

Новосибирский Государственный Университет  
Факультет Информационных Технологий

# **Техническое описание проекта по курсу ООАД**

**Komugi**

**Студенты ФИТ НГУ  
Бородин А.М.  
Пронина В.А.  
группа 19205**

**Версия 2.2.8**

## Содержание

Введение	3
Цель	3
Область действия	3
Определения и сокращения	3
Ссылки	3
Краткое описание	3
Предметная область проекта	3
Существующие проблемы	4
Предполагаемое решение	4
<b>Требования к программному решению</b>	<b>4</b>
Роли	4
Функциональные требования для роли “Пользователь”	4
Авторизация	4
Функциональные требования для роли “Авторизованный пользователь”	4
Оповещение	4
Статистика	4
Нефункциональные требования	4
<b>Обзор архитектуры</b>	<b>5</b>
Компонентная модель системы	5
Компонент 1	5
Компонент 2	5
Компоненты сторонних производителей	5
Схема развертывания приложения	5
<b>Допущения и ограничения</b>	<b>5</b>
<b>Известные проблемы</b>	<b>5</b>
Невысокая производительность приложения	6

# Техническое описание проекта по курсу ООАД

## 1. Введение

### 1.1 Цель

Данный документ представляет собой техническое описание проекта *Kotugi* 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 *Kotugi* на основе стандартного шаблона *Inteks SEP* и предназначен для использования студентами ФИТ и преподавателями курса ООАД.

### 1.3 Определения и сокращения

Таблица 1: Определения и сокращения

Термин	Описание
<i>Time management</i>	Технология организации времени и повышения эффективности его использования
<i>Deadline</i>	Крайний срок, к которому должна быть выполнена задача

### 1.4 Ссылки

В тексте содержатся ссылки на следующие документы:

[1] <Имя файла документа>, v<версия> - <описание документа>

Ссылки приводятся в виде [N], где N – номер документа в вышеприведенном списке.

### 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.

## 2. Предметная область проекта

*Тайм-менеджмент* позволяет достигать поставленных целей и является неотъемлемой частью успешной учёбы. Рациональное использование своего времени – отправная точка для продуктивности и возможности быть успевающим студентом. Учёт посещаемости и выполнения заданий дает наглядное представление об успеваемости студента.

### 2.1 Существующие проблемы

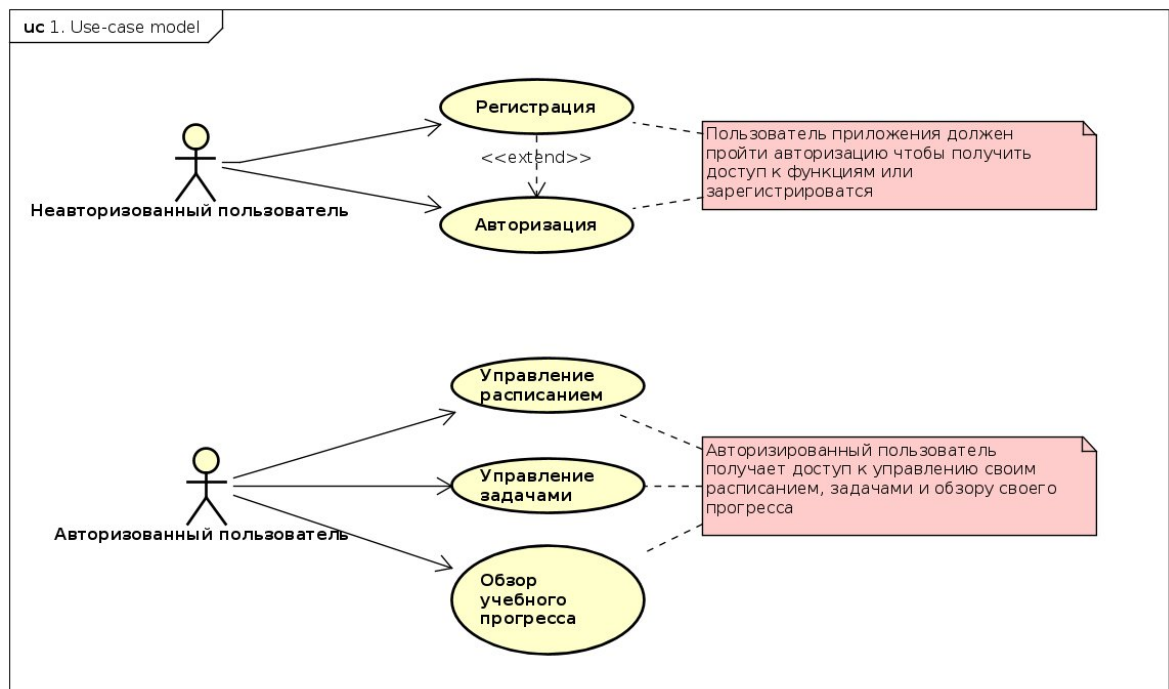
Существующие приложения, которые осуществляют контроль выполненных задач и времени на их исполнение, не реализуют всю желаемую функциональность, либо делают это в неудобном формате. Отсутствует возможность отслеживать свой прогресс, выполненных заданий за курс, а также посещаемость лекций/семинаров, нет конечной статистики. Решение проблемы можно осуществить с помощью разработки эффективного и удобного для использования студентом интерфейса.

