

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

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ
КАФЕДРА ОБЩЕЙ МАТЕМАТИКИ

**МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ЗАОЧНОЙ МАТЕМАТИЧЕСКОЙ
ШКОЛЫ НИУ «БЕЛГУ» ДЛЯ ОС ANDROID**

Выпускная квалификационная работа
обучающегося по направлению подготовки
01.03.02 Прикладная математика и информатика
очной формы обучения, группы 12001510
Фролова Виктора Алексеевича

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

БЕЛГОРОД
2019

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И АНАЛИЗ ПРОГРАММНЫХ ПРОДУКТОВ.....	6
1.1 Описание предметной области.....	6
1.2 Поиск и анализ аналогов разрабатываемого приложения.....	9
2 АНАЛИЗ АРХИТЕКТУРЫ ANDROID И ВЫБОР СРЕДЫ РАЗРАБОТКИ	14
2.1 Архитектура Android	14
2.2 Эмулятор Android SDK.....	17
3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ	24
3.1 Структура мобильного приложения.....	24
3.2 Android Манифест	24
3.3 Сборка приложения.....	27
3.4 Активности в Android приложении.....	31
3.5 Разметка приложения Android.....	35
3.6 Создание меню	39
3.7 Фрагменты.....	42
3.8 Темы для приложения Android.....	45
ЗАКЛЮЧЕНИЕ.....	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	48
ПРИЛОЖЕНИЕ А.....	51
ПРИЛОЖЕНИЕ Б.....	52

ВВЕДЕНИЕ

Математическая грамотность является очень важным аспектом различных сфер общественной жизни. С практической точки зрения математика формирует способы деятельности, с духовной - интеллектуальное развитие человека, формирование его характера и культуры в целом.

Без основ математики на сегодняшний день невозможно стать образованным человеком. Не случайно математика была названа царицей наук, ведь она является опорой для смежных наук. С каждым днем появляется все больше специальностей непосредственно связанных с применением математических знаний - в экономике, физике, информатике, микробиологии, робототехнике и многих других. Таким образом ясно просматривается тенденция увеличения роли данного предмета в жизни людей.

В современном мире все большую популярность представляют заочные школы обучения, которые предоставляют услуги высоко квалифицированных специалистов обучающимся из любых уголков страны, где проведена глобальная сеть Интернет. Так как данная технология в России распространена повсеместно и имеет сравнительно высокие скоростные показатели и низкие показатели стоимости, то данные услуги набирают все большую и большую популярность. Более того, нельзя не упоминать и тот факт, что во всем мире растут продажи мобильных устройств, а значит растет и спрос на различные мобильные приложения, которые позволяют пользователям получать необходимые им услуги практически в любое время и в любом месте, ведь на сегодняшний день человеку тяжело представить себя без устройств связи с внешним миром.

Однако стоит помнить, что на сегодняшний день не существуют и не могут существовать инструменты мобильной разработки, которые позволяют создать приложение, способное запускаться на любом мобильном

устройстве, ведь из-за погони к уникальности многие производители создали и продолжают создавать собственные операционные системы, а вследствие этого возникают проблемы с выбором определенной ОС или же с совместимостью приложения на различных устройствах.

На сегодняшний день самой популярной операционной системой для мобильных устройств можно безошибочно назвать Android, который занимает 73.52% от всего мирового рынка по данным статистики агентства Statcounter. Эта ОС приобрела такую популярность благодаря многим факторам, в том числе своей открытости, возможности индивидуальной настройки большого количества параметров интерфейса, право установки приложений в обход фирменного магазина, то есть со сторонних ресурсов.

Как ожидается в IT-индустрии, в скором времени будет наблюдаться тенденция быстрого увеличения в размере и объеме программного обеспечения для мобильных устройств, что и обеспечивает актуальность темы выпускной квалификационной работы, ведь разработка и поддержка приложений для ОС Android должна стать наиболее востребованной на рынке.

Цель работы – разработка мобильного приложения для заочной математической школы имени Софьи Ковалевской для операционной системы Android.

Для достижения поставленной цели было необходимо решить следующие задачи:

- рассмотреть предметную область и провести обзор аналогов приложения;
- выбрать инструментальную среду, в которой будет вестись разработка;
- предложить структуру мобильного приложения и спроектировать основные модули;

- разработка программного кода приложения при помощи выбранной инструментальной среды разработки;
- провести тестирование приложения.

Структура и объем работы: выпускная квалификационная работа выполнена на 68 страницах машинописного текста, состоит из введения, четырех глав, заключения и двух приложений.

В первой главе производится описание предметной области и обзор аналогов разработанного приложения.

Во второй главе проводится выбор инструментальной среды разработки и анализ специфики создания приложений для ОС Android.

Третья глава содержит описание этапов разработки, основных принципов жизненного цикла, описание основных проектных решений и разработанного программного продукта.

В приложении размещены краткий глоссарий и программный код.

1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И АНАЛИЗ ПРОГРАММНЫХ ПРОДУКТОВ

1.1 Описание предметной области

В 2005 году корпорация Google выкупила компанию Android, которая уже в 2007 году анонсировала новую открытую мобильную платформу на базе Linux, а в 2008 выпустила официальную первую версию Android. Данная операционная система позволяла запускать приложения на языке Java посредством подключения необходимых библиотек, разработанных Google. Более того, эта ОС стала поддерживать и программы написанные на других языках, таких как C и другие, благодаря инструментарию Android Native Development Kit, выпущенном в 2009 году.

С момента выхода на свет первой версии Android прошло уже больше десяти лет, а значит и версии данной операционной системы претерпели множество изменений и вариаций, что следует учитывать, если дело касается разработки под данные платформы. Облегчающим фактором является тот, что все версии этой операционной системы поддерживают обратную совместимость, таким образом, код, разработанный под младшую версию Android будет без проблем запускаться и на более поздних.

При разработке приложения на Android в первую очередь придется столкнуться с API (application programming interface – программный интерфейс приложения), который и определяет функциональность, которую сможет предоставить программа. У API существуют несколько уровней, зависящие от версии, установленной на мобильное устройство операционной системы. На 2019 год распределение используемых версий представлены ниже в таблице 1.1 и на рисунке 1.1.1, для большей наглядности.

Таблица 1.1 – Распределение версий Android

Версия	Название	API	Распределение
2.3.3 - 2.3.7	Gingerbread	10	0.3%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.3%
4.1.x	Jelly Bean	16	1.2%
4.2.x		17	1.5%
4.3		18	0.5%
4.4	KitKat	19	6.9%
5.0	Lollipop	21	3.0%
5.1		22	11.5%
6.0	Marshmallow	23	16.9%
7.0	Nougat	24	11.4%
7.1		25	7.8%
8.0	Oreo	26	12.9%
8.1		27	15.4%
9	Pie	28	10.4%

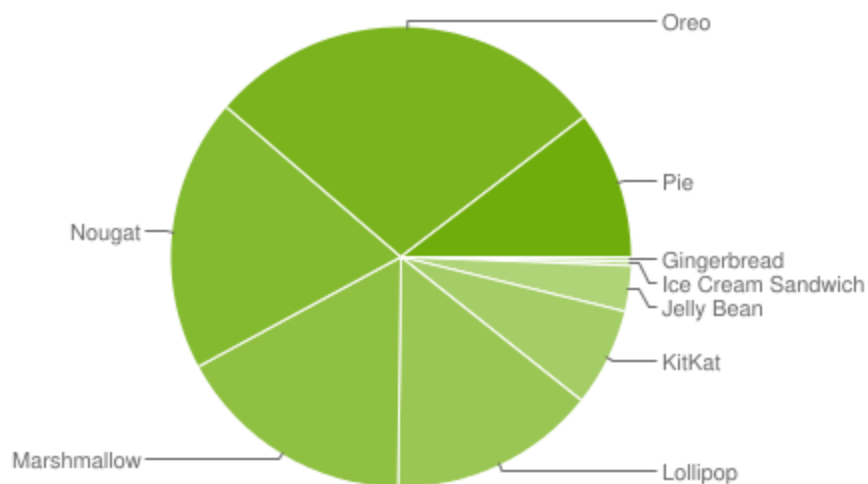


Рис. 1.1.1 Диаграмма распределения версий Android

Таким образом, если при разработке будет использоваться 15-й уровень API, то приложение сможет запускаться на 99.7% мобильных устройств на базе Android. Такое решение наиболее популярно на текущий момент, так как позволяет охватить практически всю аудиторию.

Не менее важно при разработке помнить, что устройства отличаются друг от друга не только операционными системами и их версиями, но также и экранами, которые имеют в первую очередь различное разрешение. Одной из главных задач разработчиков является адаптация приложения под максимально целесообразное количество потенциальных устройств, на которых, возможно, будет установлено данное приложение. Чтобы упростить способ разработки пользовательских интерфейсов для различных конфигураций экрана, Android делит диапазон фактических размеров и плотностей экрана на несколько сегментов, как показано в таблице 1.2, а также на рисунках 1.1.2 и 1.1.3.

Таблица 1.2 – Статистика распространенности экранов

	ldpi	mdpi	tvdpi	hdpi	xhdpi	xxhdpi	Всего
Small	0.4%				0.1%	0.1%	0.6%
Normal		0.9%	0.3%	24.0%	37.7%	23.6%	86.5%
Large		2.4%	1.9%	0.6%	1.7%	8.2%	
Xlarge		3.1%		1.3%	0.6%		5.0%
Всего	0.4%	6.4%	2.2%	25.9%	40.0%	25.4%	

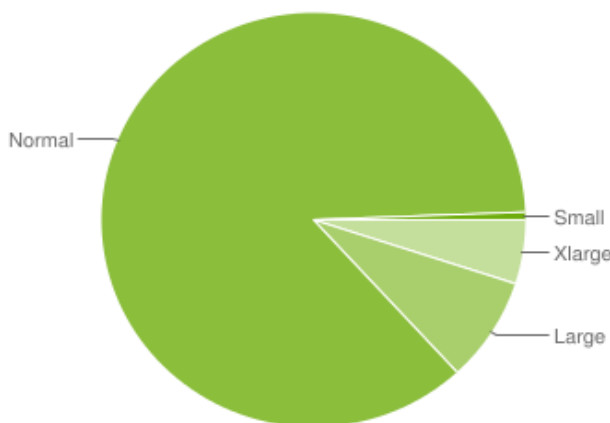


Рис. 1.1.2 Диаграмма распространенности экранов

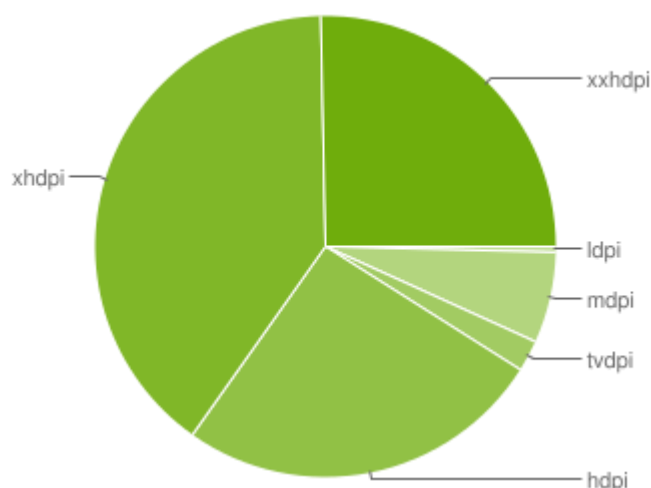


Рис. 1.1.3 Диаграмма распространенности экранов

1.2 Поиск и анализ аналогов разрабатываемого приложения

На сегодняшний день за рубежом все большее распространение получают мобильные приложения для различных учебных заведений, как для очного, так и для заочного обучения. По сути все эти приложения представляют собой своего рода справочники, то есть предоставляют пользователям информацию о местоположении школы, контактные лица, события и прочее. Все эти приложения, если и отличаются по функциональной составляющей, то не критично, поэтому сложно судить их по этому параметру, единственное, на что можно обращать внимание, это на графическую составляющую, то есть на их интерфейс. Он должен быть интуитивно понятен, приятен для глаза, информативен, но при этом не перегружен. Ниже на рисунках 1.2.1 – 1.2.6 показаны наиболее популярные за рубежом приложения различных школ/университетов, а именно Minidarie Primary School, Huraki Primary School, International School of Macao, Glendowie School, Harrabeen Sports High School и Rakaia School. Наиболее удобным интерфейсом можно считать представленные приложения на рисунках 1.2.1, 1.2.3, 1.2.5, так как они либо не перегружены огромным количеством кнопок, либо все эти кнопки дифференцированы по размеру и цветом фона, для лучшего восприятия.

