IT - 314 Software Engineering

End Term Presentation

Group 57

Topic: Online Banking System

Reporting TA: Jash Rathi

Guided by: Prof. Jayprakash Lalchandani

Team Members

Dhwanil Shah - 201901450

Yash Shah - 201901210

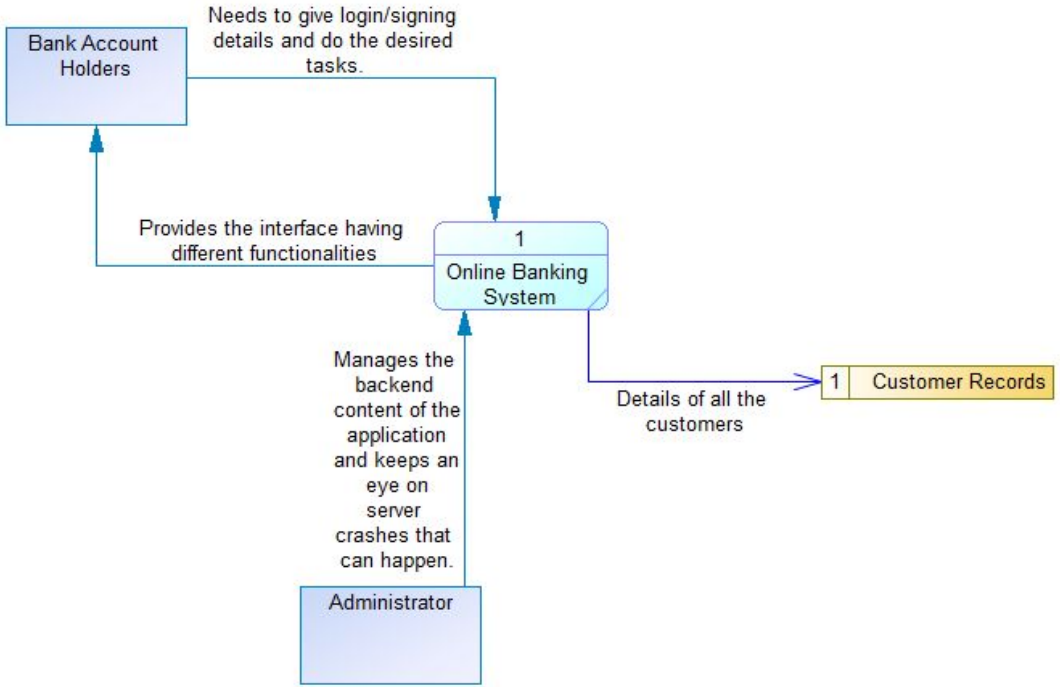
Dev Sanghvi - 201901231

Darshil Shah - 201901232

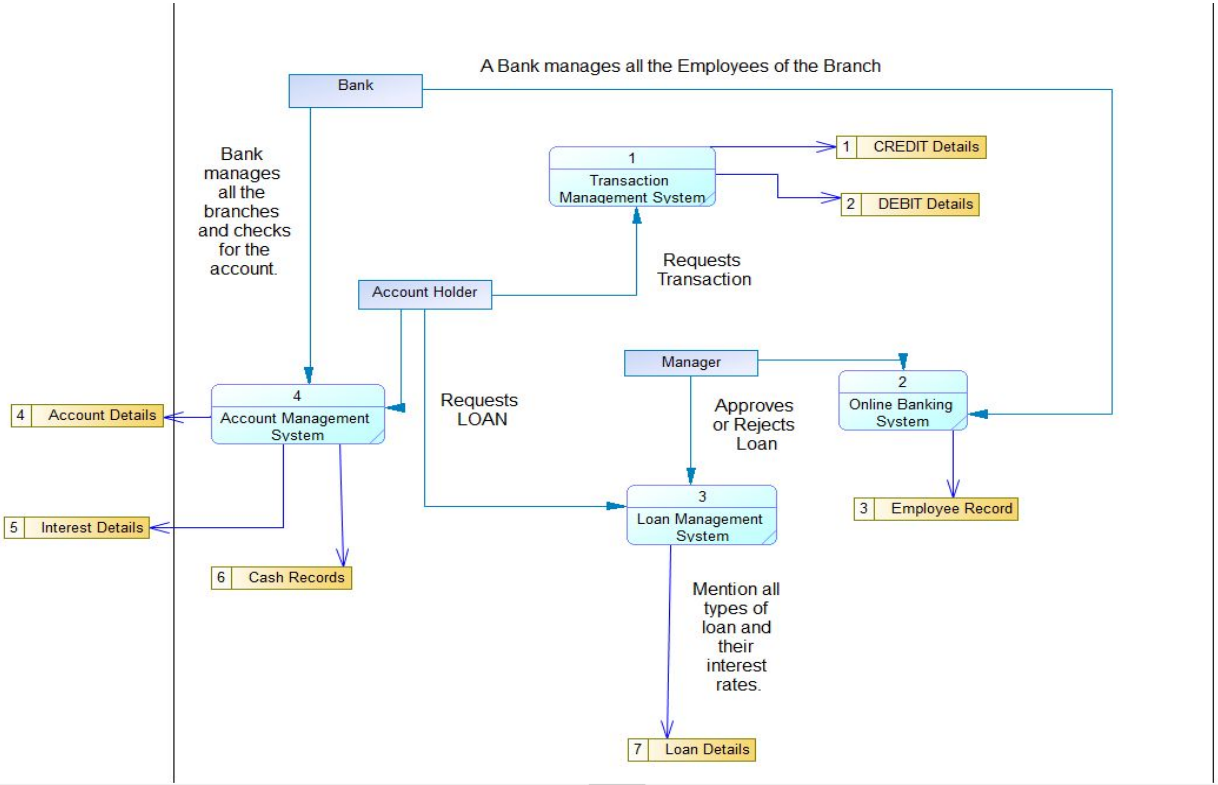
Harsh Mehta - 201901277

Jeel Faldu - 201901263

Data Flow Diagram (Level - 0: Context Diagram)



Data Flow Diagram (Level - 1: Overview Diagram)



User Stories

1. ‘forgot password’ option for existing account holders
2. KYC verification via online video call along with the document verification process.
3. Account holder should be able to perform various transactions such as transferring money to different accounts of the same or other banks.
4. Account holder should receive notifications through contact details provided during registration for all the transactions and during the sign-in process.
