

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

Ботагузов Альтаир Адилханович

Разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения.

ДИПЛОМНАЯ РАБОТА

Специальность 5В070300 – Информационные системы

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой
КБОиХИ

канд. техн. наук, доцент

_____ Н. А. Сейлова

« _____ » _____ 20__ г.

ДИПЛОМНАЯ РАБОТА

На тему: Разработка информационных образовательных систем на основе
игровых моделей и методов машинного обучения.

Специальность 5В070300 – Информационные системы

Выполнил:

Ботагузов А.А

Научный руководитель

к.т.н, доцент

_____ Косников В.А.

« _____ » _____ 20__ г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Satbayev University

Институт информационных и телекоммуникационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – Информационные системы

УТВЕРЖДАЮ

Заведующий кафедрой
КБОиХО

канд. техн. наук, доцент

_____ Н. А. Сейлова

« _____ » _____ 20__ г.

ЗАДАНИЕ

на выполнение дипломной работы

Обучающемуся: Ботагузов Альтаир Адилханович

Тема: Разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения.

Утвержден приказом Ректора Университета №762-б от 27.01.2020г.

Срок сдачи законченной работы 27.05.2020г.

Исходные данные к дипломному проекту: результаты преддипломной практики, результат обзора современного состояния по данной теме, сбор теоретического материала.

Краткое содержание дипломной работы:

- а) Современное состояние разработки информационных систем;
- б) Структура и алгоритмы функционирования системы;
- в) Информационное и программное обеспечение;
- г) Внедрение системы.

Рекомендуемая основная литература: *из 17 наименований*

ГРАФИК

подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Современное состояние разработки информационных систем	20.02.2020г.	
Структура и алгоритмы функционирования системы	01.04.2020г.	
Информационное и программное обеспечение	15.05.2020г.	
Реализация системы	27.05.2020г.	

Подписи

консультантов и нормоконтролера на законченную дипломную работу (проект) с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, Ф.И.О. (уч.степень, звание)	Дата подписания	Подпись
Разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения.	Косников В.А., кандидат технических наук, доцент		
Нормоконтролер	Бауыржан М.Б., тьютор		
Программная часть	Кабдуллин М.А., ассистент		

Научный руководитель: _____ Косников В.А.

Задание принял к исполнению обучающийся: _____ Ботагузов А.А

Дата

"27" январь 2020 г.

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Ботагузов А. А

Название: Разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения

Координатор: Вячеслав Косников

Коэффициент подобия 1: 9,7

Коэффициент подобия 2: 5,7

Замена букв: 95

Интервалы: 0

Микропробелы: 0

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....

.....
Дата

.....
Подпись Научного руководителя

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Ботагузов А. А

Название: Разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения

Координатор: Вячеслав Косников

Коэффициент подобия 1:9,7

Коэффициент подобия 2:5,7

Замена букв:95

Интервалы:0

Микропробелы:0

Белые знаки:0

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

.....
.....
.....
.....
.....

.....

Дата

.....

Подпись заведующего кафедрой /

начальника структурного подразделения

Окончательное решение в отношении допуска к защите, включая обоснование:

.....
.....
.....
.....
.....

Дата

.....
*Подпись заведующего кафедрой /
начальника структурного подразделения*

НАЗВАНИЕ:

Разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения

АВТОР:

Ботагузов А. А

НАУЧНЫЙ РУКОВОДИТЕЛЬ:

Вячеслав Косников

ПОДРАЗДЕЛЕНИЕ:

ИКИИТ

ДАТА ЗАГРУЗКИ ДОКУМЕНТА:

2020-06-03 13:06:26

ЧИСЛО ПРОВЕРЕК ДОКУМЕНТА: ⓘ

1

ПРОПУЩЕННЫЕ ВЕБ-СТРАНИЦЫ: ⓘ

■ Уровень заимствований

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



■ Предупреждение и сигналы тревоги

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв	95	показать в тексте
<small>Использование символов из другого алфавита - может указывать на способ обойти систему, поэтому следует установить их использование.</small>		
Интервалы	0	показать в тексте
<small>Количество увеличенного расстояния между буквами (просим определить является ли расстояние интацией пробела, так как исконно слова могут быть написаны слитно).</small>		
Микропробелы	0	показать в тексте
<small>Количество пробелов с нулевым размером - необходимо проверить влияют ли они на неправильное разделение слов в тексте.</small>		
Белые знаки	0	показать в тексте
<small>Количество символов, выделенных белым цветом, пожалуйста, проверьте не используются ли белые символы вместо пробела, соединяя слова (в отчете подобия система изменяет автоматически цвет букв в-коричневый, чтобы их сделать видимыми).</small>		

■ Заимствования по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и посмотрите, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз (7,35 %)

Десять самых длинных фрагментов найденных во всех доступных ресурсах.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	АВТОР	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ	
1	http://portal.tpu.ru:7777/eL/img/Tab1/Kahoot.pdf		78	1,48 %
2	https://otvetus.com/chto-takoe-windows-10-266118		62	1,18 %
3	Адаптивный образовательный онлайн-процесс на основе мобильного приложения для SMART – университета (на основе управления сложностью и весом вопросов) <i>International IT University (Информационные системы)</i>	Мадияр Сейтжанов	61	1,16 %
4	Адаптивный образовательный онлайн-процесс на основе мобильного приложения для SMART – университета (на основе управления сложностью и весом вопросов) <i>International IT University (Информационные системы)</i>	Мадияр Сейтжанов	39	0,74 %
5	https://ru.qwe.wiki/wiki/Microsoft_Word		34	0,65 %

6	http://portal.tpu.ru:7777/eL/img/Tab1/Kahoot.pdf		26	0,49 %
7	Адаптивный образовательный онлайн-процесс на основе мобильного приложения для SMART – университета (на основе управления сложностью и весом вопросов) <i>International IT University (Информационные системы)</i>	Мадияр Сейтжанов	24	0,46 %
8	https://coderlessons.com/tutorials/akademicheskii/izuchite-osnovy-kompiutera/programmnoe-obespechenie		23	0,44 %
9	https://sud.gov.kz/rus/massmedia/o-novom-vide-sudoproizvodstva-rasskazali-v-verhovnom-sude-zakonkz-14102019-g		21	0,40 %
10	https://ru.qwe.wiki/wiki/Microsoft_Word		19	0,36 %

из базы данных RefBooks (0,00 %)

Все фрагменты найдены в базе данных RefBooks, которая содержит более 3 миллионов текстов от редакторов и авторов.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	АВТОР	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ)
------------------	----------	-------	--

ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО

из домашней базы данных (0,00 %)

Все фрагменты найдены в базе данных вашего университета.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	АВТОР	ДАТА ИНДЕКСАЦИИ	ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	-------	-----------------	------------------------------

ЗАИМСТВОВАНИЙ НЕ НАЙДЕНО

из программы обмена базами данных (2,66 %)

Все фрагменты найдены в базе данных других университетов.

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ НАЗВАНИЕ БАЗЫ ДАННЫХ	АВТОР	ДАТА ИНДЕКСАЦИИ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (КОЛИЧЕСТВО ФРАГМЕНТОВ)
------------------	-------------------------------	-------	-----------------	--

1	Адаптивный образовательный онлайн-процесс на основе мобильного приложения для SMART – университета (на основе управления сложностью и весом вопросов) <i>International IT University (Информационные системы)</i>	Мадияр Сейтжанов	2019-06-11	140 (5) 2,66 %
---	--	------------------	------------	----------------

из интернета (7,06 %)

Все фрагменты найдены в глобальных интернет-ресурсах открытого доступа.

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
1	http://portal.tpu.ru:7777/eL/img/Tab1/Kahoot.pdf	132 (4) 2,51 %
2	https://ru.qwe.wiki/wiki/Microsoft_Word	83 (4) 1,58 %
3	https://otvetus.com/chto-takoe-windows-10-266118	80 (3) 1,52 %
4	https://helpexe.ru/windows/kak-ustanovit-windows-10-na-vneshnij-zhestkij-disk	33 (3) 0,63 %
5	https://coderlessons.com/tutorials/akademicheskii/izuchite-osnovy-kompiutera/programmnoe-obespechenie	23 (1) 0,44 %
6	https://sud.gov.kz/rus/massmedia/o-novom-vide-sudoproizvodstva-rasskazali-v-verhovnom-sude-zakonkz-14102019-g	21 (1) 0,40 %

АННОТАЦИЯ

Целью данной дипломной работы является разработка информационных образовательных систем на основе игровых моделей и методов машинного обучения. Информационная система является мобильным приложением и разработана на платформе Android Studio.

При разработке системы был проведен анализ существующих систем в данной сфере деятельности. Данная система разработана для улучшения системы образования, предназначена для тестирования и самообразования учеников. Система может быть загружена в Google Play Market для общего использования.

АНДАТПА

Дипломдық жұмыстың мақсаты - ойын модельдері мен машиналық оқыту әдістері негізінде ақпараттық білім беру жүйесін дамыту. Ақпараттық жүйе мобильді қосымша болып табылады және Android Studio платформасында жасалынған.

Жүйені әзірлеу кезінде осы қызмет саласындағы қолданыстағы жүйелерге талдау жасалды. Бұл жүйе білім беру жүйесін жетілдіруге арналған, студенттерді тестілеуге және өздігінен білім алуға арналған. Жүйені Google Play Market-ке жалпы пайдалану үшін жүктеуге болады.

ANNATATION

The aim of this thesis is the development of information educational systems based on game models and machine learning methods. The information system is a mobile application and is developed on the Android Studio platform. When developing the system, an analysis was made of existing systems in this field of activity. This system is designed to improve the education system, designed for testing and self-education of students. The system can be uploaded to the Google Play Market for general use.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

1 Современное состояние разработки информационных систем	8
1.1 Обзор и анализ информационных технологий для организации машинного обучения с использованием дистанционного режима	8
1.2 Анализ методов и технологий машинного обучения	10
1.3 Использование гаджетов для организации эффективного дистанционного обучения	14
1.4 Постановка задачи дипломирования	15
2 Структура и алгоритмы функционирования системы	17
2.1 Функциональная структура системы	17
3 Информационное и программное обеспечение системы	18
3.1 Информационное обеспечение	18
3.1.1 Программное обеспечение	19
3.1.2 Системное программное обеспечение	23
3.1.3 Инструментальное программное обеспечение	25
3.1.4 Прикладное программное обеспечение. Описание контрольного примера	25
4 Реализация системы	33
4.1 Инфраструктура реализации системы	33
4.2 Перспективы развития	34
ЗАКЛЮЧЕНИЕ	35
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	36

ВВЕДЕНИЕ

На сегодняшний день использование мобильных устройств является повседневной рутиной. Большинство людей имеют в наличии мобильные устройства, такие как телефоны или планшеты. Не является секретом то, что гаджеты оказывают негативный эффект на учеников, а именно отвлекают от учебного процесса. Чтобы этого избежать, гаджетам необходимо найти альтернативное применение.