### 2.2 Предполагаемое решение

Необходимо оповещать пользователя о каждой из поставленных задач, разработать дизайн, позволяющий наглядно продемонстрировать прогресс и посещение учебных занятий, а также ввести статистику выполненных/просроченных задач.

## 3. Требования к программному решению

Данный раздел описывает требования к программной системе, разрабатываемой в рамках проекта *Котиги*.



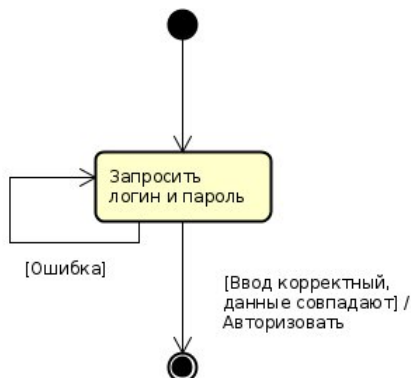
### 3.1 Роли

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:

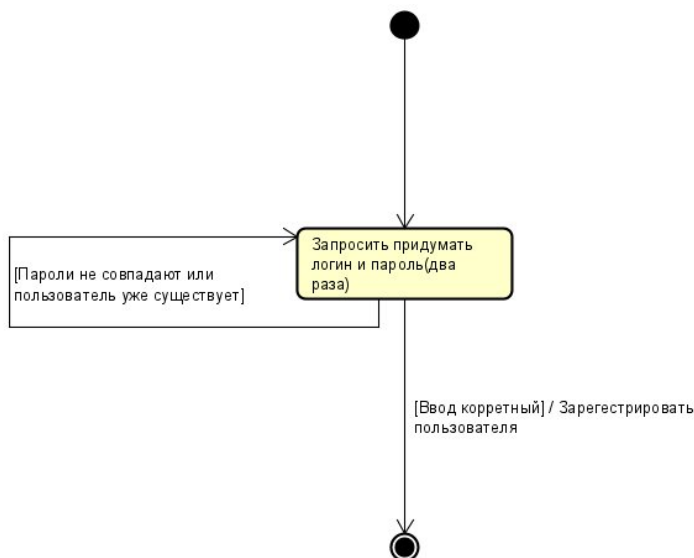
1. Неавторизованный пользователь – имеет возможность создать свой профиль или войти в уже существующий.
2. Авторизованный пользователь – использует функциональность приложения.

### 3.2 Функциональные требования для роли “Неавторизованный пользователь”

- Авторизация - предоставление доступа к профилю и всем имеющимся в нем данным.

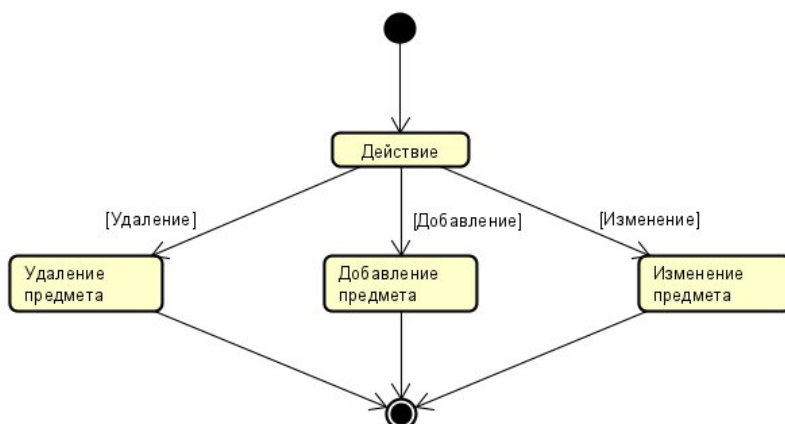


- Регистрация - создание нового профиля.

