

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

Факультет компьютерных наук

Кафедра информационных технологий управления

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

Введение.....	3
1 Постановка задачи.....	3
1.1 Требования к разрабатываемой системе	3
1.1.1 Обзор общих требований к приложению и его функциональных характеристик.....	3
1.1.2 Требования к программному обеспечению приложения	4
1.1.3 Технологический обзор: Spring, Android.....	4
2 Анализ предметной области	8
2.1 Терминология (гlossарий) предметной области	8
2.2 Обзор аналогов	8
2.2.1 Skillbox	8
3 Диаграммы	11
3.1 Диаграмма прецедентов для клиента (Use-case diagramm)	11
3.2 Диаграмма классов (Class diagramm).....	12
3.3 Диаграмма деятельности (Activity diagramm).....	13
3.4 Диаграмма последовательности (Sequence diagramm).....	15
3.5 Диаграмма состояний (Statechart diagramm)	17
3.6 ER–диаграмма	18
Заключение	19
Список использованной литературы.....	21

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, где студенты смогут находить менторов, обмениваться знаниями, задавать вопросы и получать качественную 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.

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.

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.

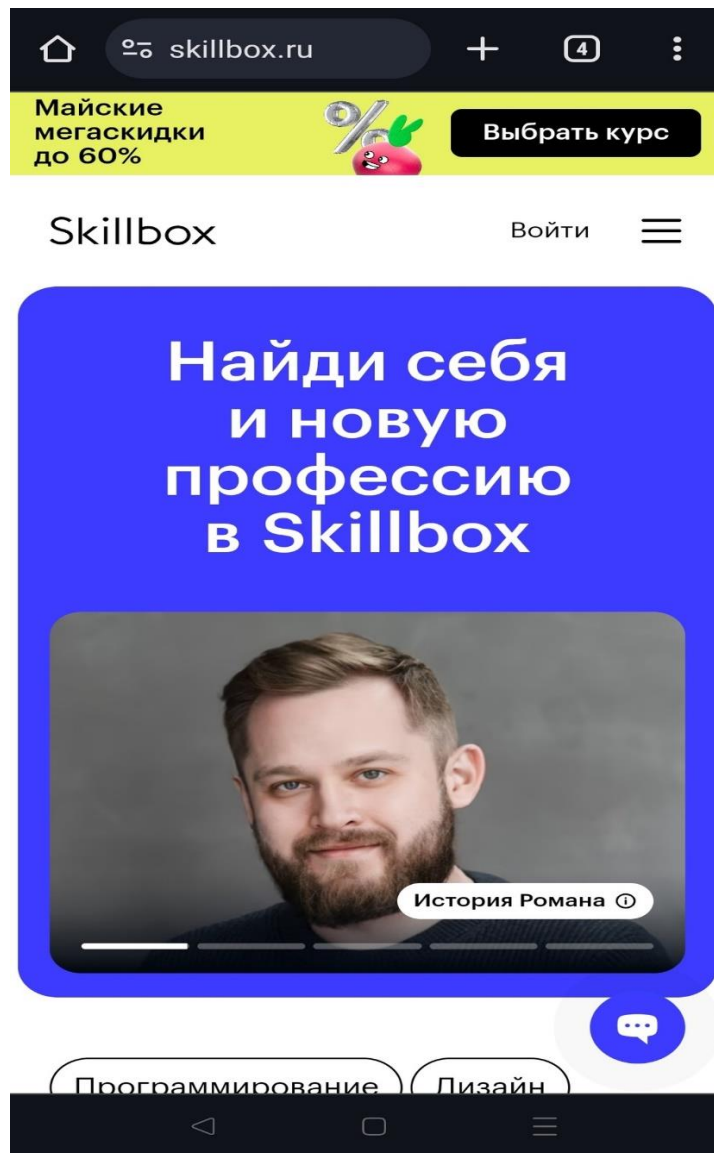


Рисунок 1 - Главная страница “Skillbox” в мобильном браузере

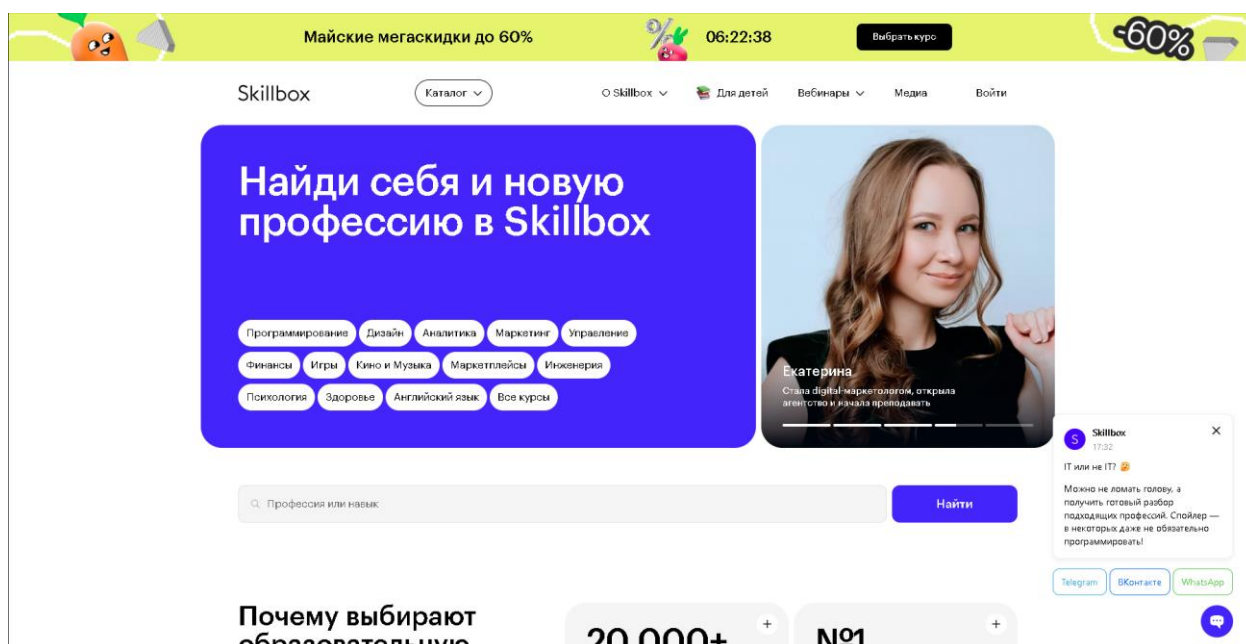


Рисунок 2 - Главная страница “Skillbox” на стационарном компьютере

3 Диаграммы

3.1 Диаграмма прецедентов для клиента (Use-case diagramm)

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.

