МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информационных систем и технологий»

Кафедра «Вычислительная техника»

Дисциплина «Информационные системы в экономике»

**КУРСОВОЙ ПРОЕКТ (РАБОТА)**

Тема «Автоматизация процессов управления информационной безопасностью»

Выполнил студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / Сулейманов М.З. /

подпись инициалы, фамилия

Курс 4 Группа ИВТАСбд-42

Направление/ специальность «Информатика и вычислительная техника»

Руководитель \_\_\_старший преподаватель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

должность, ученая степень, ученое звание

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Беляева Ирина Владимировна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

Дата сдачи:

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

Дата защиты:

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г.

Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ульяновск

2025г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет «Информационных систем и технологий»

Кафедра «Вычислительная техника»

Дисциплина «Информационные системы в экономике»

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

студенту ИВТАСбд-42 Сулейманов М.З. .

группа фамилия, инициалы

Тема проекта (работы) «Разработка обучающего мобильного приложения для предотвращения от угроз информационной безопасности»

Срок сдачи законченного проекта (работы) «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г.

Исходные данные к проекту (работе)

задание кафедры для рассмотрения аспектов научно-исследовательской работы

Содержание пояснительной записки (перечень подлежащих разработке вопросов):

1. Введение
2. Техническое задание 3. Анализ предметной области курсовой работы 4. Установление требований проекта 5. Проектирование с использованием UML 6. Проектирование базы данных 7. Методы разработки удобного программного обеспечения: 8. Заключение (подведение итогов) 9. Список литературы

Руководитель старший преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_ / И.В. Беляева /

должность подпись инициалы, фамилия

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2025г

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **/** Сулейманов М.З. **/**

подпись инициалы, фамилия

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2025г

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

**ОТЗЫВ  
руководителя на курсовой проект (работу)**

студента Сулейманова Мухтара Зейнал Оглы

фамилия, имя и отчество

Факультет ИСТ группа ИВТАСбд-42 курс 4

Дисциплина «Информационные системы в экономике»

Тема проекта (работы) «Разработка обучающего мобильного приложения для предотвращения от угроз информационной безопасности»

Отмечаются следующие моменты: актуальность темы исследования; соответствие содержания и структуры курсовой работы ее теме; степень разработанности проблемы, наиболее интересно исследованные вопросы. Оценивается степень самостоятельности и инициативы студента; умение пользоваться различными источниками информации; уровень его теоретической подготовки; умение анализировать научные материалы, делать практические выводы; знание основных концепций, научной и специальной литературы по избранной теме. Содержится оценка проекта (работы) руководителем.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель \_\_\_ старший преподаватель \_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ / И.В. Беляева /

должность, учёная степень, ученое звание подпись инициалы, фамилия

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2025г

**Оглавление**

[**Введение.** 6](#_Toc199959805)

[**1.** **Техническое задание.** 7](#_Toc199959806)

[**2.** **Анализ предметной области** 8](#_Toc199959807)

[**3.** **Установление требований проекта.** 9](#_Toc199959808)

[**3.1. Предварительные замечания к проекту.** 10](#_Toc199959809)

[3.1.1. Цели и рамки проекта. 10](#_Toc199959810)

[3.1.2. Деловой контекст. 10](#_Toc199959811)

[3.1.3. Участники проекта. 10](#_Toc199959812)

[3.1.5. Обзор документа. 10](#_Toc199959813)

[**3.2. Системные сервисы.** 10](#_Toc199959814)

[3.2.1. Рамки системы. 11](#_Toc199959815)

[3.2.2. Функциональные требования. 11](#_Toc199959816)

[**3.3. Системные ограничения.** 12](#_Toc199959817)

[3.3.1. Требования к интерфейсу. 12](#_Toc199959818)

[3.3.2. Требования к производительности 12](#_Toc199959819)

[3.3.3. Требования к безопасности 12](#_Toc199959820)

[3.3.4. Эксплуатационные требования 12](#_Toc199959821)

[3.3.5. Политические и юридические требования 12](#_Toc199959822)

[3.3.6. Другие ограничения 12](#_Toc199959823)

[**3.4. Проектные вопросы.** 12](#_Toc199959824)

[3.4.1. Открытые вопросы. 12](#_Toc199959825)

[3.4.2. Предварительный план-график. 12](#_Toc199959826)

[3.4.3. Предварительные бюджет. 12](#_Toc199959827)

[**3.5. Приложения.** 12](#_Toc199959828)

[**4.** **Проектирование с использованием UML.** 13](#_Toc199959829)

[4.1. Понятие UML-диаграм. 13](#_Toc199959830)

[4.2. UML-диаграма вариантов использования системы. 13](#_Toc199959831)

[4.3. UML-диаграмма деятельности. 15](#_Toc199959832)

[4.4. UML-диаграмма взаимодействия объектов. 16](#_Toc199959833)

[4.5. UML-диаграмма развёртывания. 17](#_Toc199959834)

[**5. Проектирование базы данных.** 19](#_Toc199959835)

[5.1 Понятие базы данных. 19](#_Toc199959836)

[5.2. Логическая модель базы данных. 19](#_Toc199959837)

[5.3. Физическая модель базы данных. 22](#_Toc199959838)

[**6. Методы разработки удобного программного обеспечения.** 24](#_Toc199959839)

[6.1. Модель ролей. 24](#_Toc199959840)

[6.2. Модель задач. 26](#_Toc199959841)

[**7. Создание проекта.** 28](#_Toc199959842)

[7.1 Структура проекта 28](#_Toc199959843)

[7.2 Backend 28](#_Toc199959844)

[7.3 Frontend 29](#_Toc199959845)

[7.4 Обзор системы 30](#_Toc199959846)

[**Заключение** 32](#_Toc199959847)

[**Список литературы** 33](#_Toc199959848)

[**Приложение** 34](#_Toc199959849)

# **Введение.**

Курсовая работа выполнена в рамках дисциплины “Информационные системы в экономике”. Данная дисциплина помогает научиться устанавливать требования будущего проекта, определять, какие роли будут иметь различные пользователи приложения, какие функциональные способности оно будет иметь, а также само создание проекта.

Информационные системы предназначены для своевременного обеспечения людей надлежащей информацией, то есть для удовлетворения конкретных информационных потребностей в рамках определённой [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C), при этом результатом функционирования информационных систем является информационная продукция — документы, информационные массивы, [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и информационные услуги

Роль информации и информационных процессов в организационно-экономической сфере настолько велика, что внедрение информационных систем на предприятии приводит к изменению и оптимизации системы управления. Формы и методы управления постоянно совершенствуются.

Данная курсовая работа затрагивает отдельные части этой дисциплины, связанные с установкой требований, проектировании и создании самого проекта (продукта).

* + - 2. **Техническое задание.**

Требуется предложить для разработки ИС и составить документ описания требований к разрабатываемой ИС. Построить UML-диаграммы к проектируемой ИС.

С помощью программы ErWin провести проектирование базы данных, которая будет использоваться в ИС. Данное проектирование включает в себя создание логической и физической модели БД.

Одним из наиболее технологичных подходов к разработке удобного пользовательского интерфейса является проектирование, ориентированное на использование.

Дизайн, ориентированный на использование, — это подход к [дизайну пользовательского интерфейса](https://en.wikipedia.org/wiki/User_interface_design), основанный на намерениях пользователя и шаблонах использования. Он анализирует пользователей с точки зрения ролей, которые они играют по отношению к системам, и использует абстрактные (основные) [варианты использования](https://en.wikipedia.org/wiki/Use_Case) для [анализа задач](https://en.wikipedia.org/wiki/Task_analysis). Он создает визуальный и [интерактивный дизайн](https://en.wikipedia.org/wiki/Interaction_design) из абстрактных [прототипов](https://en.wikipedia.org/wiki/Software_prototyping), основанных на понимании ролей пользователей и [вариантов задач](https://en.wikipedia.org/w/index.php?title=Task_case&action=edit&redlink=1).

Требуется разработать проект (продукт), ориентируясь на установленные требования и этапы проектирования.

* + - 1. **Анализ предметной области**

Информационная система экономического объекта является основой системы управления, она постоянно видоизменяется, появляются новые информационные потоки, обусловленные широким внедрением средств вычислительной техники и расширением производственных и финансовых связей предприятия. Функциональное назначение и тип информационной системы зависят от того, чьи интересы и на каком уровне она обслуживает.

Экономическая информационная система — это среда, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, персонал, технические и программные средства связи. Это система, направленная на достижение множества целей, одной из которых является производство информации, необходимой для поддержки принятия управленческих решений.

