

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук
Кафедра программирования и информационных технологий

Мобильное приложение «StreamEd»

Курсовой проект

по дисциплине

Технологии программирования

09.03.02 Информационные системы и технологии
Обработка информации и машинное обучение

6 семестр 2023/2024 учебного года

Зав. кафедрой _____ д. ф.-м. н., профессор С.Д. Махортов

Обучающийся _____ ст. 3 курса оч. отд. В. А. Батурин

Обучающийся _____ ст. 3 курса оч. отд. А. С. Пекшев

Обучающийся _____ ст. 3 курса оч. отд. Д. С. Семенов

Руководитель _____ В.С. Тарасов, ст. преподаватель __. __.20__

Воронеж 2024

Содержание

Введение.....	4
1 Постановка задачи.....	5
1.1 Цели создания системы	5
1.2 Задачи проекта.....	5
1.3 Требования к разрабатываемой системе	5
1.3.1 Требования к разрабатываемой системе	5
1.3.2 Требования к функциям выполняемым приложением	5
2 Анализ предметной области	7
2.1 Терминология	7
2.2 Обзор аналогов	8
2.2.1 MyOwnConference.....	8
2.2.2 Pruffme	10
2.2.3 МТС Линк.....	11
2.3 Моделирование системы	12
2.3.1 Диаграмма в стиле методологии IDEF0	12
2.3.2 Диаграмма прецедентов	13
2.3.3 Диаграммы последовательности.....	14
2.3.4 Диаграмма развертывания	17
2.3.5 Диаграммы состояния	18
3 Реализация.....	20
3.1 Средства реализации.....	20
3.2 Реализация базы данных	20
3.2.1 ER-диаграмма.....	20

3.3 Реализация клиентской части	21
3.3.1 Формы экранов неавторизованного пользователя	21
3.3.2 Форма экрана авторизации	22
3.3.3 Форма экрана регистрации	23
3.3.4 Форма экрана восстановления пароля	24
3.3.5 Форма экрана личной страницы	25
3.3.6 Форма экрана подключения по коду.....	26
3.3.7 Форма экрана с курсами.....	26
3.3.8 Форма экрана расписания	27
3.3.9 Форма экрана с записями трансляций.	27
3.3.10 Форма экрана с занятиями (для преподавателя).....	28
3.3.11 Форма экрана с размещением курса (для преподавателя)	29
3.3.12 Форма экрана с вебинаром.....	29
3.4 Серверная часть.....	31
3.4.1 Архитектура серверной части приложения	31
4 Аналитика	32
Заключение	33

Введение

В эпоху цифровых технологий и развития Интернета, онлайн-образование и дистанционное обучение стали неотъемлемой частью нашей жизни. Популярностью пользуются различные вебинары, онлайн-курсы и семинары, которые позволяют получить новые знания и навыки, не выходя из дома. Однако, для проведения и трансляции таких мероприятий требуются специальные платформы, которые обеспечивают комфортное и эффективное взаимодействие между преподавателем и студентами.

Онлайн-платформы для вебинаров – это программные приложения, предназначенные для проведения и трансляции онлайн-мероприятий, а также для взаимодействия со зрителями и хранения записей. Они позволяют преподавателям проводить вебинары, семинары и курсы, а студентам – подключаться к ним, задавать вопросы и получать ответы в режиме реального времени. Такие платформы также предоставляют возможность хранить записи трансляций, что позволяет студентам просматривать их в любое удобное время.

В данной курсовой работе рассматривается процесс разработки собственной онлайн-платформы для проведения и трансляции вебинаров с возможностью взаимодействия со зрителями и хранения записей. В рамках работы будут рассмотрены различные аспекты разработки такой платформы, начиная с анализа предметной области, определения его концепции и основных требований. Затем будет изучено проектирование пользовательского интерфейса и пользовательского опыта, с учетом современных тенденций и личных практик в этой области. Важное внимание будет уделено выбору и интеграции соответствующих технологий для обеспечения необходимых функций, таких как трансляция видео и аудио, чат, модерация и другие.

1 Постановка задачи

1.1 Цели создания системы

Целью данной работы является разработка онлайн-платформы для проведения и трансляции вебинаров с возможностью взаимодействия со зрителями и хранения записей.

1.2 Задачи проекта

- Регистрация и авторизация пользователей;
- Создание и редактирование курсов;
- Подписка на курсы;
- Проведение вебинаров (трансляций);
- Просмотр трансляций;
- Общение с преподавателем через чат;
- Просмотр записей трансляций;
- Модерация чата;
- Добавление и редактирование пользователей (студентов и преподавателей);
- Добавление и редактирование курсов;
- Удаление трансляций и записей.

1.3 Требования к разрабатываемой системе

1.3.1 Требования к разрабатываемой системе

- обеспечение авторизации и аутентификации пользователей;
- использование механизмов защиты от SQL-инъекций;
- использование протокола передачи данных HTTP;
- приложение должно быть построено на трехуровневой архитектуре.

1.3.2 Требования к функциям выполняемым приложением

- Неавторизованный пользователь должен обладать возможностью:
- авторизоваться/зарегистрироваться в приложении;
- просматривать список рекомендуемых курсов;
- просматривать список всех курсов.

- Авторизованный студент должен обладать возможностью:
- записаться на нужный курс;
- просматривать прямые трансляции;
- общаться в чате с преподавателем;
- задавать анонимные вопросы в чате;
- просматривать записи вебинаров.
- Авторизованный преподаватель должен обладать возможностью:
- создавать свои курсы;
- проводить в них прямые трансляции;
- Записывать трансляции;
- общаться в чате со студентами
- модерировать чат, удаляя непристойные сообщения.

2 Анализ предметной области

2.1 Терминология

Мобильное приложение – Программное обеспечение, предназначенное для работы на смартфонах, планшетах и других мобильных устройствах, разработанное для конкретной платформы.

Клиент – Это аппаратный или программный компонент вычислительной системы, посылающий запросы серверу.

Сервер – Выделенный или специализированный компьютер для выполнения сервисного программного обеспечения.

База данных – это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе. База данных обычно управляется системой управления базами данных.

СУБД – система управления базами данных. Программное обеспечение, предназначенное для создания, управления и манипулирования базами данных. СУБД обеспечивает механизмы для хранения, извлечения, обновления и управления данными, а также для обеспечения целостности данных и безопасности.

SQL-запросы – Это наборы команд для работы с реляционными базами данных.

SQL-инъекция – Внедрении в запрос произвольного SQL-кода, который может повредить данные, хранящиеся в БД или предоставить доступ к ним.

HTTP – Это протокол, позволяющий получать различные ресурсы, например HTML-документы. Протокол HTTP лежит в основе обмена данными в Интернете.

HTTPS – это защищенная версия протокола HTTP, которая использует шифрование для обеспечения безопасности и конфиденциальности передаваемых данных, таких как пароли и номера кредитных карт. HTTPS использует сертификаты SSL/TLS для шифрования и обеспечения безопасности соединения между клиентом и сервером.

Android – Это операционная система с открытым исходным кодом, созданная для мобильных устройств на основе модифицированного ядра Linux.

Аутентификация – Процедура проверки подлинности, например, проверка подлинности пользователя путем сравнения введенного им пароля с паролем, сохраненным в базе данных.

Авторизация – Предоставление определенному лицу или группе лиц прав на выполнение определенных действий.

Модерация чата – это процесс контроля и управления сообщениями в чате, чтобы обеспечить соблюдение правил и норм общения, удалять нежелательные сообщения, предотвращая нарушения.

Кодеки – это программное обеспечение или аппаратное устройство, используемое для сжатия и декодирования аудио и видео данных.

Протоколы – это наборы правил и стандартов, определяющих, как устройства в сети обмениваются данными и информацией.

