**✅ CRBT Problem Statement – Spring Core Version**

**📘 Problem Context:**

In a telecom system, **Caller Ring Back Tone (CRBT)** is a value-added service (VAS) that allows users to set a specific audio tone (song or message) that callers hear when they call them, instead of the standard ringing sound.

You are required to implement a basic CRBT service using **Spring Core** features like:

* XML-based or Annotation-based configuration
* Dependency Injection (constructor/setter)
* Service Layer abstraction
* Simple in-memory data access using Java collections and Java 8 Streams

**🎯 Functional Requirements:**

1. **Register Customer**
   * A user registers with their customerId, name, and mobileNumber.
2. **Browse Available Tones**
   * The system allows viewing the list of available tones.
3. **Subscribe to a CRBT Tone**
   * A user selects a tone from the list and subscribes.
4. **Change Tone**
   * A user can change their existing tone.
5. **Deactivate Tone**
   * A user can remove their CRBT subscription.
6. **List Subscribed Customers**
   * View the list of customers currently subscribed to CRBT.

**🧱 Entities / POJOs:**

1. Customer

java

CopyEdit

String customerId;

String name;

String mobileNumber;

Optional<CRBTTone> subscribedTone;

1. CRBTTone

java

CopyEdit

String toneId;

String title;

String artist;

Duration duration;

**🧩 Interfaces:**

* CustomerService (interface) with methods:
  + registerCustomer(Customer customer)
  + subscribeTone(String customerId, String toneId)
  + changeTone(String customerId, String toneId)
  + deactivateTone(String customerId)
  + List<Customer> getAllSubscribedCustomers()
* ToneService (interface) with methods:
  + List<CRBTTone> getAvailableTones()

**🧰 Spring Components:**

* CustomerServiceImpl — Business logic
* ToneServiceImpl — Provides available tones
* CustomerRepository — Simulates data store using Map<String, Customer>
* ToneRepository — Predefined list of tones

**💡 Tech Used:**

* Spring Core (Annotation-based or XML-based DI)
* Java 8 features:
  + Optional
  + Streams
* Collections (Map, List)
* JUnit 5 for unit testing (optional extension)

**🔁 Example Flow:**

bash

CopyEdit

1. Register Customer: Avinash (ID: C001)

2. Browse Tones:

T1 - "Shape of You"

T2 - "Channa Mereya"

T3 - "Naatu Naatu"

3. Subscribe to T2

4. Change Tone to T3

5. View Subscribed Customers -> Avinash - "Naatu Naatu"

6. Deactivate Tone