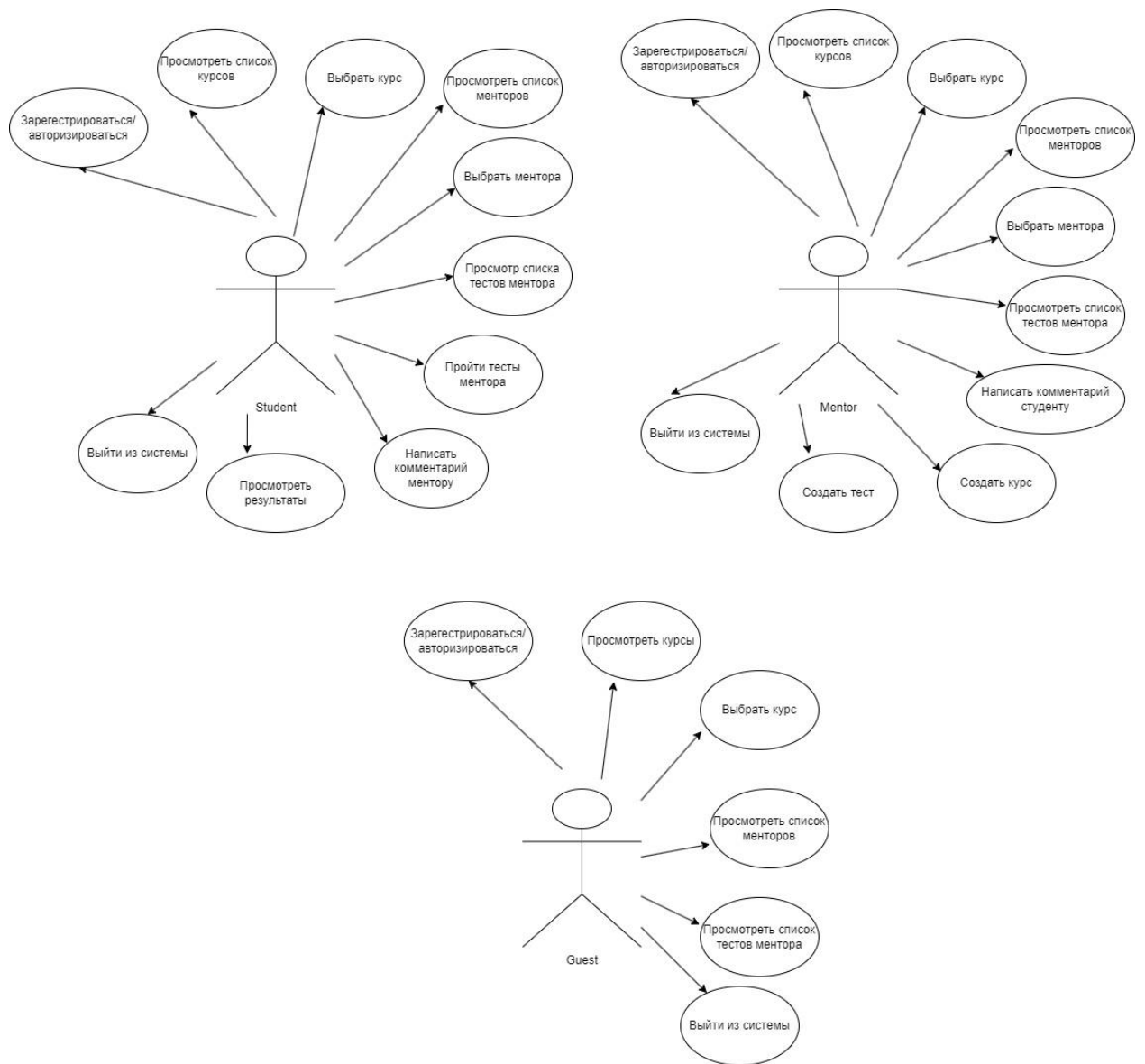


Рисунок 3 - Use-case diagramm пользователей

3.2 Диаграмма классов (Class diagramm)

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

