

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное автономное образовательное учреждение
высшего образования Санкт-Петербургский национальный
исследовательский университет информационных технологий механики и
оптики

Мега факультет трансляционных информационных технологий

Факультет информационных технологий и программирования

Лабораторная работа № 2

Современные вопросы информатики

**Разработка минимально жизнеспособного продукта: создание веб-
сайта для системы управления записями**

Этап 2

Выполнили студенты группы М34091

Джахан Исрат

Санкт-Петербург

2024

Цель этапа

1. Научиться планировать объем проекта на предстоящий семестр
2. Научиться проектировать проект для выполнения проекта

Разработка минимально жизнеспособного продукта: создание веб-сайта для системы управления записями

Требования к системе управления записями на прием

Фронтенд: Bootstrap

Бэкенд: Django

Инструмент: FullCalendar

База данных: Sqlite3

Первоначальный хостинг: Python Anywhere

Платформа для разработчиков: Github

Функциональные требования

1. **Создание записи на прием:** а. Пользователи могут создавать записи, выбирая дату и время с помощью интерфейса FullCalendar.
2. б. Записи могут быть созданы для любого человека с вводом соответствующей информации.
3. **Просмотр записей на прием:** а. Пользователи могут просматривать предстоящие записи непосредственно на интерфейсе календаря.
- б. Предстоящие записи будут выделены на календаре.
4. **Автоматическое завершение записей на прием:** а. Записи будут автоматически отмечены как завершенные после окончания назначенного дня.
- б. Записи будут отсортированы в таблице "Идущие" или "Все".
5. **Редактирование и удаление записей:** а. Пользователи могут редактировать детали записи (например, перенести или изменить информацию) прямо из календаря.
- б. Пользователи смогут удалять записи при необходимости.

- 6. **Аутентификация пользователей:** а. Пользователи должны быть аутентифицированы для создания, редактирования или удаления записей.
 - б. Функционал входа с проверкой данных должен быть реализован.
- 7. **Поиск записей:** а. Пользователи смогут искать записи с использованием фильтров, таких как имя, дата и статус.

Нефункциональные требования

- **Масштабируемость:** а. Система будет масштабируемой для обработки увеличивающегося числа записей и пользователей с течением времени.
- **Производительность:** а. Взаимодействие с календарем (загрузка, создание и редактирование записей) будет отзывчивым с минимальными задержками.
 - б. Система будет эффективно справляться с пиковыми нагрузками, особенно когда несколько пользователей взаимодействуют с календарем.
- **Безопасность:** а. Данные записей будут безопасно храниться и передаваться.
 - б. Аутентификация и авторизация пользователей должны быть реализованы с соблюдением лучших практик безопасности.
- **Удобство использования:** а. Интерфейс календаря будет интуитивно понятным и удобным, с возможностью перетаскивания для переноса записей.
 - б. Формы для создания записей будут легкими для заполнения, а сообщения об ошибках – понятными.
- **Надежность:** а. Система будет обладать высокой доступностью, особенно для функций просмотра и создания записей.
 - б. Механизмы обработки ошибок будут настроены для корректной работы в случае возникновения проблем.
- **Совместимость:** а. Система будет совместима со всеми веб-браузерами.

- **Резервное копирование и восстановление данных:** а. Данные о записях будут регулярно архивироваться, и должен быть предусмотрен процесс восстановления данных в случае сбоя.
- **Хостинг и развертывание:** а. Система будет изначально размещена на платформе PythonAnywhere, и скрипты развертывания и конфигурации будут оптимизированы для этой среды.

