МИНОБРНАУКИ РОССИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук

Кафедра программирования и информационных технологий

Техническое задание

на разработку мобильного приложения

«KnitWit»

Исполнители

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Павлов А.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Акимушкин Е.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Перепечко К.К.

Заказчик

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тарасов В.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Проскуряков Е.Д.

Воронеж 2024

[1 Терминология 4](#_Toc161136312)

[2 Общие сведения 5](#_Toc161136313)

[2.1 Полное наименование системы и название приложения 5](#_Toc161136314)

[2.2 Заказчик 5](#_Toc161136315)

[2.3 Разработчик 5](#_Toc161136316)

[2.4 Перечень документов, на основании которых создается приложение 5](#_Toc161136317)

[2.5 Плановые сроки начала и окончания работы по созданию приложения 6](#_Toc161136318)

[2.6 Порядок оформления и предъявления заказчику результатов работ по созданию приложения 6](#_Toc161136319)

[3 Назначение и цели создания приложения 8](#_Toc161136320)

[3.1 Назначение приложения 8](#_Toc161136321)

[3.2 Цели создания приложения 8](#_Toc161136322)

[3.3 Задачи приложения 8](#_Toc161136323)

[4 Требования к приложению 9](#_Toc161136324)

[4.1 Требования к структуре приложения в целом 9](#_Toc161136325)

[4.1.1 Требования к реализации 9](#_Toc161136326)

[4.1.2 Перспективы развития, модернизации приложения 10](#_Toc161136327)

[4.2 Функциональные требования к приложению 10](#_Toc161136328)

[4.3 Требования к архитектуре приложения 11](#_Toc161136329)

[4.4 Требования к обслуживающему персоналу 12](#_Toc161136330)

[4.5 Требования к оформлению и верстке экранов 12](#_Toc161136331)

[4.5.1 Общие элементы верстки 15](#_Toc161136332)

[4.6 Общие технические требования к приложению 15](#_Toc161136333)

[4.6.1 Требования по безопасности 15](#_Toc161136334)

[4.6.2 Дополнительные требования 16](#_Toc161136335)

[5 Описание приложения 17](#_Toc161136336)

[5.1 Язык приложения 17](#_Toc161136337)

[5.2 Группы пользователей приложения 17](#_Toc161136338)

[5.2.1 Экран splash 17](#_Toc161136339)

[5.2.2 Onboarding 17](#_Toc161136340)

[5.2.3 Обучение 17](#_Toc161136341)

[5.2.4 Авторизация и регистрация 18](#_Toc161136342)

[5.2.5 Каталог 18](#_Toc161136343)

[5.2.6 Главный экран курса 18](#_Toc161136344)

[5.2.7 Экран раздела курса 18](#_Toc161136345)

[5.2.8 Профиль 18](#_Toc161136346)

[5.2.9 Экран пройденных курсов 19](#_Toc161136347)

[5.2.10 Редактор курса 19](#_Toc161136348)

[5.2.11 Настройки 19](#_Toc161136349)

[6 Состав и содержание работ по созданию приложения 20](#_Toc161136350)

[7 Порядок разработки приложения 21](#_Toc161136351)

[8 Порядок контроля и приемки приложения 22](#_Toc161136352)

[9 Требования к документированию 23](#_Toc161136353)

[9.1 Перечень подлежащих разработке документов 23](#_Toc161136354)

[9.2 Вид представления и количество документов 23](#_Toc161136355)

[Приложение 24](#_Toc161136356)

1. Терминология

**Сервер, серверная часть** – компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

**Клиентская сторона** – устройство, использующее ресурсы сервера и предоставляющий пользователю возможность взаимодействия с системой.

**Модерация** – рассмотрение заявок по размещению и редактированию курсов на платформе приложения.

**Администратор** - пользователь, имеющий доступ к расширенному функционалу по модерации курсов пользователей через панель администратора.

**REST API (REST)** – стиль архитектуры программного обеспечения для построения масштабируемых веб-приложений.

**Popup** – всплывающее окно, которое появляется внезапно, без запроса пользователя, и отображается поверх экрана, на котором находился пользователь.

