

# **Техническое описание проекта по курсу ООАД**

## **«Менеджер расписания»**

**Студенты ФИТ НГУ**

**Неретин Степан Иванович**

**Кондренко Кирилл Павлович**

**Группа 21203**

**Версия 1.2**

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
1.1	Цель . . . . .	3
1.2	Область действия . . . . .	3
1.3	Определения и сокращения . . . . .	3
1.4	Ссылки . . . . .	3
1.5	Краткое описание . . . . .	3
<b>2</b>	<b>Предметная область проекта</b>	<b>5</b>
2.1	Существующие проблемы . . . . .	5
2.2	Предполагаемое решение . . . . .	5
<b>3</b>	<b>Требования к программному решению</b>	<b>6</b>
3.1	Роли . . . . .	6
3.2	Функциональные требования для роли «Пользователь браузера»	7
3.2.1	Установить расписание . . . . .	7
3.3	Функциональные требования для роли «Пользователь браузера с расписанием» . . . . .	7
3.3.1	Осуществить операцию над расписанием . . . . .	7
3.4	Нефункциональные требования . . . . .	8
<b>4</b>	<b>Обзор архитектуры</b>	<b>9</b>
4.1	Подсистемы и компоненты проекта . . . . .	9
4.1.1	Server . . . . .	10
4.1.2	Extension . . . . .	12
4.2	Компоненты сторонних производителей . . . . .	13
4.3	Схема развёртывания проекта . . . . .	13
<b>5</b>	<b>Допущения и ограничения</b>	<b>14</b>
<b>6</b>	<b>Известные проблемы</b>	<b>15</b>
6.1	Отсутствие поддержки JWT . . . . .	15
6.2	Отсутствие защиты от DOS и DDOS-атак . . . . .	15
6.3	Невозможность работы расширения без сервера . . . . .	16
6.4	Зависимость сервера и расширения от вёрстки сайта с расписанием НГУ . . . . .	16

# Глава 1

## Введение

### 1.1 Цель

Данный документ представляет собой техническое описание проекта «Менеджер расписания» и содержит основные требования к разрабатываемой в рамках проекта программной системе и описание архитектуры программного решения.

### 1.2 Область действия

Документ разработан в рамках проекта «Менеджер расписания» на основе стандартного шаблона и предназначен для использования студентами ФИТ и преподавателями дисциплины ООАД.

### 1.3 Определения и сокращения

Термин	Описание
Предметы по выбору	Предметы, разделенные по блокам, причем в каждом блоке студент расставляет их по приоритетам, но в конечном итоге из каждого блока для студента определён <b>лишь один</b> предмет

### 1.4 Ссылки

### 1.5 Краткое описание

Содержание данного документа построено таким образом, чтобы дать ответ на следующие вопросы:

- Какие проблемы предметной области должен решать будущий программный продукт

- Посредством какой функциональности системы будут достигнуто решение проблем предметной области
- Какова архитектура программного решения

Описание предметной области и проблем, для решения которых предназначен будущий программный продукт, приведены в разделе 2.

Раздел 3 содержит описание требований к программному решению, раздел 4 — описание архитектуры выбранного решения.

## Глава 2

# Предметная область проекта

Надстройка «расширения для браузера», улучшение текущего представление расписания для студентов НГУ посредством гибкого редактирования (скрытия «лишних» предметов, добавления новых).

## 2.1 Существующие проблемы

У многих студентов НГУ в расписании стоят предметы, которые они не посещают (например, это «предметы по выбору», из которых студент выбрал лишь один) или же нет предметов, на которые они ходят (например, студент может по своему желанию посещать семинары других преподавателей дополнительно к своим).

Решением этого было бы составление университетом индивидуального расписания для каждого из студентов, однако данное решение не реализуется университетом. Поэтому единственным решением в данном случае остаётся создание такой системы, с помощью которой студент сам бы мог редактировать своё расписание, при этом все изменения бы оставались и отображались там, где студент чаще всего смотрит своё расписание.

