

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

Кафедра

вычислительной техники

«Вычислительная техника»

Направление подготовки: 09.03.01 «Информатика и вычислительная техника»

Профиль подготовки: «Программное обеспечение средств вычислительной
техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ
ПОДГОТОВКИ К ЕГЭ**

Студент

ГГ 09.06.2025
(подпись, дата)

Головинов Владислав Русланович
(ФИО полностью)

Руководитель

В 09.06.2025
(подпись, дата)

Егоров В.Ю.
(фамилия, инициалы)

Нормоконтролёр

А.И. 09.06.2025
(подпись, дата)

Бычков А.С.
(фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 г. № 16)

Заведующий кафедрой

М.А.
(подпись)

М.А. Митрохин
(фамилия, инициалы)

Работа защищена с отметкой хорошо (протокол заседания ГЭК от

16 июня 2025 г. № 35)

Секретарь ГЭК

Е
(подпись)

Е.И. Гурин
(фамилия, инициалы)

Пенза 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
« 06 » 05 20 25 г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Головинов Владислав Русланович гр. 21ВВП2 факультета ФВТ

2. Тема работы мобильное приложение по подготовки к ЕГЭ

Тема утверждена приказом ПГУ № 601/о от 06.05.2025 г.

3. Руководитель бакалаврской работы к.т.н., доцент кафедры ВТ ПГУ Егоров В.Ю.

4. Задание на работу (назначение разработки, исходные данные и т.п.).

Разработать приложение «Мобильное приложение по подготовки к ЕГЭ», предоставляющее возможность изучать экзаменационный материал и проходить тесты. Система должна включать в себя функционал сбора и сохранения данных в СУБД. Регистрация и аутентификация необходима для доступа к персональным функциям пользователя. Программа должна позволять пользователю просматривать обучающий материал, проходить тесты, анализировать свою статистику успеваемости.

5. Перечень подлежащих разработке вопросов:

Произвести анализ предметной области: выявить достоинства и недостатки существующих приложений в сфере подготовки к ЕГЭ, поставить задачу на ВКР. Собрать инструментарий для проектирования и реализации мобильного приложения. Произвести проектирование, реализацию и наполнение БД SQLite. Выбрать архитектуру приложения, разработать и протестировать приложение.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	06.05.2025 – 08.05.2025	BS
2	Анализ требований	32	09.05.2025 – 11.05.2025	BS
3	Создание макета графического интерфейса	30	12.05.2025 – 14.05.2025	BS
4	Выбор технологий, применяемых при проектировании и разработке	40	15.05.2025 – 18.05.2025	BS
5	Реализация графического интерфейса	66	19.05.2025 – 22.05.2025	BS
6	Разработка основной части программы	70	23.05.2025 – 27.05.2025	BS
7	Оформление пояснительной записки	54	28.05.2025 – 30.05.2025	BS

Дата выдачи задания "06" мая 2025 г.

Руководитель бакалаврской работы BS 06.05.2025 Егоров В.Ю.
(подпись, дата)

Задание к исполнению принял студент JS 06.05.2025 Головинов В.Р.
(подпись, дата)

Работу к защите допустить
Декан факультета JS 16.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 90 листов, 21 рисунок, 5 таблиц, 5 источников и 3 приложения.

МОБИЛЬНОЕ ПРИЛОЖЕНИЕ, ОБРАБОТКА СОБЫТИЙ, ЛОГИРОВАНИЕ, БАЗА ДАННЫХ, ПАРАЛЛЕЛЬНОЕ ПРОГРАММИРОВАНИЕ, JAVA, SQLITE

Цель работы – разработка приложения «Мобильное приложение по подготовки к ЕГЭ», предоставляющее возможность изучать экзаменационный материал и проходить тесты учащимся СОШ.

Для достижения поставленной цели разработано мобильное приложение, отвечающее следующим поставленным задачам: сбор и сохранение данных в БД.

В приложении реализованы регистрация и аутентификация для доступа к персонализированным функциям пользователя.

Программа позволяет пользователю просматривать обучающий материал, проходить тесты, смотреть свою статистику успеваемости.

В результате ВКР разработано мобильное приложение, полностью соответствующее поставленной задаче. Реализована базовая функциональная программа для изучения экзаменационных заданий учащимися СОШ.

В процессе ВКР использованы язык программирования JAVA для функциональной части, SQLite для хранения данных и язык разметки XML для графического отображения.

					ПГУ 1.090301.28.001 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Мобильное приложение по подготовки к ЕГЭ Пояснительная записка	Литера	Лист	Листов
Разраб.		Головинов В.Р.		02.06.25			4	90
Пров.		Евгоров В.Ю.		02.06.25				
Н. Контр.		Бычков А.С.		02.06.25				
Уте.		Митрохин М.А.		02.06.25		ФВТ, каф. ВТ гр. 21ВВП2		

Содержание

Перечень сокращений и обозначений.....	10
Введение.....	11
1 Мобильное приложения по подготовки к ЕГЭ	12
1.1 Анализ предметной области	12
1.1.1 Анализ темы	12
1.1.2 Анализ аналогов приложений.....	14
1.1.3 Постановка задачи	22
1.2 Выбор технологий и средств разработки приложения	24
1.2.1 Выбор языка программирования.....	25
1.2.2 Выбор библиотек для разработки приложения	26
1.2.3 Выбор СУБД.....	28
1.2.4 Выбор среды разработки	29
1.3 Архитектура приложения по подготовке к ЕГЭ.....	30
1.3.1 Монолитная архитектура	30
1.3.2 Архитектура приложения.....	30
1.4 Реализация приложения	32
1.4.1 Описание работы программы	32
1.4.2 Структура БД.....	34
1.5 Разработка приложения	37
1.5.1 Структура БД.....	37
1.5.2 Описание программы	38
1.5.3 Руководство пользователя	44
2 Охрана труда.....	49
2.1 Организация рабочего пространства.....	49
2.2 Освещение: баланс между комфортом и продуктивностью.....	50
2.3 Режим работы и отдыха.....	50
Заключение	52
Список использованных источников	53
Приложение А – UML диаграммы	54

Приложение Б – Листинг программы	59
Приложение В – Презентация.....	91

Перечень сокращений и обозначений

В настоящей работе применяют следующие обозначения и сокращения

БД – база данных

ВУЗ – высшее учебное заведение

ЕГЭ – единый государственный экзамен

СУБД – система управления базами данных

API (Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой

FK (Foreign key) – внешний ключ, понятие теории реляционных баз данных, относящееся к ограничениям целостности базы данных

HTML (HyperText Markup Language) – это язык разметки, используемый для создания веб-страниц

ID (Identifier) – уникальный номер объекта

IDE (Integrated development environment) – интегрированная среда разработки

PK (Primary key) – первичный ключ, понятие теории реляционных баз данных, относящееся к ограничениям целостности базы данных

SQL (Structured Query Language) – язык структурированных запросов

UML (Unified Modeling Language) – унифицированный язык моделирования

UNIQUE – уникальное значение

XML (Extensible Markup Language) - это расширяемый язык разметки, который используется для хранения и передачи данных в структурированном виде

Введение

Не так давно интернет стал частью повседневной жизни. Уже сейчас без него невозможно представить функционирование предприятий, медучреждений, государственных структур и даже образовательных учреждений. Современный интернет невозможно представить без социальных сетей, где у пользователей широкий спектр возможностей, в них люди могут общаться, обучаться, работать, отдыхать и развлекаться.

В современном мире, где время является одним из самых ценных ресурсов, много школьников хотят попасть в лучший ВУЗ. Важным аспектом успешного поступления является хорошая подготовка к экзаменам. От результата выполнения экзаменов зависит куда возможно будет поступить в дальнейшем. По этой причине очень важно получить все необходимые знания для успешного поступления.

Введение закона «О цифровых образовательных технологиях», которое регламентирует правовые основы использования интернета и цифровых технологий в обучении, показывает, какую большую роль он играет в образовании. Школы официально переходят на электронные учебники, дистанционные уроки. Больше времени уделяется просмотру обучающих видео из интернета, а также прохождению тестовых заданий на сайтах.

В последние годы наблюдается стремительный рост популярности онлайн-сервисов подготовки к ЕГЭ. Многие школьники предпочитают выбирать данные сервисы по той причине, что они более эффективно и интересно подают учебный материал, что облегчает его усвоение. Школьники имеют возможность сами выбирать преподавателя, который будет их обучать, смотреть объяснения своих ошибок и просматривать статистику успеваемости.

Данная ВКР посвящена разработке мобильного приложения для комфортного и эффективного обучения школьников.

1 Мобильное приложения по подготовки к ЕГЭ

1.1 Анализ предметной области

1.1.1 Анализ темы

Интернет сильно изменил процесс обучения, превратившись из вспомогательного инструмента в ключевой элемент образовательной системы. В настоящее время сложно представить, как проходит учеба без его помощи. Школы официально используют электронные учебники, дневники, проводят онлайн уроки.

Много школьников желают поступить в высшее учебное заведение. По этой причине появляется огромная конкуренция на желаемые места. Для того, чтобы набрать как можно больше баллов в ЕГЭ необходимо много времени на обучения и серьезная подготовка к заданиям. В настоящее время решающую роль играет эффективность и легкость изучения материала. Приложения по подготовки к ЕГЭ могут предоставить все необходимо для данного запроса.

Приложения для подготовки к экзаменам — это удобные цифровые инструменты, которые помогают эффективно учиться, систематизировать знания и повышать шансы на успешную сдачу экзаменов. Они имеют ряд преимуществ:

- персональный подход к пользователю, могут присылать уведомление и подстраивать сложность заданий под пользователя;
- интерактивный формат обучения. Предоставляет вместо скучных конспектов увлекательные видео разборы, которые улучшают запоминание материала;
- структурированная подготовка. Приложения часто предлагают готовые планы обучения, разбитые по темам и уровням сложности. Это помогает не тратить время на составление программы и сосредоточиться на изучении материала;
- экономия времени. Не нужно искать материалы в разных источниках, всё собрано в одном месте;

- тренировка в условиях близкая к экзамену. Помогает эмоционально подготовиться;
- гибкий график. Можно обучаться в удобное время, без привязки к расписанию;
- экономия денег. Многие приложения являются бесплатными или дешевле чем репетиторы;
- доступность. В интернете много сайтов, позволяющих готовиться к экзаменам. Их легко найти, многие из них предоставляют бесплатные материалы для обучения.

Основные преимущества таких приложений заключаются в их удобстве, скорости доступа к контенту и возможности выбора. Пользователю не нужно загружать большие файлы и хранить их на устройстве, что особенно удобно при ограниченном объеме памяти. Кроме того, современные приложения по подготовке к ЕГЭ предлагают персонализированные рекомендации, основываясь на предпочтениях пользователя, что делает опыт использования более приятным.

Стоит отметить и недостатки таких приложений, а именно: необходимость подключения к интернету, так как без доступа к сети не удастся пользоваться приложением в полной мере; зависимость от сервера, если сервер выйдет из строя, то некоторые данные будут недоступны пользователю. Многие приложения являются платными или большая часть их функциональности работает только по подписке. Может быть недостаточно материала для обучения или доступ к нему ограничен по времени, как например прямой эфир, которые можно посмотреть только в настоящий момент.

Кто бы что не говорил, современные школьники охотно продолжают пользоваться приложениями по подготовке к ЕГЭ. Исследование показало, что самостоятельно готовятся менее 40% учеников. Хотя ЕГЭ и позволяет максимально уравнивать шансы детей с периферии и из центра, но получить высшие баллы без репетитора крайне сложно, отмечают эксперты. Исходя из данных статистики следует отметить тот факт, что сервисами пользуется много школьников и с каждым годом их количество растет. Со временем данная

тенденция будет сохраняться, возможно в скором времени приложения, нацеленные на обучение, станут основным способом подготовки, так как они показывают хорошие результаты. Многие школьники, обучающиеся с помощью таких приложений, очень быстро осваивают материал.

Проанализировав данную тему хочется сказать, что приложение для подготовки школьников к экзаменам является актуальной темой в настоящее время, так как будет пользоваться большим спросом среди них.

1.1.2 Анализ аналогов приложений

Обучающие приложения — это цифровые платформы, которые позволяют пользователям просматривать видео на различных устройствах, таких как смартфоны, планшеты и персональные компьютеры. На рынке существует много различных приложений, которые имеют большой функционал, упрощающий освоение материала. Эти сервисы предоставляют доступ к обширному архиву заданий с объяснениями их решения.

Чтобы выбрать варианты с чем сравнивать, нужно смотреть на какой способ обучения оно делает упор, так как каждое приложение является уникальным продуктом и имеет свои удобства, которые выделяют его среди остальных:

- удобный интерфейс. Позволяет быстро ориентироваться и понимать, что где находится;
- возможность решать задания и смотреть способ их выполнения;
- видео-объяснение материала. Дает больше полезной информации и наглядно показывает, как выполняется задание;
- индивидуальный подход. Имеется возможность общаться с преподавателем насчет заданий;
- анализ прогресса. Позволяет вести статистику слабых и сильных сторон.

Для более правильной реализации приложения, необходимо проанализировать аналоги, выявить их слабые и сильные стороны, а также определить основные требования к разрабатываемому приложению.

В этом пункте будут рассмотрены три популярных приложения для обучения, а именно:

- «Quizlet»;
- «Турбо»;
- «НоваяШкола-Тренажер».

"Quizlet" - это бесплатный сервис, который позволяет легко запоминать любую информацию, которую можно представить в виде учебных карточек. Предоставляет инструменты для учебы и обучения. Интерфейс приложения можно рассмотреть на рисунках 1 и 2.