Прокси-сервер – это сервер, который действует как посредник между пользователем и интернетом, обрабатывая запросы пользователя и передавая их от имени пользователя к интернет-ресурсам.

2.2 Обзор аналогов

Разрабатывая приложение для проведения вебинаров, необходимо рассматривать разработку с точки зрения актуальности и уникальности проекта. Для оценки этих качеств необходимо прибегнуть к рассмотрению аналогов разрабатываемого приложения, адекватно оценивая все положительные и негативные черты того или иного продукта.

2.2.1 MyOwnConference

MyOwnConference – это платформа для проведения вебинаров и онлайн-конференций, которая предоставляет возможность для проведения онлайн-мероприятий, таких как вебинары, семинары, конференции и онлайн-курсы. Платформа предлагает различные функции, такие как трансляция видео и аудио, чат, опросы, презентации, запись трансляций и другие.

MyOwnConference позволяет создавать мероприятия на большом количестве участников и предоставляет возможность для интеграции с другими сервисами, такими как CRM-системы, платежные системы и другие.



Рисунок 1 - Интерфейс вебинара «MyOwnConference»

Преимущества:

- Большое количество функций для проведения вебинаров и онлайн-конференций;
- Возможность проводить опросы и голосования во время мероприятия;
- Возможность создавать мероприятия на большое количество участников (до 10000 человек);
- Возможность интеграции с сервисами электронной коммерции для продажи билетов на мероприятия.
- Возможность записи трансляций;

Недостатки:

- Относительно высокая стоимость использования платформы;
- Ограниченные возможности для настройки дизайна и функциональности мероприятия;

- Недостаточная гибкость в планировании и управлении мероприятиями;
- Ограниченная поддержка мобильных устройств.

2.2.2 Pruffme

Pruffme – это относительно новая платформа для проведения вебинаров, которая предлагает уникальные функции, такие как создание мероприятий с использованием виртуальной реальности, искусственного интеллекта, игровых элементов и интерактивных презентаций. Платформа позволяет создавать мероприятия на большом количестве участников и предоставляет возможность для интеграции с другими сервисами.

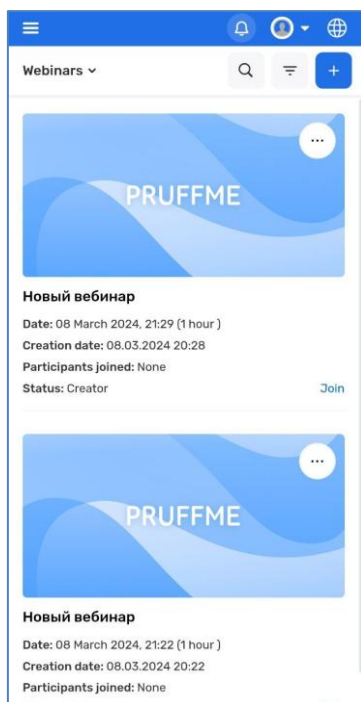


Рисунок 2 - Интерфейс сайта «Pruffme»

Преимущества:

- Большое количество функций для проведения вебинаров и онлайн-конференций;
- Мероприятия с использованием виртуальной реальности;
- Мероприятия с использованием искусственного интеллекта;
- Возможность создавать мероприятия с использованием игровых элементов;

- Возможность создавать мероприятия на большое количество участников;
- Интеграция с другими сервисами;
- Возможность записи трансляций;

Недостатки:

- Относительно высокая стоимость использования платформы;
- Ограниченные возможности для настройки дизайна и функциональности мероприятия;
- Ограниченная поддержка мобильных устройств.

2.2.3 МТС Линк

МТС Линк – это современная платформа для бизнес-коммуникаций, предлагающая комплексные инструменты для организации вебинаров, курсов, и совместной работы в команде. Платформа включает в себя функции для проведения онлайн-встреч, создания интерактивных досок, и проведения тестов, что делает её идеальным решением для образовательных и корпоративных целей. МТС Линк также поддерживает интеграцию с другими сервисами, такими как МойОфис и Битрикс24, что расширяет возможности использования платформы и упрощает работу с документами и проектами.

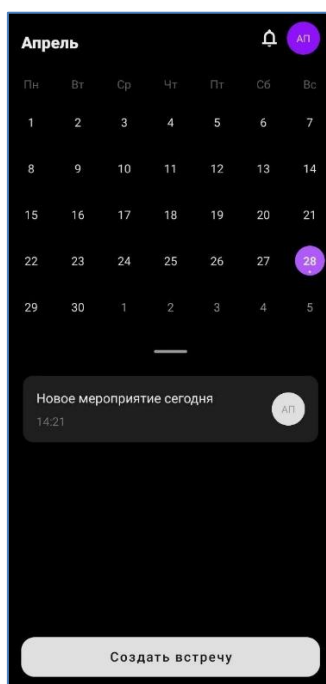


Рисунок 3 - Интерфейс мобильного приложения «МТС Линк»

Преимущества:

- Большое количество функций для проведения вебинаров и онлайн-конференций;
- Экосистема сервисов для бизнес-коммуникаций, включая вебинары, курсы, чаты, и доски, что позволяет проводить конференции, курсы, и настраивать совместную работу с помощью майнд-карт и мессенджера;
- Поддержка множества платформ;
- Возможность создавать мероприятия на большое количество участников;
- Интеграция с другими сервисами, такими как МойОфис, Битрикс24, Яндекс Метрику, что расширяет возможности использования платформы;
- Возможность записи трансляций;

Недостатки:

- Относительно высокая стоимость использования платформы;
- Ограниченные возможности для настройки дизайна и функциональности мероприятия;
- Большинство функций недоступно в мобильно приложении.
- Недостаточная оптимизация мобильного приложения

2.3 Моделирование системы

2.3.1 Диаграмма в стиле методологии IDEF0

IDF0 диаграмма представляет собой графическое представление бизнес-процесса в виде иерархической структуры функций. Основная цель IDF0 диаграммы состоит в том, чтобы показать, как различные функциональности взаимодействуют друг с другом и как они влияют на достижение целей организации. Она помогает улучшить понимание процессов и оптимизировать их для повышения эффективности организации. Данная диаграмма представлена на рисунке 4.

