**Project Documentation: "Telegram Bot for Manicure Appointments"**

**1. General Description of the Project**

* **Project Name: Telegram Bot for Manicure Appointments**
* **Description: A Telegram bot that helps users schedule manicure appointments, view prices, reviews, portfolios of nail artists, and manage their bookings. The bot is designed for both clients and salon administrators.**
* **Project Goal: To automate the process of booking manicure services and simplify the interaction between clients and administrators through Telegram.**
* **Main Technologies:**
  + **Language: Java 17**
  + **Frameworks: Spring Boot, Spring Data JPA**
  + **Database: MySQL**
  + **Telegram API via the TelegramBots library**
  + **Docker for deployment**
  + **Liquibase for database migration management**

**2. Features**

**2.1 Features for All Users (Guests)**

1. **View Prices: Users can browse the list of available services and their prices.**
2. **Reviews: Users can read reviews from other clients about the nail artists.**
3. **Gallery of Works: Users can view photos of completed works and follow links to the artists' Instagram pages for more details.**
4. **Registration/Login: Users can create a new account or log into an existing one.**

**2.2 Features for Registered Clients**

1. **Choose an Artist: After logging in, clients can select the artist they wish to book an appointment with.**
2. **View Calendar: Clients can browse available dates for up to a month ahead for their chosen artist.**
3. **Choose Time Slot: After selecting a date, clients can view available time slots for their appointment.**
4. **Choose Service: Clients can choose from available services, with the price displayed next to each option.**
5. **Confirm Appointment: After selecting a service, clients confirm their appointment.**
6. **Reminders:**
   * **The bot sends a reminder to the client one day before the appointment.**
   * **The bot sends a reminder three hours before the appointment, with details of the time and service.**
7. **Notifications for Artists:**
   * **The artist automatically receives a notification about a new booking when it is confirmed by the client.**
   * **The artist also receives reminders one day and three hours before the appointment.**
8. **Cancel/Reschedule Appointment: Clients can cancel or reschedule their appointments.**
9. **View Appointment History: Clients can see both past and upcoming appointments.**
10. **Leave a Review: After completing the service, the client is prompted to leave a review for the artist.**

**2.3 Features for Administrators**

1. **Manage Availability: Administrators can manage available dates and time slots for bookings.**
2. **Manual Booking: Administrators can book an appointment for a client or cancel an existing booking.**
3. **Client Management: Administrators can block or delete user accounts.**
4. **Send Messages to Clients: Administrators can send messages to clients via the bot (e.g., reminders or important notifications).**
5. **Review Management: Administrators can view and manage client reviews.**

**2.4 Additional Features for Administrators**

1. **Artist Management:**
   * **Add Artist: Administrators can add new artists by specifying their name, description, and other details (such as contact information and social media links).**
   * **Delete Artist: Administrators can remove an artist from the database. All associated appointments and services will either be deleted or reassigned to another artist (as chosen by the administrator).**
   * **View Artist List: Administrators can view the complete list of artists with their details.**
2. **Service Management:**
   * **Add Service: Administrators can add services for each artist, specifying the name, description, and price.**
   * **Delete Service: Administrators can delete services that are no longer provided.**
   * **Edit Service: Administrators can edit existing services by updating the description or price.**
   * **Link Services to Artists: Each artist has a specific set of services they provide.**

**3. Technical Requirements**

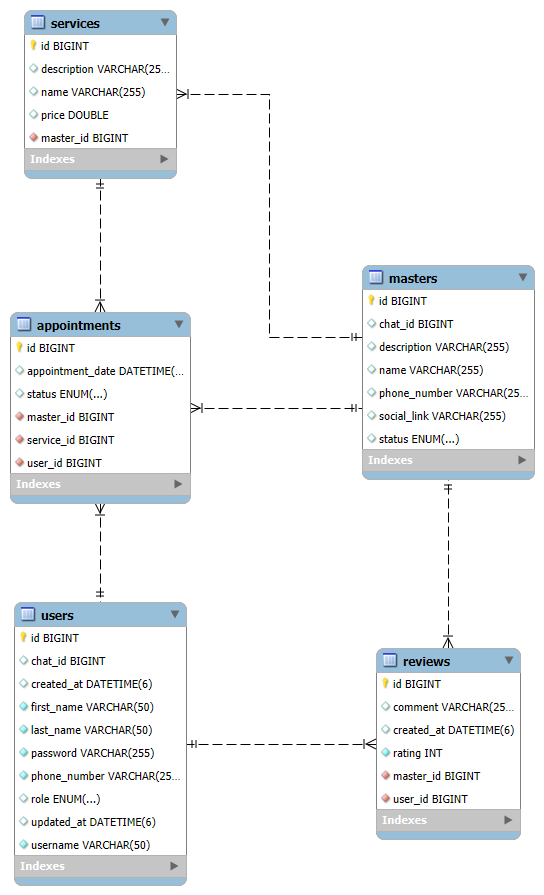
* **Telegram API: For interacting with Telegram.**
* **Java 17: The primary language for development.**
* **Spring Boot: To create the REST API and manage client interactions through the bot.**
* **Spring Data JPA: To work with the database.**
* **MySQL: The primary database to store client, artist, appointment, and service information.**
* **Docker: For easy deployment and containerization of the application.**
* **Liquibase: To manage database migrations.**
* **Instagram Integration: To redirect users to the artist’s Instagram page via Telegram.**