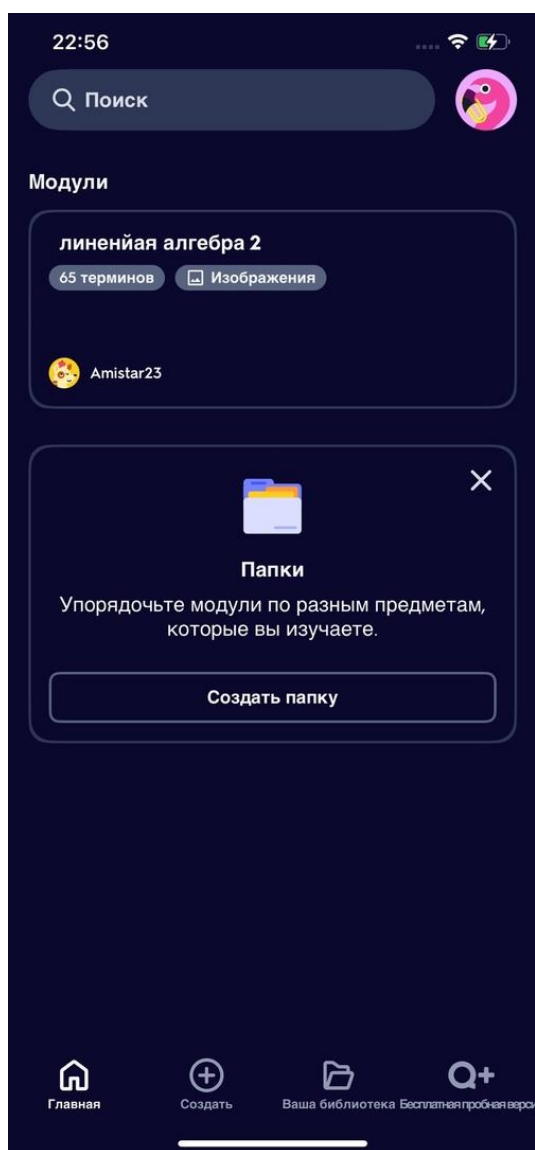


Рисунок 1 - Главная страница приложения

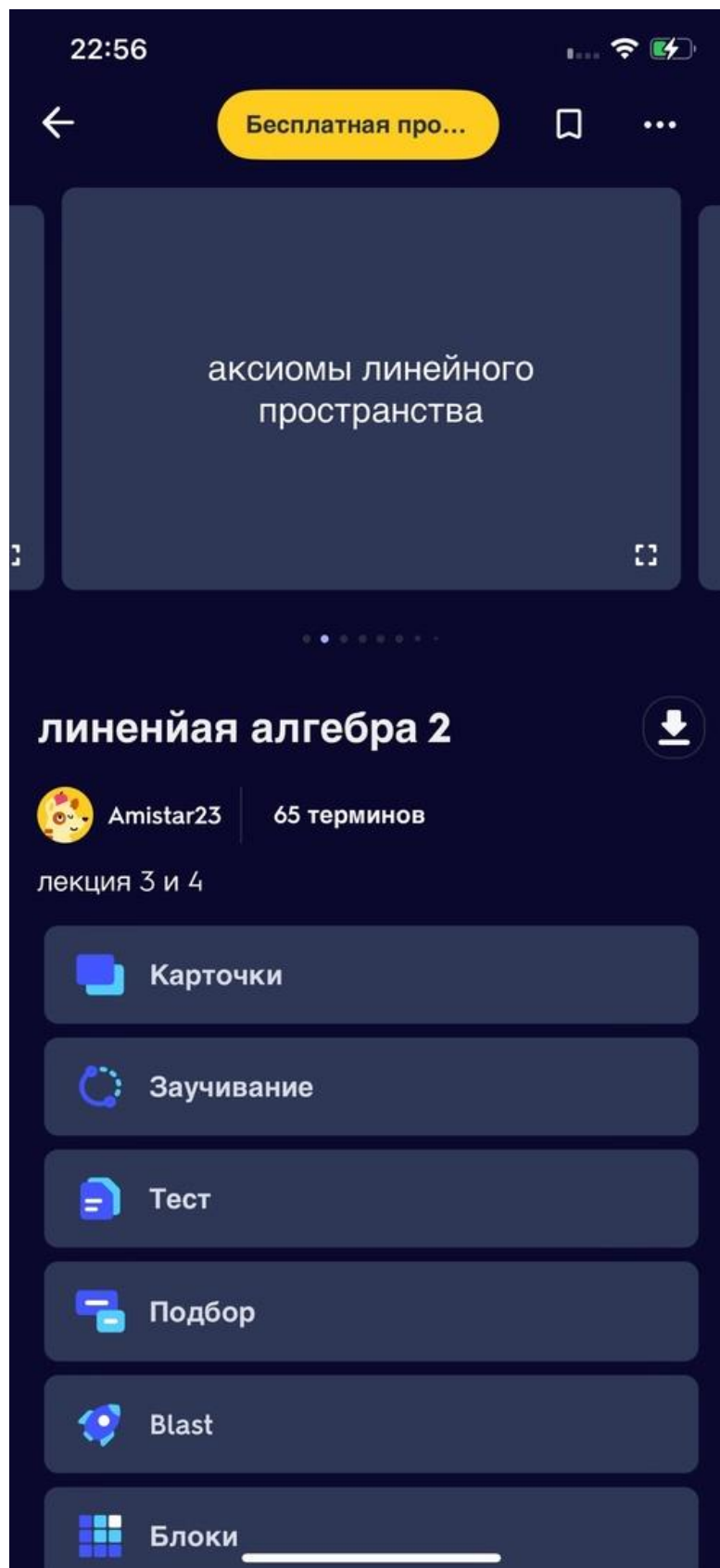


Рисунок 2 – Выбор заданий

Приложение «Quizlet» имеет большое количество различных функций, предоставляемые пользователю. Ниже представлены основные из них:

- Большой выбор карточек для изучения;

Основным недостатком данного сайта можно выделить:

- отсутствие какого-либо видео объяснений, только текстовый формат;
- нельзя выполнять задания с развернутым ответом;
- имеет устаревшие задания;
- много рекламы;
- не имеет индивидуальный подход.

Приложение «Турбо» - это мобильное приложение, разработанное онлайн-школой "Турбоподготовка". Оно предназначено для подготовки к ЕГЭ, предлагая пользователям доступ к онлайн-курсам, конспектами. Интерфейс приложения можно рассмотреть на рисунках 3 и 4.

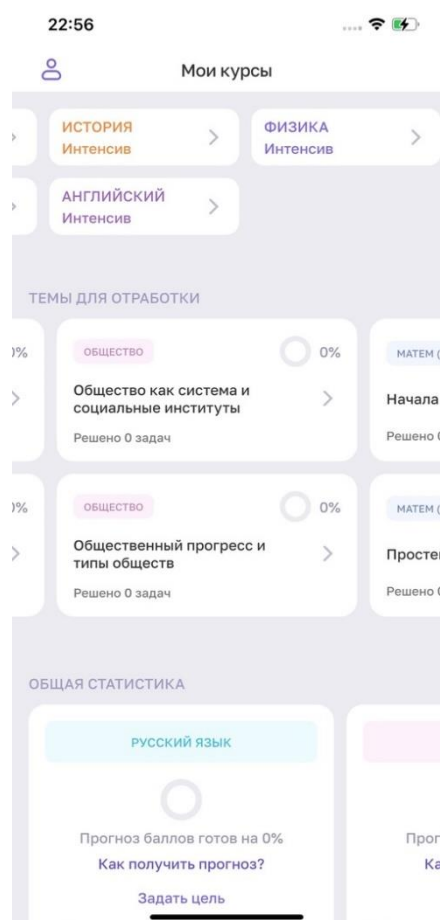


Рисунок 3 - Главная страница сайта «Турбо»

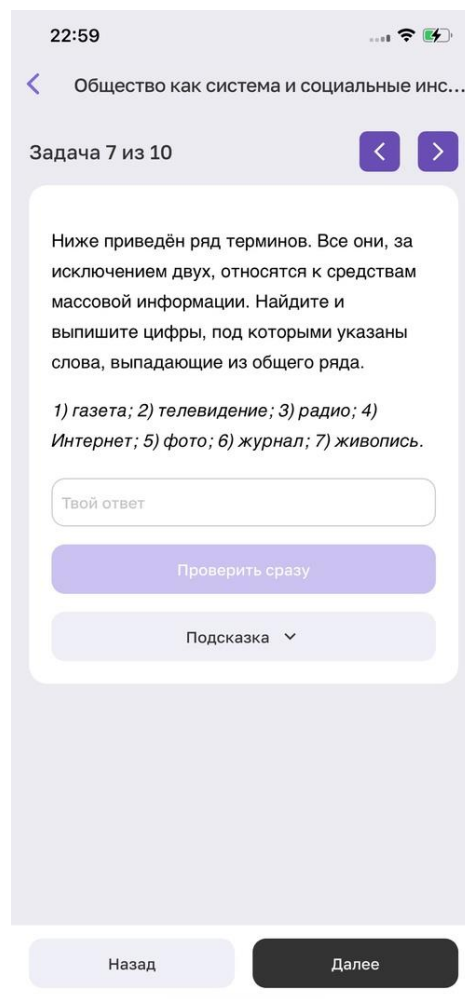


Рисунок 4 - Страница с услугами «Турбо»

Приложение «Турбо» имеет большое количество различных функций, предоставляемые пользователю. Ниже представлены основные из них:

- разные форматы обучения: онлайн-курсы, индивидуальные занятия;
- большой выбор предметов для изучения;

Основным недостатком данного сайта можно выделить:

- все курсы платные;
- качество получения учебного материала зависит от преподавателя;
- многие задания сделаны неправильно;
- тренировочные тесты сильно ограничены.

«НоваяШкола-Тренажер»- это мобильное приложение, предназначенное для подготовки к единому государственному экзамену (ЕГЭ). Оно предоставляет возможность тренироваться по всем предметам, таким как русский язык,

математика, информатика, физика, биология, английский язык, химия, история, обществознание, география и литература. Интерфейс приложения можно рассмотреть на рисунках 5 и 6.

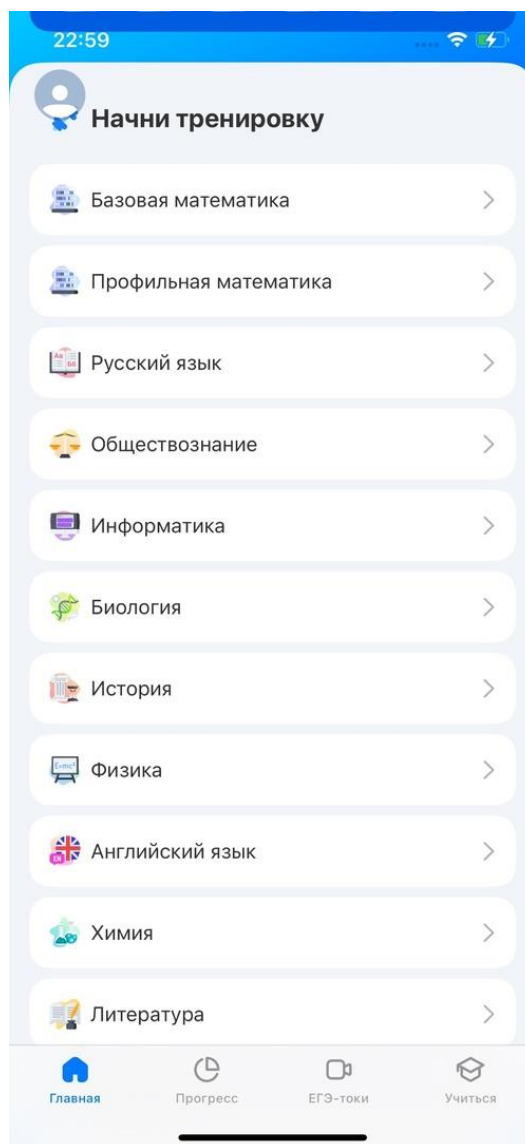


Рисунок 5 - Главная страница «НоваяШкола-Тренажер»

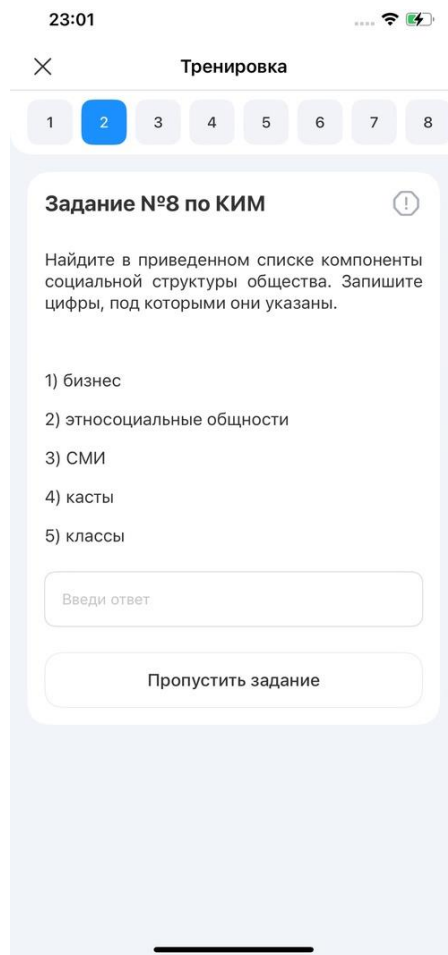


Рисунок 6 - Страница с заданиями

Сайт «НоваяШкола-Тренажер» имеет большое количество различных функций,

предоставляемые пользователю. Ниже представлены основные из них:

- большой выбор предметов;
- много заданий;

Основными недостатками данного сайта являются:

- ограниченный функционал;
- мало теории;
- нет просмотра статистики.

Исходя из анализа аналогов следует отметить, что все они являются хорошими и конкурентоспособными приложениями для подготовки к ЕГЭ. У

каждого есть свои плюсы и минусы, особые функции, которые выделяются среди других похожих приложений.

Результаты анализа существующих аналогов можно представить в таблице 1, показывающие минусы и плюсы рассматриваемых сервисов.

Таблица 1 – Достоинства и недостатки приложений по подготовке к ЕГЭ

Функционал	«Quizlet»	«Турбо»	«НоваяШкола-Тренажер»
Удобный и понятный интерфейс	-	+	+
Бесплатные занятия	+	-	-
Большой каталог заданий	-	-	+
Индивидуальный подход	-	+	+
Проверка заданий с развернутым ответом	-	-	+
Мгновенный разбор ошибок	+	-	-
Большой выбор предметов	+	+	-

Проанализировав данные онлайн-сервисов, было решено разработать собственное приложение для подготовки к ЕГЭ. Данное приложение будет соответствовать основным требованиям, которые предоставляет пользователь доступ к учебному материалу, а именно:

- удобный интерфейс;
- все основные функции, необходимые для комфортного использования;
- возможность авторизованным пользователям вести статистику своего обучения для анализа своих результатов в обучении;
- возможность удобного поиска заданий по необходимому предмету.

Помимо базового функционала, будут реализованы функции, которые отсутствуют у аналогов и которые будут привлекательны конечному пользователю.

Исходя из анализа темы, много людей используют преимущества технологий в качестве быстрого и комфортного способа получения необходимой информации для развития, поэтому было принято решение сделать приложение для этой целевой аудитории. Будет разработан продукт для подготовки к экзаменам в телефоне, который является приоритетным оборудованием в данной сфере.

1.1.3 Постановка задачи

Приложение для подготовки к ЕГЭ должно реализовывать следующий функционал:

Со стороны клиента:

- форма для авторизации и регистрации;
- возможность выбрать урок для обучения;
- возможность пройти тестовые задания;
- просмотр видео;
- изменить профиль;
- анализ статистики;
- редактирование профиля;
- поиск заданий по критериям.

Минимальные системные требования:

- операционная система: Android;
- минимальная скорость интернета: 1 Мбит/сек;
- минимальный объем оперативной памяти: 2 ГБ;
- минимальный объем свободной памяти: 150 МБ.

Данное приложение имеет функционал, который разделен на две категории, а именно:

- для оффлайн пользователя;
- для авторизованного пользователя.

Оффлайн пользователь имеет возможность пользоваться ограниченным набором функций, а именно:

- авторизация в приложении;
- просмотр обучающих тем;
- прохождение тестов;
- просмотр справочника;
- регистрация в приложении.

Возможные варианты действий неавторизованного пользователя можно рассмотреть на рисунке 7.

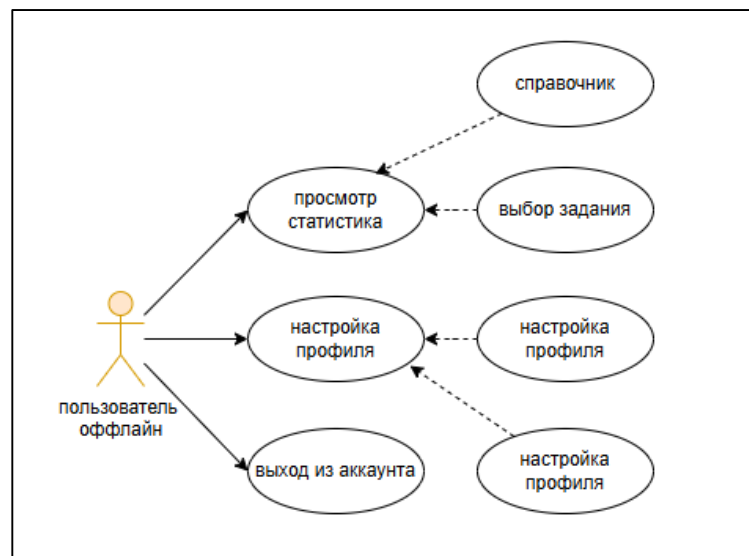


Рисунок 7 – UML диаграмма действий оффлайн пользователя

Авторизированному пользователю будет доступен расширенный функционал работы программы. Функциональные возможности, доступные авторизированному пользователю:

- просмотр видео;
- скачивание видео;
- просмотр обучающих тем;
- просмотр справочника;
- вход в приложение;
- прохождение тестов;
- прохождение экзамена;
- просмотр статистики;

- выход из аккаунта;
- настройка профиля;
- изменение персональных данных;
- удаление аккаунта;
- очистка статистики.