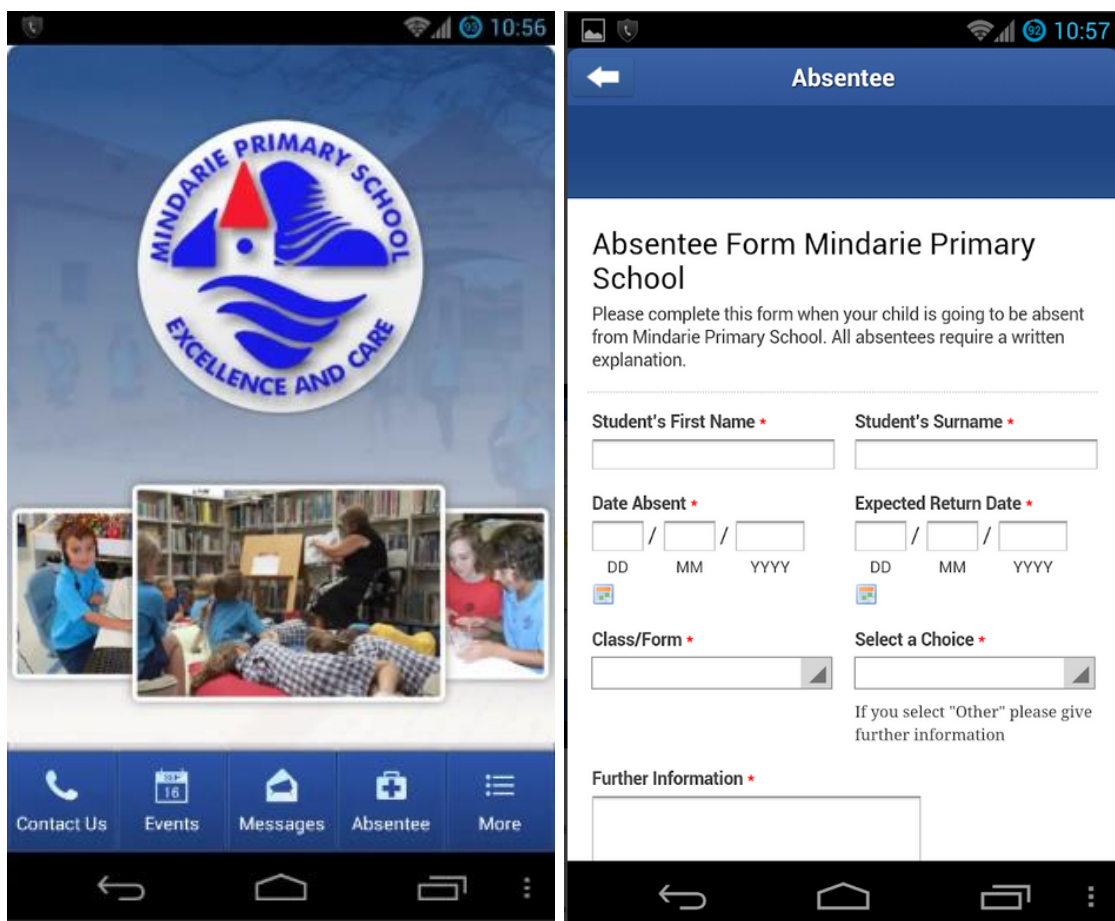


Рис 1.2.1 Минидарская начальная школа



Рис. 1.2.2 Хауракская начальная школа

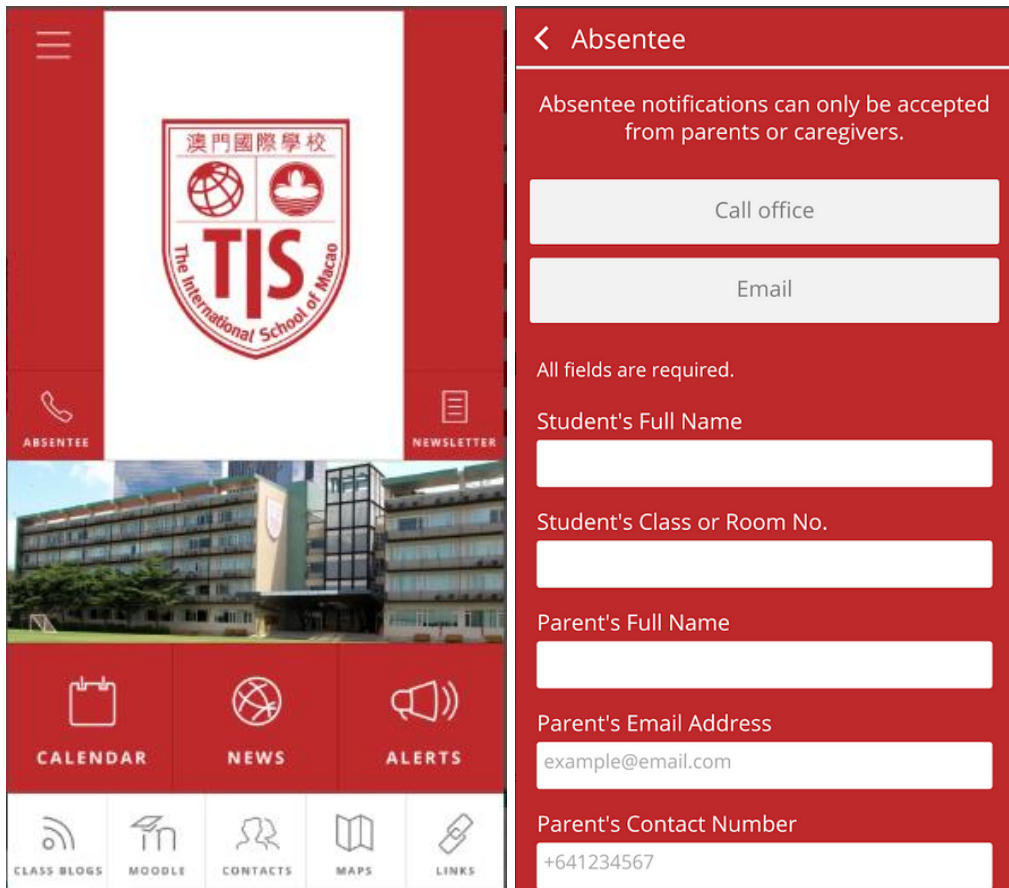


Рис 1.2.3 Международная школа Макао



Рис 1.2.4 Глендовская школа

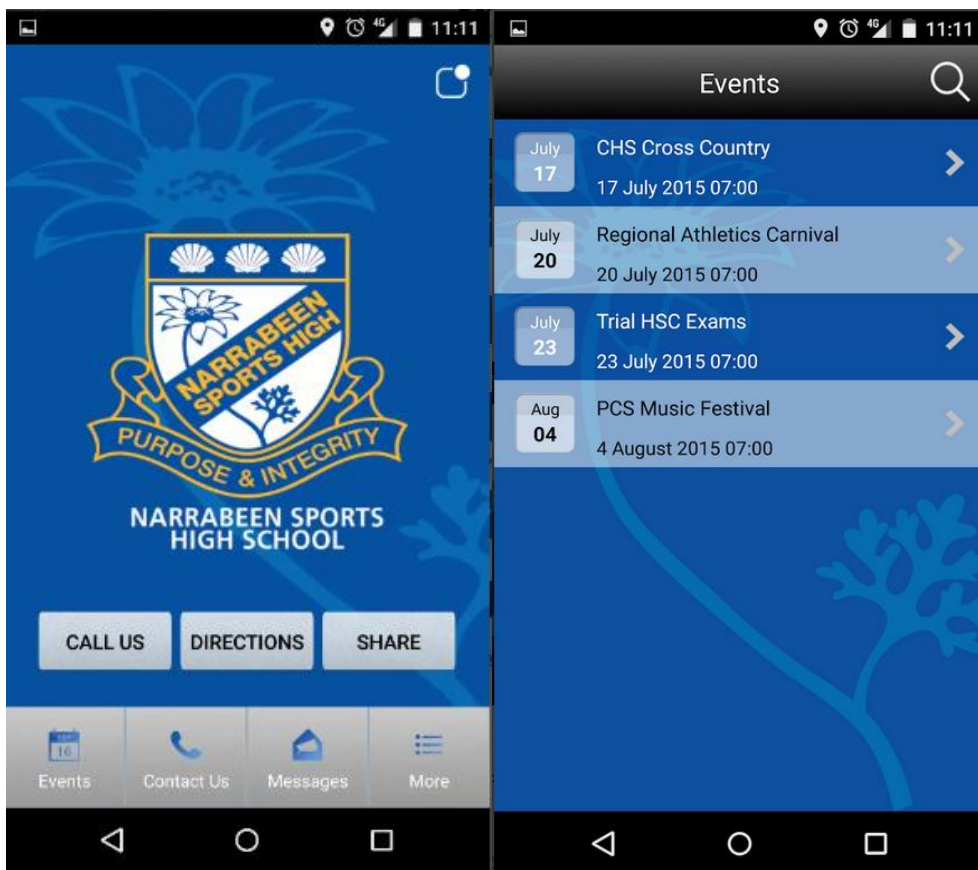


Рис. 1.2.5 Нарабьенская спортивная высшая школа

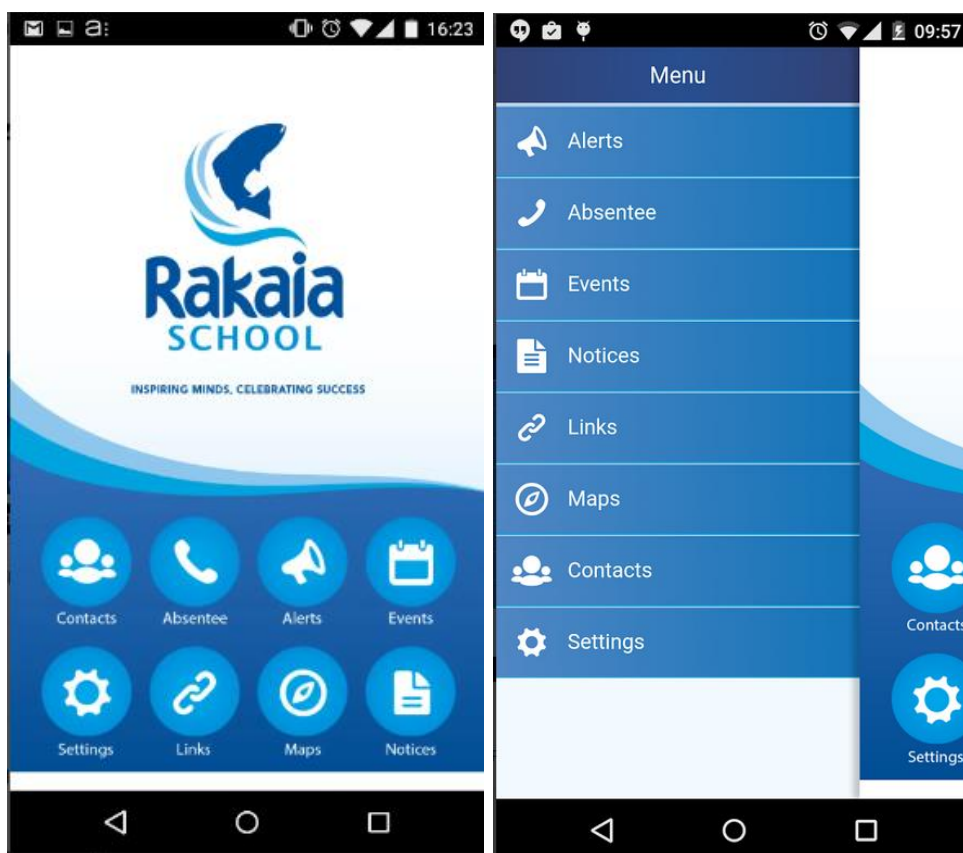


Рис. 1.2.6 Ракайская школа

Предварительно собрав статистику, в таблице 1.3 рассмотрим аналоги разрабатываемого приложения.

Таблица 1.3. Аналоги разрабатываемого приложения

Позиция в поиске	Название	Рейтинг (5 звезд)	Год публикации
1	Минидарская начальная школа	5	2018
2	Хауракская начальная школа	5	2018
3	Международная школа Макао	5	2019
4	Глендовская школа	4.5	2019
5	Нарабьенская спортивная высшая школа	2	2018
6	Ракайская школа	3	2018

С момента публикации большинства приложений данной категории в магазине Google Play Market не прошло еще даже года, что говорит о том, насколько актуальна идея разработки мобильного приложения для заочной школы. Количество пользователей же напрямую зависит от количества учеников этих школ.

2 АНАЛИЗ АРХИТЕКТУРЫ ANDROID И ВЫБОР СРЕДЫ РАЗРАБОТКИ

2.1 Архитектура Android

Android – это платформа, которая дает возможность разработчику создавать код на языке Java абстрагировано от ядра, а так же предоставляет несколько полезных функций. В первую очередь стоит отметить немалый набор API, позволяющий создавать разнообразные приложения, дает возможность повторно использовать и заменять компоненты, которые предлагают приложения платформы и сторонние производители. Кроме того, в операционной системе Android присутствует виртуальная машина Dalvik, разработанная компанией Google, которая позволяет запускать приложения. В-третьих, эта платформа предоставляет множество библиотек – графических, для разработки 2D, 3D приложений, мультимедийных, API, позволяющих использовать встроенную камеру, экран, компас, динамики, память, GPS и другие аппаратные ресурсы мобильного устройства. Конечно, это далеко не все возможности Android, так как их перечисление займет продолжительное время. Однако, это наиболее популярные и важные при разработке большинства приложений. Архитектура Android состоит из большого количества компонентов. Основные компоненты Android представлены на рисунке 2.1.1.

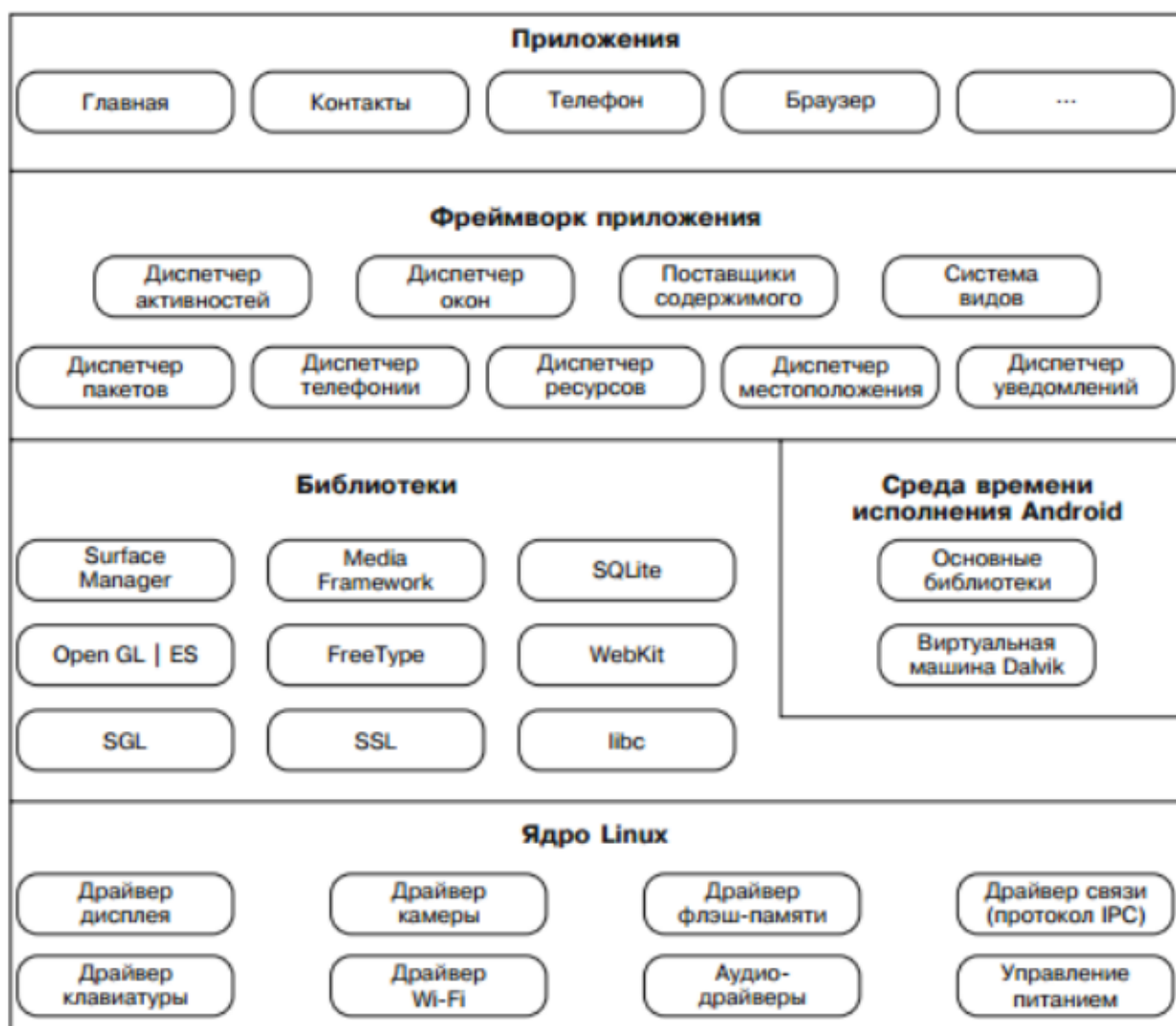


Рис. 2.1.1 Обзор архитектуры Android

Из представленного рисунка отчетливо видно, что обеспечение драйверами аппаратным компонентам системы производится ядром Linux, которое также управляет памятью, процессами, сетевой поддержкой и так далее. Каждое приложение использует свою собственную виртуальную машину Dalvik и имеет свой собственный процесс.