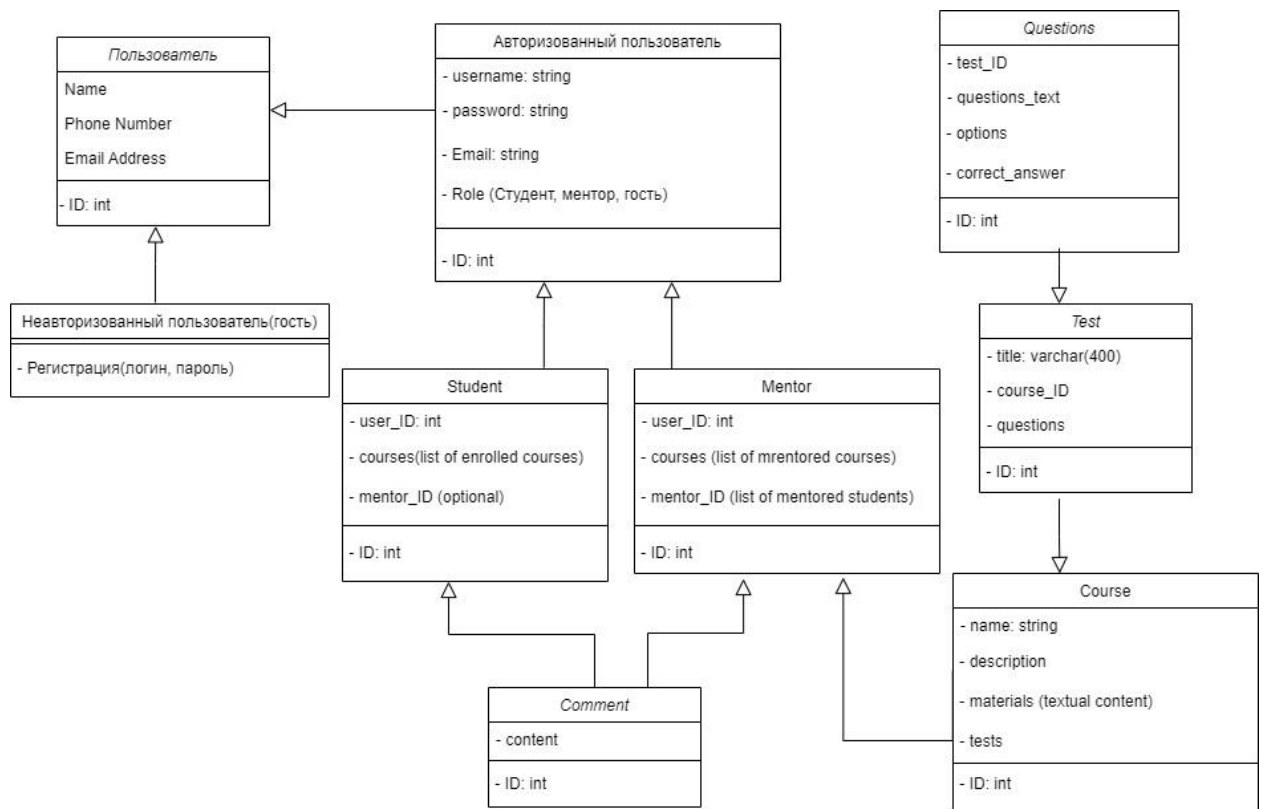


Рисунок 4 - Class diagramm

3.3 Диаграмма деятельности (Activity diagramm)

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.