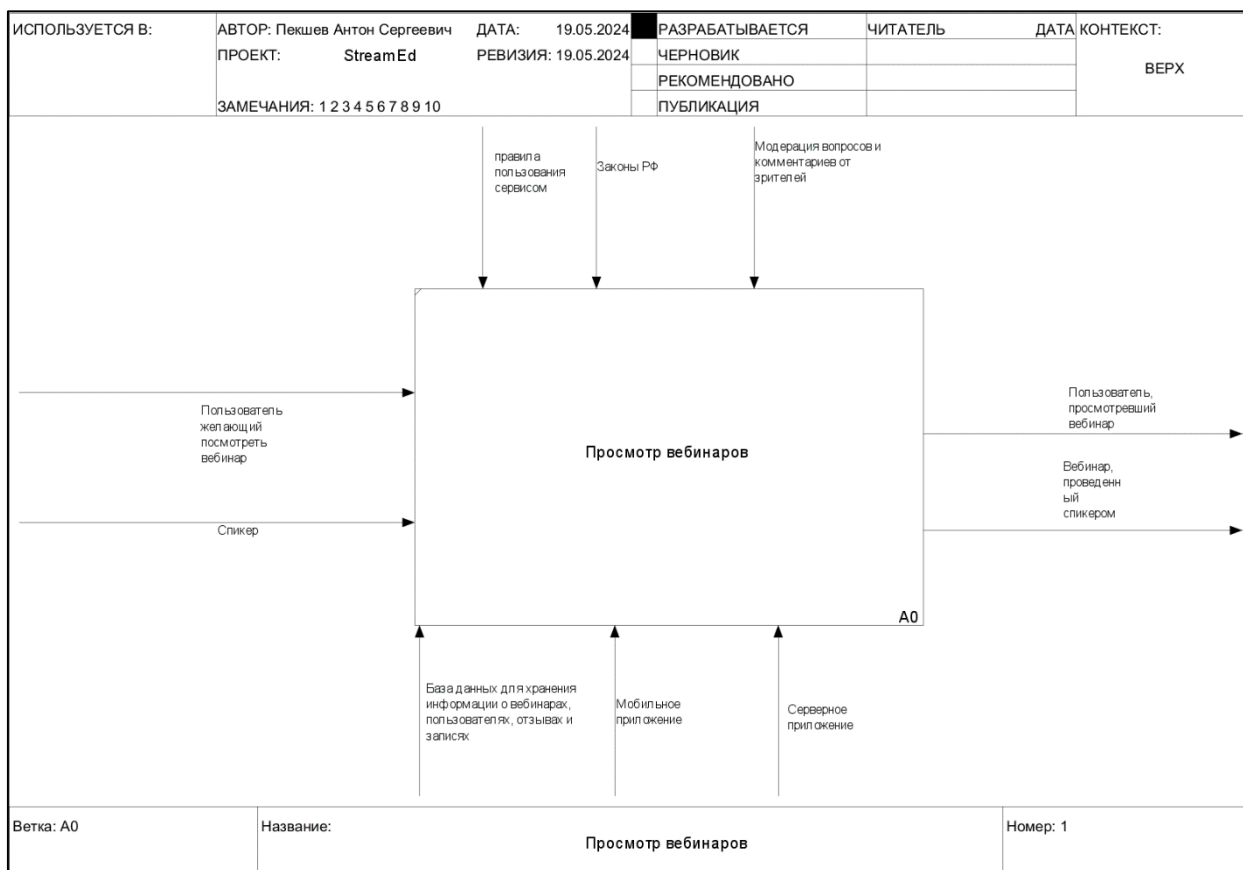


Рисунок 4 – Диаграмма в стиле методологии IDEF0

2.3.2 Диаграмма прецедентов

Рассмотрим полную диаграмму для использования приложения разными типами пользователей. В данном случае необходимость составления диаграммы прецедентов продиктована прежде всего тем, что use-case диаграмма — это инструмент для моделирования системы и понимания ее функциональности и потребностей пользователей. Они помогают в определении основных действий, которые пользователь должен совершить в системе, чтобы достичь определенных целей. Они также позволяют определить возможные риски и проблемы, которые могут возникнуть в ходе использования системы. Данная диаграмма представлена на рисунке 5.

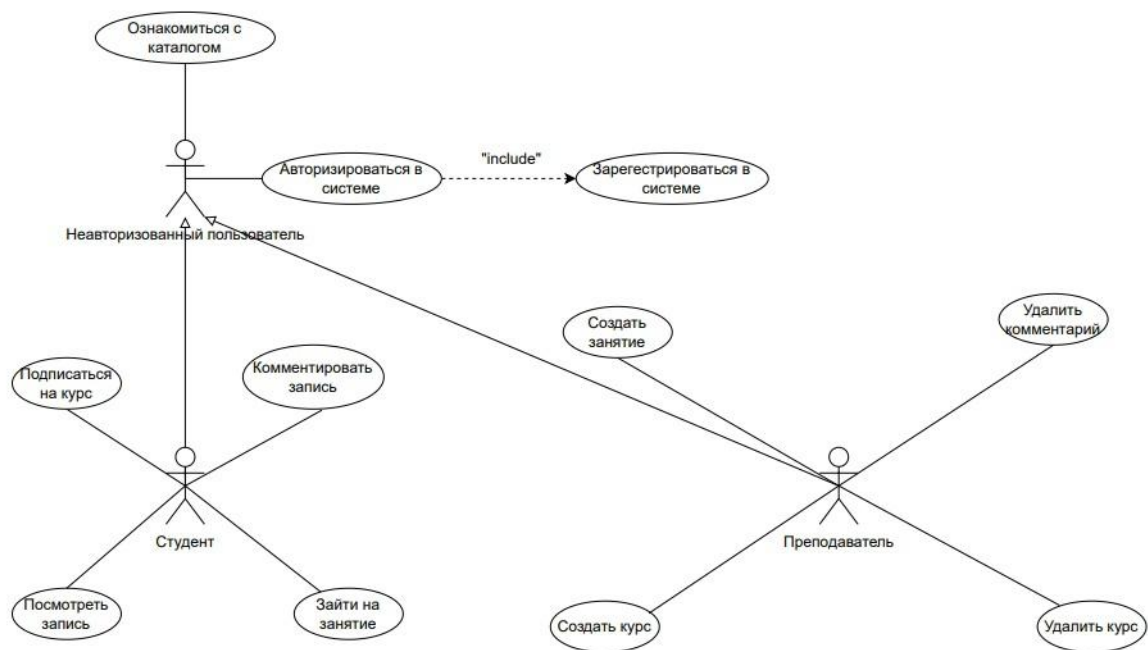


Рисунок 5 – Use-Case диаграмма пользования приложением

2.3.3 Диаграммы последовательности

Диаграмма последовательности является важным инструментом для проекта, который помогает более глубоко понимать процесс, улучшать его эффективность и упрощать взаимодействие.

Рассмотрим диаграмму последовательности приложения. Данные диаграммы представлена на рисунках 6-8.