Файлы Java и Class преобразуются с расширением DEX при помощи инструмента DX, который доступен в SDK. Это преобразование необходимо для запуска программы в Dalvik, использующий байткод DEX. В преимущества данного расширения можно записать гораздо меньший объем занимаемой памяти, в отличие от классических типов файлов, что было

достигнуто благодаря сжатию, разделению на таблицы и соединению нескольких файлов.

Dalvik получает функциональные возможности для Java приложений взаимодействуя с библиотеками ядра, которые предоставляют немалый, однако не весь, список классов.

Начиная с версии 2.2 Android получил возможность компиляции байткода «на лету», благодаря отслеживанию JIT компилятора. Это изменение позволило увеличить производительность программ, требующих больших вычислений. Данный компилятор способен использовать возможности процессора для тяжелых вычислений, например, операции с плавающей точкой. Более того, начиная с этой версии Android получил сборщик мусора Garbage Collector (GC), работающий по принципу уведомить и забрать, что иной раз может поставить разработчика в тупик, однако при тщательном изучении и должном подходе может стать очень эффективным.

Помимо основных библиотек, предлагающие возможности Java SE, существуют еще библиотеки C и C++, помогающие в создании основы применения структуры. Данные библиотеки отвечают за целый набор функций, что не сильно подходит к виртуальной машине Dalvik, поэтому API в программе представил их в виде классов Java, которые и используются при разработке.

Системные библиотеки связываются со средами выполнения фреймворком, тем самым обеспечивая связь с Android. Управление приложениями производится структурой, предлагающей сложную среду. Под эту структуру с набором API интерфейсов разработчиками создаются приложения, которые охватывают – пользовательский интерфейс, управление ресурсами и фоновыми службами, уведомления и так далее. Также у приложений, интерфейса и фоновых служб есть возможность взаимодействия с другими приложениями. Реализовано это для предоставления приложениям использования других компонентов, что позволяет снизить трудозатраты программистов.

2.2 Эмулятор Android SDK

Для разработки программного обеспечения для мобильных устройств под управлением операционной системой Android применяется комплект инструментов Android Software Development Kit. Для некоторых сред разработки, таких как Eclipse данный инструментарий нужно вручную устанавливать, в других же он присутствует по умолчанию, например, в Android Studio, где установлен плагин ADT, который используется для отладки и тестирования программ, однако есть и второй вариант использования SDK – при работе с инструментами командной строки. Благодаря эмулятору существует возможность написания полностью готового приложения, адаптированного под необходимые мобильные устройства даже без запуска и проверки на реальных физических устройствах.

На сегодняшний день эмуляторы способны имитировать различные характеристики устройств, например, камеру, Wi-Fi соединение, технологию Bluetooth и многие другие, часть из которых изображены на рисунке 2.2.1 на примере мобильного устройства Nexus 5X.

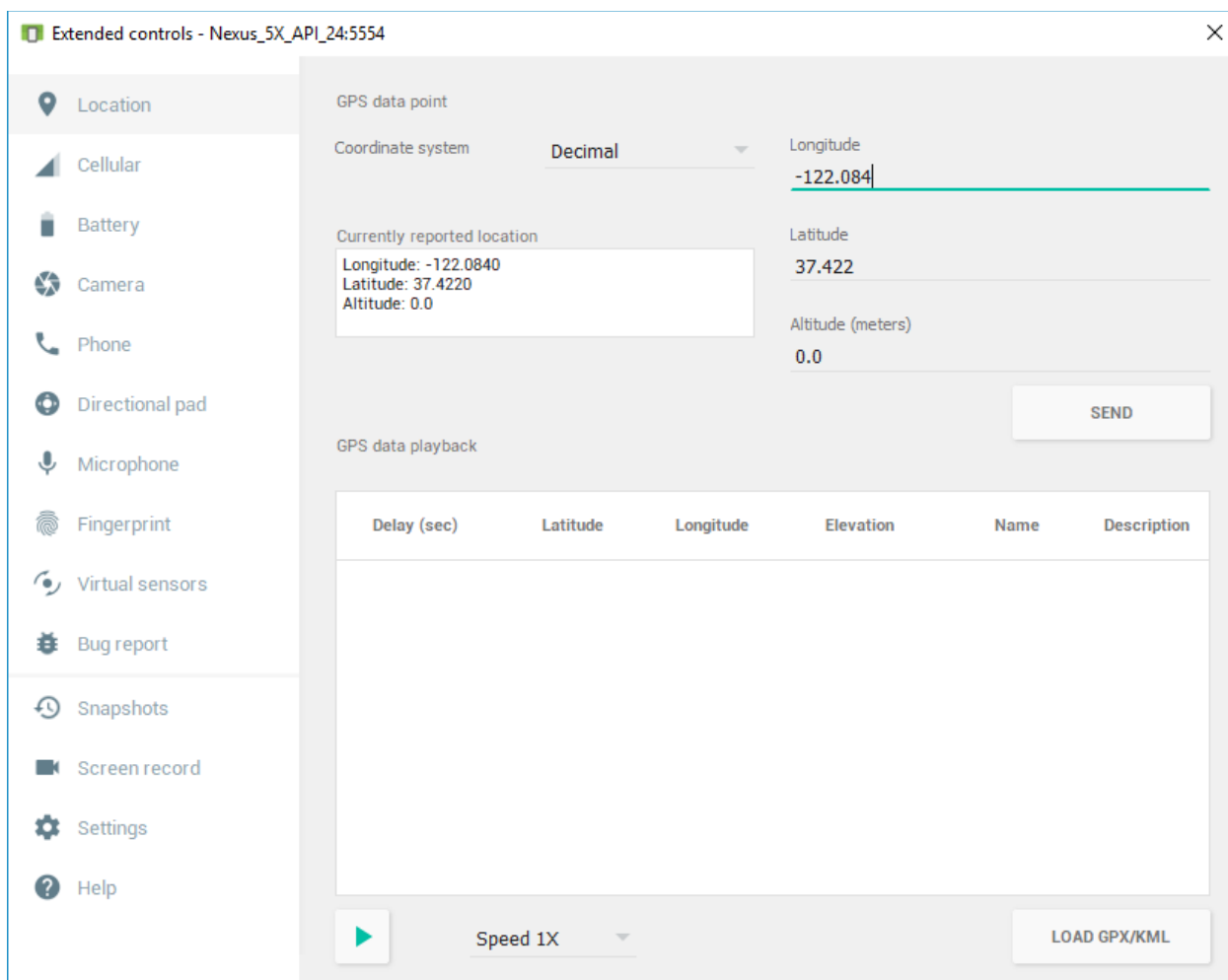


Рис. 2.2.1 Имитируемые характеристики Nexus 5X

При работе эмулятора происходит имитирование процессора, функционирующего на основе ARM, которую широко используют мобильные устройства и встроенные электронные устройства, благодаря использованию этой архитектурой сокращенных наборов команд, повышающих простоту и производительность.

SDK в Android Studio позволяет эмулировать все версии Android начиная с 2.1 Eclair и заканчивая последней версией 9.0 Pie (рисунок 2.2.2).

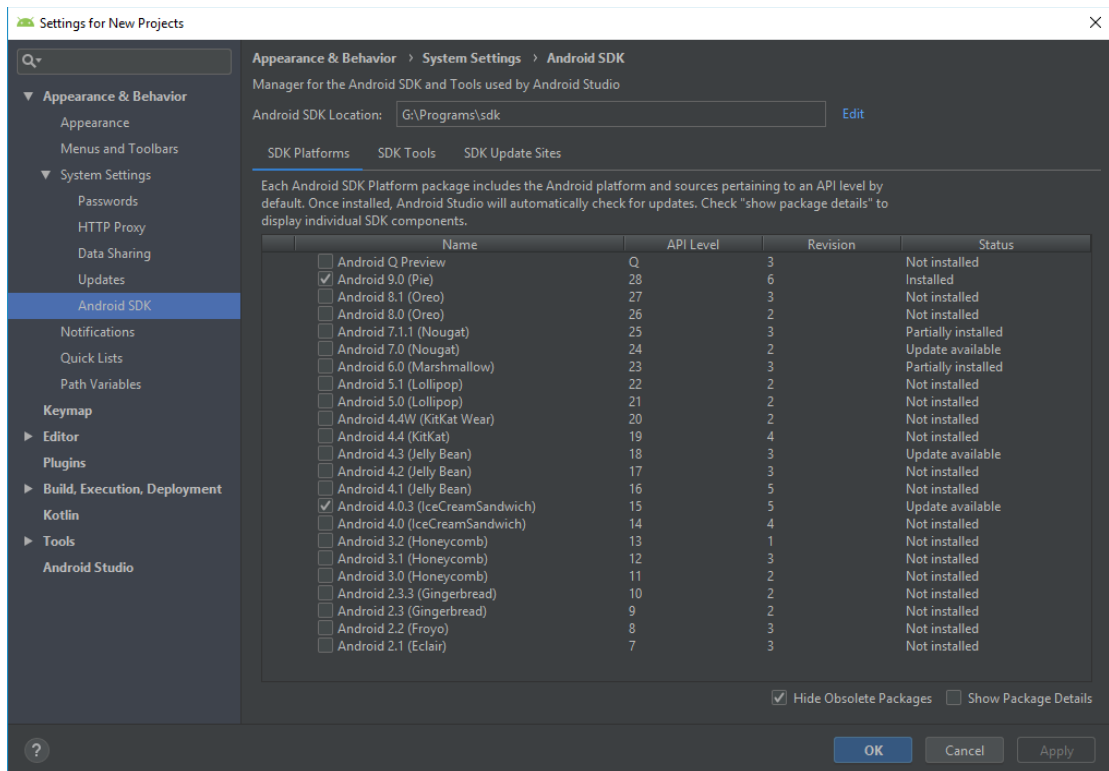


Рис. 2.2.2 Список версий Android

Более того, Android Studio позволяет имитировать огромное количество мобильных устройств, включая их функциональные и визуальные особенности (рисунки 2.2.3, 2.2.4).