**TabBar** – панель вкладок в нижней части экрана, позволяющая быстро переключаться между разделами приложения.

**Onboarding** – обучающая функциональность в приложении, появляющаяся при первом запуске для ознакомления пользователя с продуктом.

**Front-end** - клиентская часть приложения. Отвечает за получение информации с программно-аппаратной части и отображение ее на устройстве пользователя.

**Back-end** - программно-аппаратная часть приложения. Отвечает за функционирование внутренней части приложения.

1. Общие сведения
   1. Полное наименование системы и название приложения

Полное название мобильного приложения: «KnitWit».

* 1. Заказчик

Заказчик: Старший преподаватель Тарасов Вячеслав Сергеевич, Воронежский государственный университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Представитель заказчика: Ассистент Проскуряков Егор Дмитриевич, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

* 1. Разработчик

Разработчик: студент Павлов Александр Викторович, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчик: студент Акимушкин Евгений Владимирович, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

Разработчик: студент Перепечко Константин Кириллович, Воронежский Государственный Университет, Факультет Компьютерных Наук, кафедра Программирования и Информационных Технологий.

* 1. Перечень документов, на основании которых создается приложение

Данное мобильное приложение разрабатывается на основе данного технического задания и должно удовлетворять всем требованиям, указанным в нем.

* 1. Плановые сроки начала и окончания работы по созданию приложения

Начало работы по созданию системы: февраль 2024 года. Окончание работы по созданию системы: июнь 2024 года.

* 1. Порядок оформления и предъявления заказчику результатов работ по созданию приложения

К результатам труда разработчика относится:

* мобильное приложение, разработанное в соответствии с ТЗ;
* панель администратора в web-браузере, разработанная в соответствии с ТЗ;
* техническое задание;
* курсовая работа по проекту;
* видео с презентацией проекта и демонстрацией его исправной работы;
* UML диаграммы.

Заказчику передаются:

* мобильное приложение;
* панель администратора;
* техническое задание;
* курсовая работа по проекту;
* видео с презентацией проекта;
* UML диаграммы.

Результаты передаются заказчику частями по завершении каждой стадии работы по созданию приложения. Документация – в электронном виде в формате MS PDF.

Проектная документация должна быть разработана в соответствии с ГОСТ 34.602-20.

1. Назначение и цели создания приложения
   1. Назначение приложения

Мобильное приложение «KnitWit» является онлайн платформой для обучения ремеслу и хобби. Оно предоставляет доступ к видео-урокам и инструкциям для развития навыков в различных областях, обеспечивая удобство обучения и вдохновляя пользователей на творчество.

* 1. Цели создания приложения

К целям создания приложения относятся:

* создание удобной площадки для онлайн-обучения пользователей;
* предоставление доступа к обучающим материалам для развития навыков;
* стимулирование интереса пользователей к творчеству и саморазвитию;
* получение прибыли путем интеграции рекламы.
  1. Задачи приложения

Разрабатываемый проект должен решать следующие задачи:

* приобретение знаний без траты собственного времени на поиск информации;
* отслеживание прогресса при прохождении курса;
* предоставление обширной базы курсов для разных классов пользователей.

1. Требования к приложению
   1. Требования к структуре приложения в целом
      1. Требования к реализации

Система должна состоять из сервера приложения, реляционной базы данных, клиентской части.

Основной используемый стек технологий:

Back-end (серверная часть):

* Java 17;
* Spring Framework;
* PostgreSQL;
* система сборки Maven.

Язык Java был выбран, так как он не зависит от платформы. Можно создать Java-приложение на Windows, скомпилировать его в байт-код и запустить на любой другой платформе, поддерживающей JVM – виртуальную машину Java.

Основным преимуществом Spring Framework является большое количество реализованных внутренних библиотек, позволяющих быстро и качественно писать код.

В качестве базы данных была выбрана Postgres, т.к. поддерживает пользовательские объекты и их поведение, включая типы данных, функции, операции и другое. Это делает Postgres невероятно гибким и надежным. Среди прочего, он умеет создавать, хранить и извлекать сложные структуры данных.

Front-end (клиентская часть):

* Flutter.