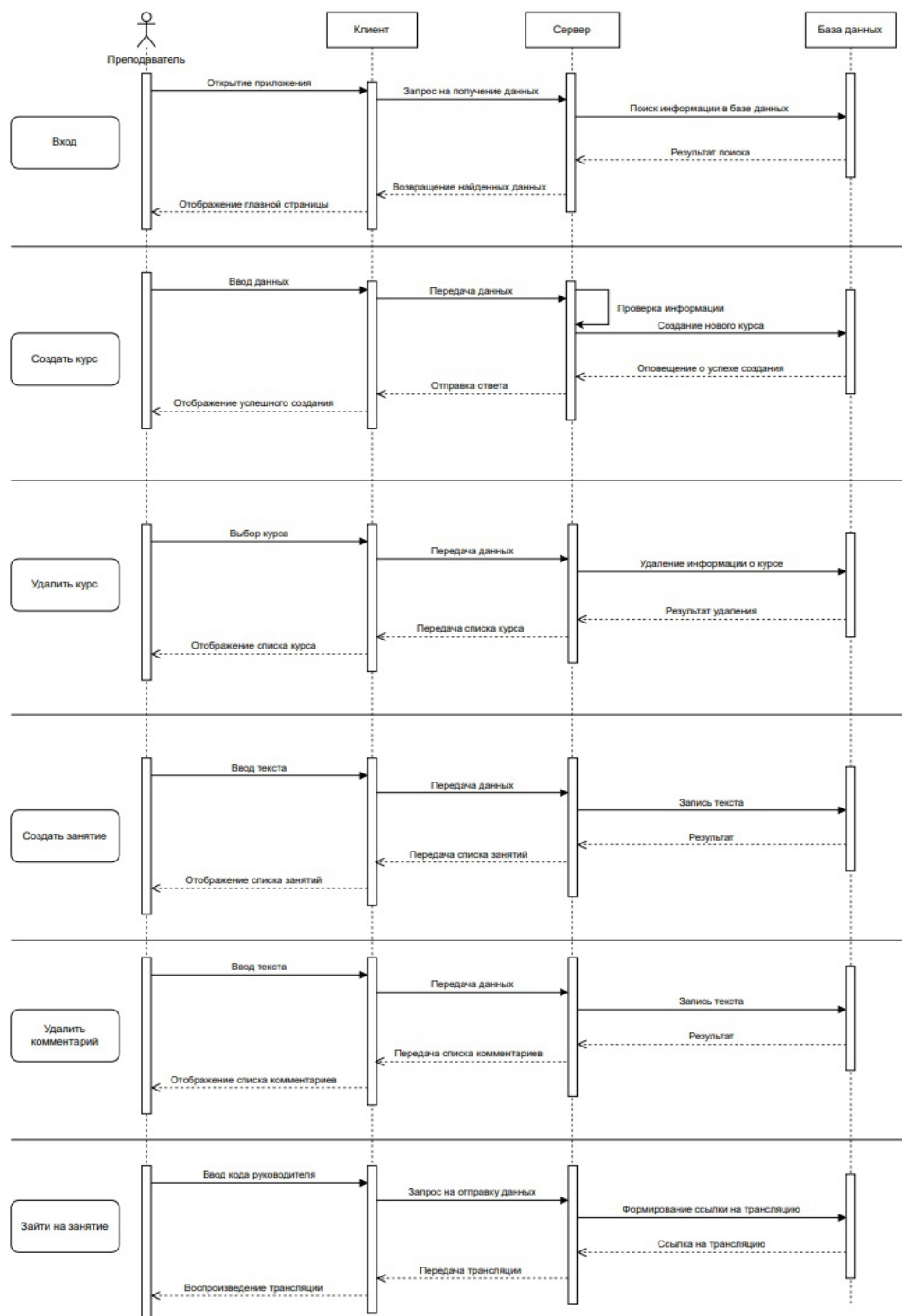


Рисунок 6 – Диаграмма последовательности преподавателя

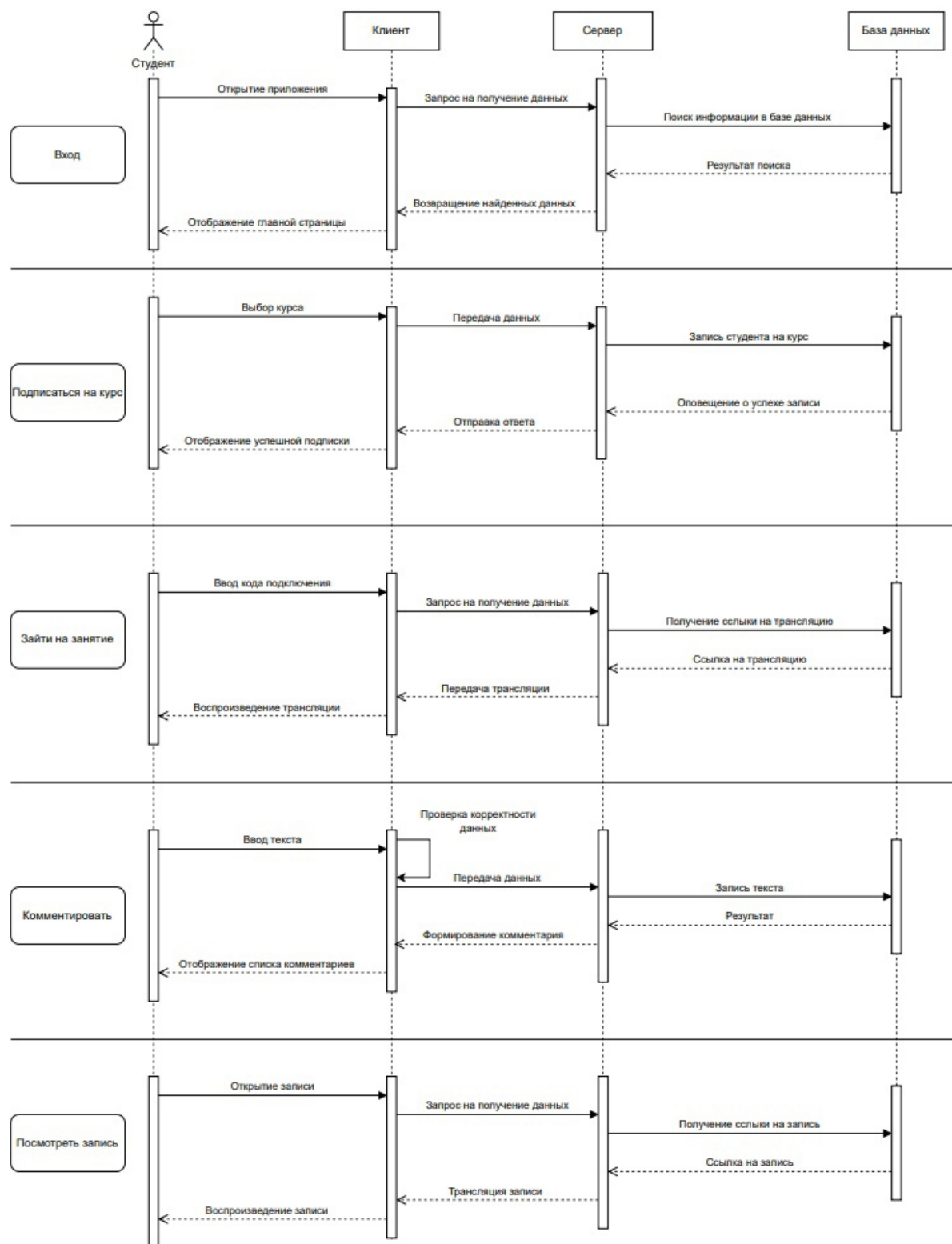


Рисунок 7 – Диаграмма последовательности студента

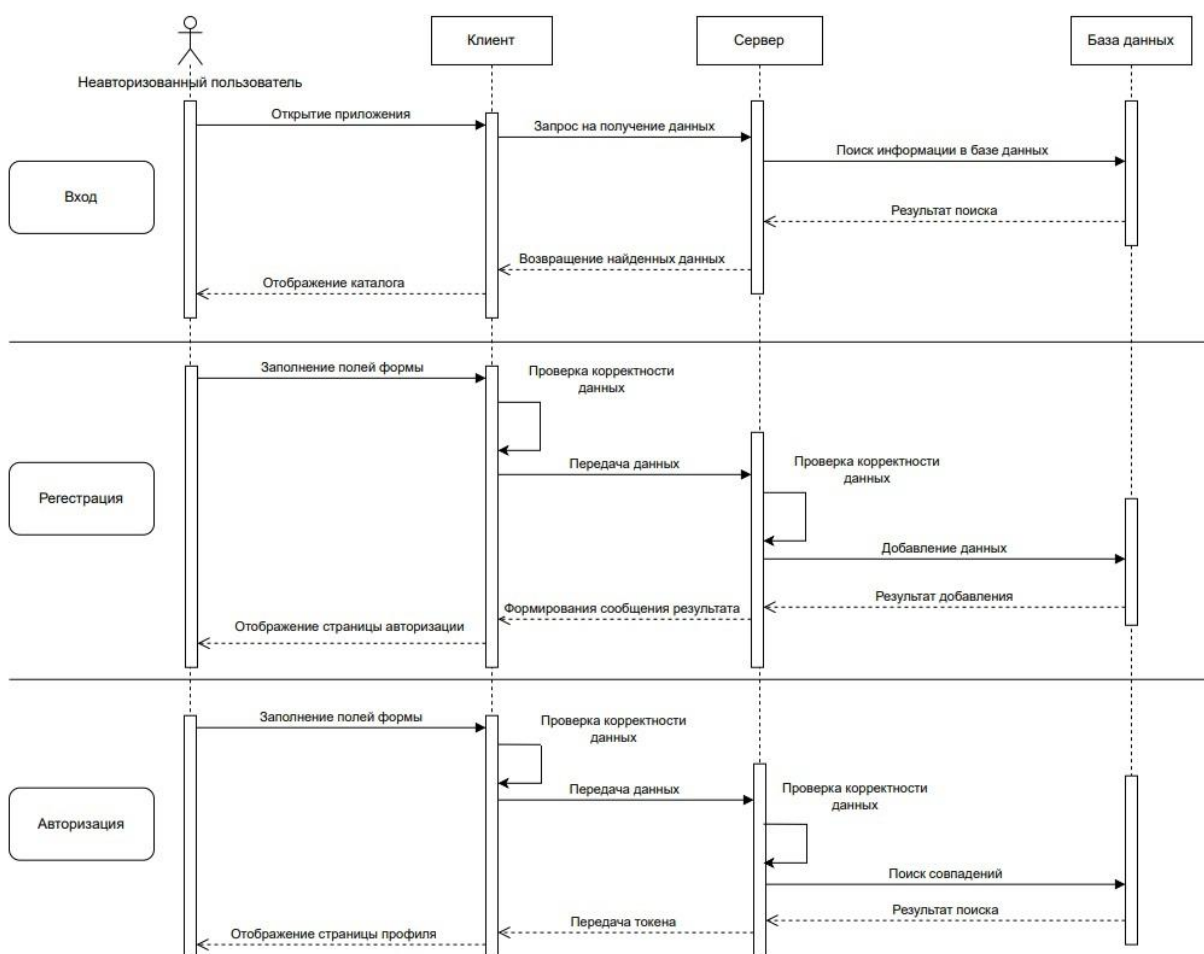


Рисунок 8 – Диаграмма последовательности неавторизованного пользователя

2.3.4 Диаграмма развертывания

Диаграмма развертывания позволяет определить требования к аппаратному обеспечению, планировать установку и настройку компонентов системы, а также оценивать ее производительность и масштабируемость. Данная диаграмма представлена на рисунке 9

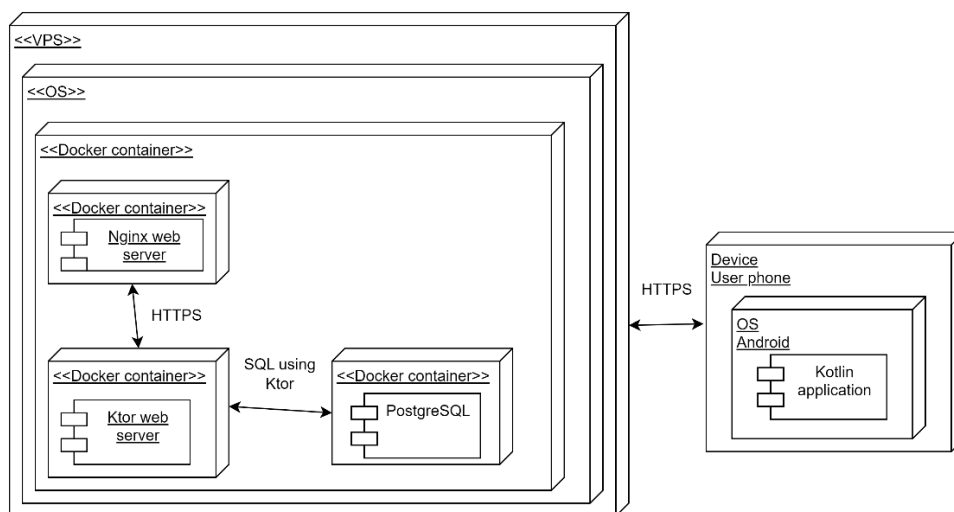


Рисунок 9 – Диаграмма разворачивания приложения

2.3.5 Диаграммы состояния

Диаграмма состояния позволяет определить возможные сценарии поведения системы, выделить ключевые состояния и переходы между ними, а также оценить ее надежность и устойчивость к ошибкам. Для нашего проекта были спроектированы 3 диаграммы для состояний преподавателя, студента и неавторизованного пользователя. Данные диаграммы представлены на рисунках 10-12



