Техническое задание на разработку сервиса

«VVSUle»

Версия 1.0

2025

История изменения документа

|  |  |  |  |
| --- | --- | --- | --- |
| Версия | Дата | Автор | Изменения |
| 1.0 | 29.10.2025 | Федянин М.А. | Начало написания документа |

Общие сведения

Система – Telegram-бот «VVSUle», сервис, который представляет собой универсальный инструмент для получения актуального расписания занятий Владивостокского государственного университета. В отличие от существующих решений, требующих перехода на внешние сайты и установки приложений на смартфон, бот предоставляет персонализированную информацию на основе группы студента непосредственно в мессенджере Telegram.

Заказчиками и исполнителями выступают Федянин Максим Андреевич, Буров Владислав Анатольевич и Нечаев Евгений Алексеевич.

Плановые сроки начала и окончания работы по созданию системы

Этапы создания информационной системы с плановыми сроками представлены в таблице.

|  |  |
| --- | --- |
| **Этап** | **Сроки** |
| Планирование и анализ требований | 15 сентября – 28 сентября (2 недели) |
| Определение требований | 29 сентября – 5 октября (1 неделя) |
| Проектирование системы | 6 октября – 26 октября (3 недели) |
| Разработка системы | 27 октября – 27 декабря (8 недель) |
| Тестирование системы | 24 ноября – 21 декабря (4 недели) |
| Развертывание системы | 12 января – 31 января (3 недели) |

Стоимость разработки системы включает в себя заработную плату разработчика, проектировщика и аналитика, и затраты на инфраструктуру развертывания, общая стоимость составляет 72 000 рублей.

Назначение и цели создания системы

Система предназначена для автоматизации процесса получения и отображения расписания занятий ВВГУ для студентов и преподавателей.

Цели создания системы с точки зрения заказчика и пользователей:

– Предоставить быстрый доступ к актуальному расписанию занятий непосредственно в Telegram.

– Обеспечить персонализацию расписания на основе учебной группы пользователя.

– Обеспечить возможность просмотра расписания на различные периоды (день, неделя, месяц).

– Интегрировать веб-приложение через Telegram Web App.

Требования к системе

Функциональные требования

**Регистрация и настройка профиля пользователя**

– Пользователь имеет возможность зарегистрироваться в системе через команду /start.

– При регистрации пользователь должен ввести свою учебную группу.

– Пользователь может изменить выбранную группу в любое время.

– Система сохраняет настройки уведомлений для каждого пользователя.

**Получение расписания занятий**

– Пользователь имеет возможность получить расписание на текущий день через команду /today.

– Пользователь может просмотреть расписание на всю неделю через команду /week.

– Система предоставляет расписание в удобном форматированном виде с указанием:

* Времени проведения занятия.
* Названия предмета.
* ФИО преподавателя.
* Номера аудитории.
* Типа занятия (лекция, практика, лабораторная).

**Управление уведомлениями**

– Пользователь может включить/отключить уведомления об изменениях в расписании.

– Система отправляет уведомления при изменении расписания для группы пользователя.

– Пользователь может настроить время отправки ежедневных напоминаний о расписании.

**Веб-приложение**

– Пользователь имеет доступ к веб-приложению через кнопку в Telegram.

– Веб-приложение предоставляет расширенный функционал:

* Просмотр расписания в табличном виде.
* Тип занятия (лекция, практика, лабораторная).

– Веб-приложение адаптировано для использования внутри Telegram Web App.

**Поиск и навигация**

– Пользователь может найти расписание для любой группы через команду /group [номер группы].

– Система предоставляет интерактивное меню с кнопками для быстрой навигации.

Нефункциональные требования

**Требования к структуре и функционированию системы**

– Описание перечня подсистем, их назначение, компоненты системы и их информационное взаимодействие между собой представлено в разделе «Архитектура системы».

– Система взаимодействует с внешними источниками расписания ВВГУ через парсинг.

– Система должна быть доступна 24/7 с редкими профилактическими остановками.

**Требования к интерфейсу**

– Аккуратный, минималистичный дизайн.

– Цветовая схема включает в себя два цвета и их градиент согласно приложению «Цветовая схема».

– Сервис должен поддерживать светлую и темную темы.

– Пользователь имеет возможность изменить тему интерфейса через соответствующую кнопку.

– Все элементы интерфейса должны быть выполнены с помощью фреймворка для упрощения разработки.

– Примеры страниц, их наполнения и визуальной составляющей представлены в приложении «Макеты интерфейса».

**Требования к локализации**

– Интерфейс системы полностью на русском языке.

– Формат дат и времени соответствует российским стандартам.

– Поддержка временных зон для корректного отображения расписания.

Архитектура системы

**Основные сведения**

Система представляет собой Telegram-бота с интеграцией веб-приложения, доступного через Telegram Web App.

Архитектура основана на микросервисном подходе и разделена на три слоя: **базовый, дополнительный и внешний.**

**Компоненты архитектуры**

**Базовый слой** включает основные компоненты, обеспечивающие функциональность и взаимодействие системы:

– **Бэкенд:** реализуется на **Python** с использованием **aiogram**, обрабатывает входящие сообщения и команды.