Главное преимущество Flutter - скорость разработки. Готовые решения данного фреймворка позволяют писать меньше кода, что значительно упрощает процесс создания приложений и существенно экономит время разработчиков.

* + 1. Перспективы развития, модернизации приложения

В долгосрочной перспективе возможны следующие направления модернизации мобильного приложения:

* добавления подписки на курсы на платной основе;
* добавление системы отзывов с комментариями от пользователей.
  1. Функциональные требования к приложению

К разрабатываемому приложению выдвигаются следующие функциональные требования:

Авторизованный пользователь обладает следующими возможностями:

* получение доступа к выбранному курсу по подписке;
* просмотр демо-материалов выбранного курса без подписки;
* фильтрация и сортировка каталога курсов;
* выход из аккаунта;
* создание собственного курса и его редактирование;
* оставлять и просматривать общую пользовательскую оценку курса;
* просмотр курсов, на которые пользователь подписан;
* возможность включения и отключения уведомлений;
* редактирование профиля пользователя.

Неавторизованный пользователь обладает следующими возможностями:

* просмотр списка курсов;
* фильтрация и сортировка каталога курсов;
* авторизация;
* регистрация.

Администратор обладает следующими возможностями:

* редактирование каталога курсов;
* создание курса и его редактирование;
* авторизация;
* модерация курсов.
  1. Требования к архитектуре приложения

Разрабатываемый проект должен удовлетворять следующим основным требованиям:

* корректная работа на устройствах с операционной системой Android 9.0 и новее;
* реализация основных задач, стоящие перед данным проектом, функционала, описанный в данном техническом задании;
* соответствие шаблону Клиент-Серверного приложения с применением «Clean architecture» и разделением на два слоя: слой представления (presentation layer) – front-end, слой доступа к данным (data layer) – back-end и связью между ними по средству Rest API.
  1. Требования к обслуживающему персоналу

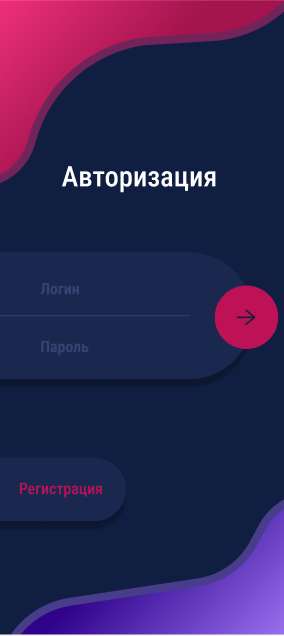
Обслуживанием приложения занимается команда разработчиков, которая обеспечивает поддержку приложения путем предоставления обновлений, в том числе при обнаружении нестабильной работы приложения, и занимается модерацией курсов.

* 1. Требования к оформлению и верстке экранов

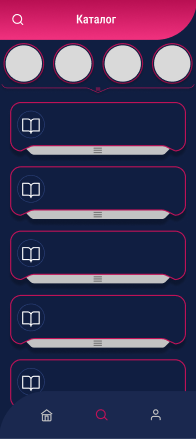
Экраны мобильного приложения должны быть оформлены в едином стиле с использованием шрифта Roboto.

Необходимо корректное и одинаковое отображение экранов мобильного приложения на устройствах с операционной системой Android 9.0 и выше.

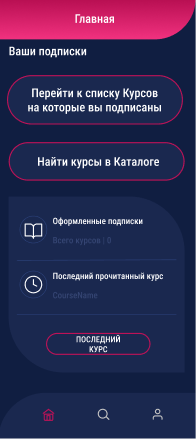
На рисунках 1-4 показаны прототипы основных типов экранов мобильного приложения:



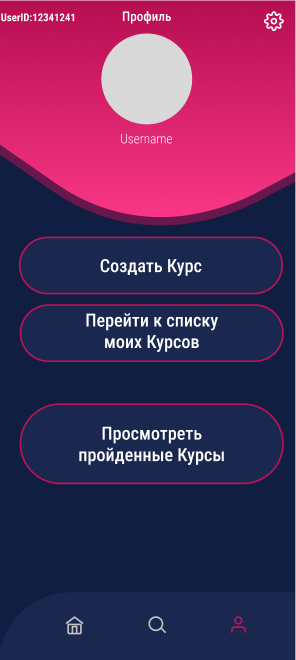
1. Экран авторизации



1. Экран вкладки каталога со списком курсов



1. Экран вкладки обучения



1. Экран вкладки профиля

Сервисная страница системы должна корректно работать в следующих браузерах:

* Google Chrome 122.0.6261.112;
* Yandex Browser 24.1.3.809.
  + 1. Общие элементы верстки

TabBar содержит кнопки перехода на экран обучения, каталога, профиля.

