

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

Факультет компьютерных наук

Кафедра информационных систем

Мобильное приложение «BrainCon»

Курсовой проект по дисциплине
«Технологии программирования»

09.03.02 Информационные системы и технологии
Информационные системы и сетевые технологии

Зав. кафедрой	_____ к.т.н., доцент Д. Н. Борисов
Обучающийся	_____ ст. 3 курса оч. отд. В. В. Малышев
Обучающийся	_____ ст. 3 курса оч. отд. Д. В. Полуянов
Обучающийся	_____ ст. 3 курса оч. отд. Е. В. Бизин
Руководитель	_____ С. И. Герасимов, преподаватель
Руководитель	_____ В.С. Тарасов, ст. преподаватель _____.20__

Воронеж 2024

Содержание

Содержание.....	2
Введение.....	4
1 Постановка задачи.....	4
1.1 Требования к разрабатываемой системе	4
1.1.1 Обзор общих требований к приложению и его функциональных характеристик.....	5
1.1.2 Требования к программному обеспечению приложения	5
1.1.3 Технологический обзор: Spring, Android.....	5
2 Анализ предметной области	9
2.1 Терминология (гlossарий) предметной области	9
2.2 Обзор аналогов	10
2.2.1 Skillbox	10
2.2.2 GeekBrains.....	14
3 Реализация.....	15
3.1 Функциональные требования к разрабатываемой системе	16
3.1.1 Регистрация и аутентификация	16
3.1.2 Просмотр курсов и материалов	16
3.1.3 Избранное	16
3.1.4 Прохождение тестов	17
3.1.5 Общение с менторами	17
3.1.6 Управление профилем.....	17
4 Диаграммы	17
4.1 Диаграмма прецедентов для клиента (Use-case diagram)	17
4.2 Диаграмма классов (Class diagram)	18
4.3 Диаграмма деятельности (Activity diagram).....	19
4.4 Диаграмма последовательности (Sequence diagram).....	21
4.5 Диаграмма состояний (Statechart diagramm)	23
4.6 ER–диаграмма	24
Заключение	25

Список использованной литературы.....	27
---------------------------------------	----

Введение

В современном мире обмен знаниями и опытом играет ключевую роль в развитии общества и личностного роста. Особенно в контексте образования, где доступ к экспертам и возможность общения с опытными профессионалами может значительно обогатить учебный процесс и расширить горизонты студентов. В силу этого возникает необходимость в создании эффективных инструментов, которые смогут облегчить процесс обмена знаниями и опытом между студентами и опытными менторами.

В этом контексте разработка мобильного приложения "BrainCon" представляет собой актуальное и перспективное направление. Данное приложение предполагает создание удобного и доступного пространства, которое будет преодолевать барьеры, связанные с социально-экономическим статусом географическим положением и другими ограничениями. В нем студенты смогут находить менторов, обмениваться знаниями, задавать вопросы и получать обратную связь от опытных специалистов в различных областях.

1 Постановка задачи

Цель данной курсовой работы заключается в анализе, проектировании и разработке мобильного приложения для обмена опытом и знаниями между студентами и менторами. В рамках данного исследования будет проведен анализ существующих аналогов, выявлены основные потребности пользователей, спроектированы функциональные и нефункциональные характеристики приложения, а также разработан прототип и реализовано его программное обеспечение.

1.1 Требования к разрабатываемой системе

1.1.1 Обзор общих требований к приложению и его функциональных характеристик

Согласно техническому заданию, приложение позволит пользователям просматривать содержимое материалов, редактировать и удалять эти материалы, а также оценивать их. Оно также должно поддерживать функции регистрации и входа для различения между анонимными и зарегистрированными пользователями.

1.1.2 Требования к программному обеспечению приложения

Требования к программному обеспечению клиентской части:

- Приложение должно устанавливаться и работать на мобильных устройствах под управлением операционной системы Android.

Требования к программному обеспечению серверной части:

- Серверная часть приложения должна быть реализована на языке программирования Java с использованием фреймворка Spring.

В качестве СУБД необходимо использовать PostgreSQL.

1.1.3 Технологический обзор: Spring, Android

Spring

1) Архитектура:

Spring Framework основан на принципах модульности и расширяемости, что делает его очень гибким и удобным для разработки приложений. Архитектура Spring состоит из следующих ключевых компонентов:

- **Core Container:** Основа фреймворка, включает в себя модули, такие как Beans, Core, Context и Expression Language (SpEL). Он предоставляет базовые функции IoC (Inversion of Control) и DI (Dependency Injection).
- **Data Access/Integration:** Содержит модули, такие как JDBC, ORM (Object-Relational Mapping), OXM (Object XML Mapping), JMS (Java

Message Service) и Transaction Management. Эти модули обеспечивают доступ к данным из различных источников и интеграцию с внешними системами.

- **Web:** Включает модули Spring Web MVC, WebSocket, Web и Portlet. Spring MVC предоставляет мощный инструмент для создания веб-приложений на основе архитектурного шаблона MVC (Model-View-Controller).
- **AOP (Aspect-Oriented Programming):** Позволяет разделять перекрестные аспекты приложения, такие как транзакции, аудит и безопасность, на отдельные компоненты и повторно использовать их в различных частях приложения.
- **Testing:** Содержит модуль Test, который облегчает написание тестов для приложений на основе Spring.

2) Принципы работы:

- **Inversion of Control (IoC) / Dependency Injection (DI):** Одним из основных принципов Spring является IoC, который делегирует управление жизненным циклом и зависимостями объектов контейнеру Spring. DI предполагает внедрение зависимостей объекта через конструктор, метод или поле, что делает классы менее связанными и более гибкими.
- **Аспектно-ориентированное программирование (AOP):** Spring поддерживает AOP, позволяя разделять аспекты приложения, такие как логирование, транзакции и безопасность, на отдельные модули и применять их к различным частям приложения.
- **Конфигурация через XML или аннотации:** Spring позволяет конфигурировать приложение как с помощью XML-файлов конфигурации, так и с помощью аннотаций. Это позволяет разработчикам выбирать подход, который лучше всего соответствует их потребностям и предпочтениям.

- **Принципы SOLID:** Spring поощряет применение принципов SOLID (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) для создания легко поддерживаемого и масштабируемого кода.
- **Поддержка тестирования:** Spring Framework облегчает написание модульных, интеграционных и функциональных тестов благодаря своей модульной архитектуре и поддержке механизмов тестирования.

Android

1) Архитектура:

Фреймворк Android основан на архитектуре, основанной на компонентах, которая позволяет разработчикам создавать масштабируемые и гибкие приложения. Основные компоненты архитектуры Android включают в себя:

- **Activity:** Это основной компонент пользовательского интерфейса в Android. Каждая активность представляет собой отдельный экран приложения, с которым пользователь может взаимодействовать.
- **Fragment:** Фрагменты представляют собой модульные части пользовательского интерфейса, которые могут быть использованы внутри активности. Они облегчают создание гибких и адаптивных макетов приложения.
- **Service:** Службы выполняют длительные операции в фоновом режиме, даже когда пользователь не взаимодействует с приложением. Они могут выполняться как в том же процессе, что и приложение, так и в отдельном процессе.
- **Broadcast Receiver:** Эти компоненты позволяют приложению реагировать на системные события или сообщения от других приложений. Они могут получать широковещательные сообщения и предпринимать соответствующие действия.

— **Content Provider:** Предоставляет единый интерфейс для управления данными приложения. Они позволяют другим приложениям получать доступ к данным и даже модифицировать их при наличии соответствующих разрешений.

2) Принципы работы:

— **Model-View-Controller (MVC):** Хотя Android не является строго MVC-фреймворком, он поддерживает архитектурный шаблон MVC. Для разделения логики приложения от пользовательского интерфейса разработчики используют классы модели (Model), классы представления (View) и классы контроллера (Controller).