Все информационные процессы, протекающие в подразделениях аппарата управления, можно разделить на два вида: формализуемые процессы, для которых существуют алгоритмы обработки информации, и не формализуемые процессы. Именно для автоматизации формализуемых процессов используются специально разрабатываемые для предприятий и организаций экономические информационные системы. Итак, экономическая информационная система представляет собой совокупность программных, технических и информационных средств, предназначенных для автоматизации операций сбора, обработки, хранения и выдачи результатной информации пользователям для принятия управленческих решений.

* + - 1. **Установление требований проекта.**

Определение требований к системе автоматизированного контроля информационной безопасности представляет собой задание, отражающее в обобщенной форме потребности службы безопасности организации. В своей основе оно формулирует концепцию системы, описывая условия ее применения в корпоративной ИТ-инфраструктуре. Ошибочное толкование требований к системе мониторинга угроз неизбежно приводит к дефектам в ее архитектуре и функциональности.

Разработка программного комплекса для защиты информации начинается с формирования документа, который должен всесторонне отражать потребности специалистов по информационной безопасности и давать разработчикам четкое понимание этих потребностей. Состав требований к системе включает сочетание текстовых описаний на естественном языке, табличных данных и схем взаимодействия. Такой формат должен быть доступен для восприятия как техническим специалистам, так и руководителям, не обладающим глубокими знаниями в области программирования. Как правило, в исходных требованиях отсутствуют строго формализованные спецификации, за исключением случаев работы с подготовленными заказчиками - их последующая формализация становится задачей для команды разработчиков.

Расхождения в понимании требований между заказчиком, конечными пользователями и разработчиками чаще всего возникают из-за различий в восприятии роли системы безопасности в общей структуре корпоративных процессов. Поэтому ключевой задачей при определении требований становится четкое описание контекста использования системы, включая ее взаимодействие с защищаемой инфраструктурой, техническими средствами и персоналом организации. В связи с этим необходимо создание документа, детально описывающего требования к проектируемой системе автоматизированного мониторинга, анализа и противодействия угрозам информационной безопасности.

**3.1. Предварительные замечания к проекту.**

3.1.1. Цели и рамки проекта.

Основной целью данного проекта является создание комплексной системы мониторинга и обеспечения информационной безопасности, предназначенной для автоматизированного выявления и предотвращения киберугроз. Разрабатываемое решение должно отличаться высокой эффективностью обнаружения аномалий, удобством эксплуатации и наглядностью представления данных.

3.1.2. Деловой контекст.

В современных условиях цифровой трансформации организации сталкиваются с необходимостью постоянного контроля защищенности корпоративной инфраструктуры. Разрабатываемая система призвана решить актуальную проблему оперативного выявления угроз безопасности и автоматизированного реагирования на инциденты, что позволит минимизировать риски компрометации данных и простоев бизнес-процессов.

3.1.3. Участники проекта.

Разработчик – Сулейманов Мухтар Зейнал Оглы (Suleymanov.az101@gmail.com)

Заказчик - Сулейманов  Мухтар  Зейнал  Оглы (Suleymanov.az101@gmail.com)

3.1.4. Идеи в отношении решений.

Программа должна быть реализована в виде мобильного приложения, которое может открываться в телефоне на базе Android.

3.1.5. Обзор документа.

В разделе «Системные сервисы» описывается, что должна делать система. В разделе «Системные ограничения» определяется, насколько система ограничена при выполнении обслуживания. В разделе «Проектные вопросы» освещаются прочие проектные вопросы.

**3.2. Системные сервисы.**

3.2.1. Рамки системы.

Рамки системы можно моделировать с помощью диаграммы контекста.

Мобильное приложение "CyberAcademy" представляет собой обучающую платформу в области информационной безопасности. Оно обеспечивает доступ к интерактивным курсам, содержит тестовые задания, реализует систему отслеживания прогресса, а также включает чат и профиль пользователя. Курсы подгружаются из локальных JSON-файлов.

Изображение выглядит как текст, диаграмма, Шрифт, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Рис.№1.Контекстная диаграмма системы по мониторингу ИБ

3.2.2. Функциональные требования.

Веб-приложение должно обеспечивать следующие возможности:

Мобильное приложение должно обеспечивать следующие возможности:

1. Регистрация и авторизация пользователя.
2. Просмотр списка доступных курсов, состоящих из разделов и уроков.
3. Прохождение тестов по завершению разделов.
4. Отображение статистики и прогресса обучения в профиле пользователя.
5. Возможность общения через встроенный чат.

**3.3. Системные ограничения.**

3.3.1. Требования к интерфейсу.

Интерфейс мобильного приложения должен быть простым и понятным

3.3.2. Требования к производительности

Особых требований к производительности мобильного приложения нет.

3.3.3. Требования к безопасности

С программой могут работать несколько человек, входя в программу под своими учетными записями.

3.3.4. Эксплуатационные требования

Мобильное приложение открываться в телефоне на базе Android.

3.3.5. Политические и юридические требования

Отсутствие упоминаний о политике. Веб-приложение должно быть политически нейтральным.

3.3.6. Другие ограничения

Нет

**3.4. Проектные вопросы.**

3.4.1. Открытые вопросы.

Нет.

3.4.2. Предварительный план-график.

01.05.2025 - 05.05.2024 - Анализ и установление требований к моб-приложению

05.05.2025 - 07.05.2025 - Спецификация требований к моб-приложению

07.03.2024 - 20.05.2025 - Кодирование моб- приложению

20.05.2025 - 22.05.2025 - Тестовая эксплуатация моб- приложения

22.05.2025 - 25.05.2025 - Ввод в эксплуатацию

3.4.3. Предварительные бюджет.

Двести тысяч рублей

**3.5. Приложения.**

Глоссарий

Мобильное приложение — это программное обеспечение, предназначенное для установки и использования на мобильных устройствах (смартфонах, планшетах), которое обеспечивает автономную или сетевую работу.

БД – база данных.

СУБД – средства управления базами данных.

Деловые документы и формы

Нет.

Ссылки

Нет.

# **Проектирование с использованием UML.**

4.1. Понятие UML-диаграм.

UML — это сокращение от Unified Modeling Language, он является стандартизированным языком моделирования, состоящим из интегрированного набора диаграмм, разработанных, чтобы помочь разработчикам систем и программного обеспечения в определении, визуализации, конструировании и документировании артефактов программных систем, а также, к примеру, для бизнес-моделирования.

UML представляет собой набор лучших инженерных практик, которые доказали свою эффективность в моделировании больших и сложных систем и является очень важной частью разработки объектно-ориентированного программного обеспечения.

UML использует в основном графические обозначения, чтобы выразить дизайн программных проектов. Использование UML помогает проектным группам общаться, изучать потенциальные проекты и проверять архитектурный дизайн программного обеспечения.

Все построенные диаграммы были построены с помощью онлайн-сервиса Diagrams.net (https://app.diagrams.net/).

4.2. UML-диаграма вариантов использования системы.

Диаграмма вариантов использования (Use Case Diagram) определяет функциональное назначение моделируемой системы или предметной области.

Данная диаграмма отображает множество актёров, взаимодействующих с проектируемой системой (программным средством) с помощью вариантов использования. Таким образом, основными элементами диаграммы вариантов использования являются актёр и вариант использования.

Актёр – это внешняя по отношению к моделируемой системе сущность, взаимодействующая с системой для решения некоторых задач. В качестве актёра может использоваться человек, другая система, устройство или программное средство. Имя актёра основано на использовании существительного.

Вариант использования определяет некоторый набор действий (операций), которые должны быть выполнены моделируемой системой или программным средством при взаимодействии с актёром. Название варианта использования базируется на неопределённой форме глагола.

