Currency Exchange Portal

Duration	:	5 Weeks
Wizard	:	Anusha Yajaman Suresh
		Mayuresh Hemant Devanpalli
		Balabaskaran Gounassegarane
Source	:	Citi - QFX

Main Heading

Problem Statement:

Clients have a requirement to their customers want to perform currency conversion feature from their account page. This portal acts as the landing page for the customer who is willing to convert their currency from their accounts. Super User will able to manage the customer profile and their accounts.

Objective:

To build a portal to perform currency conversion which can be utilized as the landing page for the customers.

Technology Stack:

Programming Language: Java, HTML, CSS, JavaScript

Framework: Spring Boot, Angular

Libraries: RESTful, JPA, Junit, Mockito, Karma, Jasmine

Activity:

Stage 1: Requirement Gathering

Associates will be interacting with corresponding mentors to understand the requirements with business impact and come up with a rough estimate plan on the deliverables with timeline.

Stage 2: Data Cleaning and Processing

As a part of this Grooming process,

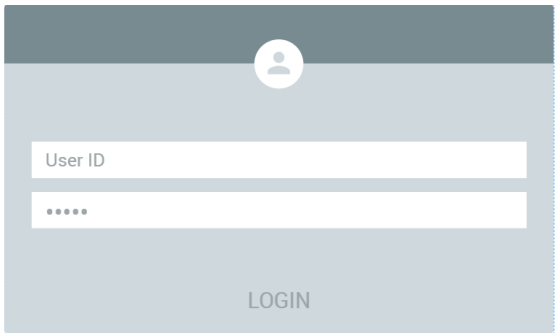
- Associates will interact with the wizards to understand the functionality, biz logic, design.
- Come-up with the design plan and get the sign off
- Various test cases to be created and it should cover E2E functionality of the application.

As part of pre-requisites,

- Create a common account for the application or individual account, repository in the github.
- Install VS code, Intellij IDEs
- Preferred DB's IDE and install the DB server.
- Install Postman
- Design table structures and create tables.
- Install SonarQube plugin and configure the rules