5. Account holder should get a detailed statement of the various transactions that occurred through his/her account based on the timeline selected.
6. Option for updating registration details for existing account holders.

User Stories (Contd.)

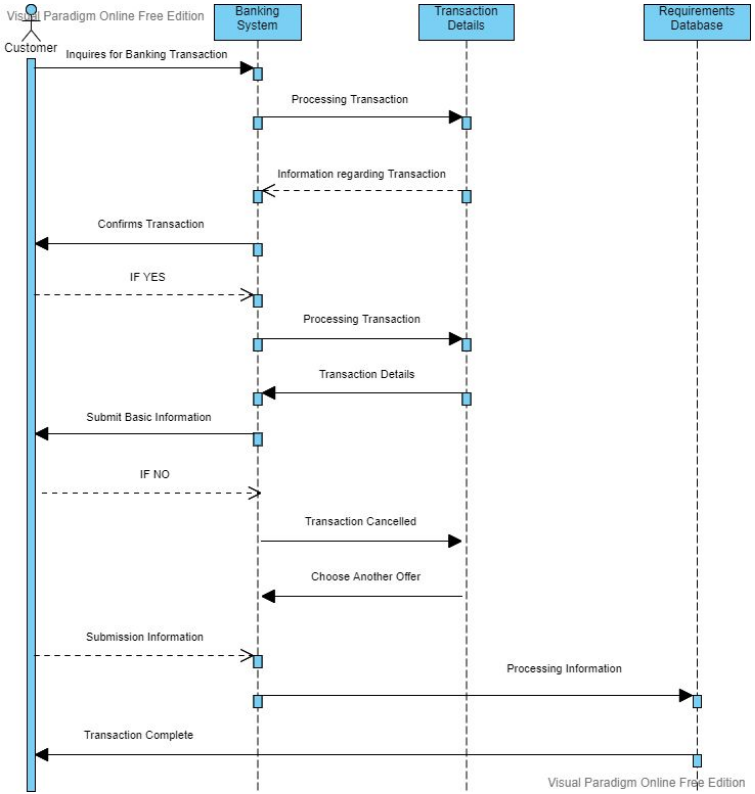
1. Applying for various loans available.
2. Loan document verification by the loan officer for accepting/rejecting a loan application.
3. As an account holder, I want the system to have an EMI calculator which calculates the EMI based on a particular loan amount, loan interest, and the duration for which the loan is taken or planning to take.
4. Account holders should be able to invest money through fixed deposits or recurring deposits.
5. Account holders should be able to pay various different bills such as mobile, telephone, electricity, gas pipeline, etc.
6. Account holders should receive OTPs as a security measure for every transaction and log-in request.