## 2.2 Предполагаемое решение

Ввиду того что чаще всего студенты смотрят своё расписание через сайт в браузере, решением будет создать «расширение для браузера», работающее в любом современном браузере.

## Глава 3

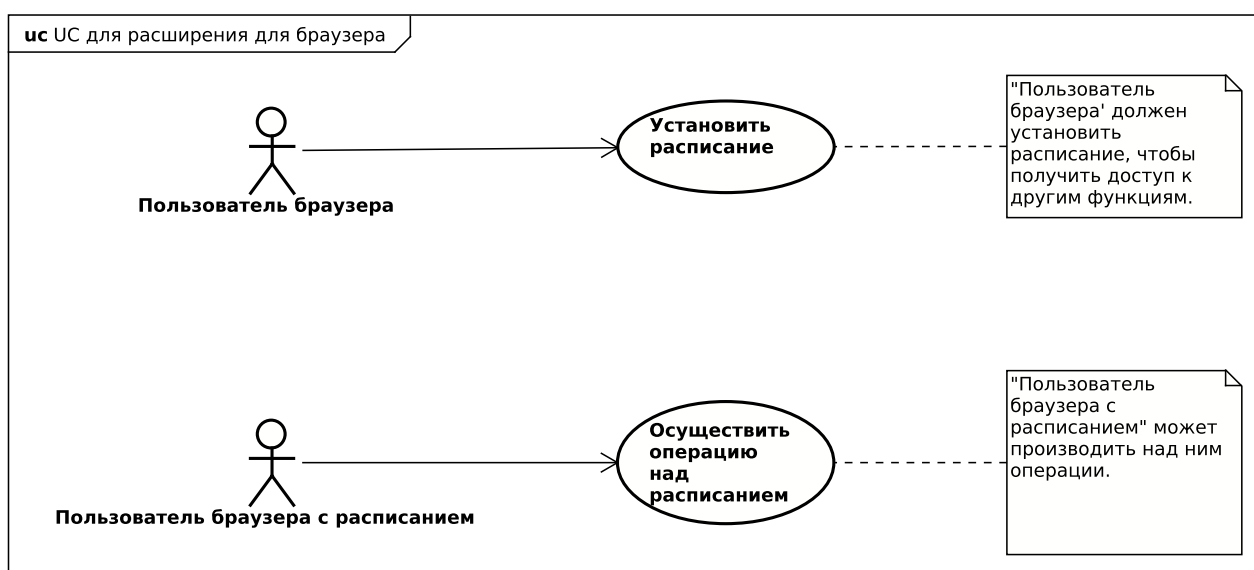
# Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта «Менеджер расписания».

### 3.1 Роли

Роль — это что-то (например: другая система) или кто-то (например: человек) вне системы, которые взаимодействуют с ней. В предлагаемой к разработке системе идентифицированы следующие роли:

1. «*Пользователь браузера*» — человек, который зашёл на сайт с расписанием группы НГУ;
2. «*Пользователь браузера с расписанием*» – пользователь браузера, для которого определено текущее расписание.



## 3.2 Функциональные требования для роли «*Пользователь браузера*»

### 3.2.1 Установить расписание

Как только «*Пользователь браузера*» зашёл на сайт с расписанием группы, текущее расписание устанавливается как расписание этой группы. Однако если по какой-то причине это не удалось сделать, то пользоваться расширением не получится. Если удалось установить расписание, то данное расписание сохраняется в кэш браузера.

Если «*Пользователь браузера*» ранее уже заходил на сайт с расписанием данной группы, то текущее расписание восстановится из кэша браузера.

## 3.3 Функциональные требования для роли «*Пользователь браузера с расписанием*»

### 3.3.1 Осуществить операцию над расписанием

«*Пользователь браузера с расписанием*» может производить над ним операции, а именно он может:

- Экспортировать расписание;
- Изменить расписанием (добавить предмет, удалить предмет, изменить предмет, установить расписание пустым, импортировать расписание).