Возможные варианты действий неавторизованного пользователя можно рассмотреть на рисунке 8.

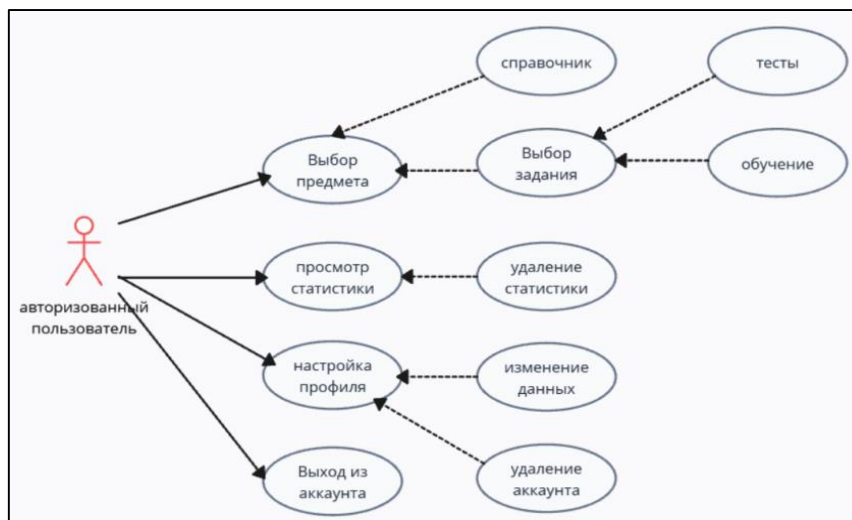


Рисунок 8 – UML диаграмма действий авторизованного пользователя

1.2 Выбор технологий и средств разработки приложения

В данном разделе будут рассмотрены технологии и средства разработки, которые были использованы для реализации мобильного приложения для подготовки к ЕГЭ.

Ниже приведены технологии и средства разработки, которые были выбраны:

- Java;
- JUnit;
- AGP;
- AppCompatActivity;
- OkHttpClient;

- Play-services-auth;
- Google-api-services-drive;
- Google-http-client-android;
- Google-api-client-android;
- SQLite;
- Android Studio.

1.2.1 Выбор языка программирования

На текущий момент есть широкий выбор языков программирования, как правило высокоуровневых, но бывают и исключения, для написания клиент-серверных программ: Java, C#, Python, C++ и др.

C++ – компилируемый и статически типизированный язык программирования. Поддерживает ООП и процедурное программирование. Отличается высоким быстродействием, но для реализации клиент-серверного приложения требует очень много времени.

C# – компилируемый и статически типизированный язык программирования. Объектно-ориентированный язык программирования. Уступает по скорости C++, но обладает более ёмким, в плане возможностей, синтаксисом.

Java – является как компилируемым, так и интерпретируемым языком программирования. Код сначала интерпретируется, а затем при помощи JIT компилятора переводится в машинный код, что позволяет запустить его почти на любой платформе, где есть реализованная Java-машина. Java обладает большим количеством фреймворков и библиотек, позволяет решать почти любые задачи.

JavaScript – потомок языка Java, что сильно отразилось на его синтаксисе. В начале своего существования использовался преимущественно как язык сценариев для интерактивных веб-страниц, однако, со временем, язык обрел популярность и для него появилось средство транслирования в машинный код – NodeJS. Это позволило использовать язык в самых различных сценариях. Язык

поддерживает большое количество библиотек и фреймворков. Также преимуществом языка является автоматическое управление памятью.

Python – интерпретируемый и динамически-строго типизируемый. Существует две реализации языка: Python 2 (поддержка прекращена) и Python 3. Язык обладает простым синтаксисом, предоставляет широкие возможности разработки самых разных алгоритмов, обладает огромным количеством библиотек под самые разные задачи (от вывода в консоль цветного текста до разработки нейронных сетей), т.к. активно поддерживается сообществом разработчиков. Язык является интерпретируемым, что негативно сказывается на скорости работы программ по сравнению со всеми вышеперечисленными языками, однако, Python позволяет использовать библиотеки и модули, написанные на других языках, например, Numpy и Pandas. Также главным преимуществом Python является встроенный Garbage Collector (сборщик мусора), который автоматически управляет памятью программы.

Go (golang) – компилируемый многопоточный язык программирования. Разработан в Google. Молодой и активно растущий язык программирования. Идеально подходит для реализации серверных приложений, однако библиотек для языка не так и много, что затрудняет его использование при реализации графического интерфейса клиентских приложений, а также нейронных сетей.

Рассмотрев все возможные варианты, и, сравнив все сильные и слабые стороны языков, был выбран язык JAVA, по той причине, что он: считается очень надежным и безопасным, а также кроссплатформенный – это позволяет охватить как можно больше пользователей своим приложением, высоко производительный, что является важным фактором при работе приложения.

1.2.2 Выбор библиотек для разработки приложения

В программе имеется набор библиотек, необходимых для реализации сложных действий, запросов, тестировки, отображении интерфейса. В набор входят также несколько библиотек, которые автоматически подключаются при разработке программы в Android Studio для базового функционала программы.

JUnit – это фреймворк для модульного тестирования (unit testing) в Java, который позволяет автоматизировано проверять корректность работы отдельных компонентов кода (методов, классов). Проверяет ожидаемые результаты, запуская тест с разными данными.

AGP (Android Gradle Plugin) – это плагин для системы автоматизации сборки Gradle, который добавляет возможность компиляции Android приложений. Он позволяет Gradle обрабатывать специфичные для Android задачи, такие как сборка APK, ресурсов и других необходимых для Android компонентов.

AppCompat – это вспомогательная библиотека для Android, которая позволяет разработчикам использовать более новые элементы UI и функции в старых версиях Android. Она предоставляет обратную совместимость для использования современного Material Design в приложениях.

OkHttp – это библиотека для Java и Android, предоставляющая простой и эффективный способ отправки и получения HTTP-запросов и ответов. Она разработана Square и часто используется в качестве альтернативы или дополнения к встроенным HTTP-клиентам SDK, таких как HttpURLConnection.

Play-services-auth – это часть Google Play Services, которая предоставляет удобные API для аутентификации пользователей через Google-аккаунт. Позволяет выполнить быстрый вход без пароля в аккаунт, а также получить token для получения доступа к Google Drive, Gmail, YouTube.

Google-api-services-drive – это официальная клиентская библиотека Google Drive API для Java, которая позволяет взаимодействовать с Google Диском (Google Drive) из вашего приложения. С её помощью можно искать, получать, загружать и скачивать файлы.

Google-http-client-android – это низкоуровневая библиотека от Google, предоставляющая оптимизированный HTTP-клиент для работы с RESTful API на Android. Реализует низкоуровневые HTTP-операции (GET, POST, PUT, DELETE). Она является частью экосистемы Google API Client Library for Java, но адаптирована под мобильные устройства с учётом ограничений Android (например, работы в основном потоке).

Google-api-client-android – это высокоуровневая библиотека от Google, которая предоставляет удобный способ взаимодействия с Google API (такими как Drive, YouTube, Gmail, Calendar и другими) из Android-приложений. Упрощает работу с API, автоматически сериализует и десериализует JSON. Она является частью Google API Client Library for Java, но оптимизирована для работы на Android-устройствах.

1.2.3 Выбор СУБД

На текущий момент предоставлено большое количество видов баз данных и систем управления ими. Существует два основных класса БД: реляционные (SQL) и не реляционные (NoSQL). Основное отличие заключается в представлении данных.

Данные в не реляционные БД хранятся, в основном, в виде словарей, где ключом является первичный ключ сущности, а значением остальные поля таблицы. Подобный подход позволяет абстрагироваться от строгой типизации и строгого формата данных. Также NoSQL БД позволяют хранить иерархические структуры данных, в то время как для реализации подобной задачи с помощью SQL БД потребовалось бы создавать дополнительные таблицы. Наиболее популярными NoSQL СУБД являются MongoDB, HBase, Cassandra.

Реляционные БД называют также табличными. SQL БД также подразделяются на два вида: строковые и колоночные. Основное различие заключается в способе хранения данных.

В колоночных БД данные одной колонки физически расположены близко друг к другу на жестком диске, это позволяет наиболее эффективно сжимать информацию. Обычно такого типа БД используются для записи логов и в аналитических целях, например, компания Яндекс специально для своего сервиса Яндекс.Метрика разработала СУБД ClickHouse. Но существенным недостатком является крайне низкая скорость обновления данных.

В строковых БД данные на физическом носителе хранятся построчно, что увеличивает время обработки и выборки, однако позволяет быстро обращаться к нужному элементу по идентификатору, изменять его или удалять. Также

некоторые строковые СУБД более эффективны по сравнению с NoSQL. Наиболее известными реляционными СУБД являются: PostgreSQL, MySQL, MS SQL.

Главным фактором в выборе базы данных является эффективность ее использование. Лучшим вариантом для дальнейшей работы можно считать SQLite по той причине, что она уже интегрирована в Android Studio и уже оптимизирована для работы на мобильных устройствах. Также большим плюсом можно считать независимость SQLite от подключения к интернету, данные базы данных хранятся в файле .db, к которому всегда можно получить доступ.

1.2.4 Выбор среды разработки

IDE (Integrated Development Environment) – это программный пакет, который предоставляет программистам интегрированную среду для разработки программного обеспечения. В него входят различные инструменты, необходимые для создания, тестирования и отладки кода, такие как текстовый редактор, компилятор, отладчик и другие. IDE упрощает процесс разработки, объединяя все необходимые инструменты в одном месте.

Android Studio – это интегрированная среда разработки (IDE) для работы с платформой Android. Данная платформа включает в себя набор библиотек и инструментов для удобной и быстрой разработки различного программного обеспечения любого уровня сложности на смартфоне. Стоит отметить, что Android studio обладает достаточным функционалом и инструментарием для разработки на разные оборудования: телефоны, планшеты, телевизоры. Имеется возможность создать детальный эмулятор требуемого оборудования, который можно настроить под свои цели.

IntelliJ IDEA – интегрированная среда разработки (IDE), написанная на Java для разработки программного обеспечения на Java, Kotlin, Groovy и других языках на основе JVM. позволяют разработчикам быстрее создавать отличное программное обеспечение за счет написания чистого, качественного кода и автоматизации задач на всех этапах разработки.

Visual Studio Code – это легковесный, но мощный редактор кода от Microsoft с поддержкой Android-разработки через расширения. Поддерживает

широкий спектр языков программирования (Java, Kotlin, Python). VS Code обладает обширной экосистемой расширений, позволяющих расширить его функциональность и добавить поддержку других языков и технологий. Хотя он не заменяет Android Studio для сложных задач, его тоже можно использовать.

В результате был выбран Android Studio, как самая эффективная среда разработки из имеющихся для Android. В этой среде разработки для данной платформы будет написана функциональная часть приложения и интерфейс пользователя.

1.3 Архитектура приложения по подготовке к ЕГЭ

1.3.1 Монолитная архитектура

Монолитная архитектура разработки является одной из традиционных подходов, в которых все компоненты программы объединены в единый блок и развертывание происходит как одно целое. При данном подходе весь код находится в одном проекте, что обеспечивает полный доступ ко всем компонентам, а также контроль над каждым действием. Благодаря этому удобно тестировать программу и следить за взаимодействием всех блоков. Монолитная архитектура обеспечивает высокую производительность в отличие от микросервисных и клиент-сервисных архитектур, так как не имеет накладных расходов на работу с удаленными компонентами. С данным подходом легко разрабатывать программу по той причине, что не надо создавать способы взаимодействия с другими интегрируемыми компонентами, доступ к которым мы не имеем.

1.3.2 Архитектура приложения

MVC (Model-View-Controller) –это популярный шаблон проектирования, который используется для разделения компонентов приложения на три основные части: модель (Model), представление (View) и контроллер (Controller). Каждая из этих частей выполняет свою определенную функцию, из-за чего упрощается разработка, поддержка и масштабируемость приложения.

Модель – это компонент, который отвечает за работу с данными приложения. Модель представляет собой структуру данных, бизнес-логику и методы доступа к данным. Она не зависит от представления или контроллера и обеспечивает абстракцию данных, что позволяет легко изменять данные без влияния на другие части приложения.

Представление – набор классов и шаблонов, отвечающих за отображение данных пользователю. Оно представляет собой интерфейс пользователя, который взаимодействует с моделью. Представление отображает данные из модели пользователю и передает пользовательские действия контроллеру.

Контроллер – это связующее звено между моделью и представлением. Он обрабатывает пользовательские запросы, взаимодействует с моделью для получения или изменения данных и передает данные в представление для отображения пользователю.

Преимущества такого разделения в сравнении с монолитной архитектурой:

- чёткое разделение ответственности: Каждый компонент отвечает за свою область;
- повышенная модульность: Можно менять реализацию одного компонента без затрагивания других;
- упрощение поддержки и тестирования. Легко тестировать бизнес-логику отдельно от UI;
- параллельная разработка: Разработчики могут работать над моделью, дизайнерами — над представлением, а контроллеры обеспечивают связь.

Диаграмма архитектуры приложения изображена на рисунке 9, а диаграмма компонентов изображена на рисунке 10.