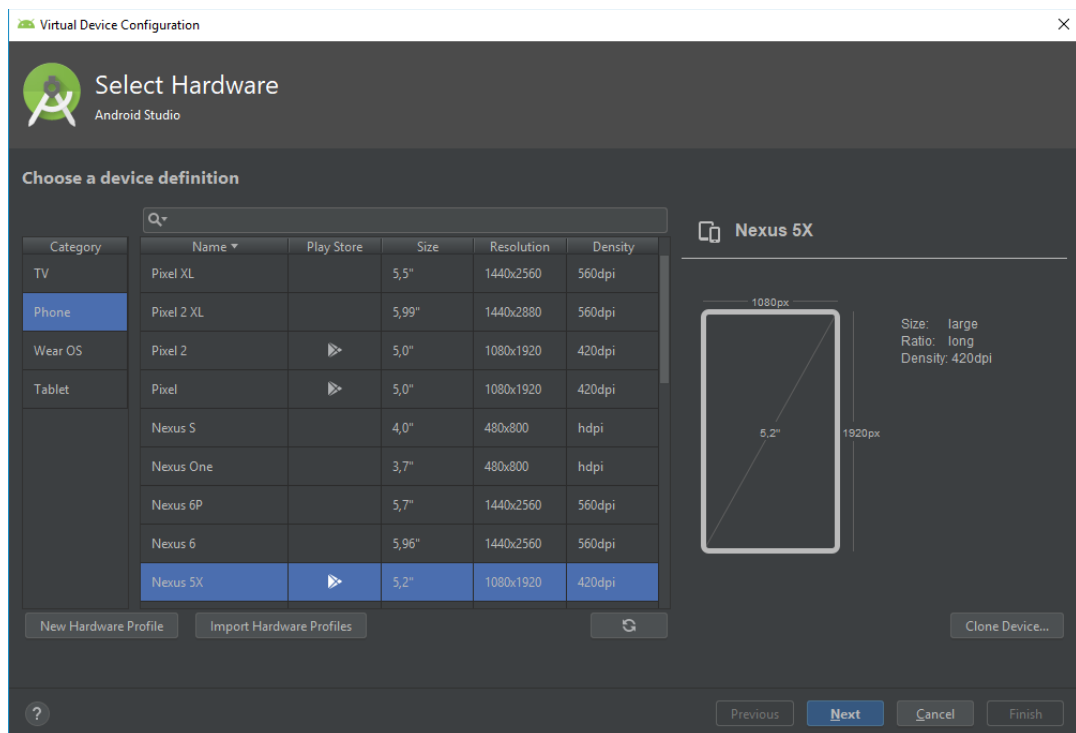


Рис. 2.2.3 Примерный список имитируемых устройств

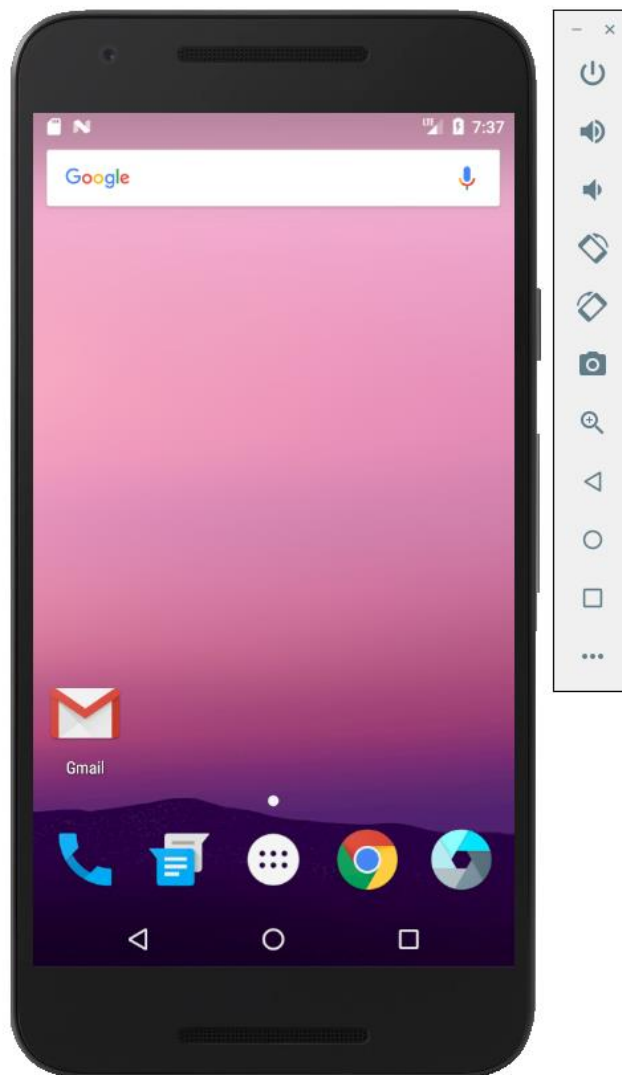


Рис. 2.2.4 Имитация Nexus 5X

2.3 Выбор среды разработки

На сегодняшний день существуют 3 наиболее популярные среды разработки под мобильные приложения для операционной системы Android:

1. Android Studio
2. RAD Studio
3. Eclipse

Выбор среды разработки очень важен, ведь от этого может зависеть как скорость написания приложения, так и его качество, наличие/отсутствие ошибок, багов.

Android Studio – интегрированная среда разработки, которую создала компания Google. Основным преимуществом является то, что эта среда разрабатывалась специально под создание приложений для Android и сразу включает в себя большое количество всех необходимых инструментов. Среди преимуществ также нужно выделить возможность видеть изменения сразу же после их совершения, поддержка консоли и огромное количество стандартных шаблонов дизайна, а также то, что функционалом данного продукта можно пользоваться абсолютно бесплатно (рис. 2.3.1).

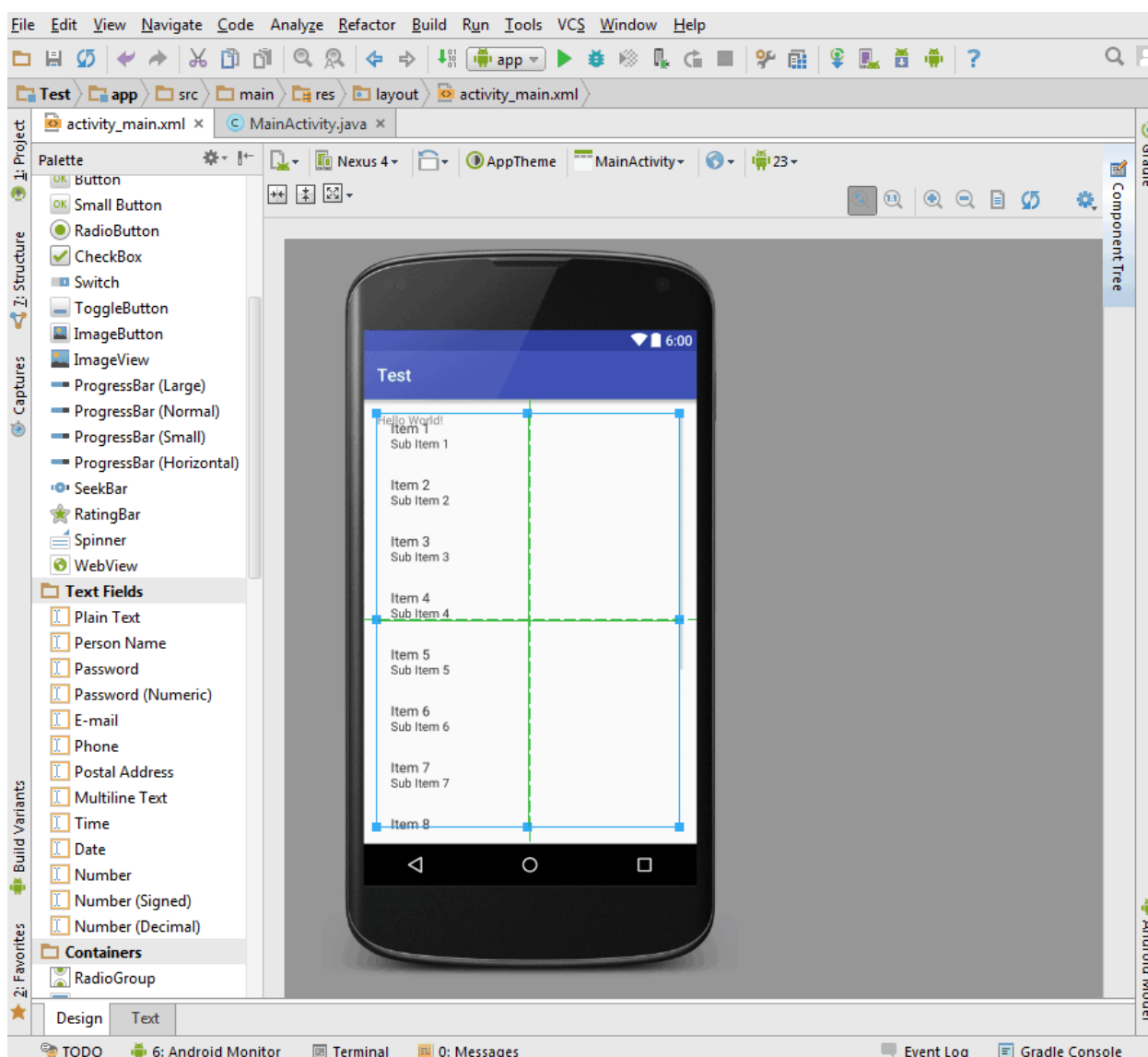


Рис. 2.3.1 Окно Android Studio

RAD Studio – среда разработки кроссплатформенных приложений, в первую очередь и для мобильных устройств (рис. 2.3.2). Одним из главных преимуществ является использование облачных сервисов, которые позволяют ускорить процесс разработки приложений. В этом программном продукте так же есть отображение изменений приложения в реальном времени. Доступна также функция переключения платформ с одной на другую. Из главных минусов можно выделить платную лицензию, но присутствует и демоверсия продукта на 30 дней.

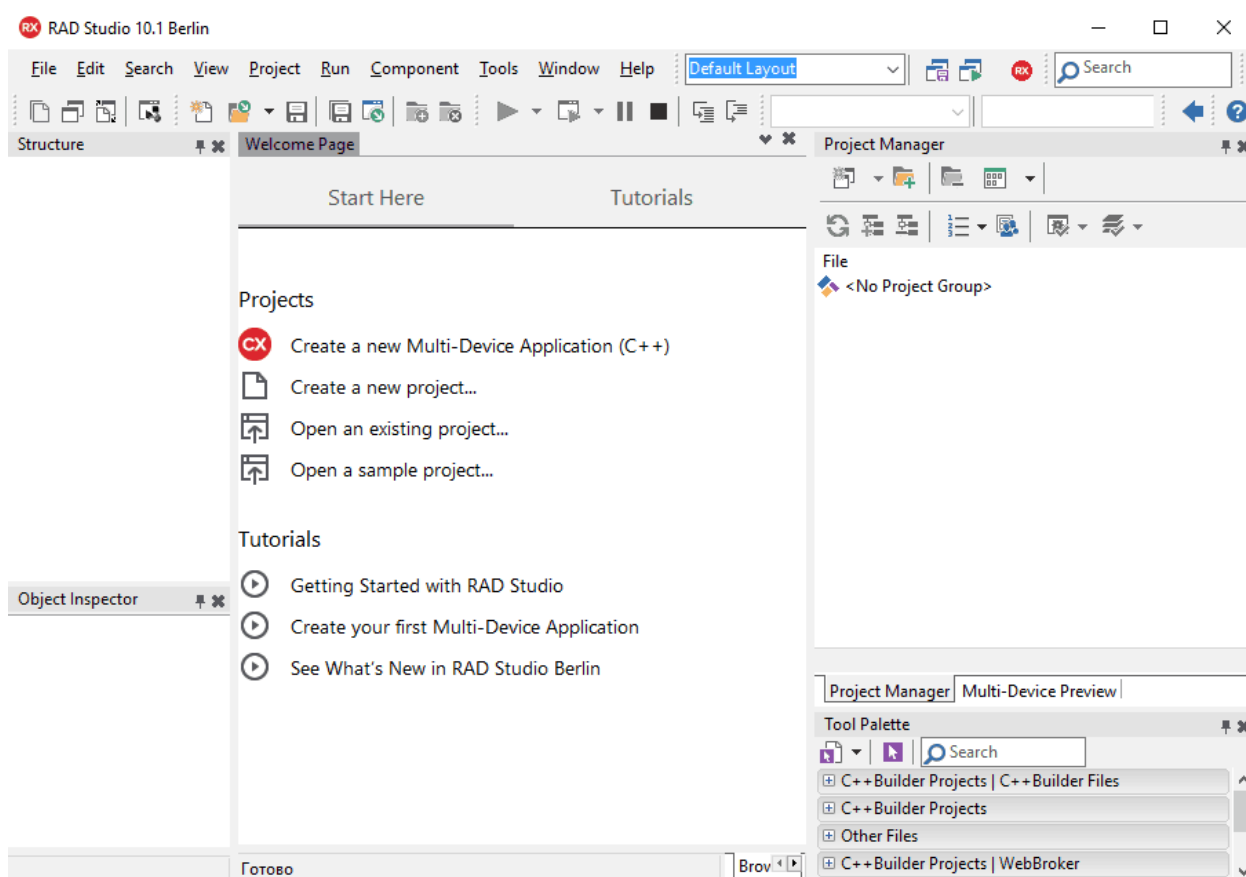


Рис. 2.3.2 Окно RAD Studio

Eclipse – наиболее популярная программная платформа для разработки приложений, в том числе и мобильных программ (рис. 2.3.3). Главными преимуществами Eclipse можно назвать – большой набор API и RCP подход, который позволяет написать практически любую программу. Также

присутствует возможность загрузить необходимые для написания приложения SDK.

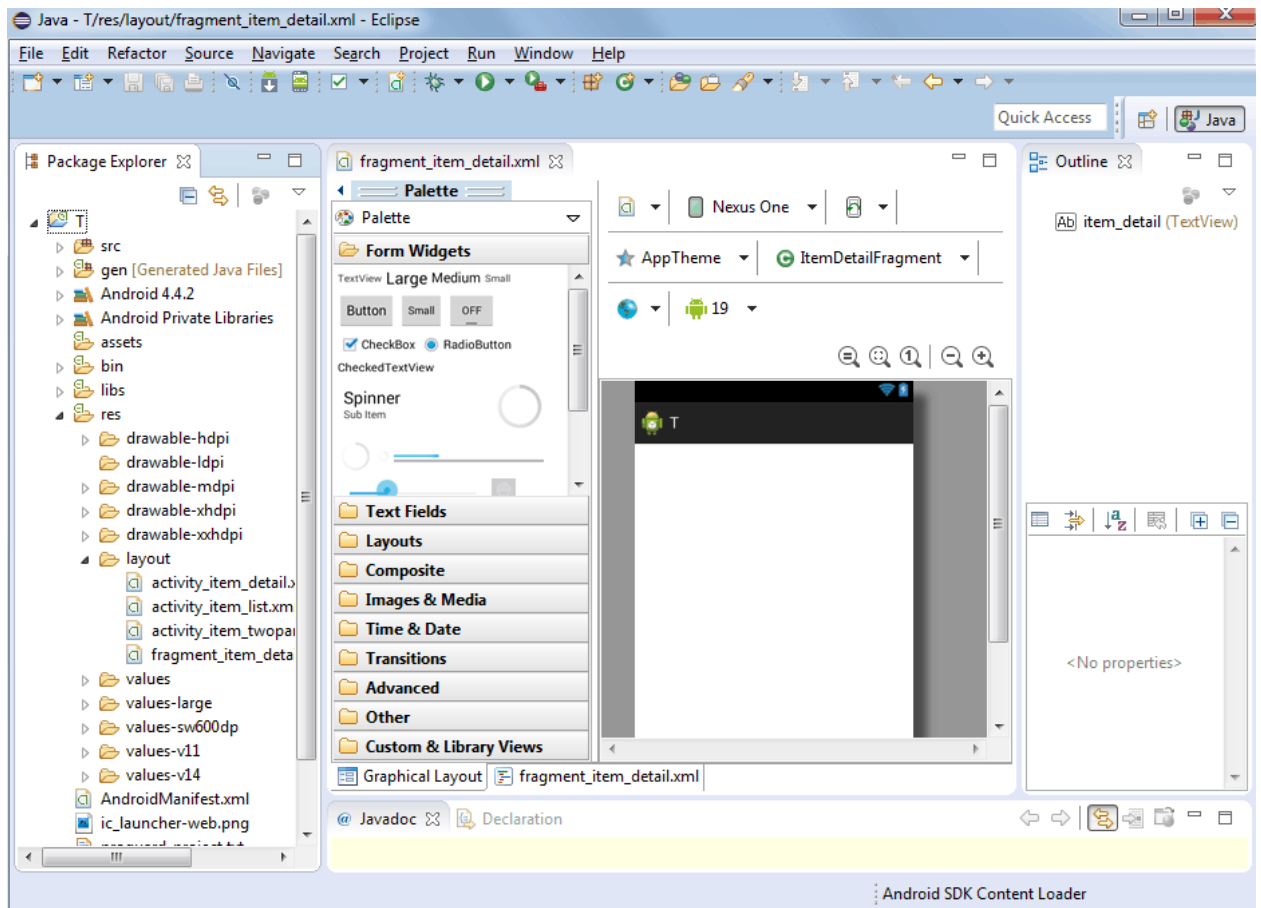


Рис. 2.3.3 Окно Eclipse

Подводя итоги сравнения лучшей программной платформой все же стоит выбрать Android Studio, так как она является наиболее удобной, созданной той же корпорацией, что и сама операционная система Android, а также содержит в себе весь необходимый инструментарий для разработки мобильных приложений.