Была построена UML-диаграмма вариантов использования:

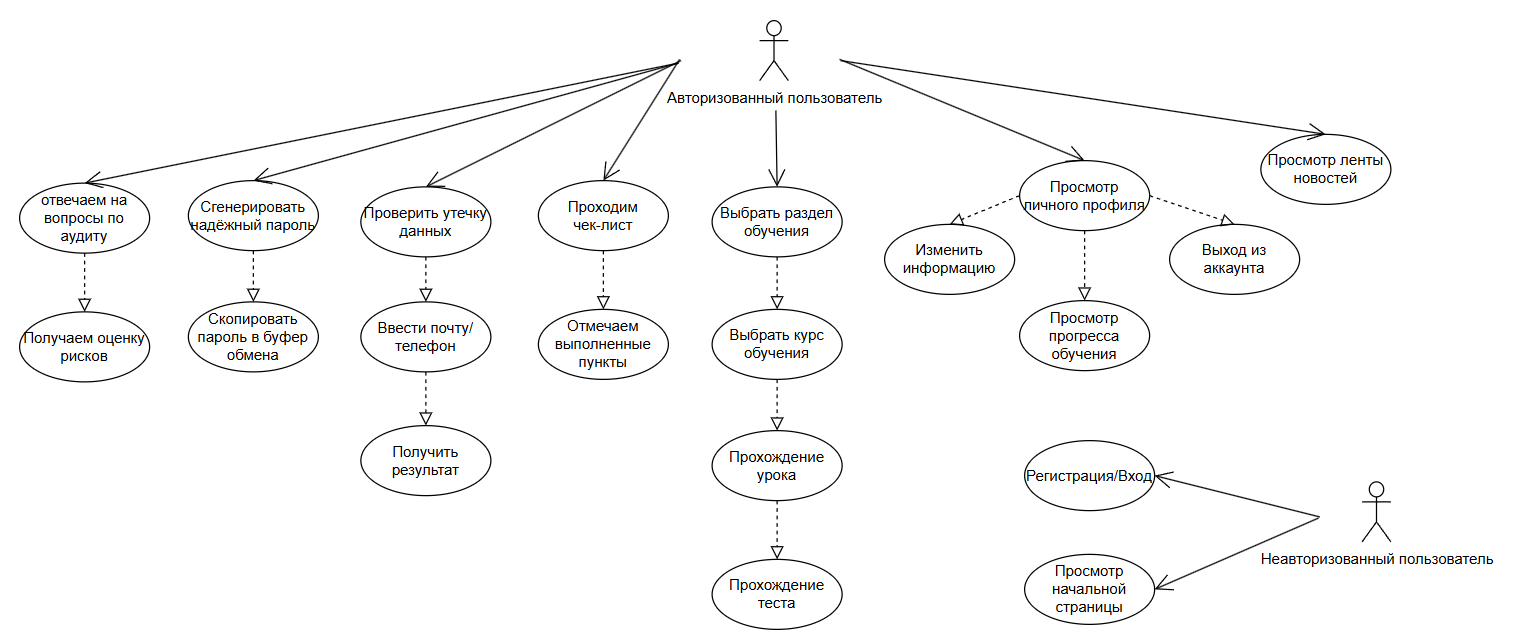


Рис №2. UML-диаграмма вариантов использования системы

Данная UML-диаграмма вариантов использования системы описывает, какие пользователи, какими возможностями обладают. Т.к. системный администратор обладает всеми возможностями авторизованного пользователя, то он соединён с ним с помощью отношения обобщения. В свою очередь авторизованный пользователь обладает всеми возможностями не авторизованного пользователя (он может только зарегистрироваться). В то же время системный администратор имеет специальные возможности, которыми обладает лишь он, такие как администрирование базой, данной в панели администратора.

4.3. UML-диаграмма деятельности.

Диаграмма деятельности — это частный случай диаграммы состояний. На диаграмме деятельности представлены переходы потока управления от одной деятельности к другой внутри системы. Этот вид диаграмм обычно используется для описания поведения, включающего в себя множество параллельных процессов.

На диаграмме деятельности могут быть представлены действия, соответствующие нескольким вариантам использования. На таких диаграммах появляется множество начальных точек, поскольку они отражают теперь реакцию системы на множество внешних событий. Таким образом, диаграммы деятельности позволяют получить полную картину поведения системы и легко оценивать влияние изменений в отдельных вариантах использования на конечное поведение системы.

Любая деятельность может быть подвергнута дальнейшей декомпозиции и представлена в виде отдельной диаграммы деятельности или спецификации (словесного описания).

Была построена UML-диаграмма деятельности:

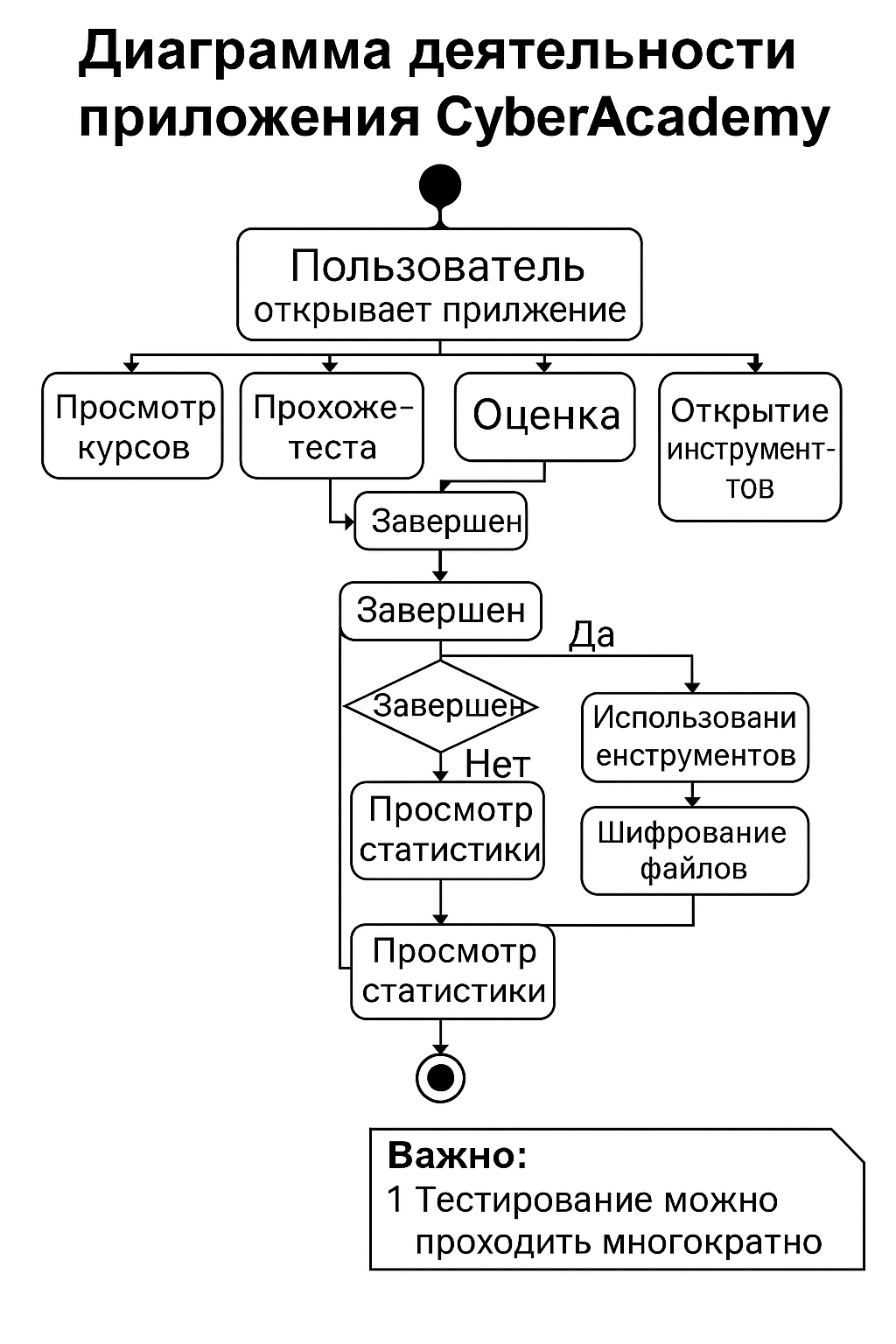


Рис №3.UML-диаграмма деятельности системы.

Данная диаграмма деятельности показывает пример деятельности системы. Сперва пользователь авторизуется в приложении. Затем, ему становятся доступны различные активности, такие как просмотр курсов, прохождения тестов, использования инструментов и отслеживание прогресса обучения.

4.4. UML-диаграмма взаимодействия объектов.

Диаграмма взаимодействия - это диаграмма, на которой представлено взаимодействие, состоящее из множества объектов и отношений между ними, включая и сообщения, которыми они обмениваются. Этот термин применяется к видам диаграмм с акцентом на взаимодействии объектов.

Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рис №4.UML-диаграмма взаимодействия объектов системы

Была построена диаграмма взаимодействия объектов на примере прохождения обучающего материала.

4.5. UML-диаграмма развёртывания.

Диаграмма развёртывания (deployment diagram) - диаграмма, на которой представлены узлы выполнения программных компонентов реального времени, а также процессов и объектов.

