**Документация к проекту "Telegram-бот для записи на маникюр"**

**1. Общее описание проекта**

* **Название проекта**: Telegram-бот для записи на маникюр.

**Описание**: Этот документ описывает проект Telegram-бота для записи на маникюр. Бот автоматизирует процесс бронирования услуг, управления записями и взаимодействия с мастерами и администраторами.

**Цель проекта**: Целью проекта является создание удобного Telegram-бота, который позволит пользователям записываться на услуги маникюра, управлять записями, получать уведомления и общаться с мастерами и администраторами.

* **Основные технологии**:
  + Язык: Java 17
  + Фреймворки: Spring Boot, Spring Data JPA
  + Базы данных: MySQL
  + Взаимодействие с Telegram API через библиотеку TelegramBots
  + Docker для деплоя
  + Liquibase для управления миграциями БД

**2. Функциональные возможности**

## **2.1. Для клиентов**

- Регистрация в боте  
- Просмотр списка услуг  
- Запись на услугу  
- Выбор мастера, даты и времени  
- Управление активными, завершенными и отмененными записями  
- Отправка отзыва о мастере  
- Получение уведомлений за день и за три часа до сеанса

## **2.2. Для мастеров**

- Просмотр своих записей  
- Перенос или отмена записей  
- Управление расписанием  
- Общение с клиентами

## **2.3. Для администраторов**

- Добавление/удаление мастеров  
- Добавление/удаление услуг  
- Управление записями клиентов  
- Отправка уведомлений клиентам и мастерам

**3. Технические требования**

* **Telegram API**: Для взаимодействия с Telegram.
* **Java 17**: Основной язык разработки.
* **Spring Boot**: Для создания REST API и взаимодействия с клиентом через бота.
* **Spring Data JPA**: Для работы с базой данных.
* **MySQL**: Основная база данных для хранения информации о клиентах, мастерах, записях и процедурах.
* **Docker**: Для удобного деплоя и контейнеризации приложения.
* **Liquibase**: Для управления миграциями базы данных.
* **Интеграция с Instagram**: Для перенаправления на страницу мастера через Telegram.

**4. Архитектура проекта**

* 1. **Классы для работы с сущностями (Models)**

**User**

* **Представляет пользователя бота (клиента).**
* **Атрибуты: id, firstName, lastName, phone, language.**

**Master**

* **Представляет мастера, который выполняет услуги.**
* **Атрибуты: id, name, specialization, availability.**

**Services**

* **Представляет услуги, которые предлагаются клиентам.**
* **Атрибуты: id, name, description, price, duration.**

**Review**

* **Представляет отзывы клиентов о мастерах и услугах.**
* **Атрибуты: id, rating, comment, userId, masterId.**

**Appointment**

* **Представляет запись на услугу.**
* **Атрибуты: id, userId, masterId, serviceId, appointmentDate, status (например, активная, завершенная, отмененная).**
  1. **Репозитории (Repositories)**

**UserRepository**

* **Интерфейс для взаимодействия с базой данных для сущности User.**
* **Использует Spring Data JPA для создания, удаления, обновления и поиска пользователей.**

**MasterRepository**

* **Интерфейс для взаимодействия с базой данных для сущности Master.**
* **Управляет данными мастеров: добавление, обновление, удаление и поиск.**

**ServiceRepository**

* **Интерфейс для взаимодействия с базой данных для сущности Services.**
* **Управляет данными услуг: добавление, обновление, удаление и поиск.**

**ReviewRepository**

* **Интерфейс для взаимодействия с базой данных для сущности Review.**
* **Управляет данными отзывов: добавление, обновление, удаление и поиск.**

**AppointmentRepository**

* **Интерфейс для взаимодействия с базой данных для сущности Appointment.**
* **Управляет данными записей: создание, удаление, поиск, обновление статуса записи.**
  1. **Контроллеры (Controllers)**

**UserController**

* **Контроллер для управления пользователями.**
* **Реализует методы для получения, создания и удаления пользователей.**

