

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ОБЩЕЙ МАТЕМАТИКИ

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОДГОТОВКИ ШКОЛЬНИКОВ
К ОЛИМПИАДАМ ПО МАТЕМАТИКЕ ДЛЯ ОС ANDROID**

Выпускная квалификационная работа
обучающегося по направлению подготовки
01.02.03 Прикладная математика и информатика
очной формы обучения, группы 07001406
Суворкина Дмитрия Андреевича

Научный руководитель
д.т.н, профессор
Аверин Г.В.

БЕЛГОРОД
2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 СОСТОЯНИЕ ВОПРОСА И АНАЛИЗ ПРОГРАММНЫХ ПРОДУКТОВ....	6
1.1 Мобильные приложения.....	6
1.2 Сайты, посвященные тестам	12
1.3 Результат анализа существующих аналогов	14
2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ.....	16
2.1 Выбор операционной системы.....	16
2.2 Выбор интегрированной среды разработки	17
2.2.1 Среда разработки Eclipse.....	17
2.2.2 Среда разработки IntelliJ IDEA	18
2.2.3 Среда разработки Android Studio.....	20
3 РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОДГОТОВКИ ШКОЛЬНИКОВ К ОЛИМПИАДАМ	22
3.1 Формирование требований к разрабатываемому приложению	22
3.2 Общая схема системы.....	22
3.3 Структура приложения	23
3.4 Особенности языка java в разработке мобильного приложения.....	23
4 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ.....	40
5.1 Тестирование навигации	40
4.2 Тестирование защиты от случайного выхода	45
4.3 Тестирование прохождения тестов	47
4.3 Тестирование работы профиля пользователя	50
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	55
ПРИЛОЖЕНИЕ А Краткий глоссарий	58
ПРИЛОЖЕНИЕ Б Техническое задание на мобильное приложение для подготовки школьников к олимпиадам	59

ВВЕДЕНИЕ

Математика — это фундаментальная наука, методы которой, активно применяются во многих естественных дисциплинах, таких как физика, химия и даже биология. Сама по себе, эта область знаний оперирует абстрактными отношениями и взаимосвязями, то есть такими сущностями, которые сами по себе не являются чем-то вещественным.

Но тем не менее, стоит только математике вступить в область любой науки о мире, она сразу воплощается в описание, моделирование и предсказание вполне себе конкретных и реальных природных процессов.

Математика позволяет развить некоторые важные умственные качества человека. Это аналитические, дедуктивные (способность к обобщению), критические, прогностические (умение прогнозировать, мыслить на несколько шагов вперед) способности. Также эта дисциплина улучшает возможности абстрактного мышления (ведь это абстрактная наука), способность концентрироваться, тренирует память и усиливает быстроту мышления.

Существует несколько видов оценивания знаний учащихся, один из них – тестирование. С помощью тестирования можно быстро и оперативно выяснить уровень знаний человека. Тестирование состоит из ряда вопросов, к которым прилагаются несколько вариантов ответов. Верный вариант ответа на вопрос, обычно, является и единственным, но существуют и вопросы с набором нескольких верных ответов.

В Интернете можно найти множество сайтов, на которых можно пройти тестирование, практически, по любой дисциплине, школьному предмету и каких-либо других видов деятельности. В мобильной области таких тестов существует гораздо меньше, так как на их создание уходит много времени и ресурсов. Но не стоит исключать данный вид тестирования, так как, практически, у каждого человека любого возраста имеется мобильный телефон, либо смартфон. Следовательно, проходить всевозможные тестирования можно с любого места, где бы человек не находился. В дороге,

на отдыхе, дома, в саду можно открыть приложение и пройти интересующий тест, тем более, что, зачастую, этот процесс не требует постоянного доступа в Интернет. Результаты такого тестирования могут записываться в память телефона и в дальнейшем отображаться в Профиле пользователя.

Цель работы – разработка приложения для прохождения тестирования по математике для операционной системы Android.

Для достижения поставленной цели в работе решены следующие задачи:

- проведен обзор аналогов приложения и выбраны технологии программной реализации мобильного приложения;
- определена операционная система и визуальная среда разработки;
- выбрана профессиональная инструментальная среда, в которой будет вестись разработка;
- разработано приложение с помощью выбранной профессиональной инструментальной среды;
- протестирован программный продукт;
- разработана программная документация к приложению.

Структура и объем работы: выпускная квалификационная работа выполнена на 72 страницах машинописного текста, состоит из введения, четырех глав, заключения и двух приложений.

В первой главе проводится обзор аналогов разработанного приложения среди различных операционных систем.

Во второй главе проводится выбор инструментальной среды разработки и языка программирования.

В третьей главе описываются этапы разработки программных продуктов в выбранной инструментальной среде разработки. Рассмотрены основные принципы жизненного цикла Android-приложения. Произведена разработка приложения.

В четвертой главе проводится тестирование программного продукта.

1 СОСТОЯНИЕ ВОПРОСА И АНАЛИЗ ПРОГРАММНЫХ ПРОДУКТОВ

Основная цель работы – разработка приложения для прохождения тестирования по математике для операционной системы Android.

Для достижения поставленной цели необходимо провести обзор аналогов приложения.

Чтобы показать актуальность решенной задачи, был произведён поиск аналогов разрабатываемого приложения. Анализ велся на основе количества скачиваний отобранных приложений.

Поиск аналогов был начат с официального магазина приложений мобильной операционной системы Android, чтобы найти приложения, выполняющие такой же или похожий функционал. Официальным магазином приложений Android является Google Play Store [5].

1.1 Мобильные приложения

Первым найденным приложением является «Тест по математике» (рисунок 1.1).

Это приложение создано группой разработчиков под названием Tiny4games. Приложение включает в себя наборы тестов по математике. Для конечного пользователя прохождение тестов предоставляется в игровой форме. На момент написания данной работы приложение установили около 100 тысяч пользователей. Средний балл приложения – 3.9.

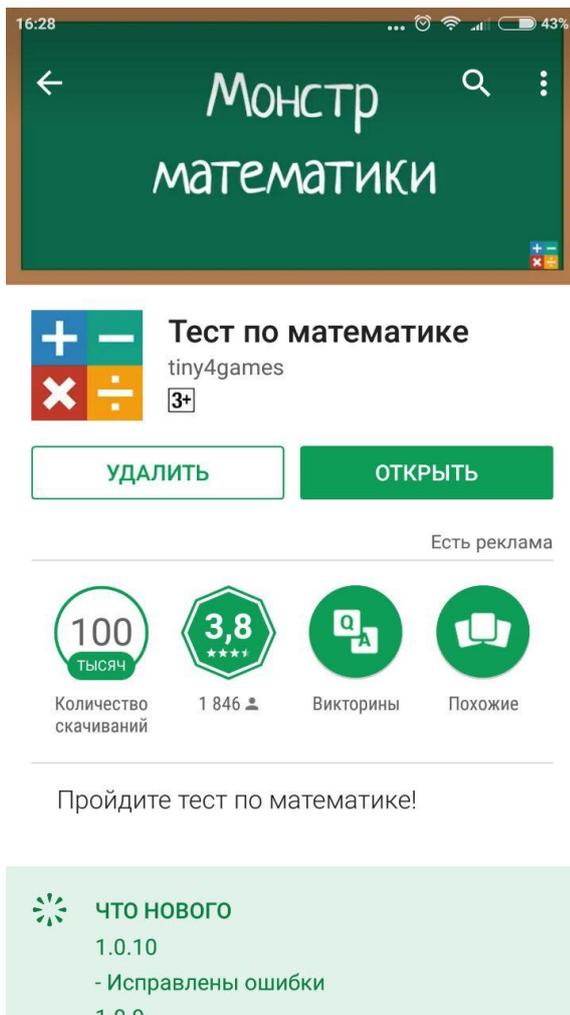


Рисунок 1.1. - Страница приложения «Тест по математике» в Google Play

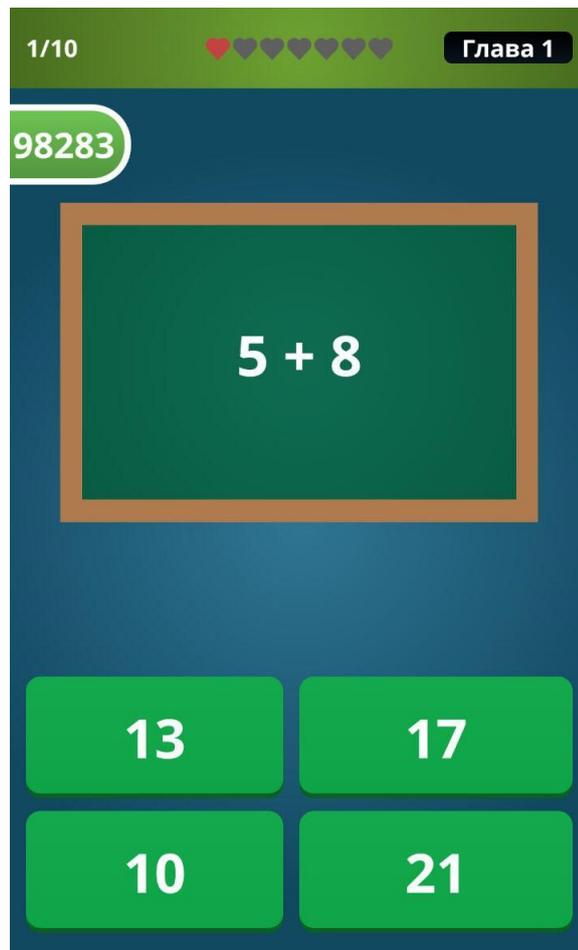


Рисунок 1.2 - Окно игрового процесса в приложении «Тест по математике»

Положительные качества данного приложения:

- Тестирование ведется в виде игры;
- наличие уровней сложности.

Недостатки данного приложения:

- приложение имеет не очень удачный дизайн;
- отсутствует возможность выбора уровня сложности теста;
- не поддерживается профиль пользователя.

На втором месте в списке результатов поиска Google Play Market находится приложение «Я умник – Тест на образованность» (рисунок 1.3).

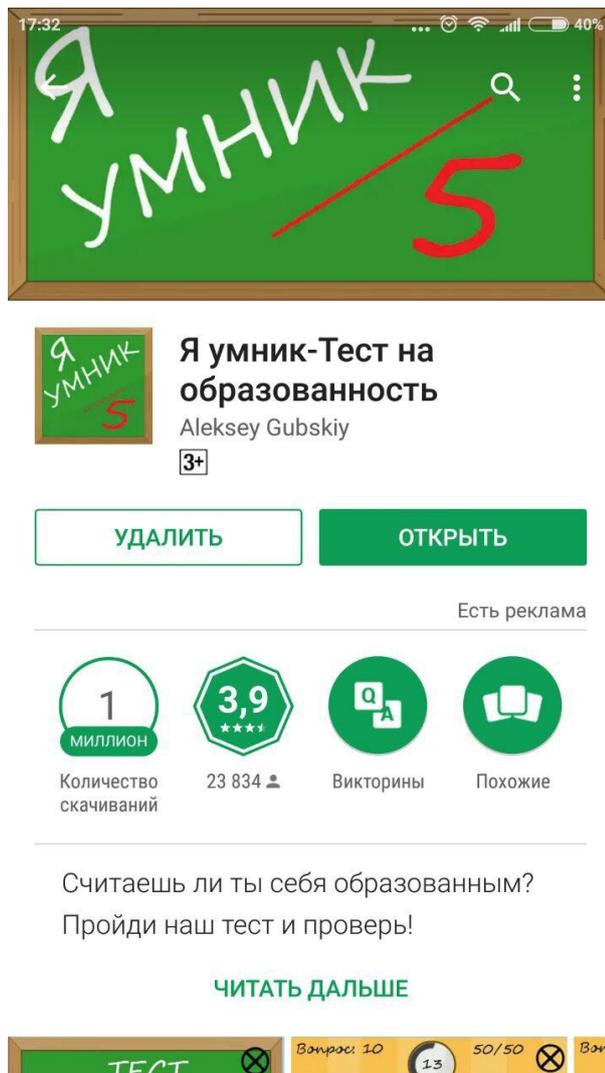


Рисунок 1.3. - Страница приложения «Я умник – Тест на образованность»

Это приложение создано одним мобильным разработчиком. Приложение включает в себя наборы тестов по нескольким дисциплинам, таким как математика, русский язык, литература, история и география. Для конечного пользователя прохождение тестов предоставляется в игровой форме. Приложение установили около 1 миллиона пользователей. Средний балл приложения – 3.9.



Рисунок 1.4. - Окно игрового процесса в приложении «Я умник – Тест на образованность»

Достоинства данного приложения:

- тестирование в виде игры;
- наличие нескольких дисциплин;
- наличие уровней сложности;
- встроенный таймер.

Недостатки данного приложения:

- дисциплины проходятся последовательно;
- отсутствует возможность выбора уровня сложности теста;
- наблюдается ограничение в разнообразии вопросов;
- не поддерживается профиль пользователя.

На третьем месте в списке находится приложение «Знаток» (рисунок 1.5).

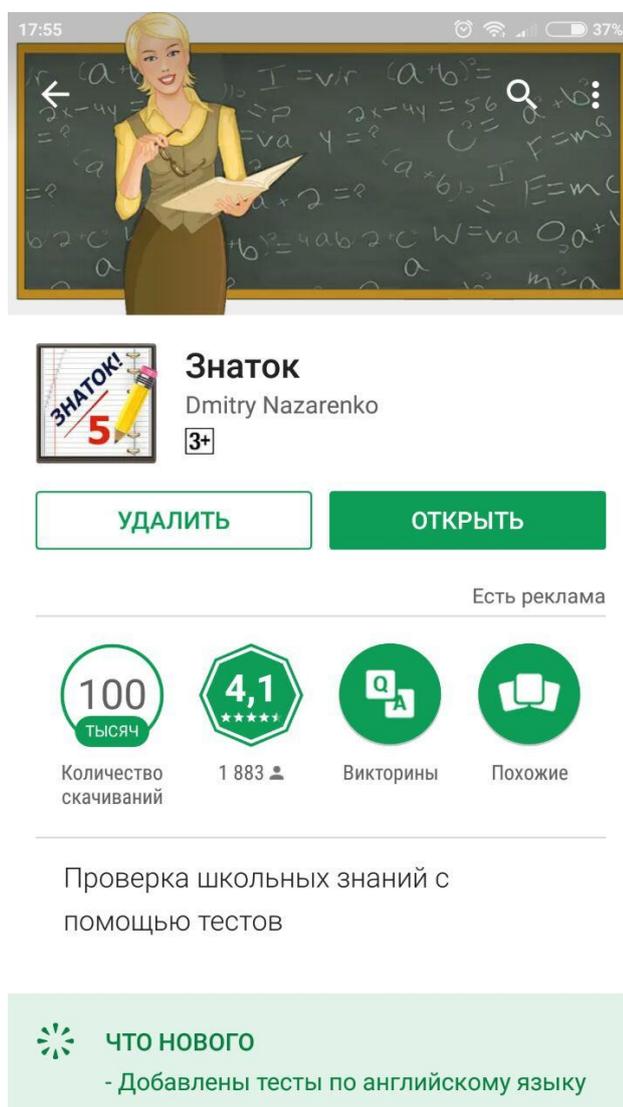


Рисунок 1.5. - Страница приложения «Знаток»

Это приложение создано одним мобильным разработчиком. Приложение включает в себя четыре набора тестов по нескольким дисциплинам, таким как математика, русский язык, география и английский. Приложение имеет приятный внешний вид, информативное меню. Для конечного пользователя прохождение тестов, также, предоставляется в игровой форме. Приложение установили около 100 тысяч пользователей. Средний балл приложения – 4,1.

