**📌 Problem Statement: Caller Ring Back Tone (CRBT) Management System**

A telecom company wants to implement a **Caller Ring Back Tone (CRBT)** service to allow its customers to personalize the tone callers hear while waiting for the call to be answered. The company needs a backend system to manage CRBT subscriptions, activations, deactivations, and billing.

**🎯 Functional Requirements:**

1. **Customer Management:**
   * Register a customer (ID, name, mobile number).
   * Fetch customer's current CRBT status (active/inactive, current tone).
2. **CRBT Tones Catalog:**
   * Maintain a collection of available CRBT tones.
   * Each tone has an ID, title, artist, price.
3. **CRBT Subscription:**
   * Allow a customer to subscribe to a tone.
   * If already subscribed, throw an exception.
   * Automatically generate a billing record.
   * Store transaction with ACID compliance.
4. **CRBT Deactivation:**
   * Allow customer to deactivate CRBT service.
   * No refund issued; mark subscription as inactive.
5. **Billing:**
   * On tone subscription, create an invoice with tone price and date.
   * All operations must be transactional and rollback-safe (ACID).
6. **Search & Filters:**
   * Use Java 8 Streams to:
     + Search tones by title/artist.
     + Get all active CRBT users.
     + List customers with bills > ₹100 in last month.

**⚙️ Technical Requirements:**

**✅ OOP Design**

* Use classes: Customer, CRBTTone, CRBTService, Invoice, BillingManager.

**✅ Exception Handling**

* Create custom exceptions: AlreadySubscribedException, InvalidToneException, CustomerNotFoundException.

**✅ Java 8 Features**

* Use **Streams**, **Optional**, **Lambda expressions**, **Collectors**.

**✅ Collections**

* Use List, Map to manage tone catalog and customer subscriptions.

**✅ JDBC + ACID + MySQL**

* Connect to MySQL using JDBC.
* Use **manual transaction management** (commit/rollback).
* Ensure:
  + **Atomicity**: All operations complete or none.
  + **Consistency**: Data must be valid before and after transactions.
  + **Isolation**: One user's action doesn't affect another's view.
  + **Durability**: Once committed, data is persistent.

**✅ JUnit 5 Tests**

* Write test cases to:
  + Subscribe tone successfully.
  + Handle duplicate subscriptions.
  + Deactivate tone.
  + Validate invoice creation.

**🧾 Database Tables**

**customers**

sql

CopyEdit

CREATE TABLE customers (

customer\_id VARCHAR(20) PRIMARY KEY,

name VARCHAR(100),

mobile VARCHAR(15)

);

**crbt\_tones**

sql

CopyEdit

CREATE TABLE crbt\_tones (

tone\_id VARCHAR(20) PRIMARY KEY,

title VARCHAR(100),

artist VARCHAR(100),

price DOUBLE

);

**subscriptions**

sql

CopyEdit

CREATE TABLE subscriptions (

subscription\_id VARCHAR(36) PRIMARY KEY,

customer\_id VARCHAR(20),

tone\_id VARCHAR(20),

is\_active BOOLEAN,

subscribed\_on DATE,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id),

FOREIGN KEY (tone\_id) REFERENCES crbt\_tones(tone\_id)

);

**invoices**

sql

CopyEdit

CREATE TABLE invoices (

invoice\_id VARCHAR(36) PRIMARY KEY,

customer\_id VARCHAR(20),

amount DOUBLE,

billing\_date DATE,

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

**📘 Sample Use Case:**

1. Alice registers as a customer.
2. She browses the CRBT catalog using searchTonesByArtist("Arijit Singh").
3. She subscribes to tone ID TONE123.
4. System:
   * Adds record to subscriptions.
   * Inserts a new invoice.
   * Wraps both in a transaction.
5. Alice tries subscribing again → AlreadySubscribedException is thrown.