При этом после любых изменений расписание сохраняется в кэш браузера, заменяя предыдущее, и страница браузера перерисовывается с учётом изменений.

## 3.4 Нефункциональные требования

Клиентские части (расширение для браузера) должны быть совместимы между собой: любой пользователь должен иметь возможность импортировать расписание, экспортированное другим пользователем.



## Глава 4

# Обзор архитектуры

Этот раздел описывает архитектуру проекта.

## 4.1 Подсистемы и компоненты проекта

Если не брать в расчёт компоненты сторонних производителей, то проект состоит из подсистем

- **Server;**
- **Extension;**

## 4.1.1 Server

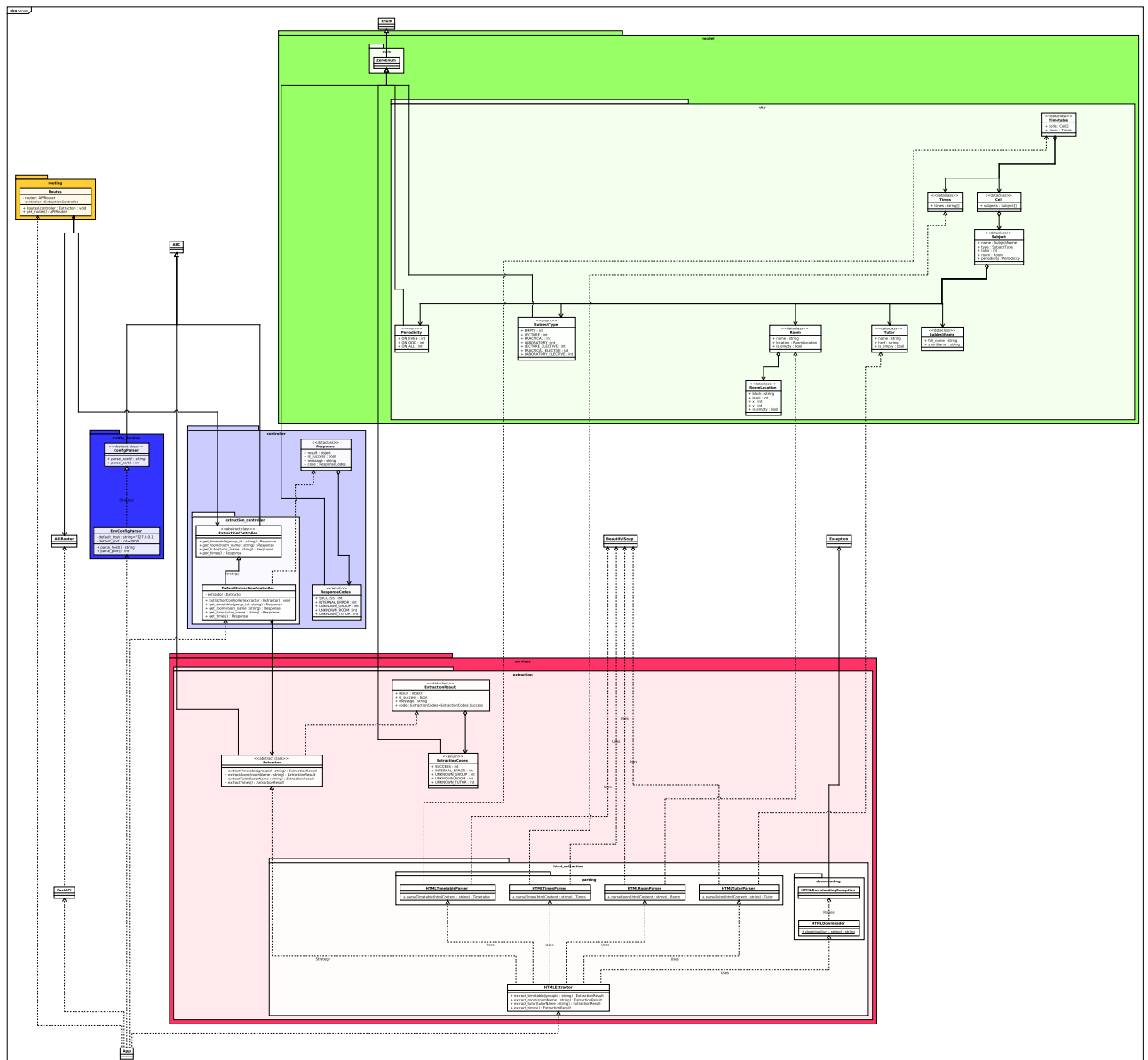


Рис. 4.1: Диаграмма классов **Server**

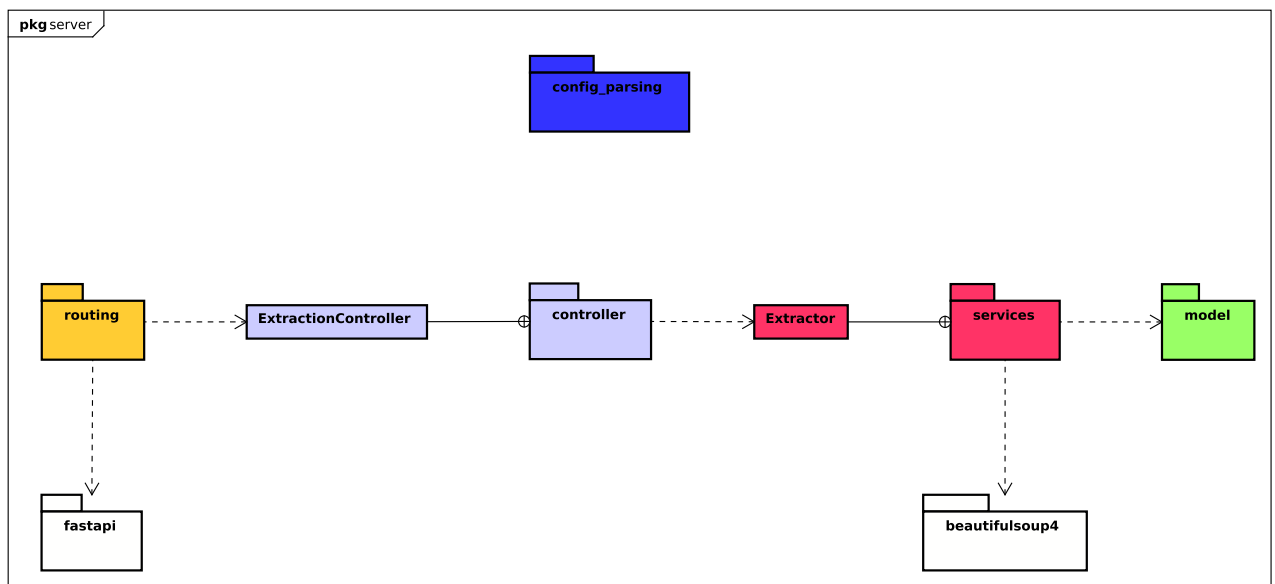


Рис. 4.2: Диаграмма пакетов **Server**

**Server** представляет собой REST-сервер с API, который по запросу парсит сайт с расписанием НГУ и возвращает информацию о расписании группы, аудитории, преподавателе или временах начала пар. Компонента сделан в «Чистой архитектуре»

- *model* — представляет собой слой с основными dto предметной области;
- *services* — представляет собой слой бизнес-логику (манипулирование классами предметной области);
- *controller* — представляет собой слой адаптеров между бизнес-логикой и запросами/ответами сервера;
- *routing* — представляет собой слой, отвечающие за межсетевое взаимодействие (принятие запросов и отправка ответов);
- *config-parsing* — представляет собой пакет, отвечающий за парсинг конфигурации сервера.