User Stories (Contd.)

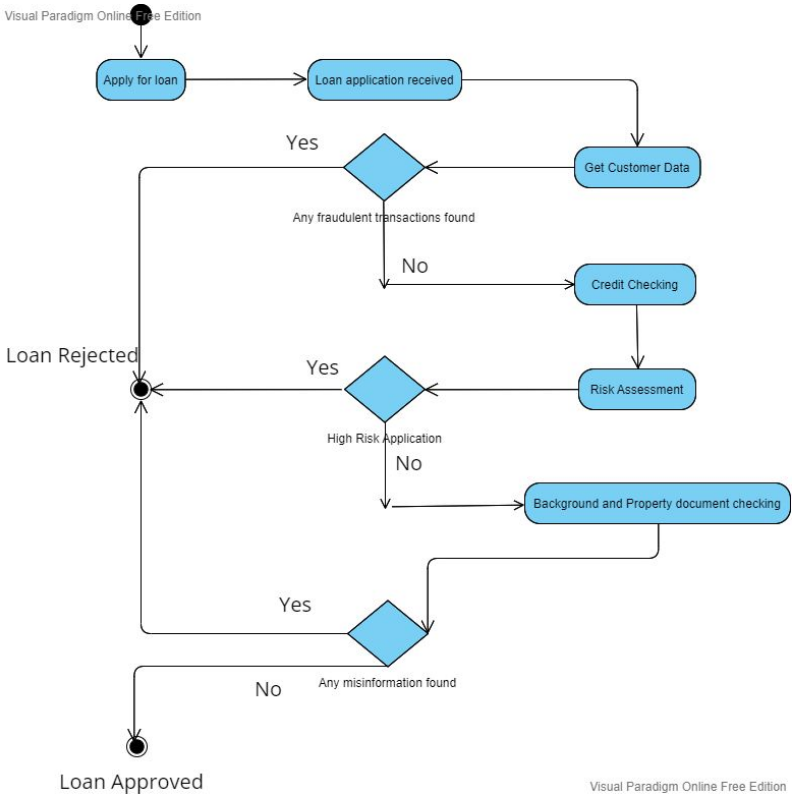
1. Account holder should be able to log out of the system when all the required transactions/tasks are completed.
2. The system should log out a particular user after 15 minutes of inactivity.
3. Account holder should be able to delete the account from the system.
4. The system must verify the authenticity of the details before deleting any account.
5. Account holder should have the option of connecting to the help desk for resolving queries.

Use Case Diagram

Sequence diagram



Activity Diagram (Loan disbursal use case)



Application Architecture

- When visiting the website that we are developing the user would be asked to initially register (if visiting the website for the first time) or login (if already is an user).
- The Login and the Sign Up tabs would initially be present on the Navigation bar which already consists of different features that can be accessed very easily and the user would not be confused of finding the features at different locations.
- The Sign Up tab consists of different field informations like the full name, date of birth, permanent address, type of account that the user wants to open, email id and a password.
- The Login tab consists of entering email id and the password that has been set initially while Signing Up.
- The Home Page would also consist the information regarding our websites' features and would be redirecting the user to specific features when clicked on it.