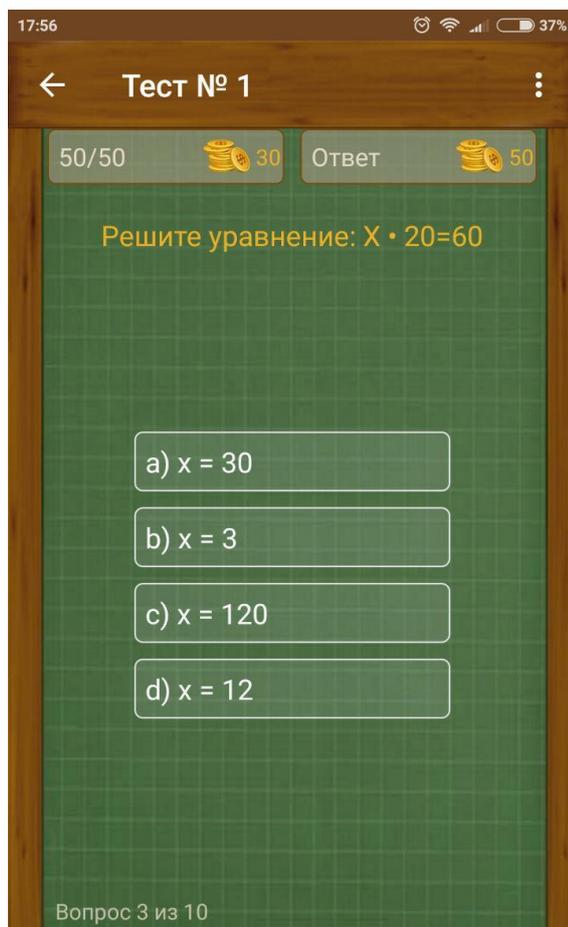


Рисунок 1.6. - Окно игрового процесса в приложении «Знаток»

Достоинства данного приложения:

- тестирование в виде игры;
- приятный визуальный интерфейс;
- поддержка нескольких дисциплин;
- наличие уровней сложности;

Недостатки данного приложения:

- дисциплины проходятся последовательно;
- отсутствует возможность выбора уровня сложности теста;
- наблюдается ограничение в разнообразии вопросов;
- не поддерживается профиль пользователя.

1.2 Сайты, посвященные тестам

Второй тип похожих существующих решений – сайты, посвященные тестам.

В интернете есть большое количество сайтов, посвященных тестам по различным предметам. Некоторые из них уже переросли в большие информационные порталы, на которых можно не только пройти интересующий тест, но и изучить теорию по интересующей дисциплине.

Например, сайт Национального исследовательского университета «Высшая школа экономики» (рисунок 1.7).

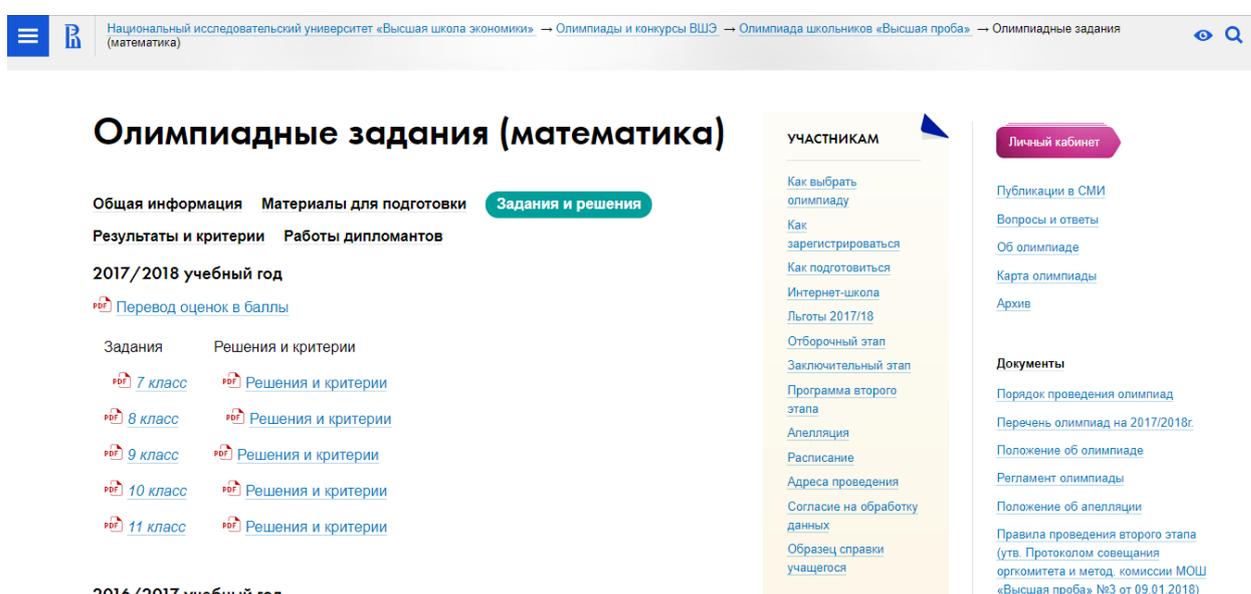


Рисунок 1.7. - Страница сайта olymp.hse.ru

Достоинства сайтов для тестирования:

- имеют большую базу тестов по различным дисциплинам;
- являются информационным центром, содержащим новости об образовании;
- многие сайты содержат форум или форумоподобный раздел сайта, созданный для общения пользователей.

Недостатки:

- необходим постоянный доступ в Интернет;

- использование сайтов для прохождения тестов ведется на персональном компьютере.

1.3 Результат анализа существующих аналогов

На основе обзора можно выделить следующие критерии для общего сравнения:

- 1) наличие более одной дисциплины;
- 2) наличие поиска;
- 3) разделение категорий тестов;
- 4) наличие профиля пользователя;
- 5) содержимое понятно новичку;
- 6) красочный и понятный интерфейс;
- 7) общение пользователей.

На основании этих критериев можно провести сравнение найденных аналогов (таблица 1.1).

Таблица 1.1. Сравнение найденных аналогов

Название	Кр.1	Кр.2	Кр.3	Кр.4	Кр.5	Кр.6	Кр.7
«Тест по математике»	-	-	-	-	+	-	-
«Я умник – Тест на образованность»	+	-	+	-	+	-	-
«Знаток»	+	-	+	-	+	+	-
Сайты, посвященные тестам	+	+	+	-	-	+	-

На основании сравнения становится понятно, что, чтобы достичь поставленную цель, разработка мобильного приложения – это лучший выбор.

Но, чтобы приложение было максимально полезным, в отличие от рассмотренных вариантов оно должно быть сделано на русском языке, с понятным новичку интерфейсом, возможностью выбора интересующей темы

тестов, профилем пользователя со статистикой. Общение пользователей не является необходимым функционалом для такого рода приложения.

Выполнив анализ приложений можно составить таблицу по нескольким показателям:

Таблица 1.2. Аналоги разрабатываемого приложения

Позиция в поиске	Название	Рейтинг (5 звезд)	Год публикации	Количество скачиваний
1	Тест по математике	3.8	2014	~100 000
2	Я умник – Тест на образованность	3.9	2014	~1 000 000
3	Знатор	4.1	2016	~100 000
4	Математейка математика - легко	4.4	2016	~100 000
5	Тесты для школы	3.7	2015	~100 000

Приложения данной категории в магазине Google Play Market за период 2-3 года набирают около 100 тысяч скачиваний, что говорит о том, что пользователей, интересующимися такими тестами, достаточно много. Это говорит об актуальности идеи по разработке программы для тестирования школьников по математике. Актуальность разрабатываемого приложения может повысить наличие в разработанном приложении тестов по Единому государственному экзамену и Обязательному государственному экзамену за прошедшие годы.

2 ВЫБОР СРЕДСТВ РАЗРАБОТКИ

2.1 Выбор операционной системы

Операционная система Android является наиболее популярной в мире и установлена на наибольшем количестве устройств (рисунок 2.1).

MONTH	ANDROID	IOS	WINDOWS PHONE	JAVA ME	SYMBIAN	OTHER
June, 2016	65.58%	27.24%	3.26%	1.81%	1.08%	1.03%
July, 2016	66.01%	27.84%	2.79%	1.44%	1.03%	0.88%
August, 2016	66.87%	27.20%	2.30%	1.62%	1.03%	0.98%
September, 2016	69.18%	25.02%	2.35%	1.43%	1.11%	0.91%
October, 2016	68.54%	25.78%	1.95%	1.54%	1.01%	1.18%
November, 2016	68.67%	25.71%	1.75%	1.57%	0.95%	1.34%
December, 2016	65.87%	29.52%	1.76%	1.44%	0.85%	0.57%
January, 2017	63.99%	32.03%	1.48%	1.14%	0.84%	0.51%
February, 2017	66.71%	29.55%	1.41%	1.09%	0.85%	0.40%
March, 2017	62.94%	33.39%	1.33%	0.99%	0.90%	0.44%
April, 2017	65.19%	31.06%	1.59%	0.95%	0.75%	0.46%
May, 2017	63.66%	33.03%	1.21%	0.90%	0.82%	0.38%
June, 2017	64.20%	32.92%	0.99%	0.82%	0.74%	0.33%

Рисунок 2.1. - Разделение рынка мобильных платформ
(июнь 2016 - июнь 2017 годов)

Учитывая эти данные, можно сказать что операционная система Android занимает большую часть рынка мобильных устройств.

После того, как выбрана операционная система, для которой будет разрабатываться приложение, необходимо определиться, для каких версий будет вестись разработка. В таблице 2.2 показано распределение пользователей по версиям Android.

Таблица 2.1. - Распределение пользователей Android по версиям ОС

Версия	Кодовое название	Уровень API	Процент пользователей
2.3.3-2.3.7	Gingerbread	10	0.4

4.0.3-4.0.4	Ice Cream Sandwich	15	0.5
4.1.x	Jelly Bean	16	1.9
4.2.x		17	2.9
4.3		18	0.8
4.4	KitKat	19	12.8
5.0	Lollipop	21	5.7
5.1		22	19.4
6.0	Marshmallow	23	28.6
7.0	Nougat	24	21.1
7.1		25	5.2
8.0	Oreo	26	0.5
8.1		27	0.2

Учитывая эти данные, можно прийти к выводу, что целесообразно установить минимальным порогом версию 5.0 (API 21).

2.2 Выбор интегрированной среды разработки

Для операционной системы Android существуют следующие основные среды разработки:

- Eclipse;
- IntelliJ IDEA;
- Android Studio.

2.2.1 Среда разработки Eclipse

Инструментом сборки проектов служит Ant (популярный инструмент для сборки ПО, написан на Java). Эта интегрированная среда разработки довольно легко настраивается, ее можно интегрировать с необходимыми для разработки SDK Android, Native Developer Kit (набор инструментов, позволяющий выполнить часть приложения используя такие языки программирования как C и C++) и машиной Java. Как только интеграция будет успешно выполнена, создать свой проект не составит труда. При создании

нового проекта для Android автоматически создается пустой «Hello World» проект, который можно даже запустить на девайсе пользователя или в эмуляторе (если такой создан). В качестве языка программирования Eclipse использует Java. Данная IDE очень простая в работе, имеет несложный интерфейс (рисунок 2.2), поэтому она подойдет новичкам.

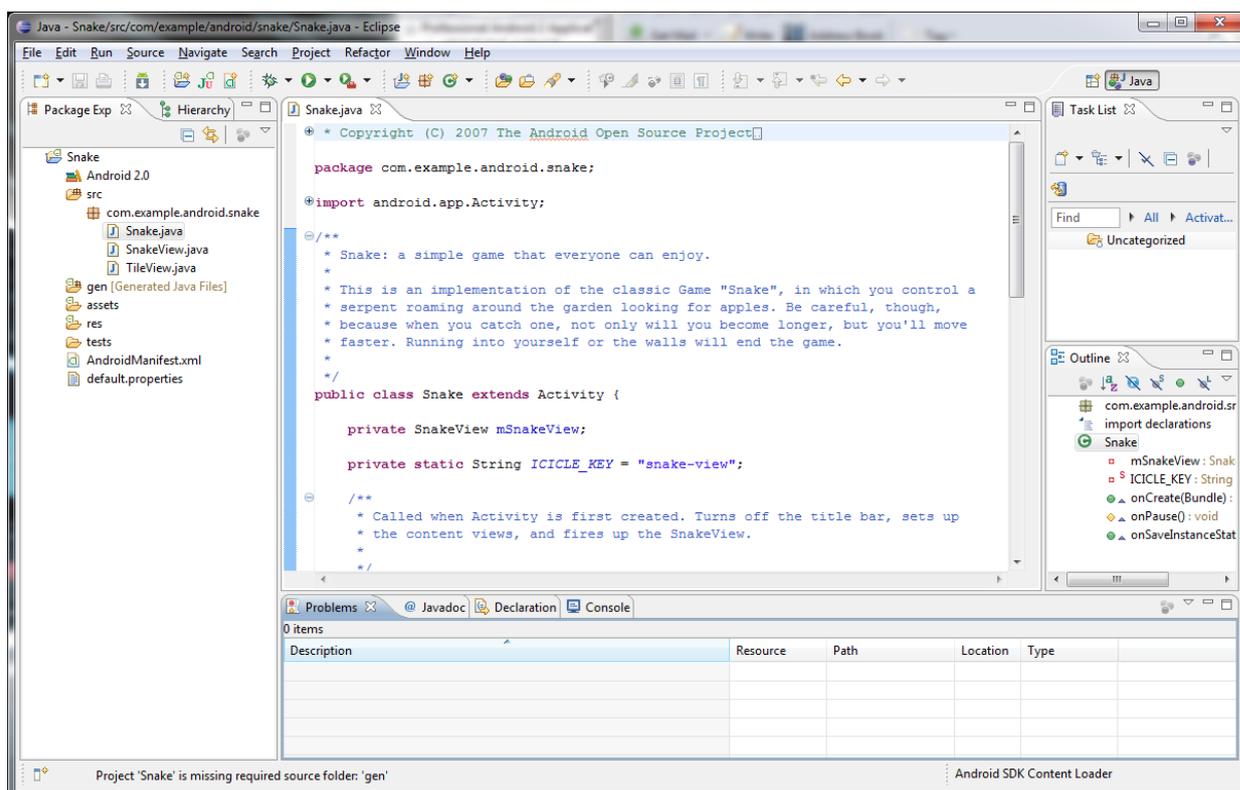


Рисунок 2.2. - Интерфейс среды разработки Eclipse

2.2.2 Среда разработки IntelliJ IDEA

Эта среда разработки является более серьезным инструментом. Продукты от JetBrains даже называются «Умными IDE (Integrated Development Environment)». Их продукты выполняют такие, на первый взгляд, элементарные функции как:

- автоматическое закрытие скобок после условия;
- групповые перезаписи методов;
- автоматическое создание шаблонных классов.

Элементарными эти функции на самом деле не являются и при долгой разработке очень облегчают жизнь разработчику. В данной IDE даже решается проблема «как назвать переменную». Например, в проекте есть базовый класс «Item», при создании из этих элементов массив, то среда разработки автоматически предложит назвать их «Items». Все это позволяет сэкономить много времени. Например, по статистике за год на одних только операциях 'rename' можно экономить около 120 часов времени.