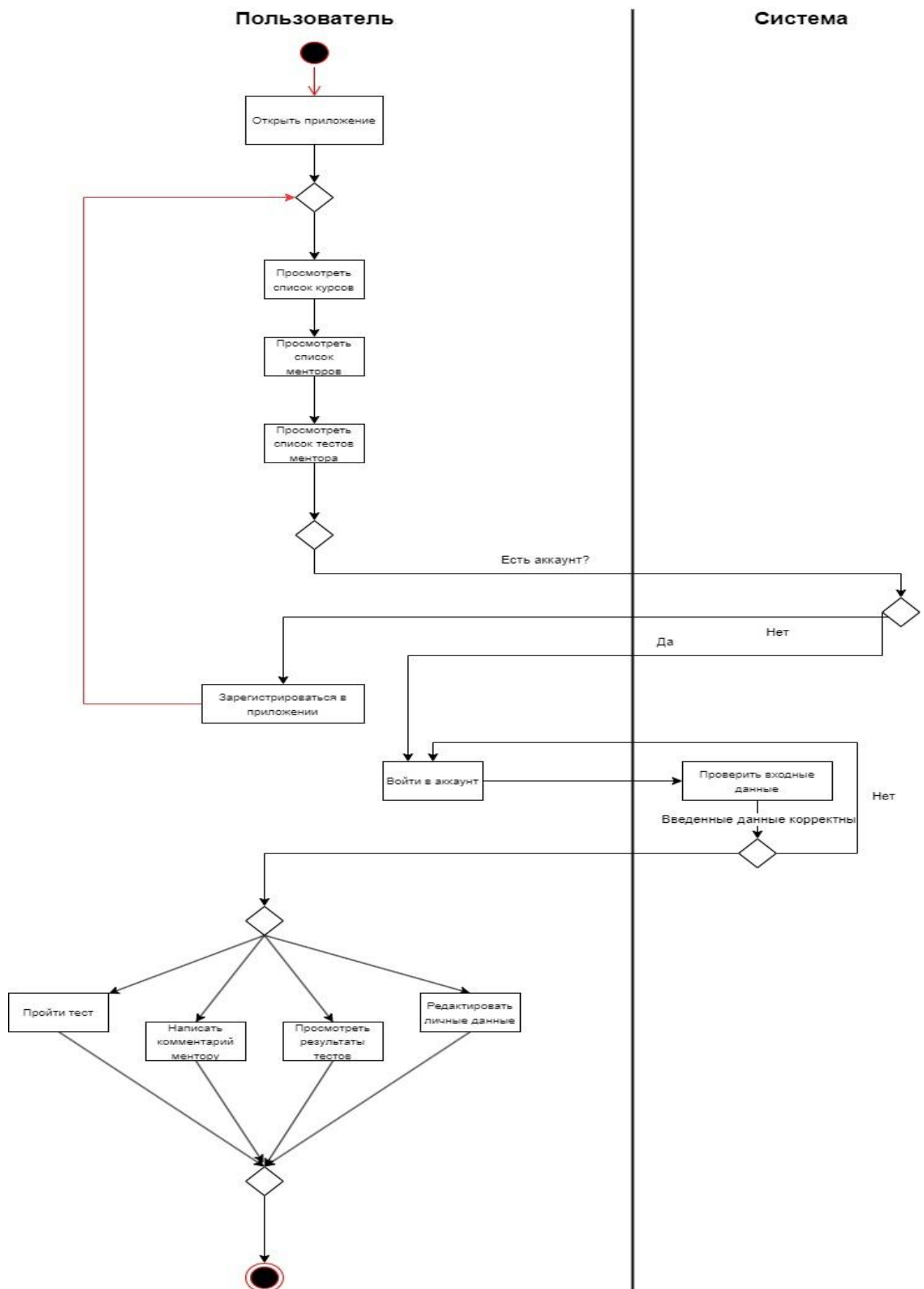


Рисунок 5 - Activity diagramm

3.4 Диаграмма последовательности (Sequence diagramm)

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.