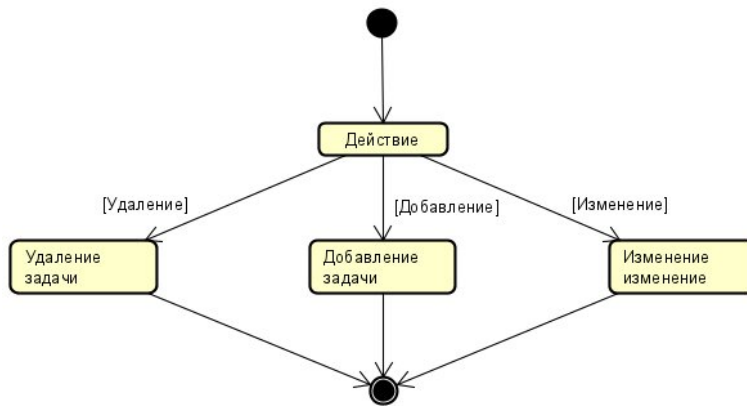


### 3.3 Функциональные требования для роли “Авторизованный пользователь”

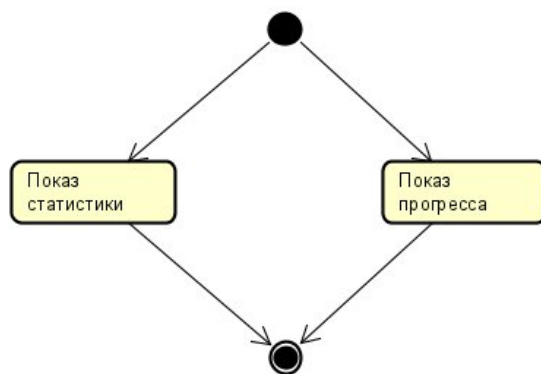
- Управление расписанием - получение информации о расписании, его изменение или удаление.



- Управление задачами - получение информации о задачах, их изменение или удаление.



- Обзор учебного прогресса - получение статистики по выполненным задачам, посещенным занятиям.



### 3.4 Нефункциональные требования

**Производительность:**

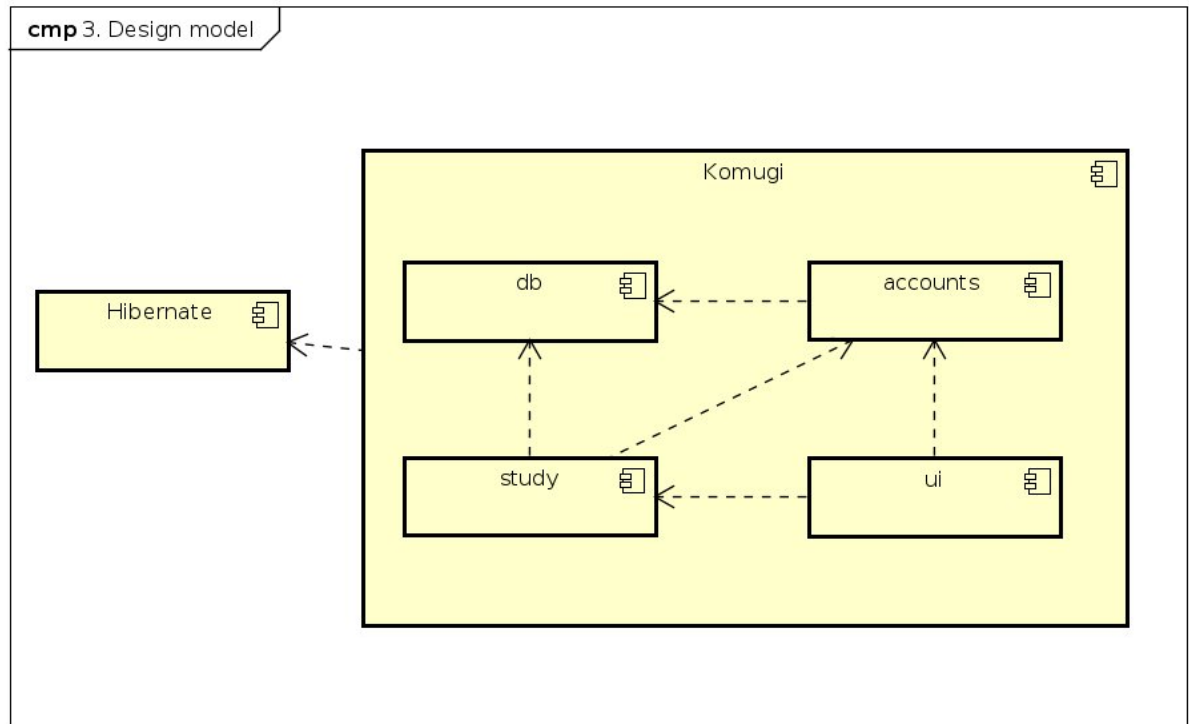
Требование к малым задержкам по времени на любую транзакцию.

