

Complaint Redressal System

This document contains the following

- Project and developer details
- Sprint planning and tasks completion
- Core concepts used in the project
- Links to the GitHub repository
- General info
- Using the application
- Technologies Used
- Demonstration of product capabilities, appearance, and user interactions
- Unique Selling Points of the application

Description

ABC Telecom Ltd. is one of India's major telecommunication service providers offering landline, mobile phone and Fiber optic broadband services across the country. Their customer services group is interested in providing a set of customer redressal services through the development of a new application using the state-of-art technologies such as Spring-boot for the development of java-based services, and UI using Angular and integrate them suitably, so that all the necessary services are taken care of through this application. They want to have an online complaint management system where the customers can raise complaints regarding their landlines and broadband services.

Scope:

The system should be a web-based application, developed using spring framework as a backend and angular as a front-end that performs the following functions:

1. Enables the Administrator to create and manages the lifecycle of different types of users
 - Customer
 - Manager
 - Engineer
2. Enables the customers to login to the portal to raise and track complaints related to the services availed by them
3. Enables the manager to login, view the complaints raised by the customers and assign the ticket to the engineers based on the PIN Code
4. Enables the engineers to pick up the tickets, work on them, enter the status of the task. They can also re-assign it to the Field Workers if they cannot resolve it from the data center.

Recommended Tools:

- Junit to perform Unit Testing
- Spring boot for web application
- MySQL Database for storing all the data and appropriate JDBC driver for connectivity

- Angular for Front End for all UIs.
- DevOps and production technologies: Git, GitHub, Docker

Requirements

1. Admin user is the super user of the system. The admin user creates other categories of user like customer, manager, and engineers. The admin user has the privilege to create, update and delete the records through the web interface and can access the entire system.
2. Customer users can perform tasks like creating a complaint, view complaints, tracking the complaint and providing feedback on the resolution of the complaint through the web UI. The feedback provision is an optional feature. Customers should provide the following minimal details such as name, address (with PIN Code), telephone / mobile number, type of problem (cannot make a call, but receive a call or can make calls, but cannot receive calls, or neither make nor receive calls). Once successfully submit the complaint, a ticket is raised and shared it with the customer. At that time, the status of the ticket will be 'RAISED'
3. Managers can pick the tickets from the active ticket list and assign them to the engineers based on the PIN code of the customer (where the service is installed). There should be different managers to take care of different PIN codes. You are expected to use at least about 5 – 10 different PIN codes, and there should be as many managers for assignment. Managers should also be able to see the status of all the tickets in all the areas.
4. Engineers can log in and view all the tickets assigned to him/her and can pick the tickets assigned to them and assign it the status of 'WIP', (work on the case, which is dependent on the type of problem and resolve) and update the status as 'RESOLVED'. In case they are not able to resolve the issue at their end, they can remark that this needs to be reassigned to Field Workers in case of a cable fault or at the customers' site, and flag it as 'ESCALATED'
5. All the activities of all the use cases should be appropriately bound by session or other alternatives. Appropriate time-out to be provided for each user.

UI Screens

Login Screen

The login screen will have the username and password fields. The usernames and passwords and roles are stored in the user table. The system should show those screens that are allowed for each category of users.

Admin Activities through UI

There has to be only one admin, and he/she can login/logout. Once logged-in, he/she should be able to maintain the lifecycles of Customer, Manager and Engineer

Manager Activities through UI

Managers should be able to login and logout. Once logged in he/she should be able to do the following, at a minimum:

- View all the tickets and status
- Assign Complaints to different Engineers
- Should be able to view of Customer feedback

Engineers Activities through UI

Engineers should be able to login and logout. Once logged in he/she should be able to do the following, at a minimum:

- View the complaints,
- View complaints based on individual customer
- Work on complaints (Off line activities) and assign the new status
- Mark the ticket status appropriately
- View the Customer feedback

Customer Activities through UI

A customer should be able to login/logout. Once logged in, he/she can view the status of the tickets raised by him/her. The customer also should be able to provide a feedback on the status RESOLVED or ESCALATED. In case there is a problem, the customer can raise a ticket on the complaint, through say, Register Complaint. Once successfully submitted, the customer should get the ticket number as the acknowledgement.

Access Levels

Appropriate users of the use cases defined in the Requirements section should have appropriate access levels. For example, Admin screens can take care of the CRUD operations on Customer, Manager and Engineer Use cases. Each of them should be able to do appropriate activities as defined above, using their UIs.

I. Sprint 1: Admin Login

1. Admin will only have login page, registration of admin will be hardcoded at the backend with the permission of production head. JWT authentication will be used for login verification.
2. Admin successful login will route to admin home page where all the desired admin functionalities will be added.
3. Admin home will have options to:
 - i. Add or remove the active users.
 - ii. Check all the raised complaints and their active status.
4. Admin can view all the details by applying filter as desired.

II. Sprint 2: Customer Login/Registration

1. Customer will have login and registration options, already registered users can login through their username and password and new users can register and create their username and password for further use. JWT authentication will be used for login verification.

2. Details entered by all users will be saved in the database, and password will be stored in encrypted format to maintain security.
3. After successful user login users will be able to:
 - i. View all the previously raised complaints by them and its status.
 - ii. Register a new complaint.
4. A logout functionality to logout current user.