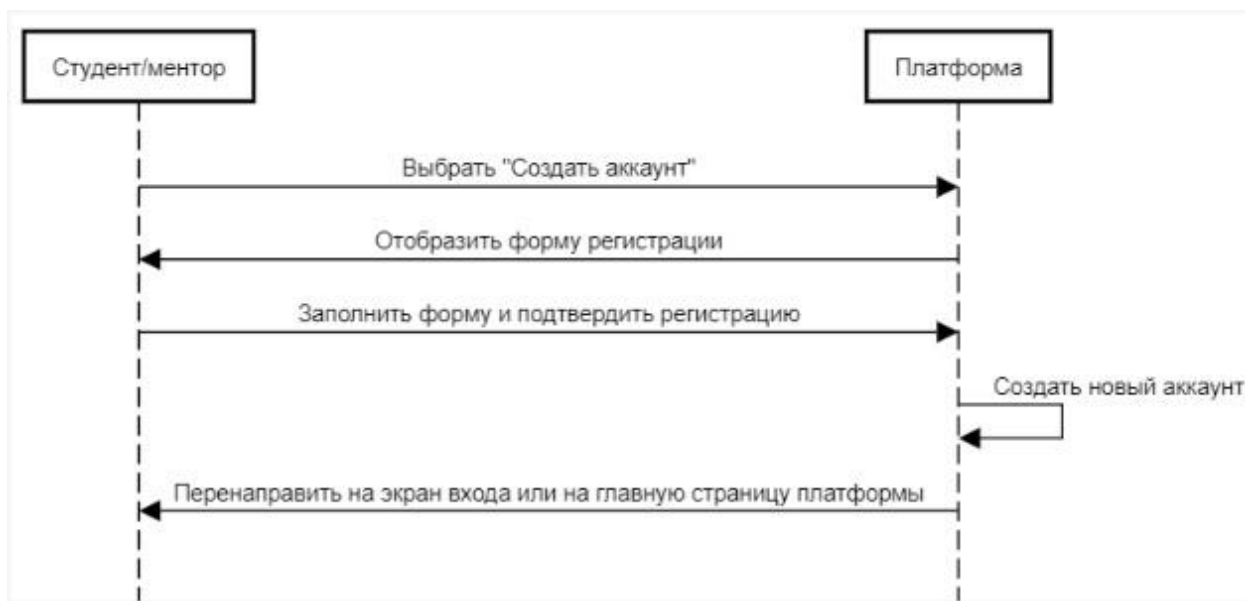


Рисунок 6 - Sequence diagram 1

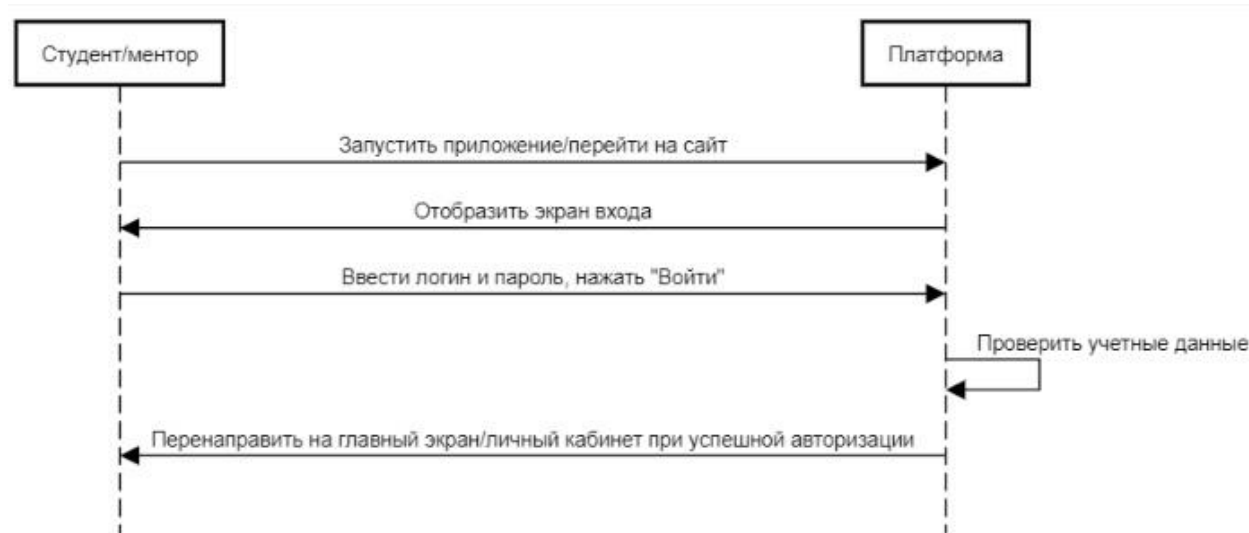


Рисунок 7 - Sequence diagramm 2

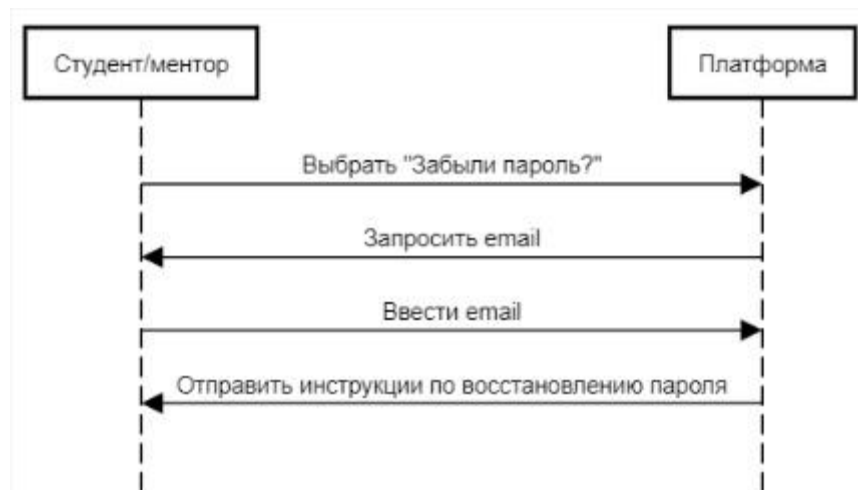


