Net Banking Application

*CDAC*

*Submitted To: - Submitted By:-*

Mr. Vasu Rupendra Kaur (200250120081)

C-DAC Bangalore Rekha Dodani (200250120028)

Sneha Kumari (200250120098)

Mansi Singh (200250120048)

**Project Objective & Description**

1. The main objective of the project is to provide a portal for online banking services.
2. As part of this project we have tried to automate the registration process for new customers.
3. New customer can apply for registration and setup up their login credentials thus completing their full profile.
4. Existing customer can login into their account.
5. Upon unsuccessful login, user we see an error message and they can re try.
6. Once Login succeeds, user sees the balance on the screen and other various services offered.
7. Users can see their last six successfully carried out recent transactions.
8. Various Services are as below

* User can add Payee
* User can transfer amount to registered payees
* User can update their personal details

1. While transferring users cannot transfer amount greater than their existing balance.

**Project Flow**

The Project is divided into two parts which consists of Angular as frontend and Spring Boot as backend with **MySQL** as database.

Use Case – Login

**Angular**

1. User is presented with the login.html **view** which prompts for username and password.
2. After User enters the credentials and hits submit the form validates for values not empty and calls the typescript **Controller**.
3. The controller makes an ajax request to service with the user credentials and subscribes on the result.
4. The **Service** makes an HTTP post request in this case to the RestAPI exposed and on return, returns it back to controller.
5. If the call succeeded and user entered correct credentials, user is redirected to accounts summary screen.
6. If the credentials entered were wrong, an error message is displayed, and user is kept on the same screen and prompted to reenter again.
7. If there was some application error, user is again kept on the same screen with a generic error.

**Spring Boot**

1. The entry point is **LoginController** that receives the username and password as a body in the post request.
2. The request body is passed to **LoginService** that further calls the **LoginRepository**.
3. **LoginRepository** fires a select call to see if there is an entry with the username and password passed in the request.
4. Internally hibernate does the entity mapping to the corresponding table as to from where the information needs to be picked up.
5. The response from repository is analyzed by service if there is one such entry, the response status is treated as successful, else it is treated as failed.
6. The same response is passed back to controller and controller passes it to back to the angular frontend application.

**Challenges**

1. While implementing the transfer funds there were lot of scenarios to consider making the transaction.
2. As to the information must be reflected at various places once the transaction is successfully carried out, like it must be reflected in the total balance and a transaction entry must be there for the sender and receiver account.
3. It is a single page application and user gets to see the last recent successful transactions carried out. The descending sorting had to be up to millisecond accurate to correctly sort the transactions based on complete date time.
4. The horizontal layout for recent transaction panel with responsive widths was achieved through fx-layout.

**Learnings**

1. Became more aware that the design importance is must before starting the implementation.
2. Learnt the complete flow as to from where and how the data travels between layers.
3. How the data gets mapped between different domain applications.
4. Learnt a lot on the date sorting because of the recent transaction feature.
5. Learnt to work as a team across remote places.
6. Learnt angular routing.
7. Learnt cors origin concept.
8. Learnt the various rest operations required and which operations suits which flow.
9. Learnt the application end to end testing.