**MasterController**

* **Контроллер для управления мастерами.**
* **Реализует методы для получения, создания, обновления и удаления мастеров.**

**ServiceController**

* **Контроллер для управления услугами.**
* **Реализует методы для получения, создания, обновления и удаления услуг.**

**ReviewController**

* **Контроллер для управления отзывами.**
* **Реализует методы для получения, создания и удаления отзывов.**

**AppointmentController**

* **Контроллер для управления записями.**
* **Реализует методы для создания, удаления и получения записей.**

**AdminController**

* **Контроллер для административных операций.**
* **Реализует методы для управления пользователями, мастерами, услугами, отзывами и записями.**
* **Доступ ограничен ролью ROLE\_ADMIN.**
  1. **Конфигурации (Configurations)**

**SecurityConfig**

* **Конфигурирует безопасность проекта.**
* **Настройка авторизации и аутентификации с использованием Spring Security.**
* **Ограничение доступа для различных ролей (например, администратор, мастер).**

**BotConfig**

* **Конфигурирует параметры Telegram-бота.**
* **Содержит токен бота и имя, используемое для взаимодействия с API Telegram.**
  1. **Сервисный слой (Service Layer)**

**UserService**

* **Содержит логику для работы с пользователями:**
  + **Регистрация пользователя.**
  + **Изменение данных пользователя.**

**MasterService**

* **Логика для работы с мастерами:**
  + **Добавление/удаление мастеров.**
  + **Обновление специализаций и доступности мастера.**

**ServiceService**

* **Логика для работы с услугами:**
  + **Добавление/удаление/обновление услуг.**
  + **Изменение цен и длительности.**

**ReviewService**

* **Логика для работы с отзывами:**
  + **Добавление отзывов.**
  + **Получение всех отзывов.**