3 РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ

3.1 Структура мобильного приложения

Общая схема приложения представлена на рисунке 3.1.1.

Разработанное приложение состоит из 5 основных окон: «Главная», «Новости», «Архив», «Преподаватели» и «Карта».

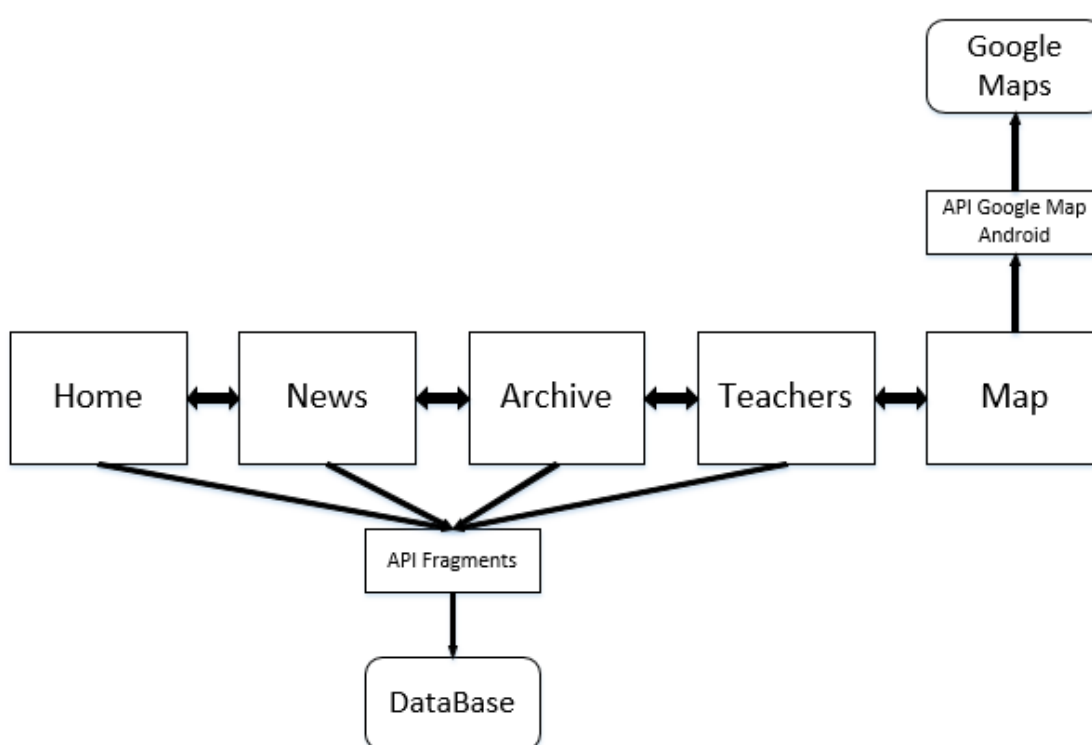


Рис. 3.1.1 Структура приложения

3.2 Android Манифест

Файл манифеста для приложения android – это файл ресурсов, который содержит все сведения, необходимые системе android о приложении. Это ключевой файл, который работает как мост между разработчиком и

платформой Android, что и помогает разработчику передать функциональность и требования нашего приложения. Это XML-файл, который должен быть назван AndroidManifest.xml и размещен в корневом каталоге приложения. Каждое приложение для Android должно иметь AndroidManifest.XML-файл. AndroidManifest.xml позволяет определить пакеты, API, библиотеки, необходимые для приложения, основные блоки приложения, такие как деятельность, услуги и т. д., сведения о разрешениях, набор классов, необходимых перед стартом.

Ниже в таблицах 3.1 и 3.2 представлены элементы, которые могут появиться в AndroidManifest.XML. Этот список ограничен, и мы не можем добавить наши собственные элементы.

Таблица 3.1 Список элементов свойств в AndroidManifest

№	Название элемента	Описание
1	uses-permission	используется для указания разрешений, запрашиваемых в целях безопасности
2	permission	используется для установки разрешений на управление доступом для определенного компонента приложения
3	permission-group	используется для установки разрешений на управление доступом для набора компонентов
4	permission-tree	ссылается на одно конкретное имя компонента, который является владельцем или родителем набора компонентов
5	instrumentation	позволяет узнать взаимодействие между системой Android и приложением
6	uses-sdk	указывает совместимость платформы приложения

7	uses-configuration	задает набор аппаратных и программных требований приложения
8	uses-feature	определяет одно требование к оборудованию и программному обеспечению и связанную с ними сущность
9	supports-screens, compatible-screens	оба этих тега устанавливают режим настройки экрана и размер экрана и так далее
10	supports-gl-texture	определяет текстуру, на основе которой фильтруется приложение

Таблица 3.2 Элементы для компонентов приложения

№	Название элемента	Описание
1	activity	имеет набор атрибутов на основе пользовательского интерфейса
2	receiver	это позволяет получать сообщения, транслируемые одним и тем же приложением или внешним объектом
3	service	имеет операцию, предоставляемую любой библиотекой или API, работает в фоновом режиме, который не виден
4	provider	предоставляет некоторую структуру для доступа к данным приложения
5	activity-alias	задает целевые действия
6	uses-library	указывает набор файлов библиотеки, необходимых для запуска приложения

Все элементы из таблицы 3.2 должны быть заключены в контейнер `<application>`.

Вся эта информация должна быть известна системе для запуска любого файла приложения. Так что этот файл должен быть создан в момент установки, а не во время запуска приложения.

AndroidManifest разрабатываемого приложения (рис. 3.2.1) должен включать в себя разрешение на использование интернета, доступ к точному местоположению, доступ к Google Maps (строки 5, 6, 16 – 18), а также две активности: MainActivity (строки 20 – 26) и MapActivity (строка 27).

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.deatheek.school">
4
5     <uses-permission android:name="android.permission.INTERNET"/>
6     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
7
8     <application
9         android:allowBackup="true"
10        android:icon="@mipmap/ic_launcher"
11        android:label="Школа им. Софьи Ковалевской"
12        android:roundIcon="@mipmap/ic_launcher_round"
13        android:supportsRtl="true"
14        android:theme="@style/AppTheme">
15
16        <meta-data
17            android:name="com.google.android.geo.API_KEY"
18            android:value="AIzaSyD8VpxfPFiTw7cdCbJyVPfhS9HitYHMa78" />
19
20        <activity android:name=".MainActivity">
21            <intent-filter>
22                <action android:name="android.intent.action.MAIN" />
23
24                <category android:name="android.intent.category.LAUNCHER" />
25            </intent-filter>
26        </activity>
27        <activity android:name=".MapActivity"></activity>
28    </application>
29
30 </manifest>
```

Рис. 3.2.1 AndroidManifest разрабатываемого приложения

3.3 Сборка приложения

Gradle — система автоматической сборки приложений для Android, которая построена на принципах Apache Ant и Apache Maven. В Eclipse используется система Ant, но большая часть разработчиков даже не замечает её присутствие. В Android Studio сборщик работает по-другому. Gradle

сопровождает разработчиков во время создания приложений постоянно. Если предыдущей средой разработки был Eclipse, то Gradle может показаться раздражительным, однако он несомненно имеет много положительного.

Gradle не является изобретением Android Studio, этот сборщик приложений был разработан гораздо раньше и использовался для приложений под Java, Scala и другие языки.

Систему сборки Grable невозможно описать в нескольких словах, так как она является очень сложной и для ее разбора написано не мало книг, но для того, чтобы с ней работать необязательно все это знать. Достаточно научиться оперировать с синтаксисом Groove, на котором пишутся команды для Gradle.

Для начала нужно найти файл build.gradle в корневом каталоге проекта. Это называется сборкой верхнего уровня ((project-level) build.gradle). Он содержит настройки, которые применяются ко всем модулям проекта. На рисунке 3.3.1 представлен скриншот этого файла в разрабатываемом приложении.

```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2
3 buildscript {
4     repositories {
5         google()
6         jcenter()
7     }
8 }
9 dependencies {
10     classpath 'com.android.tools.build:gradle:3.3.2'
11
12     // NOTE: Do not place your application dependencies here; they belong
13     // in the individual module build.gradle files
14 }
15
16
17 allprojects {
18     repositories {
19         google()
20         jcenter()
21     }
22 }
23
24
25 task clean(type: Delete) {
26     delete rootProject.buildDir
27 }
28
```

Рис. 3.3.1 Содержание файла build.gradle

Вот что происходит, шаг за шагом:

В блоке `buildscript` определяются параметры, необходимые для выполнения построения проекта.

В блоке `repositories` вы добавляете имена хранилищ, в которых Gradle должен искать используемые вами библиотеки.

Блок `dependencies` содержит необходимые зависимости плагинов, в данном случае плагины Gradle и Kotlin. В этот блок нельзя помещать зависимости модулей.

Структура блока `allprojects` аналогична структуре блока `buildscript`, но здесь определяются `repositories` для всех ваших модулей, а не для самого Gradle. Обычно раздел зависимостей для `allprojects` не определяется. Зависимости для каждого модуля различны и должны находиться в сборке уровня модуля.градля.

Теперь рассмотрим `build.gradle` файл в каталоге модуля приложения. Он содержит зависимости (библиотеки, на которые опирается модуль) и инструкции для процесса сборки. Каждый модуль определяет свою собственную `build.gradle`. Полученный код показан на рисунке 3.3.2.

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 26
5      defaultConfig {
6          applicationId "com.deatheek.school"
7          minSdkVersion 16
8          targetSdkVersion 26
9          versionCode 1
10         versionName "1.0"
11         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12     }
13     buildTypes {
14         release {
15             minifyEnabled false
16             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
17         }
18     }
19 }
20
21 dependencies {
22     implementation fileTree(dir: 'libs', include: ['*.jar'])
23     implementation 'com.android.support:appcompat-v7:26.1.0'
24     implementation 'com.android.support:design:26.1.0'
25     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
26     testImplementation 'junit:junit:4.12'
27     androidTestImplementation 'com.android.support.test:runner:1.0.2'
28     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
29     implementation 'com.google.android.gms:play-services-maps:16.1.0'
30     // volley http library
31     implementation 'com.android.volley:volley:1.1.0'
32     implementation 'com.google.code.gson:gson:2.6.2'
33 }
34
```

Рис. 3.3.2 Содержание файла build.gradle

Приведенный выше код выполняет следующие действия:

Задаёт список плагинов, необходимых для построения модуля. Com.android.application необходим для того, чтобы настроить специфические настройки процесса сборки. Здесь можно также использовать com.android.library, если разрабатывается библиотечный модуль.

В блоке android размещены все специфические для платформы опции модуля.

Параметр compileSdkVersion указывает уровень API, с которым будет скомпилировано приложение. Другими словами, нельзя использовать функции из API выше этого значения.

Параметр buildToolsVersion указывает версию компилятора. Начиная с Gradle plugin 3.0.0, это поле является необязательным. Если он не указан, Android SDK использует самую последнюю загруженную версию инструментов сборки.

Блок `defaultConfig` содержит параметры, которые будут применяться ко всем версиям сборки (например, `debug`, `release` и т. д.) вашего приложения по умолчанию.

`ApplicationId` – это идентификатор приложения. Он должен быть уникальным, чтобы успешно публиковать или обновлять приложение в Google Play Store.

Чтобы установить самый низкий поддерживаемый уровень API, использовался `minSdkVersion`. Приложение не будет доступно в Play Store для устройств, работающих на более низких уровнях API.

Параметр `targetSdkVersion` определяет максимальный уровень API, на котором было протестировано приложение, то есть приложение работает правильно на устройствах с этой версией SDK и не требует поведения обратной совместимости. Наилучший подход заключается в том, чтобы тщательно протестировать приложение с использованием новейшего API, сохраняя свой `targetSdkVersion` значение, равное `compileSdkVersion`.

`versionCode` – это числовое значение для версии приложения.

`versionName` – это удобная строка для версии приложения.

Блок `dependencies` содержит все зависимости, необходимые для данного модуля. В этот блок было дописано немало зависимостей, например:

- 'com.google.android.gms:play-services-maps:16.1.0'
- 'com.android.volley:volley:1.1.0'
- 'com.google.code.gson:gson:2.6.2'

Которые позволили приложению использовать Google Maps и другие сервисы.

3.4 Активности в Android приложении

При работе с языком программирования C, C++ или Java можно увидеть, что ваша программа начинается с функции `main ()`. Очень похожим образом система Android иницирует свою программу с помощью действия,

начинающегося с вызова метода обратного вызова onCreate (). Существует последовательность методов обратного вызова, которые запускают действие и последовательность методов обратного вызова, которые срывают действие, как показано на рисунке 3.4.1.