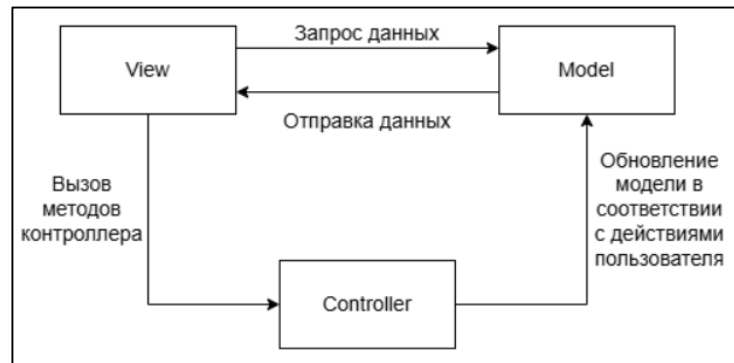


Рисунок 9 – Архитектура приложения

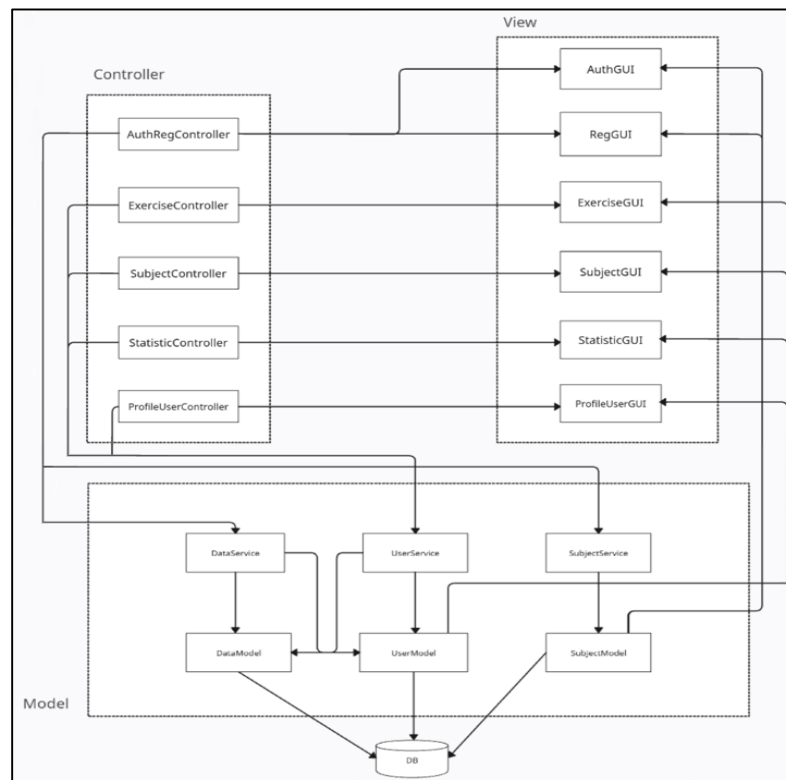


Рисунок 10 – диаграмма компонентов

1.4 Реализация приложения

1.4.1 Описание работы программы

Алгоритм работы приложения заключается в том, что пользователь взаимодействует с клиентской частью, написанной на WPF. В зависимости от того, какие действия выполняет пользователь, будет формироваться и отправляться запрос на сервер. Сервер в свою очередь будет обрабатывать запрос. Если в процессе обработки запроса происходит ошибка, сервер сообщает об этом

клиенту. В случае успешной обработки запроса сервер взаимодействует с базой данных для выполнения операций, таких как создание, редактирование, получение или удаление данных. После успешного выполнения операции с базой данных сервер возвращает клиенту необходимые данные или подтверждение выполнения операции. Если при обращении к базе данных возникает ошибка, сервер обрабатывает эту ошибку и возвращает информацию о ней клиенту.

Последовательность выполняемых действий авторизованных и не авторизованных пользователей представлена на рисунке 11.

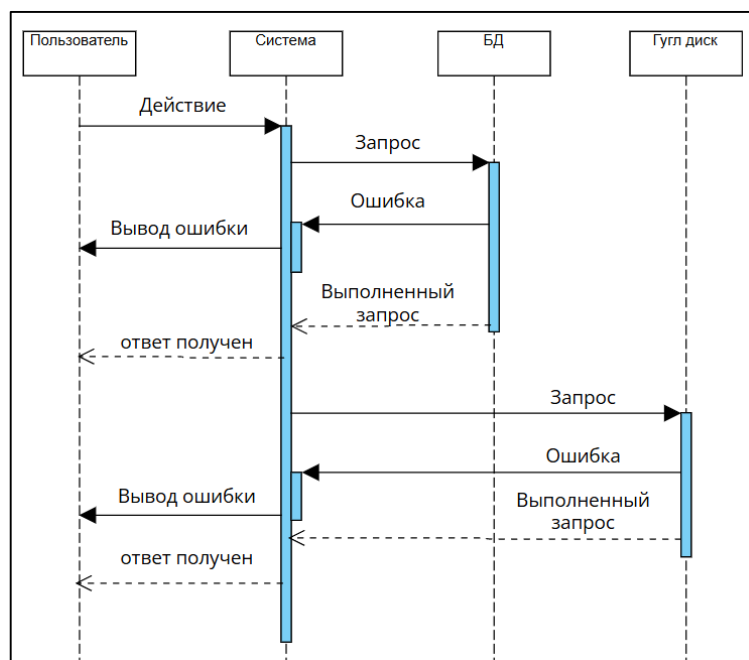


Рисунок 11 – UML диаграмма последовательности функционирования приложения

Топология системы и распределение ее компонентов по физическим узлам представлены на рисунке 12.

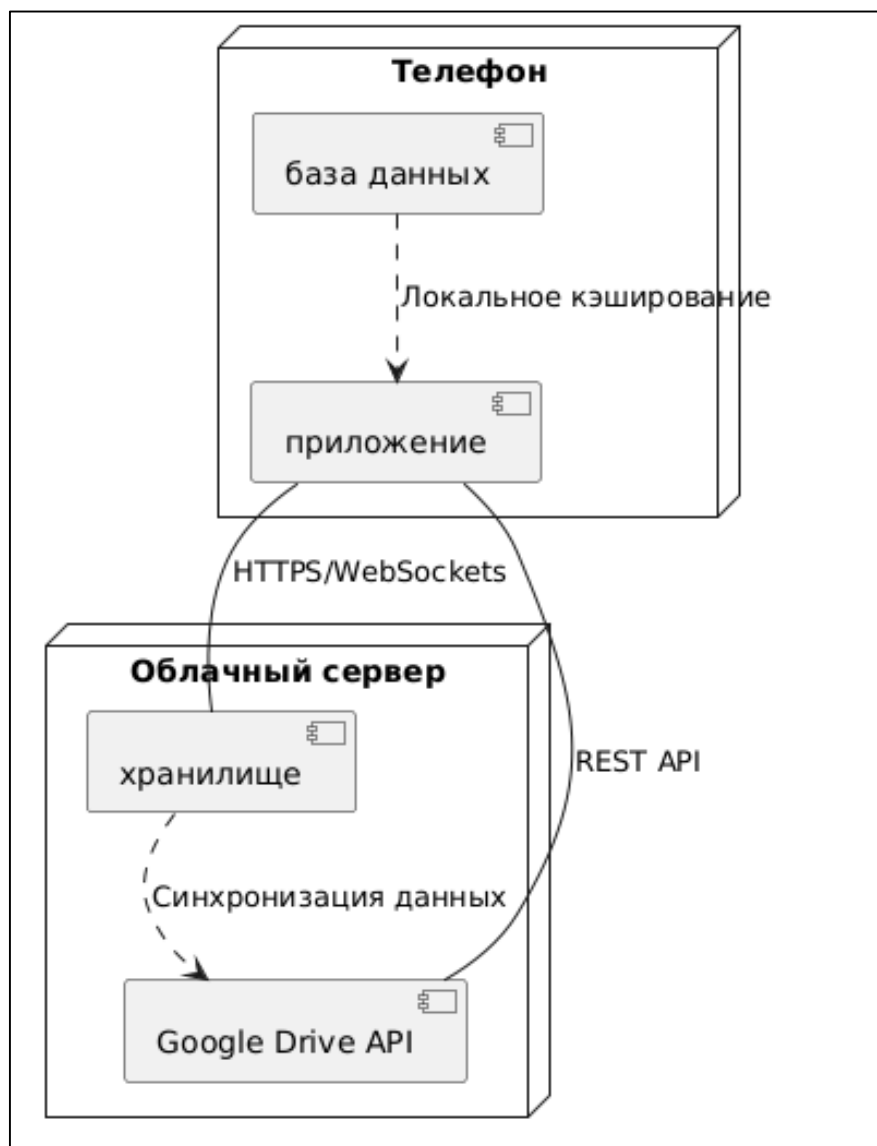


Рисунок 2 – UML диаграмма развертывания системы

1.4.2 Структура БД

База данных мобильного приложения по подготовки к ЕГЭ спроектирована для хранения информации о пользователях, предметах, заданиях, истории прохождения тестов.

В данном разделе рассматриваются основные таблицы, атрибуты и связи между ними, которые были разработаны в рамках данной работы.

Проведя анализ, было решено, что в разрабатываемом приложении будут использоваться в таблице 2, 3, 4, 5.

Таблица 2 – Атрибуты таблицы users (Пользователи системы):

№	Поле	Тип данных	Ограничения	Описание
1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Уникальный идентификатор пользователя
2	username	TEXT	NOT NULL, UNIQUE	Имя пользователя для входа в систему
3	password	TEXT	NOT NULL	Хеш пароля пользователя
4	state	TEXT	NOT NULL	Данные профиля

Правила для реализации таблицы:

- каждый пользователь должен иметь уникальные username;
- пароли хранятся в зашифрованном виде;
- данные профиля не могут быть нулевыми.

Таблица 3 – Атрибуты таблицы subject (предмет)

№	Поле	Тип данных	Ограничения	Описание
1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Уникальный идентификатор статистики
2	subjectName	TEXT	NOT NULL	Название задания
3	taskName	TEXT	NOT NULL	Название Задание
4	taskType	TEXT	NOT NULL	Название типа
5	count	INTEGER	NOT NULL	Количество
6	complete	INTEGER	NULL	Количество заданий

Правила для реализации таблицы:

- количество выполненных заданий изначально равно 0;
- таблица создается для каждого пользователя.

Таблица 4 – Атрибуты таблицы statistic (статистика заданий)

№	Поле	Тип данных	Ограничения	Описание
---	------	------------	-------------	----------

1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Уникальный идентификатор статистики
2	user_id	INTEGER	FOREIGN KEY	Ссылка на пользователя
3	subject_id	INTEGER	FOREIGN KEY	Ссылка на предмет

Правила для реализации таблицы:

- каждая статистика должна иметь уникальные user_id;
- каждая статистика должна иметь уникальные subject_id;
- каждый предмет имеет определенное количество заданий.

Таблица 5 – Атрибуты таблицы Url (ссылки на обучающие видео)

№	Поле	Тип данных	Ограничения	Описание
1	id	INTEGER	PRIMARY KEY, AUTOINCREMENT	Уникальный идентификатор видео
2	url	TEXT	UNIQUE	Путь к видео

Правила для реализации таблицы:

- имеет уникальную ссылку на предмет;
- должна использоваться для быстрого доступа к хранилищу данных.

Данная база данных имеет 4 таблицы, которое взаимодействуют друг с другом с помощью внешних ключей, что позволяет поддерживать актуальность данных. На основе данных таблиц было создана физическая модель БД, представленная на рисунке 13.

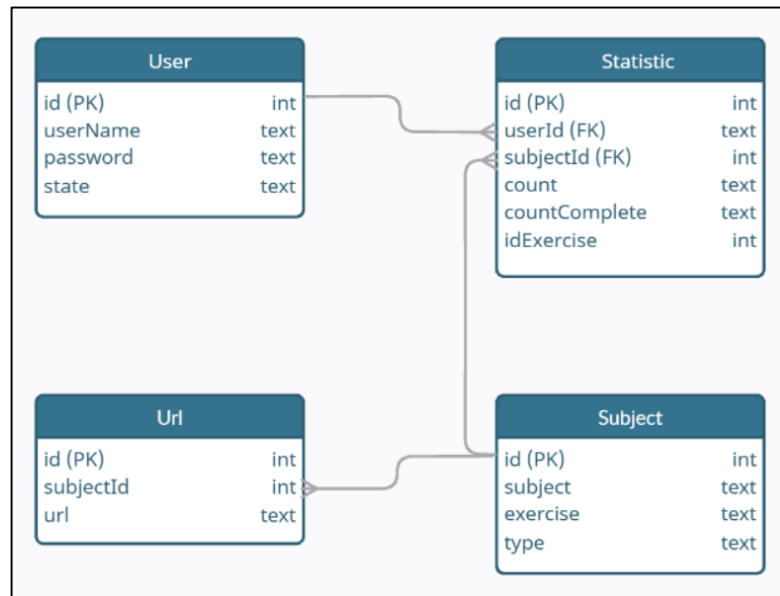


Рисунок 13 – Физическая модель БД

1.5 Разработка приложения

1.5.1 Структура БД

MyDataBaseHelper - центральный компонент доступа к данным.

DatabaseHelper наследуется от SQLiteOpenHelper и реализует паттерн DAO (Data Access Object) для каждой сущности.

Сначала мы определяем характеристики таблицы

```
private static final String DataBaseName = "myDataBase.db";
private static final int DataBaseVersion = 1; // версия базы
данных
```

Далее мы объявляем статические константы, которые содержат: названия таблиц и названия полей в таблицах.

```
public static class User
public static final String TableUser = "tableUser";
public static final String COLUMN_NAME = "name";
public static final String COLUMN_PWD = "pwd";
public static class Subject
public static final String COLUMN_NAME_SUBJECT = "subjectName";
public static final String COLUMN_NAME_TASK = "taskName";
public static final String COLUMN_NAME_TYPE = "typeName";
public static final String COLUMN_COUNT_TYPE = "count";
```

```

public static final String COLUMN_COMPLETE_TYPE = "complete";
public static class Statistic
public static final String COLUMN_USER_ID = "user_id"
public static final String COLUMN_SUBJECT_ID = "subject_id"

```

Также при создании таблиц используются статические строки и функции, являющиеся готовыми шаблонами для реализации таблиц, которые упрощают их создание и уменьшают шанс ошибиться.

```

static final String PRIMARY_KEY = " INTEGER PRIMARY KEY
AUTOINCREMENT";
public static final String ID = "_id";
public static String FOREIGN(String foreignData, String
tableName, String refData){
return "FOREIGN KEY ( " + foreignData + " ) REFERENCES " +
tableName + " ( " + refData + " );";}

```

1.5.2 Описание программы

В данном подразделе будет описан основной функционал программы. Будут описаны компоненты, отвечающие за отображение интерфейса, работу с данными.

Класс DriveServiceHelper делает API-запрос для получения доступа к гугл-диску, где хранятся обучающие видео. Данный класс играет важную роль по той причине, хранит ролики, которые весят более 100мб, тем самым программа избегает большой нагрузки.

В данной функции программа готовит запрос, который будет послан в гугл-диск для доступа к файлам.

```

FileList result = drive.files().list().setQ("mimeType =
'video/mp4'").setFields("files(id,name,webViewLink,webContentLink)")
.execute();

```

При успешном результате мы получаем необходимые данные.

Одними из важных файлов, являются файлы subject_resource, exercise_set.xml, и task_resource.

В файле subject_resource хранятся данные о предметах

```

    <resources><arrayname="subjects_image"><item>@drawable/informat
ics</item><item>@drawable/mathematics</item><item>@drawable/physics<
/ item></array><string-
arrayname="subjects_name"><item>Информатика</item><item>Математика</
item><item>Физика</item></string-array><string-array
name="subjects_description"><item>наука о методах и процессах сбора,
хранения, обработки, передачи, анализа и оценки информации с
применением компьютерных технологий</item><item>наука об отношениях
между объектами, о которых ничего не известно, кроме описывающих их
некоторых свойств, – именно тех, которые в качестве аксиом положены
в основание той или иной математической
теории</item><item>фундаментальная наука о наиболее общих законах
природы, о материи, её структуре, движении и правилах
трансформации.</item></string-array>
</resources>

```

В файле task_resource хранятся данные о видах заданий

```

<?xml version="1.0" encoding="utf-8"?><resources><string-array
name="task_name_1"><item>Анализ информационных
моделей</item><item>Построение таблиц истинности логических
выражений</item><item>Поиск информации в реляционных базах
данных</item></string-array><string-array name="task_name_2">
<item>Текстовые задачи</item><item>Единицы измерения</item>
<item>Графики и диаграммы</item></string-array><string-array
name="task_name_3"><item>Кинематика. Расчёт физической
величины</item> <item>Динамика. Расчёт физической
величины</item><item>Законы сохранения в механике. Расчёт физической
величины</item> </string-array></resources>

```

В файле exercise_set хранятся данные о заданиях

```

<?xml version="1.0" encoding="utf-8"?><resources><string-array
name="exercise_1_type_1"><item>На рисунке схема дорог Н-ского района
изображена в виде графа, в таблице содержатся сведения о дорогах
между населенными пунктами (звездочка означает, что дорога между
соответствующими городами есть).\n\t\tТак как таблицу и схему
рисовали независимо друг от друга, то нумерация населённых пунктов в
таблице никак не связана с буквенными обозначениями на графе.