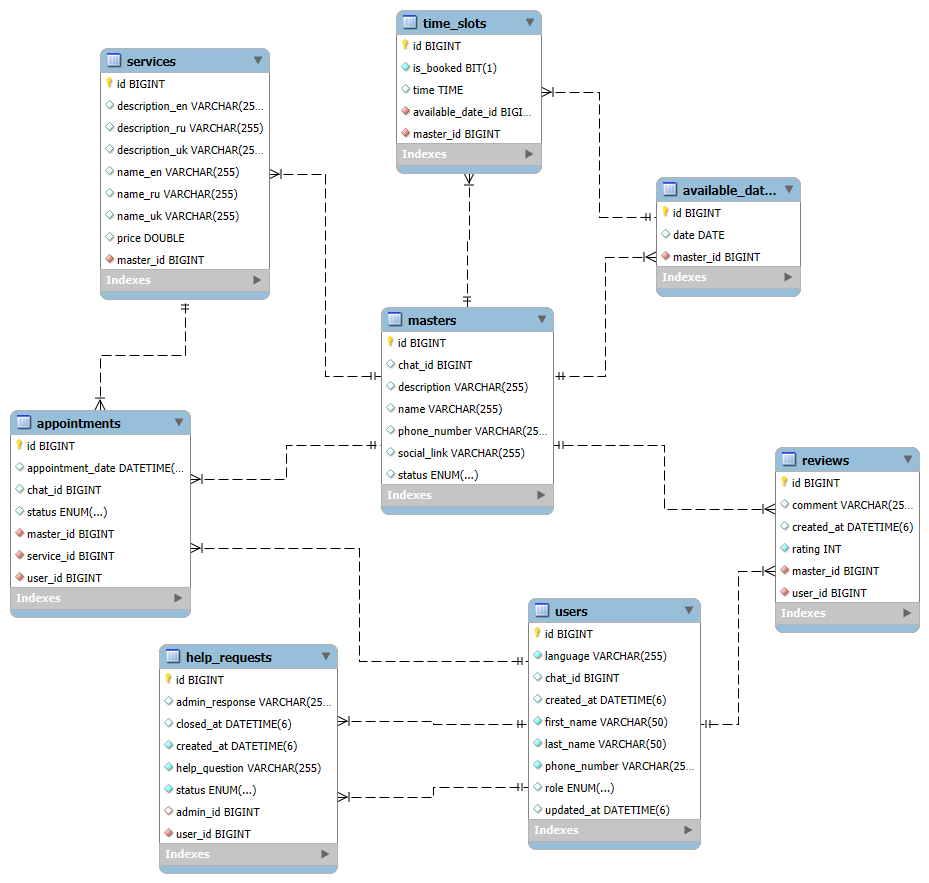
**AppointmentService**

* **Логика для работы с записями:**
  + **Создание, отмена, перенос записи.**
  + **Проверка доступности времени.**

**NotificationService**

* **Обрабатывает отправку уведомлений пользователям и мастерам.**
* **Уведомления о предстоящих услугах (за день и за 3 часа).**

**4.2 Диаграмма базы данных (ERD)**



* **Пользователи (users)**: Таблица с информацией о клиентах и администраторах.
* **Мастера (masters)**: Таблица с информацией о мастерах.
* **Процедуры (services)**: Список процедур с указанием цен, привязанных к мастерам.
* **Записи (appointments)**: Сохранение записей клиентов к мастерам с указанием даты, времени и процедуры.
* **Отзывы (reviews)**: Хранение отзывов клиентов о мастерах.
* **Помощь (help\_requests):** Таблица для запросов пользователей с возможностью назначения администратора.
* **Временные ячейки (time\_slots):** Таблица с доступными временными слотами для записи клиентов.
* **Возможные даты (available\_dates):** Таблица с датами доступности для записи клиентов.
* **4.3 Базы данных**
* **4.3.1 Таблица "Пользователи" (users)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор пользователя (первичный ключ) |
| language | VARCHAR(10) | Язык пользователя |
| role | ENUM | Роль пользователя (CLIENT, ADMIN) |
| chat\_id | BIGINT | Идентификатор чата в Telegram |
| phone\_number | VARCHAR(15) | Номер телефона пользователя |
| created\_at | TIMESTAMP | Дата и время создания записи |
| updated\_at | TIMESTAMP | Дата и время последнего обновления записи |
| first\_name | VARCHAR(50) | Имя пользователя |
| last\_name | VARCHAR(50) | Фамилия пользователя |

* **4.3.2 Таблица "Мастера" (masters)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор мастера (первичный ключ) |
| name | VARCHAR(255) | Имя мастера |
| description | TEXT | Описание мастера (например, специализация или услуги) |
| social\_link | VARCHAR(255) | Ссылка на социальные сети |
| status | ENUM | Статус мастера (ACTIVE, INACTIVE) |
| chat\_id | BIGINT | Идентификатор чата в Telegram |
| phone\_number | VARCHAR(15) | Номер телефона мастера |

* **4.3.3 Таблица "Процедуры" (services)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор процедуры (первичный ключ) |
| name\_ru | VARCHAR(255) | Название услуги на русском |
| name\_uk | VARCHAR(255) | Название услуги на украинском |
| name\_en | VARCHAR(255) | Название услуги на английском |
| description\_ru | TEXT | Описание услуги на русском |
| description\_uk | TEXT | Описание услуги на украинском |
| description\_en | TEXT | Описание услуги на английском |
| price | DOUBLE | Цена процедуры |
| master\_id | BIGINT | Идентификатор мастера (внешний ключ) |

* **4.3.4 Таблица "Записи" (appointments)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор записи (первичный ключ) |
| chat\_id | BIGINT | Идентификатор чата пользователя (Telegram) |
| user\_id | BIGINT | Идентификатор пользователя (внешний ключ) |
| master\_id | BIGINT | Идентификатор мастера (внешний ключ) |
| service\_id | BIGINT | Идентификатор процедуры (внешний ключ) |
| appointment\_date | TIMESTAMP | Дата и время записи |
| status | ENUM | Статус записи (CONFIRMED, CANCELLED, COMPLETED) |