Рисунок 10 – Диаграмма состояния преподавателя

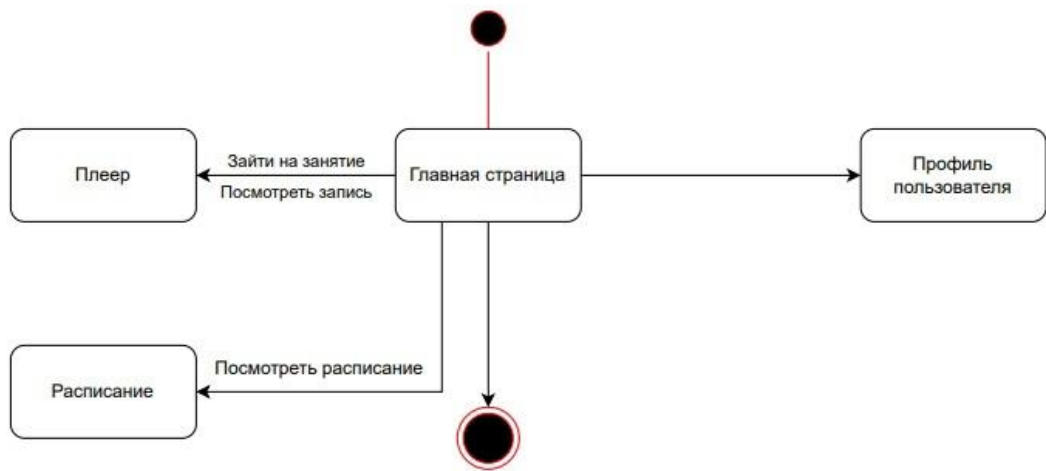


Рисунок 11 – Диаграмма состояния студента

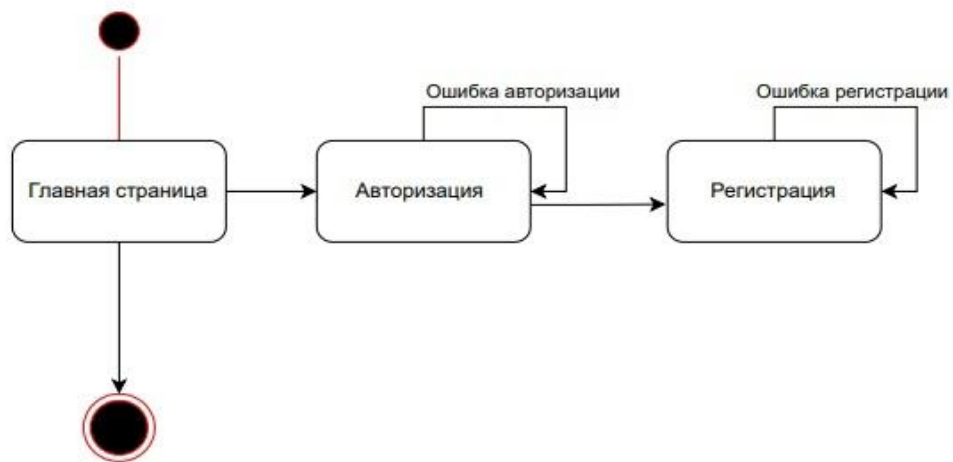


Рисунок 12 – Диаграмма состояния студента

3 Реализация

3.1 Средства реализации

Ниже приведен перечень используемых технологий.

Backend

- Kotlin;
- Ktor;
- PostgreSQL;
- Docker.

Frontend:

- Android SDK;
- Kotlin.

Инструменты для ведения документации:

- Miro;
- Draw.io;
- Ramus;
- Figma.

Дополнительный инструментарий:

- Git;
- GitHub;
- YouTrack.

3.2 Реализация базы данных

Для хранения данных была выбрана база данных PostgreSQL. Она является продуктом с открытым исходным кодом, который поддерживается многими серверами. PostgreSQL поддерживает множественные типы данных, такие как числа разной точности, тексты с различными кодировками, изображения, звуки, видео, XML-документы, JSON-объекты и многие другие.