Диаграмма развёртывания применяется для представления общей конфигурации и топологии распределенной программной системы и содержит изображение размещения компонентов по отдельным узлам системы. Кроме того, диаграмма развёртывания показывает наличие физических соединений - маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы.

Диаграмма развёртывания предназначена для визуализации элементов и компонентов программы, существующих только на этапе её исполнения (run-time). При этом представляются только те компоненты программы, которые являются исполнимыми файлами или динамическими библиотеками. Компоненты, не используемые на этапе исполнения, на диаграмме развёртывания не показываются. Так, компоненты с исходными текстами программ могут присутствовать только на диаграмме компонентов. На диаграмме развёртывания они не указываются.

Диаграмма развёртывания содержит графические изображения процессоров, устройств, процессов и связей между ними. В отличие от диаграмм логического представления, диаграмма развёртывания является единственной для системы в целом, поскольку должна отражать все особенности её реализации. Эта диаграмма, по сути, завершает процесс ООАП для конкретной программной системы и её разработка, как правило, последний этап спецификации модели. Диаграмма развёртывания разрабатывается совместно системными аналитиками, сетевыми инженерами и системотехниками.

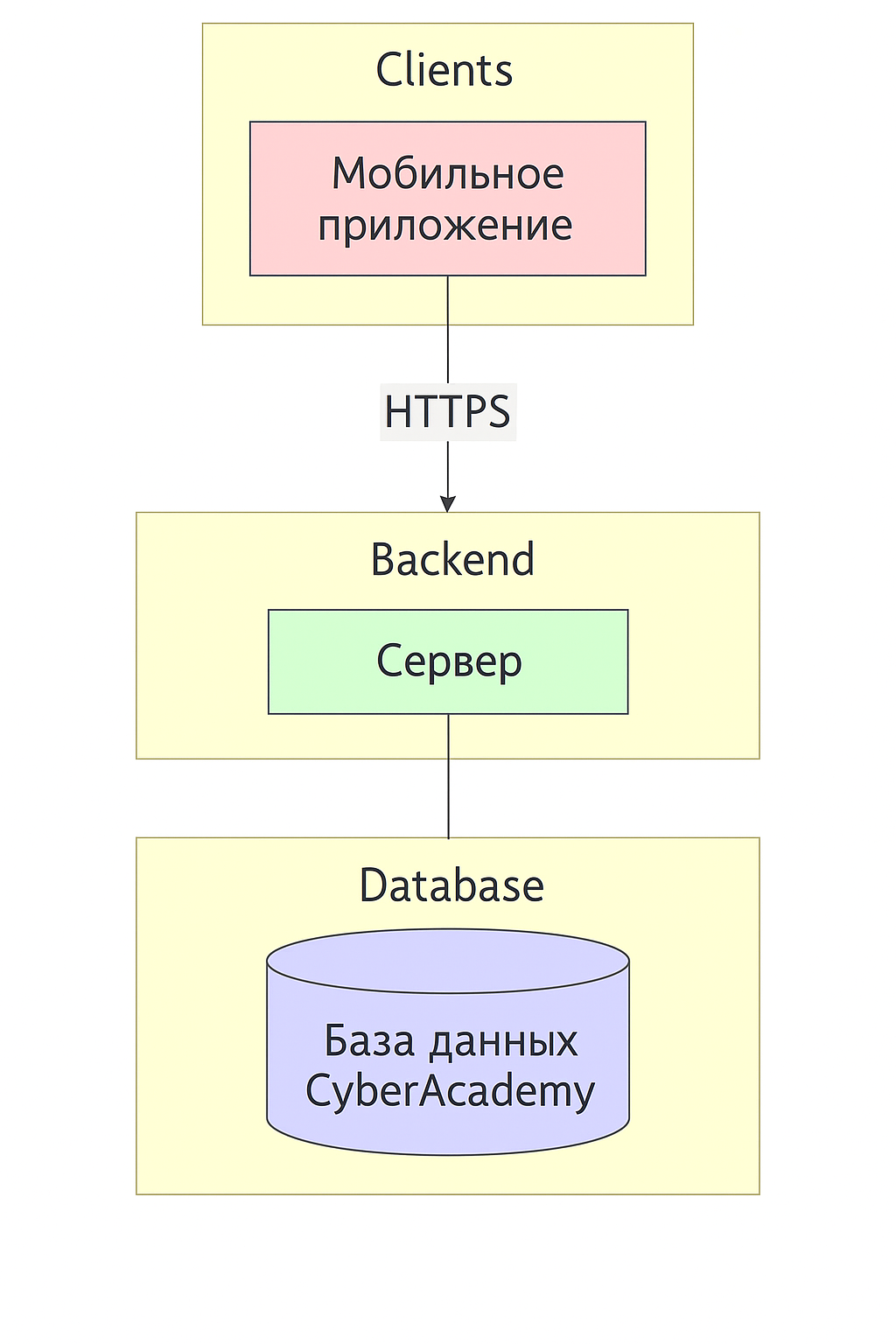


Рис №5. UML-диаграмма развёртывания системы.

На данной диаграмме развёртывания показана конфигурация обрабатывающих узлов, из которых состоит система, а так же отдельных компонентов внутри узлов, с помощью которых они и функционируют.

# **5. Проектирование базы данных.**

## 5.1 Понятие базы данных.

Цель любой информационной системы — обработка данных об объектах реального мира. Основой информационной системы является база данных. В широком смысле слова база данных — это совокупность сведений о конкретных объектах реального мира в какой-либо предметной области. Под предметной областью принято понимать часть реального мира, подлежащего изучению для организации управления его объектами и, в конечном счёте, автоматизации, например, предприятие, вуз и т. д.

Создавая базу данных, пользователь стремится упорядочить информацию по различным признакам и быстро производить выборку с произвольным сочетанием признаков. При этом очень важно выбрать правильную модель данных. Модель данных — это формализованное представление основных категорий восприятия реального мира, представленных его объектами, связями, свойствами, а также их взаимодействиями.

База данных — это информационная модель, позволяющая упорядочено хранить данные о группе объектов, обладающих одинаковым набором свойств.

Информация в базах данных хранится в упорядоченном виде. Так, в записной книжке все записи упорядочены по алфавиту, а в библиотечном каталоге либо по алфавиту (алфавитный каталог), либо в соответствии с областью знания (предметный каталог).

5.2. Логическая модель базы данных.

Логическое проектирование основано на модели логического уровня и представляет собой описание и построение схем связей между элементами данных безотносительно к их содержанию и среде хранения.

Логическая структура БД получается преобразованием концептуальной схемы в логическую схему (модель), ориентированную на выбранную СУБД.

Применительно к наиболее распространённой реляционной модели данных общий подход преобразования концептуальной схемы в логическую состоит в том, что каждую сущность, являющуюся представителем множества однотипных объектов, задают схемой отдельного отношения (таблицы), а атрибуты сущности образуют столбцы таблицы. Первичный ключ сущности образует исходный первичный ключ таблицы, который в дальнейшем может быть изменён.

Проектирование логической структуры БД должно решать задачи выбора наиболее эффективной структуры данных, обеспечения быстрого доступа к данным; исключения дублирования данных, обеспечения целостности данных таким образом, чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов.

Была построена логическая модель базы данных:

Изображение выглядит как текст, снимок экрана, диаграмма, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рис №6. Логическая модель данных.

**1. Пользователи (Users)**

Содержит информацию о зарегистрированных пользователях приложения.

* user\_id – уникальный идентификатор пользователя
* email – адрес электронной почты
* password – зашифрованный пароль
* first\_name – имя
* last\_name – фамилия
* username – имя пользователя (ник)

2. Прогресс пользователя (User\_progress)

* progress\_id – уникальный идентификатор записи
* user\_id – внешний ключ на таблицу Users
* course\_id – внешний ключ на таблицу Courses
* completion\_percent – процент завершения курса

3. Курсы (Courses)

* course\_id – уникальный идентификатор курса
* title – название курса
* description – описание

4. Уроки (Lessons)

* lesson\_id – уникальный идентификатор урока
* course\_id – внешний ключ на Courses
* title – название урока
* content\_type – тип контента (текст, видео и т.д.)