## 4.1.2 Extension

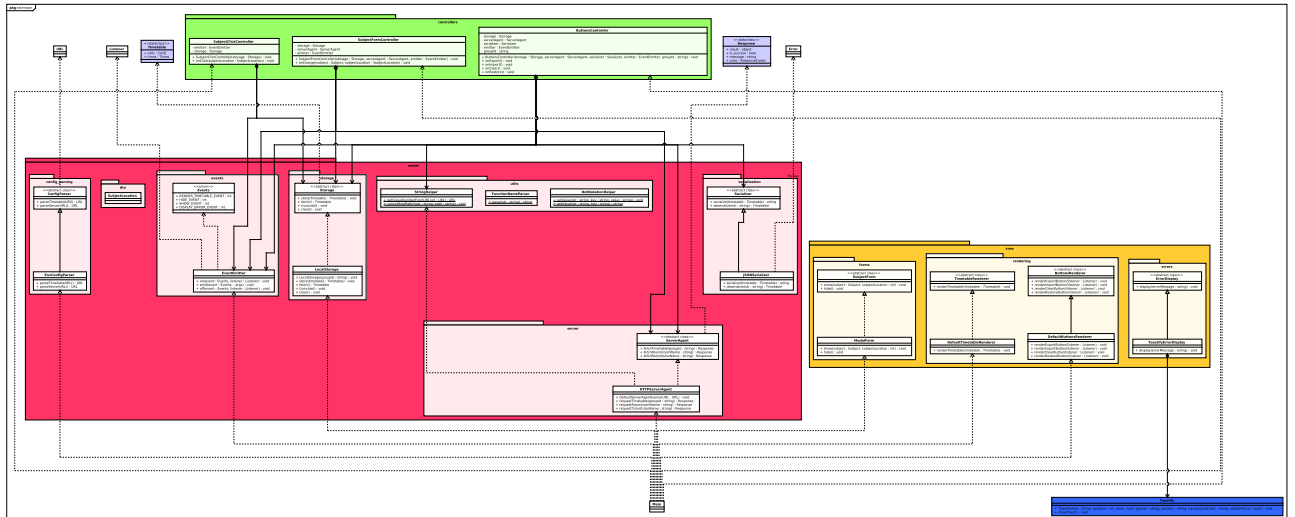


Рис. 4.3: Диаграмма классов **Extension**

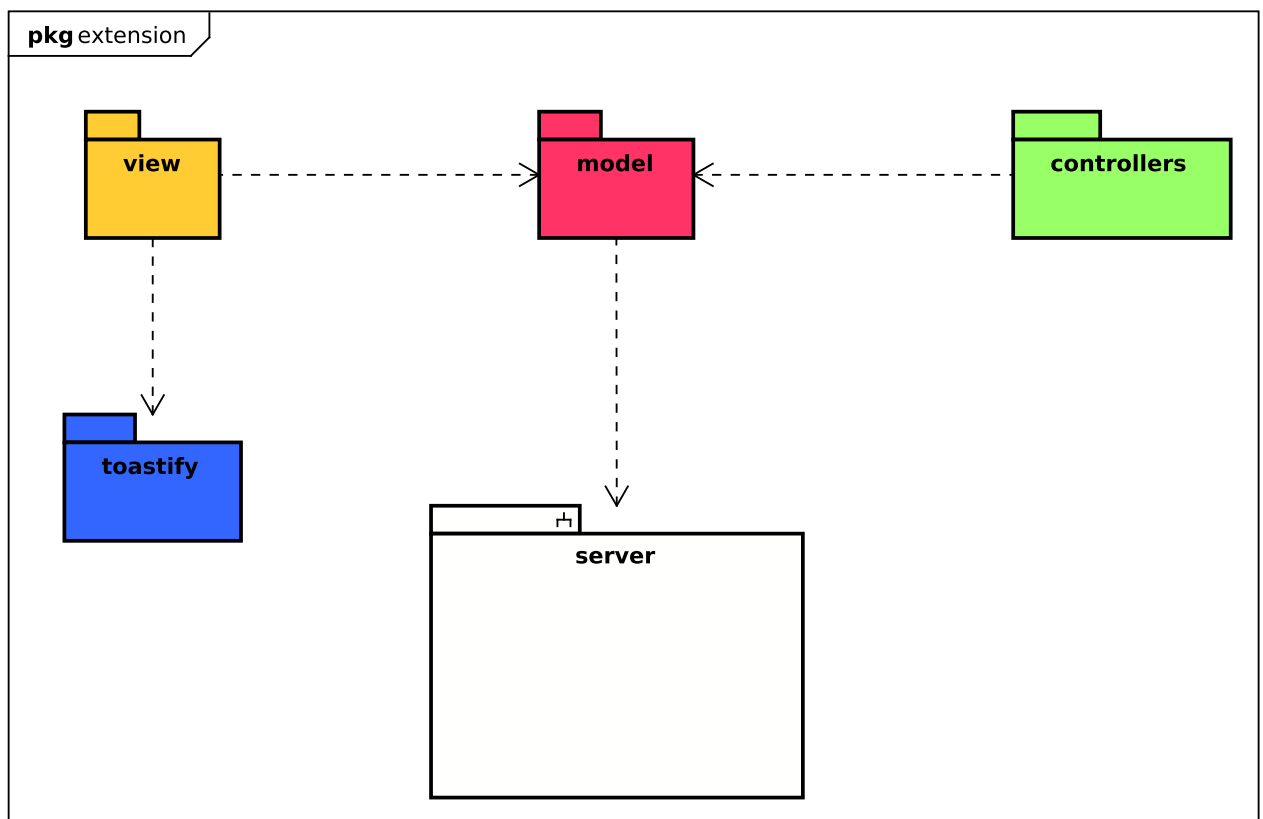


Рис. 4.4: Диаграмма пакетов **Extension**

**Extension** представляет собой расширение для браузеров, являющееся фронтендом проекта. Архитектурно **Extension** реализован с использованием паттерна MVC.

- *model* — представляет собой слой dto и бизнес-логики;
- *view* — представляет собой слой представления, то есть то что пользователь непосредственно видит и с чем взаимодействует (сообщения об ошибках, формы для ввода, ...);
- *controller* — представляет собой слой классов, отвечающих за обработку пользовательских действий (нажатия на кнопки, изменения предметов, ...);

## 4.2 Компоненты сторонних производителей

- **python-dotenv** — библиотека под Python для парсинга файлов .env. Используется в подсистеме **Server** для парсинга начальной конфигурации.
- **beautifulsoup4** — библиотека под Python для парсинга текстовых документов. Используется в подсистеме **Server** для парсинга информации из HTML-документов.
- **fastapi** — библиотека под Python для создания своего API. Используется в подсистеме **Server** для создания конечных точек.
- **Toastify** — библиотека под JS, предоставляющая способ показа различных уведомлений в браузерах. Используется в подсистеме **Extension**.

## 4.3 Схема развёртывания проекта

### Развёртывание Extension

Сборка расширения в файлы bundle.js и manifest.json и дальнейшее добавление в каталог расширений современных браузеров.

### Развёртывание Server

При изменениях в коде GitHub Actions запускает предварительно настроенный workflow. Виртуальная машина на GitHub Actions создает Docker-образы из актуального кода, собранные образы помещаются в Docker Hub. После этого останавливаются и удаляются старые Docker-контейнеры на VPS, а контейнеры с новыми Docker-образами создаются и запускаются, после чего **Server** автоматически перезапускаются.