– **Фронтенд:** реализуется на **React** + **FastAPI**, предоставляет веб-интерфейс для расширенного функционала.

– **База данных:** используется **PostgreSQL** в качестве основной реляционной СУБД для хранения структурированных данных приложения.

**Дополнительный слой** предназначен для повышения производительности, мониторинга и аналитики:

– **Кэширование:** реализуется с помощью **Redis**, используется для ускорения обработки часто запрашиваемых данных и уменьшения нагрузки на основную базу данных.

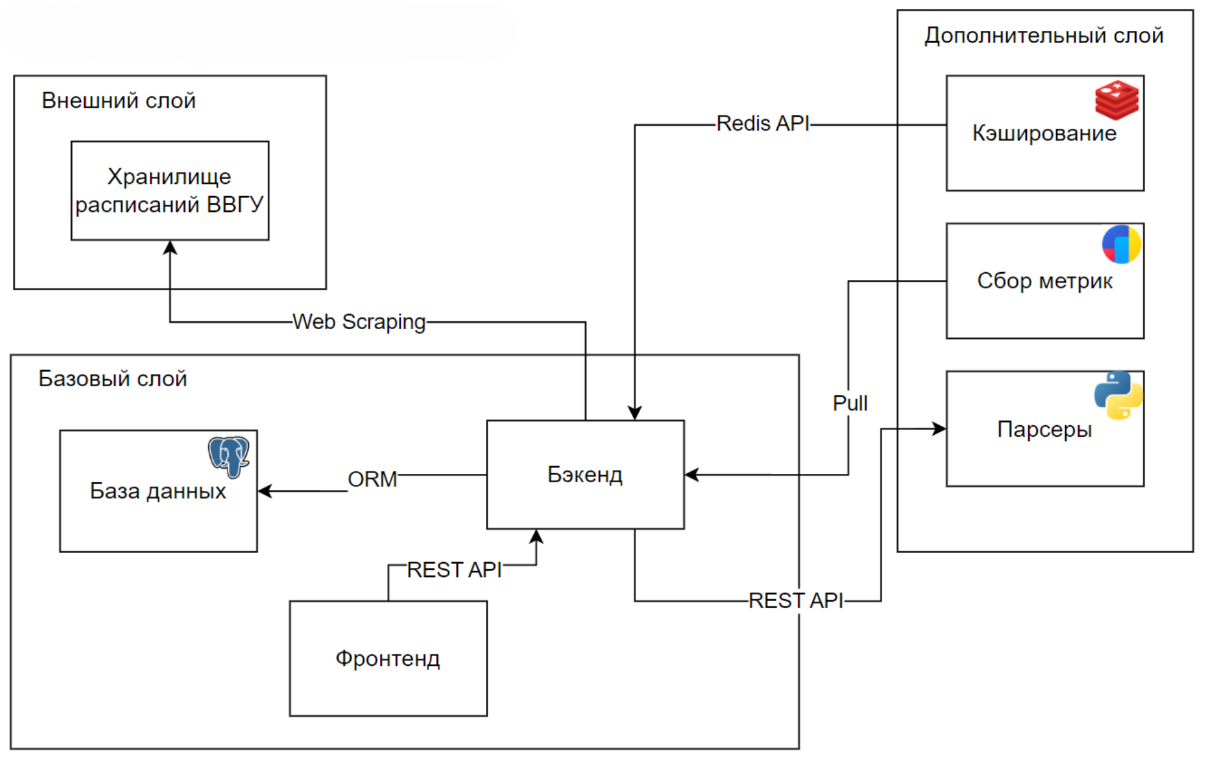
– **Сбор метрики:** осуществляется при помощи **Яндекс.Метрики**, который агрегирует показатели работы сервисов и инфраструктуры.

– **Парсеры:** реализуется в виде скриптов на Python, которые предназначены для парсинга расписания занятий с сайта ВВГУ.

**Внешний слой** содержит интеграцию с хранилищем расписаний занятий ВВГУ.

Каждый микросервис (за исключением внешних интеграций) развертывается в отдельном **Docker-контейнере**, что обеспечивает изоляцию окружений, гибкость масштабирования и удобство развертывания.

**Схема архитектуры**



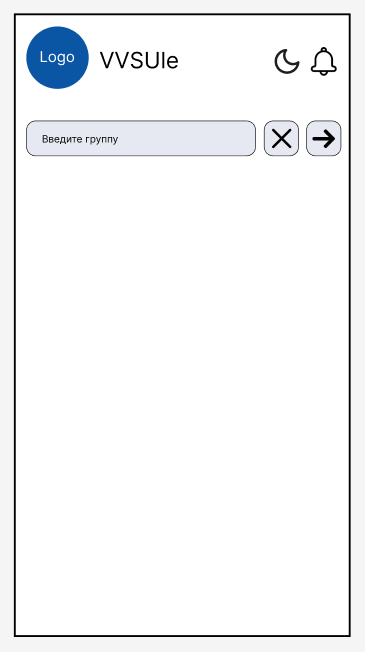
Приложения

**Цветовая схема**



**Макеты интерфейса**

– Главная страница



– Вывод расписания занятий

