**1. Общие сведения**

* **Наименование системы/ПО:** «ХроноСфера: Мобильное приложение для управления временем и задачами»
* **Основание для разработки:** Курсовой проект по дисциплине «Разработка мобильных приложений»
* **Сроки разработки:** Плановый старт – 21 февраля; плановая сдача – 5 мая

**2. Назначение и цель создания ПО**

* **Цель разработки:** Создание инструмента для повышения личной продуктивности посредством автоматизации планирования, трекинга времени и контроля выполнения задач. Приложение направлено на оптимизацию процесса самоорганизации с применением современных методик тайм-менеджмента и элементов геймификации.
* **Назначение приложения:** «ХроноСфера» предназначена для учета и управления задачами, анализа временных затрат и контроля качества выполнения посредством модерации через интеграцию с Telegram ботом. Система позволяет пользователям фиксировать длительность задач с высокой точностью, получать регулярные уведомления и мотивироваться с помощью системы достижений и индикатора «Огонёк».
* **Целевая аудитория:** Приложение ориентировано на индивидуальных пользователей – студентов, молодых специалистов и фрилансеров, заинтересованных в эффективном управлении временем и самоорганизации.

**3. Требования к системе**

**3.1. Функциональные требования**

* **Основные функции:**
  + Регистрация и авторизация пользователей.
  + Создание, редактирование и удаление задач с выбором категории и метода тайм-менеджмента (Помодоро, метод 2 минут, тайм-блоки).
  + Точное фиксирование длительности задач в секундах.
  + Формирование аналитических отчетов о выполнении задач (по дням, неделям, месяцам).
  + Интеграция с Telegram: автоматическая отправка уведомлений о завершении задач и последующая модерация.
  + Реализация системы геймификации: индикатор «Огонёк» для отслеживания последовательных дней активности и система достижений.
* **Сценарии использования (use case):**
  + Пользователь регистрируется и входит в систему.
  + Создание задачи с указанием метода работы и длительности, выбор категории.
  + Запуск таймера для отслеживания выполнения задачи.
  + После завершения задачи происходит автоматическая отправка уведомления в Telegram.
  + Модератор проверяет выполненную задачу, изменяя её статус (принята/отправлена на доработку).
  + Пользователь просматривает аналитические отчеты и систему достижений.
* **Взаимодействие с другими системами:**
  + Интеграция с Telegram для уведомлений и модерации.
  + Возможное использование push-уведомлений внутри приложения.

**3.2. Технические требования**

* **Языки разработки и фреймворки:**
  + **Frontend:** Flutter (для разработки мобильного приложения).
  + **Backend:** Python с использованием FastAPI, aiogram для интеграции с Telegram, SQLAlchemy.
* **API:** RESTful API для связи мобильного приложения с серверной частью.
* **База данных:** PostgreSQL с использованием UUID для уникальной идентификации записей.
* **Архитектура:** Клиент-серверная архитектура, предусматривающая независимую разработку мобильного приложения и backend-сервиса, с интеграцией внешних сервисов.
* **Использованные сервисы:**
  + Среда разработки – Visual Studio Code, PyCharm
  + Система контроля версий – GitHub
  + Инструменты для дизайна UI/UX – Figma

**3.3. Требования к безопасности**

* **Защита данных:**
  + Регулярное резервное копирование базы данных.
* **Аутентификация и авторизация:**
  + Контроль доступа к данным с помощью ролевой модели.

**3.4. Требования к интерфейсу**

* **UI/UX:**
  + Минималистичный дизайн с акцентом на простоту и удобство использования.
  + Светлый, почти белый фон, использование ярких цветовых тегов для визуальной идентификации категорий задач.
  + Интуитивно понятная навигация, адаптированная для устройств на базе Android.