Главной целью данной дипломной работы является вовлечение мобильных устройств в образовательный процесс. Это напрямую может способствовать улучшению качества образования и выработке определенных качеств учеников. Основной задачей данной дипломной работы является разработка мобильного приложения для проведения тестирований. Данное мобильное приложение можно использовать как для проверки знаний, так и для улучшения усвоения информации в целом. Также одним из преимуществ данной работы является возможность использования ее для дистанционных работ.

В первой главе проведен анализ существующих аналогичных информационных систем, выявлены преимущества и недостатки, а также отличия от данной работы. Также были описаны положительные моменты вовлечения гаджетов в образовательный процесс и дистанционное обучение.

Во второй главе были разработаны структуры и алгоритмы функционирования системы.

В третьей главе был описан состав информационного и программного обеспечения.

В четвертой главе была рассмотрена инфраструктура реализованной информационной системы и перспективы развития.

В качестве методов решения поставленных задач были использованы такие системы как платформа для разработки мобильных приложений, текстовые редакторы и редакторы диаграмм.

1 Современное состояние разработки информационных систем

1.1 Обзор и анализ информационных технологий для организации машинного обучения с использованием дистанционного режима

На сегодняшний день существует множество приложений для тестирования, но я выделю несколько из них, являющихся более популярными, чем другие.

Первое из них приложение под названием Kahoot!. Kahoot! Является полностью бесплатным для скачивания и использования. Это приложения позволяет создавать и проводить онлайн опросы и викторины в аудитории с помощью мобильных устройств. При этом является обязательным наличие человека, который подготовит опрос либо же викторину и запустит её.[16]

Возможности приложения:

- Создание онлайн-викторин, состоящих из вопросов с одним правильным ответом из множества.
- Проведение онлайн-дискуссии по заданной теме.
- Проведение экспресс-опроса.

Для проведения опроса, дискуссии или викторины в аудитории необходимы технические устройства:

- ПК с трансляцией изображения на экран или мобильное устройство - у преподавателя;
- смартфоны или планшеты с выходом в Интернет - у студентов.

Порядок подготовки и проведения онлайн-мероприятия в аудитории:

- Преподаватель заранее готовит вопрос (для обсуждения или проведения экспресс-опроса) или группу вопросов викторины на сайте разработчика приложения.
- В аудитории преподаватель запускает онлайн-мероприятие с ПК. На экране отображается адрес и пин-код мероприятия для студентов.
- Студенты, используя мобильные устройства, запускают мобильное приложение, вводят пин-код, свое имя и отвечают на вопросы.
- На экране мгновенно отображаются результаты.

Данное приложение не подходит для работы удаленно, вне аудитории, что является его главным недостатком. Также отсутствует какая – либо база вопросов, что исключает возможность использования приложения индивидуально для самообучения.

Второе программное обеспечение называется AnsTester.

Это приложение позволяет создавать разнообразные тесты основанные на системе начисления баллов. AnsTester изначально предназначался для проведения компьютерного тестирования в средней школе. Главным его минусом является отсутствие мобильного приложения, которое позволяло бы всем проходить тесты, используя мобильные устройства, находясь вдали от дома. Помимо тестирования, программа позволяет проводить различные опросы.[17]

Возможности программного обеспечения:

- Создание опросов и тестов.
- Проведение тестов онлайн.
- Проверка тестов.

Еще одним минусом приложения является отсутствие автоматической проверки тестов, которое позволило бы сэкономить время.

Третьим программным обеспечением является Универсальный тест 3.1.0.8. Это приложение также позволяет создавать тесты и проводить тестирования. Универсальный тест позволяет проверять знания учащихся при помощи готовых тестов на самую разнообразную тематику.

Главными минусами этого приложения являются необходимость оплаты и возможность использовать только при наличии персонального компьютера.

Возможности приложения:

- Во время тестирования перед пользователем выводится серия вопросов с вариантами ответов. Вопросы выбираются из базы в случайном порядке.
- Приложение позволяет составлять свои собственные тесты. В них можно использовать графику и тексты с форматированием.

Следующее приложение называется UniTest System. Оно позволяет составлять и проводить тесты, а также предоставляет анализ результатов тестирования. Пользователи программы могут конструировать собственные тесты, включая в них вопросы с множественными вариантами ответов, вопросы с прямым вводом ответов и т.д. Если в тесте требуется ввод ответа, программа может анализировать введённый пользователем текст на присутствие в нём ключевых слов. Тесты позволяют использования графических составляющих.

Возможности приложения:

- Создание и проведение тестов.
- Поддерживается автоматическое создание тестов на основе случайной выборки вопросов из базы данных.
- В тесты можно включать изображения, звуковые и видео - файлы.
- Анализ результатов тестирования.

Главным недостатком является отсутствие мобильного приложения.

1.2 Анализ методов и технологий машинного обучения

Машинное обучение является горячей темой в научных исследованиях и промышленности, и все время разрабатываются все больше новых методологий.

Алгоритм машинного обучения, также называемый моделью, представляет собой математическое выражение, представляющее данные в контексте проблемы, часто бизнес - проблемы. Цель состоит в том, чтобы перейти от данных к пониманию. Например, если онлайн - торговец хочет прогнозировать продажи в следующем квартале, он может использовать алгоритм машинного обучения, который прогнозирует эти продажи на основе прошлых продаж и других соответствующих данных. Аналогичным образом, производитель ветряных мельниц может визуальное контролировать важное оборудование и передавать видеоданные с помощью алгоритмов, обученных распознавать опасные щели.[1]

Ниже приведены несколько методов, которые предлагают обзор и основу в области машинного обучения[3]:

1. Регрессия.
2. Классификация.
3. Кластеризация.
4. Снижение размерности.
5. Методы ансамбля.
6. Нейронные сети и глубокое обучение.
7. Передача обучения.
8. Обучение с подкреплением.
9. Обработка естественного языка.
10. Вложения слов.

Необходимо различать две основные категории машинного обучения: контролируемое и неконтролируемое. Мы применяем контролируемые методы ML, когда у нас есть данные, которые мы хотим предсказать или объяснить. Мы делаем это, используя предыдущие данные входов и выходов, чтобы предсказать выход, основанный на новом входе. Например, вы могли бы использовать контролируемые методы ML, чтобы помочь сервисному бизнесу, который хочет предсказать количество новых пользователей, которые подпишутся на сервис в следующем месяце. В отличие от этого, неконтролируемая ОД рассматривает способы связывания и группировки точек данных без использования целевой переменной для прогнозирования. Другими словами, он оценивает данные с точки зрения черт и использует черты, чтобы сформировать группы предметов, которые похожи друг на друга. Например, вы

можете использовать неконтролируемые методы обучения, чтобы помочь розничному продавцу, который хочет сегментировать продукты с аналогичными характеристиками, без необходимости заранее указывать, какие характеристики использовать.[2]

Наиболее подходящий метод для данного мобильного приложения это кластеризация.

Используя методы кластеризации, мы попадаем в категорию неконтролируемых ML, потому что их цель - группировать или кластеризовать наблюдения, которые имеют сходные характеристики. Методы кластеризации не используют выходные данные для обучения, но вместо этого позволяют алгоритму определять выходные данные. В методах кластеризации мы можем использовать только визуализации для проверки качества решения.

Вернемся к основному функционалу приложения. Данное приложение имеет базу тестов, которые могут состоять из неограниченного количества вопросов. При выборе теста, вопросы вытравляются случайно из базы вопросов. Здесь нам и пригодится метод кластеризации. Он поможет нам сделать тесты более сбалансированными. Так как, всегда имеются сложные и более легкие вопросы, тест может получиться неудачным. Допустим, ученик зашел сдавать тест и ему попадают одни лишь легкие вопросы, в то время другому только сложные вопросы. Получается немного несправедливо. Тут вступает в дело метод кластеризации. На основе многих попыток сдачи тестов у разных людей, алгоритм решает, какие вопросы легкие, средние и сложные. Соответственно он разделяет их на группы, и составляет ограничение на появление тех или иных вопросов в большом количестве. Самым верным решением будет поделить список вопросов, и допустим, если в тесте 15 вопросов, 5 должны быть легкими, 5 средними и остальные 5 тяжелыми. Естественно, что не обязательно делить их в равном количестве. Так и работает метод кластеризации. Так как данное мобильное приложение является лишь начальной версией, машинное обучение не реализовано, но при дальнейшей разработке приложение может стать отличной автоматизированной системой для сдачи тестов.

Технологии машинного обучения.

Машинное обучение - это не отдельная методика или технология, а скорее область вычислительной науки, которая включает в себя многочисленные технологии для создания систем, которые могут извлекать уроки из данных в своей среде, а затем делать прогнозы и предпринимать действия, когда сталкиваются с новой ситуацией.

На сегодняшний день существуют определенные ключевые технологии машинного обучения, на которые следует обратить внимание, чтобы можно было перенести проекты машинного обучения на следующий уровень.[4]

TensorFlow.

TensorFlow предлагает библиотеку JS, которая помогает в разработке машинного обучения. Его API-интерфейсы помогут вам создавать и обучать модели.[4]

Я думаю, что все любители машинного обучения, которые работают с приложениями машинного обучения, знают о TensorFlow. Это библиотека машинного обучения с открытым исходным кодом, которая помогает вам разрабатывать модели ML. Команда Google разработала его. Он имеет гибкую схему инструментов, библиотек и ресурсов, которая позволяет исследователям и разработчикам создавать и развертывать приложения машинного обучения.

Особенности:

- Помогает в создании и обучении ваших моделей.
- Вы также можете запустить свои существующие модели, используя TensorFlow.js, который является конвертером моделей.
- Помогает в нейронной сети.
- Система глубокого обучения полного цикла.
- Обучайтесь, а также создавайте модели ML без особых усилий, используя высокоуровневые API, такие как Keras, с быстрым выполнением.
- Это программное обеспечение с открытым исходным кодом.
- Он также может выполнять численные вычисления с использованием графиков потока данных.
- Работать на GPU и CPU, а также на различных мобильных вычислительных платформах.
- Эффективно разверните и обучите модель в облаке.

Google Cloud ML Engine.

Если вы тренируете свой классификатор на большом количестве данных, ваш ПК или ноутбук могут работать довольно хорошо. Однако, если у вас есть миллионы или миллиарды обучающих данных? Или алгоритм довольно сложный и требует много времени в правильном исполнении? Вы должны использовать Google Cloud ML Engine для вашего спасения. Это хостинговая платформа, на которой разработчики приложений для машинного обучения и ученые данных создают и используют модели машинного обучения оптимального качества.

Особенности:

– Обеспечивает обучение модели машинного обучения, построение, глубокое обучение и прогнозное моделирование.