Реклама будет появляться по системе Popup.

* 1. Общие технические требования к приложению
     1. Требования по безопасности
* Обмен данных между клиентом и сервером должен осуществлять по протоколу https;
* Пароли пользователей должны хранится в базе данных в хешированном виде; для хеширования должен использоваться алгоритм bcrypt.
  + 1. Дополнительные требования

При первом запуске мобильного приложения пользователь должен иметь возможность ознакомиться с основными функциями приложения.

1. Описание приложения
   1. Язык приложения

Данное приложение поддерживает только русский язык.

* 1. Группы пользователей приложения

Пользователи приложения делятся на следующие группы:

* неавторизованный пользователь;
* авторизованный пользователь;
* администратор.

Функционал для данных ролей описан в пункте 4.2.

* + 1. Экран splash

Входная точка в приложение для любых пользователей. На данном экране запрашиваются системные разрешения при первом входе в приложение. Например, разрешение уведомлений и доступа к файловой системе устройства.

* + 1. Onboarding

Если пользователь первый раз запустил приложение, то показывается данная группа экранов. Данная группа экранов знакомит пользователя с ключевыми функциями приложения. Переключение экранов осуществляется с помощью свайпа влево или с помощью нажатия на кнопку «Далее». На последнем экране кнопка «Далее» заменяется на кнопку «Начать». Onboarding можно пропустить, нажав на «Х», находящийся в правом верхнем углу экрана.

* + 1. Обучение

При запуске приложения пользователь оказывается на экране обучения.

Здесь авторизованный пользователь может перейти к списку курсов, на которые он подписан, и перейти к последнему прочитанному им курсу. Также перейти к экрану каталога.

Для неавторизованного пользователя будет предложено войти в аккаунт или зарегистрироваться.

* + 1. Авторизация и регистрация

На экране регистрации пользователю будет предложено ввести свои логин, пароль и почту для создания аккаунта.

На экране авторизации пользователю необходимо ввести свои логин и пароль для входа в аккаунт.

* + 1. Каталог

На экране каталога пользователь имеет возможность находить курсы по фильтру через заданные теги или в строке поиска. После этого выводится список отфильтрованных курсов для просмотра и дальнейшего взаимодействия. Пользователь может перейти на главную страницу курса или же открыть всплывающую «шторку» с кратким описанием курса.

* + 1. Главный экран курса

На главном экране курса отображается его название, описание, автор, общая оценка, список разделов и кнопка оформления подписки. Авторизованный пользователь, который не подписан на данный курс, имеет возможность просмотреть демо-материалы курса и подписаться на курс. Авторизованный пользователь с подпиской на этот курс может изучать материал разделов курса и поставить оценку курсу.

* + 1. Экран раздела курса

Здесь отображается название раздела и его содержимое. Имеются переходы к списку разделов, следующему и предыдущему разделам, а также кнопка отметки прогресса.

* + 1. Профиль

На данном экране неавторизованному пользователю будет предложено войти в аккаунт или зарегистрироваться.

Для авторизованного пользователя на экране будут отображаться его аватар и логин.

Здесь пользователь может перейти в настройки, просмотреть список всех пройденных курсов и список своих собственных курсов, а также перейти в редактор и создать свой собственный курс.

* + 1. Экран пройденных курсов

Здесь пользователь может увидеть список всех пройденных им курсов и перейти на главный экран конкретного курса.

* + 1. Редактор курса

Здесь пользователь может создать новый или отредактировать уже существующие курсы, если он является автором или администратором.

