**Техническое задание**

**На разработку приложения**

**для онлайн-курсов**

**ФАКУЛЬТАТИВ**

**Подготовили Захарина Екатерина и Логаш Полина**

**ММФ БГУ 2022**

# 

# Оглавление

[Оглавление](#_a1taoerqaniy)

[1.Общие сведения](#_7v64kqb5p2or)

[2.Назначение системы](#_35ul2bi9s4t3)

[3.Требования к системе](#_olatc4y2fknb)

[3.1 Функциональные требования](#_4gg0ts2ka7kc)

[3.1.1 Описание пользователей системы](#_jk1ktlc91tsd)

[3.1.2 Описание вариантов использования](#_c3nt12g1w9fn)

[3.1.3 Варианты использования](#_3tsf6vkv9df2)

[3.2. Технические требования](#_g98osu4d5gy1)

[3.2.1 Архитектура системы](#_v8jqjoel3phm)

[3.2.3 Требования к пользовательскому интерфейсу](#_2kge6nwgwg56)

[4. Порядок приема системы](#_cx2j089otomw)

[4.1 Приемочные тесты](#_qjtdikmo2elh)

[4.2 Сроки исполнения](#_m9myrfrd4bui)

# 1.Общие сведения

Онлайн-курсы – приложение, с помощью которого студенты могут проходить различные курсы, а преподаватели, соответственно, создавать и вести данные курсы.

Для зачисления на курс студент должен предварительно зарегистрироваться и подать заявку на прохождение желаемого курса. Даже до регистрации доступна информация обо всех курсах. После одобрения заявки преподавателем, студент считается зачисленным курс. По мере прохождения курса ученик получает оценки от преподавателя, которые впоследствии можно будет просматривать в своем табеле.

Преподаватели после регистрации могут создавать курсы, добавлять материалы на свои курсы, одобрять заявки студентов на свои курсы, просматривать работы своих студентов, ставить им оценки и и по окончании прохождения ими курса писать на студентов отзывы.

Администраторы могут просматривать списки курсов, студентов и преподавателей, выдавать им ограничения действий на сайте на разный срок и одобрять заявки преподавателей на добавление новых курсов.

# 2.Назначение системы

Данная система создана для **онлайн** обучения

1. Организации это дает возможность не платить за аренду помещений для курсов.
2. Студентам это даёт возможность обучения любым желаемым предметам из любого места.
3. Для преподавателей также является удобством вести занятия из любой места, также просмотр выполненных заданий студентов, выставление им оценок и просмотр их табелей в разы удобнее, так как всё это осуществляется в электронном варианте, а не в бумажном.

# 3.Требования к системе

## 3.1 Функциональные требования

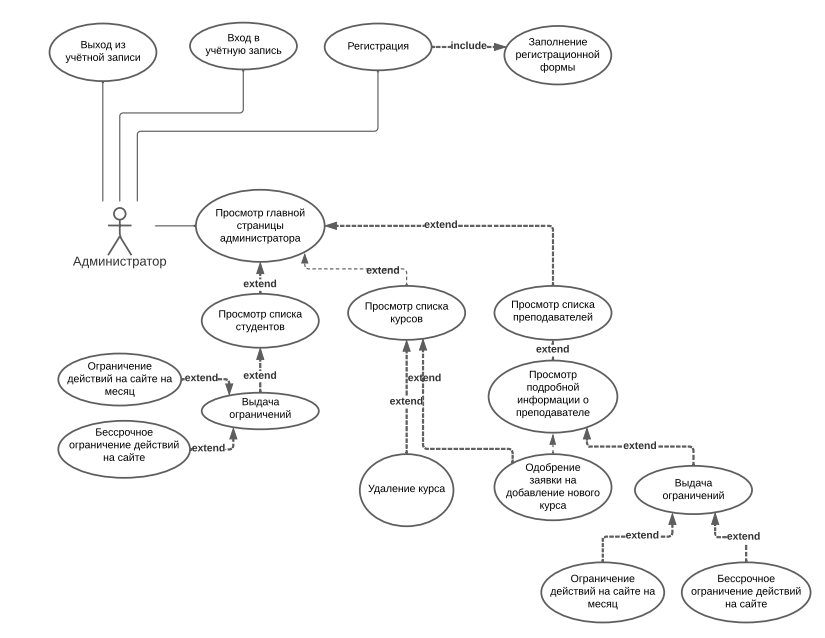
### 3.1.1 Описание пользователей системы

| **Актер** | **Описание** |
| --- | --- |
| Неавторизованный пользователь | Неавторизованный пользователь   * регистрация * авторизация |
| Студент | Авторизованный пользователь   * учится на курсе * выполняет задания * подает заявки на зачисление на курс |
| Преподаватель | Авторизованный пользователь   * подает заявку на создание нового курса * одобряет заявки на поступление на курс * добавляет и проверяет задания * пишет отзывы на студента |
| Администратор | Авторизованный пользователь   * выдает ограничения * одобряет заявку на создание нового курса |

### 

### 3.1.2 Описание вариантов использования

##### Рисунок 1 - студент и преподаватель



##### Рисунок 2 - администратор

На рисунках 1 - 2 приведены UseCase диаграммы вариантов использования нашего приложения студентом, преподавателем и администратором.