5. Тесты (Tests)

* test\_id – уникальный идентификатор теста
* lesson\_id – внешний ключ на Lessons
* score – набранный балл

6. Чек-листы (Checklists)

* checklist\_id – уникальный идентификатор
* user\_id – внешний ключ на Users
* title – заголовок чек-листа
* is\_completed – булево значение: выполнено или нет

7. Проверка утечек (Leak\_checks)

* check\_id – уникальный идентификатор
* user\_id – внешний ключ на Users
* email – проверяемый email
* leak\_source – источник утечки
* check\_date – дата проверки

8. Генератор паролей (Password\_generator)

* id – уникальный идентификатор
* length – длина пароля
* used\_special\_chars – использование спецсимволов (да/нет)

9. Лента новостей (News\_tape**)**

* news\_id – уникальный идентификатор
* title – заголовок новости
* publish\_date – дата публикации
* description – описание

Данная логическая модель данных будет использоваться при создании базы данных мобильного приложения.

Эта модель нужна для определения связей между различными атрибутами каждой таблицы. А именно: они связаны через первичные ключи (PK primary key) и внешние ключи (FK - foreign key).

5.3. Физическая модель базы данных.

Физическая модель данных зависит от конкретной СУБД, в ней содержится информация обо всех объектах базы данных. Поскольку стандартов на объекты базы данных не существует (например, нет стандарта на типы данных), физическая модель зависит от конкретной реализации СУБД и ее диалекта SQL. Следовательно, одной и той же логической модели данных могут соответствовать несколько разных физических моделей.

Основными объектами логической модели данных являются сущности, атрибуты и взаимосвязи. Физическая модель данных, как правило, создается на основе логической, поэтому каждому объекту логической модели соответствует объект физической модели. В физической модели данных сущности логической модели данных соответствует таблица, экземпляру сущности – строка в таблице, а атрибуту – колонка таблицы. Кроме перечисленных выше объектов, физическая модель может содержать объекты, тип которых зависит от СУБД: индексы, представления, последовательности, триггеры, процедуры и т.п. Если в логической модели данных не имеет большого значения, какой конкретно тип данных у атрибута, то в физической важно описать всю информацию о конкретных объектах.

Изображение выглядит как текст, диаграмма, План, Параллельный

Контент, сгенерированный ИИ, может содержать ошибки.

Рис №7. Физическая модель данных.

Физическая модель данных строится на основе логической. К логической модели данных, в каждой сущности к атрибутам добавляется присущий ему формат данных, в котором они будут храниться в столбце.

# **6. Методы разработки удобного программного обеспечения.**

Одним из наиболее технологичных подходов к разработке удобного пользовательского интерфейса является проектирование, ориентированное на использование.

Основная идея этого метода — использование специальных моделей, способствующих адекватному определению набора задач, которые необходимо решать пользователям, и способов организации информации, позволяющих упростить их решение.

Список моделей, которые используются в рамках проектирования, ориентированного на использование, следующий:  
 1) Модель ролей  
 2) Модель задач  
 3) Модель содержимого  
 4) Операционная модель  
 5) Модель реализации

6.1. Модель ролей.

Эта модель представляет собой список ролей пользователей системы. Каждая роль — это группа связанных задач и потребностей некоторого множества пользователей. Модель ролей может определять связи между ролями (роли могут уточнять друг друга, включать друг друга или просто быть похожими) и набор из одной-трёх центральных ролей, на которые, в основном, и будет нацелено проектирование.

Кроме того, каждая роль может быть снабжена профилями, указывающими различные её характеристики по отношению к контексту использования системы. Профили могут включать следующую информацию.

1)Обязанности-требования к знаниям (о предметной области, о самой системе и пр.), которым пользователь в данной роли, скорее всего, удовлетворяет.

2)Умения — уровень мастерства в работе с системой.

3)Взаимодействия — типичные варианты взаимодействия пользователя в этой роли с системой, включая их частоту, регулярность, непрерывность, концентрацию, интенсивность, сложность, предсказуемость, а также управление взаимодействием (направляется ли оно пользователем, или он только реагирует на действия системы).

4)Информация — источники, объем, направление передачи и сложность информации при взаимодействии с системой.

5)Критерии удобства-специфические критерии удобства работы для данной роли (быстрота реакции, точность указаний, удобство навигации и пр.).

6)Функции — специфические функции, возможности и свойства системы, необходимые или полезные для данной роли.

7)Возможные убытки от ошибок, которые может совершить человек в данной роли, риски использования различных функций.

Модель ролей была построена:



Рис №8. Модель ролей для системы.

Эта модель представляет собой список ролей пользователей системы. Каждая роль — это группа связанных задач и потребностей некоторого множества пользователей. Модель ролей может определять связи между ролями (роли могут уточнять друг друга, включать друг друга или просто быть похожими) и набор из одной-трёх центральных ролей, на которые, в основном, и будет нацелено проектирование.

На данной модели ролей отражены две основные роли: пользователь и администратор. Пользователь пользуется всем доступным функционалом, а администратор находится на сайте с целью администрирования БД.

6.2. Модель задач.

Модель задач при проектировании пользовательского интерфейса строится на основе сущностных вариантов использования (essential use cases). Описание сущностного варианта использования отличается от обычного тем, что в рамках его сценариев выделяются только цели и задачи пользователя, а не конкретные его действия.

Целью такого выделения является освобождение от неявных предположений о наличии определенных элементов интерфейсов, что помогает разрабатывать их именно для решаемых задач. Удобно описывать такие сценарии в виде двух последовательностей — устремлений пользователя (не действий, а задач которые он хочет решить) и обязательств системы в ответ на эти устремления.

Таблица №1. Модель задач системы CyberAcademy.

|  |  |  |
| --- | --- | --- |
| **Этап обработки инцидента ИБ** | **Действие системы** | **Результат** |
| Регистрация в системе | Проверка данных (пароль, email) | Учетная запись создана |
| Авторизация | Верификация токена/пароля | Доступ к системе предоставлен |
| Просмотр курса | Загрузка и отображение контента из JSON-файла | Урок открыт |
| Прохождение теста | Проверка ответов, расчёт результата | Пользователь получил баллы |
| Оценка прогресса | Подсчёт процента выполнения курса | Прогресс обновлён |
| Открытие чата | Подключение к модулю чата | Чат открыт для общения |
| Использование инструментов | Работа с модулями генерации паролей, проверки утечек и др. | Инструмент выполнен |
| Сохранение прогресса | Запись результата в SharedPreferences | Прогресс сохранён |

Модель задач представляет собой набор переработанных вариантов использования со связями между ними по обобщению, расширению и использованию.

В данной таблице отражается модель задач системы. Данная модель состоит из задач пользователя и обязательств сервиса в ответ на них.

# **7. Создание проекта.**

## 7.1 Структура проекта

Проект включает в себя 2 основные составляющие. Frontend, backend (Рис.10.).

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. №9. Структура проекта

## 

## 7.2 Backend

Стек технологий:

* SQLite (lessonsdb.sql)
* Файловая система и JSON
* SharedPreferences (Preferences.java)

Основные для работы файлы:

|  |  |
| --- | --- |
| Preferences.java — управление локальными настройками и флагами активности пользователя. |  |
|  |  |
| vault/TEA.java, VaultLDH.java, PasswordVaultActivity.java — логика шифрования данных и работа с хранилищем паролей.  LessonsLDH.java — загрузка данных о курсах из базы.  assets/lessonsdb.sql — структура и содержимое обучающих материалов, подгружаемых в приложение. |  |
|  |  |

## 7.3 Frontend

Стек технологий:

Material Components — визуальные элементы в стилях Material Design.

XML (Android Layouts) — визуальное описание интерфейсов.

Все файлы связанные с frontend находятся в папке layout.