* + 1. Настройки

Здесь авторизованный пользователь имеет возможность редактировать личную информацию, настроить приложение и выйти из аккаунта.

1. Состав и содержание работ по созданию приложения

Состав и содержание работ по созданию мобильного приложения включают в себя следующие этапы:

* сбор необходимой информации, постановка целей, задач приложения, которые в будущем должны быть реализованы;
* анализ предметной области, анализ конкурентов и построение структуры требований, ведущих к решению поставленных задач и целей;
* построение модели программы, описание спецификаций данных, определение связей между сущностями, разработка модели БД;
* разработка рабочего проекта, состоящего из написания кода, отладки и корректировки кода программы;
* проведение тестирования программного обеспечения.

1. Порядок разработки приложения
2. Перечень документов, предъявляемых по окончании соответствующих этапов работ

|  |  |
| --- | --- |
| Этап работы | Предъявляемые документы |
| 1 аттестация | Техническое задание |
| Промежуточный курсовой проект |
| 2 аттестация | Промежуточный курсовой проект |
| 3 аттестация | Готовый курсовой проект |

1. Порядок контроля и приемки приложения

Предварительные отчеты по работе будут проводиться во время рубежных аттестаций:

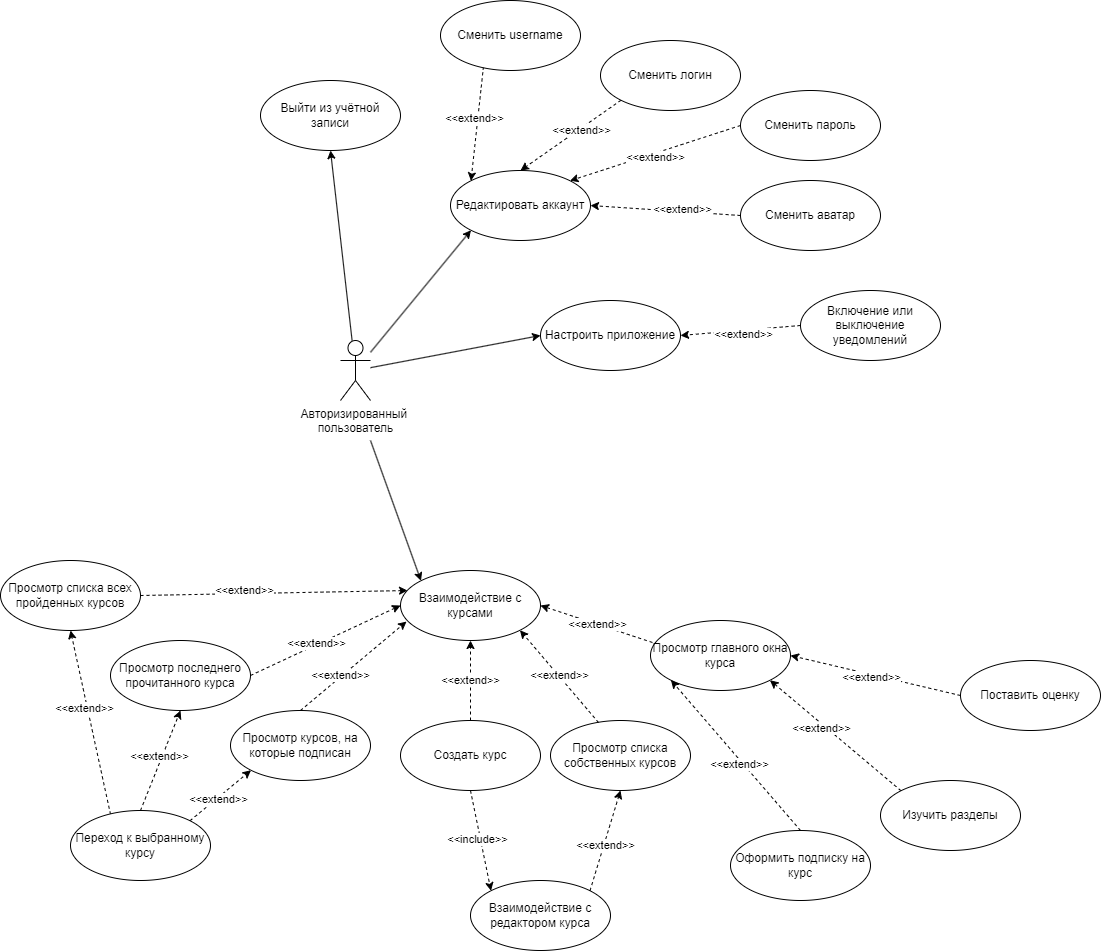
* 1 аттестация (середина марта 2024) - создан репозиторий проекта на GitHub, распределены задачи проекта в таск-менеджере YouTrack, создан проект Miro с общей логикой системы, предоставлены промежуточные результаты по курсовому проекту, готовое техническое задание, презентация проекта, видео-презентация проекта и сопроводительное письмо преподавателю практики;
* 2 аттестация (середина апреля 2024) - написана основополагающая часть кода приложения, реализована БД и ее взаимодействие с сервером, проведена отладка и доработка кода, проведено тестирование по работе приложения;
* 3 аттестация (конец мая 2024) - разработан курсовой проект, выполнены завершающие работы по доработке приложения, предоставлена готовое приложение.