**4. Требования к тестированию**

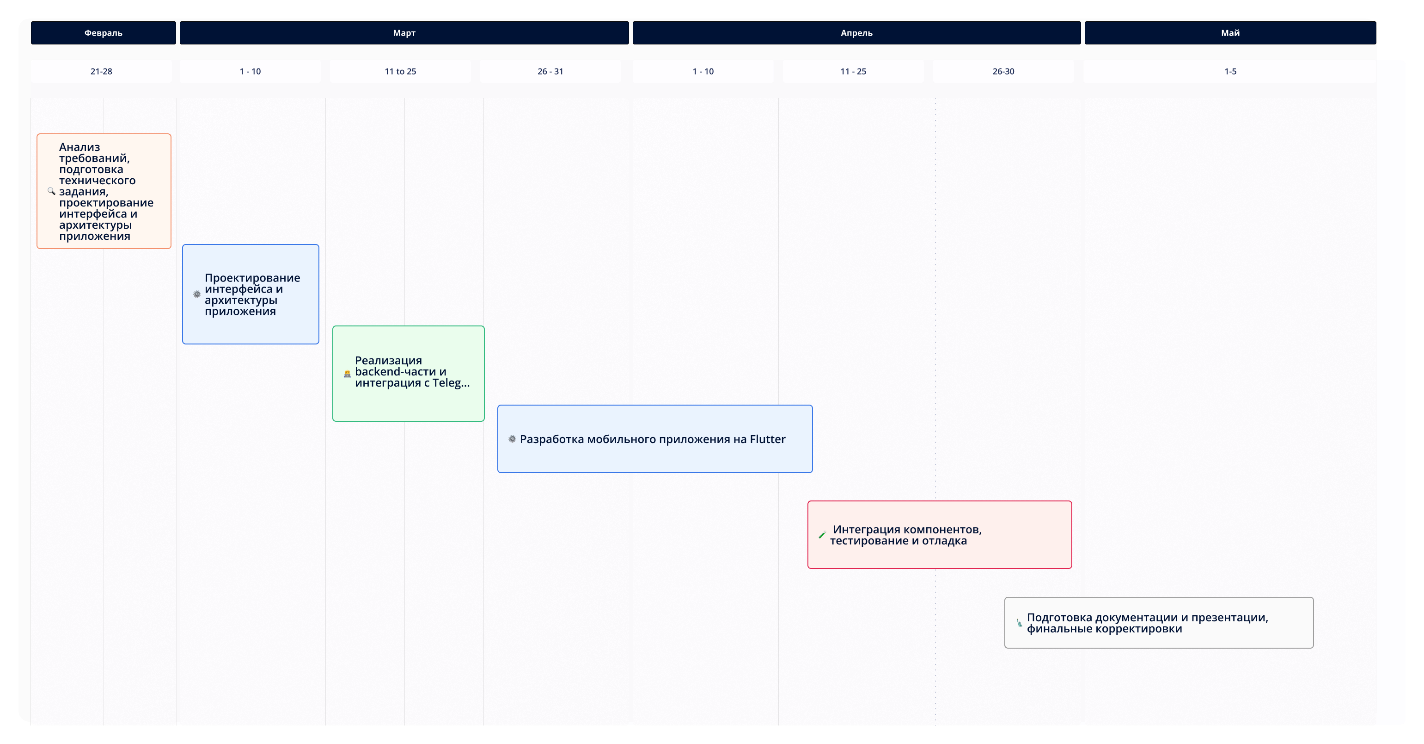
* **Методы тестирования:**
  + Интеграционное тестирование взаимодействия между мобильным приложением и backend-сервисами.
  + UI/UX тестирование с привлечением конечных пользователей.

**5. Требования к документации**

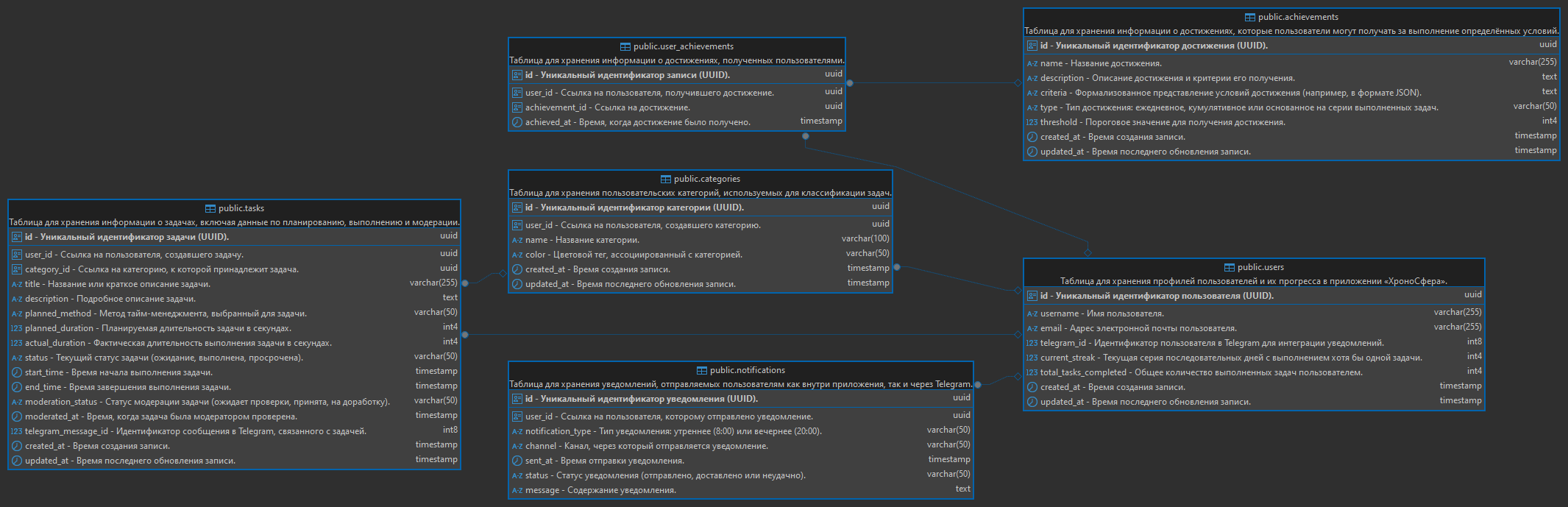
* **Перечень документов:**
  + Руководство пользователя с описанием функциональности приложения.
  + Техническая документация, включая описание архитектуры, API и схемы базы данных. (Swagger)
  + Инструкции по установке и настройке окружения.

**6. Условия эксплуатации**

* **Программные требования:**
  + Приложение разрабатывается для платформы Android.
* **Операционные требования:**
  + Минимальная версия операционной системы Android – 10 или выше.

**7. Стадии и этапы разработки**

**8. Приложения**

****