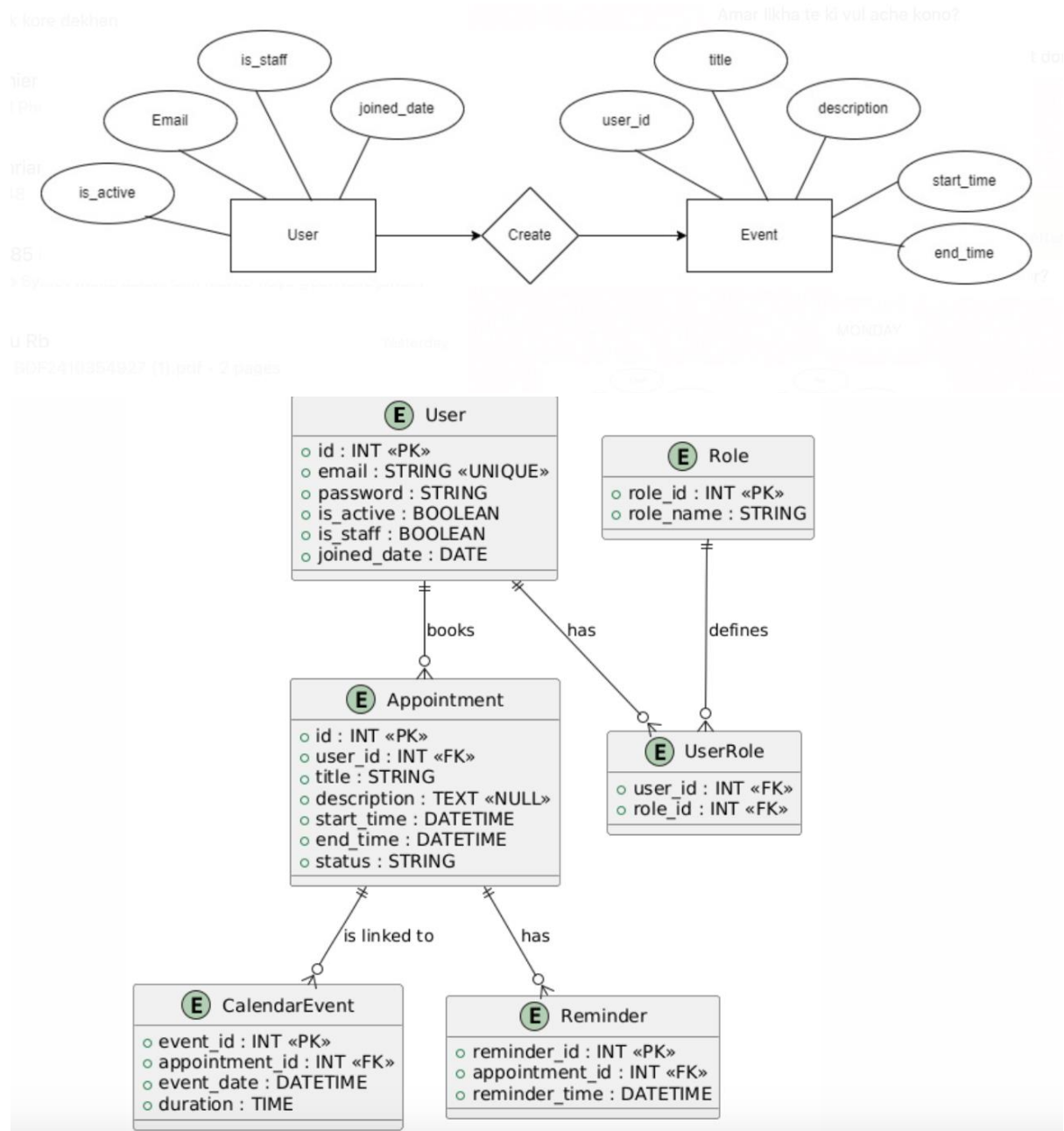
б. Кодовая база должна быть управляемой и версионизируемой с использованием GitHub.

- **Поддерживаемость:** а. Система будет спроектирована с учетом простоты поддержки, следуя лучшим практикам программирования и имея четкую документацию.

б. Будущие улучшения и исправления ошибок должны быть легко внедряемыми без влияния на существующую функциональность.

а. Интерфейс календаря будет интуитивно понятным и удобным, с возможностью перетаскивания для переноса записей.

б. Формы для создания записей будут легкими для заполнения, а сообщения об ошибках – понятными.



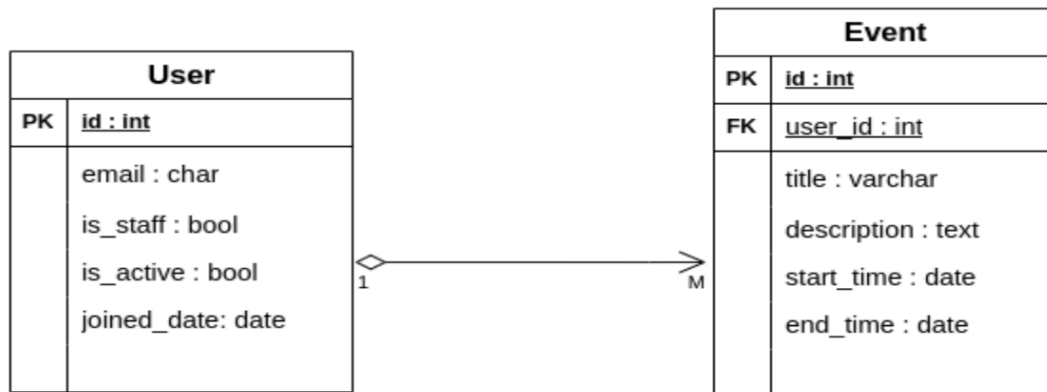
Объяснение:

- **User:** Представляет пользователей в системе с такими атрибутами, как электронная почта, пароль, статус и т.д.
- **Appointment:** Хранит информацию о каждой записи, включая пользователя, который её создал, время начала и окончания, а также статус (например, "ожидание", "завершено").
- **Role:** Представляет различные роли, которые могут быть у пользователей, такие как "Администратор", "Персонал" и т.д.
- **UserRole:** Таблица для связи пользователей с ролями.
- **CalendarEvent:** Хранит конкретные события, связанные с записями, такие как дата события и его продолжительность.
- **Reminder:** Позволяет устанавливать напоминания для записей, которые связаны с конкретной записью.

Связи:

- **User** может иметь несколько **Appointment**.
- **User** может иметь несколько **Role** через таблицу **UserRole**.
- У **Appointments** могут быть несколько **CalendarEvent** и **Reminder**

ERD Diagram:



Сущности и атрибуты:

1. User (Пользователь)

- id**: Целое число, Первичный ключ
- email**: Строка, Уникальный, Не может быть NULL
- is_active**: Булево значение, Не может быть NULL
- is_staff**: Булево значение, Не может быть NULL
- joined_date**: Дата, Не может быть NULL

2. Event (Событие)

- id**: Целое число, Первичный ключ
- user_id**: Целое число, Внешний ключ (ссылается на User.id)
- start_time**: Дата и время, Не может быть NULL
- end_time**: Дата и время, Не может быть NULL
- title**: Строка, Не может быть NULL
- description**: Текст, Может быть NULL

Отношения

- **User к Event**: Один-ко-многим
- ○ Один пользователь может создать несколько событий.