1. Требования к документированию
   1. Перечень подлежащих разработке документов

* Техническое задание;
* Курсовой проект;
* Сопроводительное письмо.
  1. Вид представления и количество документов

Документы должны быть представлены в электронном виде и опубликованы на сайте github.com в репозитории команды разработчика, а также в печатном виде. Проектная документация должна быть разработана в соответствии с ГОСТ 7.32-2001.

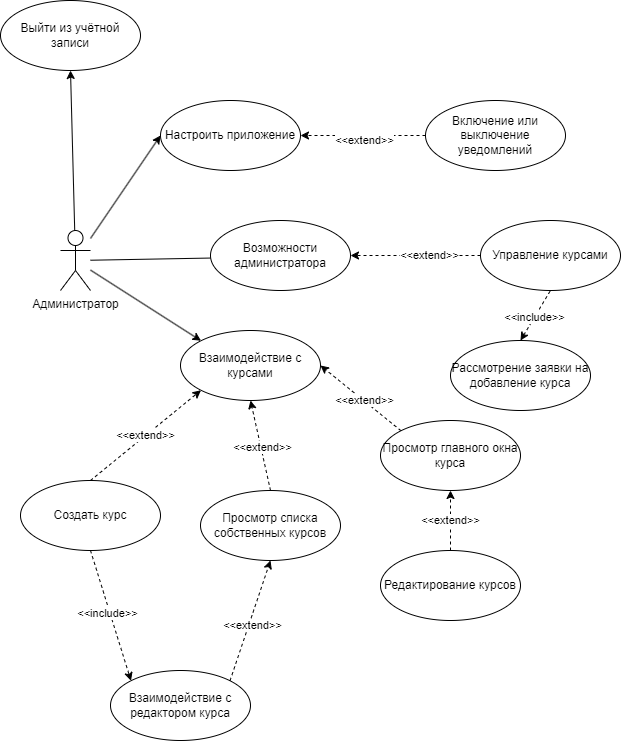
Приложение



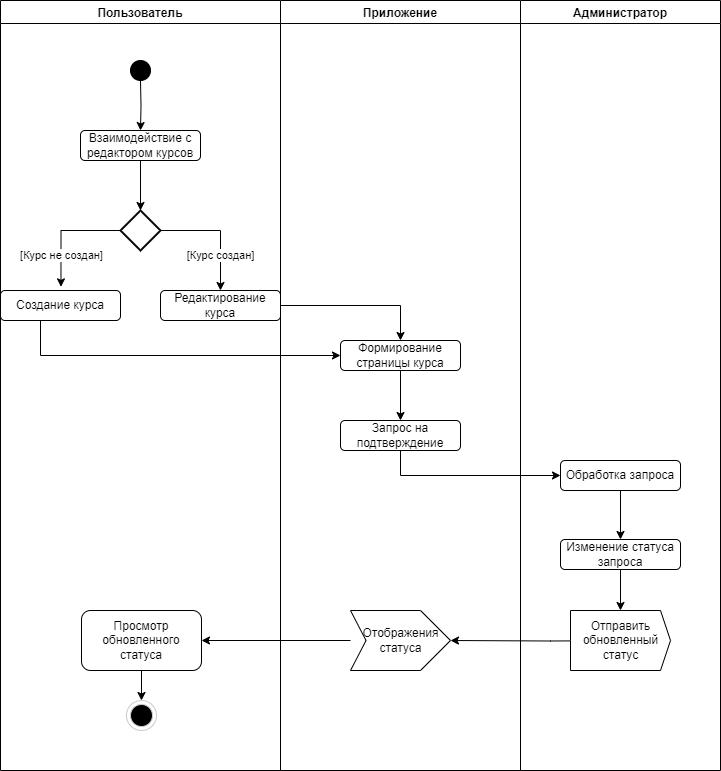
1. Диаграмма прецедентов (user)