Также, немаловажным плюсом IntelliJ IDEA является возможность смены темы оформления. Опять же, простая функция, которая сильно помогает разработчику. При бдительной и длительной работе светлый фон намного сильнее утомляет глаз, чем темно-серый (рисунок 2.3).

Однако, данный проект не является открытым. Хотя и это нельзя назвать минусом, ибо разработчики очень часто выпускают обновления с исправлениями ошибок, добавлениями новых функций и т.д.

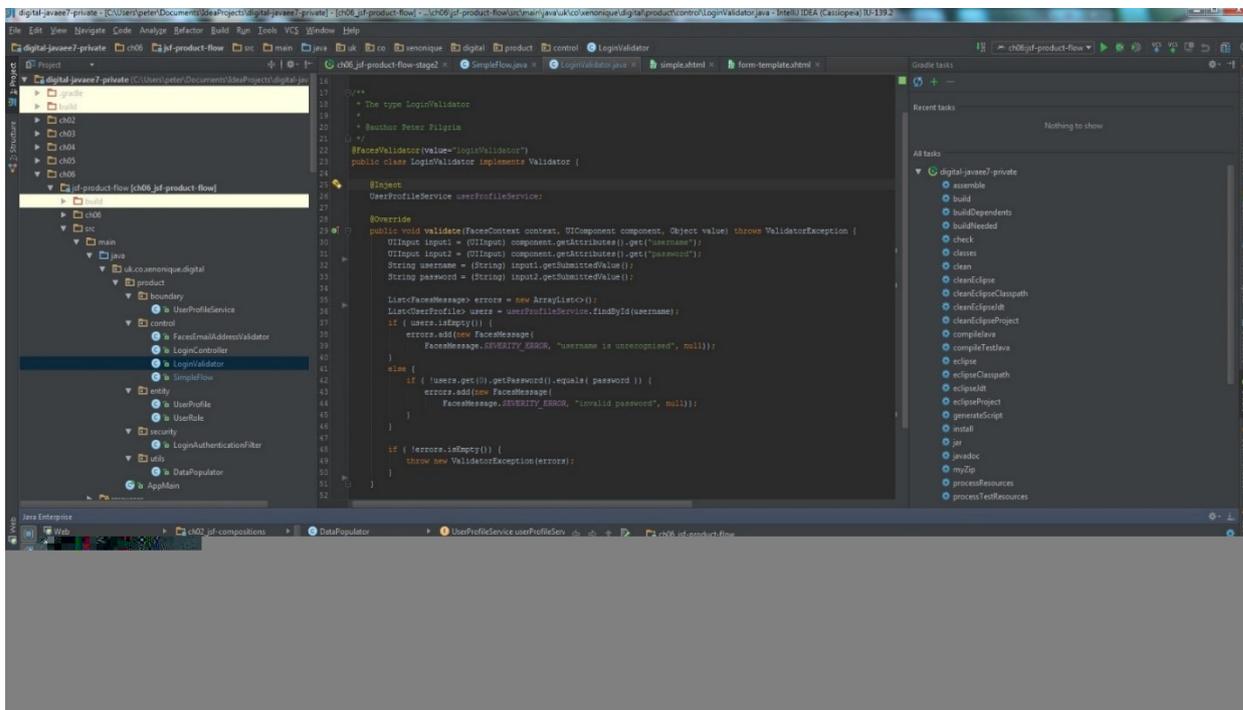


Рисунок 2.3. - Темно-серое оформление интерфейса IntelliJ IDEA

2.2.3 Среда разработки Android Studio

Android Studio – это результат сотрудничества JetBrains (разработчиков IntelliJ IDEA) и Google, который ориентирован на разработку приложений для операционной системы Android. По сути, это та же самая IDEA (рисунок 2.3), только здесь в качестве сборщика используется Gradle, а не ранее упомянутый Ant. Все внесенные в Android Studio изменения вскоре появляются в IDEA, и наоборот. Google рекомендует Android Studio для разработки приложений для своей системы.

Большим плюсом этой среды разработки является документация, которую предоставляет Google на сайте Android studio. Очень полезный материал для новичка.

После сравнения этих трех IDE, было принято решение использовать Android Studio. Она содержит все преимущества «умной» IntelliJ IDEA и имеет официальную поддержку разработчика Android.

Очень важно использовать последнюю версию этого приложения, и постоянно обновлять в процессе разработки, потому что Google добавляет поддержку новых версий Android.

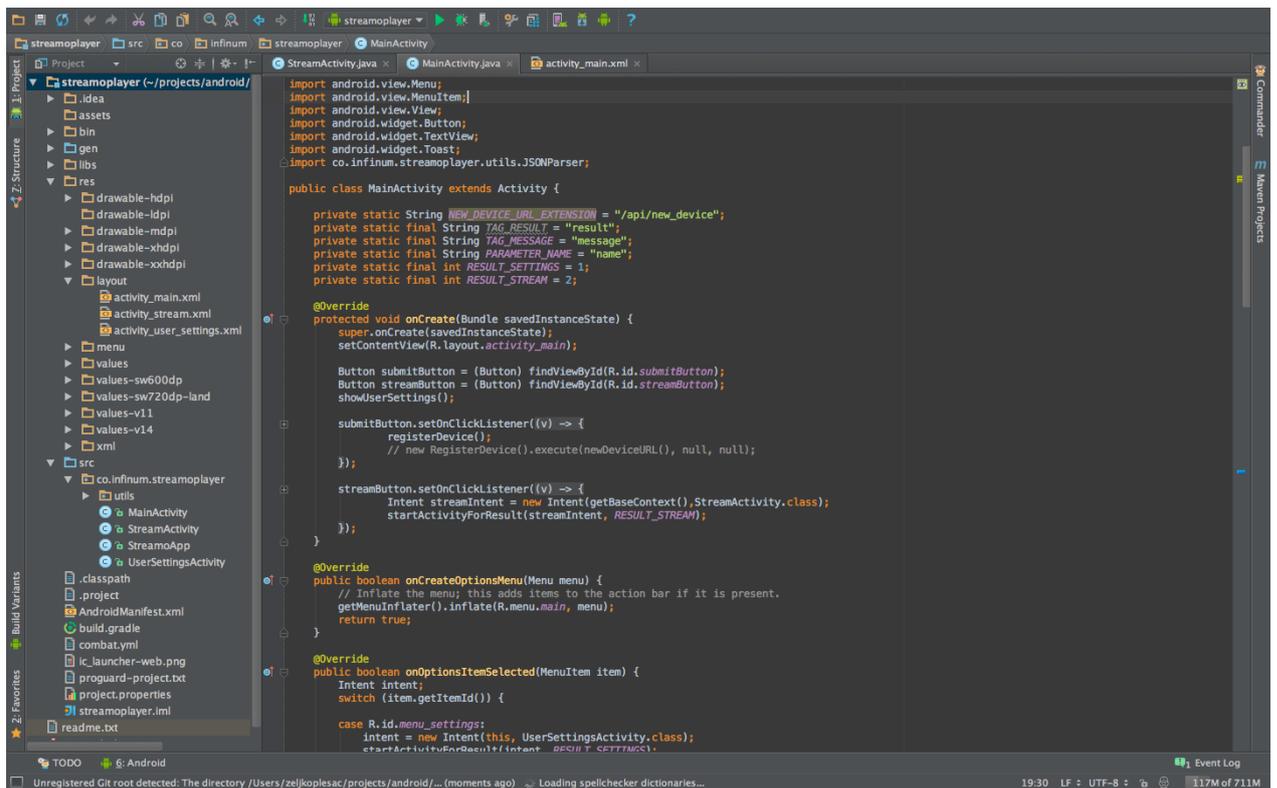


Рисунок 2.4. - Интерфейс Android Studio

3 РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОДГОТОВКИ ШКОЛЬНИКОВ К ОЛИМПИАДАМ

3.1 Формирование требований к разрабатываемому приложению

После проделанного обзора аналогов, выбора ОС и средств разработки, были сформированы требования к функциональному набору приложения:

- выбор категории тестов (Общие, олимпиады, ЕГЭ, ОГЭ);
- просмотр информации о выбранном тесте;
- возможность пропуска сложного вопроса;
- просмотр результата тестирования;
- запись результата и других статистических данных в память телефона;
- профиль пользователя с возможностью добавления фотографии пользователя и выводом статистических данных.

Необходимо сохранять интерфейс минималистическим и простым, чтобы люди, слабо знакомые с мобильными телефонами не запутались при работе с приложением.

3.2 Общая схема системы

Общая схема системы приложения представлена на рисунке 3.1.

Разработанное приложение состоит из 3 основных окон: «Информация», «Главное меню» и «Профиль пользователя». В окне «Главное меню» содержаться 4 вкладки с категориями тестов: «Общие тесты», «Олимпиады», «ЕГЭ» и «ОГЭ». В свою очередь, каждая категория включает в себя список тестов. При клике на элемент списка отображается окно «Тестирование».

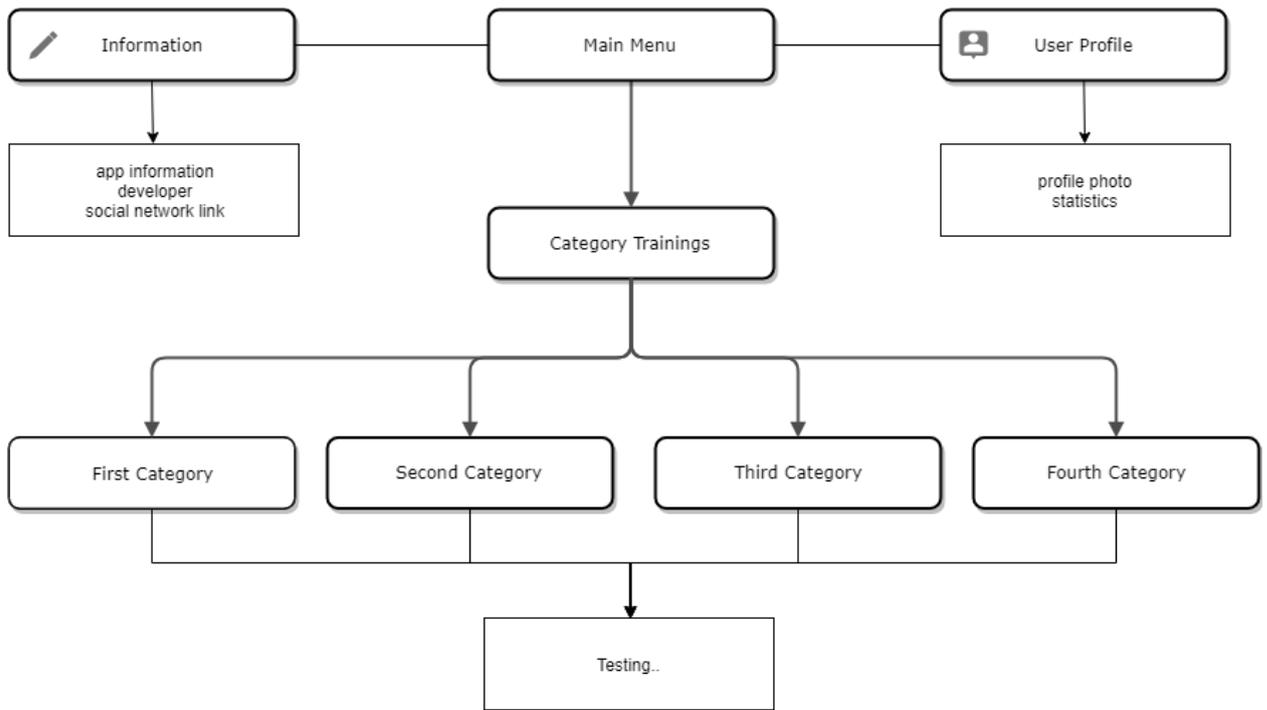


Рисунок 3.1. - Общая схема системы

3.3 Структура приложения

Разработанное приложение имеет 3 основных окна:

- окно главного меню;
- окно профиля пользователя;
- окно информации.

Детальный обзор окон и разделов разработанного приложения рассмотрен в разделе «Стадии и этапы разработки» Технического задания (Приложение Б).

3.4 Особенности языка java в разработке мобильного приложения

В приложении есть несколько классов. Главный классом и управляющим является MainActivity, который, в свою очередь, унаследован от AppCompatActivity. Данная Активность имеет свой ресурс визуализации

main_activity.xml [20]. В нем содержатся компоненты, необходимые для визуализации интерфейса [21 - 25].

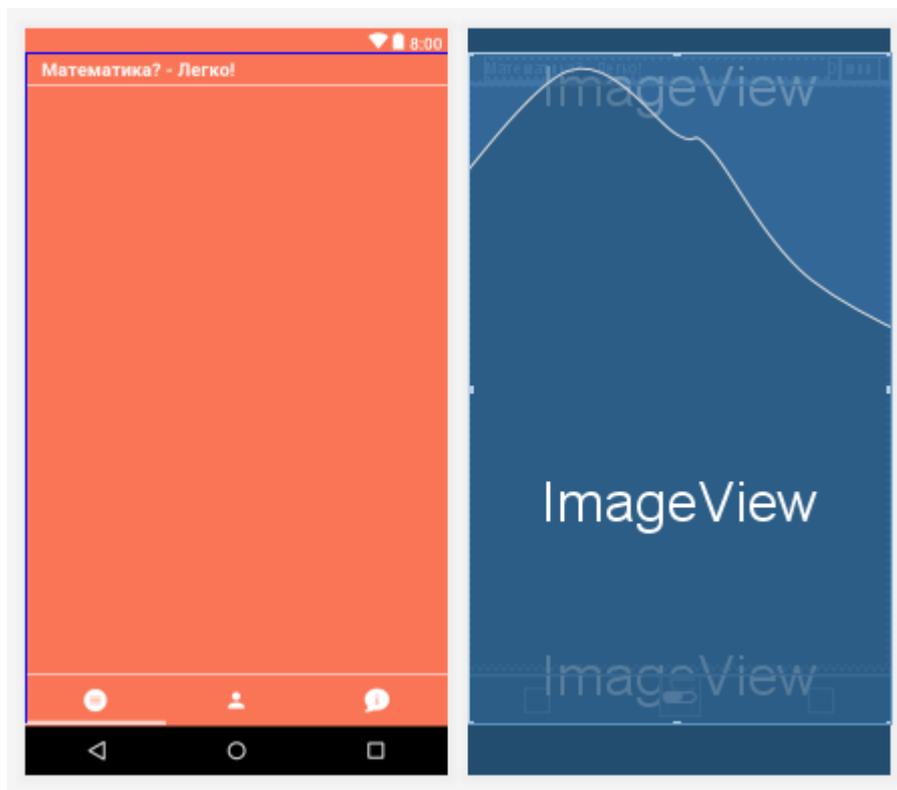


Рисунок 3.2. - Разметка MainActivity

В разметке main_activity.xml используются следующие компоненты:

1. `TextView` – компонент для вывода текста на экран, которому можно присвоить любой текст, форматировать его, изменять размер текста, цвет, стиль, шрифт.

2. `ImageView` – компонент для вывода изображения на экран. Изображение может выводиться как на фон компонента, так и на передний слой. Из основных свойств можно отметить задаваемый размер изображения в различных величинах (px, dp, pt, mm, in, sp), выравнивание изображения.

3. `TabLayout` – компонент для отображения вкладок как текстовых, так и с иконками. Переключение между вкладками происходит свайпом по горизонтали.

