

**Техническое описание проекта
по курсу ООАД**

«Менеджер расписания»

**Студенты ФИТ НГУ
Неретин Степан Иванович
Кондренко Кирилл Павлович
Группа 21203**

Версия 1.0

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	4
2.1	Существующие проблемы	4
2.2	Предполагаемое решение	4
3	Требования к программному решению	5
3.1	Роли	5
3.2	Функциональные требования для роли Роль1	5
3.2.1	<Use Case Name 1>	5
3.2.2	<Use Case Name 2>	5
3.3	Функциональные требования для роли Роль2	5
3.3.1	<Use Case Name 1>	5
3.3.2	<Use Case Name 2>	5
3.4	Нефункциональные требования	5
4	Обзор архитектуры	6
4.1	Компонентная модель системы	6
4.1.1	Компонент 1	6
4.1.2	Компонент 2	6
4.2	Компоненты сторонних производителей	6
4.3	Схема развёртывания приложения	6
5	Допущения и ограничения	7
6	Известные проблемы	8

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

1.4 Ссылки

1.5 Краткое описание

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, работающее в любом современном браузере, и телеграмм-бота, которые должны быть взаимозаменяемыми и работать друг с другом.

Глава 3

Требования к программному решению

3.1 Роли

3.2 Функциональные требования для роли Роль1

3.2.1 <Use Case Name 1>

3.2.2 <Use Case Name 2>

3.3 Функциональные требования для роли Роль2

3.3.1 <Use Case Name 1>

3.3.2 <Use Case Name 2>

3.4 Нефункциональные требования

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

Лист регистрации проверок

Дата	Версия	Описание	Автор