3.2.1 ER-диаграмма

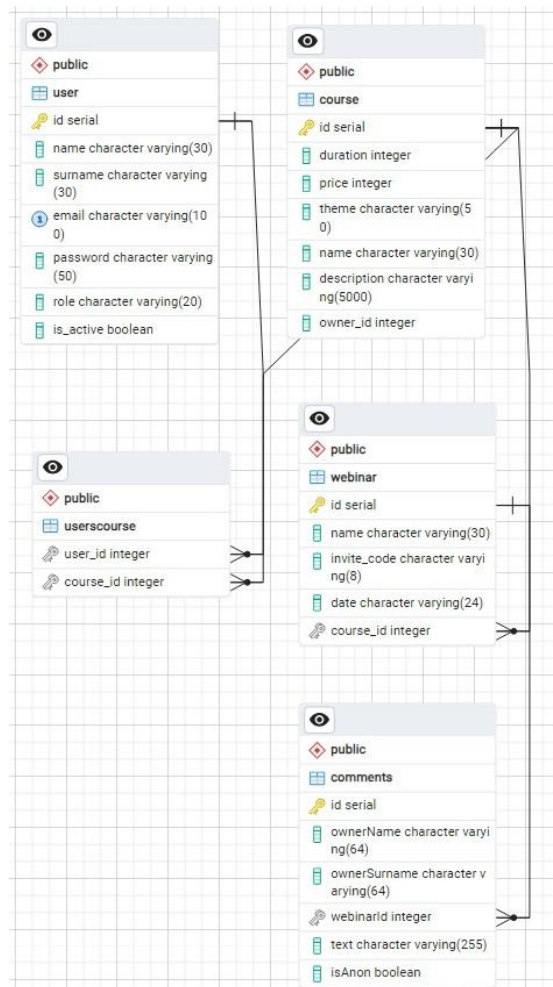


Рисунок 13 – ER диаграмма базы данных

3.3 Реализация клиентской части

Для реализации клиентской части приложения было выбрано сочетание Android SDK и языка Kotlin. Этот набор обеспечивает разработчикам множество возможностей для создания мобильных приложений под Android. Они могут использовать SDK для доступа к аппаратным возможностям устройства, взаимодействия с различными сервисами и API, разработки пользовательского интерфейса и многого другого. Kotlin, в свою очередь, предоставляет надежный и мощный язык программирования, который позволяет разработчикам создавать сложные и высокопроизводительные приложения.

3.3.1 Формы экранов неавторизованного пользователя

Пользователь (неавторизованный) имеет возможность просматривать список всех курсов. Для того, чтобы их посмотреть ему предлагают авторизоваться.

Снизу показан экран первого запуска приложения для неавторизированных пользователей

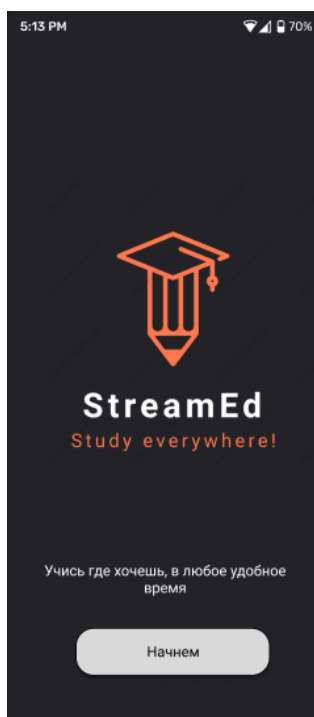


Рисунок 14 – Форма первого экрана неавторизованного пользователя

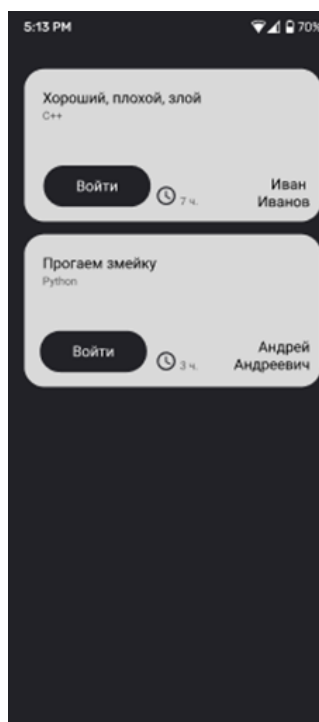


Рисунок 15 – Форма второго экрана неавторизованного пользователя

3.3.2 Форма экрана авторизации

На данном экране отображены поля с вводом логина и пароля к аккаунту, кнопка для входа, кнопка для регистрации, а также кнопка для восстановления пароля.

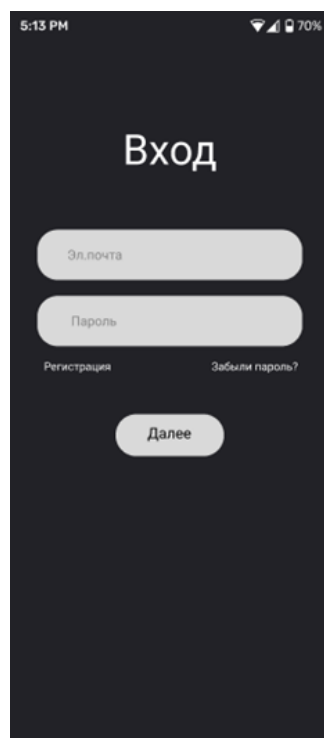


Рисунок 16 – Форма экрана авторизации

3.3.3 Форма экрана регистрации

На данном экране отображены поля для ввода имени, фамилии, почты, пароля и подтверждения пароля. Ниже расположена кнопка регистрации.

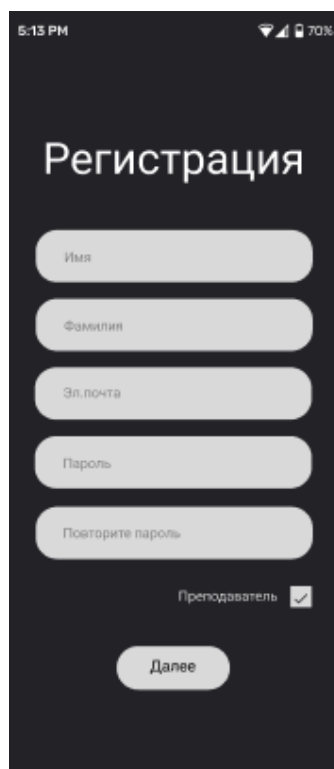


Рисунок 17 – Форма экрана авторизации

3.3.4 Форма экрана восстановления пароля

На данном экране отображены поля для ввода адреса электронной почты и кнопка отправки кода. После введения данных появляется форма для ввода полученного кода, и кнопка для подтверждения кода, а также ввода нового пароля. При успешной проверке кода пароль пользователя обновляется. В случае неудачи пользователь будет возвращен на начальную форму восстановления пароля.

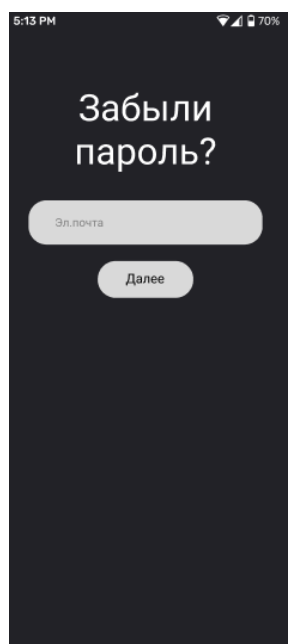


Рисунок 18 Форма экрана восстановления пароля

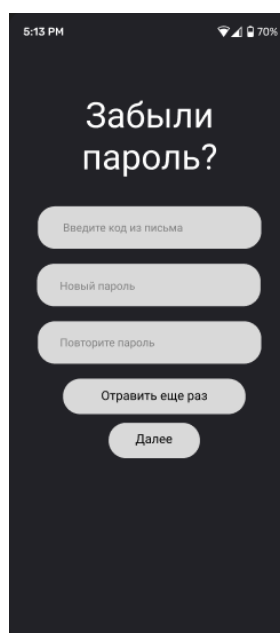


Рисунок 19 Форма экрана ввода нового пароля.