– Прогнозирование и обучение.

– Это программное обеспечение широко используется предприятиями, то есть обнаруживает облака на спутниковом изображении, быстрее реагируя на электронные письма клиентов.

– Может широко использоваться для обучения сложной модели.

Amazon Machine Learning.

Amazon Machine Learning (AML) - это облачные и надежные программные приложения для машинного обучения, которые могут использоваться всеми уровнями квалификации веб-разработчиков или разработчиков мобильных приложений. Этот управляемый сервис широко используется для создания моделей машинного обучения и создания прогнозов. В дополнение к этому он интегрирует данные из нескольких источников: Redshift, Amazon S3 или RDS.

Особенности:

– Amazon Machine Learning предоставляет мастера и инструменты визуализации.

– Поддерживает три типа моделей, а именно: мультиклассовая классификация, двоичная классификация и регрессия.

– Разрешает пользователям создавать объект источника данных из базы данных MySQL.

– В дополнение к этому он позволяет пользователям создавать объект источника данных из данных, хранящихся в Amazon Redshift.

– Основополагающими понятиями являются модели ML, источники данных, оценки, прогнозы в реальном времени и пакетные прогнозы.

Accord.NET.

Это среда машинного обучения .Net, объединенная с библиотеками обработки изображений и аудио, написанными на C#. Эта структура состоит из нескольких библиотек для широкого спектра приложений, то есть распознавания образов, статистической обработки данных и линейной алгебры. Он включает в себя Accord.Statistics, Accord.Math и Accord.MachineLearning.

Особенности:

– Состоит из более 40 непараметрических и параметрических оценок статистических распределений.

– Используется для создания приложений для компьютерного прослушивания, компьютерного зрения, обработки сигналов и статистики.

– Содержит более 35 тестов гипотез, которые включают двусторонний и односторонний тесты ANOVA, непараметрические тесты, такие как тест Колмогорова-Смирнова и многие другие.

– Он имеет более 38 функций ядра.

Apache Mahout.

Apache Mahout - математически выразительная DSL Scala и распределенная линейная алгебраическая структура. Это открытый и бесплатный проект Apache Software Foundation. Основная цель этой структуры - оперативно реализовать алгоритм для математиков, специалистов по данным и статистиков.[4]

Особенности:

– Внедрение методов машинного обучения, включая рекомендации, кластеризацию и классификацию.

– Расширяемая структура для построения масштабируемых алгоритмов.

– Включает матричные и векторные библиотеки.

– Запускает Apache Hadoop поверх парадигмы MapReduce.

1.3 Использование гаджетов для организации эффективного дистанционного обучения

Что такое электронные гаджеты?

Различные технологически управляемые устройства помогают выполнять или выполнять задачи быстрее и лучше. Они полезны во всех аспектах жизни, от бизнеса, домашнего хозяйства, правительства до образования, их полезность бесценна. В сфере образования они помогают учащимся лучше учиться и дают учителям возможность читать отличные лекции посредством эффективного обучения. К таким устройствам относятся ноутбуки, зарядные устройства, динамики Bluetooth, портативные жесткие диски, мониторы, аккумуляторы, смартфоны и наушники Bluetooth.

Образовательные возможности для обучения детей с использованием современных технологий и мобильных устройств обширные, благодаря онлайн-услугам, приложениям и подключенным гаджетам, легко превращающимся в инструменты дистанционного обучения, центры связи и исследований, музыкальные инструменты, доски для рисования и многое другое.[5]

Использование гаджетов для онлайн обучения имеет множество положительных моментов.

Улучшение сохранения знаний:

Студенты, которые занимаются и интересуются тем, что изучают, лучше сохраняют свои знания. Как уже упоминалось ранее, технологии могут способствовать активному участию в занятиях, что также является очень важным фактором для сохранения знаний. Различные формы технологии могут быть использованы для экспериментов и определения того, что лучше всего подходит для студентов с точки зрения сохранения их знаний.[6]

Поощряет индивидуальное обучение. Никто не учится одинаково из-за разных стилей обучения и разных способностей. Технология предоставляет большие возможности для того, чтобы сделать обучение более эффективным для всех с различными потребностями. Например, студенты могут учиться с собственной скоростью, пересматривать сложные концепции или пропускать их, если это необходимо. Более того, технология может предоставить больше возможностей для учеников с ограниченными возможностями. Доступ к Интернету дает студентам доступ к широкому кругу ресурсов для проведения исследований различными способами, что, в свою очередь, может повысить заинтересованность.[6]

Студенты могут изучать полезные жизненные навыки с помощью технологий.

Технологии могут помочь развить многие практические навыки, включая создание презентаций, обучение различению надежных и ненадежных источников в Интернете, ведение надлежащего онлайн-этикета и написание электронных писем.

1.4 Постановка задачи дипломирования.

Темой данной работы является мобильное приложение для самообучения и проверки знаний учеников, которое также можно использовать дистанционно. Главной задачей является создание работоспособного приложения в Android Studio, которое позволило бы тестировать учеников находящихся вне аудитории. Данное приложение должно способствовать развитию в учениках способности к самообучению. Также оно должно быть понятным и простым к использованию. Основной особенностью является автоматизация, то есть способность приложения производить работу без вмешательства разработчика либо учителя. Приложение должно содержать базу тестов, каждый из которых генерирует вопросы случайно из базы вопросов.

2 Структура и алгоритмы функционирования системы

2.1 Функциональная структура системы

При включении, после короткой загрузки, приложение открывает главный экран. На главном экране находится список тестов. После выбора теста открывается экран тестирования и начинается тест. После выбора ответа можно перейти на следующий вопрос либо выйти на главный экран. После завершения теста появится экран результатов тестирования, откуда можно отправить результаты по почте, начать тест заново или перейти к списку тестов. На рисунке 2.1 изображена схема алгоритма функционирования системы.

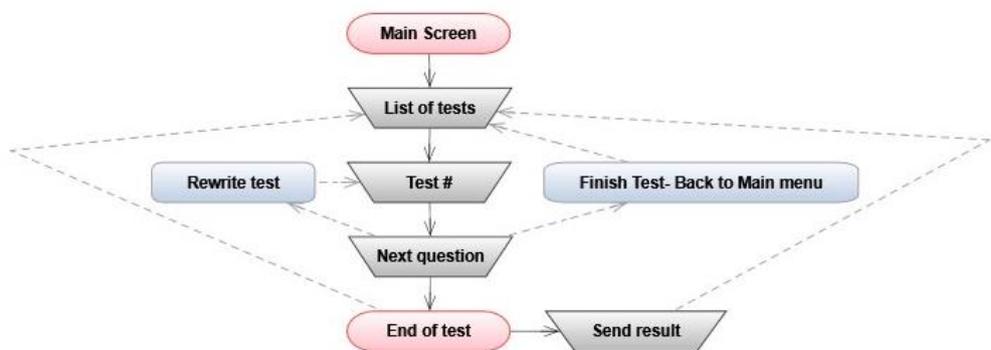


Рисунок 2.1 Алгоритм функционирования системы

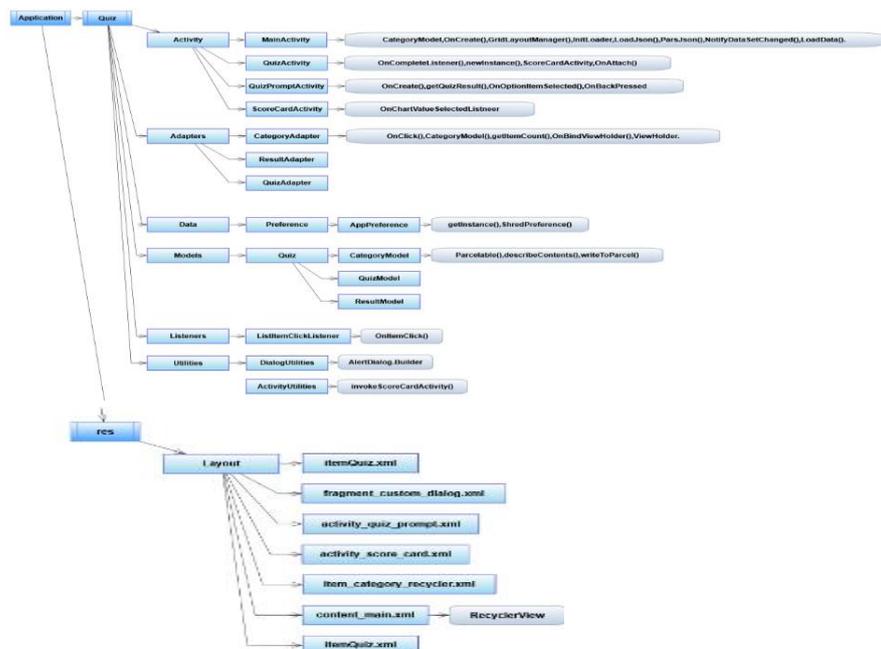


Рисунок 2.2 Функциональная структура системы

3 Информационное и программное обеспечение системы

3.1 Описание информационного обеспечения

Информационное обеспечение реализованного приложения состоит из двух файлов. Оба файла являются файлами с расширением .json. Файл JSON - это файл, в котором хранятся простые структуры данных и объекты в формате JavaScript Object Notation (JSON), который является стандартным форматом обмена данными.

В первом файле под названием quiz_category.json указано количество тестов и их наименования. Это изображено на рисунке 3.1.

```
1  {
2  "items": [
3  {
4      "question_category": "1",
5      "category_name": "Тест 1"
6  },
7  {
8      "question_category": "2",
9      "category_name": "Тест 2"
10 },
11 {
12     "question_category": "3",
13     "category_name": "Тест 3"
14 },
15 {
16     "question_category": "4",
17     "category_name": "Тест 4"
18 },
19 ]
20 }
```

Рисунок 3.1 Файл quiz_category.json

Во втором файле под названием question_set.json на каждый вопрос имеется несколько полей: поле для вопроса, вариантов ответа, номер правильного ответа и номер теста. На рисунке 3.2 показано как это выглядит.

```

3  {
4  "questionnaires": [
5  {
6  "question": "Платформа Андроид основывается на операционной системе:",
7  "answers": [
8  "Linux",
9  "MAC",
10 "Unix",
11 "Windows CE"
12 ],
13 "correct_answer": 0,
14 "question_category": "1"
15 },{
16 "question": "Архитектура платформы Андроид состоит из:",
17 "answers": [
18 "Приложений, фреймворка приложений, библиотек, среды выполнения Андроид, ядра Linux.",
19 "Приложений, библиотек, фреймворка библиотек, среды выполнения Андроид, ядра Linux.",
20 "Приложений, телефонного менеджера, библиотек, среды выполнения Андроид.",
21 "Приложений, телефонного менеджера, библиотек, ядра Linux."
22 ],
23 "correct_answer": 0,
24 "question_category": "1"

```

Рисунок 3.2 Файл question_set.json

На данный момент информационное обеспечение реализовано простым способом, однако в дальнейшем можно добавить отдельную вкладку в меню, для создания и редактирования тестов напрямую в приложении.