## Глава 5

# Допущения и ограничения

При разработке проекта были приняты *следующие допущения*

- К серверу не будут обращаться сторонние пользователи сети интернет;
- На сервер не будет совершаться никаких DOS или DDOS-атак в сети интернет;
- Сервер будет работать всегда;
- Вёрстка сайта с расписанием НГУ никогда не будет меняться.

## Глава 6

# Известные проблемы

### 6.1 Отсутствие поддержки JWT

<b>Проблема</b>	Для доступа к данным API сервера не требуется авторизация
<b>Ранг</b>	3 (низкий)
<b>Влияние на проект</b>	Любой пользователь сети интернета может получать данные с сервера, что увеличивает нагрузку на него. Как следствие, расширение для браузера может начать работать медленнее
<b>Пути решения</b>	Добавить поддержку JWT, чтобы доступ к серверу имелся лишь у авторизованных приложений (только у расширения)

### 6.2 Отсутствие защиты от DOS и DDOS-атак

<b>Проблема</b>	У сервера нет никакой защиты от DOS или DDOS-атак, поэтому при любой такой атаке он перестанет обрабатывать запросы расширения
<b>Ранг</b>	5 (средний)
<b>Влияние на проект</b>	При любой DOS или DDOS-атаке, расширения у клиентов перестанут полноценно работать
<b>Пути решения</b>	Использовать интернет сервисы для защиты от DOS и DDOS-атак. Например, CloudFlare

## 6.3 Невозможность работы расширения без сервера

<b>Проблема</b>	Если не работает сервер, то расширение не может полноценно работать
<b>Ранг</b>	7 (Высокий)
<b>Влияние на проект</b>	При критических проблемах на стороне сервера расширение не может полноценно работать
<b>Пути решения</b>	Избавиться от сервера и осуществлять парсинг сайта с расписанием НГУ на клиенте. Однако при добавлении нового клиентского фронтенда (например, мобильного приложения или телеграмм-бота) придётся дублировать логику, что может привести к расхождению спецификации и отсутствию совместимости клиентских фронтендов

## 6.4 Зависимость сервера и расширения от вёрстки сайта с расписанием НГУ

<b>Проблема</b>	Сервер и расширение напрямую зависят от вёрстки сайта с расписанием НГУ
<b>Ранг</b>	8 (Высокий)
<b>Влияние на проект</b>	Если у сайта с расписанием НГУ поменяется вёрстка, то для полноценной работы системы придётся изменять сервер и отображение страниц в расширении
<b>Пути решения</b>	Использовать API, которое используют сайты с расписанием НГУ, однако данное API если и существует, то не является открытым



# Лист регистрации изменений

Дата	Версия	Описание	Автор(ы)
06.09.2023	0.1	Заполнены разделы 1 и 2	Неретин Степан, Кондренко Кирилл
02.10.2023	0.2	Смоделирована usecase-модель	Неретин Степан, Кондренко Кирилл
02.10.2023	0.3	Заполнен раздел 3	Неретин Степан, Кондренко Кирилл
07.10.2023	0.4	Уточнён раздел 3	Кондренко Кирилл
27.10.2023	0.5	Уточнён раздел 3, и смоделирована аналитическая модель	Неретин Степан, Кондренко Кирилл
22.11.2023	0.6	Уточнён раздел 1, смоделирована дизайн-модель и заполнены разделы 4, 5, 6	Неретин Степан, Кондренко Кирилл
06.12.2023	0.7	Реализована серверная часть проекта	Кондренко Кирилл
13.12.2023	1.0	Реализована клиентская часть проекта	Неретин Степан
14.12.2023	1.1	Исправлен баг в серверной части	Кондренко Кирилл
14.12.2023	1.2	Исправлено значительное количество багов в клиентской части	Неретин Степан

# Лист регистрации проверок

Дата	Версия	Описание	Автор