

DotNet_Angular_AE_Business_Loan_B2_T5

Team Members:

- 1.Visali Mohan
- 2.Divya K
- 3.Deeba Harshini S
- 4.Mohammed Ashik S
- 5.Kaviya S
- 6.Muralikrishnan M
- 7.SARANYA S
- 8.Muthamilselvan K
- 9.Hariharan K
- 10.Kusuma Ramesh
11. NAVEEN M

Mentor : Bhuvaneswari Haritheertham

Overview :

Developing a web application to assist customers in submitting their business loan applications through an online.

Users:

- 1.Customer
- 2.Admin

Technology Stack:

Front End - Angular 10+ / React 16+

Material Design

Bootstrap/Bulma

Server Side - Spring Boot/.Net Web API/ Node Js

Database - MySQL or Oracle or MSSQL

Functional Requirements:

Customer:

- Build an application that customers can apply for business loans online.
- The customers should fill their profile.
- The customers should have an option to upload the document proofs such as Aadhaar card, pan card, pay slips, bank statements etc.
- Once the loan application is filled. The customers can submit for further approval process.
- Customer can track their loan status anytime.

Admin:

- The admin can see the list of applied loans.
- The admin can view the list of documents submitted and either approve/disapprove the loan application.
- Once the loan is sanctioned, the repayment schedule should get generated automatically.
- The admin should have provision to generate reports.

Add – On Features:

- The availability of filtering, searching, and sorting based on multiple criteria should be ensured
- The availability of email integration is necessary to notify customers.
- Multi-factor authentication for the sign-in process
- Payment Gateway

Validations:

Registration:

- Validate user input that requires a specific format such as email, phone numbers, or creditcard
- Mobile Number: Must contain 10 digits.
- Password: Must contain 8 or more characters, one or more capital letters, small letters, numbers, special characters.
- Confirm Password: Must be same as the password.

Login:

- Mail: Must be a registered mail address.
- Password: Must match with the password which was stored.
- Display informative error messages when a validation error occurs.
- Handle exceptions that occur during the execution

Customer:

- Registration: The customer needs to enter the necessary fields like type of user, username, mobile number, email, and password.
- Login: The customer uses his/her registered mail address and password to login into the portal.
- Apply Loan: The customer applies for a loan by providing the necessary details and uploading the mandatory documents.

- Loan Status: The customers can view the status of their applied loans through the unique loan ids they're provided with.
- Profile: Customers can view their personal information.

Admin:

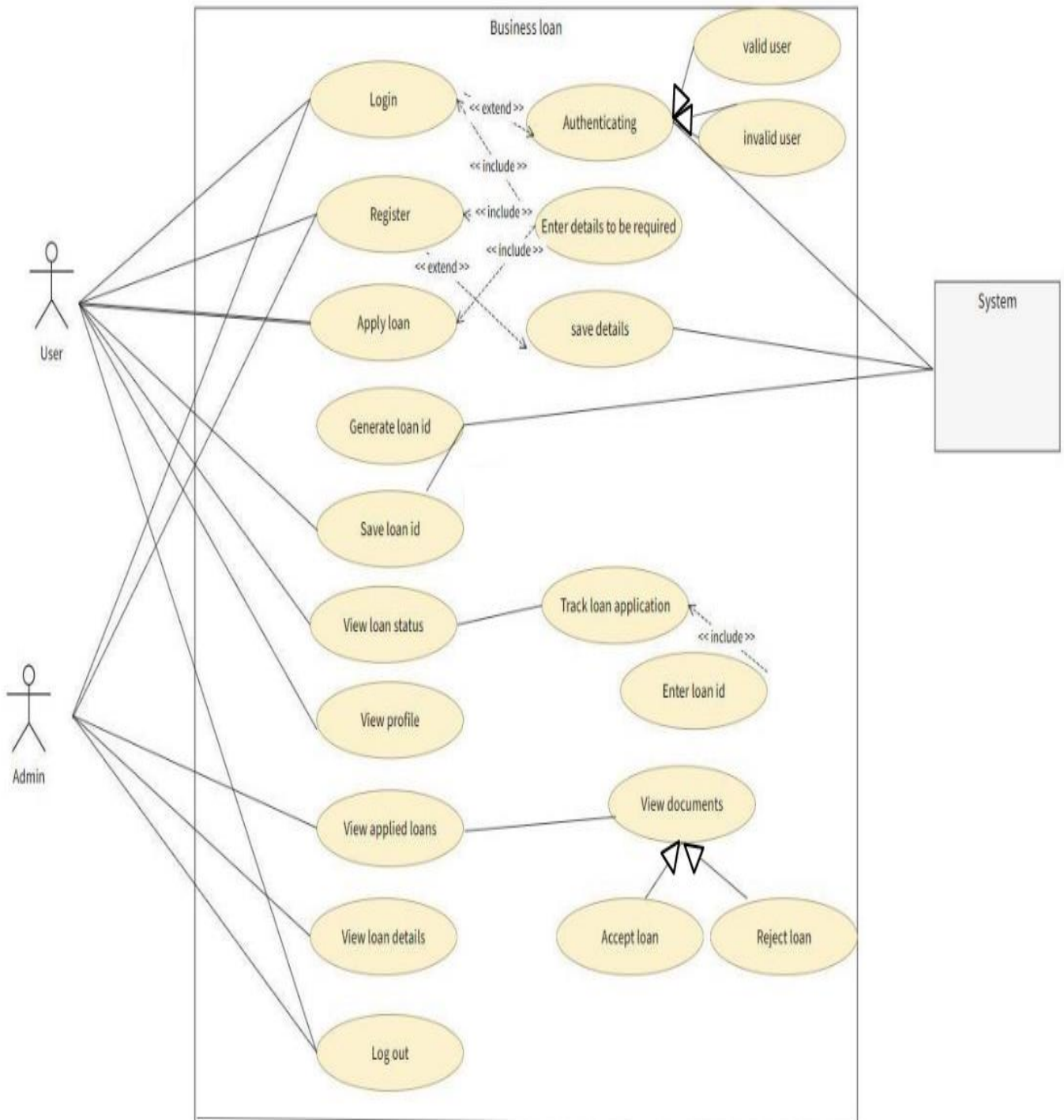
- Applied Loans: The admin can view all the applied loans under this section.