4. `NonSwipeableViewPager` – модифицированный компонент для отображения фрагментов. Напрямую взаимодействует с компонентом

TabLayout для переходов между фрагментами. Унаследован от компонента ViewPager

5. FrameLayout – слой, внутри которого помещаются компоненты и другие слои. Компоненты располагаются друг на друга

6. LinearLayout – слой, внутри которого помещаются компоненты и другие слои. Компоненты располагаются последовательно. Основными свойствами являются: смена ориентации, возможность задать сумму веса внутри располагающихся компонентов.

7. Toolbar – компонент, который обычно располагается в верху разметки, и служит для навигации по приложению [18].

8. ProgressBar – компонент, который служит для отображения прогресса загрузки информации.

Жизненный цикл приложения в Android жёстко контролируется системой и зависит от нужд пользователя, доступных ресурсов и т.д., например, пользователь хочет запустить браузер. Решение о запуске приложения принимает система. Хотя последнее слово и остаётся за системой, она подчиняется определённым заданным и логическим правилам, позволяющим определить, можно ли загрузить, приостановить приложение или прекратить его работу. Если в данный момент пользователь работает с определённым окном, система даёт приоритет соответствующему приложению. И наоборот, если окно невидимо и система решает, что работу приложения необходимо остановить, чтобы мобилизовать дополнительные ресурсы, будет прекращена работа приложения, имеющего более низкий приоритет. В Android ресурсы более ограничены, поэтому Android более жёстко контролирует работу приложений.

Активность или Фрагмент имеют свой жизненный цикл (рисунок 3.3).

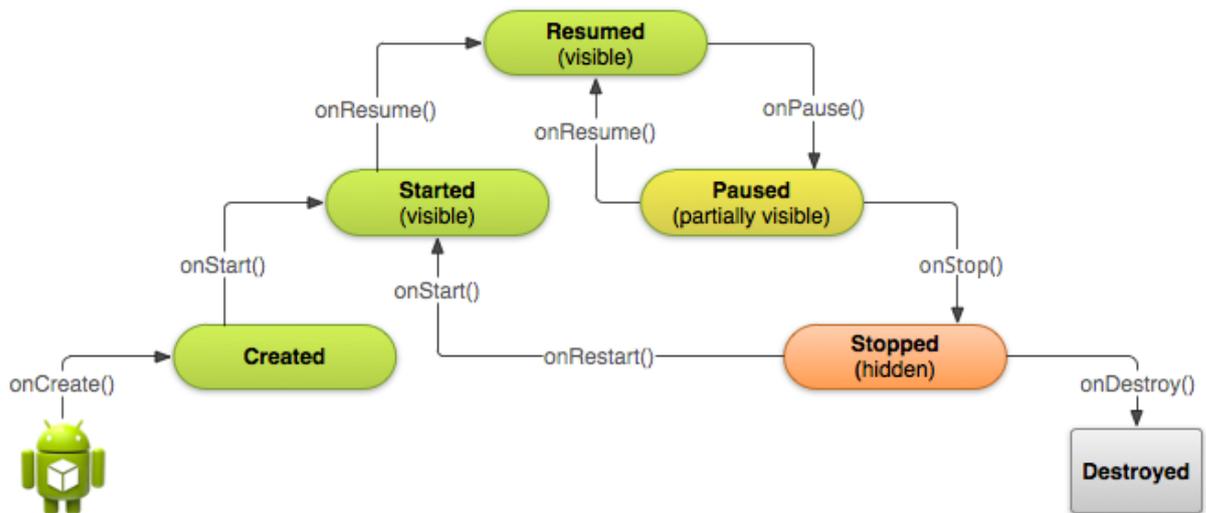


Рисунок 3.3. Жизненный цикл Activity

На рисунке 3.3 схематически изображены следующие методы жизненного цикла приложения [31]:

1. onCreate (Создание).
2. onStart (Запуск).
3. onRestart (Перезапуск).
4. onResume (Восстановление).
5. onPause (Пауза).
6. onStop (Остановка).
7. onDestroy (Уничтожение);

Метод onCreate() вызывается при создании или перезапуске активности. Система может запускать и останавливать текущие окна в зависимости от происходящих событий. Внутри данного метода настраивают статический интерфейс активности. Инициализирует статические данные активности, связывают данные со списками и т.д. Связывает с необходимыми данными и ресурсами. Задаёт внешний вид через метод setContentView().

За onCreate() всегда следует вызов onStart(), но перед onStart() не обязательно должен идти onCreate(), так как onStart() может вызываться и для возобновления работы приостановленного приложения (приложение

останавливается методом `onStop()`). При вызове `onStart()` окно еще не видно пользователю, но вскоре будет видно. Вызывается непосредственно перед тем, как активность становится видимой пользователю. Сопровождается вызовом метода `onResume()`, если активность получает передний план, или вызовом метода `onStop()`, если становится скрытой.

Метод `onResume()` вызывается после `onStart()`, даже когда окно работает в приоритетном режиме и пользователь может его наблюдать. В этот момент пользователь взаимодействует с созданным вами окном. Приложение получает монопольные ресурсы. Запускает воспроизведение анимации, аудио и видео. Также может вызываться после `onPause()`.

Когда пользователь решает перейти к работе с новым окном, система вызовет для прерываемого окна метод `onPause()`. По сути происходит свёртывание активности. Сохраняет незафиксированные данные. Деактивирует и выпускает монопольные ресурсы. Останавливает воспроизведение видео, аудио и анимацию. От `onPause()` можно перейти к вызову либо `onResume()`, либо `onStop()`.

Метод `onStop()` вызывается, когда окно становится невидимым для пользователя. Это может произойти при её уничтожении, или если была запущена другая активность (существующая или новая), перекрывающая окно текущей активности. Всегда сопровождает любой вызов метода `onRestart()`, если активность возвращается, чтобы взаимодействовать с пользователем, или метода `onDestroy()`, если эта активность уничтожается.

Если окно возвращается в приоритетный режим после вызова `onStop()`, то в этом случае вызывается метод `onRestart()`. Т.е. вызывается после того, как активность была остановлена и снова была запущена пользователем. Всегда сопровождается вызовом метода `onStart()`.

Метод вызывается по окончании работы активности, при вызове метода `finish()` или в случае, когда система уничтожает этот экземпляр активности для освобождения ресурсов. Эти два сценария уничтожения можно определить вызовом метода `isFinishing()`. Вызывается перед уничтожением активности.

Это последний запрос, который получает активность от системы. Если определённое окно находится в верхней позиции в стеке, но невидимо пользователю и система решает завершить это окно, вызывается метод `onDestroy()`. В этом случае метод удаляет все статические данные активности. Отдаёт все используемые ресурсы.

Стоит отметить, что у каждого компонента должны быть заданы свойства размера, так как они являются обязательными. Также, для использования этих компонентов в коде, необходимо задать имя. Работать с компонентами можно только в классе, которому они принадлежат, либо от имени этого класса.

Для того, чтобы использовать компонент в коде программы, его нужно объявить следующим образом [14]:

```
View view = findViewById(R.id.name),
```

где `View` - это тип компонента, `view` – название, `findViewById` – метод для нахождения компонента в разметке и `R.id.name` – уникальный идентификатор компонента.

Компоненты наследуются от суперкласса `View` и имеют ряд методов, из которых, для каждого компонента, существует обработчик нажатий `OnClickListener`. К примеру, обработчик нажатий на компонент `ImageView` будет выглядеть следующим образом:

```
ImageView image = findViewById(R.id.image);
image.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    }
});
```

Метод `onClick` отработает только тогда, когда управляющий класс зафиксирует нажатие пользователя по компоненту, которому присвоен обработчик. В методе указываются дальнейшие запланированные действия.

В разметке MainActivity входит компонент NonSwipeableViewPager – модифицированный компонент для отображения фрагментов. Напрямую взаимодействует с компонентом TabLayout для переходов между фрагментами. Унаследован от компонента ViewPager. Главное отличие NonSwipeableViewPager от его родителя ViewPager – отключение возможности переключения между фрагментами свайпом.

Следующий метод создает адаптер для отображения фрагментов. После добавления всех необходимых фрагментов, таких как FragmentMainMenu, FragmentProfile и FragmentInfo, адаптер присваивается компоненту ViewPager.

```
private void setupViewPager(ViewPager viewPager) {  
    adapter = new EventCountRight.ViewPagerMainAdapter  
(getSupportFragmentManager());  
    adapter.addFragment(new FragmentMainMenu());  
    adapter.addFragment(new FragmentProfile());  
    adapter.addFragment(new FragmentInfo());  
    viewPager.setAdapter(adapter);  
}
```

При запуске класса MainActivity первым делом вызывается метод onCreate, в котором, с помощью метода setContentView, активности присваивается файл разметки. Также, в onCreate компоненты присваивают свои уникальные идентификаторы.

Следующий метод onStart, в котором используется полезный класс под названием Hawk. Этот класс наследуется от стандартного класса SharedPreferences и используется для работы с памятью телефона. В данной программе его предназначение для сохранения статистических данных в память телефон и последующего считывания с выводом на экран.

В компоненте ViewPager, после написания предыдущего кода, находятся ссылки на другие фрагменты. Сами фрагменты наследуются от android.app.Fragment. Существует подклассы фрагментов: ListFragment, DialogFragment, PreferenceFragment, WebViewFragment и др. Фрагмент (класс

Fragment) представляет поведение или часть пользовательского интерфейса в операции (класс Activity). Разработчик может объединить несколько фрагментов в одну операцию для построения многопанельного пользовательского интерфейса и повторного использования фрагмента в нескольких операциях. Фрагмент можно рассматривать как модульную часть операции. Такая часть имеет свой жизненный цикл и самостоятельно обрабатывает события ввода. Кроме того, ее можно добавить или удалить непосредственно во время выполнения операции. Фрагмент всегда должен быть встроен в операцию, и на его жизненный цикл напрямую влияет жизненный цикл операции. Например, когда операция приостановлена, в том же состоянии находятся и все фрагменты внутри нее, а когда операция уничтожается, уничтожаются и все фрагменты. Однако пока операция выполняется (это соответствует состоянию возобновлена жизненного цикла), можно манипулировать каждым фрагментом независимо, например, добавлять или удалять их.

Во ViewPager находятся ссылки на следующие фрагменты:

1. MainMenuFragment.
2. ProfileFragment.
3. FragmentInfo.

Главное окно входа показано на рисунке 3.4.

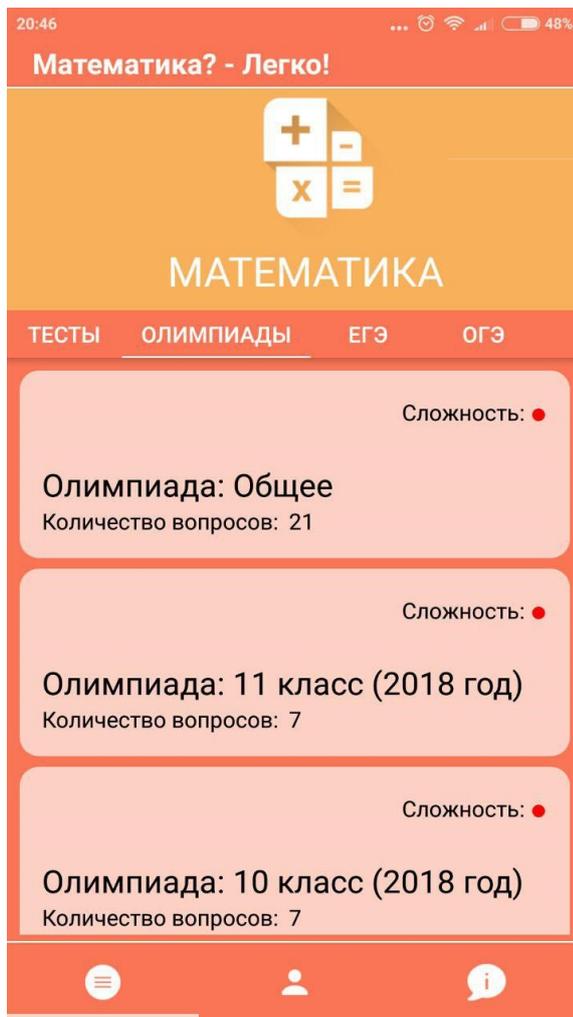


Рисунок 3.4 - Окно главного меню

Внутри фрагмента MainMenu, также, как и в MainActivity, находится ViewPager и TabLayout, с помощью которых осуществляется переход между категориями тестов, за которые отвечает фрагмент FragmentCategoryTraining.

FragmentCategoryTraining содержит список вертикально расположенных элементов категорий, а именно самих тестов. На рисунке 3.5 изображено окно категории «Тесты»:

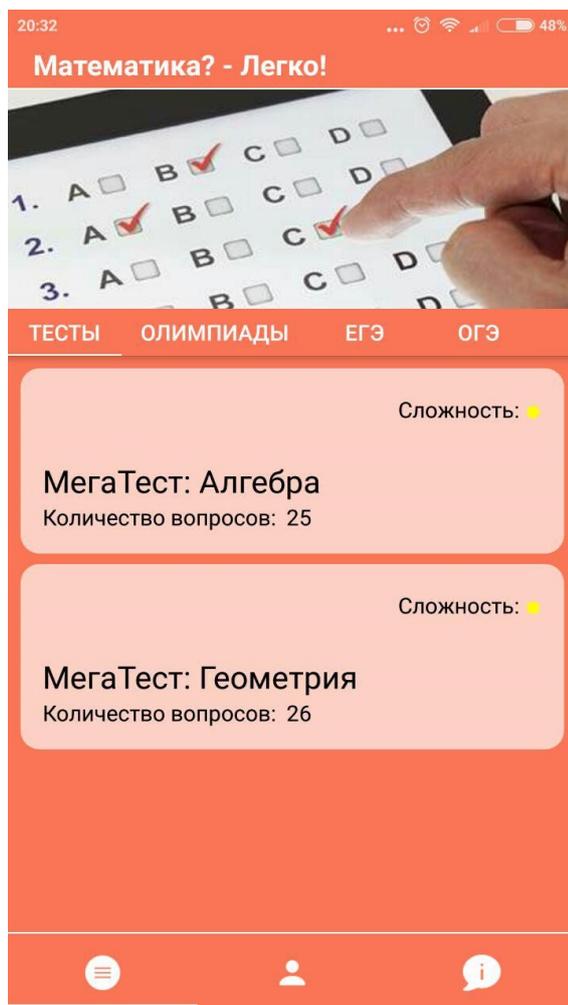


Рисунок 3.5. Окно категории «Тесты»

В этом разделе находятся тесты общей категории. Внутри программы отображение списка осуществляется путем добавления в файл разметки фрагмента компонента RecyclerView [31]. RecyclerView – это аналог стандартного ListView. Для того чтобы отобразить компоненты списка, необходимо создать отдельную разметку, как показано на рисунке 3.6.