3.1.1 Описание программного обеспечения

Программное обеспечение - это набор инструкций, данных или программ, используемых для управления компьютерами и выполнения определенных задач. В отличие от аппаратного обеспечения, которое описывает физические аспекты компьютера, программное обеспечение - это общий термин, используемый для обозначения приложений, сценариев и программ, работающих на устройстве. Программное обеспечение можно рассматривать как переменную часть компьютера, а аппаратную часть - неизменяемую часть.

Программное обеспечение часто делится на категории. Под прикладным программным обеспечением понимаются загруженные пользователем программы, которые удовлетворяют потребности или потребности. Примеры приложений включают офисные пакеты, программы для работы с базами данных, веб-браузеры, текстовые процессоры, инструменты разработки программного обеспечения, редакторы изображений и коммуникационные платформы.[7]

Системное программное обеспечение включает в себя операционные системы и любую программу, которая поддерживает прикладное программное обеспечение.

Термин промежуточное программное обеспечение иногда используется для описания программирования, которое является посредником между прикладным и системным программным обеспечением или между двумя различными типами прикладного программного обеспечения. Например, промежуточное программное обеспечение может использоваться для отправки запроса удаленной работы из приложения на компьютере, на котором установлена операционная система одного типа, в приложение на компьютере с другой операционной системой.

Дополнительной категорией программного обеспечения являются утилиты, которые представляют собой небольшие полезные программы с ограниченными возможностями. Некоторые утилиты поставляются с операционными системами. Как и приложения, утилиты, как правило, устанавливаются отдельно и могут использоваться независимо от остальной части операционной системы.

Проект был реализован на платформе для разработки программного обеспечения Android Studio.

Android Studio - это официальная интегрированная среда разработки (IDE) для операционной системы Google Android, созданная на основе программного обеспечения JetBrains IntelliJ IDEA и разработанная специально для разработки под Android. Он доступен для загрузки в операционных системах Windows, macOS и Linux. Он заменяет Eclipse Android Development Tools (ADT) в качестве основной IDE для разработки собственных приложений Android.

Android Studio была анонсирована 16 мая 2013 года на конференции Google I / O. Он находился на ранней стадии предварительного просмотра, начиная с версии 0.1 в мае 2013 года, затем вступил в бета-версию, начиная с версии 0.8, выпущенной в июне 2014 года. Первая стабильная сборка была выпущена в декабре 2014 года, начиная с версии 1.0.[8]

В текущей стабильной версии предусмотрены следующие функции:

- Поддержка сборки на основе Gradle.
- Android-рефакторинг и быстрые исправления.
- Инструменты Lint для отслеживания производительности, удобства использования, совместимости версий и других проблем.
- Возможности интеграции ProGuard и подписи приложений.

– Мастера на основе шаблонов для создания общих дизайнов и компонентов Android.

– Богатый редактор макетов, который позволяет пользователям перетаскивать компоненты пользовательского интерфейса, возможность предварительного просмотра макетов на нескольких конфигурациях экрана.

– Поддержка создания приложений Android Wear.

– Встроенная поддержка Google Cloud Platform, обеспечивающая интеграцию с Firebase Cloud Messaging (ранее «Google Cloud Messaging») и Google App Engine.

– Android Virtual Device (эмулятор) для запуска и отладки приложений в студии Android.

3.1.2 Системное программное обеспечение

Системное программное обеспечение включает программы, предназначенные для управления самим компьютером, такие как операционная система, утилиты управления файлами и операционная система диска (или DOS).

Системное программное обеспечение - это программное обеспечение, которое предоставляет платформу для других программ. Примерами могут быть операционные системы, антивирусные программы, программы для форматирования дисков, переводчики компьютерных языков и т. д. Они обычно готовятся производителями компьютеров. Это программное обеспечение состоит из программ, написанных на языках низкого уровня, используемых для взаимодействия с оборудованием на самом базовом уровне. Системное программное обеспечение служит интерфейсом между оборудованием и конечными пользователями.[10]

Данный проект реализован на операционной системе Windows 10.

Windows 10 - это серия операционных систем, созданная американской многонациональной технологической компанией Microsoft и выпущенная как часть семейства операционных систем Windows NT. Он является преемником Windows 8.1 (2013), выпущенной почти двумя годами ранее и выпущенной в производство 15 июля 2015 года, и широко выпущенной для широкой публики 29 июля 2015 года. Windows 10 стала доступна для загрузки через MSDN и Technet и доступны в качестве бесплатного обновления для розничных копий пользователей Windows 8 и Windows RT через Windows Store. Windows 10 получает новые сборки на постоянной основе, которые доступны для пользователей без дополнительной оплаты, в дополнение к дополнительным

тестовым сборкам Windows 10, которые доступны для инсайдеров Windows. Устройства в корпоративных средах могут получать эти обновления медленнее или использовать вехи долгосрочной поддержки, которые получают только критические обновления, такие как исправления безопасности, в течение десятилетнего срока службы расширенной поддержки.[9]

Одна из наиболее заметных функций Windows 10 - это поддержка универсальных приложений, расширение приложений в стиле Metro, впервые появившееся в Windows 8. Универсальные приложения могут быть разработаны для работы с несколькими семействами продуктов Microsoft с практически идентичным кодом - включая ПК, планшеты, смартфоны, встроенные системы, Xbox One, Surface Hub и Mixed Reality. Пользовательский интерфейс Windows был переработан для обработки переходов между интерфейсом, ориентированным на мышь, и интерфейсом, оптимизированным для сенсорного экрана, на основе доступных устройств ввода. «В частности, на ПК «2 в 1» оба интерфейса содержат обновленное меню «Пуск», которое включает элементы традиционной Windows 7. Меню «Пуск» с фрагментами Windows 8. В Windows 10 также представлены веб-браузер Microsoft Edge, система виртуальных рабочих столов, функция управления окном и рабочим столом под названием «Просмотр задач», поддержка входа в систему для распознавания отпечатков пальцев и лиц, новые функции безопасности для корпоративных сред и DirectX 12.

Служебной программой или же утилитой для составления данного отчета является Microsoft Word.

Microsoft Word, также называемый просто Word, представляет собой текстовый процессор, разработанный Microsoft. Впервые он был выпущен 25 октября 1983 года под названием Multi-Tool Word для систем Xenix.

Среди его возможностей Word включает в себя встроенную проверку орфографии, тезаурус, словарь и утилиты для манипулирования и редактирования текста.

Несколько более поздних версий Word предоставляют пользователям возможность создавать свои собственные шаблоны форматирования, позволяя им определить файл, в котором заголовок, заголовок, абзац и другие элементы дизайна отличаются от стандартных шаблонов Word. Например, Normal.dot - это основной шаблон, из которого создаются все документы Word. Он определяет поля по умолчанию, а также расположение текста и шрифтов по умолчанию. Хотя normal.dot уже установлен с определенными значениями по умолчанию, пользователь может изменить

normal.dot на новые значения по умолчанию. Это изменит другие документы, созданные с использованием шаблона, обычно непредвиденными способами.[9]

3.1.3 Инструментальное программное обеспечение

Инструментальное ПО это набор программ, которые используются для разработки, ремонта или улучшения других программ или оборудования. Традиционно набор программных инструментов удовлетворял только основные потребности во время разработки программы: типичный набор мог бы состоять из текстового редактора, компилятора, загрузчика ссылок и некоторой формы инструмента отладки. Такой набор концентрируется исключительно на этапе разработки программы и обычно предоставляется системой разработки программ.

В настоящее время признано, что программные средства могут помочь во всех действиях на всех этапах жизненного цикла программного обеспечения, включая действия по управлению и обеспечению качества. Таким образом, полный набор будет посвящен таким вопросам, как спецификация требований, проектирование, валидация, контроль конфигурации и управление проектом. Такие инструменты часто являются частью интегрированной среды разработки программного обеспечения.

Основным и единственным инструментальным программным обеспечением в данном проекте является язык программирования Java. Java является основным языком программирования для платформы Android Studio.

Java - это язык программирования высокого уровня, разработанный Sun Microsystems. Первоначально он был разработан для разработки программ для приставок и портативных устройств, но впоследствии стал популярным выбором для создания веб-приложений. Синтаксис Java похож на C ++, но является строго объектно-ориентированным языком программирования. Например, большинство программ на Java содержат классы, которые используются для определения объектов, и методы, которые назначаются отдельным классам. Java также известен как более строгий, чем C ++, то есть переменные и функции должны быть явно определены. Это означает, что исходный код Java может создавать ошибки или «исключения» легче, чем другие языки, но он также ограничивает другие типы ошибок, которые могут быть вызваны неопределенными переменными или неназначенными типами.

В отличие от исполняемых файлов Windows (.EXE-файлов) или приложений Macintosh (.APP-файлов), Java-программы не запускаются

непосредственно операционной системой. Вместо этого программы Java интерпретируются виртуальной машиной Java или JVM, которая работает на нескольких платформах. Это означает, что все программы Java являются мультиплатформенными и могут работать на разных платформах, включая компьютеры Macintosh, Windows и Unix. Однако JVM должна быть установлена для всех приложений Java или апплетов. К счастью, JVM включена как часть Java Runtime Environment (JRE), которая доступна для бесплатной загрузки.[12]

3.1.4 Прикладное программное обеспечение. Описание контрольного примера

Одна из первых вещей, которые нужно понять в отношении термина «прикладное программное обеспечение», заключается в том, что он чрезвычайно широк.

Прикладное программное обеспечение обычно определяется как любая программа или количество программ, разработанных для конечных пользователей. Вот и все, в двух словах. В этом смысле любую программу конечного пользователя можно назвать «приложением». Отсюда вековая поговорка: «для этого есть приложение». Люди часто используют термин «прикладное программное обеспечение», чтобы говорить о пакетах или группах отдельных программных приложений, используя другой термин «прикладная программа» для обозначения отдельных приложений.

Это связано с тем, что слово «программа» соотносится с дискретной счетной единицей, а слово «программное обеспечение» часто используется для обозначения нескольких отдельных программ.

Примеры прикладного программного обеспечения включают такие элементы, как Microsoft Word, Microsoft Excel или любой из используемых веб-браузеров для навигации в Интернете или сами наборы программного обеспечения, если они предназначены для конечных пользователей. Microsoft Word 2016.

Microsoft Word или MS-WORD (часто называемый Word) - это программа для обработки текстовых изображений, которую пользователи могут печатать. Изготовлено компьютерной компанией Microsoft. Его цель - позволить пользователям печатать и сохранять документы.[13]

Подобно другим текстовым процессорам, он имеет полезные инструменты для создания документов.