```

Определите номера населенных пунктов А и Г в таблице. В ответе запишите числа в порядке возрастания без разделителей.

На рисунке слева изображена схема дорог Н-ского района, в таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет. Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам А и Г на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

На рисунке схема дорог изображена в виде графа, в таблице звёздочкой обозначено наличие дороги между населёнными пунктами. Отсутствие звёздочки означает, что такой дороги нет. Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какие номера населённых пунктов соответствуют населённым пунктам Б и В. В ответе запишите эти два номера в порядке возрастания без пробелов и знаков препинания.

Пример. Пусть населённым пунктам Д и Е соответствуют номера П1 и П2. Тогда в ответе нужно написать «12».

На рисунке слева изображена схема дорог Н-ского района, в таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет. Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам В и С на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

На рисунке слева изображена схема дорог Н-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет. Каждому населённому пункту на схеме

соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам Е и G на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.</item>

<item>На рисунке слева изображена схема дорог N-ского района. В таблице звёздочкой обозначено наличие дороги из одного населённого пункта в другой. Отсутствие звёздочки означает, что такой дороги нет.\n\t\tКаждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам Е и F на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.</item>

</string-array>

<arrayname="exercise_1_type_1_image">

<item>@drawable/info_1_1_1</item>

<item>@drawable/info_1_1_2</item>

<item>@drawable/info_1_1_3</item>

<item>@drawable/info_1_1_4</item>

<item>@drawable/info_1_1_5</item>

<item>@drawable/info_1_1_6</item>

</array>

<string-arrayname="exercise_1_type_1_answer">

<item>Ответ35</item><item>Ответ67</item><item>Ответ26</item><item>От
вет 26</item><item>Ответ45</item> <item>Ответ46</item> </string-
array>

<string-arrayname="explanation_1_type_1"><item>Сопоставим населённые пункты графа и населённые пункты в таблице. Нам необходимо определить номера населённых пунктов А и G. Из В ведут пять дорог. Таким образом, В – 4. Проверим первый пункт: из первого пункта есть дорога во второй, а из второго есть путь в три пункта. Получается, что D – 2. Следовательно, номера 1, 2 и 6 не подходят. Остаются два населённых пункта – 3 и 5. Это и есть ответ. Записываем ответ в порядке возрастания без разделителей.</item><item>Сопоставим населённые пункты графа и населённые пункты в таблице. Необходимо

определить номера населенных пунктов А и Г. Из F ведут шесть дорог. Таким образом, F – 3. Заметим, что из пунктов А и Г нет дороги в населённые пункты С и Е, из которых идут 2 дороги. Следовательно, пункты 6 и 7 – это либо А, либо Г.

Заметим, что Е – единственная вершина степени 2, которая связана с вершинами третьей степени Д и К, связанными с остальными вершинами степени 2. Значит, Е соответствует П5. Далее рассмотрим два варианта. 1. Пусть Д соответствует П4, а К соответствует П7. В – единственная вершина степени 3, в которую есть дорога из Д, следовательно, В соответствует П2. Б – единственная вершина степени 3, в которую есть дорога из К, следовательно, Б соответствует П6.

2. Пусть Д соответствует П7, а К соответствует П4. В – единственная вершина степени 3, в которую есть дорога из Д, следовательно, В соответствует П6. Б – единственная вершина степени 3, в которую есть дорога из К, следовательно, Б соответствует П2. Таким образом, населённым пунктам Б и В соответствуют П2 и П6.

Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам В и С на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам Е и Г на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

Каждому населённому пункту на схеме соответствует его номер в таблице, но неизвестно, какой именно номер. Определите, какие номера населённых пунктов в таблице могут соответствовать населённым пунктам Е и F на схеме. В ответе запишите эти два номера в возрастающем порядке без пробелов и знаков препинания.

</stringarray><stringarrayname="exercise_1_type_2">

<item>На рисунке схема дорог Н-ского района изображена в виде

графа, в таблице содержатся сведения о длинах этих дорог (в километрах). \n\t\tТак как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта В в пункт Е. В ответе запишите целое число – так, как оно указано в таблице.</item>

<item>На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах) \n\t\tТак как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта Г в пункт Е. В ответе запишите целое число – так, как оно указано в таблице..</item>

<item>На рисунке справа схема дорог Н-ского района изображена в виде графа, в таблице содержатся сведения о длинах этих дорог (в километрах).\n\t\tТак как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта В в пункт Г. В ответе запишите целое число – так, как оно указано в таблице.</item>

</string-array>

<array name="exercise_1_type_2_image">

<item>@drawable/info_1_2_1</item>

<item>@drawable/info_1_2_2</item>

<item>@drawable/info_1_2_3</item>

</array>

<string-array

name="exercise_1_type_2_answer"><item>Ответ20</item><item>Ответ

40</item><item>Ответ 55</item></string-array> <string-array

name="explanation_1_type_2"><item>Пункт В – единственный пункт с пятью дорогами, значит, ему соответствует П6, а пункт Е – единственный с четырьмя дорогами, значит, ему соответствует П4. Длина дороги из П6 в П4 равна 20.</item><item>Пункт Г – единственный пункт с тремя дорогами, значит, ему соответствует П2, а пункт Е –

единственный с четырьмя дорогами, значит, ему соответствует П4. Длина дороги из П2 в П4 равна 40.

Пункт В – единственный пункт с пятью дорогами, значит, ему соответствует П6, а пункт Г – единственный с тремя дорогами, значит, ему соответствует П2. Длина дороги из П6 в П2 равна 55.

```
</string-array></resources>
```

1.5.3 Руководство пользователя

Чтобы начать работу в приложении, нужно зарегистрироваться или авторизоваться в системе. Для этого в поля «Имя пользователя» и «Пароль» введем учетные данные пользователя, после чего нажмем кнопку «Войти». Для регистрации нажмем кнопку «Зарегистрироваться».

При попытке входа с хотя бы одним незаполненным полем на экране кнопка входа будет неактивна (рисунок 14).

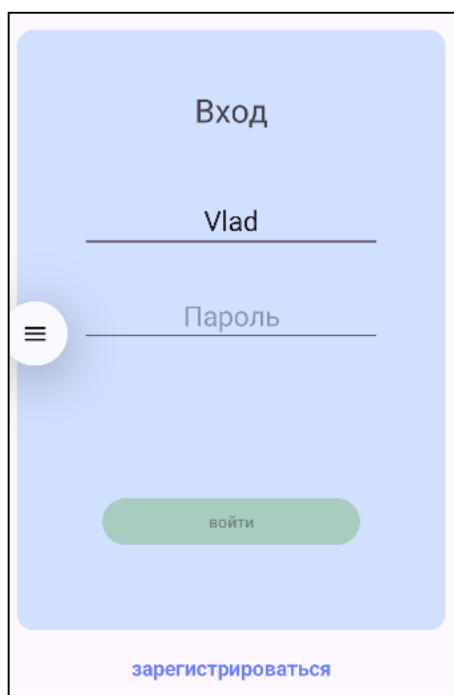


Рисунок 14 – аутентификация

При попытке регистрации нового аккаунта с хотя бы одним не заполненным полем кнопка регистрации будет неактивна (рисунок 15).

Регистрация

Vlad

Пароль

Войти

Войти

This is a mobile app registration screen. It has a light blue background with rounded corners. At the top, the title 'Регистрация' is centered. Below it are two input fields: the first contains 'Vlad' and the second is labeled 'Пароль'. To the left of the password field is a circular menu icon with three horizontal lines. Below the input fields is a green rounded button with the text 'Войти'. At the very bottom of the screen, on a light pink bar, is a blue link labeled 'Войти'.

Рисунок 15 – регистрация

При попытке регистрации нового аккаунта, который уже существует, будет выведено предупреждение (рисунок 16).

Регистрация

Vlad

123

Имя или пароль уже существуют

Войти

Войти

This is the same registration screen as in Figure 15, but with an error state. The password field now contains '123'. Below the password field, the text 'Имя или пароль уже существуют' is displayed. The green 'Войти' button has turned into a dark blue button. The bottom pink bar with the blue 'Войти' link remains the same.

Рисунок 16 – ошибка при регистрации

При попытке входа в аккаунт, который еще не существует, будет выведено предупреждение (рисунок 17).

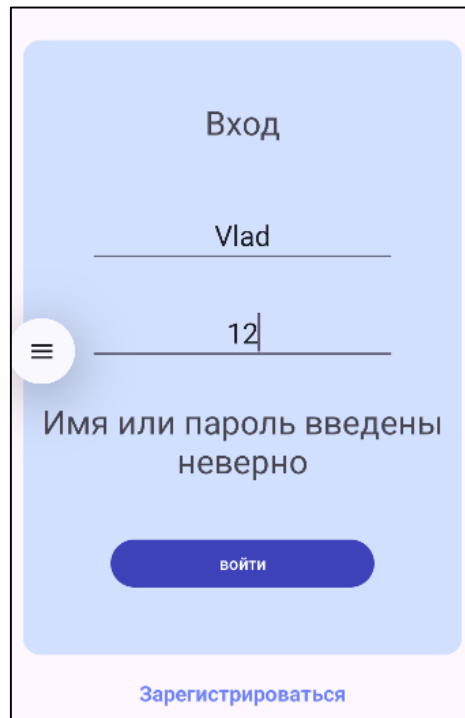


Рисунок 17 – ошибка при аутентификации

Если регистрация успешно пройдена, на экране появится соответствующее уведомление (рисунок 18).

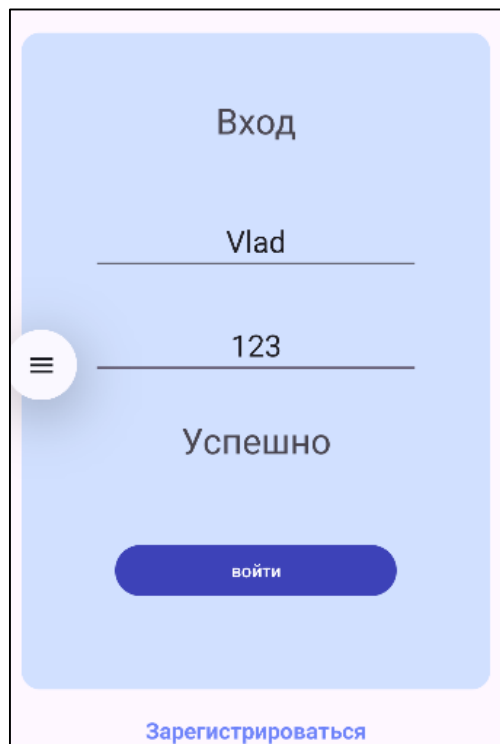


Рисунок 18 – успешный вход

После успешной авторизации откроется главное меню программы (рисунок19). Можно выбрать предмет для обучения, поменять данные профиля, чтобы это сделать необходимо нажать по той или иной кнопке действия.

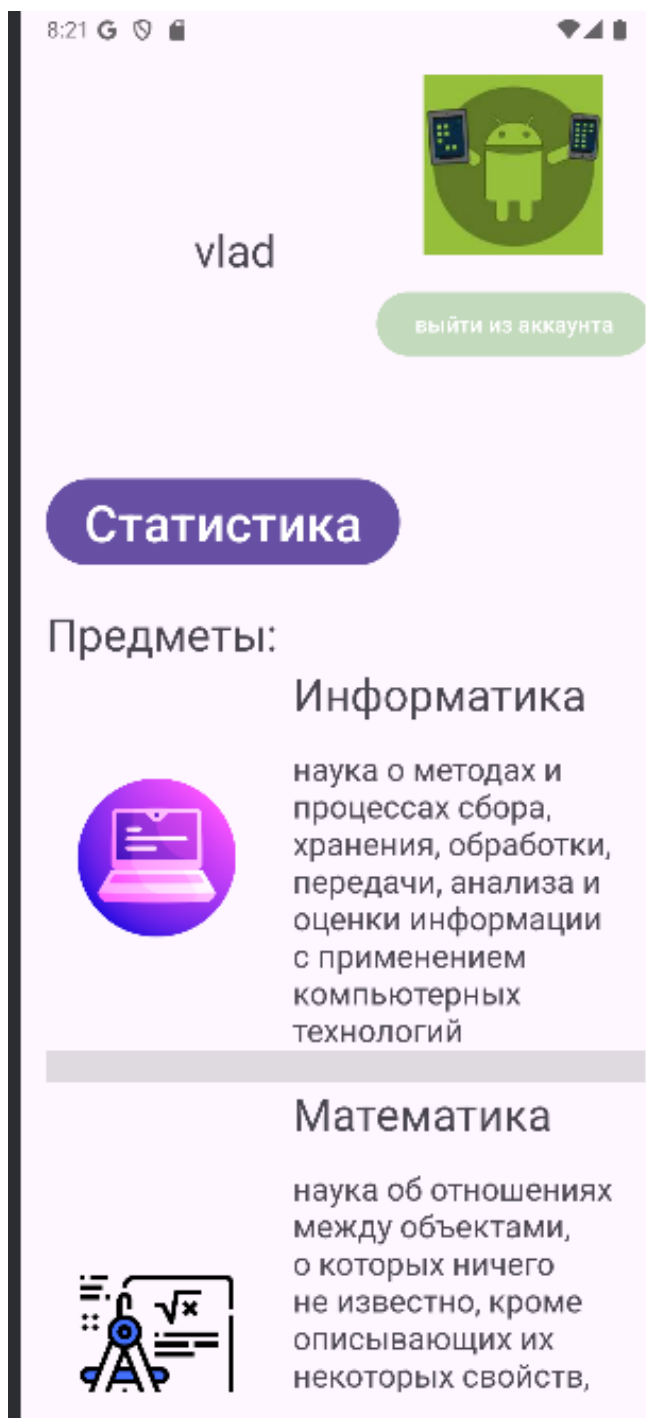


Рисунок 19 – выбор предмета

После выбора предмета пользователю доступен выбор задания или прохождение теста. Для выбора нужно нажать на желаемый курс (рисунок 20).

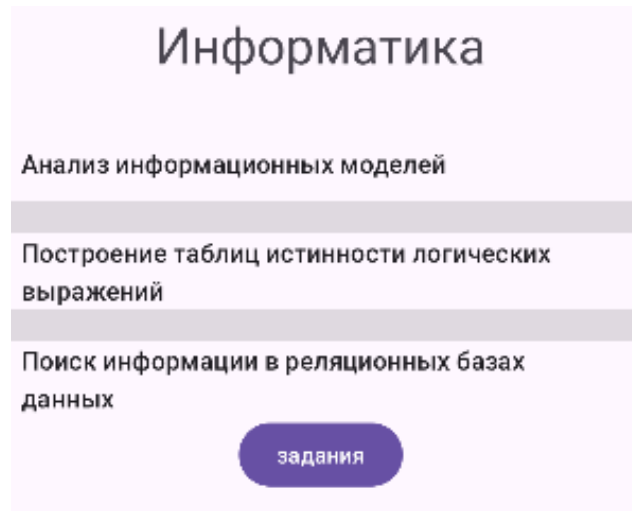


Рисунок 20 – выбор задания

После выбора тестов пользователю доступны задания для решения. Если нажать на ответ с пустым вводом появится предупреждение (рисунок 21).

буквенными обозначениями на графе. Определите номера населенных пунктов А и G в таблице. В ответе запишите числа в порядке возрастания без разделителей.