**4. Project Architecture**

**4.1 Main Components**

* **BotController: The main controller for handling incoming messages from Telegram.**
* **UserService: Manages user functionality such as registration, login, and appointment viewing.**
* **AdminService: Handles administrative tasks such as managing bookings and blocking users.**
* **AppointmentService: Manages appointment booking (choosing date, time, and service).**
* **NotificationService: Handles reminders for upcoming appointments.**
* **ReviewService: Manages customer reviews.**
* **MasterService: Manages artists—adding, deleting, and viewing artist information.**
* **ProcedureService: Manages services—adding, deleting, and editing available services.**
* **Database (MySQL): Stores users, artists, schedules, services, appointments, and reviews.**

**4.2 Database Schema (ERD)**



* **Users (users): Table for client and administrator information.**
* **Artists (masters): Table for nail artist information.**
* **Services (services): List of services with prices, linked to specific artists.**
* **Appointments (appointments): Stores client appointments with the selected artist, including date, time, and service.**
* **Reviews (reviews): Stores client reviews of nail artists.**

**4.3 Database Tables**

**4.3.1 Users Table (users)**

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| **id** | **BIGINT** | **Unique identifier for the user (primary key)** |
| **username** | **VARCHAR(255)** | **User's username (unique)** |
| **password** | **VARCHAR(255)** | **User's password** |
| **role** | **ENUM** | **User role (CLIENT, ADMIN)** |
| **chat\_id** | **BIGINT** | **Telegram chat ID** |
| **phone\_number** | **VARCHAR(15)** | **User's phone number** |
| **created\_at** | **TIMESTAMP** | **Date and time when the account was created** |
| **updated\_at** | **TIMESTAMP** | **Date and time when the account was last updated** |
| **first\_name** | **VARCHAR(50)** | **User's first name** |
| **last\_name** | **VARCHAR(50)** | **User's last name** |

**4.3.2 Artists Table (masters)**

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| **id** | **BIGINT** | **Unique identifier for the artist (primary key)** |
| **name** | **VARCHAR(255)** | **Artist's name** |
| **description** | **TEXT** | **Description of the artist (e.g., specialization)** |
| **social\_link** | **VARCHAR(255)** | **Link to the artist's social media profile** |
| **status** | **ENUM** | **Status of the artist (ACTIVE, INACTIVE)** |
| **chat\_id** | **BIGINT** | **Telegram chat ID** |
| **phone\_number** | **VARCHAR(15)** | **Artist's phone number** |

**4.3.3 Services Table (services)**

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| **id** | **BIGINT** | **Unique identifier for the service (primary key)** |
| **name** | **VARCHAR(255)** | **Service name** |
| **description** | **TEXT** | **Service description** |
| **price** | **DECIMAL(10,2)** | **Price of the service** |
| **master\_id** | **BIGINT** | **ID of the associated artist (foreign key)** |

**4.3.4 Appointments Table (appointments)**

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| **id** | **BIGINT** | **Unique identifier for the appointment (primary key)** |
| **user\_id** | **BIGINT** | **ID of the client (foreign key)** |
| **master\_id** | **BIGINT** | **ID of the artist (foreign key)** |
| **service\_id** | **BIGINT** | **ID of the selected service (foreign key)** |
| **appointment\_date** | **TIMESTAMP** | **Date and time of the appointment** |
| **status** | **ENUM** | **Status of the appointment (CONFIRMED, CANCELLED, COMPLETED)** |

**4.3.5 Reviews Table (reviews)**

| **Field** | **Data Type** | **Description** |
| --- | --- | --- |
| **id** | **BIGINT** | **Unique identifier for the review (primary key)** |
| **user\_id** | **BIGINT** | **ID of the client (foreign key)** |
| **master\_id** | **BIGINT** | **ID of the artist (foreign key)** |
| **rating** | **INT** | **Review rating (from 1 to 5)** |
| **comment** | **TEXT** | **Review comment** |
| **created\_at** | **TIMESTAMP** | **Date and time the review was created** |

**5. Administrator Interactions with Artists and Services**

1. **Add Artist: The administrator can add a new artist through the bot or management panel.**
2. **Delete Artist: The administrator removes an artist, and their services are either deleted or reassigned to another artist.**
3. **Add Service: For each artist, the administrator can add specific services with corresponding descriptions and prices.**
4. **Edit Service: The administrator can update the name, description, and price of existing services.**

**6. Using @BotFather**

**To create and configure the bot in Telegram, @BotFather is used:**

1. **Find @BotFather on Telegram and create a new bot using the /newbot command.**
2. **Choose a name for the bot and receive a token for API access.**
3. **Set up the token in the Spring Boot project for Telegram API integration.**

**7. Use Cases**

1. **Client Appointment Booking: A client selects an artist, date, time, and service, and then confirms the appointment.**
2. **Cancel Appointment: A client can cancel their appointment using a designated command.**
3. **Reminders: The bot automatically sends a reminder to the client one day and three hours before the appointment.**
4. **Leave a Review: After the service, the client is prompted to leave a review.**
5. **Manage Artists: The administrator adds, removes, or updates artists through the bot or interface.**
6. **Manage Services: The administrator adds, removes, or edits services for each artist.**

**8. Development Plan**

1. **Week 1: Create the bot via @BotFather and integrate with the Telegram API.**
2. **Week 2: Develop client functionality: registration, booking appointments.**
3. **Week 3: Develop the administrative panel: managing artists, services, appointments, and reviews.**
4. **Week 4: Implement the reminder system and add the ability to edit services and artists.**
5. **Week 5: Testing, debugging, and deployment to the server.**