* **4.3.5 Таблица "Отзывы" (reviews)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор отзыва (первичный ключ) |
| user\_id | BIGINT | Идентификатор пользователя (внешний ключ) |
| master\_id | BIGINT | Идентификатор мастера (внешний ключ) |
| rating | INT | Оценка отзыва (от 1 до 5) |
| comment | TEXT | Комментарий отзыва |
| created\_at | TIMESTAMP | Дата и время создания отзыва |

* **4.3.6 Таблица "Помощь" (help\_requests)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор отзыва (первичный ключ) |
| user\_id | BIGINT | Пользователь, создавший запрос |
| admin\_id | BIGINT | Администратор, который отвечает за запрос (nullable) |
| help\_question | TEXT | Вопрос, заданный пользователем |
| admin\_response | TEXT | Ответ администратора |
| created\_at | TIMESTAMP | Время создания запроса |
| closed\_at | TIMESTAMP | Время закрытия запроса |
| status | VARCHAR(10) | Статус запроса (ожидает, открыт, закрыт) |

* **4.3.7 Таблица "Временные ячейки" (time\_slots)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор отзыва (первичный ключ) |
| available\_date\_id | BIGINT | Связь с доступной датой |
| time | TIME | Время записи |
| is\_booked | BOOLEAN | Флаг, показывающий, забронирован ли слот |
| master\_id | BIGINT | Мастер, к которому привязан временной слот |

* **4.3.8 Таблица "Возможные даты" (available\_dates)**

| **Поле** | **Тип данных** | **Описание** |
| --- | --- | --- |
| id | BIGINT | Уникальный идентификатор отзыва (первичный ключ) |
| date | DATE | Дата доступности для записи |
| master\_id | BIGINT | Мастер, для которого доступна эта дата |

**5. Примеры взаимодействия администратора с мастерами и процедурами**

1. **Добавление мастера**: Администратор через интерфейс бота или панель управления может добавить нового мастера.
2. **Удаление мастера**: Администратор удаляет мастера, и его процедуры либо удаляются, либо переназначаются другим мастерам.
3. **Добавление процедуры**: Для каждого мастера можно добавить индивидуальный набор процедур с указанием цены и описания.
4. **Редактирование процедуры**: Администратор может редактировать название, описание и цену существующих процедур.

# **6. API-эндпоинты**

1. Для взаимодействия с ботом доступны следующие API-эндпоинты:  
   - POST /api/appointments - создание записи  
   - GET /api/appointments - просмотр всех записей  
   - DELETE /api/appointments/{id} - удаление записи  
   - POST /api/users - регистрация пользователя  
   - GET /api/users - получение информации о пользователе

**7. Сценарии использования**

1. **Запись клиента на процедуру**: Клиент выбирает мастера, дату, время и процедуру, затем подтверждает запись.
2. **Отмена записи**: Клиент может отменить свою запись через соответствующую команду.
3. **Напоминания**: Бот автоматически отправляет клиенту уведомление за день и за три часа до процедуры.
4. **Оставление отзыва**: После завершения процедуры клиент получает запрос на оставление отзыва.
5. **Управление мастерами**: Администратор добавляет, удаляет или обновляет мастеров через бот или интерфейс.
6. **Управление процедурами**: Администратор добавляет, удаляет или редактирует процедуры для каждого мастера.

**8. План разработки**

1. **Неделя 1**: Создание бота через @BotFather, интеграция с Telegram API.
2. **Неделя 2**: Разработка функционала для клиентов: регистрация, запись на процедуры.
3. **Неделя 3**: Разработка административной части: управление мастерами, процедурами, записями и отзывами.
4. **Неделя 4**: Реализация системы напоминаний и добавление возможности редактировать процедуры и мастеров.
5. **Неделя 5**: Тестирование, отладка проекта и деплой на сервер.