Admin can

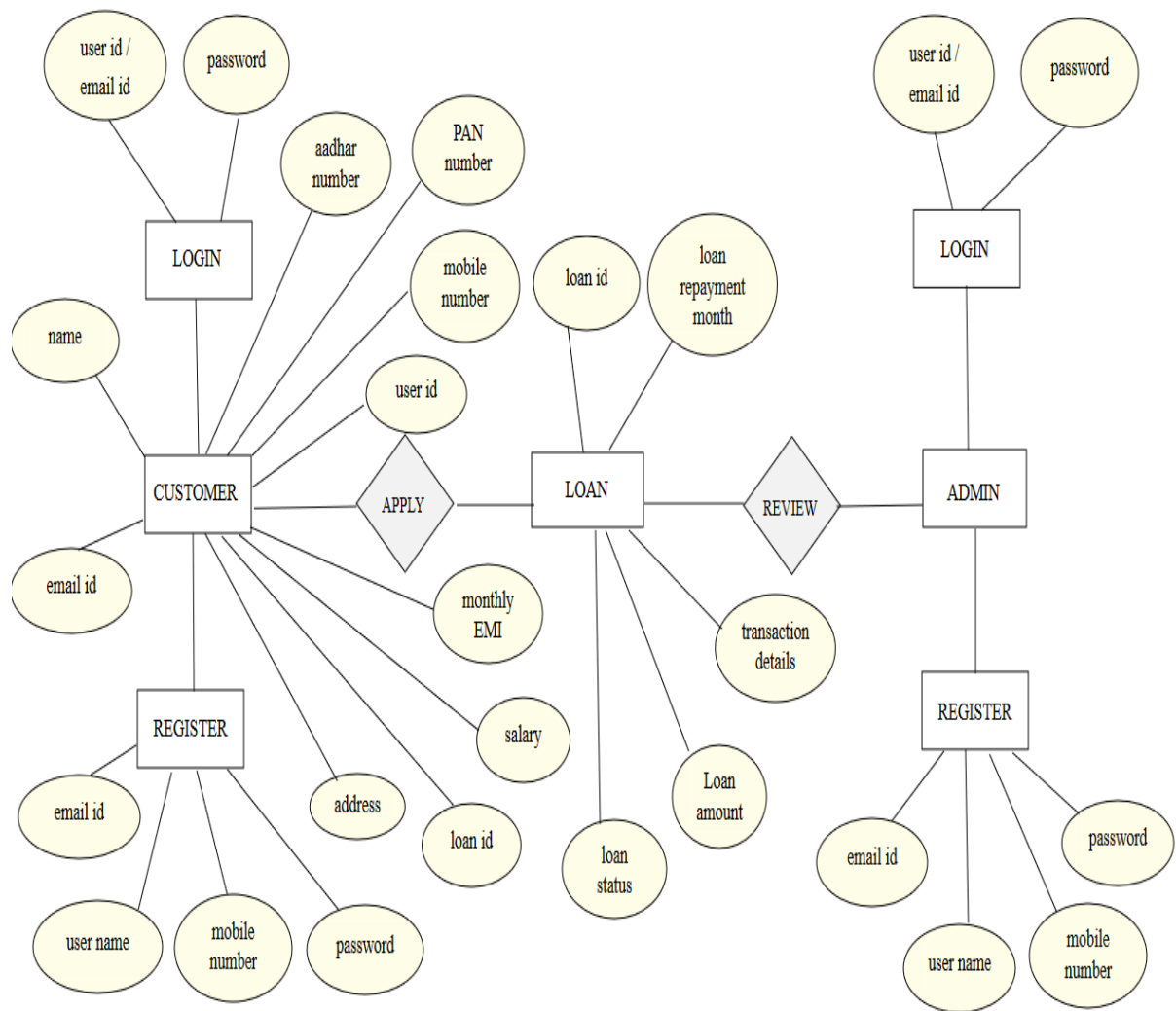
approve or disapprove the loans based on the bank constraints.