Рисунок 8 - Sequence diagramm 3

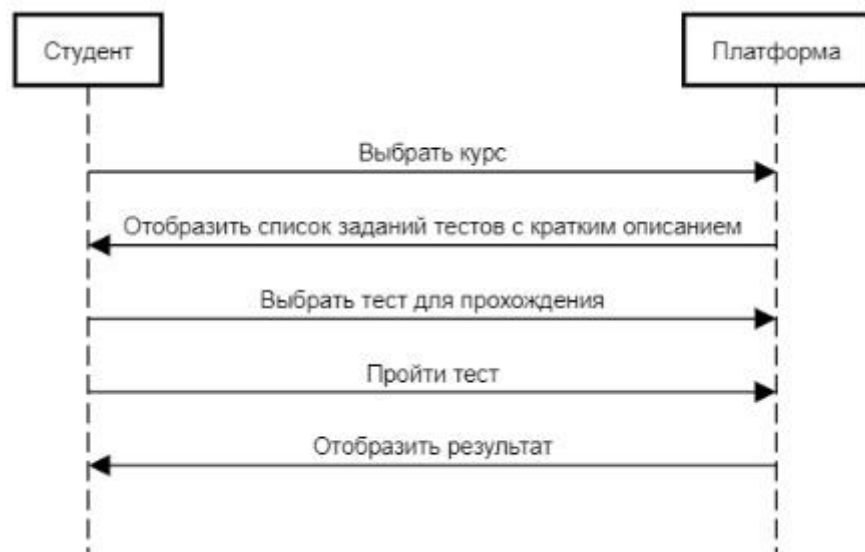


Рисунок 9 - Sequence diagram 4



Рисунок 10 - Sequence diagram 5

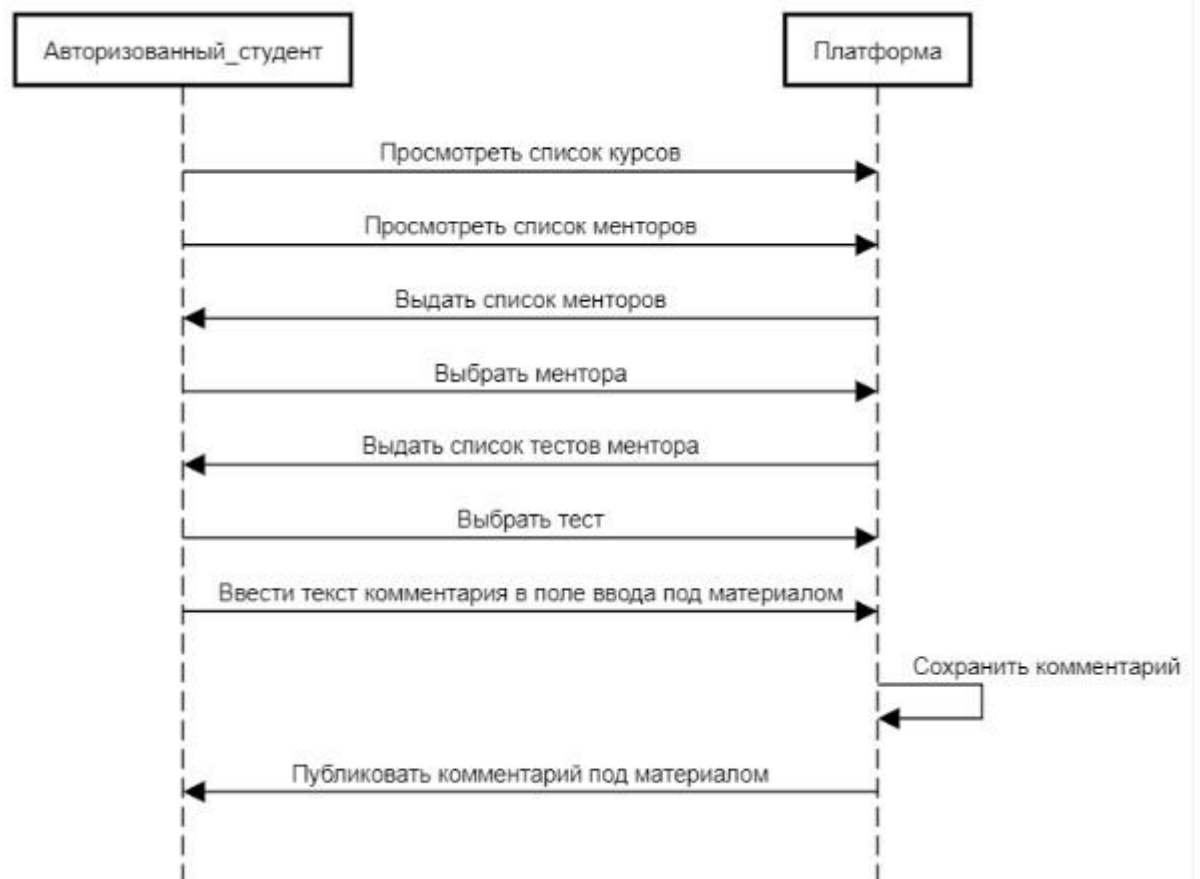