- Проверка орфографии и грамматики, подсчет слов (это также подсчитывает буквы и строки).
- Распознавание речи.
- Вставляет картинки в документы.
- Выбор шрифтов.
- Специальные коды.
- Веб-страницы, графики и т. д.
- Таблицы.
- Отображает синонимы слов и может читать текст.
- Печатает по-разному.

Большинство людей знают, что Microsoft Word - это программа для обработки текстов, используемая для создания текстовых документов. Фактически, Microsoft Word стал мерой для всех программ обработки текста на рынке сегодня из-за всех функций и инструментов, которые он предлагает. Однако, что многие люди не понимают, о Microsoft Word, это то, что это больше, чем простая программа обработки текста. Это также развилось в основную программу публикации рабочего стола.

Вы можете не только создавать письма, заметки, отчеты и книги в Microsoft Word, вы также можете создавать листовки, брошюры, плакаты и другие типы публикаций, которые требуют графики и текста. Инструменты, доступные в Word, позволяют делать больше, чем просто вставлять изображения и графику. Вы также можете обрезать, корректировать цвет и удалять фоны, чтобы создать нужную публикацию - и все это без необходимости прикасаться к программе редактирования фотографий или графики.

Instrumind Think Composer.

Это интересный продукт во многих отношениях. Он может создавать широкий спектр типов диаграмм и визуальных моделей сразу после установки, он позволяет пользователям указывать новые типы диаграмм и даже способен генерировать код, когда это уместно и определено пользователем.

Визуальные модели, включенные в программное обеспечение, включают: бизнес-модель, диаграмму классов, концептуальную карту, модель данных, кухню быстрого питания, блок-схему, генеалогическое древо, карту ума, организационную диаграмму, диаграмму последовательности, временную шкалу, диаграмму вариантов использования, веб-среду «универсальная» диаграмма.

ThinkComposer называет каждый из типов диаграмм «доменами». Каждый предопределенный домен предоставляет доступ к набору элементов

диаграммы, но вы можете добавить к нему набор и определить новые домены, используя очень много дополнительных элементов и форм диаграммы, показанных справа:

Каждый узел (в основном называемый в ThinkComposer «объектом») может содержать целую диаграмму, позволяющую пользователям переходить по нескольким уровням или перемещаться по последовательным вкладкам. Объекты на диаграмме могут включать текст, изображения, вложенные файлы, ссылки, настраиваемые поля или таблицы.

Пользовательский интерфейс необычен и не соответствует стилю типичной программы Windows. В верхней части находится набор элементов управления в виде ленты, и, хотя это не лента в стиле Microsoft, ее легко понять.

Слева находится линейный список объектов на диаграмме, а под ним панель, показывающая, если объект выбран, как он связан с другими объектами. Справа находится палитра доступных объектов, а под ним панели маркеров и других предметов, которые можно добавить, чтобы украсить диаграмму. Стоит следить за панелью подсказок в правом нижнем углу окна. Это часто указывает на элементы управления, которые доступны вам в любое время.

Описание контрольного примера.

Приложение начинает свою работу с приветственного экрана, в то же время происходит загрузка приложения.

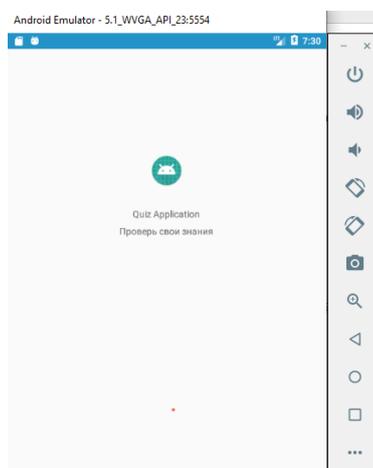


Рисунок 3.1.1 Приветственный экран

После того как приложение включилось, пользователя направляет на главный экран, где находятся варианты тестов на выбор.

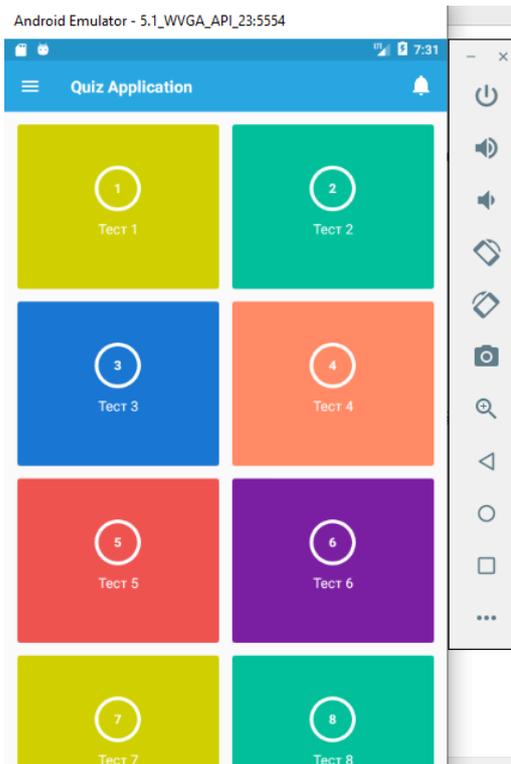


Рисунок 3.1.2 Экран выбора теста

База каждого теста может содержать неограниченное количество вопросов. Вопросы выбираются случайно из базы вопросов, которая может быть заполнена посредством загрузки json файла либо напрямую в коде приложения в файле question_set.json.

Question_set.json выглядит так:

```

1
2
3 {
4   "questionnaires": [
5     {
6       "question": "Платформа Андроид основывается на операционной системе:",
7       "answers": [
8         "Linux",
9         "MAC",
10        "Unix",
11        "Windows CE"
12      ],
13      "correct_answer": 0,
14      "question_category": "1"
15    }, {
16      "question": "Архитектура платформы Андроид состоит из:",
17      "answers": [
18        "Приложений, фреймворка приложений, библиотек, среды выполнения Андроид, ядра Linux.",
19        "Приложений, библиотек, фреймворка библиотек, среды выполнения Андроид, ядра Linux.",
20        "Приложений, телефонного менеджера, библиотек, среды выполнения Андроид.",
21        "Приложений, телефонного менеджера, библиотек, ядра Linux."
22      ],
23      "correct_answer": 0,
24      "question_category": "1"
25    }, {

```

Рисунок 3.1.3 Question_set.json

Здесь на каждый вопрос содержится несколько полей, которые включают вопрос, варианты ответов, номер верного ответа и номер теста.

Слева вверху располагается кнопка меню, которая открывает меню приложения. Здесь можно предоставить ссылки на различные материалы и ресурсы. Данное меню является стандартом для всех мобильных приложений.

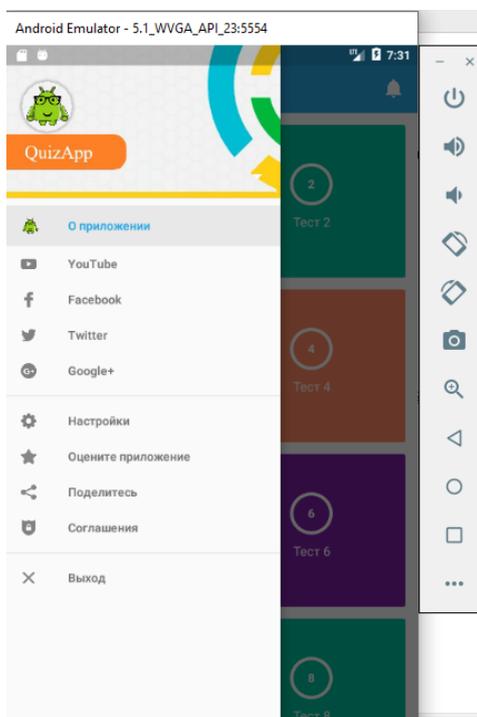


Рисунок 3.1.4 Меню приложения

После выбора теста будет предоставлен выбор, пройти тест снова, если вы его уже проходили, либо отказаться и вернуться на главное меню.

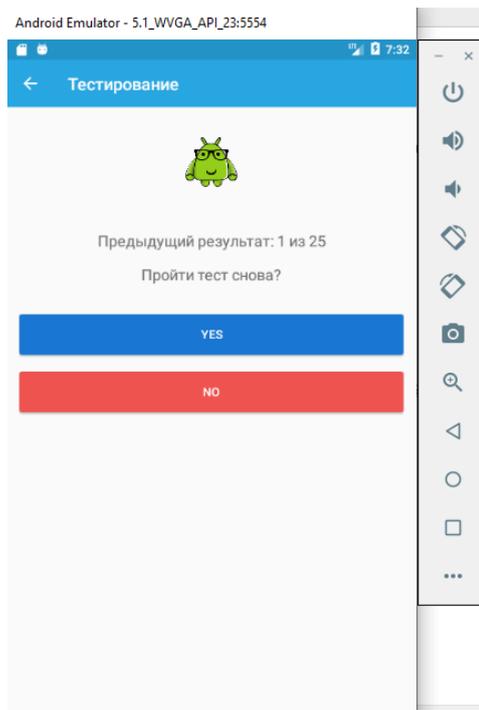


Рисунок 3.1.5 Диалоговое окно сдачи повторного тестирования

После нажатия кнопки да пользователь начинает прохождение тестирования. На этом экране содержится несколько полей, поле для вопроса и для вариантов ответа. При нажатии стрелки назад в левом верхнем углу пользователя перенесет на экран выбора теста. Чтобы перейти на следующий вопрос нужно нажать на кнопку следующий вопрос внизу экрана.

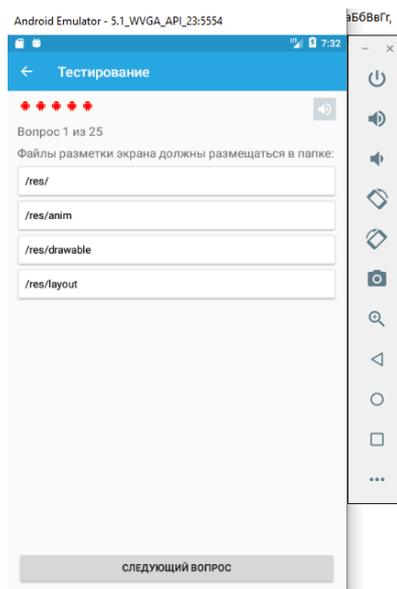


Рисунок 3.1.6 Экран тестирования

После выбора варианта ответа, при правильном выборе, ответ перекрасится в зелёный цвет, при неправильном, в красный и будет показан правильный вариант ответа.

