Telecom Service Management System Requirements Document

Purpose:

The purpose of this application is to enable users to perform telecom-related operations. This system should facilitate the following functionalities:

a. Service Management: Allow users to add, modify, and manage telecom services.

b. Customer Management: Enable users to manage customer information and subscriptions.

c. Issue Resolution: Provide tools for identifying and resolving customer issues efficiently.

Technology Stack: Springboot and MySQL or Cassandra, Microservice

1.1.1 Service API Controller:

This controller allows users to:

1. **View all available telecom services.**
2. **Retrieve details of a service by its name.**
3. **Save new service information.**

1.1.2 Customer API Controller:

This controller enables users to:

1. **Access customer information and subscriptions.**
2. **Search for customers by name or subscription.**
3. **Save customer information.**

1.1.3 Issue API Controller:

This controller facilitates users to:

1. **Find all reported issues.**
2. **Search for issues related to a specific service or customer.**
3. **Save new issue details.**

Software Requirements: Springboot, RESTful Web Services, MySQL or Cassandra, Postman.

Scope of the project: Develop a server-side application.

Assumptions, Dependencies, Risks / Constraints:

1. While adding info, if id is already existing, it should throw a custom exception.
2. For all rest endpoints receiving @RequestBody, validation check must be done and must throw custom exception if data is invalid
3. Avoid modifying test resources, as they are used for auto-evaluation.
4. Implement all business validations on DTO objects.
5. Implement all database operations on model/entity objects.

6. Perform model to DTO conversion in the service layer.

URL’s Exposed in the controller

ServiceController

Url Exposed	Purpose
/service/	View all available telecom services
/service/name	Retrieve details of a service by name
/service/save	Save new service information

CustomerController

Url Exposed	Purpose
/customer/	View customer information and subscriptions
/customer/name	Search for customers by name or subscription
/customer/save	Save customer information

IssueController

Url Exposed	Purpose
/issue/	Find all reported issues
/issue/service/{serviceName}	Search for issues related to a specific service
/issue/customer/{customerName}	Search for issues related to a specific customer
/issue/save	Save new issue details

Business Validations:

1. Name, address, and contact information must not be null, with a minimum of 5 characters and a maximum of 30 characters.
2. Service details, customer subscriptions, and all other fields must not be null, with a minimum of 5 characters and a maximum of 30 characters.

Resources Available in the platform

Name	Resource	Remarks
Package Structure		
Controller	ServiceController, CustomerController, IssueController,	Controller classes for handling all requests
model	Service, Customer, Issue	Entity/Document class
dto	ServiceDTO, CustomerDTO, IssueDTO	An DTO classes for validations and for data transfer
exception	InsertionFailed, ResourceNotFound	Custom Exception class
repository	ServiceRepository, CustomerRepository, IssueRepository	Repository interface
service	ServiceService, CustomerService, IssueService	Service Layer

Note:

1. **Implement only service methods and controller classes.**
2. **Do not add any method or change REST endpoint patterns.**
3. **Utilize Netflix Eureka for service discovery.**