**Документація до проекту "Telegram-бот для запису на манікюр"**

**1. Загальний опис проекту**

* **Назва проекту: Telegram-бот для запису на манікюр.**
* **Опис: Telegram-бот допомагає користувачам записуватися на манікюр, переглядати ціни, відгуки, фото робіт майстрів та управляти своїми записами. Бот призначений як для клієнтів, так і для адміністраторів салону.**
* **Мета проекту: Автоматизувати процес запису на послуги манікюру та спростити взаємодію клієнтів і адміністраторів через Telegram.**
* **Основні технології:**
  + **Мова: Java 17**
  + **Фреймворки: Spring Boot, Spring Data JPA**
  + **Бази даних: MySQL**
  + **Взаємодія з Telegram API через бібліотеку TelegramBots**
  + **Docker для деплоя**
  + **Liquibase для управління міграціями БД**

**2. Функціональні можливості**

**2.1 Функціонал для всіх користувачів (гостей)**

1. **Перегляд цін: Клієнти можуть бачити список доступних процедур та їхню вартість.**
2. **Відгуки: Перегляд відгуків інших клієнтів про майстрів.**
3. **Галерея робіт: Перегляд фотографій готових робіт з можливістю перейти на Instagram-сторінку майстра для більш детального перегляду.**
4. **Реєстрація/вхід: Користувач може створити новий акаунт або увійти в існуючий.**

**2.2 Функціонал для зареєстрованих клієнтів**

1. **Вибір майстра: Після авторизації клієнт може вибрати майстра, до якого він хоче записатися.**
2. **Перегляд календаря: Можливість переглядати доступні дати на місяць вперед для запису до обраного майстра.**
3. **Вибір часу: Після вибору дати клієнт бачить доступні часові інтервали для запису.**
4. **Вибір процедури: Клієнт вибирає потрібну процедуру, поруч з якою вказана ціна.**
5. **Підтвердження запису: Після вибору процедури клієнт підтверджує запис.**
6. **Нагадування:**
   * **За день до процедури бот надсилає сповіщення про майбутній запис.**
   * **За три години до запису бот надсилає нагадування з вказівкою часу та процедури.**
7. **Сповіщення для майстрів:**
   * **Майстру автоматично приходить сповіщення про новий запис у момент його підтвердження клієнтом.**
   * **Також майстру надсилаються нагадування про запис за день і за три години до процедури.**
8. **Скасування/перенесення запису: Клієнт може скасувати або перенести свою запис.**
9. **Перегляд історії записів: Клієнт може бачити свої минулі та майбутні записи.**
10. **Залишити відгук: Після завершення процедури клієнту буде запропоновано залишити відгук про майстра.**

**2.3 Функціонал для адміністратора**

1. **Відкриття/закриття дат і часу для запису: Адміністратор може управляти доступними датами та часовими слотами для запису клієнтів.**
2. **Запис клієнта вручну: Адміністратор може записати клієнта на процедуру або скасувати існуючу запис.**
3. **Управління доступом клієнтів: Можливість блокувати або видаляти акаунти клієнтів.**
4. **Відправка повідомлень клієнтам: Адміністратор може надсилати клієнтам повідомлення через бота (наприклад, нагадування або важливі сповіщення).**
5. **Перегляд відгуків: Адміністратор може переглядати та управляти відгуками клієнтів.**

**2.4 Додаткові можливості для адміністратора**

1. **Управління майстрами:**
   * **Додавання майстра: Адміністратор може додавати нових майстрів, вказавши їх ім'я, опис та інші дані (наприклад, контактні дані, посилання на соціальні мережі).**
   * **Видалення майстра: Адміністратор може видалити майстра з бази даних. При цьому всі записи та процедури, пов'язані з майстром, будуть також видалені або переназначені іншим майстрам (на вибір адміністратора).**
   * **Перегляд списку майстрів: Адміністратор може бачити повний список майстрів з їхніми даними.**
2. **Управління процедурами:**
   * **Додавання процедури: Адміністратор може додавати процедури для кожного майстра окремо, вказавши назву, опис та ціну.**
   * **Видалення процедури: Адміністратор може видаляти процедури, які більше не надаються.**
   * **Редагування процедури: Адміністратор може змінювати існуючі процедури, оновлюючи їхній опис або ціну.**
   * **Прив'язка процедур до майстрів: Кожен майстер має індивідуальний набір процедур, які він надає.**