— **Activity Lifecycle:** Активности в Android имеют жизненный цикл, который включает в себя различные состояния, такие как создание, запуск, приостановка, возобновление и уничтожение. Разработчики могут переопределить методы жизненного цикла активности, чтобы управлять её поведением в различных состояниях.

— **Фрагменты и управление макетами:** Фрагменты позволяют разделить пользовательский интерфейс на независимые компоненты, что облегчает создание гибких и адаптивных макетов. Управление макетами осуществляется через XML-ресурсы или программно.

— **Адаптеры и списковые представления:** Для отображения списков данных в приложении используются списковые представления (ListView, RecyclerView) в сочетании с адаптерами. Адаптеры связывают данные с представлениями и обеспечивают их отображение.

— **Асинхронная работа и потоки:** в Android для выполнения длительных операций в фоновом режиме рекомендуется использовать асинхронные задачи (AsyncTask), потоки или службы.

— **Манифест приложения:** в манифесте приложения (AndroidManifest.xml) описываются все компоненты приложения, его настройки, разрешения и зависимости от внешних ресурсов.

2 Анализ предметной области

2.1 Терминология (гlossарий) предметной области

Пользователь — неавторизованный пользователь сайта, не имеющий полного доступа к функциям сайта.

Авторизованный пользователь (студент, ментор) – авторизованный пользователь сайта, имеющий полный доступ к функциям сайта.

Студент — пользователь приложения BrainCon, зарегистрированный для прохождения курсов. Студенты могут искать и выбирать курсы, добавлять их в избранное, получать доступ к учебным материалам, задавать вопросы менторам.

Ментор — специалист, зарегистрированный в приложении BrainCon, который помогает студентам в процессе обучения. Менторы создают и ведут курсы, отвечают на вопросы студентов, проводят вебинары и дают обратную связь по выполненным заданиям, способствуя развитию навыков и знаний студентов.

Курсы — вкладка в приложении BrainCon, где пользователи могут просматривать все доступные образовательные программы.

Тесты — раздел внутри обучающих курсов в приложении BrainCon, предназначенный для оценки знаний и навыков обучающихся.

Профиль (аккаунт) — страница пользователя приложения с персональными данными о нем.

Backend (серверная часть) – программно-аппаратная часть сервиса, которая хранится на сервере, обрабатывает полученные данные и отправляет ответ обратно.

Frontend (клиентская сторона) – интерфейс с набором функций, с которым взаимодействует пользователь.

Избранное — функция в приложении BrainCon, позволяющая пользователям сохранять любимые курсы для быстрого доступа.

2.2 Обзор аналогов

Прежде, чем начинать разработку приложения BrainCon, необходимо проанализировать уже имеющиеся платформы для понимания их плюсов и минусов. После сбора информации можно переходить к этапу разработки с учетом сделанных выводов по изученным аналогам.

2.2.1 Skillbox

Skillbox - это образовательная платформа, специализирующаяся на курсах по цифровым навыкам, веб-разработке, дизайну, маркетингу, анимации, программированию и другим смежным областям. Платформа предлагает онлайн-курсы как для начинающих, так и для опытных специалистов, а также корпоративные обучающие программы.

Курсы на платформе Skillbox разработаны экспертами и практикующими специалистами в соответствующих областях. Они представлены в различных форматах, включая видеоуроки, учебные материалы, практические задания и проекты. Большинство курсов на Skillbox ориентированы на практическое применение полученных знаний в реальных проектах.

Отсутствие мобильного приложения у Skillbox ограничивает возможности студентов получать доступ к образовательным материалам и курсам в любое время и в любом месте через свои мобильные устройства. Это может быть особенно неудобно для занятых людей, которые не всегда могут использовать компьютер или ноутбук для обучения. Вместо этого, студенты могут чувствовать себя ограниченными в своих возможностях обучения, когда доступ к платформе ограничен только через мобильный браузер или стационарный компьютер. Веб-сайт Skillbox может быть неоптимизирован для мобильных устройств, что может привести к неудобству при просмотре материалов курсов или выполнении заданий на небольших экранах.

Так же стоит отметить, что при использовании платформой в мобильном браузере становится очень явной проблема информационной перегруженности. Пользователям может быть сложно найти нужный курс или материал среди множества предложений. Это может затруднить фокусировку на конкретных темах и усложнить процесс обучения.

