**Спецификация технических требований (SRS) для веб-сервиса отслеживания занятости компьютерных классов 806 кафедры**

**1. Введение**

**1.1. Назначение**

Данный документ описывает технические требования к веб-сервису, который автоматизирует сбор, хранение, редактирование и отображение расписания компьютерных классов 806 кафедры МАИ.

**1.2. Область применения**

Проект предназначен для студентов, преподавателей и администрации кафедры, обеспечивая доступ к актуальному расписанию занятий и возможностям его редактирования.

**1.3. Определения и сокращения**

* **Scraper** – модуль, загружающий данные с сайта МАИ.
* **Парсер** – компонент, обрабатывающий и структурирующий данные.
* **CRUD** – операции создания, чтения, обновления и удаления данных.

**2. Функциональные требования**

**2.1. Модуль загрузки расписания**

* Получение данных с сайта МАИ (Selenium/Urllib).
* Кэширование загруженных данных.
* CLI-интерфейс для ручного запуска парсинга расписания.

**2.2. Модуль обработки и хранения данных**

* Преобразование данных в структурированный формат.
* Хранение в базе данных (SQLite/PostgreSQL).

**2.3. Модуль синхронизации**

* Экспорт расписания в Google Calendar и Google Spreadsheets.
* Автоматическая синхронизация при изменениях.

**2.4. Модуль редактирования (CRUD)**

* Добавление новых занятий.
* Редактирование существующих занятий.
* Удаление неактуальных занятий.
* Автоматическое применение изменений в БД.

**2.5. Модуль уведомлений**

* Отправка email/Telegram-уведомлений при изменении расписания.

**3. Нефункциональные требования**

**3.1. Производительность**

* Обновление расписания не должно превышать 5 минут.

**3.2. Надежность**

* Встроенный механизм логирования изменений.
* Резервное копирование БД.

**3.3. Масштабируемость**

* Возможность интеграции новых источников расписания.

**4. Архитектура системы**

**4.1. Общая структура**

1. **Scraper** → получает данные с сайта.
2. **Парсер** → обрабатывает и сохраняет их в БД.
3. **Сервер API (FastAPI/Django)** → предоставляет данные фронтенду и другим сервисам.
4. **Frontend (React/Vue.js)** → отображает расписание и позволяет его редактировать.
5. **Google API (Calendar, Sheets)** → синхронизирует данные.

**5. Технологический стек**

| **Компонент** | **Технология** |
| --- | --- |
| **Backend** | Python (FastAPI) |
| **Парсинг** | Selenium/Urllib |
| **База данных** | SQLite/PostgreSQL |
| **Frontend** | React/Vue.js |
| **Синхронизация** | Google API |

**6. Риски и их решение**

| **Риск** | **Решение** |
| --- | --- |
| Изменение структуры сайта МАИ | Разработка гибкого парсера |
| Недоступность Google API | Локальное кэширование данных |
| Ошибки в расписании | Журналирование и восстановление |

**7. Заключение**

Настоящий документ описывает архитектуру и технические требования к системе. Дальнейшая работа включает в себя разработку архитектурных диаграмм, описание API и выбор инструментов для реализации.