	1	2	3	4	5	6
1		*		*		
2	*			*		*
3				*	*	
4	*	*	*		*	*
5			*	*		
6		*		*		

Введите ответ

Ответить

Рисунок 21 – тестовые задания

2 Охрана труда

Работа за компьютером требует соблюдения мер безопасности для сохранения здоровья, особенно для разработчиков и других специалистов, проводящих за экраном большую часть дня. Игнорирование этих правил может привести к проблемам со зрением, нарушениям осанки, развитию туннельного синдрома и другим заболеваниям. Рассмотрим ключевые аспекты безопасной организации рабочего процесса.

2.1 Организация рабочего пространства

Правильно оборудованное рабочее место — основа профилактики профессиональных заболеваний.

Требования к помещению:

- расположение. Предпочтительны помещения с окнами на север или северо-восток;

- площадь. Не менее 4,5 м² на одно рабочее место для обеспечения свободного движения и комфорта;

- освещение. Обязательно сочетание естественного и искусственного света.

Для защиты от бликов используйте жалюзи или плотные шторы;

- вентиляция. Регулярное проветривание для снижения концентрации вредных веществ (например, формальдегида) и поддержания уровня кислорода;

- шум. Уровень не должен превышать 50 дБ — это минимизирует стресс и повышает концентрацию.

- микроклимат:

- 1) температура: 20–24°C (оптимально — 22°C);

- 2) влажность: 40–60% (максимум — 75%);

- 3) скорость движения воздуха: до 0,1 м/с.

Электробезопасность:

- заземление рабочего места для нейтрализации электромагнитного излучения от монитора и периферии;

- использование современных устройств с низким уровнем излучения.

2.2 Освещение: баланс между комфортом и продуктивностью

Неправильное освещение вызывает усталость глаз, головные боли и снижает эффективность работы.

Рекомендации:

- естественный свет: Рабочий стол располагайте так, чтобы свет из окна падал слева;

- искусственное освещение:

- 1) общая освещенность 300–500 лк;
- 2) равномерное распределение света. Избегайте бликов;
- 3) используйте лампы теплого спектра (4000–6500 К);
- 4) люминесцентные лампы должны быть оснащены рассеивателями.

2.3 Режим работы и отдыха

Регулярные перерывы и физическая активность — залог сохранения здоровья.

Физические требования:

- правильная поза: спина прямая, экран на уровне глаз, расстояние до монитора — 50–70 см;
- каждые 60–90 минут делайте перерыв на 10–15 минут: разминка для глаз, шеи и спины.

Психологические аспекты:

- используйте техники тайм-менеджмента (например, Pomodoro) для снижения стресса;
- избегайте переутомления: нормируйте рабочие часы.

Организационные меры:

- планируйте задачи так, чтобы чередовать работу за компьютером с другими видами активности;

– отдыхайте в выходные дни без использования гаджетов.

Соблюдение норм безопасности при работе за компьютером не только сохраняет здоровье, но и повышает продуктивность. Организуйте эргономичное пространство, следите за освещением, делайте перерывы и используйте специализированные программы. Помните: профилактика всегда проще, чем лечение последствий.

Заключение

В ходе выполнения данной выпускной квалификационной работы было разработано приложение, реализующие основной функционал обучения учебного материала для подготовки к ЕГЭ. Приложение включает модули регистрации и аутентификации пользователей, выбор предметов для обучения, выбор задания, прохождения тестовых заданий, анализ статистики.

Проект демонстрирует применение технологий JAVA, SQLite, XML для создания мобильного приложения. Интерфейс взаимодействия с применением REST API, который получает доступ к гугл-диску. Дизайн программы выполнен в стиле минимализм. Также была спроектирована СУБД для хранения полученных и обработанных данных.

Проект имеет значительный потенциал для дальнейшего развития, включая выполнение обучения в игровом формате, подключение ИИ для обеспечения выполнения работ с развернутым ответом, разработку системы уведомлений, расширение функционала и настроек. Дальнейшая работа также может быть направлена на повышение безопасности и оптимизацию производительности.

Список использованных источников

1. Android Studio: официальная документация [Электронный ресурс]. – Режим доступа: <https://developer.android.com/reference> (дата обращения 28.04.2025).
2. SQLite. Официальная документация [Электронный ресурс]. – Режим доступа: <https://www.sqlite.org/> (дата обращения 20.05.2025).
3. Java Официальная документация [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/en/java/> (дата обращения 25.05.2025).
4. Google API Официальная документация [Электронный ресурс]. – Режим доступа: <https://developers.google.com/explorer> (дата обращения 12.05.2025).
5. СанПиН 2.2.2/2.4.1340-03 «Гигиенические требования к персональным электронно-вычислительным машинам и организации работы».

Приложение А

(Обязательное)

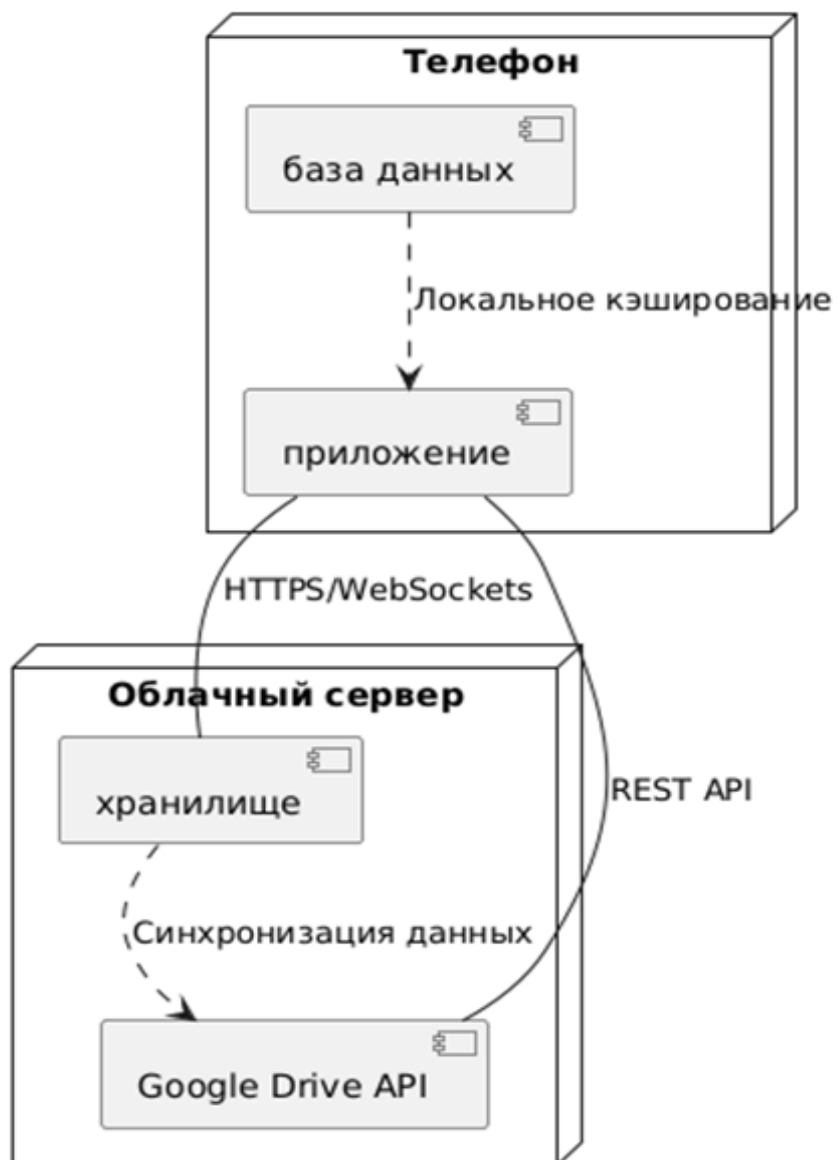
Мобильное приложение по подготовки в ЕГЭ

UML-диаграммы

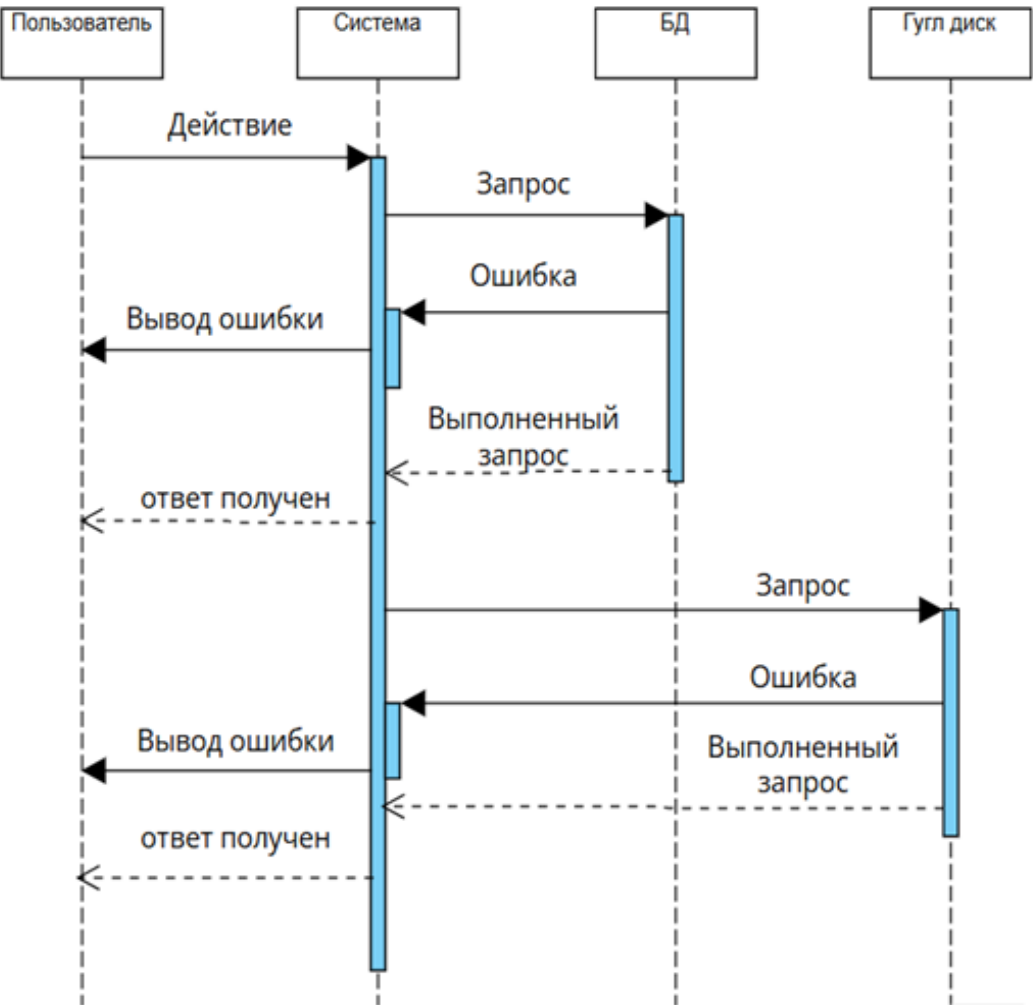
The diagrams illustrate the user flows for three different user roles in a system:

- неавторизованный пользователь (Unauthorized User):**
 - Initial actions: Регистрация (Registration), Авторизация (Authorization).
 - Post-authentication flows:
 - Выбор предмета (Subject Selection) → справочник (Reference) → тесты (Tests).
 - Выбор задания (Task Selection) → обучение (Learning) → тесты (Tests).
- авторизованный пользователь (Authorized User):**
 - Initial actions: просмотр статистики (View Statistics), настройка профиля (Profile Settings), Выход из аккаунта (Logout).
 - Post-authentication flows:
 - удаление статистики (Delete Statistics) → просмотр статистики (View Statistics).
 - изменение данных (Change Data) → настройка профиля (Profile Settings).
 - удаление аккаунта (Delete Account) → Выход из аккаунта (Logout).
- администратор (Administrator):**
 - Initial actions: кэширование видео (Cache Video), API запросы (API Requests), работа с базой данных (Database Management), обработка запросов (Request Processing).
 - Post-authentication flows:
 - получение видео (Get Video) → кэширование видео (Cache Video).
 - хранение видео (Store Video) → API запросы (API Requests).
 - хранение персональных данных (Store Personal Data) → работа с базой данных (Database Management).
 - хранение статистики (Store Statistics) → обработка запросов (Request Processing).

Приложение А.2 – Диаграмма развертывания

[illegible]

Приложение А.3 – Диаграмма последовательностей



ПУ 1.090301.28.003 Д14									
Изм.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.
Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.	Исх.
Модуль: приложение для подготовки УМК – Диаграмма последовательности									
ФБТ, п. 218ВГ2									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									
Исх.									

Приложение Б

(Обязательное)

Мобильное приложения для подготовки к ЕГЭ

Листинги программы

Файл DBManager.java

```
package com.example.thesis.DataBase;

import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class DBManager {

    private MyDataBaseHelper dbHelper;

    private Context context;

    private SQLiteDatabase database;

    public class WHERE{
        public WHERE(){
            where = " WHERE ";
        }
        private String where;

        public void add(String column, String operator, String data,
String logOperator) {

            where += column + " " + operator + " ";

            if (dbHelper.table.isText(column)) {
                where += "\"" + data + "\" ";
            } else {
                where += data + " ";
            }
            if(!logOperator.isEmpty()){
                where += logOperator + " ";
            }
        }
        public String getWHERE(){
            return where;
        }
    }

    public DBManager(Context context) {
        this.context = context;
    }
}
```

```

public String dataToString(String ... arg){
    return dbHelper.table.dataToString(arg);
}

public void open(String NameTable) throws SQLException {
    dbHelper = new MyDataBaseHelper(context, NameTable);
    database = dbHelper.getWritableDatabase();
}

public void close() {
    dbHelper.close();
}

public void insert(String column, String data) {
    String r = dbHelper.table.INSERT(column, data);
    database.execSQL(dbHelper.table.INSERT(column, data));
}

public void replace(String column, String data) {
    database.execSQL(dbHelper.table.REPLACE(column, data));
}

public void update(String data, String where) {
    database.execSQL(dbHelper.table.UDPATE(data, where));
}

public Cursor select(String data, String where) {
    return database.rawQuery(dbHelper.table.SELECT(data,
where), null);
}
}

```

Файл MyDataBaseHelper.java

```

package com.example.thesis.DataBase;

import android.content.Context;
import android.database.sqlite.*;

import java.util.Objects;

public class MyDataBaseHelper extends SQLiteOpenHelper {

    private static final String DataBaseName = "myDataBase.db"; //
название бд

```

```

        private static final int DataBaseVersion = 1; // версия базы
данных
        //public static final String TableRegName = "tableReg";          //
название таблицы
        public static final String TableVideoName = "tableVideo";      //
название таблицы

// Таблица "Регистрация"
public static class Reg {
    public static final String TableReg = "tableReg";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_PWD = "pwd";
}

// Таблица "Видео"
public static class Video {
    public static final String TableVideo = "tableVideo";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_URL = "url";
    public static final String COLUMN_PWD = "pwd";
}

// Таблица "Предмет"
public static class Subject {
    public static final String TableVideo = "tableSubject";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_NAME = "name";
    public static final String COLUMN_COUNT_TASK = "count";
    public static final String COLUMN_COMPLETE_TASK =
"complete";
}

// Таблица "Избранное"
public static class Favorites{
    public static final String TableVideo = "tableFavorite";
    public static final String COLUMN_ID = "_id";
    public static final String COLUMN_SUBJECT_ID = "subject_id";
//(FK)
    public static final String COLUMN_C_Task = "count";
    public static final String COLUMN_Complete_Task =
"complete";
}

// Таблица "Статистика"