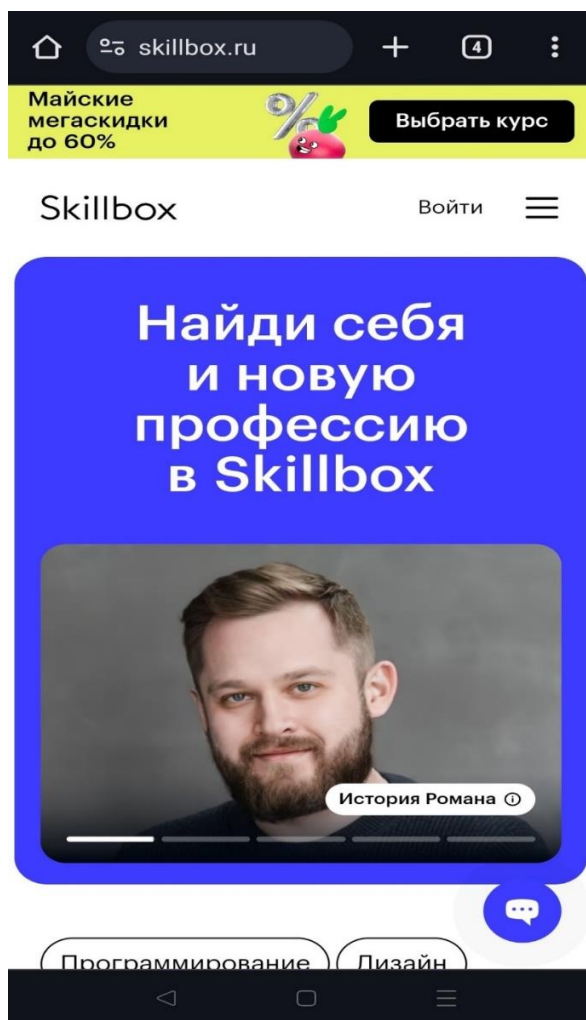


Рисунок 1 - Главная страница “Skillbox” в мобильном браузере

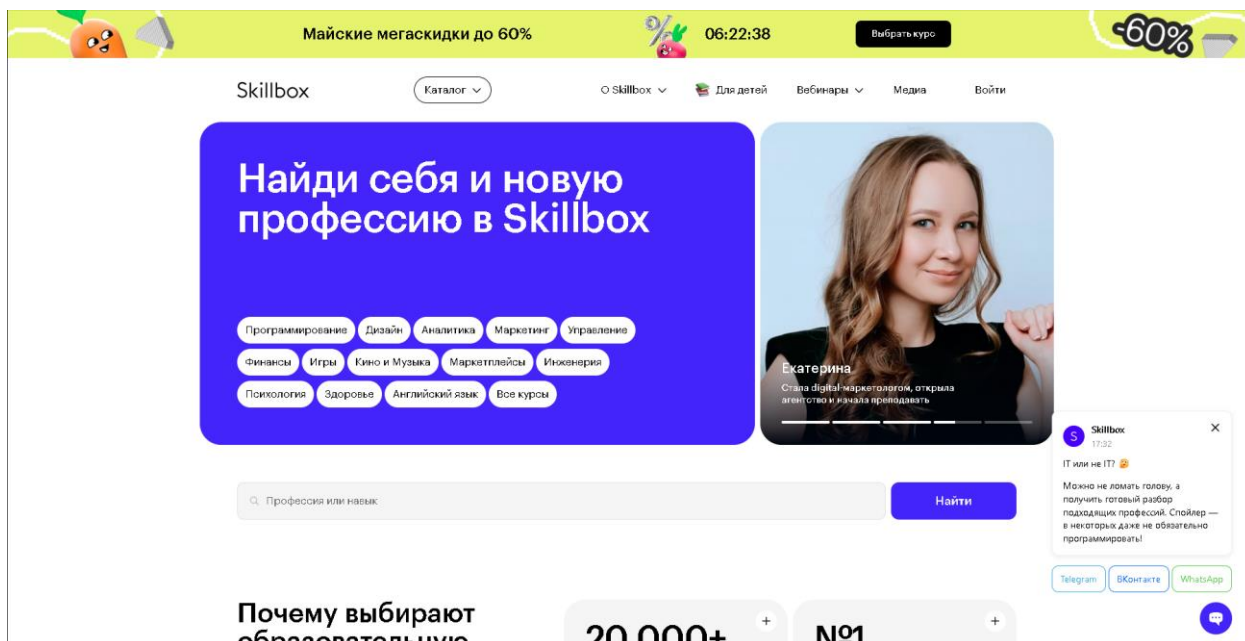


Рисунок 2 - Главная страница “Skillbox” на стационарном компьютере

Следующим недостатком является практика непрозрачной бизнес-модели. Для того, чтобы объяснить суть, мы провели небольшое расследование.

На данной платформе вам могут предложить первые полгода обучения бесплатно, а монетизировать свои навыки можно будет уже после 4-х месяцев изучения курса. Однако, для того, чтобы поступить на обучение, нужно заключить с банком кредитный договор:

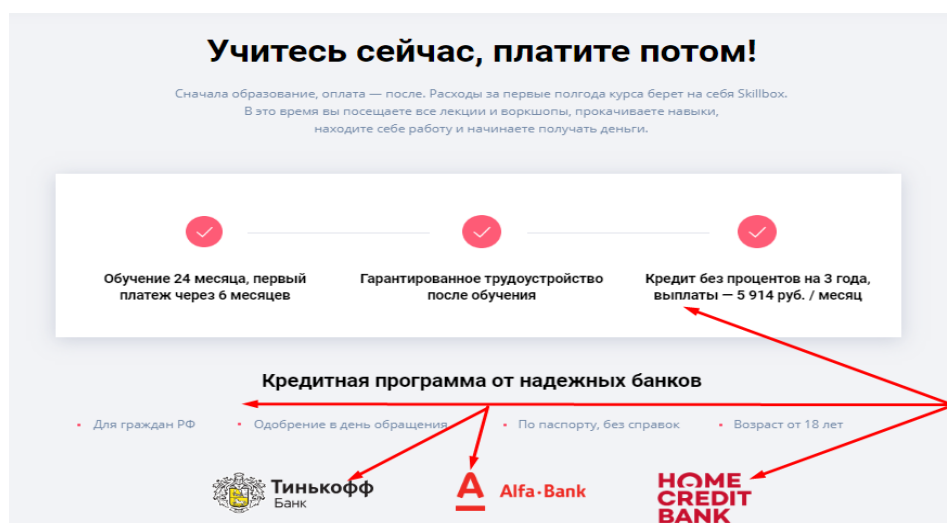


Рисунок 3 - Скриншот с платформы Skillbox в курсе “Профессия SMM”

Skillbox на словах рекламирует “кредит без процентов”, однако, записавшись на курс SMM и получив дополнительное соглашение, выявляется обратное:

Стороны договорились о нижеследующем:

1. Стоимость услуг составляет 00 000,00 (_____) рублей на основании Кредитного договора №____ от ____ г. с АО «НАЗВАНИЕ БАНКА».
2. Считать, что Сумма Вознаграждения Исполнителя в размере 00 000,00 (_____) рублей подлежит перечислению Заказчику на его банковские реквизиты в АО «НАЗВАНИЕ БАНКА» в течение 20 (двадцати) календарных дней с момента поступления денежных средств по программе курса «_____» на расчетный счет Исполнителя.
3. Заказчик за свой счет осуществляет выплату процентов, начисляемых банком за пользование кредитными средствами, сумма кредита и процентов по нему указываются в кредитном договоре между Заказчиком и банком.
4. При возникновении разногласий между условиями Договора и условиями настоящего Соглашения подлежат применению условия настоящего Соглашения.
5. Настоящее Соглашение составлено в 2 (двух) экземплярах, по одному для каждой из Сторон.
6. Настоящее Соглашение вступает в силу при соблюдении всех следующих условий: подписание Соглашения Сторонами; акцепт Оферты Заказчиком; получение Исполнителем средств от АО «НАЗВАНИЕ БАНКА» за оплату услуг Исполнителя за Заказчика. Соглашение действует в течение срока действия Договора оказания услуг.
7. Стороны согласовали, что Исполнитель может подписать Соглашение с использованием аналогов собственноручной подписи, в том числе факсимиле.

Рисунок 4 - Скриншот дополнительного соглашения со Skillbox

В дополнительном соглашении прописано условие о процентах по кредиту. Оплачивает их Заказчик, то есть студент.

Следующая проблема – стоимость обучения. На скриншоте ниже указана полная стоимость курса со скидкой и размер ежемесячного платежа:

Скриншот веб-страницы Skillbox. В центре экрана на красном фоне информация о курсе: скидка 40%, полная стоимость курса 305 572 Р, цена в рассрочку 5 914 Р/мес, сумма первого взноса 183 343 Р. Справа на белом фоне форма «Запишитесь на курс» с полем «Набор до: 15 марта», списком преимуществ (первая оплата через 6 месяцев, 2 года обучения = 2 года опыта, гарантия трудоустройства, бессрочный доступ к материалам), полями для ввода имени, телефона и email, а также кнопкой «Записаться». Внизу формы мелким шрифтом указано: «Отправляя данную форму, вы даете согласие на обработку своих персональных данных».

Рисунок 5 - Скриншот с платформы Skillbox в курсе “Профессия SMM”

Если первая оплата предполагается через 6 месяцев, то оплата за 18 платных месяцев обучения составит $5914 \cdot 18 = 106452$ р. Если учитывать платежи за первые 6 месяцев, которые вносит банк, то всё равно не получается заявленная сумма. $5914 \cdot 24 = 141936$ р.

Кредит указан на 3 года. $5914 \cdot 36 = 212\,904$ р. это платеж за три года. Опять не совпадает.

Если первые полгода бесплатны, то все равно не сходится. $5914 \cdot 30 = 177\,420$ р.

2.2.2 GeekBrains

GeekBrains является одной из ведущих платформ для онлайн-обучения, предлагая разнообразные курсы в области IT, программирования, дизайна, маркетинга и других направлений. Несмотря на свои известные достижения, GeekBrains сталкивается с рядом серьезных проблем, которые могут затруднить обучение пользователей. Ниже приведен подробный анализ основных проблем, основанный на отзывах студентов данной платформы:

- Непредсказуемые изменения в расписании: Частые и непредсказуемые переносы занятий, введение каникул без предварительного уведомления, а также отмены или замены курсов приводят к тому, что студенты теряют мотивацию и могут не успевать освоить материал в запланированные сроки. Например, курс по "Алгоритмам и структурам данных на Python" был сначала перенесен, а затем вообще удален без уведомления студентов, что вызвало у них негативную реакцию.
- Кураторы и менеджеры часто не предоставляют своевременную и полезную информацию, ограничиваясь шаблонными ответами типа "Уточню у коллег" или "Передам коллегам ваши вопросы". Это создает у студентов ощущение, что их проблемы игнорируются, а поддержка отсутствует.
- Проблемы с юридическими аспектами: Так же, как и Skillbox, GeekBrains требует предоплату за обучение, включая оплату в кредит

под видом рассрочки, и не предоставляет ясной и понятной процедуры возврата средств в случае несоответствия услуг ожиданиям. Это вызывает у студентов необходимость обращаться в суд для защиты своих прав и возврата средств.