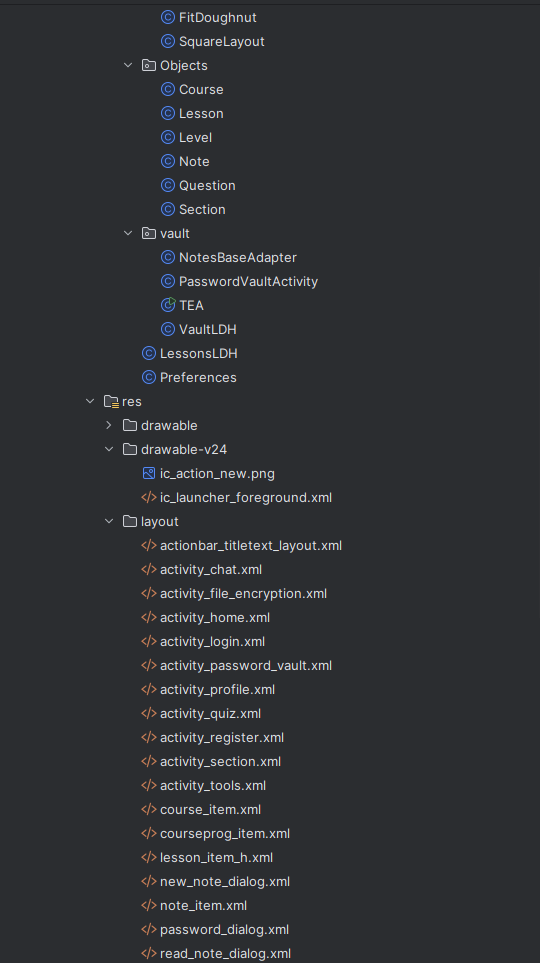


Рис. 10. Структура frontend

## 

## 7.4 Обзор системы

В данном приложении реализованы следующие ключевые особенности:

1. авторизация пользователей

Изображение выглядит как текст, снимок экрана, Устройство связи, Мобильное устройство

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 11. Авторизация пользователей

1. Личный кабинет пользователя

Изображение выглядит как электроника, текст, гаджет, Мобильное устройство

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 12. Прогресс пользователя

1. Раздел с курсами

Изображение выглядит как текст, снимок экрана, Устройство связи, Мобильное устройство

Контент, сгенерированный ИИ, может содержать ошибки.

Рис. 13. Раздел с курсами

# **Заключение**

В курсовой работе были установлены требования к создаваемому проекту, спроектированы UML-диаграммы, логическая и физическая модели базы данных, разработан удобный интерфейс, а также реализован сам проект электронной выставки изделий ручной работы.

Таким образом, во время выполнения были улучшены навыки работы с языком высокого уровня, освоено проектирование UML-диаграмм, базы данных, а также модели ролей и задач. Приобретены знания по дисциплине «Информационные системы в экономике».

# **Список литературы**

1) Информационные системы в экономике : учебник для вузов / В. Н. Волкова, В. Н. Юрьев, С. В. Широкова, А. В. Логинова ; под редакцией В. Н. Волковой, В. Н. Юрьева. — Москва : Издательство Юрайт, 2023. — 402 с.

2) Бодров, О.А Предметно-ориентированные экономические информационные системы / О.А Бодров. - М.: ГЛТ, 2013. - 244 c.

3) Вдовин, В.М. Предметно-ориентированные экономические информационные системы: Учебное пособие / В.М. Вдовин, Л.Е. Суркова, А.А. Шурупов. - М.: Дашков и К, 2016. - 388 c.

4) Васильков, А.В. Информационные системы и их безопасность: Учебное пособие / А.В. Васильков, А.А. Васильков, И.А. Васильков. - М.: Форум, 2013. - 528 c.

# **Приложение**

**FileEncryptionActivity.java**

|  |
| --- |
| package com.gcorso.cyberacademy.activities;  import android.app.Activity; import android.content.Intent; import android.net.Uri; import android.os.Bundle; import android.widget.Button; import android.widget.Toast;  import androidx.annotation.Nullable; import androidx.appcompat.app.AppCompatActivity;  import com.gcorso.cyberacademy.R;  import java.io.InputStream; import java.io.OutputStream; import java.security.SecureRandom;  import javax.crypto.Cipher; import javax.crypto.CipherInputStream; import javax.crypto.CipherOutputStream; import javax.crypto.KeyGenerator; import javax.crypto.SecretKey; import javax.crypto.spec.SecretKeySpec;  public class FileEncryptionActivity extends AppCompatActivity {  private static final int *PICK\_FILE\_TO\_ENCRYPT* = 101;  private static final int *CREATE\_ENCRYPTED\_FILE* = 102;  private static final int *PICK\_FILE\_TO\_DECRYPT* = 103;  private static final int *CREATE\_DECRYPTED\_FILE* = 104;   private Uri fileToProcessUri; // исходный файл  private boolean isEncrypting = true;  private byte[] secretKey; // не хранить так в реальном проекте!   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_file\_encryption*);   Button btnEncrypt = findViewById(R.id.*btnEncrypt*);  Button btnDecrypt = findViewById(R.id.*btnDecrypt*);   generateKey();   btnEncrypt.setOnClickListener(v -> {  isEncrypting = true;  Intent intent = new Intent(Intent.*ACTION\_OPEN\_DOCUMENT*);  intent.setType("\*/\*");  startActivityForResult(intent, *PICK\_FILE\_TO\_ENCRYPT*);  });   btnDecrypt.setOnClickListener(v -> {  isEncrypting = false;  Intent intent = new Intent(Intent.*ACTION\_OPEN\_DOCUMENT*);  intent.setType("\*/\*");  startActivityForResult(intent, *PICK\_FILE\_TO\_DECRYPT*);  });  }   private void generateKey() {  try {  KeyGenerator keyGen = KeyGenerator.*getInstance*("AES");  keyGen.init(128, new SecureRandom());  SecretKey key = keyGen.generateKey();  secretKey = key.getEncoded();  } catch (Exception e) {  e.printStackTrace();  }  }   @Override  protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {  super.onActivityResult(requestCode, resultCode, data);   if (resultCode == Activity.*RESULT\_OK* && data != null && data.getData() != null) {  Uri uri = data.getData();  if (requestCode == *PICK\_FILE\_TO\_ENCRYPT* || requestCode == *PICK\_FILE\_TO\_DECRYPT*) {  fileToProcessUri = uri;  // Спросим у пользователя, куда сохранить результат  Intent intent = new Intent(Intent.*ACTION\_CREATE\_DOCUMENT*);  intent.setType("\*/\*");  intent.putExtra(Intent.*EXTRA\_TITLE*, isEncrypting ? "encrypted\_" + System.*currentTimeMillis*() + ".enc" : "decrypted\_" + System.*currentTimeMillis*());  startActivityForResult(intent, isEncrypting ? *CREATE\_ENCRYPTED\_FILE* : *CREATE\_DECRYPTED\_FILE*);  } else if (requestCode == *CREATE\_ENCRYPTED\_FILE*) {  encryptFile(fileToProcessUri, uri);  } else if (requestCode == *CREATE\_DECRYPTED\_FILE*) {  decryptFile(fileToProcessUri, uri);  }  }  }   private void encryptFile(Uri inputUri, Uri outputUri) {  try {  InputStream in = getContentResolver().openInputStream(inputUri);  OutputStream out = getContentResolver().openOutputStream(outputUri);   SecretKeySpec keySpec = new SecretKeySpec(secretKey, "AES");  Cipher cipher = Cipher.*getInstance*("AES");  cipher.init(Cipher.*ENCRYPT\_MODE*, keySpec);   CipherOutputStream cos = new CipherOutputStream(out, cipher);   byte[] buffer = new byte[4096];  int bytesRead;  while ((bytesRead = in.read(buffer)) != -1) {  cos.write(buffer, 0, bytesRead);  }   cos.flush();  cos.close();  in.close();  out.close();   Toast.*makeText*(this, "Файл зашифрован и сохранён!", Toast.*LENGTH\_LONG*).show();  } catch (Exception e) {  e.printStackTrace();  Toast.*makeText*(this, "Ошибка шифрования: " + e.getMessage(), Toast.*LENGTH\_LONG*).show();  }  }   private void decryptFile(Uri inputUri, Uri outputUri) {  try {  InputStream in = getContentResolver().openInputStream(inputUri);  OutputStream out = getContentResolver().openOutputStream(outputUri);   SecretKeySpec keySpec = new SecretKeySpec(secretKey, "AES");  Cipher cipher = Cipher.*getInstance*("AES");  cipher.init(Cipher.*DECRYPT\_MODE*, keySpec);   CipherInputStream cis = new CipherInputStream(in, cipher);   byte[] buffer = new byte[4096];  int bytesRead;  while ((bytesRead = cis.read(buffer)) != -1) {  out.write(buffer, 0, bytesRead);  }   out.flush();  cis.close();  in.close();  out.close();   Toast.*makeText*(this, "Файл расшифрован и сохранён!", Toast.*LENGTH\_LONG*).show();  } catch (Exception e) {  e.printStackTrace();  Toast.*makeText*(this, "Ошибка расшифровки: " + e.getMessage(), Toast.*LENGTH\_LONG*).show();  }  } } |