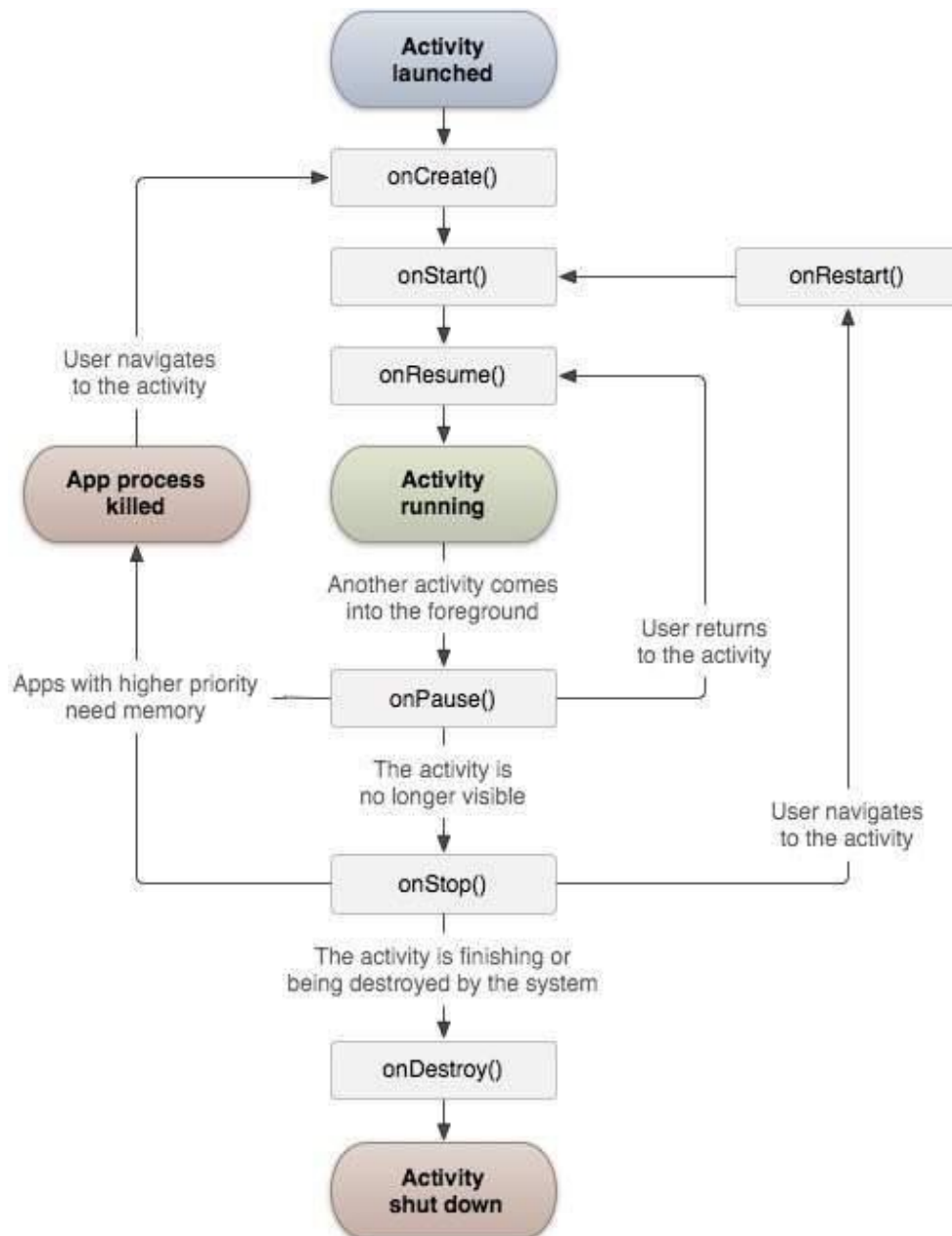


Рис. 3.4.1 Жизненный цикл активности Android

Класс Activity определяет следующие обратные вызовы, т. е. события. Вам не нужно реализовывать все методы обратного вызова. Однако важно

понимать каждый из них и реализовать те, которые гарантируют, что приложение будет вести себя так, как ожидают пользователи.

Таблица 3.3 Методы жизненного цикла

№	Название	Описание
1	onCreate()	Это первый метод и вызывается при первом создании действия
2	onStart()	Этот метод вызывается, когда действие становится видимым для пользователя
3	onResume()	Это вызывается, когда пользователь начинает взаимодействовать с приложением
4	onPause()	Приостановленное действие не получает ввода пользователя и не может выполнить код и вызывается при приостановке текущего действия и возобновлении предыдущего действия
5	onStop()	Этот метод вызывается, когда активность больше не видна
6	onDestroy()	Этот метод до того, как действие будет уничтожено системой
7	onRestart()	Этот метод вызывается при перезапуске действия после его остановки

Разберем все вышесказанное на примере создаваемого приложения. После открытия приложения запускается стартовая активность, указанная в манифесте, то есть MainActivity. Изначально эта активность имеет всего один метод – это onCreate и выглядит он, как представлено на рисунке 3.4.2.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
}

```

Рис. 3.4.2 Метод onCreate

В дальнейшем в MainActivity будет добавлен элемент fragment, содержащий контент, а также меню, через которое будет осуществляться навигация приложения, в том числе и переход на MapActivity.

В MapActivity были подключены следующие библиотеки com.google.android.gms.maps.*, а также переписан метод onMapReady, который при запуске активности открывает карту, находит и отмечает на ней НИУ БелГУ. Для того, чтобы это заработало, необходимо зарегистрироваться на ресурсе Google, создать свой проект и подключить необходимые API, в данном случае – Android Maps API (рис. 3.4.3).

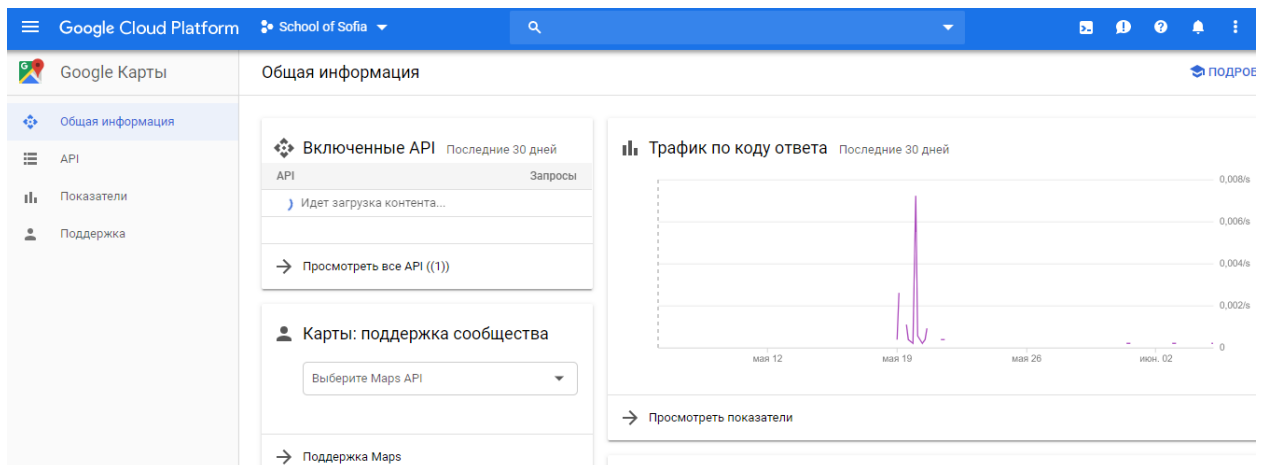


Рис. 3.4.3 Созданный проект в Google и подключенный API

Далее нужно получить ключ и передать его приложению, таким образом Google начнет передавать нашей программе необходимые данные. Итоговый результат изображен на рисунке 3.4.4.



Рис. 3.4.4 MapActivity

3.5 Разметка приложения Android

Android обеспечивает надежную поддержку для разработки приложений на основе пользовательского интерфейса. Android предоставляет различные виджеты, которые программист может использовать для создания желаемого макета и интерфейса. Эти элементы могут быть созданы с помощью языка программирования напрямую, или через файлы XML-макета. Рассмотрим оба метода и выделим их отличия.

В Android XML-макет-это файл, который определяет различные виджеты, используемые в пользовательском интерфейсе, и отношения между этими виджетами и их контейнерами. Android относится к layout файлов в качестве ресурсов. Следовательно, макеты хранятся в папке layout. Если используется Android Studio, она создает файл макета XML по умолчанию (activity_main.xml) в папке layout, которая выглядит как следующий XML-код. Файлы макета выступают в качестве входных данных для инструмента Android Asset Packaging Tool (AAPT), который создает R.java файл для всех ресурсов.

Макеты на основе XML очень полезны, если знать компоненты пользовательского интерфейса во время компиляции. Если необходимы компоненты пользовательского интерфейса во время выполнения, их можно добавить с помощью XML-кода.

XML-макет имеет следующие преимущества:

- XML-очень популярный и широко используемый формат. Таким образом, многие разработчики вполне комфортно с ним;
- Обеспечивает отделение пользовательского интерфейса от логики кода. Это обеспечило гибкость для изменения одного без существенного влияния на другое;
- Создание XML-вывода проще, чем написание прямого кода, что упрощает использование инструментов перетаскивания для создания интерфейсов для приложений android.

Пользовательский интерфейс в Android – это иерархия групп представлений и представлений. Группы представлений будут промежуточными узлами в иерархии, а представления-терминальными узлами. Пример кода продемонстрирован на рисунке 3.5.1.

```
1 <ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:style="http://schemas.android.com/tools"
3     android:id="@+id/scroll"
4     android:layout_width="fill_parent"
5     android:layout_height="wrap_content">
6     <HorizontalScrollView
7         android:id="@+id/hscroll"
8         android:layout_width="fill_parent"
9         android:layout_height="fill_parent">
10
11     <LinearLayout
12         android:layout_width="fill_parent"
13         android:layout_height="wrap_content"
14         android:orientation="vertical">
15         <TextView
16
17             android:id="@+id/textviewTitle1"
18             android:layout_width="wrap_content"
19             android:layout_height="wrap_content"
20             android:textSize="26sp"
21             android:isScrollContainer="true"
22             android:textStyle="bold"
23             android:layout_marginLeft = "15sp"
24             android:layout_marginTop = "27sp"/>
25         <TextView
26             android:id="@+id/textviewT1"
27             android:layout_width="wrap_content"
28             android:layout_height="wrap_content"
29             android:textSize="18sp"
30             android:isScrollContainer="true"
31             android:textStyle="italic"
32             android:layout_marginLeft = "15sp"/>
```

Рис. 3.5.1 Код fragment_four.xml

Например, в приведенном выше fragment_four.xml-файл, LinearLayout – это viewgroup, а TextView – это представление.

Android предоставляет следующие стандартные макеты (viewgroups), которые можно использовать в приложении Android:

- AbsoluteLayout
- FrameLayout
- LinearLayout
- RelativeLayout
- TableLayout

В Absolute Layout, можно указать точные координаты каждого элемента, который мы хотим разместить. В абсолютной компоновке мы дадим точные координаты X и Y каждого элемента управления.

LinearLayout используется для размещения одного элемента на каждой строке. Таким образом, все элементы будут размещены упорядоченно сверху вниз. Это очень широко используемый макет для создания форм на Android.

Используя Relative Layout, мы можем указать положение элементов по отношению к другим элементам или по отношению к родительскому контейнеру.

Чтобы указать положение относительно родительского контейнера, мы используем `android:layout_alignParentTop = "true"` и `android:layout_alignParentLeft = "true"` для выравнивания элементов в левом верхнем углу родительского контейнера. Затем, чтобы выровнять по отношению к другому элементу, мы можем использовать свойства `android:layout_alignLeft = "@+id/otherelement"` и `android:layout_below="@+id/otherelement"`.

В разрабатываемом приложении при разметке `fragment_four` использовались следующие макеты:

- Scroll View – контейнер компоновки для иерархии представлений, которая может прокручиваться пользователем, позволяя ей быть больше физического отображения. HorizontalScrollView – это `FrameLayout`, то есть вы должны поместить в него один объект, содержащий все содержимое для прокрутки; этот объект может сам быть менеджером макетов со сложной иерархией объектов. В нашем случае используется дочерний элемент `LinearLayout` в горизонтальной ориентации, представляющий горизонтальный массив элементов верхнего уровня, которые пользователь может прокручивать;

- Linear Layout – описан выше;
- Text View – контейнер для отображения текста;
- Image View – контейнер для отображения изображений.

Полученный результат изображен на рисунке 3.5.2.



Рис. 3.5.2 Полученный результат

3.6 Создание меню

Android Bottom Navigation View-это еще один способ навигации по связанным страницам в мобильном приложении для android. Он подходит для навигации между 3 до 5 страниц. Если планируется добавить более 5 различных страниц активности или фрагментов в пункт назначения верхнего уровня, настоятельно рекомендуется использовать Android Navigation Drawer.

Ниже приводится подробное описание того, какие материальные рекомендации говорят нам о том, где должен использоваться `BottomNavigationView` для поддержания материального дизайна в приложении:

- `BottomNavigationView` следует использовать для обеспечения быстрой навигации между видами верхнего уровня приложения. Только те пункты назначения, которые требуют прямого доступа, должны быть в `BottomNavigationView`.

- В `BottomNavigationView` может быть только от 3 до 5 пунктов назначения верхнего уровня. Если есть 1 или 2 назначения верхнего уровня, тогда следует использовать `TabLayout`. Если в приложении больше 5, перейти к боковой панели ящика.

- Каждый параметр должен иметь ширину представления, разделенную на количество действий (максимум 168dp и минимум 80dp). Высота 56dp и каждый значок 24 x 24dp. Он должен иметь заполнение 6dp над значком (активный вид), 8dp над значком (неактивный вид), 10dp под текстом, 12dp слева и справа от текста. Он должен иметь высоту 8dp. Поскольку закусочные имеют более низкую высоту (6dp), они появляются за нижней навигационной панелью.

- Активный значок должен быть окрашен основным цветом приложения. Черный или белый цвет должен использоваться, если нижняя панель навигации уже окрашена.

- Если представление находится в фокусе, должны отображаться значок и текстовая метка представления. Если есть только три действия, значки и текстовые метки должны отображаться в любое время. Если есть четыре или пять действий, неактивные представления должны отображаться только в виде значков.

- Навигация по BottomNavigationView должна сбросить состояние задачи. Если пользователь нажимает на активное действие, он должен перейти к пользователю в верхней части представления.

Разработчики должны быть осторожны при объединении нижней навигации с вкладками. Это может вызвать путаницу при навигации по приложению.

Для меню приложения были нарисованы иконки в программе Adobe Photoshop (рис. 3.6.1).

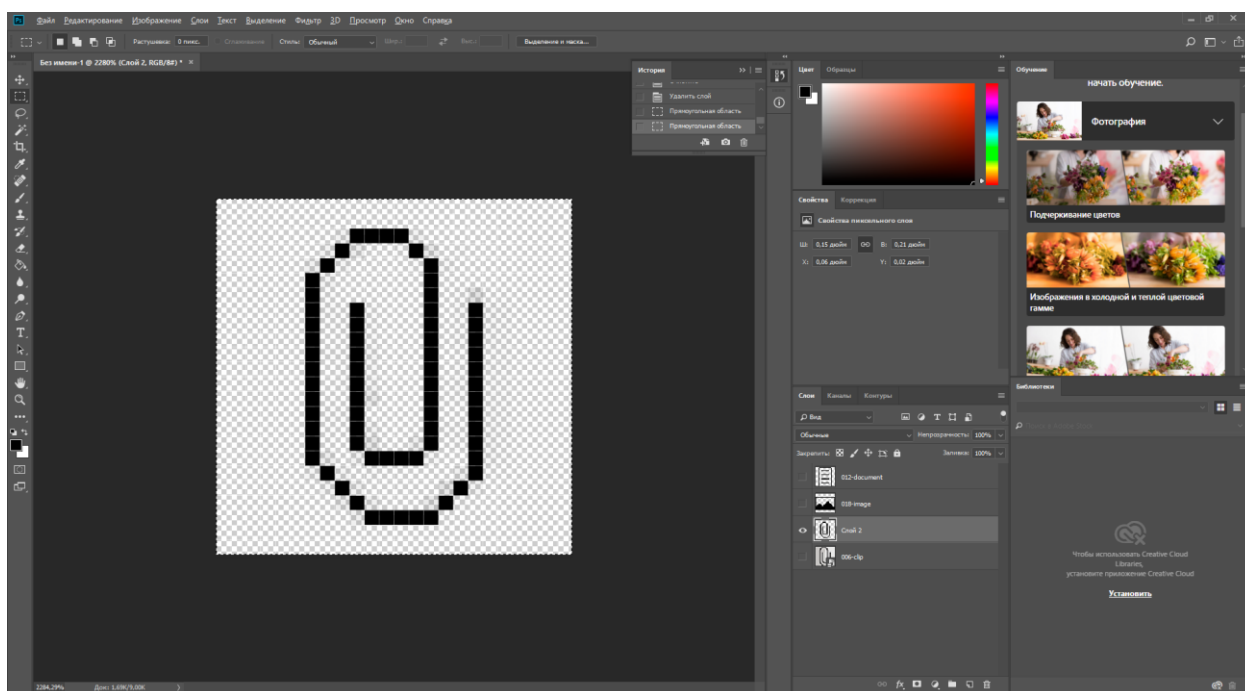


Рис. 3.6.1 Процесс создания иконок для меню

Далее в navigation.xml эти иконки были подключены к меню (рис. 3.6.2). Конечный результат представлен на рисунке 3.6.3.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3
4
5     <item
6         android:id="@+id/navigation_one"
7         android:icon="@drawable/ic_home_black_24dp"
8         android:title="Главная" />
9
10    <item
11        android:id="@+id/navigation_two"
12        android:icon="@drawable/news2_24dp"
13        android:title="Новости" />
14
15    <item
16        android:id="@+id/navigation_three"
17        android:icon="@drawable/archive_24dp"
18        android:title="Архив" />
19
20    <item
21        android:id="@+id/navigation_four"
22        android:icon="@drawable/ic_person_white_24dp"
23        android:title="Преподаватели" />
24
25    <item
26        android:id="@+id/navigation_map"
27        android:icon="@drawable/ic_dashboard_black_24dp"
28        android:title="" />
29
30 </menu>