**Поддерживаемость:**

Возможность использовать приложение в большинстве современных браузеров.

## 4. Обзор архитектуры

Этот раздел описывает архитектуру системы.



### 4.1.1 Компонентная модель системы

#### 4.1.1.1 UI

Этот компонент содержит классы, реализующие пользовательский интерфейс, а именно способы просмотра событий/меню и переключения между ними.

#### 4.1.1.2 Study

Этот компонент содержит классы, реализующие сбор данных и его обработку для высчитывания статистики, а также работа с данными конкретного пользователя.

#### 4.1.1.3 Accounts

Этот компонент содержит классы, реализующие авторизацию и регистрацию пользователя в приложении.

#### 4.1.1.4 DB

Этот компонент содержит классы, реализующие исполнение запросов в базе данных.

### 4.1.2 Компоненты сторонних производителей

#### 4.1.2.1 Hibernate

Хранение и управление данными.

### 4.1.3 Схема развертывания приложения

[Здесь приводится Deployment diagram - диаграмма развертывания системы, со связями между узлами и указанием способа связи (протокола). На диаграмме обязательно указать, какие компоненты находятся на том или ином узле]

## 5. Допущения и ограничения

[Краткое описание допущений, которые подразумевает данный проект, и любых ограничений (например, по бюджету, участникам, требуемому оборудованию, срокам и т.п.), накладываемых на его выполнение.]

Пример: При разработке проекта принято допущение, что число транзакций в единицу времени значительно (более чем в 10 раз) снижается в ночное время, что позволяет в период с 01:00 до 6:00 производить автоматическое обновление программного обеспечения системы, требующее полной перезагрузки и остановки сервиса на период до 5 минут.



## 6. Известные проблемы

Ниже приводятся известные на данный момент проблемы и недоработки выработанного программного решения, а также возможные пути их устранения в последующих итерациях проекта.

### 6.1 Невысокая производительность приложения

Проблема	Производительность приложения экспоненциально деградирует при общем числе пользователей выше 10000 и числе одновременных сессий выше 100.
Ранг	10 (высокий)
Влияние на проект	Невозможность использования системы при числе пользователей более 10000.
Пути решения	Кластеризация веб-сервера и сервера базы данных, а также применение load balancer в точке маршрутизации запроса к веб-серверу.

## Лист регистрации изменений

Дата	Версия	Описание	Автор
12.09.2021	1.0.0	Заполнен Vision проекта, расписана роль и функциональные требования к ней	Пронина В.А.
16.09.2021	1.1.0	Изменение Vision и функциональных требований	Пронина В.А.
05.10.2021	1.4.2	Завершена 3 секция и начато заполнение 4 секции.	Бородин А.М.

## Лист регистрации проверок

Дата	Версия	Описание	Автор
16.09.2021	1.1.0	Приняты Vision и роли	Андреев А.В.
7.10.2021	1.4.2	Приняты Анализ и Дизайн	Андреев А.В.
2.12.2021	2.2.8	Принята реализация	Андреев А.В.