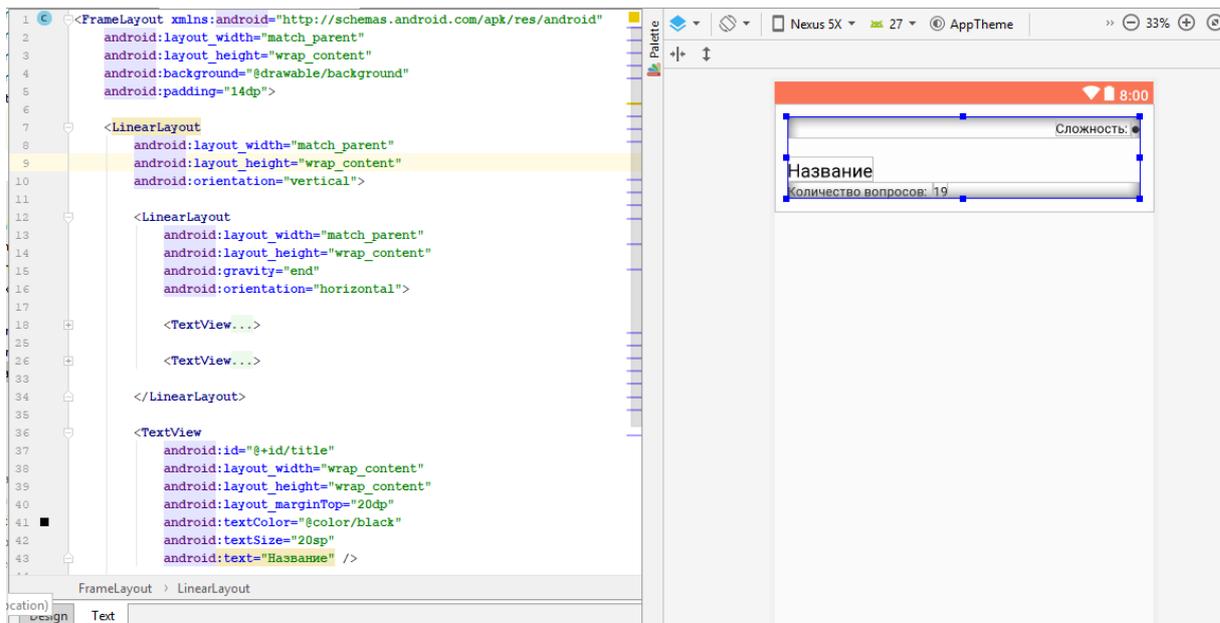


Рисунок 3.6. - Часть кода разметки элемента списка категорий тестов и визуальное отображение компонентов на экране эмулятора ОС Android

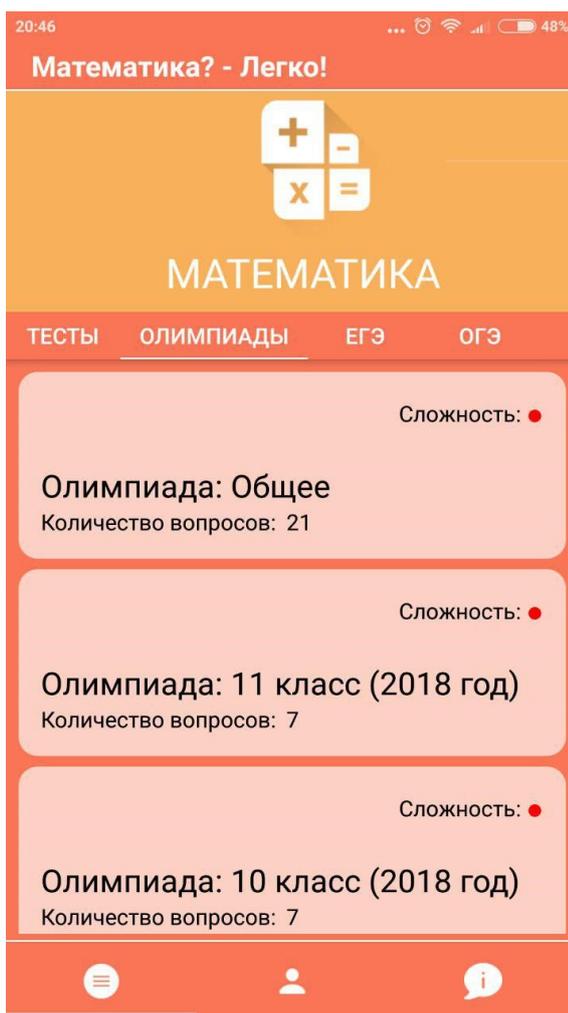


Рисунок 3.7. - Окно категории «Олимпиады»

У каждого компонента RecyclerView должен быть свой Adapter и Holder. Holder нужен для того чтобы объявить все используемые в элементе списка компоненты, задать им имена и присвоить уникальные идентификаторы. Также создаются Get-методы, для доступа к объявленным компонентам вне Holder'a. В свою очередь, Adapter нужен для формирования самого списка тестов. На рисунке 3.8 изображен список тестов категории «ЕГЭ».

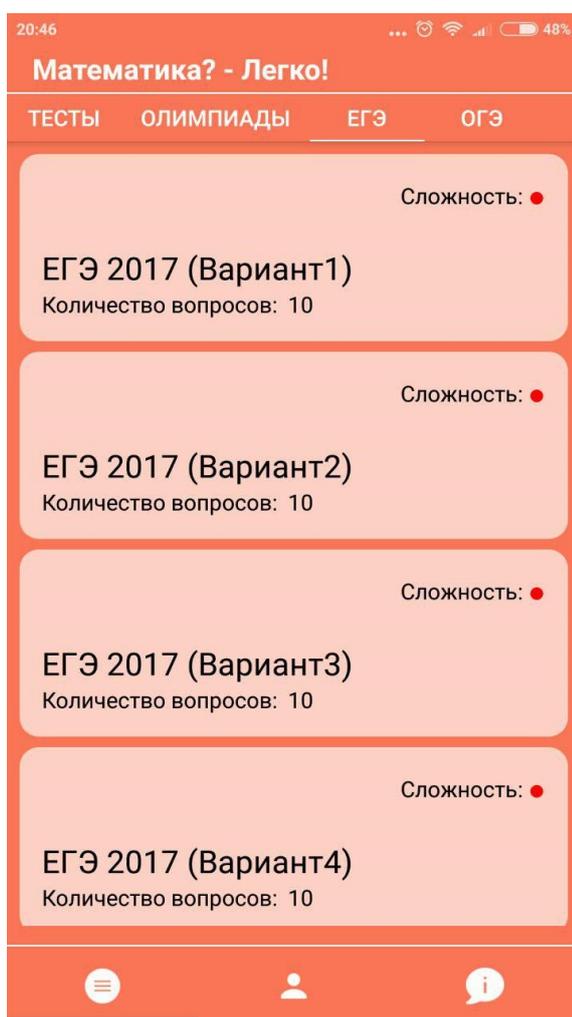


Рисунок 3.8. - Окно категории «ЕГЭ»

Добавлены несколько тестов для подготовки к ЕГЭ по математике на основе тестов за предыдущий год. В элементе списка в правом верхнем углу располагается индикатор сложности теста. Он зависит от номера школьного

класса, для которого, собственно, предназначается тест. На рисунке 3.9 изображен список тестов категории «ОГЭ».

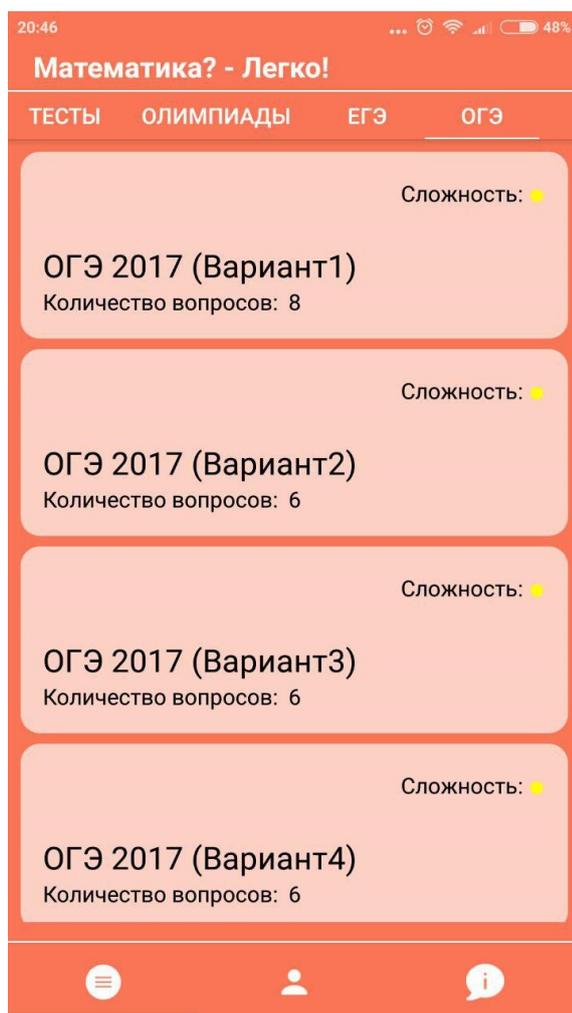


Рисунок 3.9. - Окно категории «ОГЭ»

Добавлены несколько тестов для подготовки к ОГЭ по математике на основе тестов за предыдущий год.

На следующем рисунке 3.10 показана разметка фрагмента FragmentTraining.

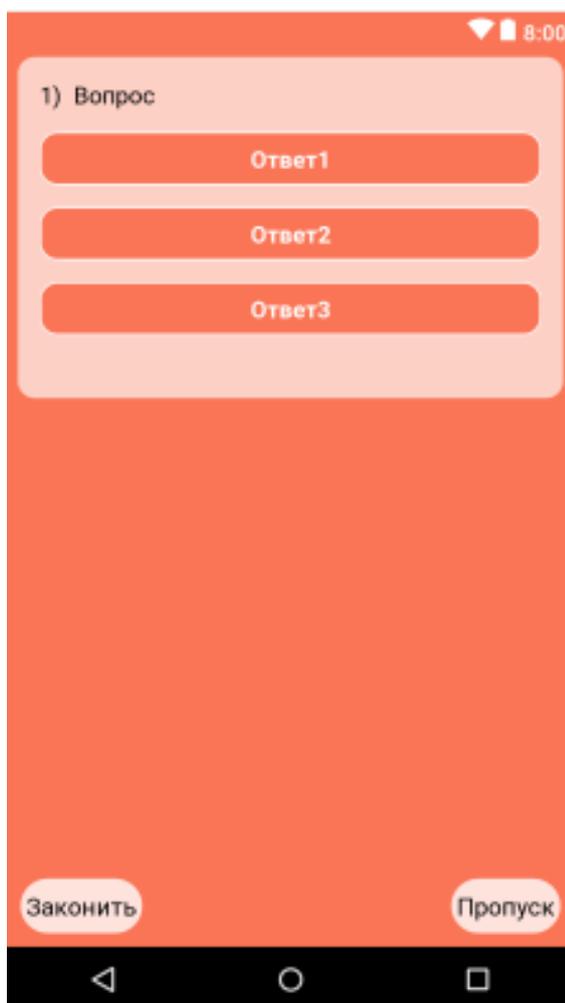


Рисунок 3.10. - Окно прохождения теста

В файле разметки располагаются компоненты: `FrameLayout`, `LinearLayout` (orientation - vertical), `ImageView`, `TextView`, `Button`.

Стоит обратить внимание на компонент `ImageView`, хоть его и не видно, но он располагается между вопросом и ответами. За видимость компонента отвечает свойство `View.Visibility`, параметрами которого являются: `visible`, `invisible` и `gone`. `Visible` присвоится компоненту, который должен оставаться в видимости для пользователя, `invisible` – это полная противоположность `visible`. `Gone` скрывает компонент с зоны видимости, создавая эффект замещения.

На следующем рисунке 3.11 показан макет профиля пользователя.

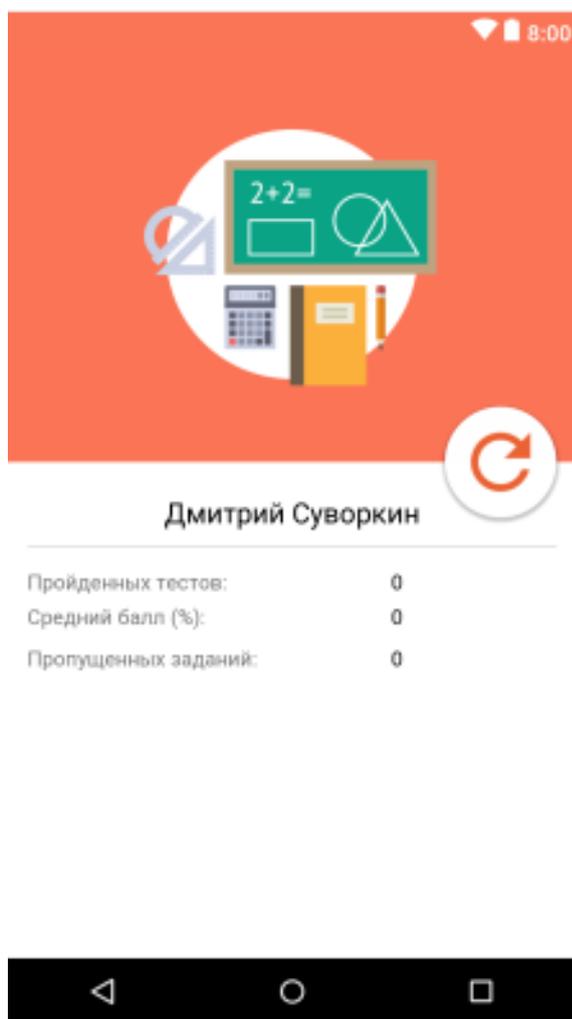


Рисунок 3.11. - Макет профиля пользователя

Раздел «Профиль пользователя» (FragmentProfile) включает в себя область для фотографии пользователя и список статистических данных. При клике в области для фото пользователя, в новом окне открывается галерея смартфона, в которой необходимо выбрать фото для загрузки. После выбора фото окно галереи закрывается. Далее, фото загружается в свою область и сохраняется в память телефона, для дальнейшей автоматической загрузки при перезапуске приложения. Статистические данные включают в себя показатель количества пройденных тестов, средний процент прохождения теста и количество использованных пропусков при прохождении теста.

Разметка раздела содержит следующие компоненты:

1. FrameLayout.

2. `LinearLayout`.
3. `ImageView`.
4. `TextView`.
5. `CoordinatorLayout` [19].
6. `ConstrainLayout`.
7. `FloatingActionButton`.

`ConstrainLayout` – компонент, позволяющий размещать компоненты относительно других с использованием различных видов анимации. В данном примере, при прокрутке списка статистических данных вниз – окно изображения профиля пользователя уменьшается до тех пор, пока не выйдет за рамки области видимости. А компонент с именем профиля двигается вверх и упирается в верхнюю границу. `FloatingActionButton` – компонент, наследуемый от `View.Button`. Без каких-либо манипуляций с кодом может выполнять некоторые действие, к примеру, уменьшаться. В обработчике нажатий на кнопку находится определенный код для обновления статистических данных.

Следующий раздел – «Информация», который содержит информацию об авторе разрабатываемого приложения, ссылки на сайт института и университета в картинках, а также, ссылки на социальные сети разработчика (рисунок 3.12).



Рисунок 3.12. - Раздел «информация»

Разметка раздела содержит несколько компонентов: `ImageView`, `TextView` и `LinearLayout`. `ImageView` хранят изображения, `TextView` текстовый надписи, а `LinearLayout` эти компоненты в горизонтальной и вертикальной ориентации соответственно.

4 ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ

Поскольку в приложении отсутствуют как таковые входные данные, то не может быть ошибок ввода неверных данных. Также, ввиду отсутствия подключения к каким-либо удаленным серверам, тестирование подключения и нагрузки сервера тоже не нужно.

В данном приложении необходимо протестировать работу навигации по разделам приложения, по категориям тестов, прохождения тестов и корректность вывода статистических данных в Профиле пользователя с загрузкой и выводом изображения профиля пользователя в поле, а также проверить требование к надежности из Технического задания (Приложение Б).