Таким образом, BrainCon стремится избежать перечисленные проблемы, обеспечивая удобный формат изучения материалов и эффективную поддержку от менторов, а также философия нашего проекта предполагает доступ к обучению для студентов вне зависимости от их местоположения и материального положения.

3 Реализация

Реализация BrainCon включает в себя использование различных технологий и инструментов, обеспечивающих функционирование приложения на серверной и клиентской сторонах, а также управление данными. Вот более подробное описание используемых средств:

- Java: Java является основным языком программирования для разработки серверной части приложения. Ее мощные инструменты и высокая производительность обеспечивают надежную работу приложения.
- Spring Framework: Spring Framework предоставляет не только инструменты для управления зависимостями и внедрения зависимостей, но и фреймворк для создания веб-приложений. С помощью Spring можно легко создавать веб-контроллеры, сервисы, обрабатывать запросы и управлять транзакциями базы данных.
- Android: Для клиентской части приложения используется платформа Android. Она обеспечивает удобный и интуитивно понятный пользовательский интерфейс, позволяя пользователям взаимодействовать с функционалом **BrainCon** на мобильных устройствах. Android также предоставляет доступ к различным функциям устройства, таким как камера, геолокация и другие.

- PostgreSQL: Для хранения данных приложения используется реляционная база данных PostgreSQL. Она обеспечивает надежное хранение и управление данными, поддерживает различные типы данных и операции, такие как добавление, изменение и удаление данных. PostgreSQL также обладает масштабируемостью и высокой производительностью, что позволяет обрабатывать большие объемы информации.

3.1 Функциональные требования к разрабатываемой системе

3.1.1 Регистрация и аутентификация

- Регистрация пользователя: Пользователь должен иметь возможность зарегистрироваться в приложении, указав свои данные, такие как имя, адрес электронной почты и пароль.
- Аутентификация пользователя: Зарегистрированные пользователи должны иметь возможность войти в систему, вводя свои учетные данные.

3.1.2 Просмотр курсов и материалов

- Просмотр доступных курсов: Пользователь должен иметь возможность просматривать список доступных курсов, включая их описания, длительность и стоимость.
- Просмотр материалов курсов: После выбора курса пользователь должен иметь доступ к учебным материалам, таким как лекции, видеоуроки, задания и т. д.

3.1.3 Избранное

- Добавление курсов в избранное: Пользователь должен иметь возможность добавлять курсы в список "Избранное" для удобного и быстрого доступа к ним.
- Просмотр избранных курсов: Пользователь должен иметь доступ к списку избранных курсов в отдельной вкладке приложения.

3.1.4 Прохождение тестов

- Прохождение встроенных тестов: После изучения материалов курса пользователь должен иметь возможность проходить тесты для проверки своих знаний и навыков.
- Получение результатов тестов: Пользователь должен получать обратную связь о результатах прохождения тестов, включая правильные и неправильные ответы.

3.1.5 Общение с менторами

- Задавание вопросов менторам: Пользователь должен иметь возможность задавать вопросы и просить совета у менторов по темам курсов.
- Получение обратной связи: Пользователь должен получать обратную связь от менторов по своим вопросам и выполненным заданиям.

3.1.6 Управление профилем

- Редактирование профиля: Пользователь должен иметь возможность редактировать свой профиль, включая имя, адрес электронной почты, пароль и другие данные.

4 Диаграммы

4.1 Диаграмма прецедентов для клиента (Use-case diagram)

Диаграмма прецедентов представляет собой диаграмму, которая моделирует функциональность системы, показывая ее взаимодействие с пользователями, внешними сущностями, которые взаимодействуют с системой. Диаграмма прецедентов фокусируется на функциональных возможностях системы.

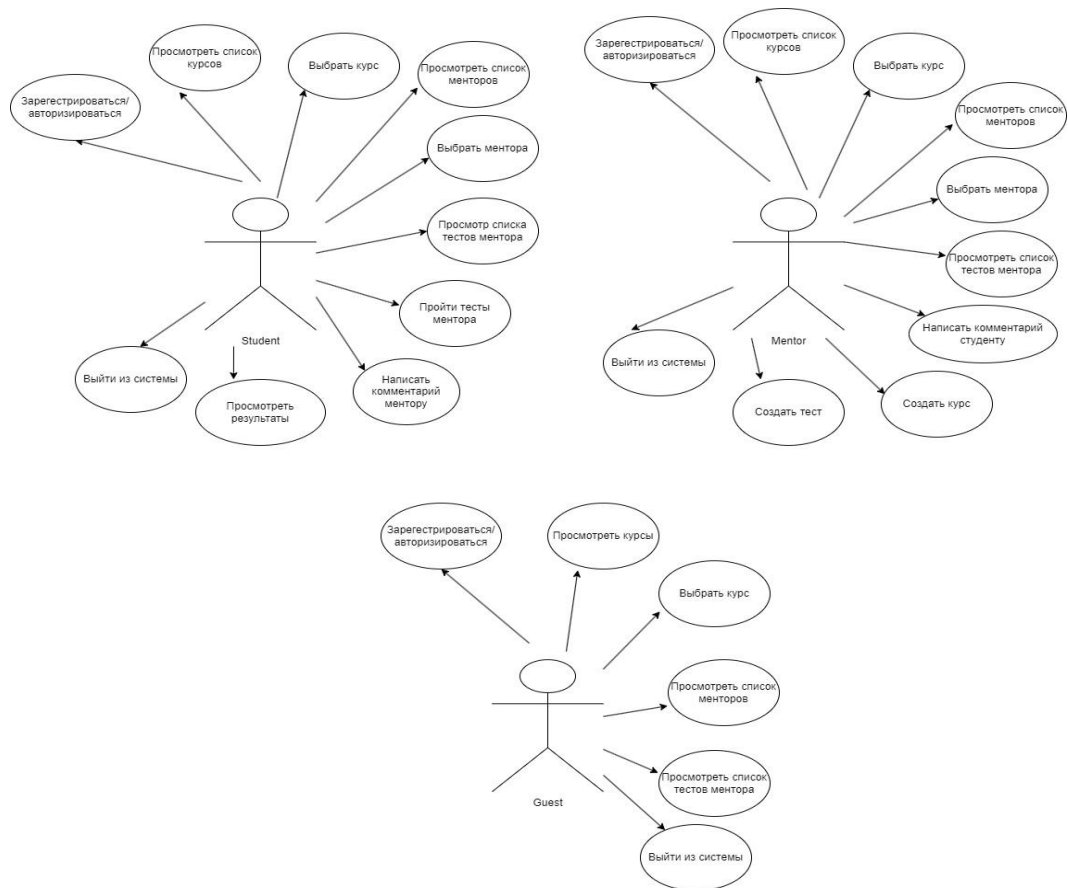


Рисунок 6 - Use-case diagram пользователей

4.2 Диаграмма классов (Class diagram)

Диаграмма классов представляет собой графическую модель, которая отображает структуру системы, описывая классы, их атрибуты, методы и взаимоотношения между классами.