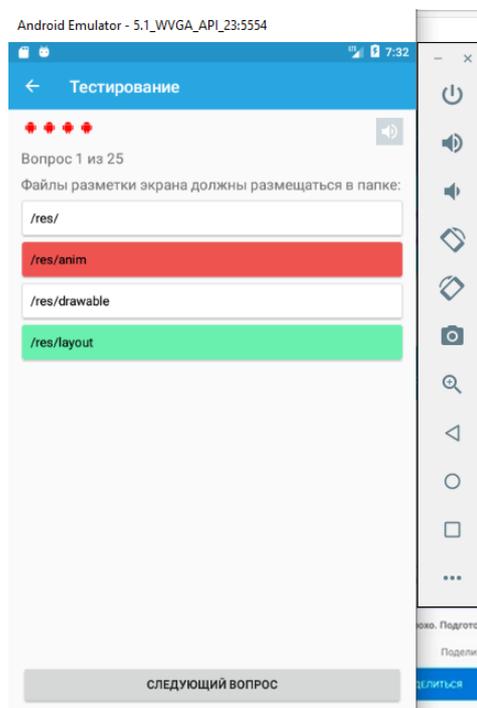


Рисунок 3.1.7 Выбор ответа

После завершения теста откроется экран результатов теста. Он содержит небольшую диаграмму, предоставляющую информацию о том, сколько ответов было выбрано верно, сколько неверно и сколько было пропущено. Ниже имеется список вопросов, с указанными пользователем, а также правильными вариантами ответов. Также пользователю предоставляется выбор: поделиться результатом посредством почты или начать тест заново.

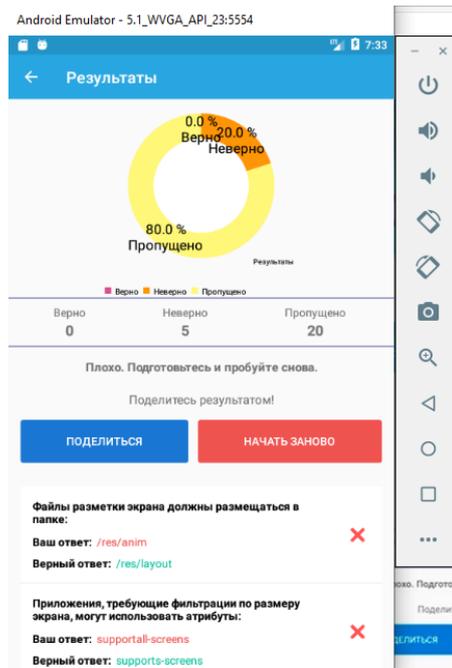


Рисунок 3.1.8 Экран результатов тестирования

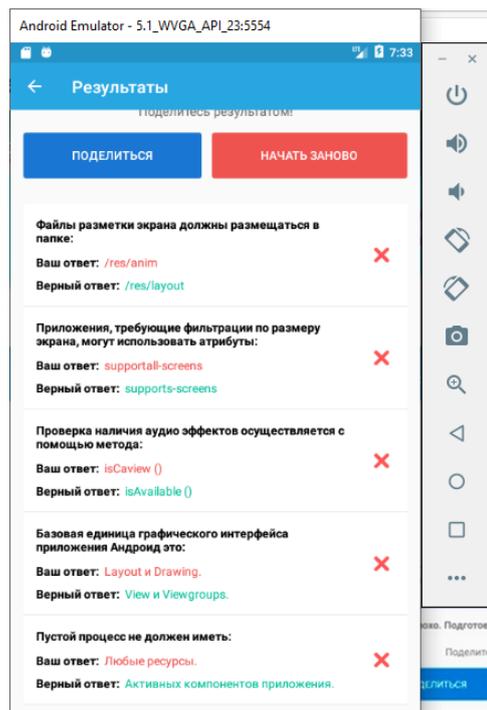


Рисунок 3.1.9 Экран результатов тестирования

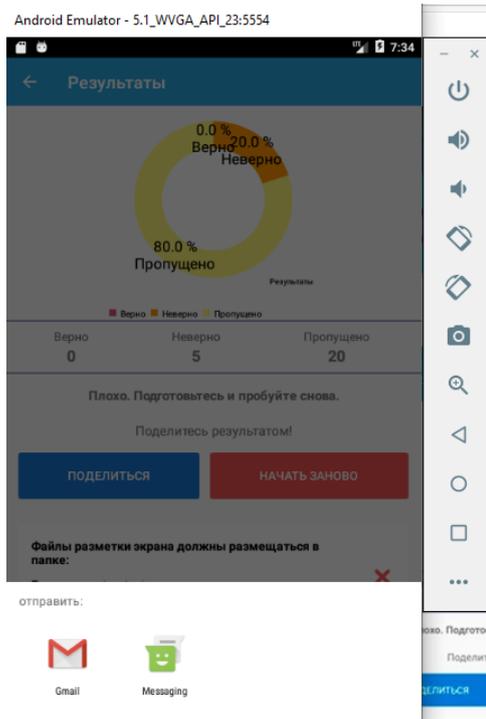


Рисунок 3.1.10 Отправление результатов

4 Реализация системы

4.1 Инфраструктура реализации системы

Виртуальная мобильная инфраструктура (VMI) - это мобильно-ориентированная технология, которая запускает мобильные приложения в мобильной операционной системе (ОС) / виртуальной машине, расположенной на удаленном сервере. По сути, это создает виртуальное устройство, работающее в центре обработки данных, к которому может обращаться приложение тонкого клиента на мобильных конечных точках. VMI расширяет те же принципы, которые позволяли Virtual Desktop Infrastructure (VDI) запускать настольные приложения на настольных компьютерах и мобильных устройствах - только на этот раз к мобильным приложениям осуществляется удаленный доступ с мобильных устройств, таких как смартфоны, планшеты, фаблеты и носимые устройства.

Платформа содержит настраиваемую мобильную ОС, которая размещается на виртуальных машинах (VM). Эти виртуальные машины работают на гипервизоре на централизованном хост-компьютере. Каждая виртуальная машина может запускать несколько пользовательских сессий, а также принимать гостей, используя разные операционные системы. Брокер соединений используется для связи между данными организации и мобильными устройствами, подключенными к сети.

В дополнение к выполнению собственных пользовательских сеансов эта система также отвечает за активацию / аутентификацию пользователей и жизненный цикл инфраструктуры (то есть платформы, виртуальных машин и хранилища). Несколько сеансов пользователей могут выполняться одновременно на одной платформе. Каждый пользователь изолирован в закрытой изолированной программной среде, а каждое авторизованное приложение изолировано от других приложений в изолированной программной среде.

В архитектуре VMI данные отсоединяются от ОС в центральном хранилище. Это дает ИТ-отделу большую гибкость, позволяя каждый раз запускать пользовательские сеансы на другой виртуальной машине. ИТ-отделам также проще управлять централизованным хранилищем и создавать его резервные копии. Данные обычно шифруются с использованием отдельных ключей для каждого пользователя, гарантируя, что даже администратор не сможет получить доступ к данным.

Клиентское приложение и все виртуализированные мобильные приложения доставляются в виде одного тонкого клиентского приложения на мобильное устройство пользователя. Приложение тонкого клиента работает независимо от ОС удаленной платформы и может работать на любом мобильном устройстве и ОС (обычно поддерживаются iOS и Android). Пользовательский интерфейс представляет собой виртуальную имитацию приложений, работающих на сервере, и отображается на устройстве пользователя в виде плоского изображения. Это изображение нельзя анализировать или манипулировать на уровне устройства. Такие функции, как печать экрана, заблокированы. С сервера на устройство передаются только команды отображения: мобильные устройства получают информацию о пикселях от удаленных приложений и, в свою очередь, отправляют информацию о ключе, жесте, местоположении и устройстве. Никакие приложения или данные не сохраняются и не хранятся на самом устройстве. Запатентованный протокол обмена данными UX через IP от Nubo используется для передачи ввода пользователя обратно на удаленный сервер для обеспечения максимальной безопасности.[15]

4.2 Перспективы развития.

За последний год на долю использования платформ Android приходится 73.54%. Это говорит о том, что создание мобильного приложения именно на Android является преимущественно хорошим решением. Приложение было разработано на Android Studio, и в дальнейшем имеет множество вариантов изменения и улучшений по отзывам и требованиям. Данная разработка является начальной версией, следовательно, есть возможность его дальнейшего совершенствования.

ЗАКЛЮЧЕНИЕ

В дипломной работе на основе данных, полученных при изучении предметной области, было разработано и реализовано мобильное приложение на платформе Android Studio и языке программирования Java. Составлен и реализован ряд алгоритмов для выполнения основных функций.

Функции приложения:

- составление и проведение тестов;
- анализ проведенного тестирования;
- отправление результатов;

– Для разработки программного продукта использованы следующие технологии:

- Android Studio 3.6.3
- Java SE 11.0.7

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 <https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0e0000>
- 2 https://ru.wikipedia.org/wiki/Машинное_обучение
- 3 <https://www.osp.ru/cio/2018/05/13054535/>
- 4 <https://www.analyticsinsight.net/the-6-most-important-ai-technologies-in-machine-learning/>
- 5 <https://mightygadget.co.uk/how-electronic-gadgets-are-useful-for-education/>
- 6 <https://www.webanywhere.co.uk/blog/2016/02/top-6-benefits-technology-classroom/>
- 7 <https://searcharchitecture.techtarget.com/definition/software>
- 8 https://en.wikipedia.org/wiki/Android_Studio
- 9 <https://searchenterprisedesktop.techtarget.com/definition/Windows-10>
- 10 <https://www.geeksforgeeks.org/system-software/>
- 11 https://en.wikipedia.org/wiki/Windows_10#User_interface_and_desktop
- 12 <https://techterms.com/definition/java>
- 13 https://en.wikipedia.org/wiki/Microsoft_Word
- 14 <https://www.techopedia.com/definition/4224/application-software>
- 15 <https://nubosoftware.com/what-is-vmi>
- 16 <https://portal.tpu.ru/eL/img/Tab1/Kahoot.pdf>
- 17 <http://softru.ru/down/o-1708.html>

Приложение А

BaseActivity

```
import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.LinearLayout;
import android.support.v7.widget.Toolbar;

import info.fandroid.quizapp.quizapplication.R;

public class BaseActivity extends AppCompatActivity {

    private Context context;
    private Activity activity;

    private Toolbar toolbar;
    private LinearLayout loadingView, noDataView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        activity = BaseActivity.this;
        context = activity.getApplicationContext();
    }

    public void initToolbar(boolean isEnabled) {
        toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        getSupportActionBar().setDisplayHomeAsUpEnabled(isEnabled);
    }

    public void setToolbarTitle(String title) {
        if (getSupportActionBar() != null) {
            getSupportActionBar().setTitle(title);
        }
    }

    public void enableUpButton() {
        if (getSupportActionBar() != null) {
            getSupportActionBar().setDisplayHomeAsUpEnabled(true);
        }
    }
}
```