### 3.1.3 Таблицы вариантов использования

**Регистрация**

| Actor | Незарегистрированный пользователь |
| --- | --- |
| Goal in context | Зарегистрироваться |
| Trigger event | Необходимо зарегистрироваться |
| Pre-condition | Нет |
| Success end condition | Новый пользователь зарегистрирован |
| Failed end condition | В регистрации отказано с указанием причин |

#### 

#### 

#### Основной сценарий регистрации

| **Шаг** | **Актер** | **Описание действия** |
| --- | --- | --- |
| 1 | Незарегистрированный пользователь | Заходит на сайт |
| 2 | Сайт | Выдаёт регистрационную форму |
| 3 | Незарегистрированный пользователь | Вводит следующую информацию:  1. Логин  2. Пароль  3. Повтор пароля  4. Адрес электронной почты  5. Вводит капчу  6. Нажимает на кнопку зарегистрироваться |
| 4 | Сайт | Проверяет уникальность логина и совпадение паролей. В случае успешной проверки, регистрирует нового пользователя на сайте |
| 5 | Сайт | Отсылает на адрес электронной почты сообщение с запросом о подтверждении регистрации. Возвращает Главную форму зарегистрированного пользователя. |

**Альтернативный сценарий регистрации**

| № | Variable | Possible variations |
| --- | --- | --- |
| 1 | Обязательные поля формы не заполнены. | Приложение сообщает (подчёркивает красным), какое поле заполнено неверно и предлагает его заполнить. |
| 2 | Поля формы заполнены неверно. | Приложение сообщает (подчёркивает красным), какое поле заполнено неверно и предлагает его заменить.  Причины:  1. Адрес электронной почты имеет неправильный формат.  2. Несуществующий адрес электронной почты.  3. Неверно записана капча. |
| 3 | Пароль и повтор пароля не совпадают. | Приложение сообщает (подсвечивает красным) об ошибке при наборе пароля. |
| 4 | Введенный логин пользователя уже существует. | Приложение информирует, что такой логин уже существует и предлагает ввести другой. |

#### Добавление заявки на прохождения курса

| Actor | Студент |
| --- | --- |
| Goal in context | Добавиться на курс |
| Trigger event | Желание проходить курс |
| Pre-condition | Подать заявку |
| Success end condition | Получит доступ к материалам курса |
| Failed end condition | Получит отказ с указанием причин |

#### 

#### 

#### Основной сценарий добавления заявки на прохождение курса (для студента)

| **Шаг** | **Актер** | **Описание действия** |
| --- | --- | --- |
| 1 | Студент | На странице курса нажимает кнопку подать заявку “Подать заявку” |
| 2 | Система | Отсылает заявку преподавателю |
| 3 | Преподаватель | Просматривает заявку, одобряет заявку, нажав кнопку “Одобрить” |
| 4 | Система | Добавляет пользователя на курс |
| 5 | Система | Отображает студенту, что заявка принята |

**Альтернативный сценарий добавления заявки на прохождение курса (для студента)**

| № | Variable | Possible variations |
| --- | --- | --- |
| 1 | Преподаватель | По некоторым причинам преподаватель не хочет или не может добавить студента на курс и после просмотра заявки нажимает кнопку “Отклонить” |
| 2 | Система | Отображает студенту, что заявка отклонена |

#### Добавление заявки на добавление курса (для преподавателя)

| Actor | Преподаватель |
| --- | --- |
| Goal in context | Стать преподавателем нового курса |
| Trigger event | Необходимо получить одобрение заявки на добавление нового курса |
| Pre-condition | Подать заявку на добавление нового курса |
| Success end condition | Добавлен новый курс с данным преподавателем в качестве куратора |
| Failed end condition | Сообщение с текстом об отказе в добавлении нового курса |

#### Основной сценарий заявки на добавление нового курса

| **Шаг** | **Актер** | **Описание действия** |
| --- | --- | --- |
| 1 | Преподаватель | На странице со списком курсов нажать “Добавить новый курс” |
| 2 | Система | Отсылает заявку администратору |
| 3 | Администратор | Просматривает заявку, одобряет заявку, нажав кнопку “Одобрить” |
| 4 | Система | Отображает преподавателю, что заявка принята |