Рисунок 11 - Sequence diagram 6

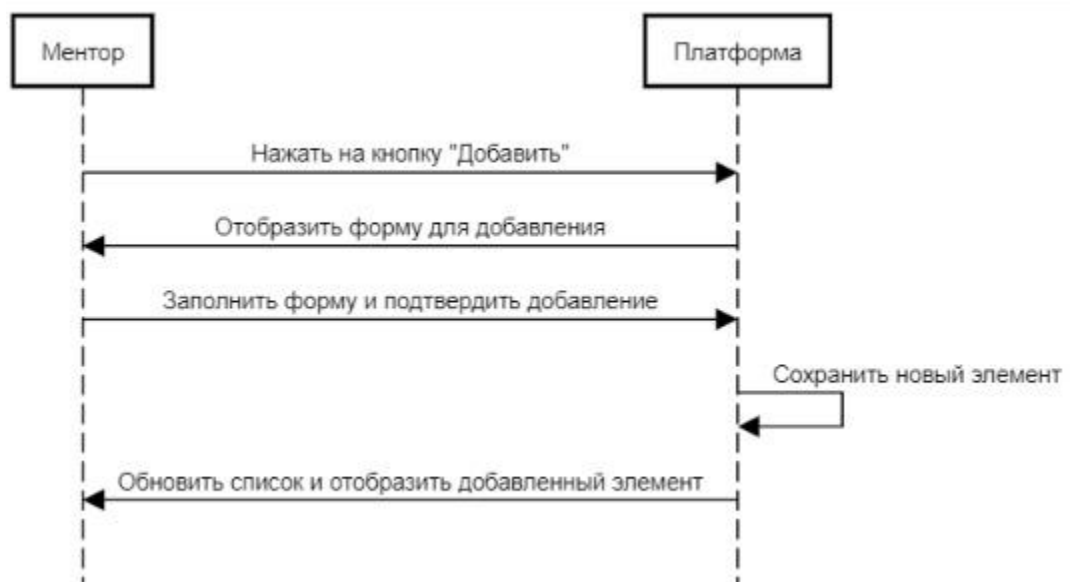


Рисунок 12 - Sequence diagram 7

3.5 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.