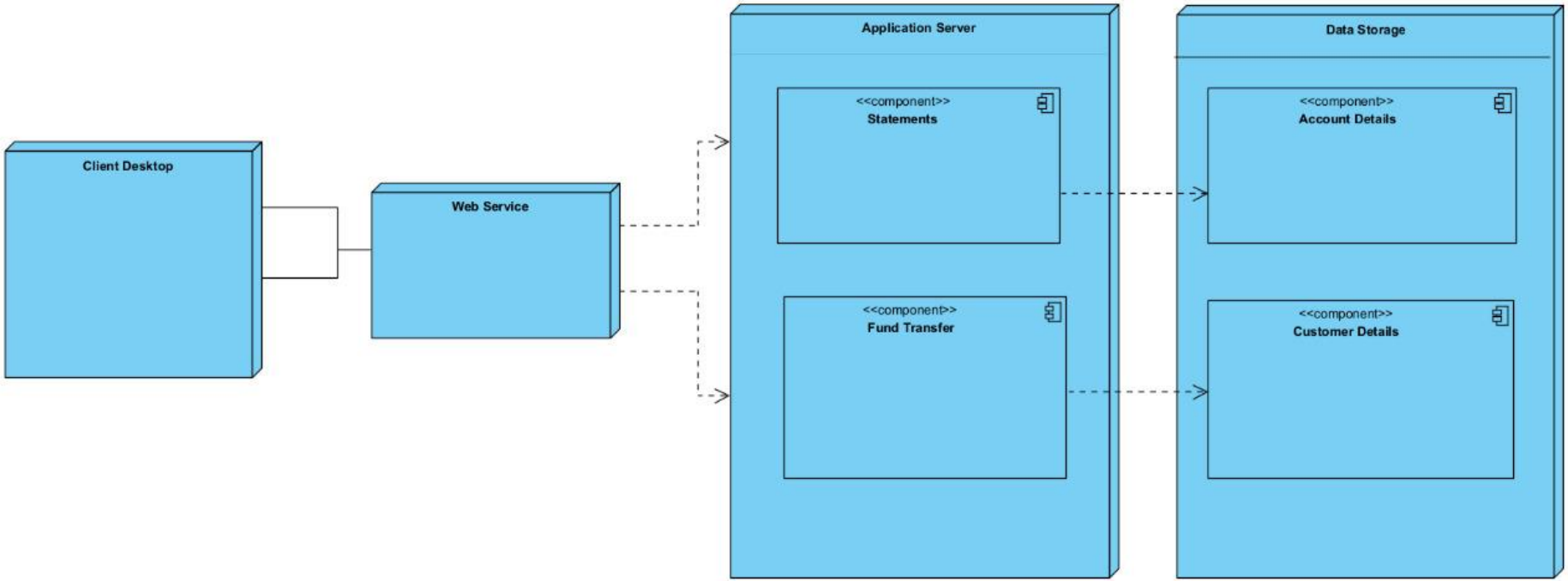
Technologies / Tools and Libraries

- HTML
  - ➢ The Front-End of the website is developed mostly using HTML. The different elements that would be inserted in the website would solely be added using HTML tags.
- CSS, Bootstrap
  - ➢ The designing of the site i.e. the colors added, the sizes of the buttons, the styling of different tags, etc. is done using CSS and Bootstrap (which provides built-in CSS framework).
- JavaScript
  - ➢ The functioning of buttons on clicking it, the working that is provided by different tags, verification of passwords of users, etc. work will be done using JavaScript.
- NodeJS
  - ➢ Here NodeJS is used mainly to develop the backend of the system which shows the different ways how a database is connected to the system.
- PostgreSQL
  - ➢ The PostgreSQL is used to connect the system to a data store where all the data is stored and modified whenever required.