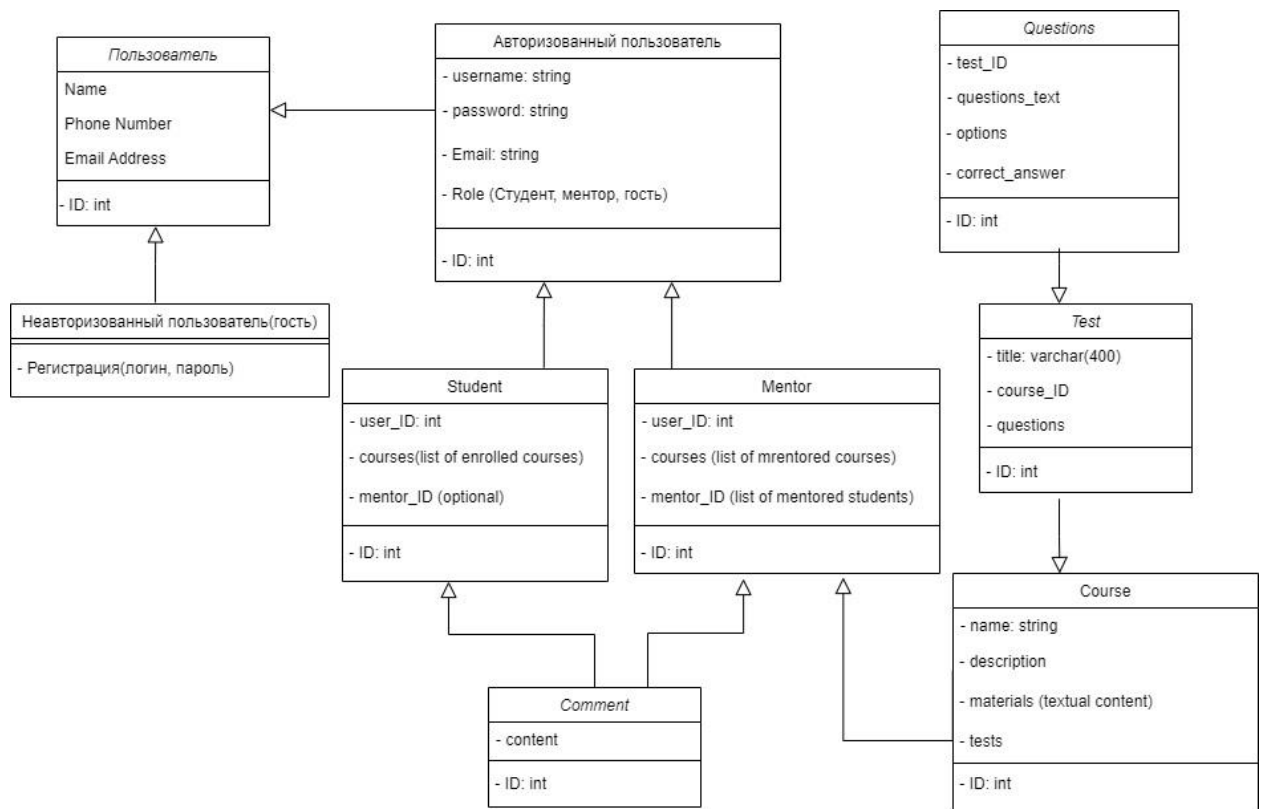


Рисунок 7 - Class diagram

4.3 Диаграмма деятельности (Activity diagram)

Диаграмма деятельности представляет собой графическую модель, которая позволяет описывать последовательность действий, процессы и поведение системы. Эта диаграмма используется для моделирования бизнес-процессов, алгоритмов и составных частей системы.

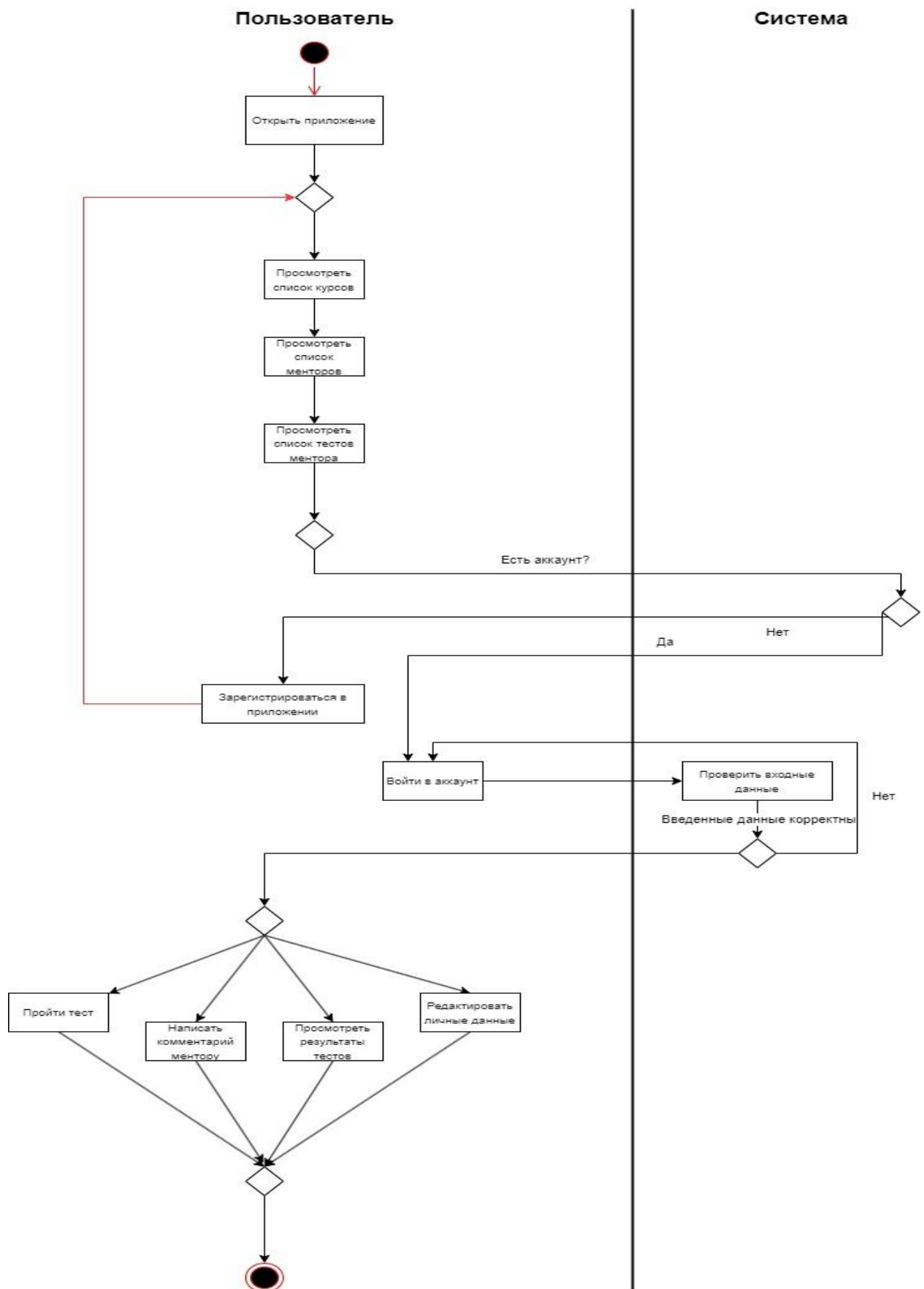


Рисунок 8 - Activity diagram

4.4 Диаграмма последовательности (Sequence diagram)

Диаграмма последовательности представляет собой диаграмму, которая моделирует взаимодействие между объектами в определенной последовательности времени. Она позволяет показать, как объекты обмениваются сообщениями и взаимодействуют друг с другом для выполнения определенной функциональности.