**HomeActivity.java**

|  |
| --- |
| package com.gcorso.cyberacademy.activities;  import android.content.Intent; import android.os.Bundle; import androidx.annotation.NonNull; import com.google.android.material.bottomnavigation.BottomNavigationView; import androidx.appcompat.app.ActionBar; import androidx.appcompat.app.AppCompatActivity; import android.view.Gravity; import android.view.MenuItem; import android.view.View; import android.widget.ListView; import android.widget.TextView;  import com.gcorso.cyberacademy.LessonsLDH; import com.gcorso.cyberacademy.R; import com.gcorso.cyberacademy.adapters.CoursesListAdapter; import com.gcorso.cyberacademy.Objects.Course;  import java.util.List;  public class HomeActivity extends AppCompatActivity {   CoursesListAdapter coursesListAdapter;  LessonsLDH lessonsLDH;   private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  = new BottomNavigationView.OnNavigationItemSelectedListener() {   @Override  public boolean onNavigationItemSelected(@NonNull MenuItem item) {  Intent intent;  int id = item.getItemId();  if (id == R.id.*navigation\_explore*) {   return true;  } else if (id == R.id.*navigation\_chat*) {  intent = new Intent(HomeActivity.this, ChatActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_tools*) {  intent = new Intent(HomeActivity.this, ToolsActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_profile*) {  intent = new Intent(HomeActivity.this, ProfileActivity.class);  startActivity(intent);  return true;  }  return false;  }  };   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_home*);   lessonsLDH = LessonsLDH.*getInstance*(this);  List<Course> courses = lessonsLDH.getCourses();    coursesListAdapter = new CoursesListAdapter(this, courses);  ListView listCourses = findViewById(R.id.*courses\_list*);  listCourses.setAdapter(coursesListAdapter);   BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);   final ActionBar abar = getSupportActionBar();  View viewActionBar = getLayoutInflater().inflate(R.layout.*actionbar\_titletext\_layout*, null);  ActionBar.LayoutParams params = new ActionBar.LayoutParams(  ActionBar.LayoutParams.*WRAP\_CONTENT*,  ActionBar.LayoutParams.*MATCH\_PARENT*,  Gravity.*CENTER*);  TextView textviewTitle = (TextView) viewActionBar.findViewById(R.id.*actionbar\_textview*);  textviewTitle.setText("Explore");  abar.setCustomView(viewActionBar, params);  abar.setDisplayShowCustomEnabled(true);  abar.setDisplayShowTitleEnabled(false);    }  } |

**ProfileActivity.java**

|  |
| --- |
| package com.gcorso.cyberacademy.activities;  import android.content.Intent; import android.os.Bundle; import androidx.annotation.NonNull; import com.google.android.material.bottomnavigation.BottomNavigationView; import androidx.appcompat.app.ActionBar; import androidx.appcompat.app.AppCompatActivity; import android.view.Gravity; import android.view.MenuItem; import android.view.View; import android.widget.TextView;  import com.gcorso.cyberacademy.LessonsLDH; import com.gcorso.cyberacademy.R; import com.gcorso.cyberacademy.adapters.CoursesProgGridAdapter; import com.gcorso.cyberacademy.layout.ExpandableHeightGridView; import com.gcorso.cyberacademy.layout.FitDoughnut; import com.gcorso.cyberacademy.Objects.Course; import com.gcorso.cyberacademy.Objects.Level;  import java.util.ArrayList; import java.util.List;  public class ProfileActivity extends AppCompatActivity {   private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  = new BottomNavigationView.OnNavigationItemSelectedListener() {   @Override  public boolean onNavigationItemSelected(@NonNull MenuItem item) {  Intent intent;  int id = item.getItemId();  if (id == R.id.*navigation\_explore*) {  intent = new Intent(ProfileActivity.this, HomeActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_chat*) {  intent = new Intent(ProfileActivity.this, ChatActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_tools*) {  intent = new Intent(ProfileActivity.this, ToolsActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_profile*) {   return true;  }  return false;  }  };    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_profile*);   final ActionBar abar = getSupportActionBar();  View viewActionBar = getLayoutInflater().inflate(R.layout.*actionbar\_titletext\_layout*, null);  ActionBar.LayoutParams params = new ActionBar.LayoutParams(  ActionBar.LayoutParams.*WRAP\_CONTENT*,  ActionBar.LayoutParams.*MATCH\_PARENT*,  Gravity.*CENTER*);  TextView textviewTitle = (TextView) viewActionBar.findViewById(R.id.*actionbar\_textview*);  textviewTitle.setText("Profile");  abar.setCustomView(viewActionBar, params);  abar.setDisplayShowCustomEnabled(true);  abar.setDisplayShowTitleEnabled(false);   BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);  navigation.setSelectedItemId(R.id.*navigation\_profile*);   LessonsLDH lessonsLDH = LessonsLDH.*getInstance*(this);  Level level = lessonsLDH.getLevel();   // set up the overall level  FitDoughnut doughnut = (FitDoughnut) findViewById(R.id.*doughnuttot*);  doughnut.animateSetPercent((float) level.getPerctot());  TextView tvperctot = findViewById(R.id.*tvpercentage*);  String p = Integer.*toString*(level.getPerctot())+ "%";  tvperctot.setText(p);  TextView tvLev = findViewById(R.id.*tvlevel*);  tvLev.setText(level.getLiv());  TextView tvProg = findViewById(R.id.*tvprogress*);  String prog = Integer.*toString*(level.getProg()) + " / " + Integer.*toString*(level.getTot());  tvProg.setText(prog);   ExpandableHeightGridView gridCourses = findViewById(R.id.*gridCoursesProg*);  List<Course> courses = new ArrayList<>();  List<String> coursesTitles = lessonsLDH.getCoursesNames();  for(int i = 0; i<coursesTitles.size(); i++){  courses.add(new Course(coursesTitles.get(i), level.getPerccourses()[i]));  }  CoursesProgGridAdapter coursesAdapter = new CoursesProgGridAdapter(courses);  gridCourses.setAdapter(coursesAdapter);  gridCourses.setExpanded(true);    } } |

**ToolsActivity.java**

|  |
| --- |
| package com.gcorso.cyberacademy.activities;  import android.content.Intent; import android.os.Bundle; import androidx.annotation.NonNull; import com.google.android.material.bottomnavigation.BottomNavigationView; import androidx.appcompat.app.ActionBar; import androidx.appcompat.app.AppCompatActivity; import android.view.Gravity; import android.view.LayoutInflater; import android.view.MenuItem; import android.view.View; import android.widget.Button; import android.widget.EditText; import android.widget.RelativeLayout; import android.widget.TextView; import android.widget.Toast;  import com.gcorso.cyberacademy.R; import com.gcorso.cyberacademy.vault.PasswordVaultActivity;  public class ToolsActivity extends AppCompatActivity {    private BottomNavigationView.OnNavigationItemSelectedListener mOnNavigationItemSelectedListener  = new BottomNavigationView.OnNavigationItemSelectedListener() {   @Override  public boolean onNavigationItemSelected(@NonNull MenuItem item) {  Intent intent;  int id = item.getItemId();  if (id == R.id.*navigation\_explore*) {  intent = new Intent(ToolsActivity.this, HomeActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_chat*) {  intent = new Intent(ToolsActivity.this, ChatActivity.class);  startActivity(intent);  return true;  } else if (id == R.id.*navigation\_tools*) {  return true;  } else if (id == R.id.*navigation\_profile*) {  intent = new Intent(ToolsActivity.this, ProfileActivity.class);  startActivity(intent);  return true;  }  return false;  }  };   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_tools*);   final ActionBar abar = getSupportActionBar();  View viewActionBar = getLayoutInflater().inflate(R.layout.*actionbar\_titletext\_layout*, null);  ActionBar.LayoutParams params = new ActionBar.LayoutParams(  ActionBar.LayoutParams.*WRAP\_CONTENT*,  ActionBar.LayoutParams.*MATCH\_PARENT*,  Gravity.*CENTER*);  TextView textviewTitle = (TextView) viewActionBar.findViewById(R.id.*actionbar\_textview*);  textviewTitle.setText("Tools");  abar.setCustomView(viewActionBar, params);  abar.setDisplayShowCustomEnabled(true);  abar.setDisplayShowTitleEnabled(false);   BottomNavigationView navigation = (BottomNavigationView) findViewById(R.id.*navigation*);  navigation.setOnNavigationItemSelectedListener(mOnNavigationItemSelectedListener);  navigation.setSelectedItemId(R.id.*navigation\_tools*);   RelativeLayout itemVault = findViewById(R.id.*itemvault*);  itemVault.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  LayoutInflater inflater = ToolsActivity.this.getLayoutInflater();  View dialogView = inflater.inflate(R.layout.*password\_dialog*, null);  final android.app.AlertDialog dialog = new android.app.AlertDialog.Builder(ToolsActivity.this).setView(dialogView).show();   final EditText input = dialogView.findViewById(R.id.*fieldpassword*);   Button btOpen = dialogView.findViewById(R.id.*btOpen*);  btOpen.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {   String passwordstr = input.getText().toString();   if(passwordstr.length()>0){   while(passwordstr.length()<16){  passwordstr = passwordstr + " ";  }  Intent openReading = new Intent (ToolsActivity.this, PasswordVaultActivity.class);  openReading.putExtra("password", passwordstr);  startActivity(openReading);  dialog.dismiss();  }    }  });  Button btCancel = dialogView.findViewById(R.id.*btCancel*);  btCancel.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  dialog.cancel();  }  });  }  });   RelativeLayout itemData = findViewById(R.id.*itemdata*);  itemData.setOnClickListener(itemEmptyListener);   Intent intent = new Intent(ToolsActivity.this, FileEncryptionActivity.class);  startActivity(intent);  }   View.OnClickListener itemEmptyListener = new View.OnClickListener() {  @Override  public void onClick(View view) {  Toast.*makeText*(ToolsActivity.this, "Sorry, this tool is not available yet.", Toast.*LENGTH\_LONG*).show();  }  }; } |