```

```

        public static class Statistic {
            public static final String TableVideo = "tableStatistic";
            public static final String COLUMN_ID = "_id";
            public static final String COLUMN_USER_ID = "user_id";
// (FK)
            public static final String COLUMN_SUBJECT_ID = "subject_id";
// (FK)
        }

        public MyTable table;
        public MyDataBaseHelper(Context context, String nameTable) {
            super(context, DataBaseName + ".db", null, DataBaseVersion);

            if (Objects.equals(nameTable, TableRegName)) {
                table = new RegTable(TableRegName);
            }
            else {
                table = new VideoTable(TableVideoName);
            }
        }

        @Override
        public void onCreate(SQLiteDatabase db) {
            MyTable tableRegUser = new RegTable(TableRegName);
            MyTable tableVideo = new RegTable(TableVideoName);

            db.execSQL(tableRegUser.CreateTable());
            db.execSQL(tableVideo.CreateTable());
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion, int
newVersion) {
            db.execSQL("DROP TABLE IF EXISTS " + TableRegName);
            db.execSQL("DROP TABLE IF EXISTS " + TableVideoName);
            onCreate(db);
        }

        //public MyTable findTable(String tableName) throws
TypeNotPresentException {

        //            if (tableName.equals(TableRegName)) {
        //                return tableReg;

```

```

        //      }
        //      else if(tableName.equals(TableVideoName)){
        //          return tableVideo;
        //      }
        //      else{
        //          throw new TypeNotPresentException("Not Found Table
Name", null);/
        //      }
        //};
    }

```

Файл DriveServiceHelper

```

package com.example.thesis.DriveApi;

import android.content.Context;
import android.os.AsyncTask;
import android.util.Log;

import com.example.thesis.R;
import com.google.api.client.http.javanet.NetHttpTransport;
import com.google.api.client.json.gson.GsonFactory;
import com.google.api.services.drive.Drive;
import com.google.api.services.drive.DriveScopes;
import com.google.api.services.drive.model.File;
import com.google.api.services.drive.model.FileList;
import com.google.auth.http.HttpCredentialsAdapter;
import com.google.auth.oauth2.GoogleCredentials;

import java.io.IOException;
import java.io.InputStream;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.ExecutionException;

public class DriveServiceHelper {

    private static final String TAG = "DriveServiceHelper";
    private static final List<String> SCOPES =
Collections.singletonList(DriveScopes.DRIVE_READONLY);

    public interface DriveListener {
        void onFilesLoaded(List<File> files);
        void onError(Exception e);
    }

    public static void loadDriveFiles(Context context, DriveListener

```

```

listener) throws ExecutionException, InterruptedException {
    new DriveTask(context, listener).execute().get();
}

private static class DriveTask extends AsyncTask<Void, Void,
List<File>> {
    private final Context context;
    private final DriveListener listener;
    private Exception exception;

    DriveTask(Context context, DriveListener listener) {
        this.context = context;
        this.listener = listener;
    }

    @Override
    protected List<File> doInBackground(Void... voids) {
        try {
            // 1. Загружаем учетные данные
            InputStream keyStream =
context.getResources().openRawResource(R.raw.google_drive_key);
            GoogleCredentials credentials =
GoogleCredentials.fromStream(keyStream)
                .createScoped(SCOPES);

            // 2. Создаем сервис Drive
            Drive drive = new Drive.Builder(
                new NetHttpTransport(),
                GsonFactory.getDefaultInstance(),
                new HttpCredentialsAdapter(credentials))
                .setApplicationName("MyVideo")
                .build();

            // 3. Получаем список файлов
            FileList result = drive.files().list()
                .setQ("mimeType = 'video/mp4'") // ищем
только mp4
                .setFields("files(id, name, webViewLink,
webContentLink)")
                .execute();

            return result.GetFiles();
        } catch (IOException e) {
            Log.e(TAG, "Error loading files", e);
            exception = e;
            return null;
        }
    }
}

```

```

        }
    }

    @Override
    protected void onPostExecute(List<File> files) {
        if (exception != null) {
            listener.onError(exception);
        } else {
            listener.onFilesLoaded(files);
        }
    }
}
}

```

Файл LoginActivity.java

```

package com.example.thesis.Login;

import android.content.Intent;
import android.database.Cursor;
import android.os.Bundle;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.thesis.DataBase.DBManager;
import com.example.thesis.DataBase.MyDataBaseHelper;
import com.example.thesis.DataBase.RegTable;
import com.example.thesis.Profile.ProfileActivity;

import com.example.thesis.R;

public class LogInActivity extends AppCompatActivity {

    public EditText editName, editPwd;
    private Button button;
    private TextView err, signInUp, clickableText;
    public DBManager dbManager;

```

```

private boolean isReg = false; // 0 - вход | 1 - регистрация

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable(this);
    setContentView(R.layout.activity_log_in_up);

    dbManager = new DBManager(this);
    dbManager.open(MyDataBaseHelper.TableRegName);

    editName = (EditText) findViewById(R.id.editName);
    editPwd = (EditText) findViewById(R.id.editPwd);
    err = (TextView) findViewById(R.id.textErr);
    signInUp = (TextView) findViewById(R.id.textInUp);
    clickableText = (TextView)
findViewById(R.id.clickable_text);
    button = (Button) findViewById(R.id.button);

    editName.addTextChangedListener(textWatcher);
    editPwd.addTextChangedListener(textWatcher);

    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
        Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
        return insets;
    });
}

    TextWatcher textWatcher = new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence s, int start, int
count, int after) {}

        @Override
        public void onTextChanged(CharSequence s, int start, int

```

```

before, int count) {
    check();
}

@Override
public void afterTextChanged(Editable s) {}
};

// Вкл/выкл возможность нажать на кнопку
private void check() {
    String username = editName.getText().toString();
    String password = editPwd.getText().toString();

    if (!username.isEmpty() && !password.isEmpty()) {
        button.setEnabled(true);

button.setBackgroundTintList(getResources().getColorStateList(R.color.buttonReady));
        } else {
            button.setEnabled(false);

button.setBackgroundTintList(getResources().getColorStateList(R.color.buttonNotReady));
        }
    }

// вход/регистрация
public void Change(View view) {

    isReg = !isReg;

    if (isReg) {
        signInUp.setText("Регистрация");
        clickableText.setText("Авторизация");
    }
    else {
        signInUp.setText("Авторизация");
        clickableText.setText("Зарегистрироваться");
    }

    editName.setText("");
    editPwd.setText("");
    err.setText("");
}

public void SignInUp(View view) {

```

```

String name = editName.getText().toString();
String pwd = editPwd.getText().toString();

if(name.isEmpty() || pwd.isEmpty()){
    err.setText("Поле не может быть пустым");
    return;
}
if (!(name.matches("[a-zA-Z]+")) || (!pwd.matches("[a-zA-Z0-9]+"))){
    err.setText("Нельзя ввести такое имя");
    return;
}

String logInUp;
if(isReg){
    logInUp = "OR";
}
else{
    logInUp = "AND";
}

DBManager.WHERE where = dbManager.new WHERE();

if(isReg) {
    where.add(RegTable.Cname, "=", name, "");
    Cursor curs =
dbManager.select(dbManager.dataToString(RegTable.ID, RegTable.Cname,
RegTable.Cpwd), where.getWHERE());
    curs.moveToFirst();
    if (curs.moveToFirst()) {
        err.setText("Имя уже существует");
        return;
    } else {
        err.setText("Успешно");
    }

dbManager.insert(dbManager.dataToString(RegTable.Cname,
RegTable.Cpwd),
                dbManager.dataToString(name, pwd));

    Intent intent = new Intent(this,
ProfileActivity.class);
    intent.putExtra("USERNAME", name);
    startActivity(intent);
}

```

```

    }
    else{
        where.add(RegTable.Cname, "=", name, logInUp);
        where.add(RegTable.Cpwd, "=", pwd, "");
        Cursor curs =
dbManager.select(dbManager.dataToString(RegTable.ID, RegTable.Cname,
RegTable.Cpwd), where.getWHERE());
        curs.moveToFirst();
        if (curs.moveToFirst()) {
            err.setText("Успешно");

            Intent intent = new Intent(this,
ProfileActivity.class);
            intent.putExtra("USERNAME", name);
            startActivity(intent);
        } else {
            err.setText("Имя или пароль введены неверно");
        }
    }
}
}

```

Файл ProfileActivity.java

```

package com.example.thesis.Profile;

import android.content.Intent;
import android.content.res.TypedArray;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.thesis.DriveApi.DriveServiceHelper;
import com.example.thesis.DriveApi.VideoStorage;
import com.example.thesis.Login.LogInActivity;

```

```

import com.example.thesis.R;
import com.example.thesis.SubjectTeaching.LessonsActivity;
import com.google.api.services.drive.model.File;

import java.util.ArrayList;
import java.util.List;
import java.util.concurrent.ExecutionException;

public class ProfileActivity extends AppCompatActivity implements
DriveServiceHelper.DriveListener {

    VideoStorage storage = new VideoStorage("info");
    ArrayList<Subject> subjects = new ArrayList<Subject>();
    ListView subjectsList;
    Intent intent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_profile);

        try {
            DriveServiceHelper.loadDriveFiles(this, this);
        } catch (ExecutionException e) {
            throw new RuntimeException(e);
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }

        intent = new Intent(this, LessonsActivity.class);
        TextView userName = findViewById(R.id.UserName);
        String Name = getIntent().getStringExtra("USERNAME");

        // Отображение данных
        if (Name != null) {
            userName.setText(Name);
        }

        setInitialData();
        subjectsList = findViewById(R.id.subjectsList);
        SubjectAdapter subjectAdapter = new SubjectAdapter(this,
R.layout.subject_item, subjects);
        subjectsList.setAdapter(subjectAdapter);
    }
}

```

```

        AdapterView.OnItemClickListener itemListener = new
AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View v,
int position, long id) {

        // получаем выбранный пункт
        Subject selectedState =
        (Subject)parent.getItemAtPosition(position);
        Toast.makeText(getApplicationContext(), "Был выбран
предмет " + selectedState.getName(),
            Toast.LENGTH_SHORT).show();

        intent.putExtra("SUBJECT", position);
        intent.putExtra(VideoStorage.class.getName(),
storage);

        startActivity(intent);

        //intent.putExtra(VideoStorage.getNameForIntent(),
storage.subjectAt(position));
        //startActivity(intent);
    }
};
subjectsList.setOnItemClickListener(itemListener);

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
    Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
    return insets;
});
}

public void SignOut(View view){
    Intent loginIntent = new Intent(this, LogInActivity.class);
    // Очищаем весь стек Activity
    loginIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK |
Intent.FLAG_ACTIVITY_CLEAR_TASK);

    startActivity(loginIntent);
    finish(); // Закрываем текущую Activity
}

```

```

private void setInitialData(){

    TypedArray subjects_image =
    getResources().obtainTypedArray(R.array.subjects_image);
    String[] name =
    getResources().getStringArray(R.array.subjects_name);
    String[] description =
    getResources().getStringArray(R.array.subjects_description);

    for (int i = 0; i < subjects_image.length(); i++) {
        int resId = subjects_image.getResourceId(i, -1);
        if (resId != -1) {
            subjects.add(new Subject (name[i], description[i],
resId));
        }
    }
}

@Override
public void onFilesLoaded(List<File> files) {
    runOnUiThread() -> {
        storage.SetVideo(this, files);
    });
}

@Override
public void onError(Exception e) {
    runOnUiThread() -> {
        Toast.makeText(this, "Error: " + e.getMessage(),
Toast.LENGTH_SHORT).show();
    });
}
}

```

Файл Subject.java

```

package com.example.thesis.Profile;

//школьный предмет
public class Subject {

    private String name; // название предмета
    private String description; // описание предмета
    private int imageResource; // картинка предмета

```

```

    public Subject(String name, String description, int
imageResource) {
        this.name=name;
        this.description=description;
        this.imageResource=imageResource;
    }

    public String getName() {
        return this.name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getDescription() {
        return this.description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public int getImageResource() {
        return this.imageResource;
    }

    public void setImageResource(int imageResource) {
        this.imageResource = imageResource;
    }
}

```

Файл SubjectAdapter

```

package com.example.thesis.Profile;

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;

```

```

import com.example.thesis.R;

import java.util.List;

public class SubjectAdapter extends ArrayAdapter<Subject> {

    private final List<Subject> subjects;
    private final int layout;
    private final LayoutInflater inflater;

    public SubjectAdapter(Context context, int resource,
List<Subject> subjects) {
        super(context, resource, subjects);
        this.subjects = subjects;
        this.layout = resource;
        this.inflater = LayoutInflater.from(context);
    }

    public View getView(int position, View convertView, ViewGroup
parent) {

        View view=inflater.inflate(this.layout, parent, false);

        TextView nameView = view.findViewById(R.id.name_subject);
        ImageView imageView = view.findViewById(R.id.image_subject);
        TextView descriptionView =
view.findViewById(R.id.description_subject);

        Subject subject = subjects.get(position);

        imageView.setImageResource(subject.getImageResource());
        nameView.setText(subject.getName());
        descriptionView.setText(subject.getDescription());

        return view;
    }
}

```

Файл Exercise.java

```

package com.example.thesis.SubjectTeaching.Exercise;

public class Exercise {

    private String condition;

```

```

private int image;
private String answer;
private String correctAnswer;
private String explanation;
public boolean isExpanded;

    public Exercise(String condition, int image, String
correctAnswer, String explanation) {
        this.condition = condition;
        this.image = image;
        this.answer = "";
        this.correctAnswer = correctAnswer;
        this.explanation = explanation;
        isExpanded = false;
    }

    public String getCondition() {
        return condition;
    }
    public int getImage() {
        return image;
    }

    public void setAnswer(String answer){
        this.answer = answer;
    }
    public String getAnswer() {
        return answer;
    }
    public String getCorrectAnswer() {
        return correctAnswer;
    }
    public String getExplanation() {
        return explanation;
    }

    public class ExerciseInfo{
        public String subject; //название предмета
        public String name; //название задания
        public String subName; //название типа задания
        public int number; //номер задания
        public boolean isComplete; //задание решено

        ExerciseInfo(String subject, String name, String subName,
int number, boolean isComplete){

```

```

        this.subject = subject;
        this.name = name;
        this.subName = subName;
        this.number = number;
        this.isComplete = isComplete;
    }
}

```

Файл ExerciseActivity.java

```

package com.example.thesis.SubjectTeaching.Exercise;

import android.content.res.TypedArray;
import android.os.Bundle;
import android.widget.ListView;

import androidx.activity.EdgeToEdge;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.thesis.R;
import com.example.thesis.SubjectTeaching.Task.Task;
import com.example.thesis.SubjectTeaching.Task.TaskAdapter;

import java.util.ArrayList;
import java.util.List;

public class ExerciseActivity extends AppCompatActivity {

    ArrayList<Exercise> exercises = new ArrayList<Exercise>();
    ListView exercisesList;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_exercise);

        exercisesList = findViewById(R.id.exerciseList);
    }
}

```

```

        ExerciseAdapter taskAdapter = new ExerciseAdapter(this,
exercises, true);
        // устанавливаем адаптер
        exercisesList.setAdapter(taskAdapter);

        determineExercise();

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
            return insets;
        });
    }

    void determineExercise() {

        TypedArray imageExample;
        String[] condition = new String[]{"");
        String[] correctAnswer = new String[]{"");
        String[] explanation = new String[]{"");
        List<Exercise> exerciseItems = new ArrayList<>();

        //1 задание
        //1 тип задания
        condition =
getResources().getStringArray(R.array.exercise_1_type_1);
        imageExample =
getResources().obtainTypedArray(R.array.exercise_1_type_1_image);
        correctAnswer =
getResources().getStringArray(R.array.exercise_1_type_1_answer);
        explanation =
getResources().getStringArray(R.array.explanation_1_type_1);

        for(int i = 0; i < condition.length; i++){
            exercises.add(new Exercise(condition[i],
imageExample.getResourceId(i, -1), correctAnswer[i],
explanation[i]));
        }

        condition =