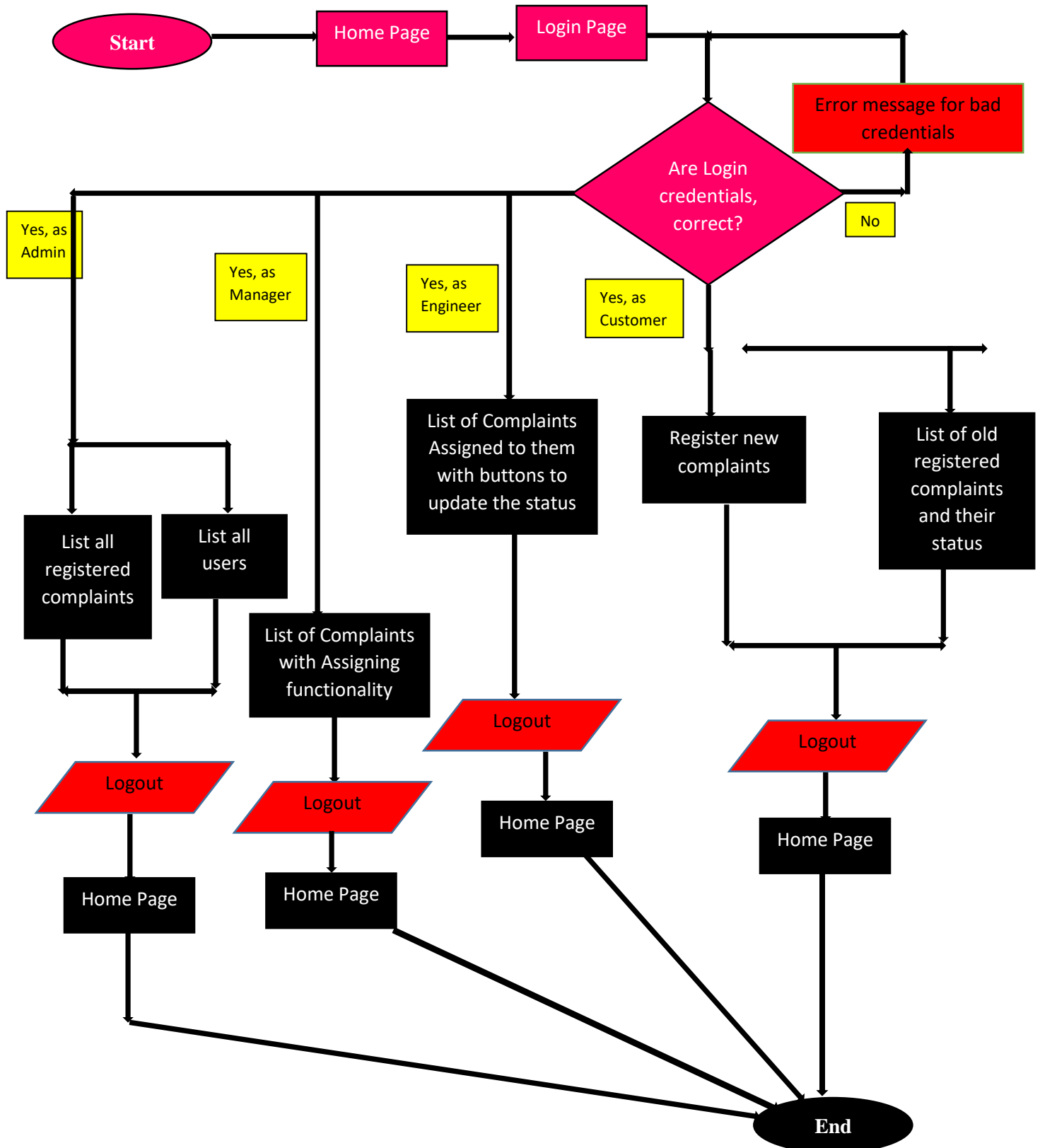
III. **Sprint 3: Manager Login**

1. Manager can login and view all the complaints and assign complaints to suitable engineer.
2. Name of the assignee will be visible along with complaints

IV. **Sprint 4: Engineer Login**

1. Engineer can view all the complaints specifically assigned to them.
2. With the click of a button they can update status of complaint as WIP(work in progress)/ In review/ Resolved.

Flow chat



Developer Details:

Mahendra Kumar Singh

Mahendrakumarsingh9893@gmail.com

Technologies Used

HTML5

CSS3

Bootstrap5

Angular

JSON

TypeScript

VsCode(Editor)

Links to the GitHub repository:

<https://github.com/Mahendra1272/ComplaintRedressalSystem.git>

Pushing the code to GitHub Repository:

- Open Git Bash and navigate to the folder where you have created your files.
cd ComplaintRedressalSystem \ ComplaintRedressalSystem
- Initialize the repository using the below command **git init**
- Add all the files to your git repository using the below command **git add .**
- Commit the changes using the below command **git commit -m "Initial commit"**
- Add the URL for the remote repository where your local repository will be pushed **git remote add origin https://github.com/Mahendra1272/ComplaintRedressalSystem .git**
- Push the files to the folder you initially created using below command **git push -u origin master**