Тестирование проводилось на смартфоне Xiaomi Redmi Note 3 Pro со следующими техническими и аппаратными характеристиками:

1. Размер экрана: 5.5 дюйм;
2. Разрешение экрана: 1920*1080 px;
3. Процессор: Qualcomm Snapdragon 650 MSM8956, 1800 МГц
4. Количество ядер: 6;
5. Видеопроцессор: Adreno 510;
6. Объем встроенной памяти: 32 Гб;
7. Объем оперативной памяти: 3 Гб;
8. Операционная система: Android;
9. Версия ОС: 6.1.

5.1 Тестирование навигации

Как видно на рисунке 4.1 и рисунке 4.2 окна меняются в зависимости от выбранной свайпом категории тестов.

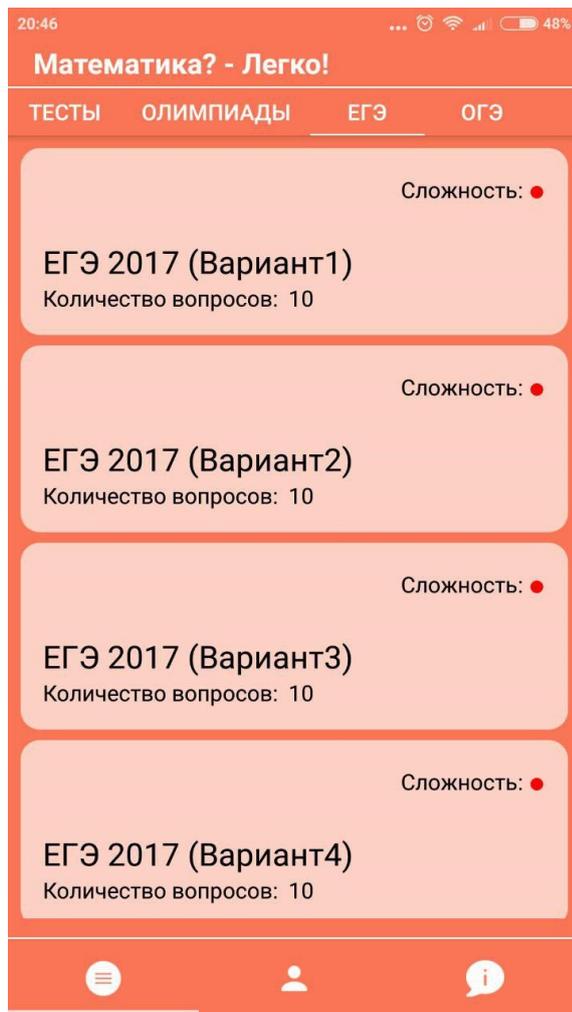


Рисунок 4.1. - Окно категории «ЕГЭ»

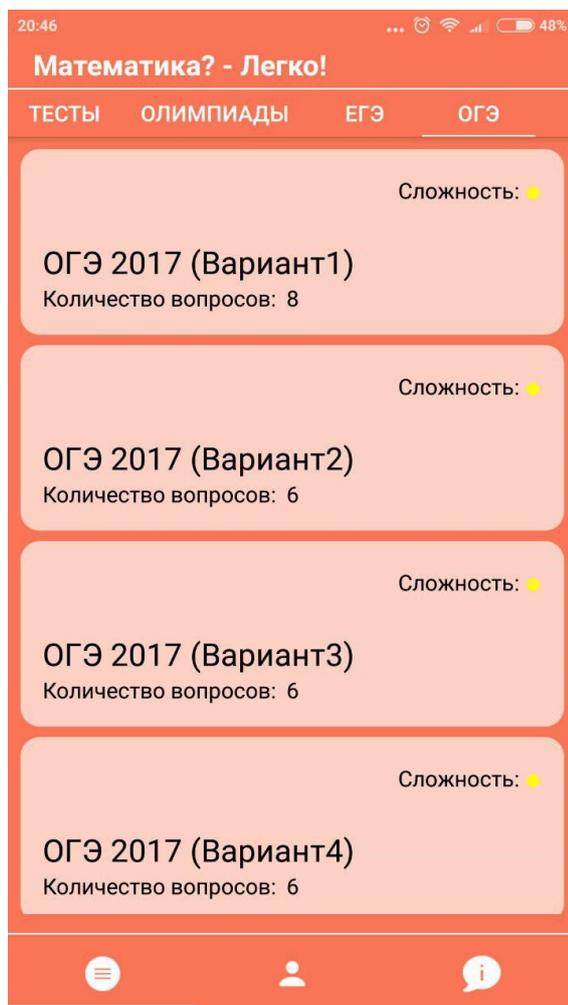


Рисунок 4.2. - Окно категории «ОГЭ»

Также необходимо протестировать навигацию между разделами приложения. На рисунках 4.3 – 4.5 зафиксированы переходы между разделами «Меню», «Профиль пользователя» и «Информация».

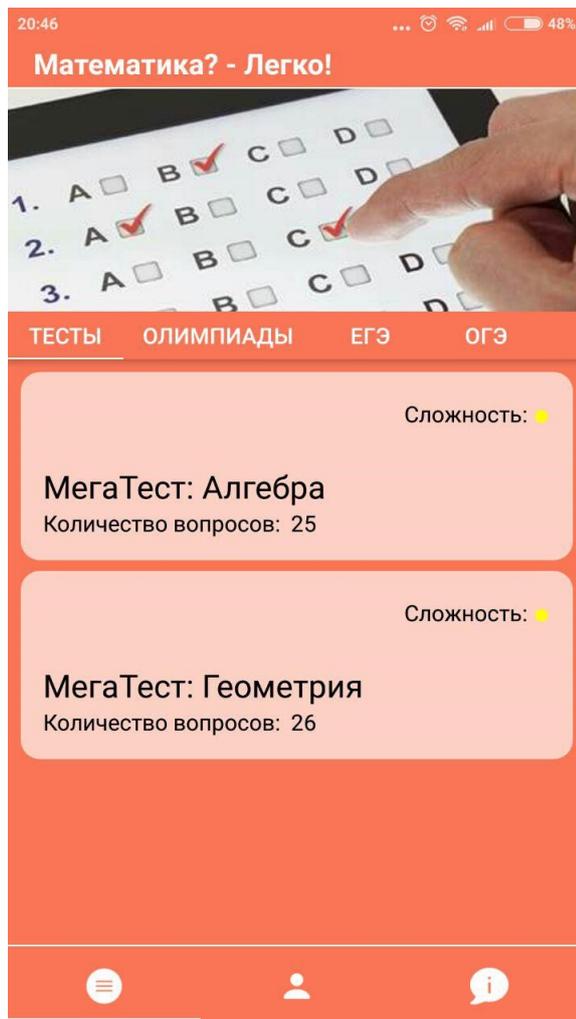


Рисунок 4.3. - Раздел «Меню»

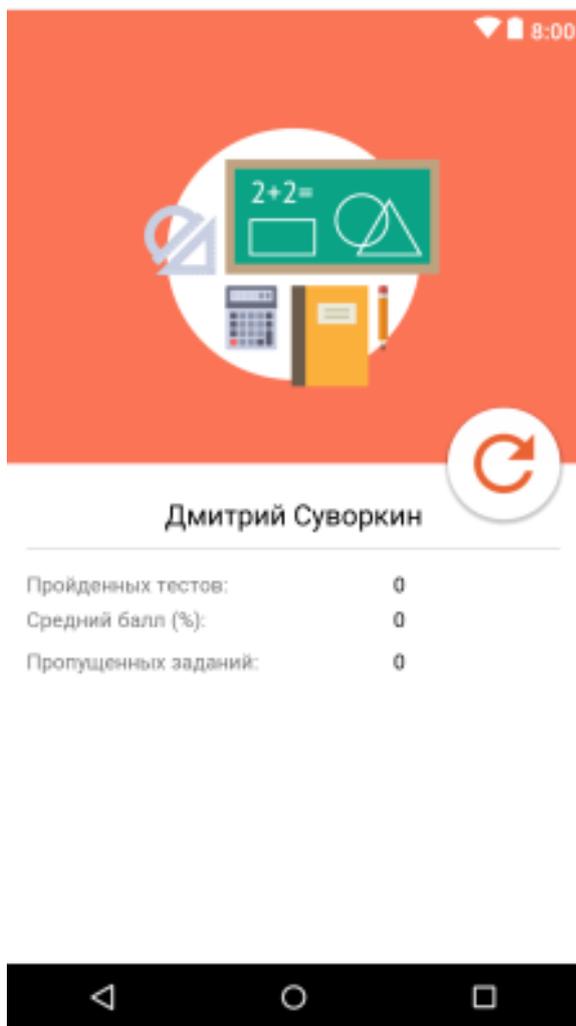


Рисунок 4.4. - Раздел «Профиль пользователя»



Рисунок 4.5. Раздел «Информация»

Внизу экрана имеются иконки разделов, описанных выше. Горизонтальный индикатор указывает на открытую вкладку.

4.2 Тестирование защиты от случайного выхода

Защита от случайного выхода срабатывает в двух случаях: при попытке выхода из области прохождения теста и при выходе из самого приложения.

На рисунках 4.6 и 4.7 изображено окно с подтверждением выхода.

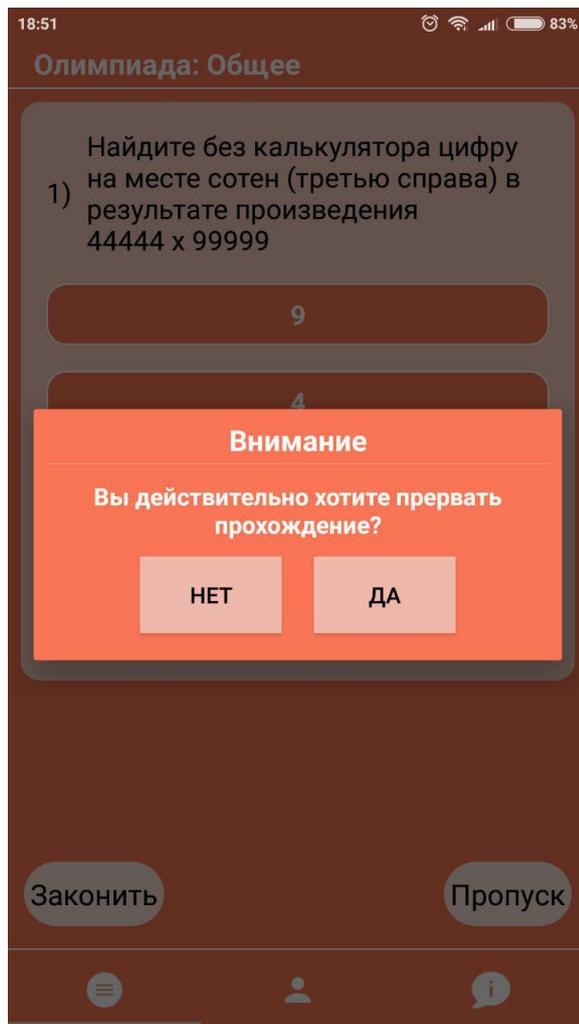


Рисунок 4.6. - Окно подтверждения прерывания прохождения теста

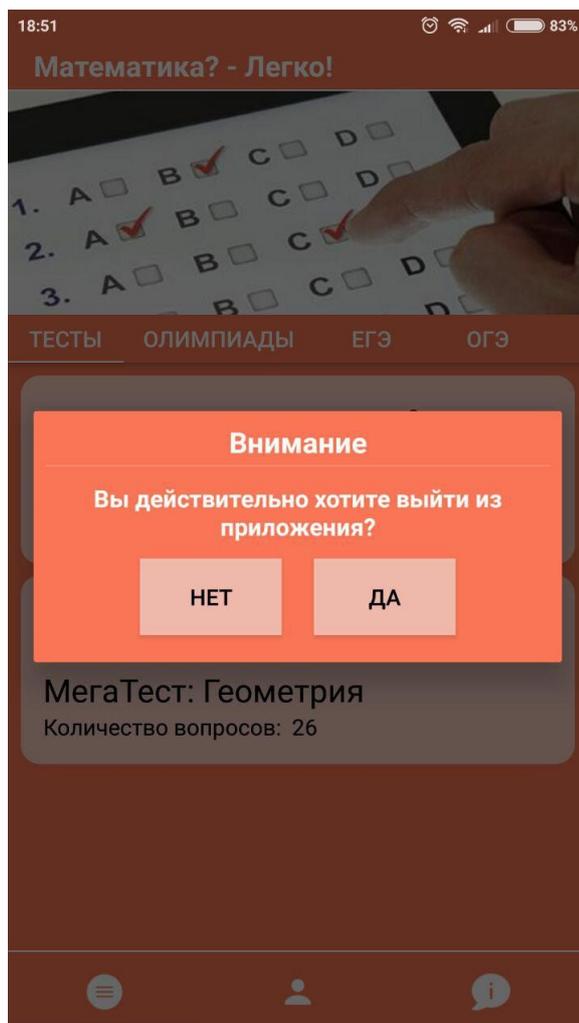


Рисунок 4.7. - Окно подтверждения выхода из приложения

При клике на соответствующую кнопку в окне, будет произведен выход из приложения или теста, либо сворачивания окна подтверждения. При клике не в области окна засчитывается клик на кнопку «Нет».

4.3 Тестирование прохождения тестов

Для тестирования прохождения был выбран тест «Олимпиада: 7 класс (2018 год)» категории «Олимпиады» раздела приложения «Меню».

На следующем рисунке изображено окно первого вопроса с ответами.

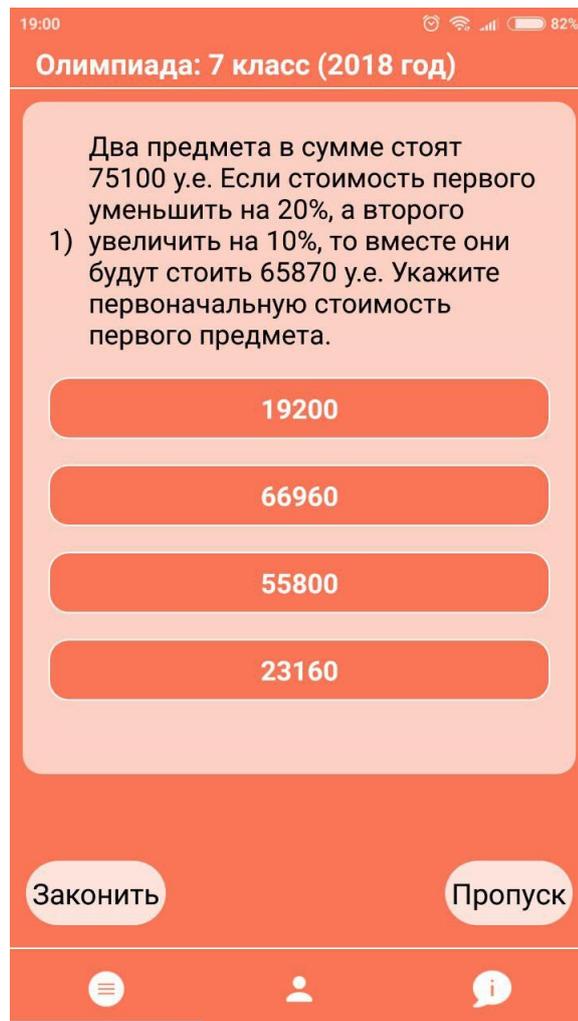


Рисунок 4.8. - Окно с первым вопросом к тесту «Олимпиада: 7 класс (2018 год)» категории «Олимпиады»

Как видно на рисунке 4.8, при открытии тестов в верхнюю область приложения (Toolbar) выводится название самого теста.

В одном из следующих заданий присутствует изображения. Окно с вопросом показано на рисунке 4.9.