3.3.5 Форма экрана личной страницы

На данном экране указана информация о пользователе (имя, фамилия, почта) рядом расположена кнопка редактирования пользователя, выхода из аккаунта и удаления аккаунта.

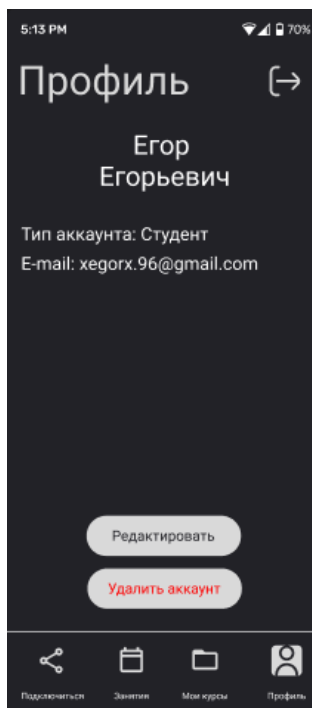


Рисунок 20 Форма экрана личной страницы.

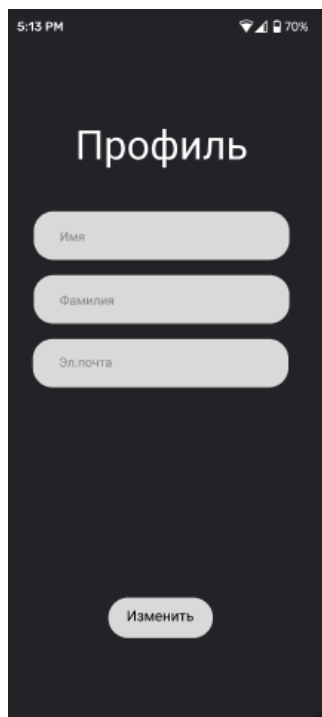


Рисунок 21 Форма редактирования личной страницы.

3.3.6 Форма экрана подключения по коду

На данном экране расположены поле ввода кода и кнопка для подключения к трансляции по этому коду.

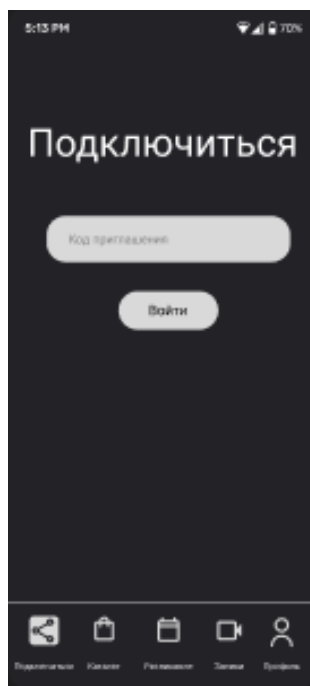


Рисунок 22 форма экрана подключения

3.3.7 Форма экрана с курсами

На данном экране расположены курсы. Имеется возможность подписаться и отписаться от курсов.

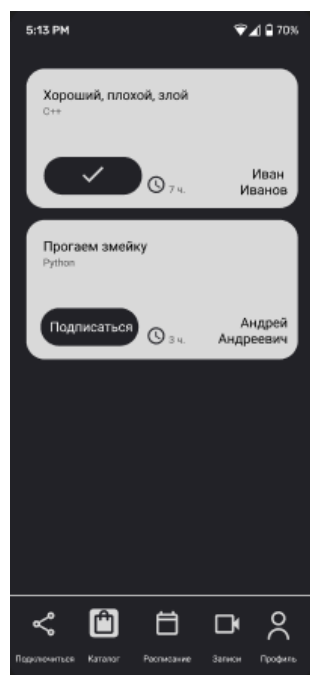


Рисунок 23 форма экрана с каталогом курсов

3.3.8 Форма экрана расписания



Рисунок 24 форма экрана с расписанием

На данном экране курсы и дата их проведения для отслеживания нужных вебинаров. Форма экрана с расписанием.

3.3.9 Форма экрана с записями трансляций.

На этом экране расположены поля с трансляциями. По нажатию кнопки можно посмотреть в записи нужную нам трансляцию.



Рисунок 25 форма экрана с записями трансляций

3.3.10 Форма экрана с занятиями (для преподавателя)

На этом экране можно добавить занятие в свой преподавательский курс. После того, как нажата кнопка «Добавить занятие», появляется экран с созданием занятия.



Рисунок 26 Форма экрана с добавлением занятий в свои курсы.

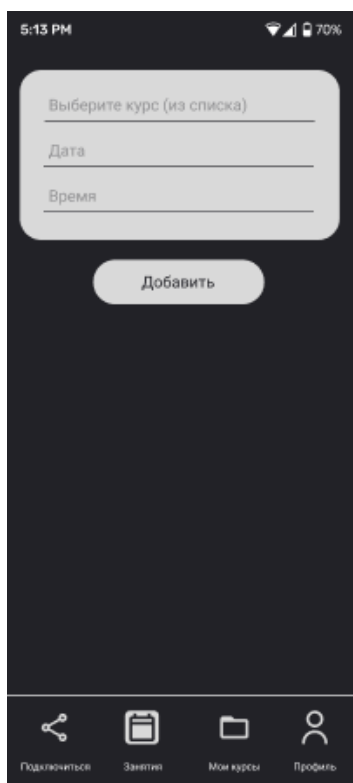


Рисунок 27 форма экрана с созданием задания для курса.

3.3.11 Форма экрана с размещением курса (для преподавателя)

На экране расположена форма экрана для создания преподавательского курса. Необходимые поля для ввода: название, длительность курса, направление, стоимость.

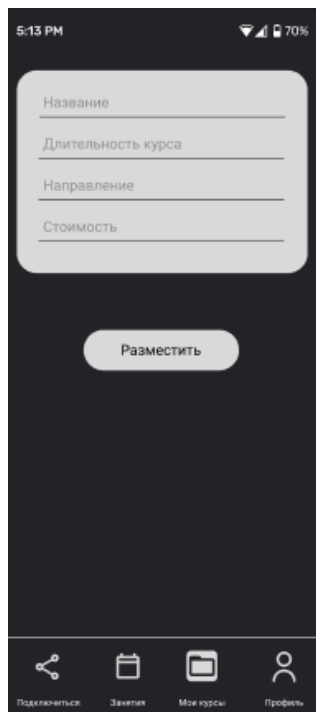


Рисунок 28 форма экрана с созданием преподавательского курса.

3.3.12 Форма экрана с вебинаром

На этом экране происходит проведение прямой трансляций. Имеется возможность отправить сообщение в чат, а также галочка для отправки анонимного сообщения для студента.

Можно просматривать участников при нажатии на соответствующую кнопку.

От лица преподавателя можно модерировать чат, удаляя нежелательные сообщения.

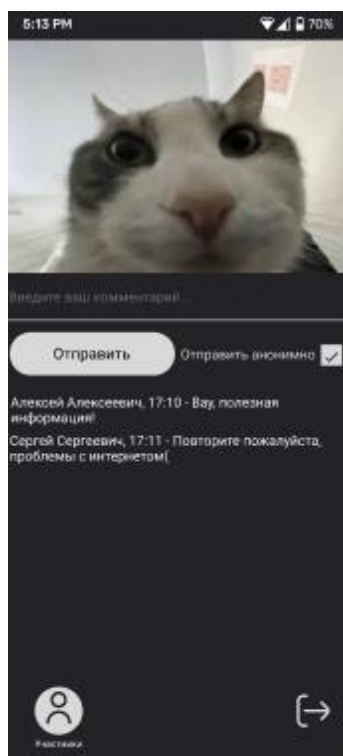


Рисунок 29 Форма экрана с трансляцией (студент).

