

Name: Johan George Eapen

Employee Id: 81693

# Project: Telecom DND (Do Not Disturb) Registry System

## 1. Introduction

The purpose of this document is to provide a detailed Low-Level Design (LLD) for the Telecom DND Registry System. The system simulates the enforcement of Do Not Disturb regulations by telecom operators, enabling customers to register and manage DND preferences that block promotional calls and SMS.

The application uses Python for business logic and MySQL for persistent data management, with a console-based user interface. The system supports real-time filtering and logging of promotional communication attempts to ensure compliance and auditability.

## 2. Module Overview

The project consists of the following modules:

### 2.1 Customer Management

- Manages customer registration including personal details and phone numbers.
- Supports CRUD operations on customer data stored in MySQL.

### 2.2 DND Subscription Management

- Allows customers to subscribe or unsubscribe from DND services.
- Supports blocking of all calls, all SMS, or selective categories (Banking, Insurance, Real Estate, Education, Entertainment, etc.).

### 2.3 Promotional Communication Simulation & Filtering

- Simulates telemarketer calls and SMS attempts specifying telemarketer ID, message type, and category.
- Checks each attempt against customer DND settings in real-time to permit or block communication.
- Logs blocked attempts for regulatory compliance.

## 2.4 Logging, Reporting & User Interface

- Maintains logs of all blocked promotional attempts including timestamps and metadata.
- Provides a console-based menu-driven interface for users and admins to manage system features and generate reports.

## 3. Architecture Overview

### 3.1 Architectural Style

- Backend: Python CLI application implementing core business logic
- Database: Relational Database using MySQL for persistent storage
- Interface: Console-based Command Line Interface (text menus)

### 3.2 Component Interaction

1. Customers interact with the system via console menus for registration and DND management.
2. Promotional attempts are simulated through system inputs.
3. System queries MySQL database to enforce DND rules and log blocked attempts.
4. Reports are generated from database logs and displayed on the console.

## 4. Module-Wise Design

### 4.1 Customer Management Module

#### 4.1.1 Features

- Add, update, view, and delete customer profiles
- Manage customer phone numbers

#### 4.1.2 Data Flow

- User inputs customer data through console menus
- Backend validates and stores data in customers table in MySQL
- Customer information retrieved for DND operations

### 4.1.3 Entities

- Customer
  - CustomerID (Primary Key)
  - Name
  - Phone Number

## 4.2 DND Subscription Management Module

### 4.2.1 Features

- Manage DND subscriptions: full blocking or category-based blocking
- Enable customers to customize blocking preferences

### 4.2.2 Data Flow

- Customers update DND preferences via menu-driven CLI
- Preferences stored in `dnd_preferences` table linked to customers
- On communication attempts, preferences are checked to allow or block messages

### 4.2.3 Entities

- DND Preference
  - CustomerID (Foreign Key)
  - Block All Calls (Boolean)
  - Block All SMS (Boolean)
  - Blocked Categories (List / JSON or table)

## 4.3 Promotional Communication Simulation & Filtering Module

### 4.3.1 Features

- Simulate telemarketer call/SMS with metadata (TelemarketerID, Category, Type)
- Real-time enforcement of DND rules
- Block or allow promotional communications accordingly

#### 4.3.2 Data Flow

- Telemarketer attempts entered via console input
- System queries DND registry, determines delivery or blocking
- Blocked attempts logged into database

#### 4.3.3 Entities

- Blocked Attempt
  - AttemptID (Primary Key)
  - CustomerID (Foreign Key)
  - TelemarketerID
  - Attempt Type (Call/SMS)
  - Category
  - Timestamp

### 4.4 Logging, Reporting & User Interface Module

#### 4.4.1 Features

- Logs all blocked attempts with relevant details
- Generates audit reports for customers and admins
- Provides menu-driven text interface for all interactions

#### 4.4.2 Data Flow

- Blocked attempts are inserted into the `blocked_attempts` table
- Reports queries retrieve and summarize data for display
- User selections trigger commands presented via CLI menus

#### 4.4.3 Entities

- Utilizes Blocked Attempt entity from previous module for reporting

### 5. User Interface Design

- Console-based CLI menus for:
  - Customer registration and profile management

- DND subscription management with selectable categories
- Simulating telemarketing call/SMS attempts
- Viewing logs and generating reports

## 6. Non-Functional Requirements

### 6.1 Performance

- Able to handle multiple customer records and simulate promotion attempts with minimal latency in a local environment.

### 6.2 Scalability

- Designed with modularity and database-backed storage to allow easy scaling.

### 6.3 Security

- Secures customer data via controlled access through CLI. (Authentication can be added in future.)

### 6.4 Usability

- Intuitive CLI-based menus allowing easy navigation for end users and admins.

## 7. Assumptions and Constraints

### 7.1 Assumptions

- Promo attempts and customer interactions are simulated via CLI.
- All users trust the integrity of input data.
- Network and telecom infrastructure are out of scope.

### 7.2 Constraints

- System operates in a local or test environment only; no cloud deployment.
- No live integration with telecom networks or SMS gateways currently.
- User access control and authentication are not implemented (future scope).