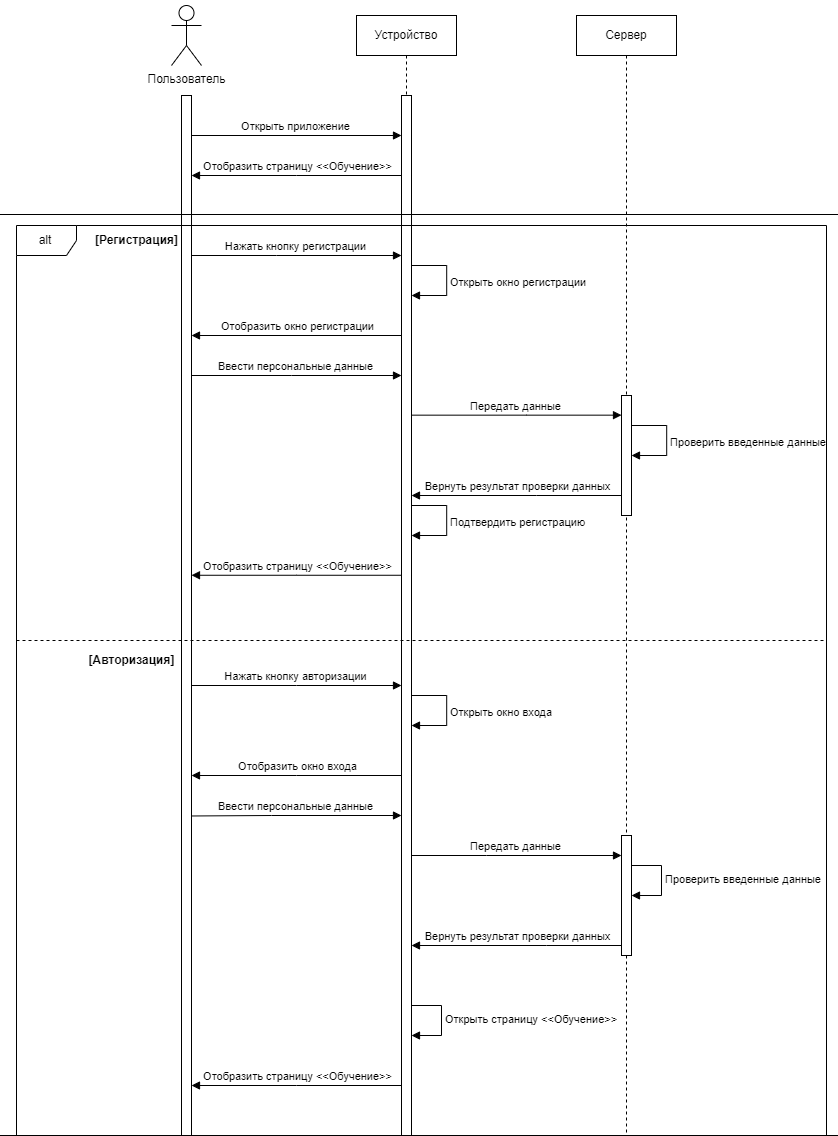
1. Диаграмма прецедентов (unauthorized user)