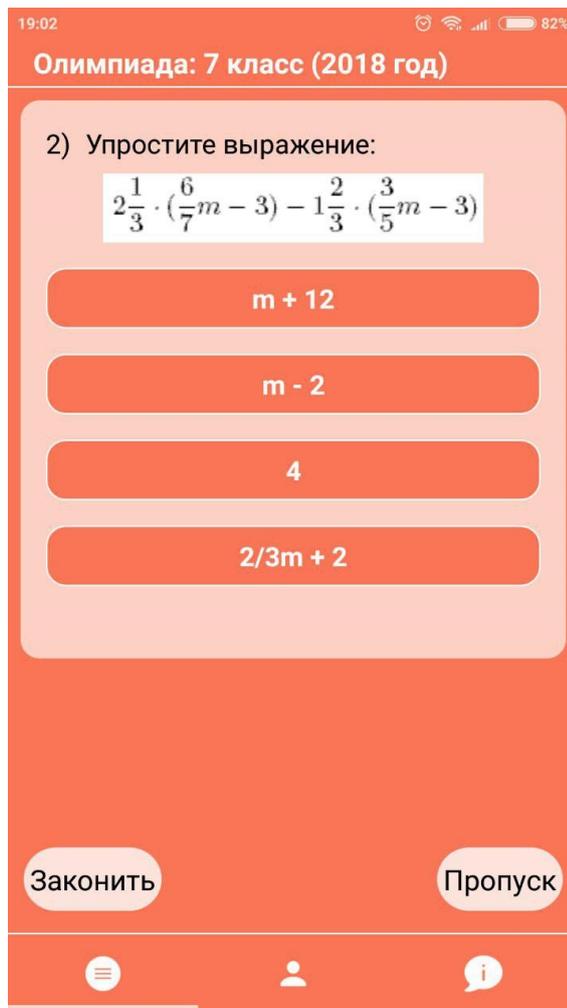


Рисунок 4.9. - Окно со вторым вопросом к тесту «Олимпиада: 7 класс (2018 год)» категории «Олимпиады» с картинкой

Завершить прохождение теста можно двумя способами – кликом на кнопку «Завершить» или ответить на все вопросы теста. При клике на кнопку «Завершить» тестирование завершается и на экран выводится окно с результатами тестирования. При окончании ответа на все вопросы теста результирующее окно открывается автоматически.

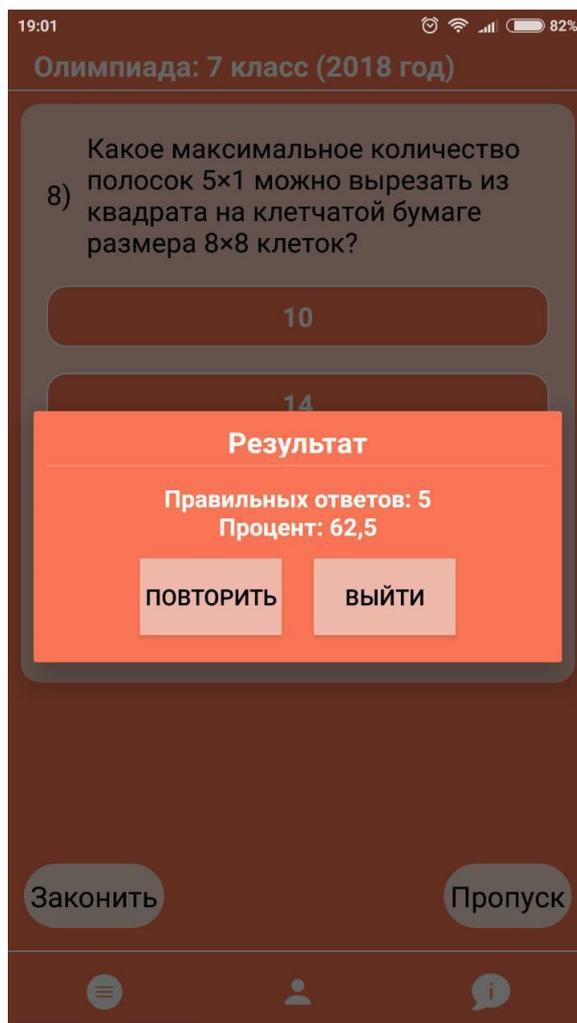


Рисунок 4.10. - Окно с результатом тестирования

В окне с результатом тестирования можно выйти в главное меню при клике на кнопку «Выйти» и повторить тест кликом на кнопку «Повторить».

4.3 Тестирование работы профиля пользователя

В разделе «Профиль пользователя» тестированию подвергнется механизм смены изображения профиля пользователя и вывод статистических данных.

При клике на текущее изображение профиля пользователя произойдет открытие Галереи устройства со списком всех изображений.

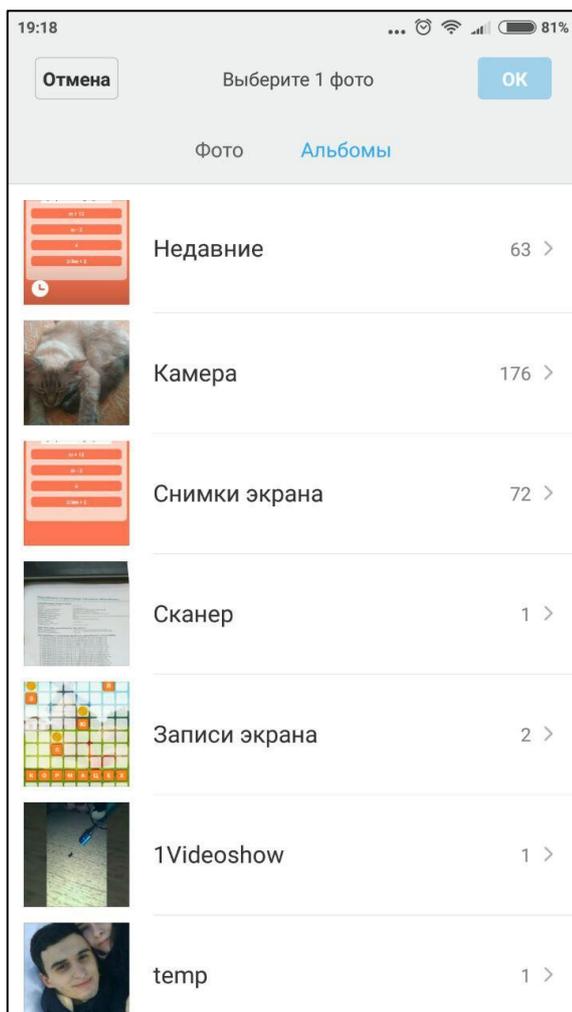


Рисунок 4.11. - Окно галереи устройства

После выбора подходящего изображения и клика кнопки «Ок» деятельность возвращается к приложению. Результат загрузки изображения показан на рисунке 4.12.

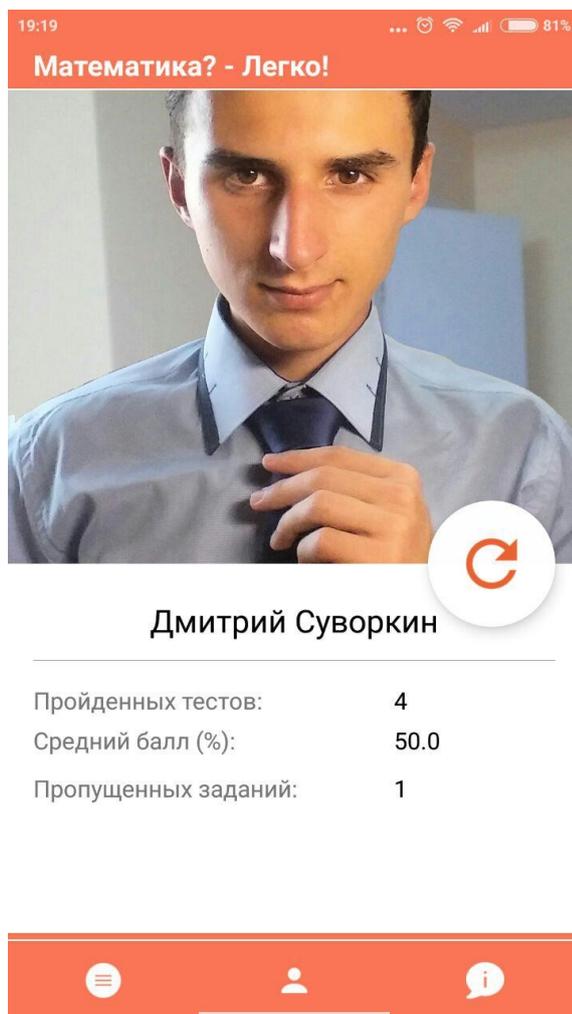


Рисунок 4.12. - Результирующее окно загрузки фото профиля пользователя

На рисунке 4.12 также изображен вывод и подсчет статистических данных. Можно понять, что количество пройденных тестов равно четырем, средний балл прохождения тестов – 50 процентов и количество пропущенных заданий – одно.

ЗАКЛЮЧЕНИЕ

В ходе работы над выпускной квалификационной работой были получены следующие результаты:

- проведенный обзор аналогов показал, что мобильное приложение – это наилучший вариант для достижения поставленной цели. Разработаны технические требования к приложению;
- в результате оценки количества устройств с различными операционными системами на рынке, была выделена наиболее популярная версия;
- в результате ознакомления со средствами разработки для ОС Android, была выбрана Android Studio IDE;
- с помощью Android Studio было разработано приложение, удовлетворяющее требованиям технического задания;
- приложение было протестировано на работоспособность, также, было показано, что все элементы приложения работают в штатном режиме;

Все поставленные задачи в работе были выполнены, разработано приложение для прохождения тестирования по математике для операционной системы Android.

Проектируемое приложение создано для людей, желающих узнать свой уровень знаний по дисциплине «Математика». Интерфейс максимально простой, не нагружен лишними изображениями или рекламой, чтобы пользователи, которые плохо умеют пользоваться мобильными устройствами не растерялись при использовании программы. Также реализован простой и удобный механизм перехода между разделами приложения и категориями тестов.

Как и любой другой продукт, разработанную программу можно совершенствовать, вводя новые функции и улучшая работу существующих. В сравнении с аналогами, единственный критерий, по которому приложение им

уступает – это возможность общения. Поэтому, в будущем можно ввести возможность пользователям оставлять комментарии к тестам и ввести новый раздел «Форум» для общения пользователей. Помимо своей основной функции – общения, этот функционал может служить, также, хорошим средством обратной связи с разработчиком.

Дальнейшим этапом развития приложения может стать выпуск версии для операционной системы iOS, так как это вторая операционная система на рынке.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Google Inc. Google Play Store. Главная страница магазина приложений. URL: <https://play.google.com/store/apps> (дата обращения 16.04.2018).
2. Idc.com. Smartphone OS Market Share. URL: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp/> (дата обращения 11.02.2016).
3. Голощапов А.Л. Google Android. Создание приложений для смартфонов и планшетных ПК. 2-е изд., перераб. и доп. – Спб.: БХВ-Петербург, 2014.
4. Андрей Солнцев. «Почему IDEA лучше Eclipse». URL: <https://habrahabr.ru/post/112749/> (дата обращения 02.05.2016).
5. Wikipidea. «Android Studio». URL: https://ru.wikipedia.org/wiki/Android_Studio (дата обращения 02.05 2016).
6. Александр Климов. «Разработка под Android». URL: <http://developer.alexanderklimov.ru/android/> (дата обращения 17.04.2016)
7. Олимпиадные задания (математика) – Олимпиада школьников «Высшая проба» – Национальный исследовательский университет «Высшая школа экономики» URL: <https://olymp.hse.ru/mmo/tasks-math>
8. Тесты по математике по разделу: Элементарная математика URL: testmat.ru/mat_tem.php?tem=1
9. Вариант 1. Задание 7. ЕГЭ 2017 Математика, И.В. Ященко. 36 вариантов. Решение URL: self-edu.ru/ege2017_36.php?id=1_7
10. Статистика версий Android за январь 2018 года — Новости на FullHub URL: <https://fullhub.ru/news/android-stat-jan-2018-98606.html>
11. Draw.io URL: <https://startpack.ru/application/draw-io>
12. Фрагменты | Android Developers URL: <https://developer.android.com/guide/components/fragments?hl=RU>

13. GitHub - gjiazhe/PanoramaImageView: An imageView can auto scroll with device rotating. URL: <https://github.com/gjiazhe/PanoramaImageView>
14. Android: Доступ к View внутри фрагмента URL: developer.alexanderklimov.ru/android/theory/fragment-view.php
15. Обновление данных в ViewPager при перелистывании вкладок - Программирование Android – Киберфорум URL: www.cyberforum.ru/android-dev/thread1977451.html
16. Android Arsenal URL: <https://android-arsenal.com>
17. Drawable URL: <https://android-dev-examples.blogspot.ru/2014/09/android-selector-drawable.html>
18. Bottom navigation - Components - Material Design URL: <https://material.io/guidelines/components/bottom-navigation.html#bottom-navigation-specs>
19. Mastering the Coordinator Layout · Saúl Molinero URL: saulmm.github.io/mastering-coordinator
20. Рецепты под Android: Как вкусно приготовить LayoutManager / Блог компании EastBanc Technologies / Хабрахабр URL: <https://habrahabr.ru/company/eastbanctech/blog/267497/>
21. Architecture Components URL: startandroid.ru/ru/courses/architecture-components.html
22. ViewModel и LiveData: паттерны и антипаттерны / Хабрахабр URL: <https://habrahabr.ru/post/338590/>
23. Основные компоненты Андроид URL: <https://developer.android.com/guide/components/?hl=ru>
24. Android Development Tutorial. Часть 1 / Хабр URL: <https://habr.com/post/115022/>
25. Основные компоненты Android-приложения URL: <https://geekbrains.ru/events/573>
26. View-model URL:

<https://developer.android.com/reference/android/view/View>

27. Download Android Studio and SDK tools | Android Developers URL:

<https://developer.android.com/studio/>

28. Google Play | Android Developers URL:

<https://developer.android.com/distribute/>

29. Реализация Custom View-компонента в Android | TuHub URL:

<https://tuhub.ru/posts/realizatsiya-custom-view-komponenta-v-android>

30. Операции | Android Developers URL:

<https://developer.android.com/guide/components/activities?hl=ru>

31. Урок 23. Activity Lifecycle. В каких состояниях может быть Activity URL: <http://startandroid.ru/ru/uroki/vse-uroki-spiskom/60-urok-23-activity-lifecycle-v-kakih-sostojanijah-mozhet-byt-activity.html>

32. Урок 3. Верстка экранов. Виды Layout | Школа Android URL:

<https://android-school.ru/урок-3-верстка-экранов-виды-layout/>

Краткий глоссарий

1) Активность - это компонент приложения, который выдает экран, и с которым пользователи могут взаимодействовать для выполнения каких-либо действий, например набрать номер телефона, сделать фото, отправить письмо или просмотреть карту. Каждой операции присваивается окно для прорисовки соответствующего пользовательского интерфейса. Обычно окно отображается во весь экран, однако его размер может быть меньше, и оно может размещаться поверх других окон.

2) Фрагмент - представляет поведение или часть пользовательского интерфейса в операции (класс Activity). Разработчик может объединить несколько фрагментов в одну операцию для построения многопанельного пользовательского интерфейса и повторного использования фрагмента в нескольких операциях. Фрагмент можно рассматривать как модульную часть операции.

3) Галерея – системная утилита любого устройства на ОС Android для просмотра фотографий и изображений.