**3. Технічні вимоги**

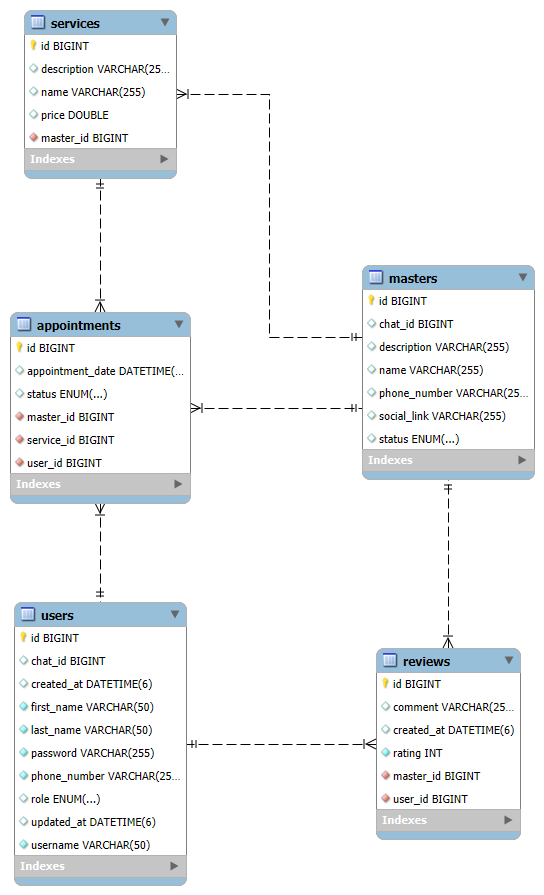
* **Telegram API: Для взаємодії з Telegram.**
* **Java 17: Основна мова розробки.**
* **Spring Boot: Для створення REST API та взаємодії з клієнтом через бота.**
* **Spring Data JPA: Для роботи з базою даних.**
* **MySQL: Основна база даних для зберігання інформації про клієнтів, майстрів, записи та процедури.**
* **Docker: Для зручного деплоя та контейнеризації додатку.**
* **Liquibase: Для управління міграціями бази даних.**
* **Інтеграція з Instagram: Для перенаправлення на сторінку майстра через Telegram.**

**4. Архітектура проекту**

**4.1 Основні компоненти**

* **BotController: Основний контролер для обробки вхідних повідомлень з Telegram.**
* **UserService: Логіка роботи з користувачами — реєстрація, вхід, перегляд записів.**
* **AdminService: Логіка для адміністратора — управління записями, блокування користувачів.**
* **AppointmentService: Управління записями на процедури (вибір дати, часу, процедури).**
* **NotificationService: Система нагадувань про майбутні записи.**
* **ReviewService: Логіка для управління відгуками клієнтів.**
* **MasterService: Управління майстрами — додавання, видалення та перегляд майстрів.**
* **ProcedureService: Управління процедурами — додавання, видалення та редагування процедур.**
* **База даних (MySQL): Зберігання користувачів, майстрів, розкладів, процедур, записів та відгуків.**

**4.2 Діаграма бази даних (ERD)**



* **Користувачі (users): Таблиця з інформацією про клієнтів та адміністраторів.**
* **Майстри (masters): Таблиця з інформацією про майстрів.**
* **Процедури (services): Список процедур з вказівкою цін, прив'язаних до майстрів.**
* **Записи (appointments): Збереження записів клієнтів до майстрів з вказівкою дати, часу та процедури.**
* **Відгуки (reviews): Зберігання відгуків клієнтів про майстрів.**

**4.3 Бази даних**

**4.3.1 Таблиця "Користувачі" (users)**

| **Поле** | **Тип даних** | **Опис** |
| --- | --- | --- |
| **id** | **BIGINT** | **Унікальний ідентифікатор користувача (первинний ключ)** |
| **username** | **VARCHAR(255)** | **Ім'я користувача (унікальне)** |
| **password** | **VARCHAR(255)** | **Пароль користувача** |
| **role** | **ENUM** | **Роль користувача (CLIENT, ADMIN)** |
| **chat\_id** | **BIGINT** | **Ідентифікатор чату в Telegram** |
| **phone\_number** | **VARCHAR(15)** | **Номер телефону користувача** |
| **created\_at** | **TIMESTAMP** | **Дата та час створення запису** |
| **updated\_at** | **TIMESTAMP** | **Дата та час останнього оновлення запису** |
| **first\_name** | **VARCHAR(50)** | **Ім'я користувача** |
| **last\_name** | **VARCHAR(50)** | |  | | --- | |  |  |  | | --- | | **Прізвище користувача** | |