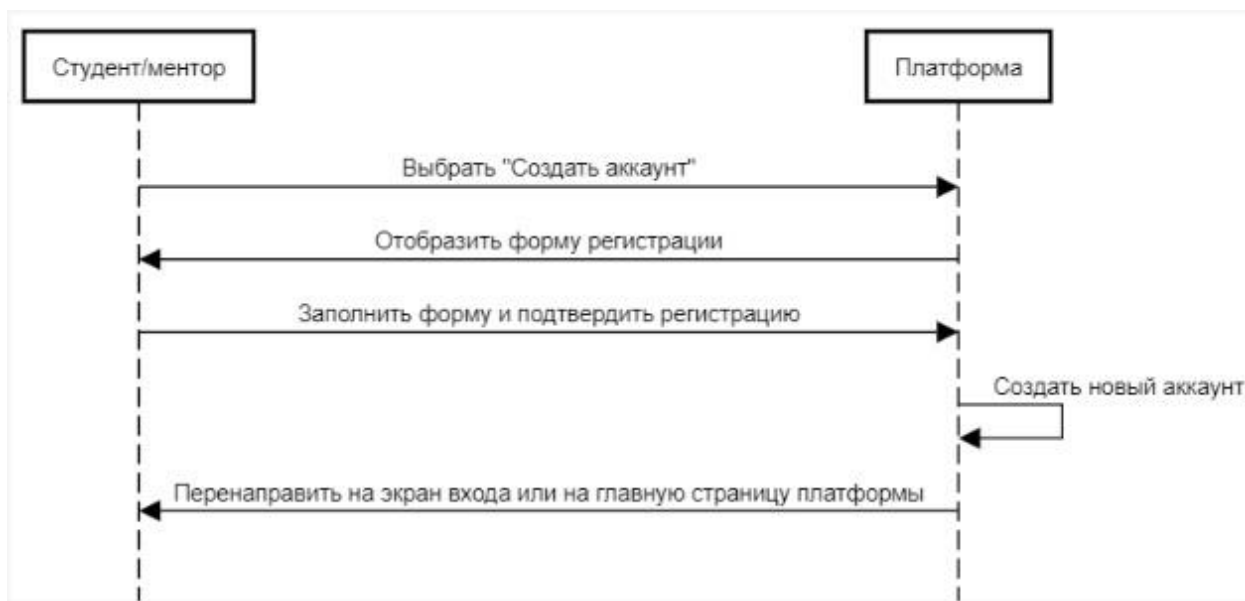


Рисунок 9 - Sequence diagram 1

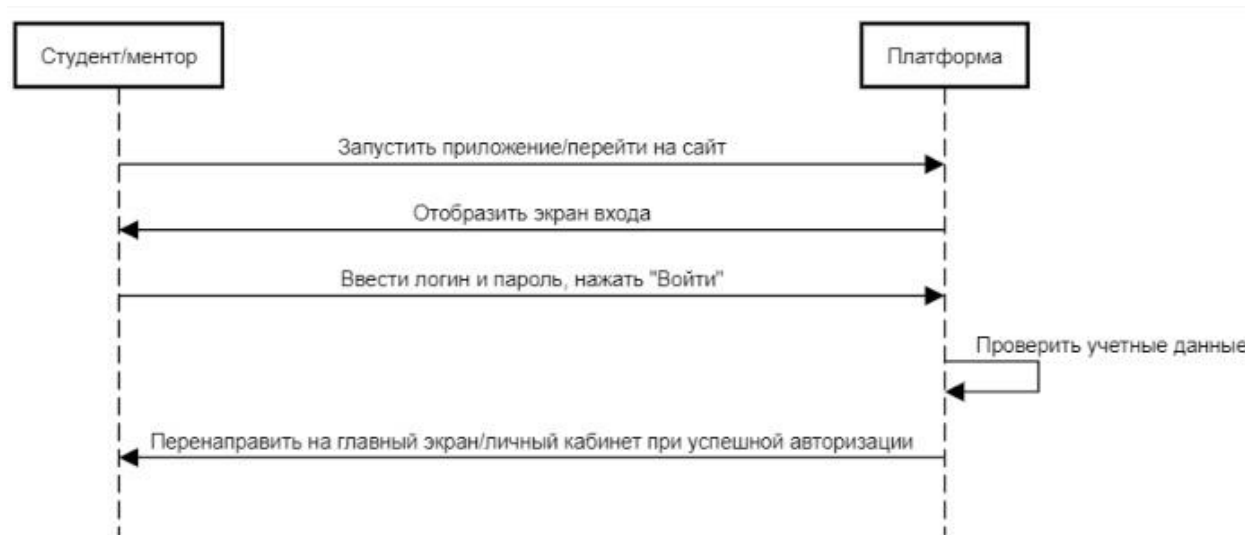


Рисунок 10 - Sequence diagram 2

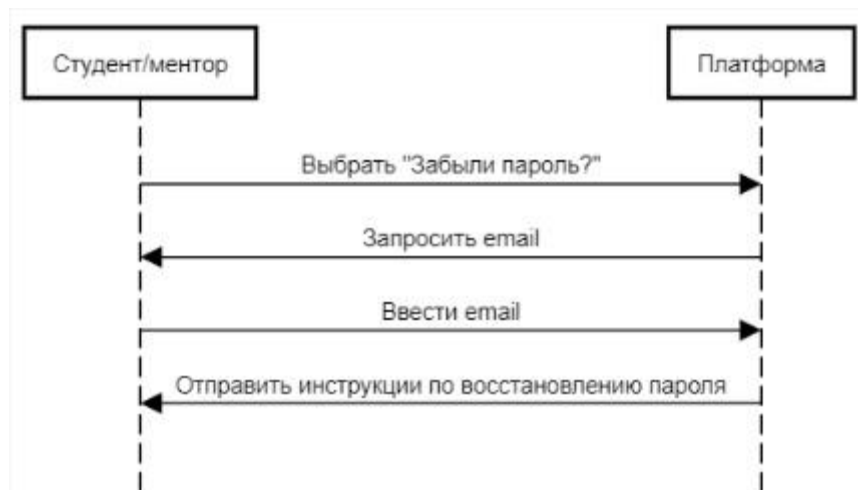


Рисунок 11 - Sequence diagram 3

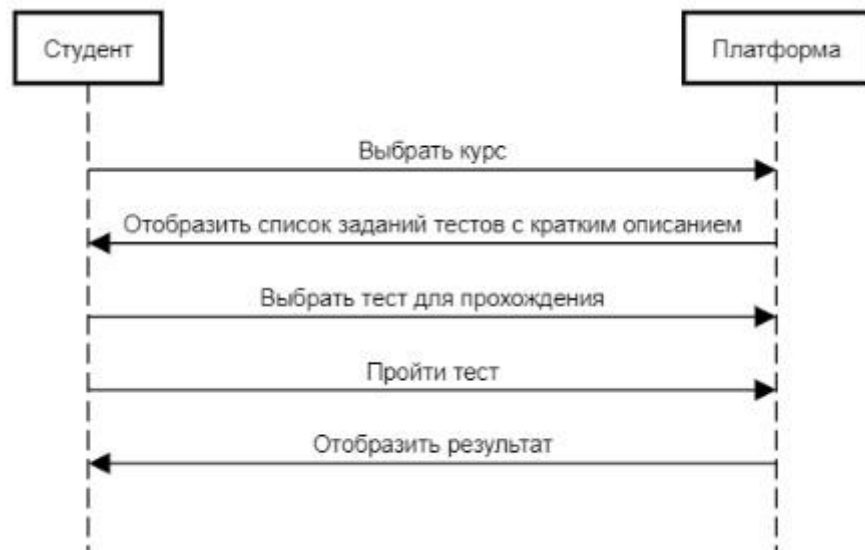


Рисунок 12 - Sequence diagram 4



Рисунок 13 - Sequence diagram 5

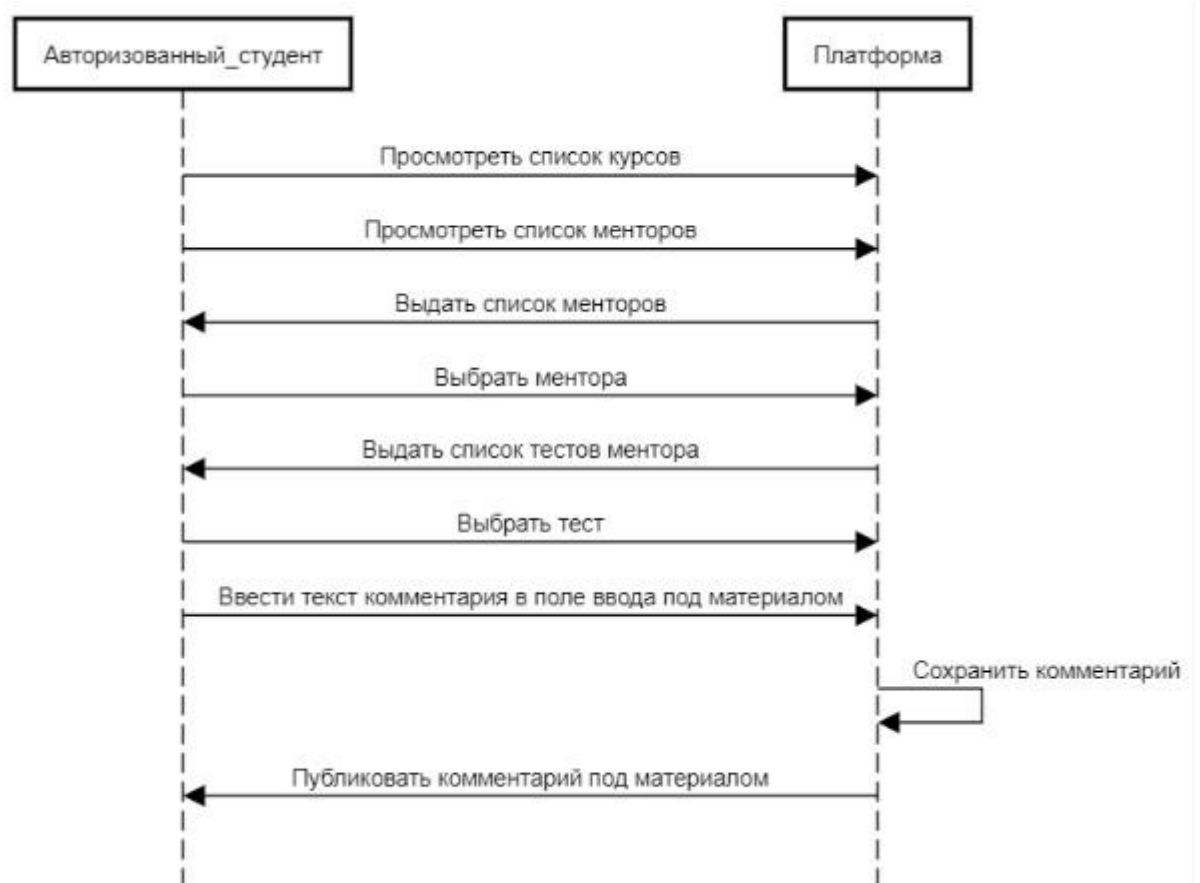


Рисунок 14 - Sequence diagram 6

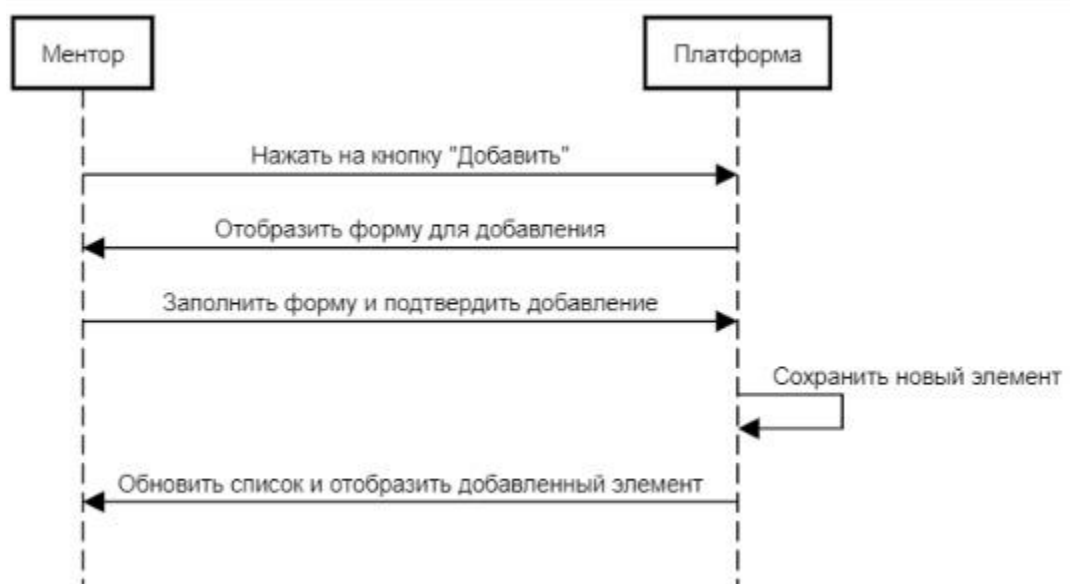


Рисунок 15 - Sequence diagram 7

4.5 Диаграмма состояний (Statechart diagramm)

Диаграмма состояний представляет собой диаграмму, которая моделирует поведение объекта или системы в различных состояниях и

переходах между этими состояниями. Она описывает, как объект или система реагирует на различные события, изменяя свое состояние.

На рисунке 13 изображена диаграмма состояния системы корпоративного сайта IT компании.