**Альтернативный сценарий добавление заявки на добавление курса**

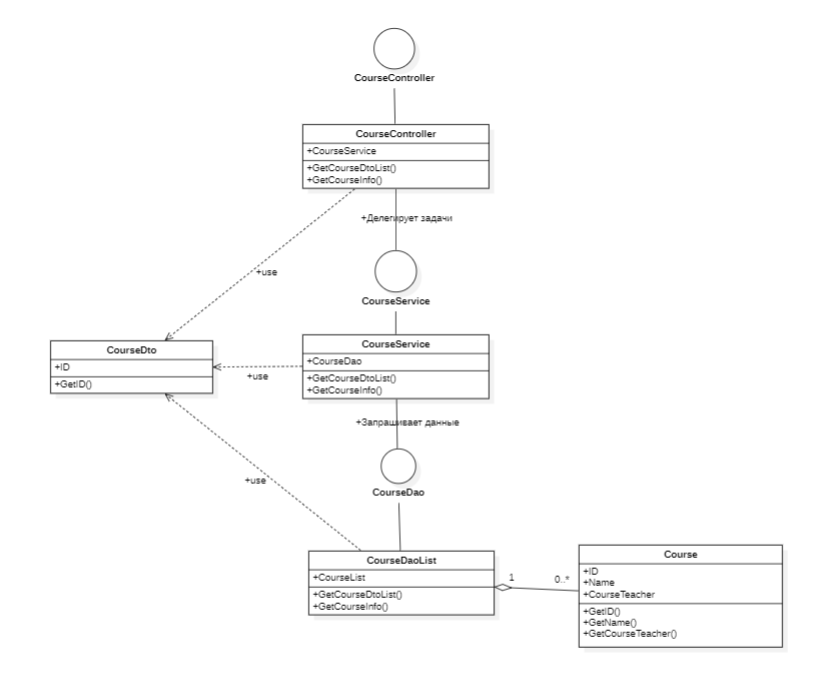
| № | Variable | Possible variations |
| --- | --- | --- |
| 1 | Администратор | По некоторым причинам преподаватель не хочет или не может добавить новый курс и после просмотра заявки нажимает кнопку “Отклонить” |
| 2 | Система | Отображает подавателю, что заявка отклонена |

## 3.2. Технические требования

### 3.2.1 Архитектура системы

##### 

##### Рисунок 3 - диаграмма классов 1



##### Рисунок 4 - диаграмма классов 2

##### 

##### Рисунок 5 - диаграмма последовательностей

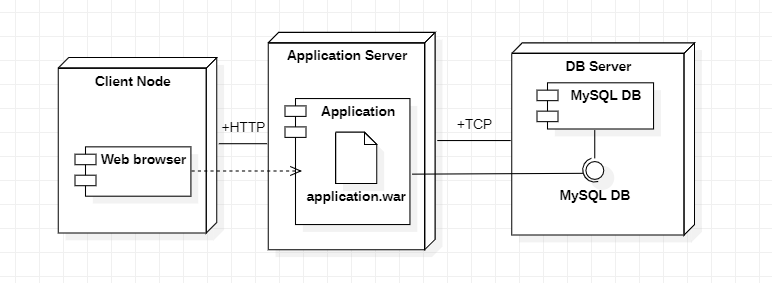
На рисунке 4 приведена диаграмма последовательности при одобрении заявки студента на добавление на курс. Преподаватель одобряет заявку и отправляет ее в CourseController, который далее передает ее в CourseService, а тот в CourseDao. CourseDao связывается с удаленной базой данных и отправляет туда данные об одобренной заявке и возвращает результат (успешно/неуспешно), который передается через все стадии к администратору и преподавателю, создавшему заявку.

##### Рисунок 6 - диаграмма деятельности 1

##### *Рисунок* 7 *- диаграмма деятельности* 2

##### *Рисунок* 8 *- диаграмма* состояний

##### *Рисунок* 9 *- диаграмма компонентов*

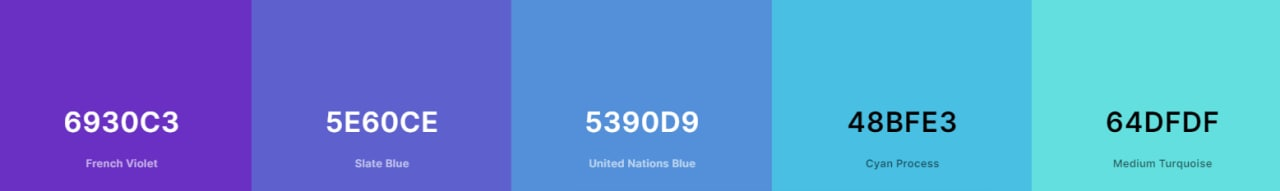


##### Рисунок 10 - диаграмма развертывания

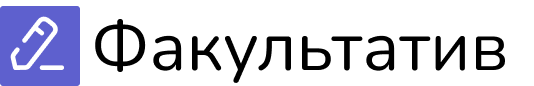
### 

### 3.2.3 Требования к пользовательскому интерфейсу

Дизайн приложения должен быть выполнен с использованием синих и фиолетовых оттенков, помимо стандартных белого и черного. Логотипом выступает название сервиса.