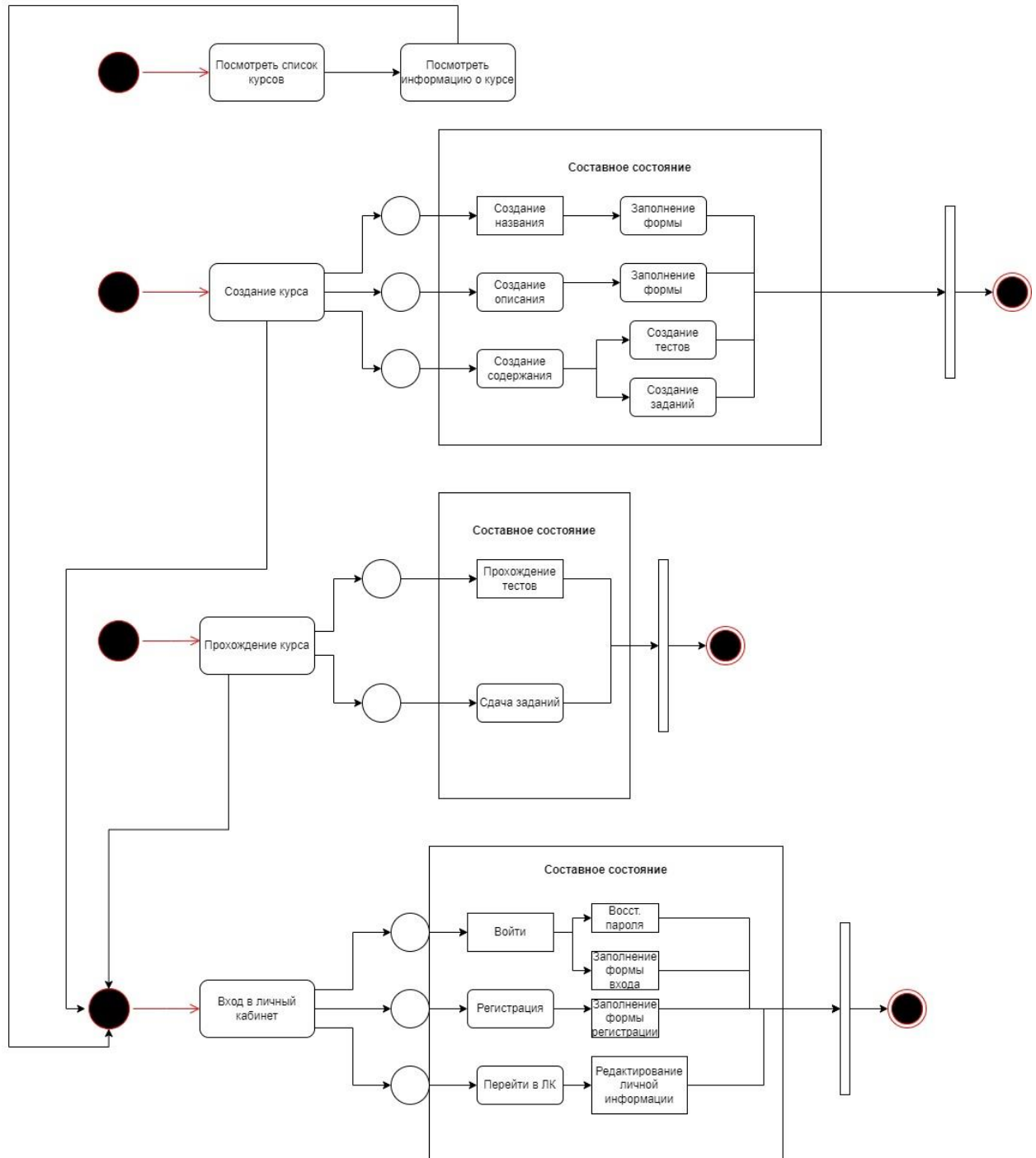


Рисунок 13 - Statechart diagramm

3.6 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.

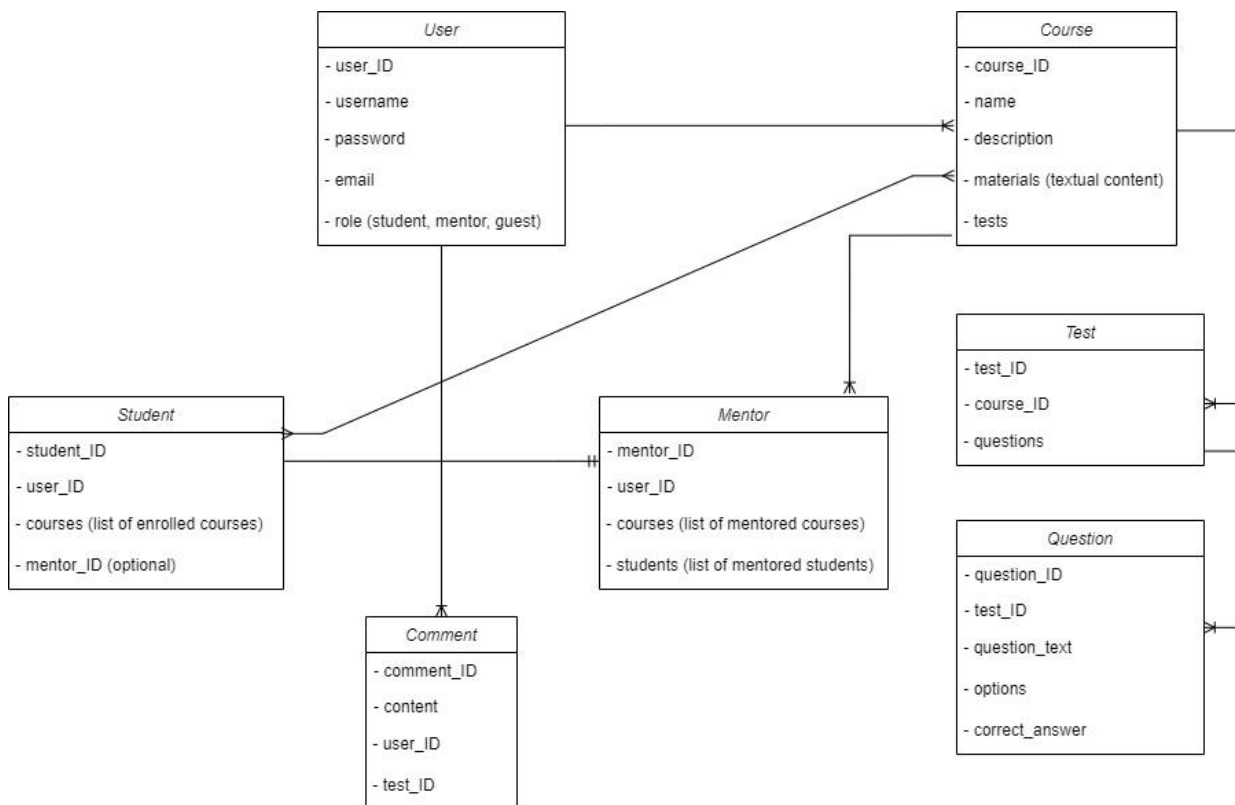


Рисунок 14 - ER-диаграмма

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 [Электронный ресурс]. — <https://skillbox.ru/> — (Дата обращения: 24.04.2024).