```

Рис. 3.6.2 Navigation.xml

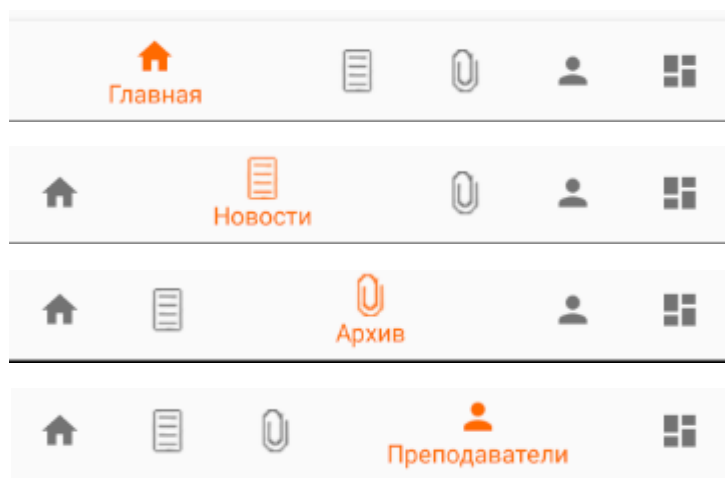


Рис. 3.6.3 Меню

3.7 Фрагменты

Android инициировал фрагменты в Android 3.0 (API level 11), главным образом для поддержки более динамичных и гибких дизайнов пользовательского интерфейса.

Фрагменты позволяют такие конструкции без требования, чтобы управлять сложными изменениями в порядке просмотра клевания. Разделив макет действия на фрагменты, позволяется изменять внешний вид действия во время выполнения и сохранять эти изменения в заднем стеке, управляемом действием.

Фрагмент не работает сам по себе. Каждый фрагмент должен быть размещен классом Activity, хотя у фрагмента есть свой жизненный цикл, но он возобновляется, когда его активность хоста возобновляется.

Фрагмент может быть добавлен, удален и заменен в классе Activity, поэтому его использование помогает решить некоторую начальную проблему с активностью, например, когда разные представления в одном и том же действии хотят передавать данные друг другу.

В следующем примере мы рассмотрим, как фрагменты взаимодействуют с активностью хоста. Существует несколько способов достижения этого, например, вызов метода `getActivity ()` в классе `Fragment` для получения экземпляра действия хоста. После получения дескриптора действия можно вызвать любые методы доступа класса `Activity`.

В нашем приложении создано пять фрагментов (рис. 3.7.1) для пяти окон, которые управляются файлами `java` (рис. 3.7.2). Переключение между фрагментами происходит по довольно простому алгоритму, представленному на рисунке 3.7.3.

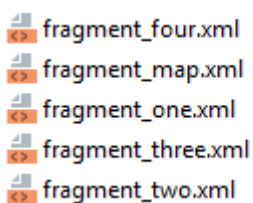


Рис. 3.7.1 Файлы фрагментов

- ▼ fragment
 - FragmentFour
 - FragmentMap
 - FragmentOne
 - FragmentThree
 - FragmentTwo

Рис. 3.7.2 Файлы java

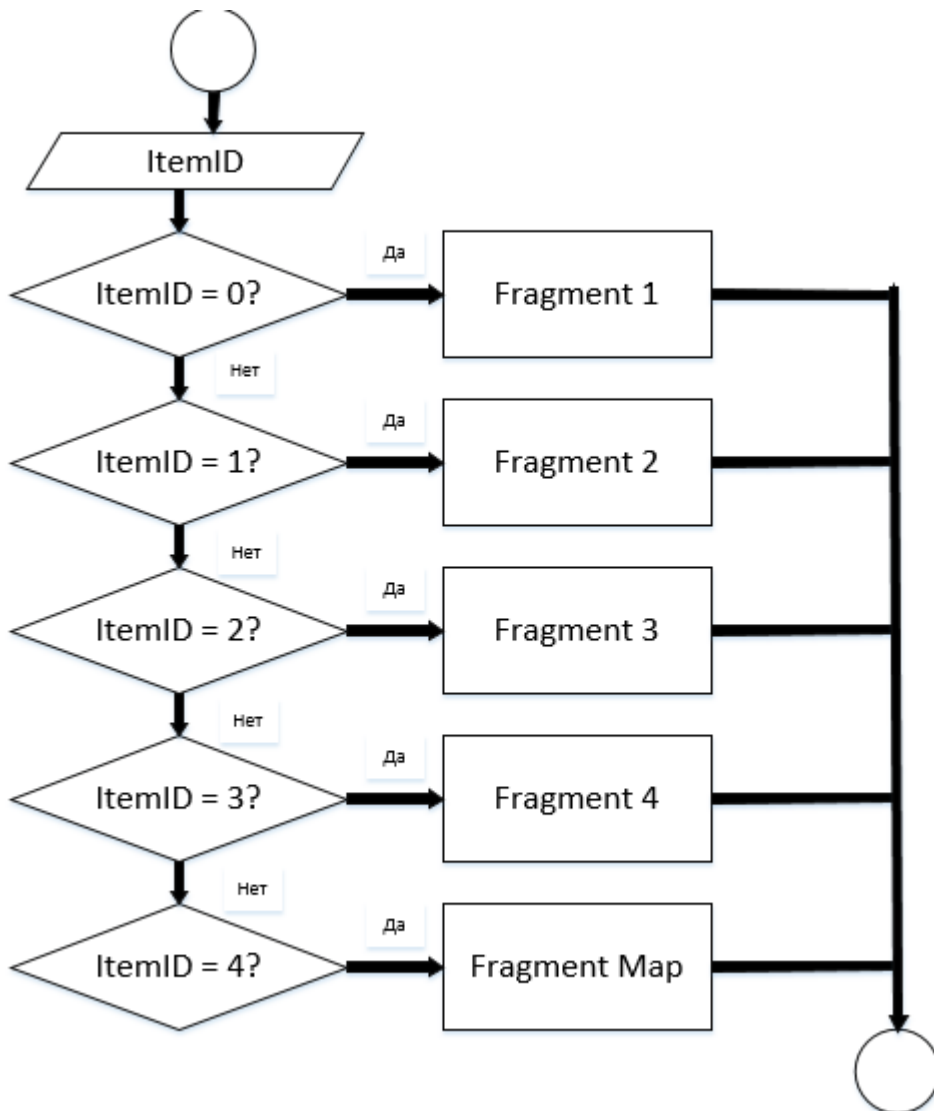


Рис 3.7.3 Блок-схема алгоритма меню

В данном алгоритме отсутствует проверка условия на выход значения из диапазона 0-4, так как появление данного события невозможно.

3.8 Темы для приложения Android

Для новичков в разработке Android и Material Design, управление темой и стилем приложения может быть немного запутанным. В Android Studio 1.4 был выпущен новый инструмент, называемый Theme Editor, чтобы помочь этой задаче (рисунок 3.8.1).

Редактор тем – это графический инструмент. Эта первая версия фокусируется на редактировании и обновлении цветов темы материала. Изменения автоматически обновляются и обновляются в colors.xml.

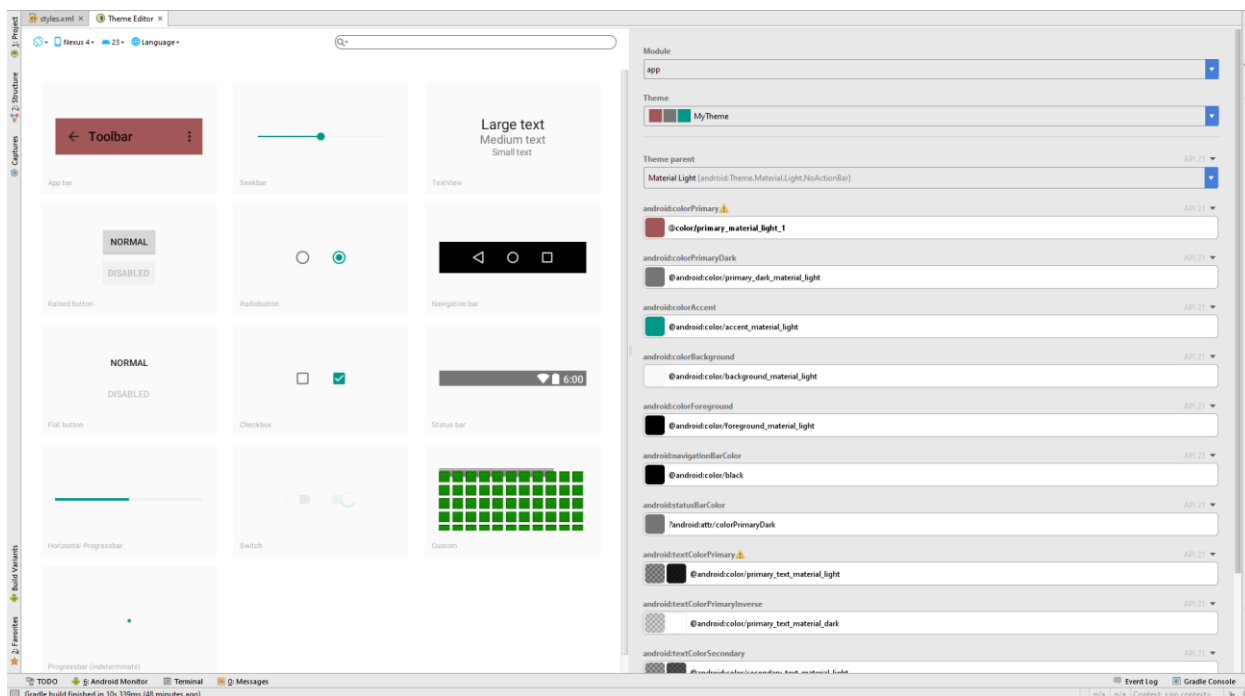


Рис. 3.8.1 Theme Editor

В нашем приложении настроены две темы – дневная и ночная. Их смену было решено производить автоматически, ориентируясь на текущее время с учетом часовых поясов. Пример дневной и ночной темы представлен ниже на рисунке 3.8.2.

