

Техническое описание проекта по курсу ООАД

«Менеджер расписания»

Студенты ФИТ НГУ

Неретин Степан Иванович

Кондренко Кирилл Павлович

Группа 21203

Версия 1.5

Содержание

1	Введение	3
1.1	Цель	3
1.2	Область действия	3
1.3	Определения и сокращения	3
1.4	Ссылки	3
1.5	Краткое описание	3
2	Предметная область проекта	5
2.1	Существующие проблемы	5
2.2	Предполагаемое решение	5
3	Требования к программному решению	6
3.1	Роли	6
3.2	Функциональные требования для роли « <i>Пользователь телеграмм</i> »	7
3.2.1	Указать расписание	7
3.3	Функциональные требования для роли « <i>Пользователь телеграмм с расписанием</i> »	7
3.3.1	Осуществить операцию над расписанием	7
3.4	Функциональные требования для роли « <i>Пользователь браузера</i> »	8
3.4.1	Установить расписание	8
3.5	Функциональные требования для роли « <i>Пользователь браузера с расписанием</i> »	8
3.5.1	Осуществить операцию над расписанием	8
3.6	Нефункциональные требования	8
4	Обзор архитектуры	9
4.1	Компонентная модель системы	9
4.1.1	Компонент 1	9
4.1.2	Компонент 2	9
4.2	Компоненты сторонних производителей	9
4.3	Схема развёртывания приложения	9
5	Допущения и ограничения	10
6	Известные проблемы	11

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 лишь один предмет
Телеграмм-бот	Приложение внутри приложения «Telegram», которое позволяет совершать определённые действия в чате с ним
Расширение для браузера	Компьютерная программа, расширяющая функциональные возможности браузера
API	Описание способов взаимодействия одной компьютерной программы с другими

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 в виде «телеграмм-бота» или «расширения для браузера», улучшение текущего представления 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 и в приложении от университета для телефона, решением будет создать «расширение для браузера», работающее в любом современном браузере, и «телеграмм-бота», которые должны быть взаимозаменяемыми и работать друг с другом.