**QuizActivity.java**

|  |
| --- |
| package com.gcorso.cyberacademy.activities;  import android.content.Intent; import android.os.Bundle; import androidx.appcompat.app.AppCompatActivity; import android.view.LayoutInflater; import android.view.View; import android.widget.Button; import android.widget.ImageView; import android.widget.RelativeLayout; import android.widget.TextView;  import com.gcorso.cyberacademy.LessonsLDH; import com.gcorso.cyberacademy.Preferences; import com.gcorso.cyberacademy.R; import com.gcorso.cyberacademy.layout.FitDoughnut; import com.gcorso.cyberacademy.Objects.Level;  import java.util.ArrayList; import java.util.List;  public class QuizActivity extends AppCompatActivity {  String[] questions;  String lessontitle;  LessonsLDH lessonsLDH;  int curq;  int lessonid;  public static final String *Q\_SEP* = "<<-->>";  public static final String *A\_SEP* = "<->";  boolean[] results;   TextView lessontitleTv;  TextView questiontitleTv;  Button[] answersBt;  ImageView[] resqIv;  RelativeLayout emptyview;  ImageView closeBt;    @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_quiz*);   Intent intent = getIntent();  lessonid = intent.getIntExtra("lessonid", 0);   lessonsLDH = LessonsLDH.*getInstance*(this);  String q = lessonsLDH.getQuestion(lessonid);  lessontitle = lessonsLDH.getLessonTitle(lessonid);   questions = q.split(*Q\_SEP*);  curq = 0;   lessontitleTv = findViewById(R.id.*lesson\_title*);  questiontitleTv = findViewById(R.id.*question\_title*);   answersBt = new Button[]{findViewById(R.id.*answer1*), findViewById(R.id.*answer2*), findViewById(R.id.*answer3*), findViewById(R.id.*answer4*)};  resqIv = new ImageView[]{findViewById(R.id.*resq0*), findViewById(R.id.*resq1*), findViewById(R.id.*resq2*),  findViewById(R.id.*resq3*), findViewById(R.id.*resq4*),findViewById(R.id.*resq5*),findViewById(R.id.*resq6*),  findViewById(R.id.*resq7*),findViewById(R.id.*resq8*),findViewById(R.id.*resq9*)};  emptyview = findViewById(R.id.*emptyview*);  emptyview.setVisibility(View.*GONE*);   lessontitleTv.setText(lessontitle);   results = new boolean[]{false, false, false, false, false, false, false, false, false, false};   emptyview.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  nextQuestion();  }  });   setCurrentQuestion();   closeBt = findViewById(R.id.*closebt*);  closeBt.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  Intent intent = new Intent(QuizActivity.this, HomeActivity.class);  startActivity(intent);  }  });  }   private void nextQuestion(){  if(curq == 9){  int tot = 0;  for(int i = 0; i<10; i++){  if(results[i]) {  tot++;  }  }   int ch = 5\*lessonsLDH.updateResult(lessonid, tot);   LayoutInflater inflater = this.getLayoutInflater();  View dialogView = inflater.inflate(R.layout.*result\_dialog*, null);   final android.app.AlertDialog dialog = new android.app.AlertDialog.Builder(QuizActivity.this).setView(dialogView).show();   TextView tvCompl = dialogView.findViewById(R.id.*tvcompl*);  TextView tvScore = dialogView.findViewById(R.id.*tvscore*);  TextView tvResult = dialogView.findViewById(R.id.*tvresult*);   if(tot == 10){  tvCompl.setText("Perfect!");  } else if (tot>=7){  tvCompl.setText("Well Done!");  } else {  tvCompl.setText("Try Again!");  tvCompl.setTextColor(getResources().getColor(R.color.*wrong*));  TextView tvExtra = dialogView.findViewById(R.id.*tvextra*);  tvExtra.setVisibility(View.*VISIBLE*);  }   String res = "Score:\n" + Integer.*toString*(tot) + "/10";  tvResult.setText(res);   String change = "+" + Integer.*toString*(ch) + "\n" + Preferences.*SCORE\_NAME*;  tvScore.setText(change);   // level  Level level = lessonsLDH.getLevel();  FitDoughnut doughnut = (FitDoughnut) dialogView.findViewById(R.id.*doughnuttot*);  doughnut.animateSetPercent((float) level.getPerctot());  TextView tvperctot = dialogView.findViewById(R.id.*tvpercentage*);  String p = Integer.*toString*(level.getPerctot())+ "%";  tvperctot.setText(p);  TextView tvLev = dialogView.findViewById(R.id.*tvlevel*);  tvLev.setText(level.getLiv());  TextView tvProg = dialogView.findViewById(R.id.*tvprogress*);  String prog = Integer.*toString*(level.getProg()) + " / " + Integer.*toString*(level.getTot());  tvProg.setText(prog);   Button btTryAgain = dialogView.findViewById(R.id.*btTryAgain*);  btTryAgain.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  dialog.dismiss();  Intent intent = new Intent(QuizActivity.this, SectionActivity.class);  intent.putExtra("sectionn", 0);  intent.putExtra("lessonid", lessonid);  startActivity(intent);  }  });   Button btHome = dialogView.findViewById(R.id.*btHome*);  btHome.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  dialog.dismiss();  Intent intent = new Intent(QuizActivity.this, HomeActivity.class);  startActivity(intent);  }  });   } else {  emptyview.setVisibility(View.*GONE*);  curq++;  setCurrentQuestion();  }  }   private void setCurrentQuestion(){  String[] parts = questions[curq].split(*A\_SEP*);  questiontitleTv.setText(parts[0]);  final int correctans;   if(parts.length == 2){  // true or false  answersBt[0].setBackground(getResources().getDrawable(R.drawable.*answerbox*));  answersBt[1].setBackground(getResources().getDrawable(R.drawable.*answerbox*));  answersBt[0].setText("True");  answersBt[1].setText("False");  answersBt[2].setVisibility(View.*GONE*);  answersBt[3].setVisibility(View.*GONE*);   if(parts[1].charAt(0)== 'F'){  correctans = 1;  } else {  correctans = 0;  }  } else {  // 4 multiple choices   List<Integer> list = new ArrayList<Integer>();  list.add(0);  list.add(1);  list.add(2);  list.add(3);  java.util.Collections.*shuffle*(list);   correctans = list.indexOf(0);   for(int i = 0; i<4; i++){  answersBt[i].setVisibility(View.*VISIBLE*);  answersBt[i].setBackground(getResources().getDrawable(R.drawable.*answerbox*));  answersBt[i].setText(parts[list.get(i)+1]);  }  }   answersBt[correctans].setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  answersBt[correctans].setBackground(getResources().getDrawable(R.drawable.*answerboxcorrect*));  results[curq] = true;  resqIv[curq].setBackground(getResources().getDrawable(R.drawable.*qcorrect*));  emptyview.setVisibility(View.*VISIBLE*);  }  });   for(int i = 0; i<4; i++){  if(i != correctans){  final int finalI = i;  answersBt[i].setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View view) {  answersBt[correctans].setBackground(getResources().getDrawable(R.drawable.*answerboxcorrect*));  answersBt[finalI].setBackground(getResources().getDrawable(R.drawable.*answerboxwrong*));  resqIv[curq].setBackground(getResources().getDrawable(R.drawable.*qwrong*));  emptyview.setVisibility(View.*VISIBLE*);  }  });  }  }  } } |