```

```

getResources().getStringArray(R.array.exercise_1_type_2);
    imageExample =
getResources().obtainTypedArray(R.array.exercise_1_type_2_image);
    correctAnswer =
getResources().getStringArray(R.array.exercise_1_type_2_answer);
    explanation =
getResources().getStringArray(R.array.explanation_1_type_2);
    for(int i = 0; i < condition.length; i++){
        exercises.add(new Exercise(condition[i],
imageExample.getResourceId(i, -1), correctAnswer[i],
explanation[i]));
    }
}

void exerciseInfo(){

}

}

```

Файл ExerciseAdapter.java

```

package com.example.thesis.SubjectTeaching.Exercise;

import android.content.Context;
import android.os.Build;
import android.text.Editable;
import android.text.TextWatcher;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.TextView;

import androidx.annotation.NonNull;

import com.example.thesis.R;

import java.util.List;

```

```

public class ExerciseAdapter extends ArrayAdapter<Exercise> {

    private final List<Exercise> exercises;
    private final Context context;

    private final boolean state; //state: 0 - example | 1 - exercise

    public ExerciseAdapter(Context context, List<Exercise>
exercises, boolean state) {
        super(context, R.layout.exercise_item, exercises);
        this.exercises = exercises;
        this.context = context;
        this.state = state;
    }

    @NonNull
    public View getView(int position, View convertView, ViewGroup
parent) {
        ViewHolder holder;

        if (convertView == null) {
            LayoutInflater inflater = LayoutInflater.from(context);
            convertView = inflater.inflate(R.layout.exercise_item,
parent, false);

            holder = new ViewHolder(convertView, state);

            convertView.setTag(holder);
        } else {
            holder = (ViewHolder) convertView.getTag();
        }

        display(holder, position);

        return convertView;
    }

    public void display(ViewHolder holder, int position){
        Exercise exercise = exercises.get(position);

        holder.condition.setText(exercise.getCondition());
        holder.image.setImageResource(exercise.getImage());

        if(state) {

```

```

        holder.answerContainer.setVisibility(exercise.isExpanded
? View.VISIBLE : View.GONE);
        holder.answer.setText(exercise.getAnswer());

holder.correctAnswer.setText(exercise.getCorrectAnswer());
holder.explanation.setText(exercise.getExplanation());

holder.answer.addTextChangedListener(new TextWatcher() {

    @Override
    public void onTextChanged(CharSequence s, int start,
int before, int count) {
        if(count != 0) {
            int t = position;
            exercise.setAnswer(s.toString());
        }
    }
    @Override
    public void beforeTextChanged(CharSequence s, int
start, int count, int after) {}
    @Override
    public void afterTextChanged(Editable s) {}
});

holder.btn.setOnClickListener(v -> {
    if (Build.VERSION.SDK_INT >=
Build.VERSION_CODES.VANILLA_ICE_CREAM) {
        if(!holder.answer.getText().isEmpty()) {
            holder.solution.setVisibility(View.VISIBLE);
        }
        else{
            holder.answer.setHint("Введите ответ");
        }
    }
});

holder.solution.setOnClickListener(v -> {
    exercise.isExpanded = !exercise.isExpanded;
    holder.solution.setVisibility(View.VISIBLE);
    notifyDataSetChanged(); // Обновляем список
});

}

}

```

```

public static class ViewHolder {

    TextView solution;
    TextView condition;
    ImageView image;
    EditText answer;
    Button btn;
    View answerContainer;
    TextView correctAnswer;
    TextView explanation;

    public ViewHolder(View view, boolean state){
        //основные виджеты задания
        condition = view.findViewById(R.id.exercise_condition);
        image = view.findViewById(R.id.exercise_image);

        //дополнительные
        solution = view.findViewById(R.id.solution);
        answer = view.findViewById(R.id.exercise_answer);
        btn = view.findViewById(R.id.exercise_btn);
        answerContainer =
view.findViewById(R.id.exercise_layoutAnswer);
        correctAnswer =
view.findViewById(R.id.exercise_correctAnswer);
        explanation =
view.findViewById(R.id.exercise_explanations);
        if(!state) {
            solution.setVisibility(View.GONE);
            answer.setVisibility(View.GONE);
            btn.setVisibility(View.GONE);
            answerContainer.setVisibility(View.GONE);
            correctAnswer.setVisibility(View.GONE);
            explanation.setVisibility(View.GONE);
        }
    }
}

```

Файл Statistic.java

```

package com.example.thesis.SubjectTeaching.Exercise;

import java.util.ArrayList;

```

```

public class Statistic {

    String nameUser;    //имя пользователя
    String[] subjects; //название предмета
    String[] nameExercise; //название задания
    String[] nameTypeExercise; //название типа задания
    ArrayList<ArrayList<Integer>> countExercise; //количество
заданий
    ArrayList<ArrayList<Integer>> countCompleteExercise;
//количество выполненных заданий
    ArrayList<ArrayList<Integer>> idExercise;

}

```

Файл Task.java

```

package com.example.thesis.SubjectTeaching.Task;

import com.example.thesis.SubjectTeaching.Exercise.Exercise;

public class Task {

    private String name; // название задания
    private String description; // описание задания
    private Exercise exercise; // задание
    public Task(String name, String description, Exercise exercise){
        this.name=name;
        this.description=description;
        this.exercise = exercise;
    }

    public String getName() {
        return this.name;
    }
    public String getDescription() {return this.description;}
    public Exercise getExercise(){ return this.exercise;}
}

```

Файл TaskActivity.java

```

package com.example.thesis.SubjectTeaching.Task;

import android.content.res.TypedArray;
import android.os.Bundle;

```

```

import android.util.DisplayMetrics;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.widget.AdapterView;
import android.widget.ListView;
import android.widget.Toast;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import com.example.thesis.DriveApi.VideoStorage;
import com.example.thesis.R;
import com.example.thesis.SubjectTeaching.Exercise.Exercise;

import java.util.ArrayList;
import java.util.List;

public class TaskActivity extends AppCompatActivity {

    ArrayList<Task> tasks = new ArrayList<Task>();
    ListView tasksList;
    WebView webView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_task);

        webView = findViewById(R.id.webView3);
        webView.getSettings().setJavaScriptEnabled(true);
        WebSettings webSettings = webView.getSettings();
        webSettings.setMediaPlaybackRequiresUserGesture(true); //
Автопроизведение
        webSettings.setAllowFileAccess(false); // Доступ к локальным
файлам
        webSettings.setAllowContentAccess(false); // Доступ к
контенту

        DisplayMetrics displayMetrics = new DisplayMetrics();

```

```

getWindowManager().getDefaultDisplay().getMetrics(displayMetrics);
    int screenWidth = displayMetrics.widthPixels;
    int screenHeight = displayMetrics.heightPixels;

    webView.getLayoutParams().width = (int) (screenWidth * 9);
    webView.getLayoutParams().height = (int) (screenHeight *
0.4);
    webView.requestLayout();

    determineTask();
    tasksList = findViewById(R.id.TaskList);
    TaskAdapter taskAdapter = new TaskAdapter(this,
R.layout.task_item, tasks);
    // устанавливаем адаптер
    tasksList.setAdapter(taskAdapter);

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
    Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
    return insets;
});
}

private void determineTask(){

    Bundle arguments = getIntent().getExtras();

    String[] name = new String[]{" "};
    String[] description = new String[]{" "};
    List<Exercise> exerciseList = new ArrayList<>();
    int numberTask = arguments.getInt("TASK");
    if (numberTask == 0) {
        name =
getResources().getStringArray(R.array.task_1_names);
        description =
getResources().getStringArray(R.array.task_1_explanations);
    }
    else if(numberTask == 1){
        name =
getResources().getStringArray(R.array.task_2_names);
        description =

```

```

getResources().getStringArray(R.array.task_2_explanations);
    }
    else if(numberTask == 2){
        name =
getResources().getStringArray(R.array.task_3_names);
        description =
getResources().getStringArray(R.array.task_3_explanations);
    }

    TypedArray imageExample;
    String[] condition = new String[]{" "};

    //1 тип задания
    condition =
getResources().getStringArray(R.array.exercise_1_type_1);
    imageExample =
getResources().obtainTypedArray(R.array.exercise_1_type_1_image);
    exerciseList.add(new Exercise(condition[0],
imageExample.getResourceId(0, -1), null, null));

    //2 тип задания
    condition =
getResources().getStringArray(R.array.exercise_1_type_2);
    imageExample =
getResources().obtainTypedArray(R.array.exercise_1_type_2_image);
    exerciseList.add(new Exercise(condition[0],
imageExample.getResourceId(0, -1), null, null));

    VideoStorage.VideoInfo info = null;
    info = (VideoStorage.VideoInfo)
arguments.getSerializable(VideoStorage.getNameForIntent());
    webView.loadUrl(info.getUrlView());

    for (int i = 0; i < name.length; i++) {
        tasks.add(new Task (name[i], description[i],
exerciseList.get(i)));
    }
}
}

```

Файл TaskAdapter.java

```

package com.example.thesis.SubjectTeaching.Task;

```

```

import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;

import com.example.thesis.R;
import com.example.thesis.SubjectTeaching.Exercise.Exercise;
import com.example.thesis.SubjectTeaching.Exercise.ExerciseAdapter;

import java.util.ArrayList;
import java.util.List;

public class TaskAdapter extends ArrayAdapter<Task> {

    private final Context context;
    private final List<Task> tasks;
    private final int layout;
    private final LayoutInflater inflater;
    ExerciseAdapter exerciseAdapter;

    public TaskAdapter(Context context, int resource, List<Task>
subjects) {
        super(context, resource, subjects);
        this.context = context;
        this.tasks = subjects;
        this.layout = resource;
        this.inflater = LayoutInflater.from(context);

        List<Exercise> listExercise = new ArrayList<>();
        for(int i = 0; i < subjects.size(); i++){
            listExercise.add(subjects.get(i).getExercise());
        }
        this.exerciseAdapter = new ExerciseAdapter(context,
listExercise, false);
    }

    public View getView(int position, View convertView, ViewGroup
parent) {
        ViewHolder holder;
        ExerciseAdapter.ViewHolder holderExercise;

        if (convertView == null) {
            LayoutInflater inflater = LayoutInflater.from(context);

```

```

        convertView = inflater.inflate(R.layout.task_item,
parent, false);
        holder = new ViewHolder(convertView);
        holderExercise = new
ExerciseAdapter.ViewHolder(convertView, false);
        convertView.setTag(R.id.id1, holder);
        convertView.setTag(R.id.id2, holderExercise);
    } else {
        holder = (ViewHolder) convertView.getTag(R.id.id1);
        holderExercise = (ExerciseAdapter.ViewHolder)
convertView.getTag(R.id.id2);
    }

    Task task = tasks.get(position);
    holder.nameView.setText(task.getName());
    holder.descriptionView.setText(task.getDescription());

    exerciseAdapter.display(holderExercise, position);

    return convertView;
}

static class ViewHolder {
    TextView nameView;
    TextView descriptionView;

    ViewHolder(View view){
        nameView = view.findViewById(R.id.name_task);
        descriptionView =
view.findViewById(R.id.description_task);
    }
}
}

```

Файл LessonsActivity.java

```

package com.example.thesis.SubjectTeaching;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView;
import android.widget.ListView;

import androidx.activity.EdgeToEdge;

```

```

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

import android.widget.ArrayAdapter;

import com.example.thesis.DriveApi.VideoStorage;
import com.example.thesis.Profile.ProfileActivity;
import com.example.thesis.R;
import com.example.thesis.SubjectTeaching.Exercise.ExerciseActivity;
import com.example.thesis.SubjectTeaching.Task.TaskActivity;

public class LessonsActivity extends AppCompatActivity {

    private String[] nameTask;
    VideoStorage videoStorage;
    Intent intent;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_lessons);

        intent = new Intent(this, TaskActivity.class);

        determineSubject();

        ListView tasksList = findViewById(R.id.tasksList);
        // создаем адаптер
        ArrayAdapter<String> adapter = new ArrayAdapter(this,
            android.R.layout.simple_list_item_1, nameTask);
        // устанавливаем для списка адаптер
        tasksList.setAdapter(adapter);
        // добавляем для списка слушатель
        tasksList.setOnItemClickListener(new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View v,
int position, long id)
            {
                intent.putExtra("TASK", position);

                intent.putExtra(VideoStorage.VideoInfo.class.getName(),
videoStorage.subjectAt(position));
            }
        });
    }
}

```

```

        startActivity(intent);
    }
});

ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main),
(v, insets) -> {
    Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
    v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom);
    return insets;
});
}

public void onExercises(View view) {
    Intent intent = new Intent(this, ExerciseActivity.class);
    startActivity(intent);
}
private void determineSubject(){

    Bundle arguments = getIntent().getExtras();
    if(arguments!=null) {
        videoStorage = (VideoStorage)
arguments.getSerializable(VideoStorage.class.getName());
    }

    int numberSubject = arguments.getInt("SUBJECT");
    if (numberSubject == 0) {
        nameTask =
getResources().getStringArray(R.array.task_name_1);
    }
    else if(numberSubject == 1){
        nameTask =
getResources().getStringArray(R.array.task_name_2);
    }
    else if(numberSubject == 2){
        nameTask =
getResources().getStringArray(R.array.task_name_3);
    }
}
}
}

```

Приложение В

(Обязательное)

Мобильное приложения для подготовки к ЕГЭ

Презентация

Мобильное приложение по подготовки к ЕГЭ

Автор работы:

Студент гр. 21ВВП2 Головинов Владислав Русланович

Научный руководитель:

к.т.н. доцент каф. ВТ Егоров Валерий Юрьевич

Цель работы и актуальность разработки

Цель работы – разработать мобильное
приложение по подготовке к ЕГЭ

- Удобный и понятный материал для обучения
- Возможность пройти тесты
- Возможность проанализировать свою статистику

Анализ существующих аналогов

• РЕШУ ЕГЭ

• Фоксфорд

• Незнайка

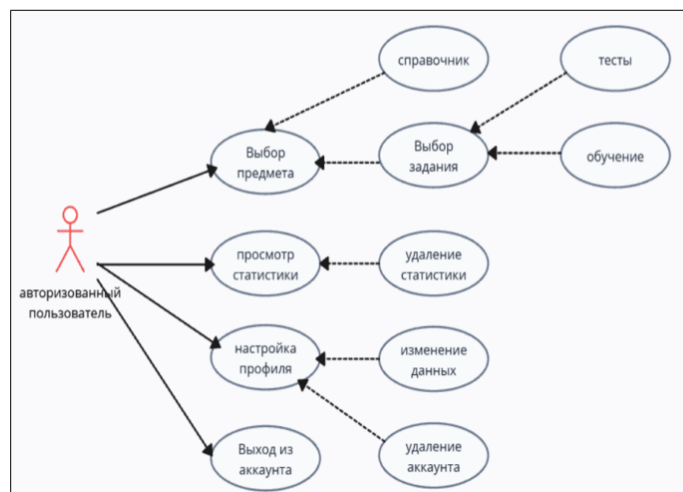


Фоксфорд
Онлайн-школа



Возможности приложения

- авторизация
- регистрация
- просмотр обучающих роликов
- прохождение тестов



Выбор технологий и средств разработки приложения

- Android Studio
- Java
- XML
- SQLite



Спасибо за внимание!