##### Рисунок 11 - цветовая палитра

**

##### Рисунок 12 - логотип

##### 

##### Рисунок 13 - стартовый загрузочный экран

##### 

##### Рисунок 14 - экран авторизации

* приложение должно одинаково хорошо отображаться и выглядеть на экранах с разными разрешениями. Минимальная ширина страницы, при которой проверяется корректность отображения приложения - 360 рх, максимальная ширина страницы, при которой проверяется корректность отображения приложения - 1920 рх.
* интерактивность элементов, с которыми пользователи могут взаимодействовать, изменение внешнего вида самого элемента и состояния курсора при наведении, использование разных стилей для активного и неактивного состояния элемента
* обязательное требование к интерактивности: плавное изменение внешнего вида элемента при наведении и клике не влияющее на соседние элементы
* читабельность текста: минимальный размер шрифта на любом разрешении экрана не менее 14 рх, достаточная контрастность цвета фона и цвета шрифта в активном и неактивном состоянии.

**3.2.3 Нефункциональные требования**

**OC:**

Microsoft Windows 7, 8, 8.1, 10, 11

Linux

MacOS 10

**Требуемый язык программирования:** Java

**Для хранения данных:** должен использоваться PostgreSQL

**Для аналитики -** PostgreSQL Analytics

**Требования к безопасности:**

- разграничить роли администратора, студента и преподавателя

- все данные о пользователе должны шифроваться

- исходный код приложения приложения должен шифроваться

- валидация данных форм с уведомлением пользователя об ошибках заполнения в удобном формате

- осуществление резервного копирования файлов

- все пароли баз данных должны храниться только в шифрованном виде

**Требования к надежности:** Желательно иметь автоматическую систему сохранения баз данных. В случае, если хостинг не отвечает, грамотно обрабатывать ошибку и выдавать сообщение о том, что база данных недоступна.

**Требования к производительности:** Приложение должно функционировать на следующих браузерах:

* Yandex Browser 21
* Safari 5
* Opera 76
* Chrome 90
* Mozilla Firefox 100
* Internet Explorer 11

# 

# 4. Порядок приема системы

## 4.1 Приемочные тесты

**Тест кейс: Одобрение заявки на курс**

Pre-condition: быть авторизованным в приложении как администратор

1. Зайти в приложение **→** должна появиться панель администратора
2. Перейти на страницу заявок на одобрение на добавление курса **→** должна открыться форма заполнения заявки
3. Нажать на одну из заявок **→** Должно появиться подробная информация о заявке и две кнопки “Одобрить” и “Отклонить”
4. Нажать на кнопку “Одобрить”

Ожидаемый результат: администратор одобрит заявку на добавление нового курса, новый курс появится на сайте, преподавателю, добавившему заявку, придёт оповещение о том, что его курс одобрен.

**Тест кейс: Проверка домашнего задания**

Pre-condition: быть авторизованным в системе как преподаватель

1. Зайти в приложение → должна появиться главная страница
2. Нажать на кнопку “Мои курсы” → должна появиться страница с курсами, которые ведет данный преподаватель
3. Нажать на название одного из курсов **→** должна появиться страница с подробной информацией о курсе: материалы курса, список студентов, проходящих курс и домашние задания студентов.
4. Нажать на кнопку “Домашние задания” →должен появиться список с названиями домашних работ
5. Нажать на название одного из домашних заданий → должен появиться таблица со столбцом студентов, которые должны выполнить данное домашнее задание, столбцом с прикрепленными файлами работ студентов, столбцом с оценками за задание и столбцом с комментариями к заданию каждого из студентов.
6. Нажать на один из прикрепленных файлов → должна открыться новая страница с файлом или файл должен скачаться на устройство преподавателя

Ожидаемый результат:

## 

## 

## 4.2 Сроки исполнения

| **№** | **Название этапа** | **Сроки этапа** |
| --- | --- | --- |
| 1 | Изучение предметной области. Написание плана разработки | 14.02.2022 - 28.02.2022 |
| 2 | Проектирование системы | 01.03.2022 - 12.03.2022 |
| 3 | Разработка базы данных | 13.03.2022 - 23.03.2022 |
| 4 | Написание серверной части проекта (запросов, обработка запросов и т.п.) | 24.03.2022 - 24.04.2022 |
| 5 | Отрисовка дизайна проекта и создание UI части | 25.04.2022 - 20.05.2022 |