**Java Microservices with Spring Boot Training Project**

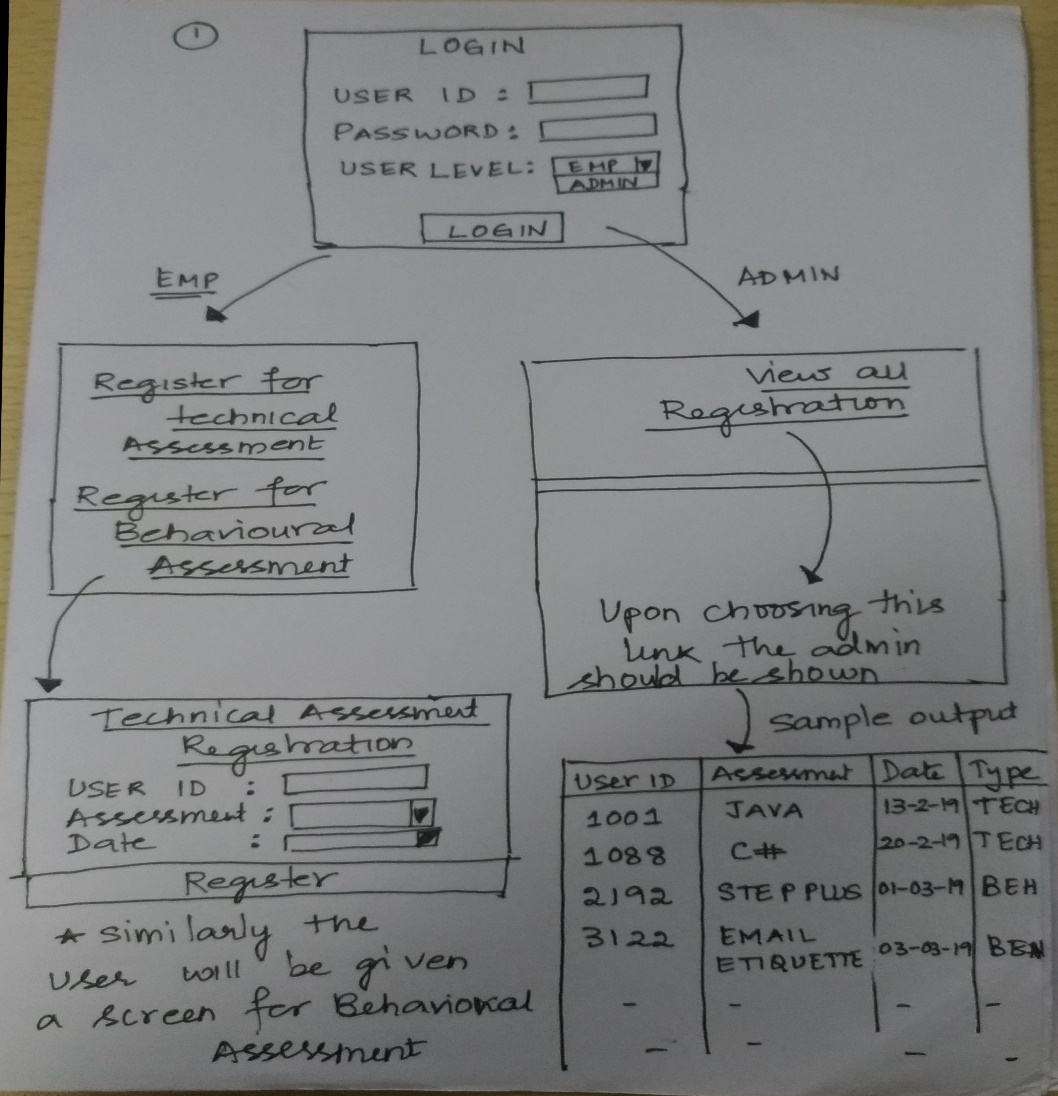
**Title:** Assessment Management

**Requirement**: Assessment Management project helps the employees and the assessment team administrator to manage few assessment related tasks.

Broad Functionalities:

1. Employees, once successfully logged in can register for either “Technical” or “Behavioral” Assessments.
2. Assessment Team Admin, once successfully logged in can view a tabular list of all the registrations done by the employees.

**Flow and Sample screens:**



Expectation is that this application will grow exponentially in time and may be required to be managed/scaled independently in future.

Hence, the requirement is to architect and implement it as Java based Microservices. You can use Spring Boot framework to implement the same.

Apart from implementing the Microservices based on the functionalities discussed earlier,

1. Please implement Eureka as your Registry and Discovery service so that microservices can find/discover each other and interact with each other if required.
2. Implement Client-side load balancer “Ribbon” wherever applicable.
3. Identify some important methods that can lead to cascading failures and guard them by implementing Hystrix – Software Circuit Breaker/Fault Tolerance mechanism.

**Assumptions:**

1. The users are already registered. So you can use a static list of registered Employees/Admins for this application.
2. The choice of how many number of Microservices you want to implement for the given scenario is left to you.

Hint: You can have 3 microservices:

1. Login Microservice
2. Employee Microservice
3. Admin Microservice

(Employee Microservice can be further implemented into different “Technical” or “Behavioral” Microservice. The choice is yours)

1. You can use a static list of “Technical” or “Behavioral” assessments for registration.

E.g.: Technical -> “Java”, “C#”, “PHP”, “PERL” etc.,

Behavioral -> “Step Plus”, “Email Etiquette”, “Learning Agility” etc.,

1. Implement appropriate validations to ensure that only upon providing correct userId and password and user type, the user will be logged on successfully and further allowed to proceed with the application. Otherwise appropriate Error messages are displayed.

**Other Details:**

1. You can use any IDE of your choice (STS/IntelliJ/Eclipse) etc.,
2. You can use any collections/database as your data store.