4) Смартфон - Устройство, объединяющее в себе функции персонального органайзера и мобильного телефона; смарт-телефон.

5) Google Play Market - магазин приложений, игр, книг, музыки и фильмов компании Google и других компаний, позволяющий владельцам устройств с операционной системой Android устанавливать и приобретать различные приложения.

6) Форум - мероприятие, проводимое для обозначения или решения каких-либо в достаточной степени глобальных проблем.

Интерфейс - общая граница между двумя функциональными объектами, требования к которой определяются стандартом; совокупность средств, методов и правил взаимодействия между элементами системы.

Техническое задание на мобильное приложение

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ОБЩЕЙ МАТЕМАТИКИ

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОДГОТОВКИ ШКОЛЬНИКОВ
К ОЛИМПИАДАМ ПО МАТЕМАТИКЕ ДЛЯ ОС ANDROID**

Техническое задание на мобильное приложение для подготовки школьников
к олимпиадам

Разработал:
Суворкин Д.А.

ОГЛАВЛЕНИЕ

1 ВВЕДЕНИЕ.....	62
1.1 Наименование программы	62
1.2 Краткая характеристика области применения	62
2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ.....	62
2.1 Цель создания приложения	62
2.2 Основание для проведения разработки.....	62
2.3 Наименование и условное обозначение темы разработки.....	63
2.4 Задачи проекта.....	63
3 НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	63
3.1 Функциональное назначение	63
3.2 Эксплуатационное назначение	64
4 ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ ...	64
4.1 Требования к функциональным характеристикам	64
4.1.1 Требования к составу выполняемых функций.....	64
4.1.2 Требования к организации входных данных.....	64
4.1.3 Требования к организации выходных данных	65
4.1.4 Требования к временным характеристикам	65
4.2 Требования к надежности	65
4.2.1 Требования к обеспечению надежного функционирования программы	65
4.2.2 Требования к отказоустойчивости программы.....	65
5 УСЛОВИЯ ЭКСПЛУАТАЦИИ	65
5.1 Климатические условия эксплуатации	65
5.2 Требования к составу и параметрам технических средств	66
5.3 Требования к информационной и программной совместимости.....	66
5.4 Требования к исходным кодам и языкам программирования.....	66
6 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	66
7 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ	67

8 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ.....	69
8.1 Стадии разработки	69
8.2 Этапы разработки	70
8.3 Структура приложения	70
9 СИСТЕМА АДМИНИСТРИРОВАНИЯ И УЧЕТА ПОСЕЩАЕМОСТИ ПРИЛОЖЕНИЯ.....	72
9.1 Система администрирования приложения	72
9.2 Обеспечение посещаемости приложения.....	72
10 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ.....	73

1 ВВЕДЕНИЕ

1.1 Наименование программы

Наименование – «Математика? Легко!».

1.2 Краткая характеристика области применения

Программа предназначена для свободного распространения и использования всеми пользователями мобильной операционной системы Android.

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

2.1 Цель создания приложения

Приложение создается для закрепления знаний обучающегося по математике и подготовке к государственным экзаменам в виде теста.

Для достижения цели необходимо:

- иметь оборудование, подходящее системным требованиям для разработки приложения;
- проанализировать состояния рынка мобильных приложений по выбранной категории приложения;
- обеспечить разработку программного обеспечения;
- обеспечить подготовку материала и методических рекомендаций по составлению тестов по математике;

2.2 Основание для проведения разработки

Основание для проведения разработки является Приказ об утверждении тем, руководителей и консультантов выпускных квалификационных работ студентов образовательных программ НИУ «БелГУ» от 23.04.2018 № 2169-УР.

2.3 Наименование и условное обозначение темы разработки

Наименование темы разработки – «Мобильное приложение для подготовки школьников к олимпиадам по математике для ОС Android».

2.4 Задачи проекта

- изучить основы создания приложений для выбранной операционной системы;
- определиться со средой разработки и языком программирования;
- составить функциональные и пользовательские требования;
- спроектировать схему приложения;
- разработать приложения согласно установленным требованиям и схеме;
- протестировать продукт на устройствах выбранной операционной системы с разными версиями и размерами экрана;
- составление документации к продукту;
- публикация приложения в официальный магазин выбранной операционной системы с сопровождающим описанием.

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1 Функциональное назначение

Функциональным назначением программы является предоставление пользователю возможности прохождения тестирования по математике различных категорий.

3.2 Эксплуатационное назначение

Программа может использоваться на всех смартфонах и планшетах Android выше версии 5.0 (по состоянию на момент сдачи выпускной квалификационной работы).

Конечными пользователями программы могут быть все владельцы описанных выше устройств.

4 ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Программа должна обеспечивать возможность выполнения перечисленных ниже функций:

- а) навигацию между разделами приложения;
- б) навигацию между категориями;
- в) сохранение и вывод статистических данных;
- г) загрузка изображения профиля пользователя.

4.1.2 Требования к организации входных данных

Требования к организации входных данных программы не предъявляются.

4.1.3 Требования к организации выходных данных

Выходные данные должны храниться во внутренней базе данных, размещенной в папке приложения в соответствии с требованиями операционной системы.

4.1.4 Требования к временным характеристикам

Требования к временным характеристикам программы не предъявляются.

4.2 Требования к надежности

4.2.1 Требования к обеспечению надежного функционирования программы

Надежное функционирование программы должно быть обеспечено выполнением пользователем совокупности организационно-технических мероприятий, приведенных ниже:

- а) своевременной зарядкой батареи мобильного устройства;
- б) соблюдением условий эксплуатации.

4.2.2 Требования к отказоустойчивости программы

Программа должна быть защищена от случайного выхода.

5 УСЛОВИЯ ЭКСПЛУАТАЦИИ

5.1 Климатические условия эксплуатации

Надежное функционирование программы обеспечивается при соблюдении условий эксплуатации мобильного устройства.

5.2 Требования к составу и параметрам технических средств

Программа разрабатывается для следующих типов мобильных устройств:

- а) мобильный телефон;
- б) планшет.

При этом к устройству предъявляются следующие требования:

- а) объем памяти RAM – не менее 256 Мб;
- б) объем памяти ROM – не менее 512 Мб;
- в) разрешение экрана мобильного устройства – не менее 4 дюймов.

5.3 Требования к информационной и программной совместимости

Для корректной работы разрабатываемого программного средства необходимо устройство на базе операционной системы Android версии от версии 5.0 и выше (до максимальной актуальной версии по состоянию на момент сдачи выпускной квалификационной работы).

5.4 Требования к исходным кодам и языкам программирования

Исходные коды программы должны быть реализованы на языке Java и языке разметки XML. В качестве интегрированной среды разработки программы должна быть использована среда Android Studio.

6 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации должен включать в себя:

- а) техническое задание;
- б) руководство оператора;
- в) текст программы.

7 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Проект планируется выполнить с использованием языка программирования Java (www.java.com) и расширяемым языком разметки XML.

Преимущества языка Java:

- Объектно-ориентированный: в Java все является объектом.

Дополнение может быть легко расширено, так как он основан на объектной модели.

- Платформонезависимый: в отличие от многих других языков, включая C и C++, Java, когда был создан, он не компилировался в платформе конкретной машины, а в независимом от платформы байт-коде. Этот байт код распространяется через интернет и интерпретируется в Java Virtual Machine (JVM), на которой он в настоящее время работает.

- Простой: процессы изучения и введение в язык программирования Java остаются простыми. Если Вы понимаете основные концепции объектно-ориентированного программирования, то он будет прост для Вас в освоении.

- Безопасным: методы проверки подлинности основаны на шифровании с открытым ключом.

- Архитектурно-нейтральный: компилятор генерирует архитектурно-нейтральные объекты формата файла, что делает скомпилированный код исполняемым на многих процессорах, с наличием системы Java Runtime.

- Портативный: архитектурно-нейтральный и не имеющий зависимости от реализации аспектов спецификаций — все это делает Java портативным. Компилятор в Java написан на ANSI C с чистой переносимостью, который является подмножеством POSIX.
- Прочный: выполняет усилия, чтобы устранить ошибки в различных ситуациях, делая упор в основном на время компиляции, проверку ошибок и проверку во время выполнения.
- Многопоточный: функции многопоточности, можно писать программы, которые могут выполнять множество задач одновременно. Введение в язык Java этой конструктивной особенности позволяет разработчикам создавать отлаженные интерактивные приложения.
- Интерпретированный: Java байт-код переводится на лету в машинные инструкции и нигде не сохраняется. Делая процесс более быстрым и аналитическим, поскольку связывание происходит как дополнительное с небольшим весом процесса.
- Высокопроизводительный: введение Just-In-Time компилятора, позволило получить высокую производительность.
- Распространенный: предназначен для распределенной среды интернета.
- Динамический: программирование на Java считается более динамичным, чем на C или C++, так как он предназначен для адаптации к меняющимся условиям. Программы могут выполнять обширное количество во время обработки информации, которая может быть использована для проверки и разрешения доступа к объектам на время выполнения.

Преимущества языка разметки XML:

- язык разметки, который позволяет стандартизировать вид файлов-данных в виде текста, который будет понятен человеку;
- поддержка Юникод;

- возможность описания записей, списков, деревьев, форматированного текста;
- описание структур и имен полей, как описание значения полей, иными словами – это самодокументируемый формат;
- строго определенные требования к анализу и синтаксис, позволяющие быть простым, непротиворечивым и эффективным;
- основан на стандартах международного уровня;
- иерархическая структура подходит для описания любых документов, кроме видео и аудио файлов, растровых изображений, двоичных данных и сетевых структур данных;
- простой текст, свободный от всяких ограничений и лицензирования;
- подмножество SGML;
- свободен от всяких требований на порядок расположения атрибутов в элементах;
- реализация парсеров для всех языков программирования;
- XSLT стандартный механизм преобразования;
- поддерживается на низком программном уровне.

8 СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

8.1 Стадии разработки

Разработка должна быть проведена в три стадии:

1. проведение предварительного исследования;
2. разработка технического задания;
3. рабочее проектирование.

8.2 Этапы разработки

На стадии проведения предварительного исследования необходимо сформировать требования к программе путем обзора существующих аналогов, выбора операционной системы, выбора средств разработки.

На стадии разработки технического задания должен быть выполнен этап разработки настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены следующие этапы работ:

1. проектирование программы;
2. разработка программы;
3. разработка программной документации;
4. испытание программы.

8.3 Структура приложения

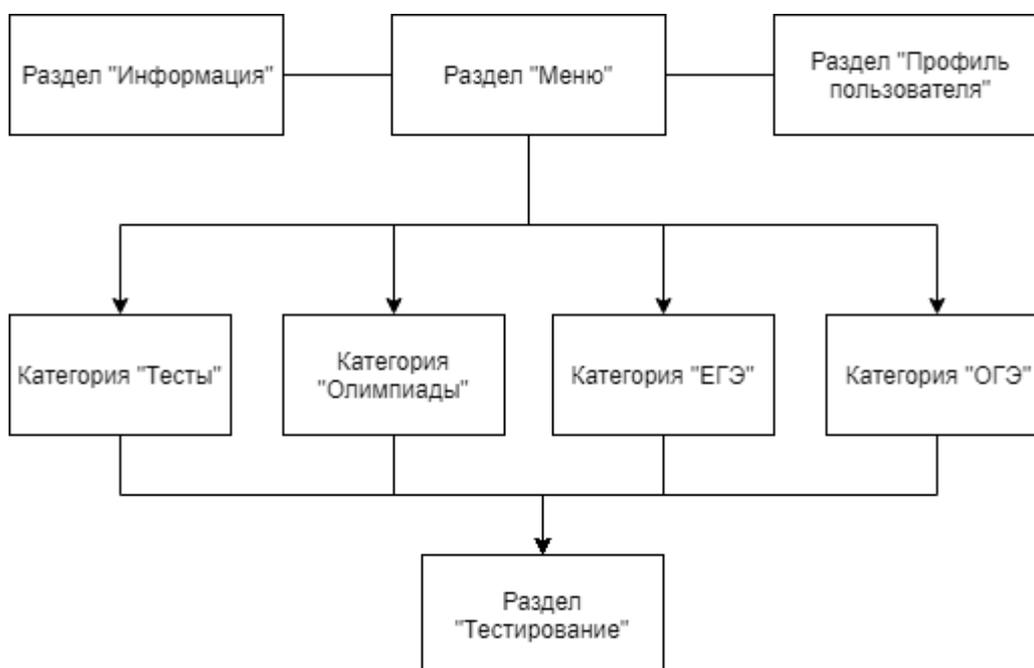


Рисунок П.Б.8.1

Раздел «Информация» (FragmentInformation) содержит информацию об авторе разрабатываемого приложения, ссылки на сайт института и университета в картинках. Также, ссылки на социальные сети разработчика.

Раздел «Профиль пользователя» (FragmentProfile) включает в себя область для фотографии пользователя и список статистических данных. При клике в области для фото пользователя, в новом окне открывается галерея смартфона, в которой необходимо выбрать фото для загрузки. После выбора фото окно галереи закрывается. Далее, фото загружается в свою область и сохраняется в память телефона, для дальнейшей автоматической загрузки при перезапуске приложения. Статистические данные включают в себя показатель количества пройденных тестов, средний процент прохождения теста и количество использованных пропусков при прохождении теста.

Раздел «Меню» содержит четыре окна категорий тестов. Переключения между окнами происходит горизонтальной прокруткой. В данном разделе присутствует область с названием меню, изображение, характеризующее категорию тестов, горизонтальный список имен категорий и область для вывода списка выбранной категории тестов.

Категории тестирований содержат вертикально прокручивающийся список тестов выбранной категории. Элемент списка включает в себя текстовые поля с названием теста и количеством вопросов, трехцветный индикатор сложности теста.

Раздел «Тестирование» содержит фрагмент вопроса с ответами и две кнопки «Закончить» и «Пропустить». Во фрагменте находятся текстовые поля с номером задания и вопросом. Некоторые вопросы сопровождаются заданиями с изображением. В каждом задании теста есть, как минимум, 3 ответа. При клике на ответ во фрагмент загружается следующее задание до тех пор, пока вопросы теста не закончатся. При клике на кнопку «Пропустить» во фрагмент загружается следующее задания, а текущее помещается в очередь заданий на последнее место. При клике на кнопку «Закончить» прохождение

тестирования прерывается, и появляется окно с результатами прохождения теста с возможностью выхода в главное меню и повторным прохождением теста.

9 СИСТЕМА АДМИНИСТРИРОВАНИЯ И УЧЕТА ПОСЕЩАЕМОСТИ ПРИЛОЖЕНИЯ

9.1 Система администрирования приложения

Вся информация размещена во встроенной в приложение базе данных. Также, в приложении есть динамические методы, для быстрого создания тестов. Для создания теста, необходимо в специальном методе указать название теста, рекомендуемый класс для прохождения, загрузить в приложение изображение задания, если имеется, указать в методе ссылку на ресурс изображения и инициализировать ответы на вопросы с указанием правильного варианта.

9.2 Обеспечение посещаемости приложения

Для обеспечения высокой посещаемости приложения предполагается провести следующие мероприятия:

- контекстная реклама в сети «Instagram»;
- реклама в других приложениях подобного жанра;
- реклама приложения в учебных заведениях среднего образования;
- обмен баннерами с Интернет-проектами сходной тематики и сайтами учебных и научных организаций РФ

10 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

По завершению разработки программы и программной документации необходимо провести испытания программы. Испытания должны доказать работоспособность функций программы, а также проверить выполнено ли требование к надежности.