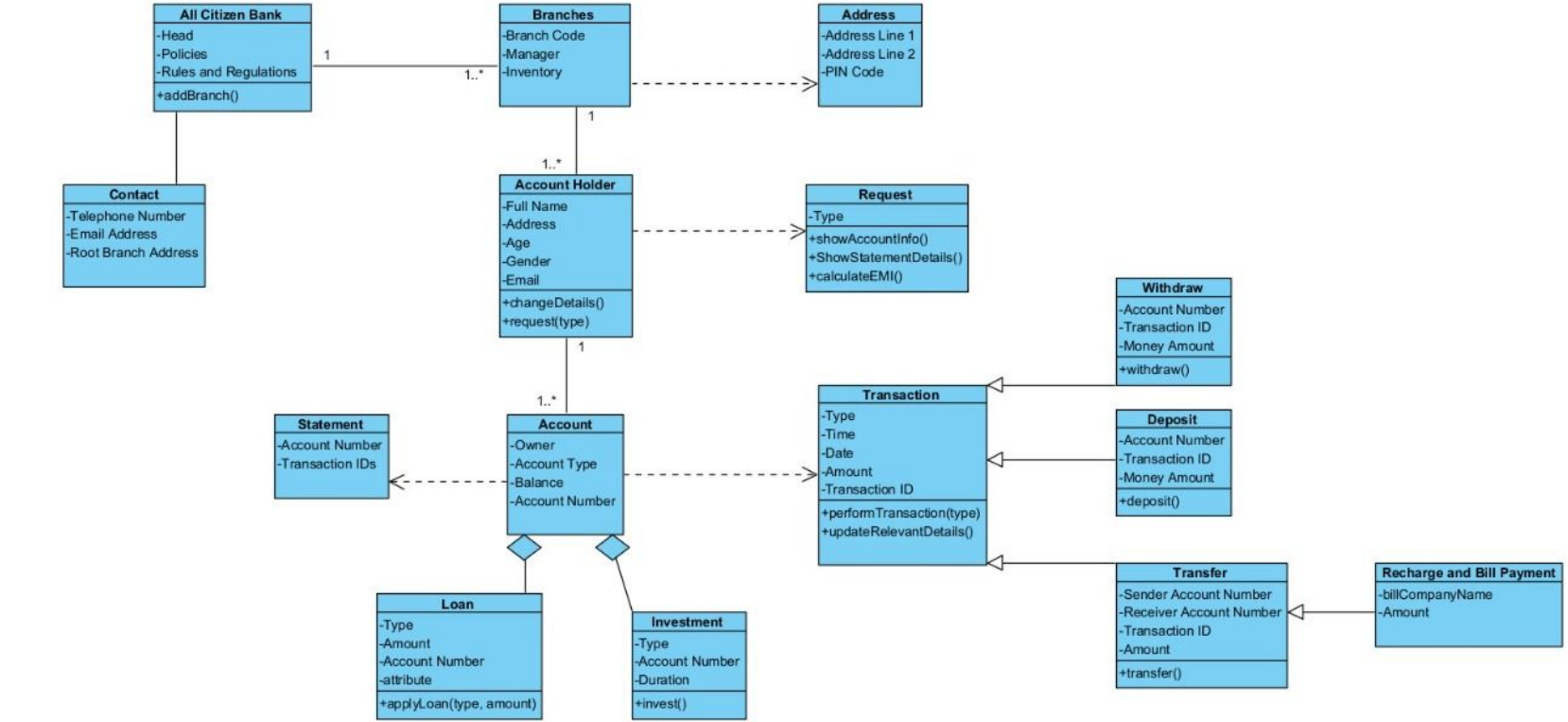
Summary of Implementation

- We have implemented the front-end designing of the system to a greater extent.
- The website is quite user-friendly in terms of access of functionalities and use of the functionalities.
- It mainly has different functionalities like login/signup and validation of login and signup, transferring of money to other account holder, etc.
- The whole of the frontend of the website has been implemented using HTML, CSS (Bootstrap) and JavaScript.
- The backend is developed using NodeJs, ExpressJs with NodeJs and the database is implemented using PostgreSQL.
- PostGreSQL is used mainly to show the uniformity that is to be maintained as all the fields in the information of user should be available and relations must be made between different tables strongly.
- We have successfully completed the connection of PostgreSQL with the signup/login page and transfer money page.