- Loan Status: The status of each loan can be checked under this section using just the loan id
- Approve Loan - The admin can verify the loan application
- Verify Documents - The admin can verify the documents before approving the loan
- Edit Loan - Edit the loans
- delete Loan - delete the loans v.
- Generate Schedule- Generate repayment schedule based on the number of months
- Edit Schedule- Edit the repayment schedule
- Delete Schedule - delete the repayment schedule.

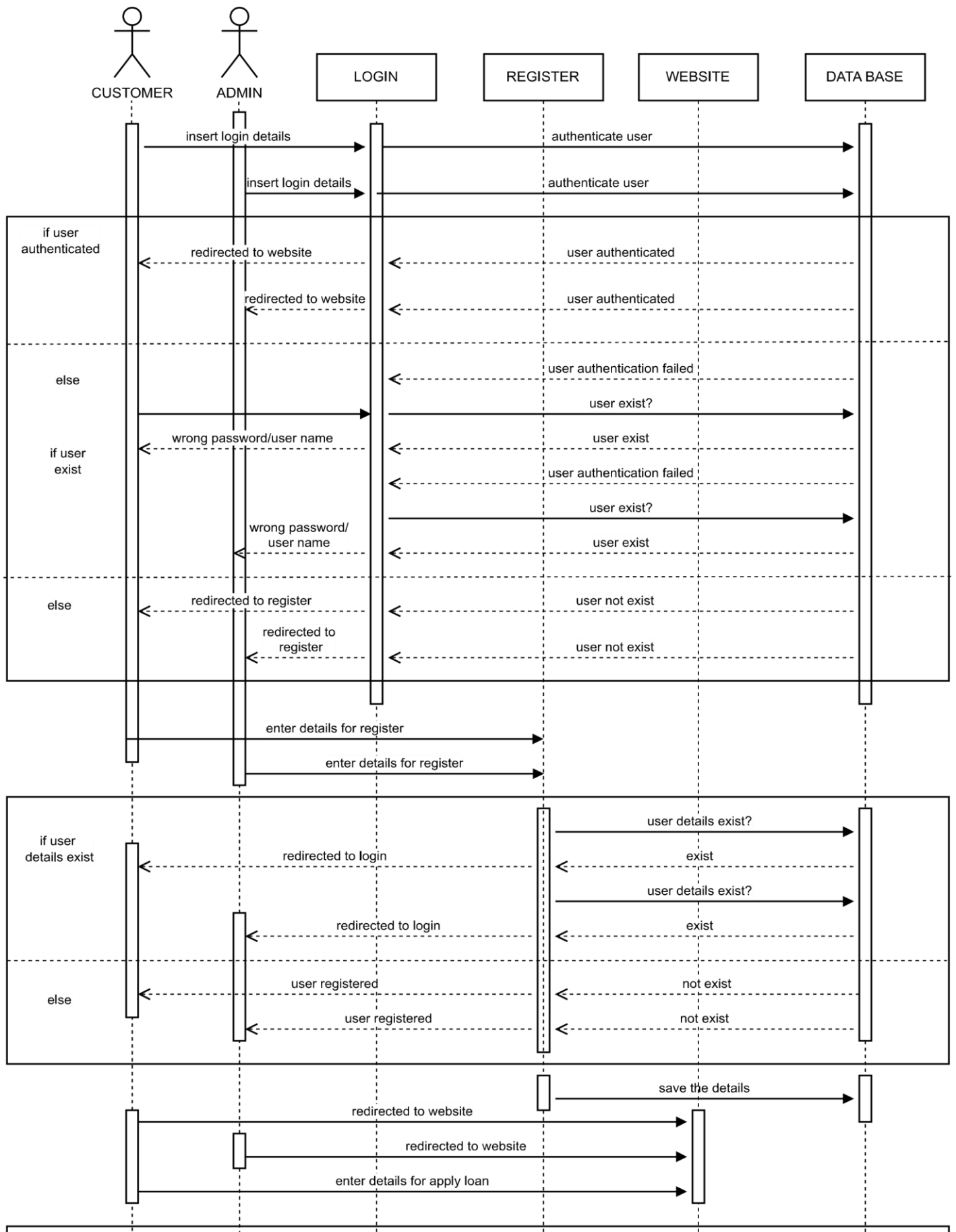
UML – Use Case Diagram:

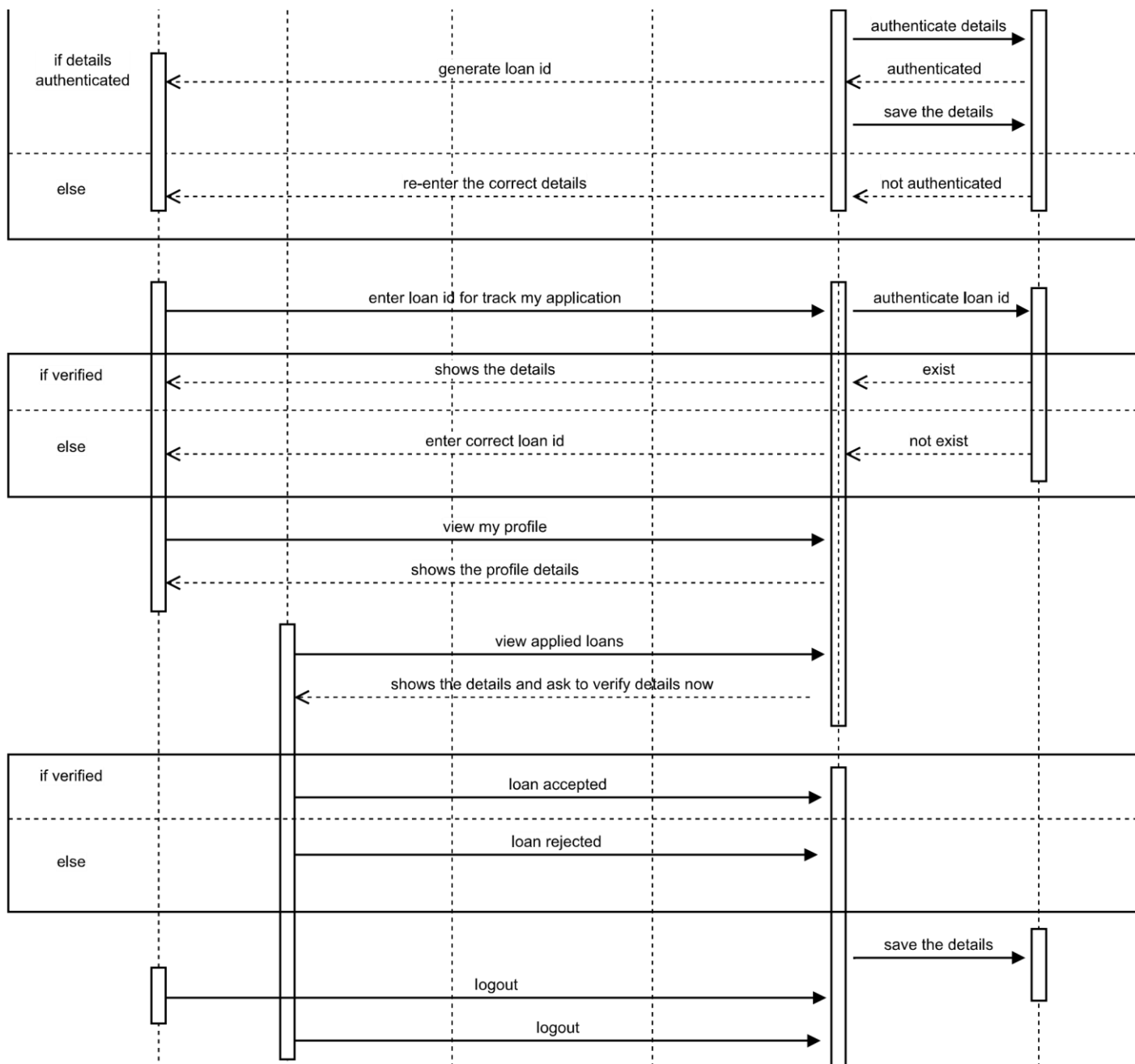


ER- DIAGRAM:



SEQUENCE DIAGRAM:





Back end:

Methods:

Authentication Controller:

- isUserPresent
- isAdminPresent
- saveUser
- saveAdmin

User Controller:

- addUser
- getUser
- editUser
- deleteUser

Admin Controller:

- approvedLoan
- verifyDocuments
- generateSchedule

Server Side:

Model Layer:

- UserModel
- LoginModel
- AdminModel
- LoanApplicantModel
- LoanApplicantModel
- DocumentModel

Controller Layer:

- AuthController
- UserController
- loanController
- AdminController