Продолжение приложения А

```
        getSupportActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

public void initLoader() {
    loadingView = (LinearLayout) findViewById(R.id.loadingView);
    noDataView = (LinearLayout) findViewById(R.id.noDataView);
}

public void showLoader() {
    if (loadingView != null) {
        loadingView.setVisibility(View.VISIBLE);
    }

    if (noDataView != null) {
        noDataView.setVisibility(View.GONE);
    }
}

public void hideLoader() {
    if (loadingView != null) {
        loadingView.setVisibility(View.GONE);
    }
    if (noDataView != null) {
        noDataView.setVisibility(View.GONE);
    }
}

public void showEmptyView() {
    if (loadingView != null) {
        loadingView.setVisibility(View.GONE);
    }

    if (noDataView != null) {
        noDataView.setVisibility(View.VISIBLE);
    }
}

@Override
public void onBackPressed() {
    super.onBackPressed();
}
}
```

Продолжение приложения А

CustomUrlActivity

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.view.MenuItem;
import android.webkit.WebView;

import info.fandroid.quizapp.quizapplication.R;
import info.fandroid.quizapp.quizapplication.constants.AppConstants;
import info.fandroid.quizapp.quizapplication.listeners.WebListener;
import info.fandroid.quizapp.quizapplication.web.WebEngine;

public class CustomUrlActivity extends BaseActivity {

    private Activity activity;
    private Context context;
    private String pageTitle, pageUrl;

    private WebView webView;
    private WebEngine webEngine;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        initVar();
        initView();
        initFunctionality();
    }

    private void initVar() {
        activity = CustomUrlActivity.this;
        context = activity.getApplicationContext();

        Intent intent = getIntent();
        if (intent != null) {
            pageTitle = intent.getStringExtra(AppConstants.BUNDLE_KEY_TITLE);
            pageUrl = intent.getStringExtra(AppConstants.BUNDLE_KEY_URL);
        }
    }
}
```

Продолжение приложения А

```
}  
}
```

```
private void initView() {  
    setContentView(R.layout.activity_custom_url);  
    initWebEngine();  
    initLoader();  
    initToolbar(true);  
    setToolbarTitle(pageTitle);  
    enableUpButton();  
}
```

```
public void initWebEngine() {  
    webView = (WebView) findViewById(R.id.webView);
```

```
    webEngine = new WebEngine(webView, activity);  
    webEngine.initWebView();
```

```
    webEngine.initListeners(new WebListener() {
```

```
        @Override
```

```
        public void onStart() {  
            showLoader();  
        }  
    }
```

```
        @Override
```

```
        public void onLoaded() {  
            hideLoader();  
        }  
    }
```

```
        @Override
```

```
        public void onProgress(int progress) {  
  
        }  
    }
```

```
        @Override
```

```
        public void onNetworkError() {  
            showEmptyView();  
        }  
    }
```

```
        @Override
```

```
        public void onPageTitle(String title) {
```

Продолжение приложения А

```
    }
  });
}

private void initFunctionality() {
    webEngine.loadPage(pageUrl);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            finish();
            return true;

        default:
            return super.onOptionsItemSelected(item);
    }
}
}
```

MainActivity

```
import android.app.Activity;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.support.v4.content.LocalBroadcastManager;
import android.support.v7.widget.GridLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.mikepenz.materialdrawer.AccountHeader;
import com.mikepenz.materialdrawer.AccountHeaderBuilder;
import com.mikepenz.materialdrawer.Drawer;
```

Продолжение приложения А

```
import com.mikepenz.materialdrawer.DrawerBuilder;
import com.mikepenz.materialdrawer.model.DividerDrawerItem;
import com.mikepenz.materialdrawer.model.PrimaryDrawerItem;
import com.mikepenz.materialdrawer.model.ProfileDrawerItem;
import com.mikepenz.materialdrawer.model.SecondaryDrawerItem;
import com.mikepenz.materialdrawer.model.interfaces.IDrawerItem;
import com.mikepenz.materialdrawer.model.interfaces.IProfile;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;

import info.fandroid.quizapp.quizapplication.R;
import info.fandroid.quizapp.quizapplication.adapters.CategoryAdapter;
import info.fandroid.quizapp.quizapplication.constants.AppConstants;
import info.fandroid.quizapp.quizapplication.data.sqlite.NotificationDbController;
import info.fandroid.quizapp.quizapplication.listeners.ListItemClickListener;
import info.fandroid.quizapp.quizapplication.models.notification.NotificationModel;
import info.fandroid.quizapp.quizapplication.models.quiz.CategoryModel;
import info.fandroid.quizapp.quizapplication.utilities.ActivityUtilities;
import info.fandroid.quizapp.quizapplication.utilities.AppUtilities;

public class MainActivity extends BaseActivity {

    private Activity activity;
    private Context context;

    private Toolbar toolbar;

    private RelativeLayout mNotificationView;
    private AccountHeader header = null;
    private Drawer drawer = null;

    private ArrayList<CategoryModel> categoryList;
    private CategoryAdapter adapter = null;
    private RecyclerView recyclerView;
```

Продолжение приложения А

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    activity = MainActivity.this;
    context = getApplicationContext();
    categoryList = new ArrayList<>();

    mNotificationView = (RelativeLayout) findViewById(R.id.notificationView);
    recyclerView = (RecyclerView) findViewById(R.id.rvContentScore);
    recyclerView.setLayoutManager(new GridLayoutManager(activity, 2,
GridLayoutManager.VERTICAL, false));

    adapter = new CategoryAdapter(context, activity, categoryList);
    recyclerView.setAdapter(adapter);

    initLoader();
    loadData();
    initListener();

    final IProfile profile = new ProfileDrawerItem().withIcon(R.drawable.ic_dev);

    header = new AccountHeaderBuilder()
        .withActivity(this)
        .withTranslucentStatusBar(true)
        .withHeaderBackground(R.drawable.header)
        .withOnAccountHeaderProfileImageListener(new
AccountHeader.OnAccountHeaderProfileImageListener() {
            @Override
            public boolean onProfileImageClick(View view, IProfile profile,
boolean current) {
                ActivityUtilities.getInstance().invokeCustomUrlActivity(activity,
CustomUrlActivity.class,
                    getResources().getString(R.string.site),
getResources().getString(R.string.site_url), false);
                return false;
            }
        })
}
```

Продолжение приложения А

```
@Override
public boolean onProfileImageLongClick(View view, IProfile profile,
boolean current) {
    return false;
}
})
.addProfiles(profile)
.build();

drawer = new DrawerBuilder()
    .withActivity(this)
    .withToolbar(toolbar)
    .withHasStableIds(true)
    .withAccountHeader(header)
    .addDrawerItems(
        new PrimaryDrawerItem().withName("О
приложении").withIcon(R.drawable.ic_dev).withIdentifier(10).withSelectable(false
),
        new
SecondaryDrawerItem().withName("YouTube").withIcon(R.drawable.ic_youtube).
withIdentifier(20).withSelectable(false),
        new
SecondaryDrawerItem().withName("Facebook").withIcon(R.drawable.ic_facebook)
.withIdentifier(21).withSelectable(false),
        new
SecondaryDrawerItem().withName("Twitter").withIcon(R.drawable.ic_twitter).with
Identifier(22).withSelectable(false),
        new
SecondaryDrawerItem().withName("Google+").withIcon(R.drawable.ic_google_plu
s).withIdentifier(23).withSelectable(false),
        new DividerDrawerItem(),
        new
SecondaryDrawerItem().withName("Настройки").withIcon(R.drawable.ic_settings
).withIdentifier(30).withSelectable(false),
        new SecondaryDrawerItem().withName("Оцените
приложение").withIcon(R.drawable.ic_rating).withIdentifier(31).withSelectable(fa
lse),
        new
SecondaryDrawerItem().withName("Поделитесь").withIcon(R.drawable.ic_share).
withIdentifier(32).withSelectable(false),
```

Продолжение приложения А

```
        new
SecondaryDrawerItem().withName("Соглашения").withIcon(R.drawable.ic_privacy_policy).withIdentifier(33).withSelectable(false),

        new DividerDrawerItem(),
        new
SecondaryDrawerItem().withName("Выход").withIcon(R.drawable.ic_exit).withIdentifier(40).withSelectable(false)
    )
    .withOnDrawerItemClickListener(new
Drawer.OnItemClickListener() {
        @Override
        public boolean onItemClick(View view, int position, IDrawerItem drawerItem) {

            if (drawerItem != null) {
                Intent intent = null;
                if (drawerItem.getIdentifier() == 10) {
                    ActivityUtilities.getInstance().invokeNewActivity(activity, AboutDevActivity.class, false);

                } else if (drawerItem.getIdentifier() == 20) {
                    AppUtilities.youtubeLink(activity);
                } else if (drawerItem.getIdentifier() == 21) {
                    AppUtilities.facebookLink(activity);
                } else if (drawerItem.getIdentifier() == 22) {
                    AppUtilities.twitterLink(activity);
                } else if (drawerItem.getIdentifier() == 23) {
                    AppUtilities.googlePlusLink(activity);
                } else if (drawerItem.getIdentifier() == 30) {
                    // TODO: invoke SettingActivity
                } else if (drawerItem.getIdentifier() == 31) {
                    AppUtilities.rateThisApp(activity);
                } else if (drawerItem.getIdentifier() == 32) {
                    AppUtilities.shareApp(activity);
                } else if (drawerItem.getIdentifier() == 33) {

                    ActivityUtilities.getInstance().invokeCustomUrlActivity(activity, CustomUrlActivity.class,
                        getResources().getString(R.string.privacy),
```

Продолжение приложения А

```
getResources().getString(R.string.privacy_url), false);
        } else if (drawerItem.getIdentifier() == 40) {

        }
    }

    return false;
}
})
.withSavedInstance(savedInstanceState)
.withShowDrawerOnFirstLaunch(false)
.withShowDrawerUntilDraggedOpened(false)
.build();
}
```

@Override

```
public void onBackPressed() {
    if (drawer != null && drawer.isDrawerOpen()) {
        drawer.closeDrawer();
    } else {
        AppUtilities.tapPromtToExit(this);
    }
}
```

```
private void loadData() {
    showLoader();
    loadJson();
}
```

```
private void loadJson() {
    StringBuffer sb = new StringBuffer();
    BufferedReader br = null;
    try{
        br = new BufferedReader(new
InputStreamReader(getAssets().open(AppConstants.CONTENT_FILE)));
        String temp;
        while ((temp = br.readLine()) != null)
            sb.append(temp);

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
```