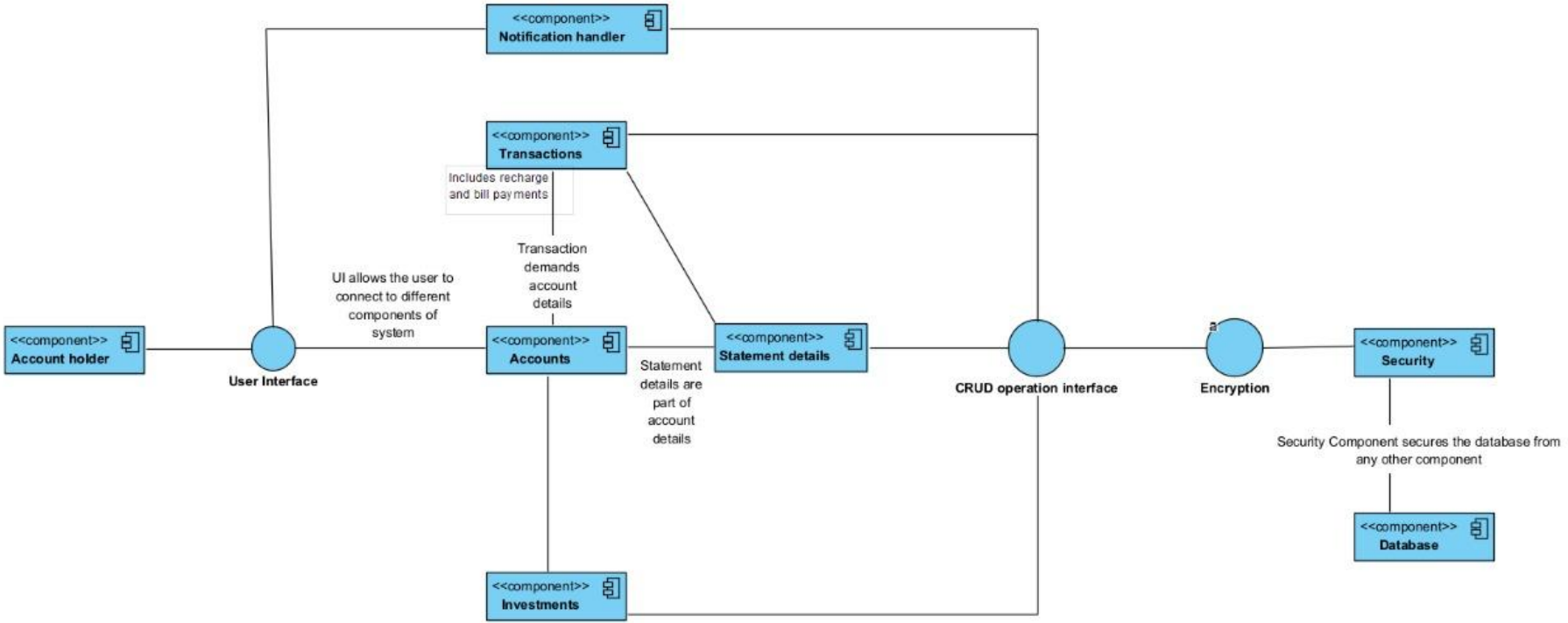
Deployment Diagram



Class Diagram



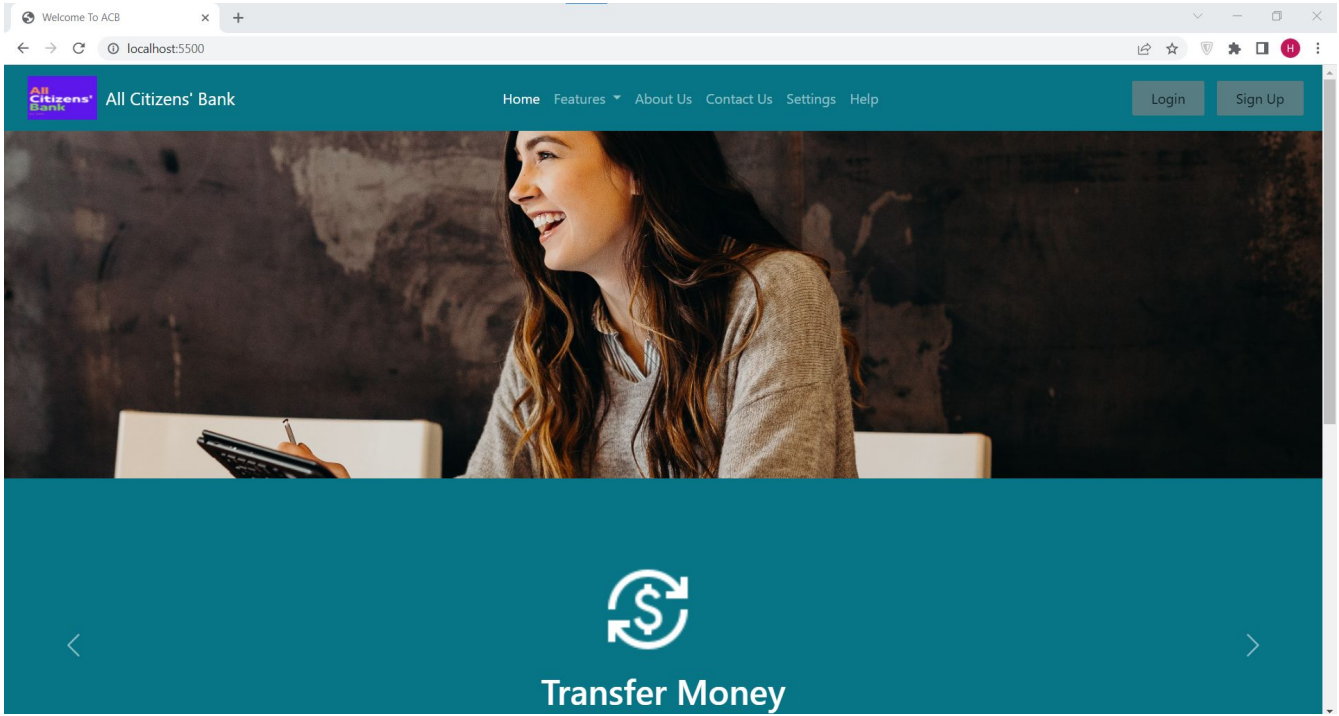
Component Diagram



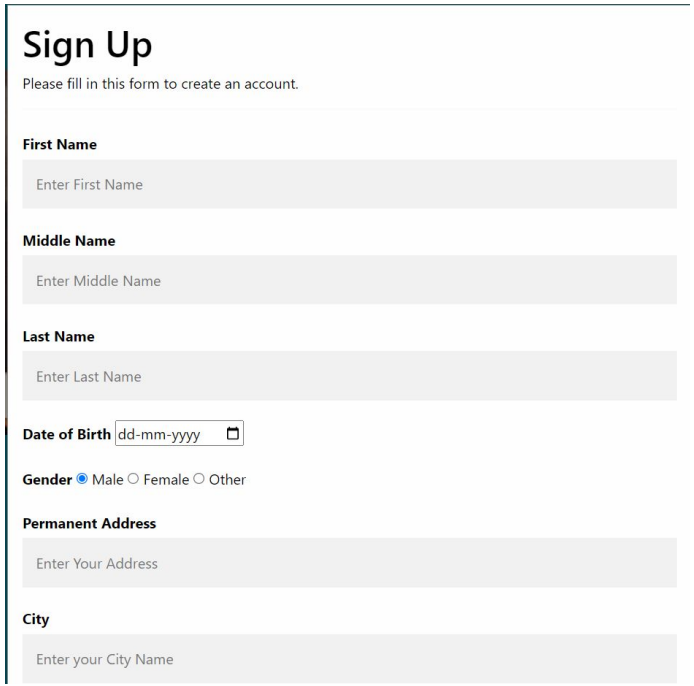
