HP QC dashboard Implementation

Overview:

The solution for the given problem is implemented by following technologies.

1. **Spring Security** – Provide authentication and role based authorization.
2. **Spring MVC –** Providing service layer and web configuration.
3. **Spring Boot/REST –** Micro services Implementation.
4. **Hibernate –** Provide DAO Layer
5. **Cassandra –** Big data repository to store login session details.
6. **Oracle/MySQL –** Providing data model and repository for the app data.
7. **Angular JS/ Bootstrap -** Providing UI and responsive charts.
8. **Logging- Log** back to log application logs.

Browsers

AngularJS/ JSP

Access Denied

Spring Security

Invalid

Authenticated

**Spring MVC**

spring

*Cassandra Big data*

Cassandra

View Controller

Spring Micro Services

Hibernate

*Mysql/Oracle*

Technical Design:

The following roles are used in this application.

1. When the user requests the application URL, it will reach **webserver**.
2. The request will flow Webserver to **spring security**.
3. Authenticate if valid user and request will flow to **Spring MVC.**
4. If the user is invalid, Access will be denied for that request.
5. One the user authenticated, **spring security** will authorize the users by validating roles.
6. Once authorization completed, Application will connect with **Cassandra** Big data repository.
7. The application will store logged in user unstructured data of into **Cassandra repository**.
8. The request will flow into **Spring MVC/ REST services** .
9. Then the request flow, **Micro services** to **Hibernate** to fetch the details.
10. Based the authorization, the user navigated to the dashboard and details will be displayed.

Functional Design:

Following roles are available in the implementation.

1. Project Manager – Project Dashboard / Chart view and create new agile project.
2. Add /Modify Project
3. Project Dashboard View
4. Quality Analyst - Issue management.
5. Add/Modify Issue
6. Issue Dashboard view
7. Business Analyst- Test case management.
8. Add/Modify Test case
9. Test case Dashboard view