Продолжение приложения А

```
    try {
        br.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
parseJson(sb.toString());
}

private void parseJson(String jsonData) {
    try {
        JSONObject jsonObject = new JSONObject(jsonData);
        JSONArray jsonArray =
            jsonObject.getJSONArray(AppConstants.JSON_KEY_ITEMS);

        for (int i = 0; i < jsonArray.length(); i++) {
            JSONObject object = jsonArray.getJSONObject(i);

            String categoryId =
                object.getString(AppConstants.JSON_KEY_CATEGORY_ID);
            String categoryName =
                object.getString(AppConstants.JSON_KEY_CATEGORY_NAME);

            categoryList.add(new CategoryModel(categoryId, categoryName));
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    hideLoader();
    adapter.notifyDataSetChanged();
}

private void initListener() {

    //notification view click listener
    mNotificationView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            ActivityUtilities.getInstance().invokeNewActivity(activity,
                NotificationListActivity.class, false);
        }
    });
}
```

Продолжение приложения А

```
// recycler list item click listener
adapter.setItemClickListener(new ListItemClickListener() {
    @Override
    public void onItemClick(int position, View view) {

        CategoryModel model = categoryList.get(position);
        ActivityUtilities.getInstance().invokeCommonQuizActivity(activity,
        QuizPromptActivity.class, model.getCategoryId(), true);
    }
});
}
// received new broadcast
private BroadcastReceiver newNotificationReceiver = new BroadcastReceiver()
{

    @Override
    public void onReceive(Context context, Intent intent) {
        initNotification();
    }
};

public void initNotification() {
    NotificationDbController notificationDbController = new
NotificationDbController(context);
    TextView notificationCount = (TextView)
findViewById(R.id.notificationCount);
    notificationCount.setVisibility(View.INVISIBLE);

    ArrayList<NotificationModel> notiArrayList =
notificationDbController.getUnreadData();

    if (notiArrayList != null && !notiArrayList.isEmpty()) {
        int totalUnread = notiArrayList.size();
        if (totalUnread > 0) {
            notificationCount.setVisibility(View.VISIBLE);
            notificationCount.setText(String.valueOf(totalUnread));
        } else {
            notificationCount.setVisibility(View.INVISIBLE);
        }
    }
}
```

Продолжение приложения А

```
@Override
protected void onResume() {
    super.onResume();

    //register broadcast receiver
    IntentFilter intentFilter = new IntentFilter(AppConstants.NEW_NOTI);

    LocalBroadcastManager.getInstance(this).registerReceiver(newNotificationReceiver, intentFilter);

    initNotification();

    // load full screen ad
    // TODO: AdsUtilities.getInstance(mContext).loadFullScreenAd(mActivity);
}
}
```

QuizActivity

```
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.support.annotation.RequiresApi;
import android.support.v4.app.FragmentManager;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.text.Html;
import android.util.Log;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.TextView;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.io.BufferedReader;
import java.io.IOException;
```

Продолжение приложения А

```
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
```

```
import info.fandroid.quizapp.quizapplication.R;
import info.fandroid.quizapp.quizapplication.adapters.QuizAdapter;
import info.fandroid.quizapp.quizapplication.constants.AppConstants;
import info.fandroid.quizapp.quizapplication.data.preference.AppPreference;
import info.fandroid.quizapp.quizapplication.listeners.ListItemClickListener;
import info.fandroid.quizapp.quizapplication.models.quiz.QuizModel;
import info.fandroid.quizapp.quizapplication.models.quiz.ResultModel;
import info.fandroid.quizapp.quizapplication.utilities.ActivityUtilities;
import info.fandroid.quizapp.quizapplication.utilities.BeatBox;
import info.fandroid.quizapp.quizapplication.utilities.DialogUtilities;
import info.fandroid.quizapp.quizapplication.utilities.SoundUtilities;
```

```
public class QuizActivity extends BaseActivity implements
DialogUtilities.OnCompleteListener {
```

```
    private Activity mActivity;
    private Context mContext;
    private ImageButton btnSpeaker;
    private Button btnNext;
    private RecyclerView mRecyclerQuiz;
    private TextView tvQuestionText;
    private TextView tvQuestionTitle;
    private ImageView imgFirstLife, imgSecondLife, imgThirdLife,
imgFourthLife, imgFifthLife;
```

```
    private QuizAdapter mAdapter = null;
    private List<QuizModel> mItemList;
    ArrayList<String> mOptionList;
    ArrayList<String> mBackgroundColorList;
```

```
    private int mQuestionPosition = 0;
    private int mQuestionsCount = 0;
    private int mScore = 0, mWrongAns = 0, mSkip = 0;
    private int mLifeCounter = 5;
    private boolean mUserHasPressed = false;
    private boolean mIsSkipped = false, mIsCorrect = false;
```

Продолжение приложения А

```
private String mQuestionText, mGivenAnsText, mCorrectAnsText,  
mCategoryId;
```

```
private ArrayList<ResultModel> mResultList;
```

```
private BeatBox mBeatBox;
```

```
private List<SoundUtilities> mSounds;
```

```
private boolean isSoundOn;
```

```
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

```
    //TODO: initializeRewardedAds();
```

```
    //TODO: loadRewardedVideoAds();
```

```
    initVar();
```

```
    initView();
```

```
    loadData();
```

```
    initListener();
```

```
}
```

```
private void initVar() {
```

```
    mActivity = QuizActivity.this;
```

```
    mContext = mActivity.getApplicationContext();
```

```
    Intent intent = getIntent();
```

```
    if (intent != null) {
```

```
        mCategoryId =
```

```
intent.getStringExtra(AppConstants.BUNDLE_KEY_INDEX);
```

```
    }
```

```
    mItemList = new ArrayList<>();
```

```
    mOptionList = new ArrayList<>();
```

```
    mBackgroundColorList = new ArrayList<>();
```

```
    mResultList = new ArrayList<>();
```

```
    mBeatBox = new BeatBox(mActivity);
```

```
    mSounds = mBeatBox.getSounds();
```

Продолжение приложения А

```
}

private void initView() {
    setContentView(R.layout.activity_quiz);

    imgFirstLife = (ImageView) findViewById(R.id.firstLife);
    imgSecondLife = (ImageView) findViewById(R.id.secondLife);
    imgThirdLife = (ImageView) findViewById(R.id.thirdLife);
    imgFourthLife = (ImageView) findViewById(R.id.fourthLife);
    imgFifthLife = (ImageView) findViewById(R.id.fifthLife);
    btnSpeaker = (ImageButton) findViewById(R.id.btnSpeaker);
    btnNext = (Button) findViewById(R.id.btnNext);

    tvQuestionText = (TextView) findViewById(R.id.tvQuestionText);
    tvQuestionTitle = (TextView) findViewById(R.id.tvQuestionTitle);

    mRecyclerQuiz = (RecyclerView) findViewById(R.id.rvQuiz);
    mRecyclerQuiz.setLayoutManager(new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false));

    mAdapter = new QuizAdapter(mContext, mActivity, mOptionList,
mBackgroundColorList);
    mRecyclerQuiz.setAdapter(mAdapter);

    initToolbar(true);
    setToolbarTitle(getString(R.string.quiz));
    enableUpButton();
    initLoader();
}

@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)
private void loadData() {
    showLoader();

    isSoundOn =
AppPreference.getInstance(mActivity).getBoolean(AppConstants.KEY_SOUND,
true);
    setSpeakerImage();

    loadJson();
}
```

Продолжение приложения А

```
@RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)  
private void setSpeakerImage() {  
    if (isSoundOn) {  
        btnSpeaker.setImageResource(R.drawable.ic_speaker);  
    } else {  
        btnSpeaker.setImageResource(R.drawable.ic_speaker_not);  
    }  
}
```

```
public void initListener() {  
    btnSpeaker.setOnClickListener(new View.OnClickListener() {  
        @RequiresApi(api = Build.VERSION_CODES.LOLLIPOP)  
        @Override  
        public void onClick(View view) {  
            isSoundOn = !isSoundOn;  
        }  
    });  
}
```

```
AppPreference.getInstance(mActivity).setBoolean(AppConstants.KEY_SOUND,  
isSoundOn);  
    setSpeakerImage();  
};
```

```
btnNext.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if (!mUserHasPressed) {  
            FragmentManager manager = getSupportFragmentManager();  
            DialogUtilities dialog =  
DialogUtilities.newInstance(getString(R.string.skip_text),  
getString(R.string.skip_prompt), getString(R.string.yes), getString(R.string.no),  
AppConstants.BUNDLE_KEY_SKIP_OPTION);  
            dialog.show(manager,  
AppConstants.BUNDLE_KEY_DIALOG_FRAGMENT);  
        } else {  
            updateResultSet();  
            setNextQuestion();  
        }  
    }  
});
```

Продолжение приложения А

```
mAdapter.setOnClickListener(new ListItemClickListener() {
    @Override
    public void onItemClick(int position, View view) {
        if (!mUserHasPressed) {
            int clickedAnswerIndex = position;
            if (mItemList.get(mQuestionPosition).getCorrectAnswer() != -1) {
                for (int currentItemIndex = 0; currentItemIndex <
mOptionList.size(); currentItemIndex++) {
                    if (currentItemIndex == clickedAnswerIndex && currentItemIndex
== mItemList.get(mQuestionPosition).getCorrectAnswer()) {
                        mBackgroundColorList.set(currentItemIndex,
AppConstants.COLOR_GREEN);
                        mScore++;
                        mIsCorrect = true;
                        if (isSoundOn) {

mBeatBox.play(mSounds.get(AppConstants.BUNDLE_KEY_ZERO_INDEX));
                    }
                    } else if (currentItemIndex == clickedAnswerIndex &&
!(currentItemIndex == mItemList.get(mQuestionPosition).getCorrectAnswer())) {
                        mBackgroundColorList.set(currentItemIndex,
AppConstants.COLOR_RED);
                        mWrongAns++;
                        if (isSoundOn) {

mBeatBox.play(mSounds.get(AppConstants.BUNDLE_KEY_SECOND_INDEX));
                    }
                    }
                    decreaseLifeAndStatus();
                    } else if (currentItemIndex ==
mItemList.get(mQuestionPosition).getCorrectAnswer()) {
                        mBackgroundColorList.set(currentItemIndex,
AppConstants.COLOR_GREEN);
                        ((LinearLayoutManager)
mRecyclerView.getLayoutManager()).scrollToPosition(currentItemIndex);
                    }
                }
            } else {
                mBackgroundColorList.set(clickedAnswerIndex,
AppConstants.COLOR_GREEN);
                mScore++;
                mIsCorrect = true;
            }
        }
    }
}
```

Продолжение приложения А

```
mBeatBox.play(mSounds.get(AppConstants.BUNDLE_KEY_ZERO_INDEX));
    }

    mGivenAnsText =
mItemList.get(mQuestionPosition).getAnswers().get(clickedAnswerIndex);
    mCorrectAnsText =
mItemList.get(mQuestionPosition).getAnswers().get(mItemList.get(mQuestionPo
sition).getCorrectAnswer());

    mUserHasPressed = true;
    mAdapter.notifyDataSetChanged();
    }
}

});

}
```