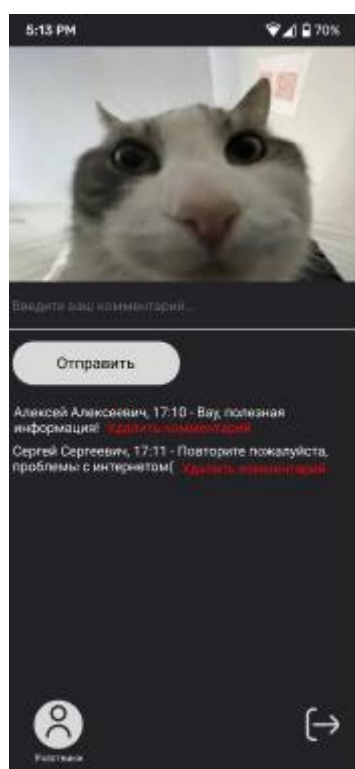


Рисунок 30 Форма экрана с трансляцией (преподаватель).

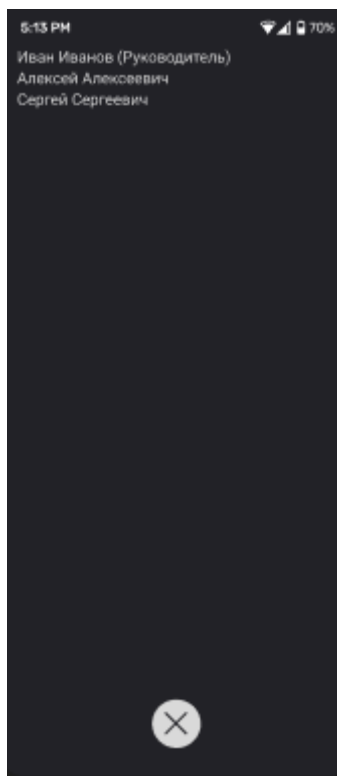


Рисунок 31 Форма экрана с просмотром участников.

3.4 Серверная часть

В качестве языка был выбран строго типизированный объектно-ориентированный язык программирования Kotlin. Он остается очень популярным языком программирования в этой мобильной разработке благодаря своим мощным возможностям и широкому спектру инструментов для разработки. К тому же существует достаточное количество библиотек, написанных на Kotlin, которые в перспективе можно легко интегрировать в проект.

3.4.1 Архитектура серверной части приложения

Серверная часть приложения реализована соответственно многоуровневой архитектуре. Она разделяет приложение на несколько слоев или уровней, каждый из которых отвечает за определенные аспекты приложения, такие как представление, бизнес-логика, доступ к данным и т.д. Это позволяет улучшить разделение ответственности, упростить тестирование и отладку, а также сделать приложение более гибким и масштабируемым.

4 Аналитика

Используя инструмент AppMetrica были созданы воронки для анализа количества пользователей, которые, открыв приложение, прошли весь пользовательский путь по регистрации и приложению.

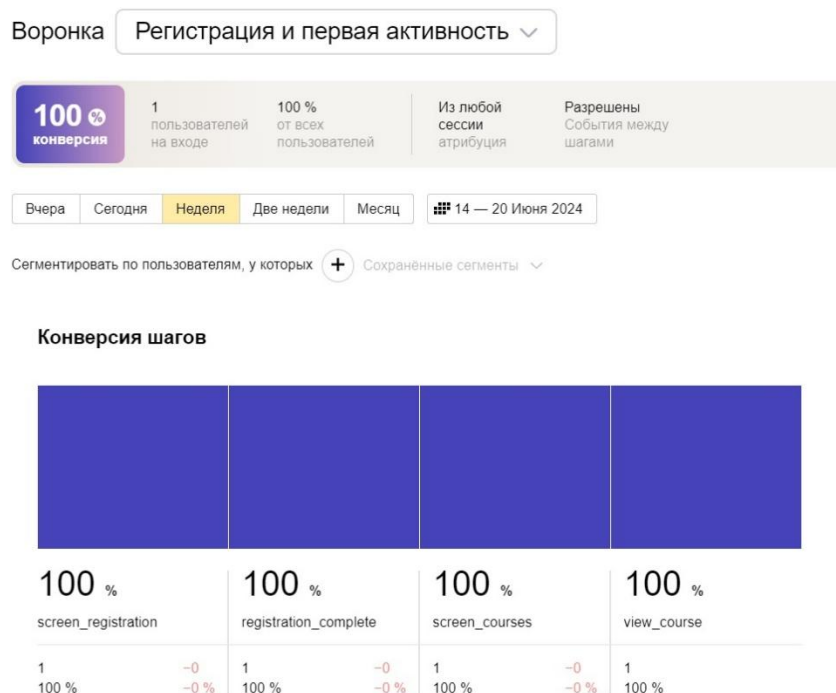


Рисунок 32 Воронка «Регистрация и первая активность»

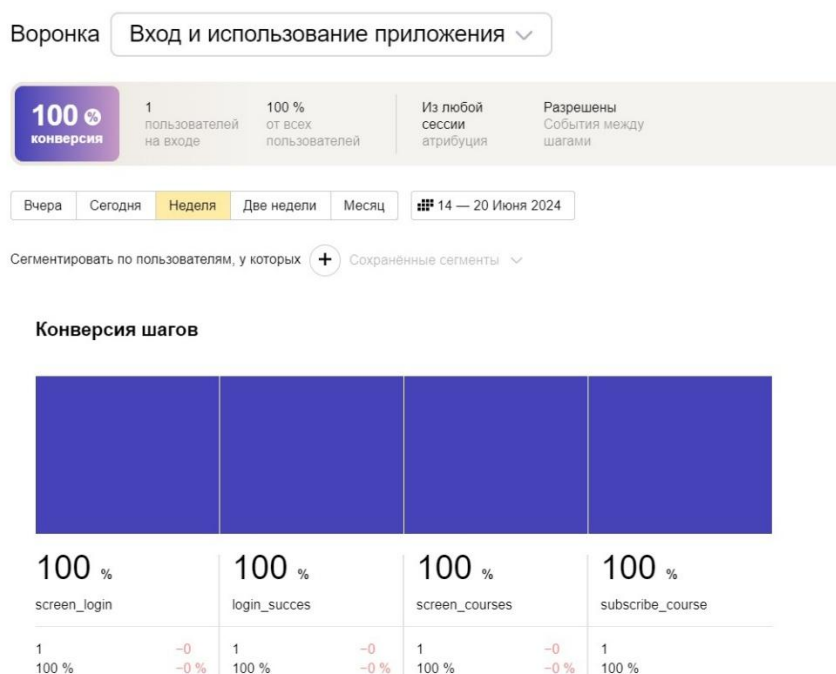


Рисунок 33 Воронка «Вход и использование приложения»

Заключение

В ходе выполнения данного курсового проекта был выполнен анализ предметной области и аналогов разрабатываемого приложения.

Для разработки приложения были разработаны макеты интерфейса, выбрана платформа приложения, построены UML диаграммы.

Для контроля версий был создан репозиторий GitHub.

При разработке приложения были реализованы следующие задачи:

- Регистрация и авторизация пользователей;
- Создание и редактирование курсов;
- Подписка на курсы;
- Проведение вебинаров (трансляций);
- Просмотр трансляций;
- Общение с преподавателем через чат;
- Просмотр записей трансляций;
- Модерация чата;
- Добавление и редактирование пользователей (студентов и преподавателей);
- Добавление и редактирование курсов;
- Удаление трансляций и записей.

Backend часть приложения и база данных были размещены в контейнере Docker compose на хостинге.

Разработанное приложение удовлетворяет поставленным требованиям.

Все поставленные задачи были выполнены.