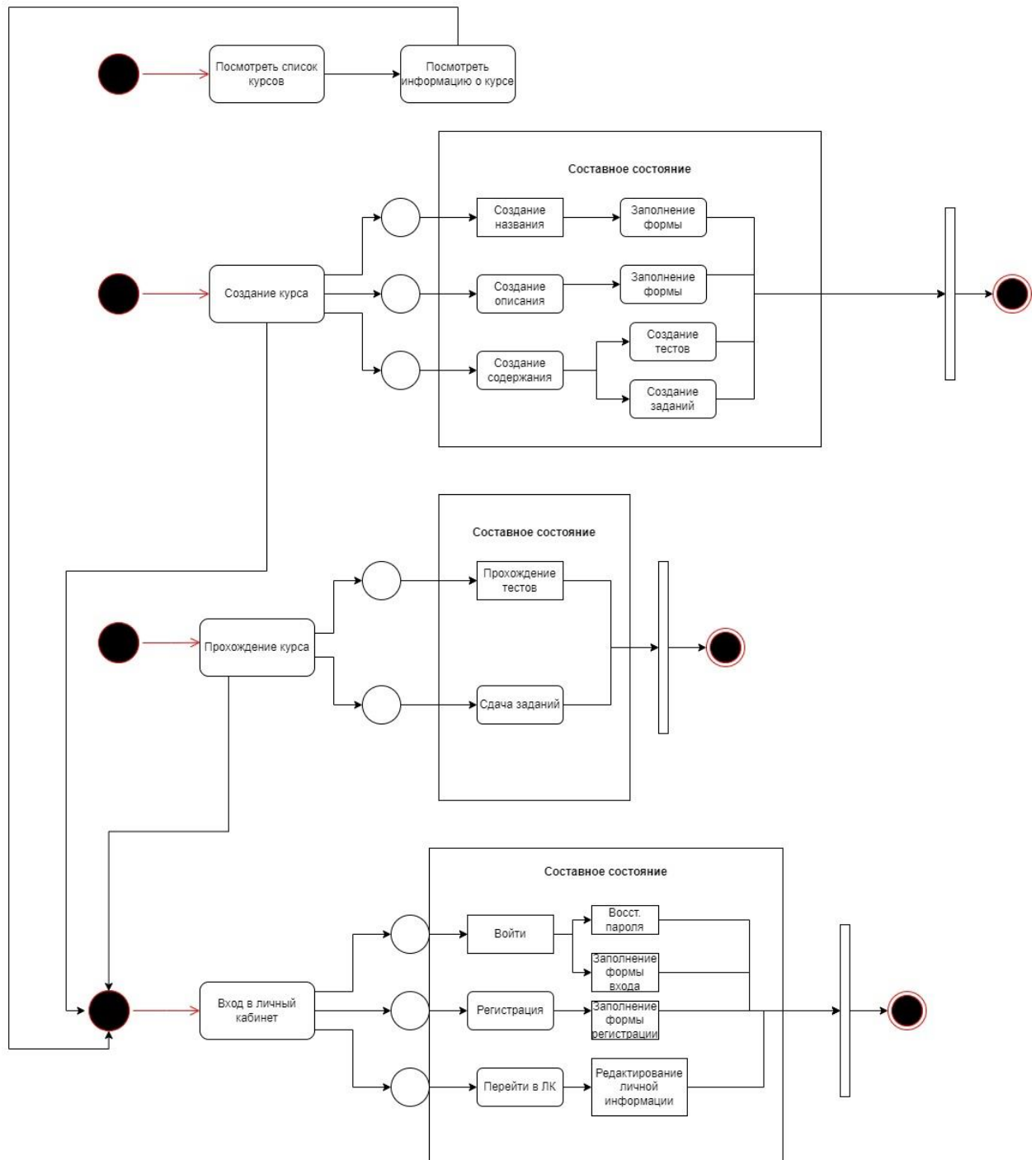


Рисунок 16 - Statechart diagram

4.6 ER-диаграмма

ER-диаграмма - это графическая модель, используемая в базах данных для описания сущностей и их взаимосвязей.

На рисунке 14 изображена ER-диаграмма системы корпоративного сайта IT компании.

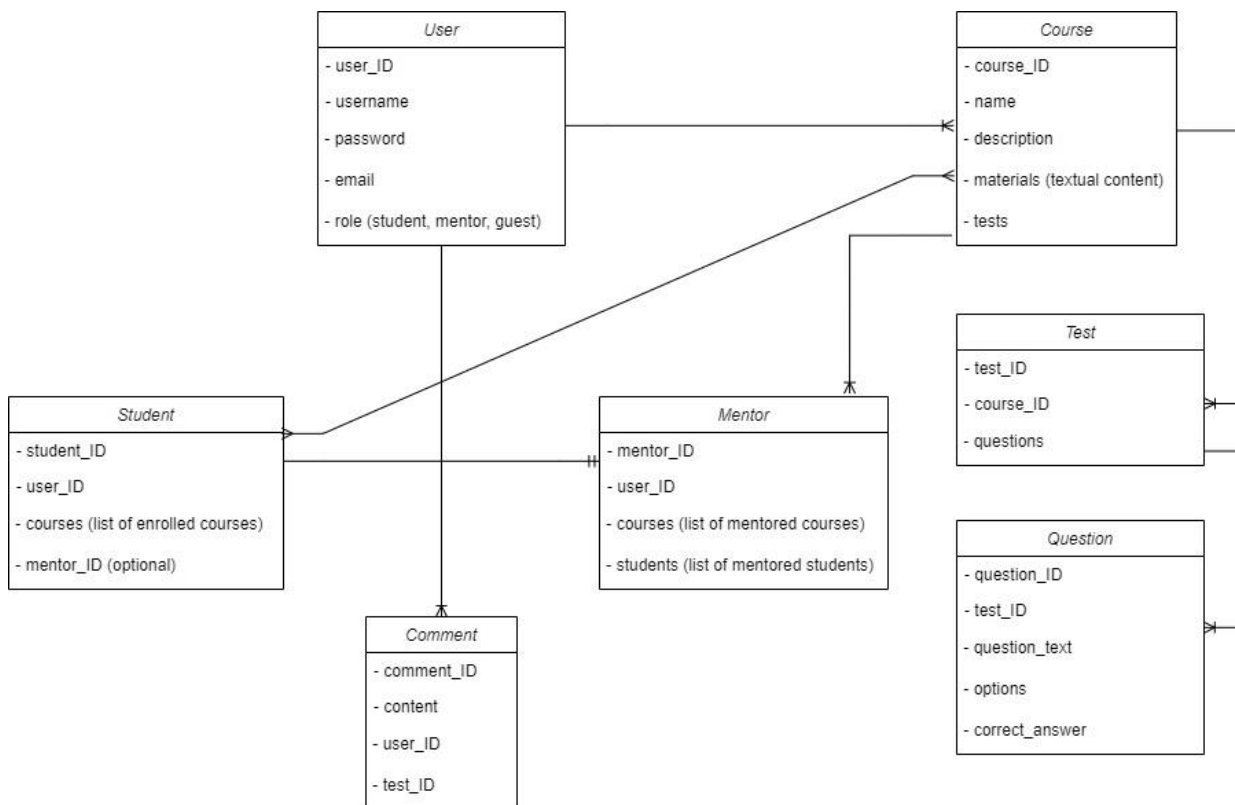


Рисунок 17 - ER-диаграмма

Заключение

В ходе данной курсовой работы была разработана система для создания онлайн-платформы обмена опытом и знаниями между студентами и менторами в различных областях. Основной целью проекта было создание удобного и функционального ресурса, который позволяет пользователям легко и быстро находить менторов, обмениваться опытом и получать качественные консультации. Была реализована система авторизации и регистрации пользователей, возможность просмотра профилей менторов и их компетенций, просмотра доступных курсов и программ обучения, возможность оценки и отзыва о менторе, а также функционал публикации и просмотра полезных статей и рекомендаций.

Платформа была разработана с использованием клиент-серверной архитектуры, где клиенты взаимодействуют с веб-сервером через браузер. Были выполнены требования к функциональным и нефункциональным характеристикам, такие как интуитивно понятный интерфейс, удобная навигация, высокая скорость загрузки страниц и адаптивный дизайн, соответствующий потребностям и предпочтениям пользователей.

В результате разработки была создана система, которая значительно упрощает процесс поиска и получения консультаций от менторов в различных областях знаний. Это способствует активному обмену опытом и повышению качества обучения студентов, а также облегчает работу менторов, предоставляя им возможность эффективно делиться своим опытом и знаниями.

Список использованной литературы

1. Skillbox [Образовательный портал]. — <https://skillbox.ru/> — (Дата обращения: 24.04.2024).
2. GeekBrains [Образовательный портал]. — https://gb.ru/?utm_referrer=https%3A%2F%2Fyandex.ru%2F — (Дата обращения: 01.05.2024).
3. vc.ru [Статья “10 мифов о Skillbox, о которых должен знать каждый”]. — <https://vc.ru/life/219320-10-mifov-o-skillbox-o-kotoryh-dolzhen-znat-kazhdyy> — (Дата обращения: 30.04.2024).
4. dzen.ru [Статья “Почему я не пошла учиться в Skillbox. Три подвоха онлайн-университета.”]. — <https://dzen.ru/a/XmueVmlhvXnb64XH> — (Дата обращения: 30.04.2024).
5. pikabu.ru [Статья “GeekBrains царство ботов и воров”]. — https://pikabu.ru/story/geekbrains_tsarstvo_botov_i_vorov_9355186 — (Дата обращения: 03.05.2024).
6. vc.ru [Статья “GeekBrains отказали в возврате денежных средств.”]. — <https://vc.ru/claim/1178786-geekbrains-otkazali-v-vozvrate-denezhnyh-sredstv> — (Дата обращения: 03.05.2024).