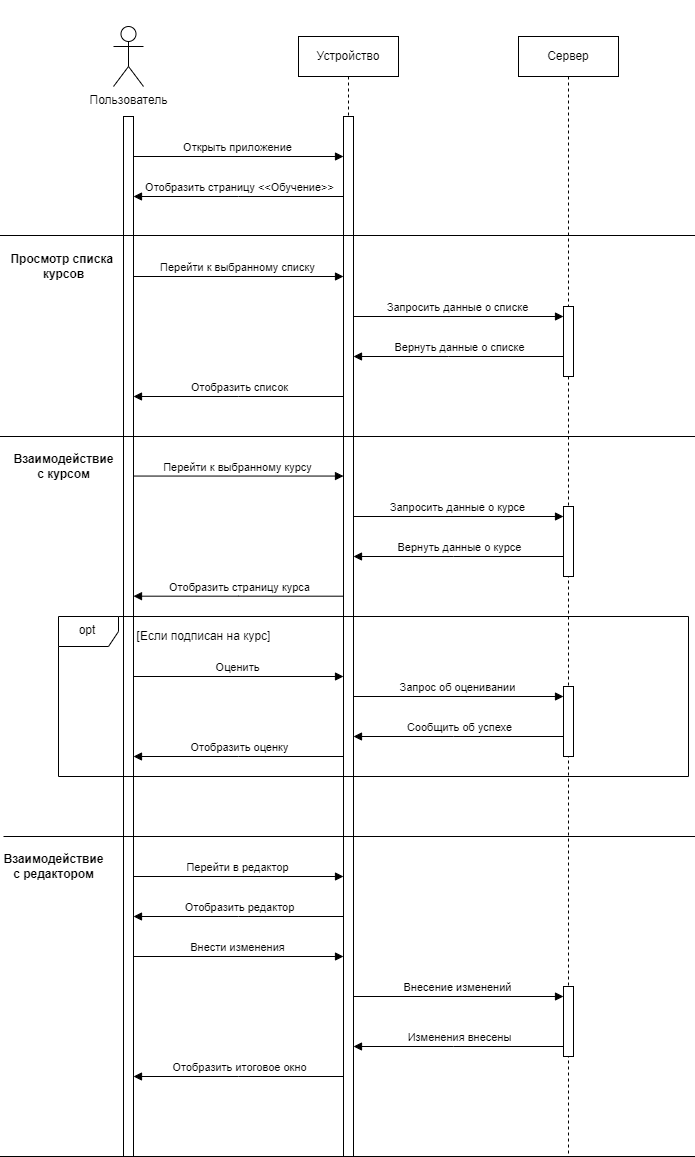
1. Диаграмма прецедентов (admin)



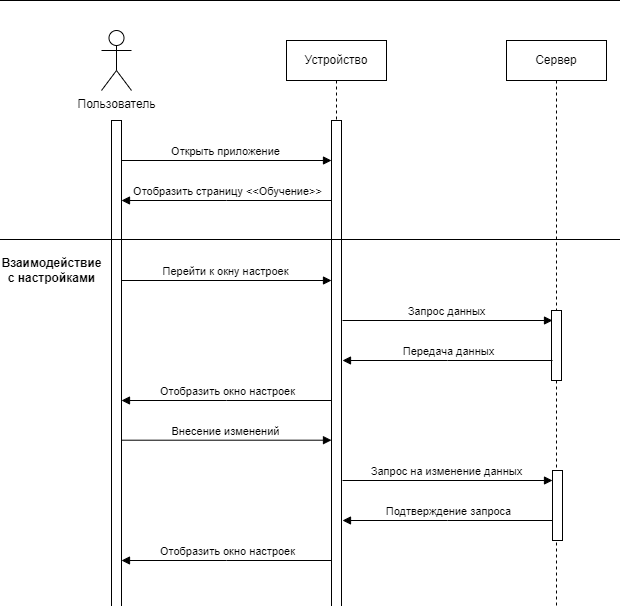
1. Диаграмма активности (взаимодействие с редактором курса)



1. Диаграмма последовательности



1. Диаграмма последовательности (продолжение)



1. Диаграмма последовательности (продолжение)