**4.3.2 Таблиця "Майстри" (masters)**

| **Поле** | **Тип даних** | **Опис** |
| --- | --- | --- |
| **id** | **BIGINT** | **Унікальний ідентифікатор майстра (первинний ключ)** |
| **name** | **VARCHAR(255)** | **Ім'я майстра** |
| **description** | **TEXT** | **Опис майстра (наприклад, спеціалізація або послуги)** |
| **social\_link** | **VARCHAR(255)** | **Посилання на соціальні мережі** |
| **status** | **ENUM** | **Статус майстра (ACTIVE, INACTIVE)** |
| **chat\_id** | **BIGINT** | **Ідентифікатор чату в Telegram** |
| **phone\_number** | **VARCHAR(15)** | **Номер телефону майстра** |

**4.3.3 Таблиця "Процедури" (services)**

| **Поле** | **Тип даних** | **Опис** |
| --- | --- | --- |
| **id** | **BIGINT** | **Унікальний ідентифікатор процедури (первинний ключ)** |
| **name** | **VARCHAR(255)** | **Назва процедури** |
| **description** | **TEXT** | **Опис процедури** |
| **price** | **DECIMAL(10, 2)** | **Ціна процедури** |
| **master\_id** | **BIGINT** | **Ідентифікатор майстра (зовнішній ключ)** |

**4.3.4 Таблиця "Записи" (appointments)**

| **Поле** | **Тип даних** | **Опис** |
| --- | --- | --- |
| **id** | **BIGINT** | **Унікальний ідентифікатор запису (первинний ключ)** |
| **user\_id** | **BIGINT** | **Ідентифікатор користувача (зовнішній ключ)** |
| **master\_id** | **BIGINT** | **Ідентифікатор майстра (зовнішній ключ)** |
| **service\_id** | **BIGINT** | **Ідентифікатор процедури (зовнішній ключ)** |
| **appointment\_date** | **TIMESTAMP** | **Дата та час запису** |
| **status** | **ENUM** | **Статус запису (CONFIRMED, CANCELLED, COMPLETED)** |

**4.3.5 Таблиця "Відгуки" (reviews)**

| **Поле** | **Тип даних** | **Опис** |
| --- | --- | --- |
| **id** | **BIGINT** | **Унікальний ідентифікатор відгуку (первинний ключ)** |
| **user\_id** | **BIGINT** | **Ідентифікатор користувача (зовнішній ключ)** |
| **master\_id** | **BIGINT** | **Ідентифікатор майстра (зовнішній ключ)** |
| **rating** | **INT** | **Оцінка відгуку (від 1 до 5)** |
| **comment** | **TEXT** | **Коментар відгуку** |
| **created\_at** | **TIMESTAMP** | **Дата та час створення відгуку** |

**5. Приклади взаємодії адміністратора з майстрами та процедурами**

1. **Додавання майстра: Адміністратор через інтерфейс бота або панель управління може додати нового майстра.**
2. **Видалення майстра: Адміністратор видаляє майстра, і його процедури або видаляються, або переназначаються іншим майстрам.**
3. **Додавання процедури: Для кожного майстра можна додати індивідуальний набір процедур з вказівкою ціни та опису.**
4. **Редагування процедури: Адміністратор може редагувати назву, опис і ціну існуючих процедур.**

**6. Використання @BotFather**

**Для створення та налаштування бота в Telegram використовується @BotFather:**

1. **Знайти @BotFather в Telegram і створити нового бота через команду /newbot.**
2. **Придумати ім'я для бота та отримати токен для доступу до API.**
3. **Налаштувати токен у Spring Boot проекті для інтеграції з Telegram API.**

**7. Сценарії використання**

1. **Запис клієнта на процедуру: Клієнт вибирає майстра, дату, час і процедуру, потім підтверджує запис.**
2. **Скасування запису: Клієнт може скасувати свою запис через відповідну команду.**
3. **Нагадування: Бот автоматично надсилає клієнту сповіщення за день і за три години до процедури.**
4. **Залишення відгуку: Після завершення процедури клієнт отримує запит на залишення відгуку.**
5. **Управління майстрами: Адміністратор додає, видаляє або оновлює майстрів через бот або інтерфейс.**
6. **Управління процедурами: Адміністратор додає, видаляє або редагує процедури для кожного майстра.**

**8. План розробки**

1. **Тиждень 1: Створення бота через @BotFather, інтеграція з Telegram API.**
2. **Тиждень 2: Розробка функціоналу для клієнтів: реєстрація, запис на процедури.**
3. **Тиждень 3: Розробка адміністративної частини: управління майстрами, процедурами, записами та відгуками.**
4. **Тиждень 4: Реалізація системи нагадувань та додавання можливості редагувати процедури і майстрів.**
5. **Тиждень 5: Тестування, налагодження проекту та деплой на сервер.**