Software technical

Specifications and Features

CMD – Hackathon

Table of Contents

1 Introduction

CMD is a global telehealth platform enabling doctors and patients to connect with each other virtually and in-person.

2 Levels

2.1 Level 1: 60 mins

1. Feature to be implemented – Add a new Clinic
2. Create a Spring boot project and add all required dependencies in pom.xml
3. Follow a Layered Architecture – Entities, Dtos, mapper, service, service impl, repository, controller, exceptions
4. Check in the project in your branch
5. Follow the right naming conventions – Anything you name should be of high quality

2.2 Level 2: 75 mins

1. Clinic will have the following information
    a. Clinic ID – (Unique no : CL202200001)
    b. Clinic Name
    c. Business Name
    d. Street Address
    e. City
    f. State
    g. Country
    h. Zip Code (US Fomat)
    i. Latitude
    j. Longitude
    k. Date Created
    l. Services Offered (Consultation, Xray, Blood Test, Covid Test, MRI Scan)
        i. Service ID
        ii. Service Name
        iii. Service Code
        iv. Service Description
        v. Average Price
        vi. Is Active (Is it being provided now)
        vii. Above would be master data however each clinic would be configured with the services offered
2. Create the required models
3. Check in the code

2.3 Level 3: 60 mins

1. Create a rest api ClinicController which will have the method post to create a clinic
2. The REST API Standards have to be followed (We shouldn't have a method CreateClinic)
3. Add all the necessary dependencies for the controller

4. Follow the right standards and annotations
5. Create the DTOs needed for the functionality at the UI
6. Create your mappers for DTOs to models
7. Check in the code

2.4 Level 4: 75 mins

1. Implement the end-to-end flow
2. Business layer should have the logic of appointment date and time checked
3. Data should be stored in the database in PostgreSQL
4. High Quality Coding Standards to be followed
    a. Naming conventions
    b. Logic
    c. Right use of the Language
5. Unit Testable implementation
6. Generate Swagger. Take a screen shot and check in the same
7. Check in the Code

2.5 Level 5: 90 mins

1. Create a unit test project
2. Write test cases for the use case implemented
3. Should have sufficient coverage of the cases – positive and negative
4. Implement unit tests for the feature implemented
5. Execute the test cases
6. Check in Code

2.6 Level 6: 90 mins

1. Error logger must be implemented
2. Log4j or any other error logger can be implemented
3. Warning, Error with priority must be logged
4. Clients calling the Controller should handle the exceptions and use global exception and send the response accordingly.
5. Information that should be logged
    a. Message
    b. Priority – Critical, High, Medium, Low
    c. Message Type – Info, Warning, Error
    d. Message Date and Time
    e. Project
    f. Class
    g. Method

2.7 Level 7: 90 mins

1. Clinics are frequently searched in the application along with services offered
2. We should be storing the clinics and its services which is accessible for a city in a data structure enabling easy query
3. We should also cache the data as clinics are not changed every day and it's a rare update that happens.
4. Should be serialized and stored in cache memory
5. When a clinic Is added all its relevant information should be added to cache as well.