank.

- **Welcome User**

  After the user successfully authorized himself, he/she will be provided with a welcome page offering different tasks. (Here, this interface will provide many of the functionalities, which the customer needs in the software). He/She has to choose a task to carry on.

- **Check the Balance**

  After logging in, if the user wants to check his balance he will have to click the balance check link. He/She will be able to see the list of their accounts connected by that particular customer_id. They can select the specific account of which they want to select the balance.

- **Transfer Balance**

  If a user wants to transfer his money to some other account, then this module will provide him with this opportunity. The customer needs to input the account details of the receiver. After this process, the server will check the balance of the user and if the transfer balance will be less than the account balance then the transfer will take place else he will be alarmed that he/she has a low balance.

- **Loan Facilities**

  If a user wants to take a loan then he needs to select the type of loan he wants and the amount required to borrow, someone from the bank will contact the customer over the phone.

Interface Requirements:

5.1.1) Hardware Interface

As it is a web application it can be accessed in any of the Operating systems that the user has. Any type of Web Browser will be able to access our online banking application, but preferably Google Chrome will give the best user experience.

Server Side

- **Operating System – Windows 9x/xp**
- **Processor - Pentium 3.0 GHz or higher.**
- **RAM - 256 Mb or more.**
- **Hard Drive - 10 GB or more.**

Client Side

- **Operating System - Windows xp or above, MAC or Linux/Unix.**
- **Processor - Pentium III or 2.0 Ghz or higher.**
- **RAM - 256 Mb or more.**

5.1.2) Software Interface

- **Client Side**

  HTML, CSS, JavaScript, ReactJS

- **Web Server**

  hg JavaScript, Node.js, Express.js

- **Database**

  MongoDB

5.1.3) Communication Interface

The customer must connect to the Internet to access the Website.

a) Dialup Modem of 52 kbps.
b) Broadband Internet
c) Dialup or Broadband Connection with an Internet Provider

5.1.4) Front-End Interface

In our online banking application, when a user comes he/she will see the following major components -

a) Landing page for everyone
b) Login Panel
c) Features Panel after Login - For users
d) Administration panel after login - For adminsh

4.2) Non-Functional Requirements

Those requirements which are not the functionalities of a system but are the characteristics of a system are called the non - functionalities. Every software application has some non-functionalities. Just fulfilling the requirements of the user is not a good task, keeping the system accurate, easy to maintain, reliable, and secure is also a basic part of software engineering. An online banking application must have the following non-functional requirements so that it could be said as a complete system.

a) **Conformance to specific standards.**

   This application must fulfill the requirements of instructors and should give a high score in performance.

b) **Performance Constraints**

   The application must give an optimum amount of performance. It should use less memory and will be easily accessed by the user. Memory Management should be done wisely so that none of the memory parts goes wasted.

c) **Hardware Limitations**

   It should be designed in such a way that cheap hardware must be installed to access and use it effectively. It should be platform-independent. There should be no hardware limitations. It should be designed to work with low-specification hardware so that it could easily work with high-specification hardware.

d) **Maintainable**

   Each of the new features should be designed in such a way that a new module can easily be integrated with it.

e) **Reliable**

   It's important to provide a failure-free execution of the program to the customer. Most importantly to provide operational reliability of our application to the customer.

f) **Testable**
g) **Security**

It’s important to ensure that the customer’s valuable information does not get out into the public domain.