Individual contribution

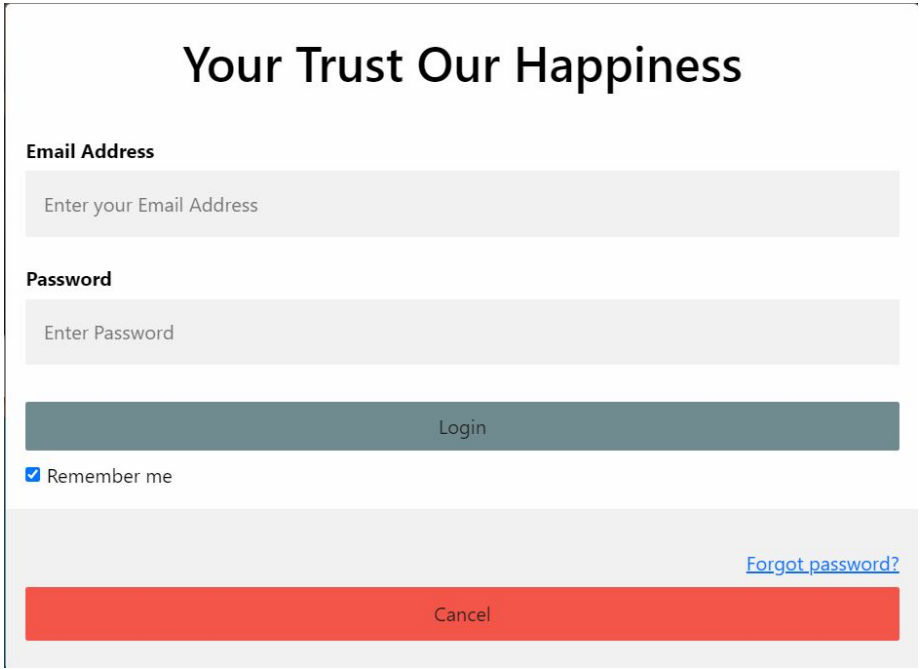
- Dhwanil Shah -
- Yash Shah
- Dev Sanghvi
- Darshil Shah
- Jeel Faldu
- Harsh Mehta