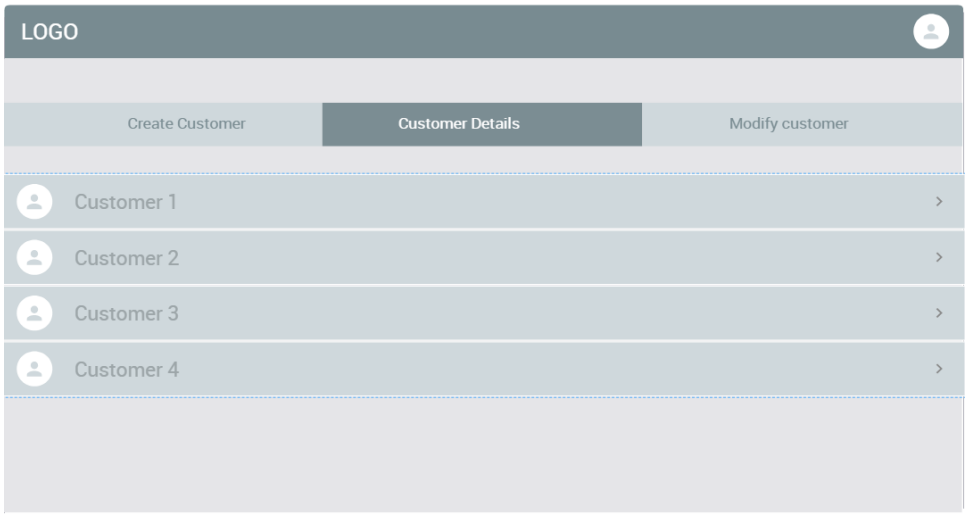
Stage 3: Front-end Design

1.Login Screen



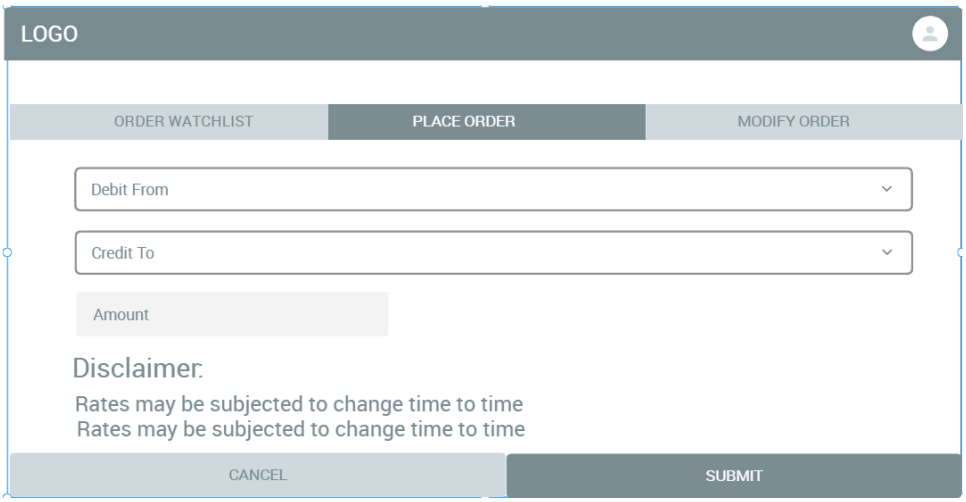
2.Admin page

- Create customer page
- List of customer page
- Modify account page

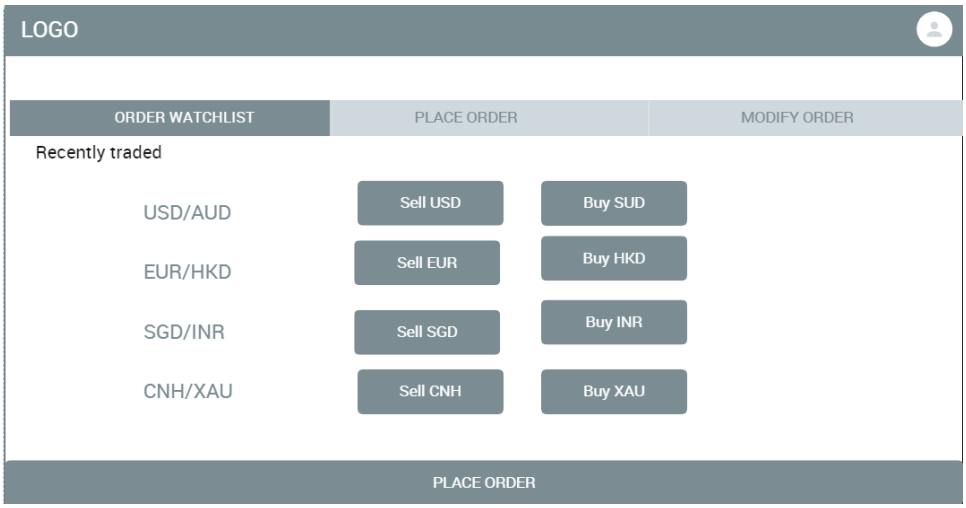


3.Customer page

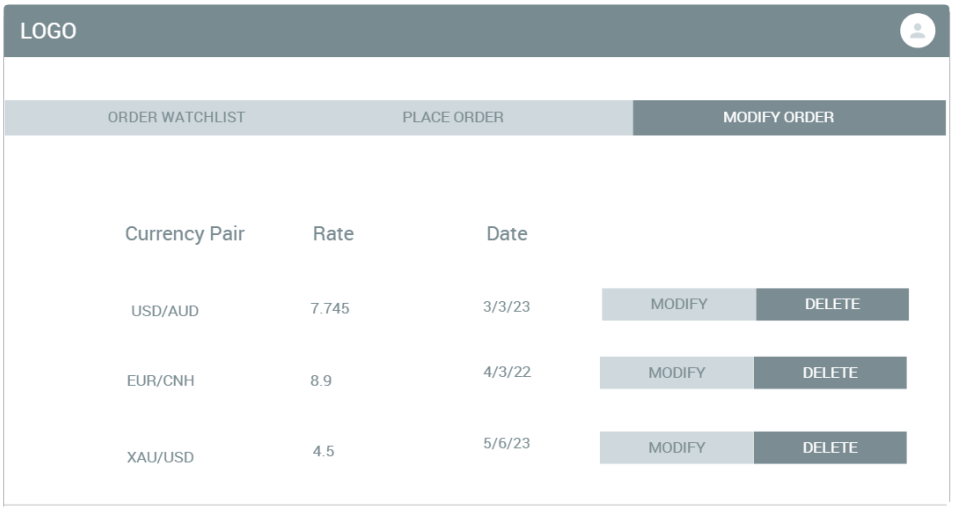
- List of accounts
- Place Order page
- Order Watchlist page



4.Order list



5.Modify order



Stage 3: APIs Design

- customer creation (personal details, customer no., account numbers)
- account creation (account no., currency type)
- list of users (includes all account details)
- modify account
- delete account
- list of accounts (for individual customer)
- order place (by customer)
- list all orders (placed by customer)
- get currency pairs (supported currency conversion)

Stage 4: Authentication and Authorization

- Implement user authentication mechanisms to secure user accounts and protect sensitive information.
- Set up user creation and login functionality, including password hashing and session management.
- Implement authorization to restrict access to certain features or actions based on user roles and permissions.

Stage 5: Data Storage

- Establish a connection to the chosen database system (e.g., MySQL, PostgreSQL, MongoDB).
- Design the database schema and create tables or collections to store travel entries, user information, and other relevant data.
- Implement database operations for CRUD (Create, Read, Update, Delete) functionalities.

Stage 6: Testing and Deployment

- Unit Testing: Develop and execute unit tests to ensure individual components of your application work correctly.
- Integration Testing: Perform integration testing to verify the interaction between different components of your application.
- Deployment: Deploy your application to a dev environment, following best practices for scalability, performance, and security.

Stage 7: Continuous Integration/Continuous Deployment (CI/CD)

- Set up a CI/CD pipeline to automate the process of building, testing, and deploying your application whenever changes are made to the codebase.