Глава 3

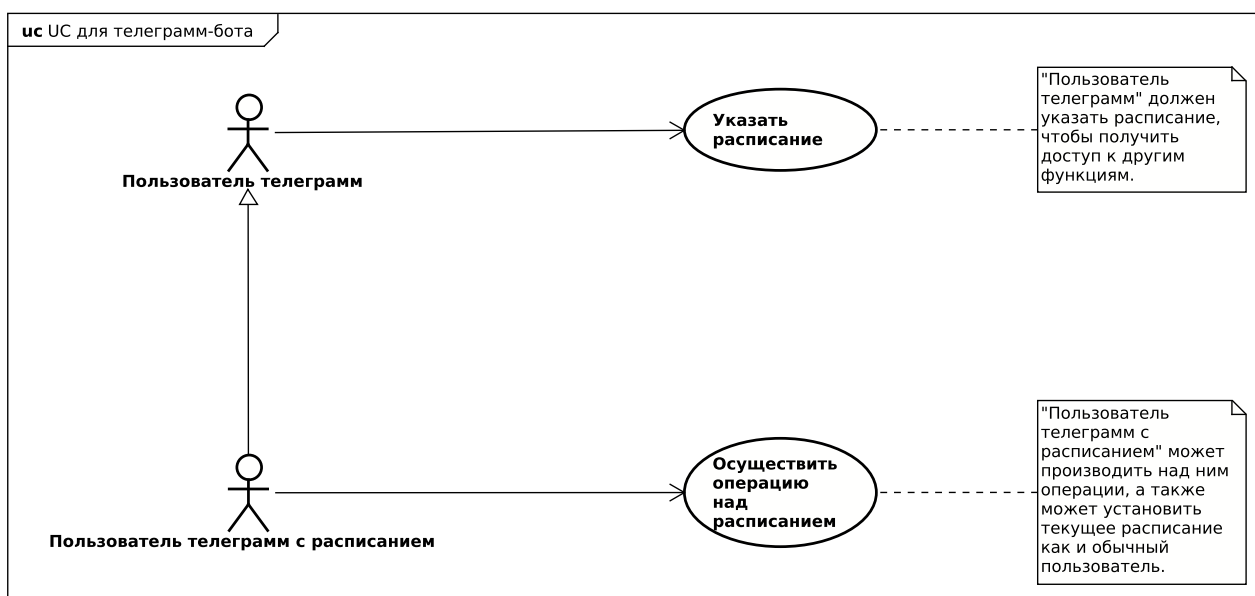
Требования к программному решению

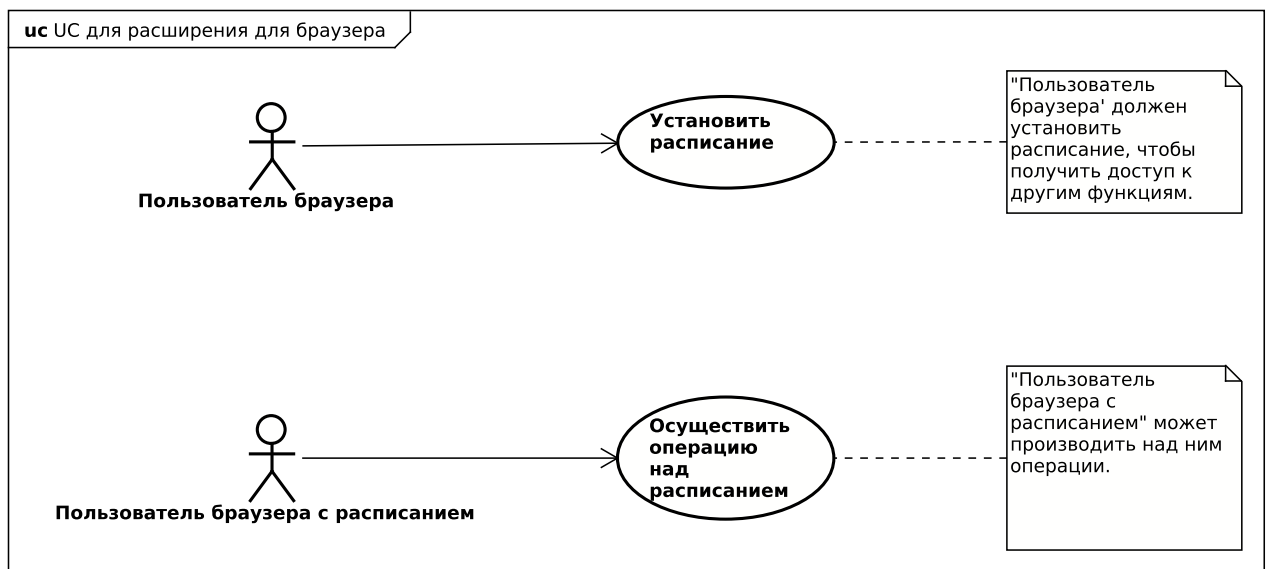
Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта «Менеджер расписания».

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 телеграмм» — человек, добавивший к себе бота в телеграмм;
2. «Пользователь телеграмм с расписанием» — пользователь телеграмм, указавший текущее расписание;
3. «Пользователь браузера» — человек, который зашёл на сайт с расписанием группы НГУ;
4. «Пользователь браузера с расписанием» — пользователь браузера, для которого определено текущее расписание.





3.2 Функциональные требования для роли «Пользователь телеграмм»

3.2.1 Указать расписание

«Пользователь телеграмм» должен указать расписание, чтобы получить доступ к другим функциям. Он может сделать несколькими способами:

- Импортировать расписание из файла с расписанием (подготовленного ранее с помощью телеграмм-бота или расширения для браузера);
- Указать номер группы;
- Оставить расписание пустым.

3.3 Функциональные требования для роли «Пользователь телеграмм с расписанием»

3.3.1 Осуществить операцию над расписанием

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

- Показать расписание;
- Экспортировать расписание;
- Изменить расписание (добавить предмет, удалить предмет, изменить предмет).

«Пользователь телеграмм с расписанием» также может установить текущее расписание как и «Пользователь телеграмм».

3.4 Функциональные требования для роли «*Пользователь браузера*»

3.4.1 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.

3.5 Функциональные требования для роли «*Пользователь браузера с расписанием*»

3.5.1 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).

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

3.6 Нефункциональные требования

Проект состоит из двух частей — телеграмм-бота и расширения для браузера, однако они должны иметь схожий пользовательский интерфейс и быть совместимы между собой, чтобы, например, можно было импортировать расписание, которое было экспортировано в другой части проекта.

Глава 4

Обзор архитектуры

4.1 Компонентная модель системы

4.1.1 Компонент 1

4.1.2 Компонент 2

4.2 Компоненты сторонних производителей

4.3 Схема развёртывания приложения

Глава 5

Допущения и ограничения

Глава 6

Известные проблемы

Лист регистрации изменений

Дата	Версия	Описание	Автор
06.09.2023	1.0	Заполнены разделы 1 и 2	Неретин Степан, Кондренко Кирилл
30.09.2023	1.1	Смоделирована usecase-модель телеграмм-бота	Кондренко Кирилл
02.10.2023	1.2	Смоделирована usecase-модель расширения для браузера	Неретин Степан, Кондренко Кирилл
02.10.2023	1.3	Заполнен раздел 3	Неретин Степан, Кондренко Кирилл
07.10.2023	1.4	Уточнён раздел 3	Кондренко Кирилл
22.10.2023	1.5	Уточнён раздел 3, и смоделирована аналитическая модель для телеграмм-бота	Кондренко Кирилл

Лист регистрации проверок

Дата	Версия	Описание	Автор