Application navigation using UI/UX

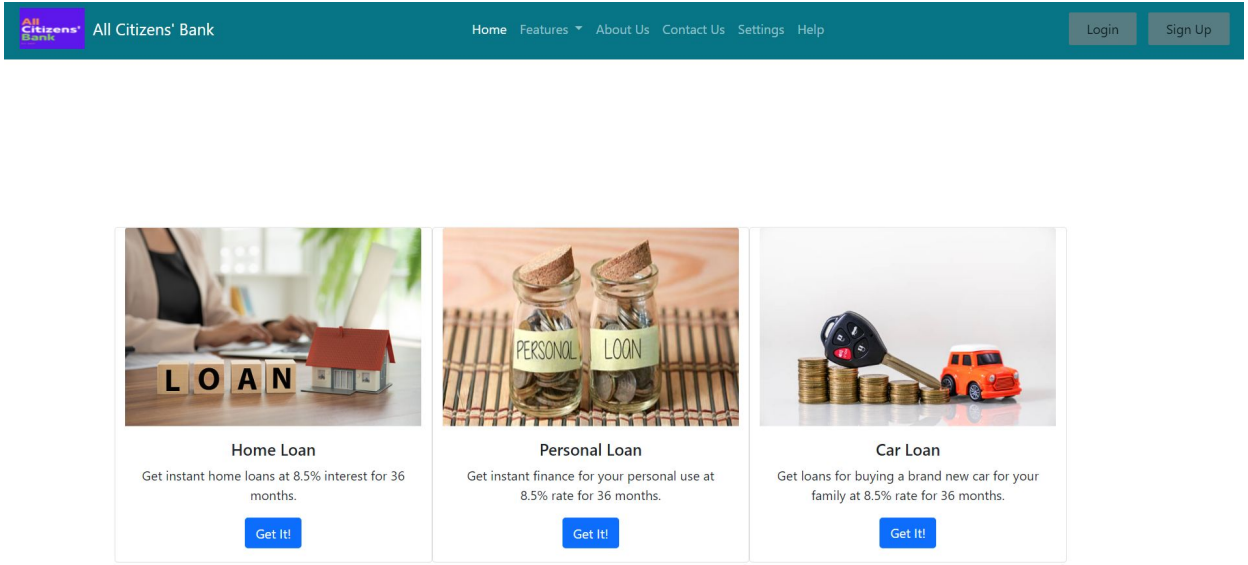


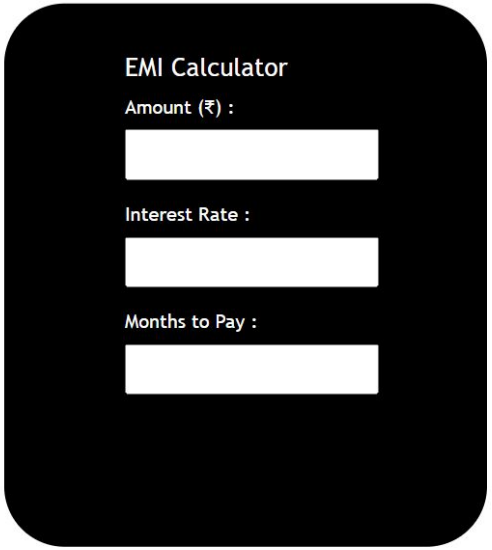
Application navigation (Contd.)



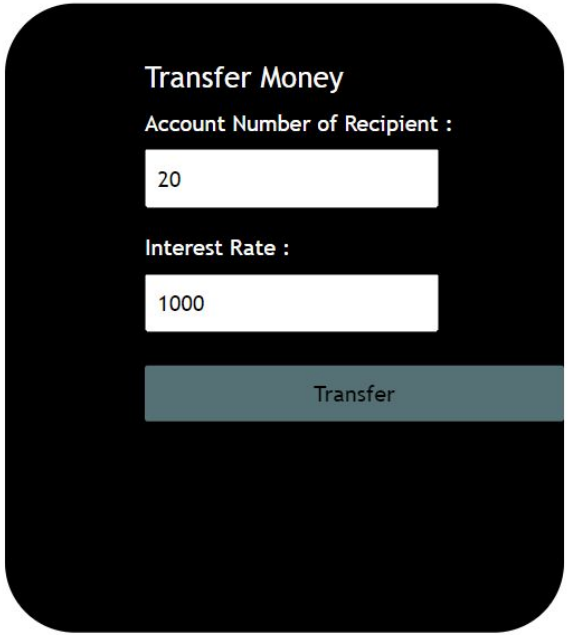


Application navigation (Contd.)





Application navigation (Contd.)



Summary of test plan

Test cases should be designed in such a way that it should cover all the possibilities including the boundaries.

- User will only allow to login if it's account exist and email_id and password are provided correctly.
- After Sign Up operation a new entry is created in the database and so after this if the user login, login operation will be successfully implemented.
- After performing the operation of transferring funds, balance of both recipients gets updated.
- Transfer money operation will only work if sending as well as receiving parties have their account.
- Amount of money transferred should always be less than/equal to balance of the user.

Summary of test plan (Contd.)

- There are different kinds of loan provided and user can apply for a specific loan according to its requirement.
- EMI calculator is working correctly and efficiently and hence all EMI related details is available to the user.
- Query Panel (Customer Care) is also available on our website so that it helps the user to encounters any problem while performing any transaction.
- Recharge and bill payments functionalities will be available in the **future**.
- In order to achieve Consistency and Fault tolerance we will also try to convert our SQL database to NoSQL database in **future**.

Thank You