**Complaint Redressal System**

1. You can see the principles of interaction with the program in the screenshots
2. For implementing this project, we need 5 sprints. Sprint duration – 1 week
   1. Sprint 1-developing architecture of solution
   2. Sprint 2-developing front-end design
   3. Sprint 3-developing front-end business logic
   4. Sprint 4-developing back-end side
   5. Sprint 5-testing and refactoring
3. Core concepts used in project:
   1. Spring Boot
   2. Spring Security
   3. Spring Web
   4. Spring Data JPA
   5. Lombok
   6. Angular
   7. Google material design
   8. JSON Web Tokens (JWT)
4. Features:

* 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.
* 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’
* 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.
* 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”
* 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.

**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 be able to login and logout. Once logged in he/she 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 be able to login and logout. Once logged in he/she 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 be able to login/logout. Once logged in, he/she can view the status of the tickets raised by him/her. The customer also 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 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.