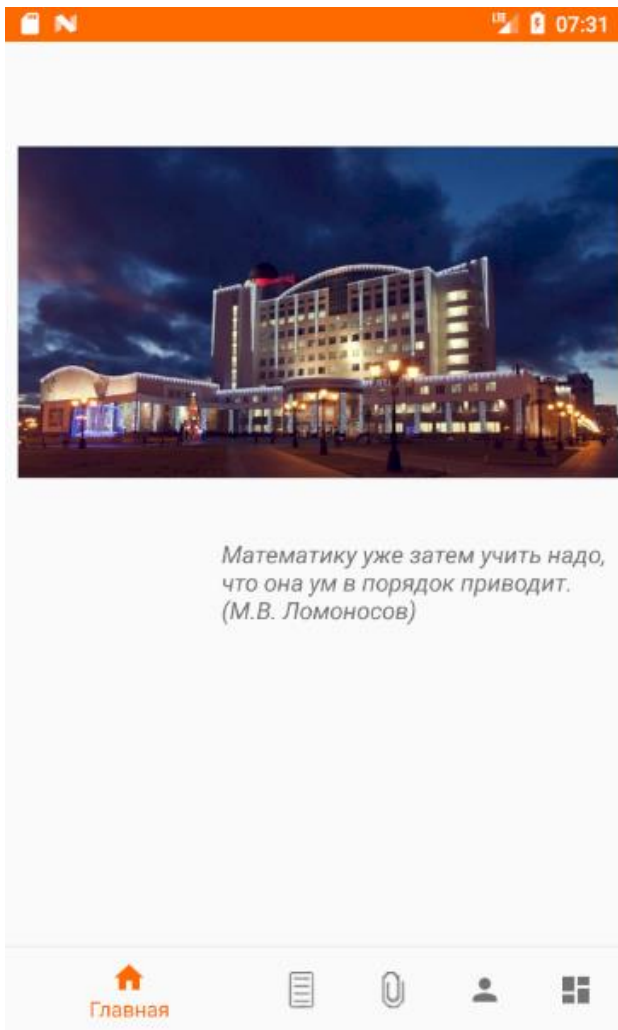


Рис. 3.8.2 Дневная и ночная темы приложения

ЗАКЛЮЧЕНИЕ

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

- было найдено и проанализировано шесть аналогичных приложений, доступных на Google Play Market, которые были выполнены с учетом уже старых графических решений, а также обладали перегруженностью интерфейса, что тоже является недостатком;
- в результате оценки количества устройств с различными операционными системами на рынке, была выделена наиболее популярная версия;
- для разработки мобильного приложения для ОС Android из трех наиболее распространенных сред была выбрана Android Studio, отличающаяся удобным интерфейсом и хорошими функциональными возможностями;
- с помощью Android Studio было разработано приложение, удовлетворяющее требованиям технического задания;

Данное приложение состоит из пяти активных вкладок, которые содержат различную информацию о заочной школе такую, как информацию о преподавательском составе, местоположении школы и так далее. Для приложения был создан свой дизайн, который также предусматривает смену дневной и ночной тем, что позволяет пользоваться этим приложением с большим комфортом для глаз.

Все поставленные задачи в работе были выполнены, разработано приложение заочной математической школы имени Софьи Ковалевской для операционной системы Android.

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-](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/60-urok-23-activity-lifecycle-v-kakih-sostojaniyah-mozhet-byt-activity.html)

[activity-lifecycle-v-kakih-sostojaniyah-mozhet-byt-activity.html](http://startandroid.ru/ru/uroki/vse-uroki-spiskom/60-urok-23-activity-lifecycle-v-kakih-sostojaniyah-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.

```

package com.deatheek.school;

import android.content.Intent;
import android.support.annotation.NonNull;
import
android.support.design.widget.BottomNavigationView;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentTransaction;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.app.AppCompatActivity;
import android.view.MenuItem;

import com.deatheek.school.fragment.FragmentFour;
import com.deatheek.school.fragment.FragmentOne;
import com.deatheek.school.fragment.FragmentThree;
import com.deatheek.school.fragment.FragmentTwo;

public class MainActivity extends AppCompatActivity
{

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        //toolbar = getSupportActionBar();

        BottomNavigationView navigation =
        (BottomNavigationView)
        findViewById(R.id.navigation);

        navigation.setOnNavigationItemSelectedListener(mOn
        NavigationItemSelectedListener);

        // attaching bottom sheet behaviour - hide / show
        on scroll
        //CoordinatorLayout.LayoutParams layoutParams
        =
        (CoordinatorLayout.LayoutParams)
        navigation.getLayoutParams();
        //layoutParams.setBehavior(new
        BottomNavigationViewBehavior());
    }
}

```

```

// Check that the activity is using the layout version
with the fragment_container FrameLayout
getDelegate().setLocalNightMode(
AppCompatActivity.MODE_NIGHT_AUTO);

// load the store fragment by default
//toolbar.setTitle("Fragment 1");
loadFragment(new FragmentOne());
}

private
BottomNavigationView.OnNavigationItemSelectedListener
mOnNavigationItemSelectedListener
=
new
BottomNavigationView.OnNavigationItemSelectedListener() {

    @Override
    public boolean
onNavigationItemSelectedListener(@NonNull MenuItem item)
{
    Fragment fragment;
    switch (item.getItemId()) {
        case R.id.navigation_one:
            //toolbar.setTitle("Fragment 1");
            fragment = new FragmentOne();
            loadFragment(fragment);
            return true;
        case R.id.navigation_two:
            //toolbar.setTitle("Fragment 2");
            fragment = new FragmentTwo();
            loadFragment(fragment);
            return true;
        case R.id.navigation_three:
            //toolbar.setTitle("Fragment 3");
            fragment = new FragmentThree();
            loadFragment(fragment);
            return true;
        case R.id.navigation_four:
            //toolbar.setTitle("Fragment 4");
            fragment = new FragmentFour();
            loadFragment(fragment);
            return true;
        case R.id.navigation_map:

```

```

        Intent intentMap = new
Intent(MainActivity.this, MapActivity.class);
        startActivity(intentMap);
        break;
    }

    return false;
}
};

/**
 * loading fragment into FrameLayout
 *
 * @param fragment
 */
private void loadFragment(Fragment fragment) {
    // load fragment
    FragmentTransaction transaction =
getSupportFragmentManager().beginTransaction();
    transaction.replace(R.id.frame_container,
fragment);
    transaction.addToBackStack(null);
    transaction.commit();
}
}
package com.deatheek.school;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

import
com.google.android.gms.maps.CameraUpdateFactory;

import com.google.android.gms.maps.GoogleMap;

import
com.google.android.gms.maps.OnMapReadyCallback;

import
com.google.android.gms.maps.SupportMapFragment;

import com.google.android.gms.maps.model.LatLng;

import
com.google.android.gms.maps.model.MarkerOptions;

public class MapActivity extends AppCompatActivity
implements OnMapReadyCallback {

    private GoogleMap mMap;

    @Override

    protected void onCreate(Bundle savedInstanceState)
{

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.fragment_map);

// Obtain the SupportMapFragment and get
notified when the map is ready to be used.

SupportMapFragment mapFragment =
(SupportMapFragment) getSupportFragmentManager()

.findFragmentById(R.id.map);
mapFragment.getMapAsync(this);
}

/**
 * Manipulates the map once available.
 * This callback is triggered when the map is ready to
be used.
 * This is where we can add markers or lines, add
listeners or move the camera. In this case,
 * we just add a marker near Sydney, Australia.
 * If Google Play services is not installed on the
device, the user will be prompted to install
 * it inside the SupportMapFragment. This method
will only be triggered once the user has
 * installed Google Play services and returned to the
app.
 */
@Override
public void onMapReady(GoogleMap googleMap) {
    mMap = googleMap;

    LatLng belgu = new LatLng(50.593488,
36.575734);
    mMap.addMarker(new
MarkerOptions().position(belgu).title("НИУ БелГУ"));

mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(belgu, 17));

}
}
package com.deatheek.school.fragment;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;

```

```

import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.deatheek.school.R;

public class FragmentFour extends Fragment {

    public FragmentFour() {
        // Required empty public constructor
    }

    public static FragmentOne newInstance(String
param1, String param2) {
        FragmentOne fragment = new FragmentOne();
        Bundle args = new Bundle();
        fragment.setArguments(args);
        return fragment;
    }

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

    }

    @Override
    public View onCreateView(LayoutInflater inflater,
ViewGroup container,
        Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        View view =
inflater.inflate(R.layout.fragment_four,
        container, false);

        TextView textviewtitle1 = (TextView)
view.findViewById(R.id.textviewTitle1);
        textviewtitle1.setText("Аверин Геннадий
Викторович");

        TextView textviewtitle2 = (TextView)
view.findViewById(R.id.textviewTitle2);
        textviewtitle2.setText("Борисовский Иван
Петрович");

        TextView textview2 = (TextView)
view.findViewById(R.id.textviewT2);
        textview2.setText("Преподаватель
математики");

        ImageView image2 = (ImageView)
view.findViewById(R.id.image2);
        image2.setImageResource(R.drawable.image2);

        TextView textview2 = (TextView)
view.findViewById(R.id.textview2);
        textview2.setText("Образовательное
учреждение: \n" +
        "Белгородский государственный \n" +
        "педагогический институт \n" +
        "им. М.С. Ольминского, 1992\n" +
        "Специальность: математика\n" +
        "Квалификация: учитель математики и \n"
+
        "Николаевский ордена Трудового
Красного \n" +
        "Знамени кораблестроительный институт
\n" +
        "им. адм. С.О. Макарова, 1980\n" +
        "Специальность: холодильные и \n" +
        "компрессорные машины и установки\n" +
        "Квалификация: инженер-механик\n" +
        "доктор технических наук, профессор\n" +
        "Должность: заведующий кафедрой общей
\n" +
        "математики, профессор\n" +
        "Опыт преподавания: 19 лет");

        TextView textview1 = (TextView)
view.findViewById(R.id.textviewT1);
        textview1.setText("Преподаватель
математики");

        ImageView image1 = (ImageView)
view.findViewById(R.id.image1);
        image1.setImageResource(R.drawable.image1);

        TextView textview1 = (TextView)
view.findViewById(R.id.textview1);
        textview1.setText("Образовательное
учреждение: \n" +

```

```

\n" +
        "информатики и вычислительной техники
\n" +
        "и звание учитель средней школы\n" +
\n" +
        "кандидат физико-математических наук,
\n" +
        "доцент\n" +
        "Должность: заведующий кафедрой\n" +
        "математики, доцент\n" +
        "Опыт преподавания: 23 года");

```

```

        TextView textViewTitle3 = (TextView)
view.findViewById(R.id.textViewTitle3);
        textViewTitle3.setText("Витохина Наталья
Николаевна");

```

```

        TextView textView3 = (TextView)
view.findViewById(R.id.textView3);

```

```

        textView3.setText("Преподаватель
математики");

```

```

        TextView textView3 = (TextView)
view.findViewById(R.id.textView3);

```

```

        textView3.setText("Образовательное
учреждение: \n" +

```

```

        "Белгородский государственный \n" +
        "университет, 2002\n" +
        "Специальность: математика\n" +
        "Квалификация: учитель математики \n" +
        "и информатики\n" +
        "кандидат физико-математических наук\n"
+
        "Должность: доцент кафедры математики
\n" +
        "Опыт преподавания: 15 лет");

```

```

        ImageView image3 = (ImageView)
view.findViewById(R.id.image3);
        image3.setImageResource(R.drawable.image3);

```

```

        TextView textViewTitle4 = (TextView)
view.findViewById(R.id.textViewTitle4);
        textViewTitle4.setText("Красовская Людмила
Владимировна");

```

```

        TextView textView4 = (TextView)
view.findViewById(R.id.textView4);

```

```

        textView4.setText("Преподаватель
информатики");

```

```

        TextView textView4 = (TextView)
view.findViewById(R.id.textView4);

```

```

        textView4.setText("Образовательное
учреждение: \n" +
\n" +
        "Дагестанский государственный
\n" +
        "технический" +
        "университет, 2000\n" +
        "Специальность: вычислительные
машины, \n" +
        "комплексы, системы и сети\n" +
        "Квалификация: инженер\n" +
        "кандидат технических наук, доцент\n" +
        "Должность: заведующий кафедрой\n" +
        "информатики, естественнонаучных \n" +
        "дисциплин и методик \n" +
        "преподавания, доцент \n" +
        "Опыт преподавания: 18 лет");

```

```

        ImageView image4 = (ImageView)
view.findViewById(R.id.image4);
        image4.setImageResource(R.drawable.image4);

```

```

        TextView textViewTitle5 = (TextView)
view.findViewById(R.id.textViewTitle5);

```

```

        textViewTitle5.setText("Некрасова Ирина
Викторовна");

```

```

        TextView textView5 = (TextView)
view.findViewById(R.id.textView5);

```

```

        textView5.setText("Преподаватель
математики");

```

```

        TextView textView5 = (TextView)
view.findViewById(R.id.textView5);

```

```

        textView5.setText("Образовательное
учреждение: \n" +
        "Белгородский государственный \n" +
        "университет, 2001\n" +
        "Специальность: математика\n" +
        "Квалификация: учитель математики \n" +
        "и информатики\n" +
        "кандидат физико-математических наук\n"
+
        "Должность: доцент кафедры общей \n" +
        "математики \n" +
        "Опыт преподавания: 17 лет");

```

```

        ImageView image5 = (ImageView)
view.findViewById(R.id.image5);
        image5.setImageResource(R.drawable.image5);

```

```

        TextView textViewtitle6 = (TextView)
view.findViewById(R.id.textViewTitle6);

        textViewtitle6.setText("Остапенко Светлана
Ивановна");

        TextView textView6 = (TextView)
view.findViewById(R.id.textViewT6);

        textView6.setText("Преподаватель
математики");

        TextView textView6 = (TextView)
view.findViewById(R.id.textView6);

        textView6.setText("Образовательное
учреждение: \n" +

                "Белгородский государственный \n" +
                "университет, 2007\n" +

                "Специальность: математика\n" +

                "Квалификация: учитель математики и \n"
+

                "информатики" +

                "кандидат педагогических наук,\n" +

                "Должность: доцент кафедры математики
\n" +

                "Опыт преподавания: 11 лет");

        ImageView image6 = (ImageView)
view.findViewById(R.id.image6);

        image6.setImageResource(R.drawable.image6);

        TextView textViewtitle7 = (TextView)
view.findViewById(R.id.textViewTitle7);

        textViewtitle7.setText("Полунин Виктор
Александрович");

        TextView textView7 = (TextView)
view.findViewById(R.id.textViewT7);

        textView7.setText("Преподаватель
математики");

        TextView textView7 = (TextView)
view.findViewById(R.id.textView7);

        textView7.setText("Образовательное
учреждение: \n" +

                "Белгородский государственный \n" +
                "университет, 2006\n" +

                "Специальность: математика\n" +

                "Квалификация: бакалавр\n" +

                "кандидат физико-математических наук\n"
+

                "Должность: доцент кафедры \n" +
                "дифференциальных уравнений \n" +

                "Опыт преподавания: 8 года");

        ImageView image7 = (ImageView)
view.findViewById(R.id.image7);

        image7.setImageResource(R.drawable.image7);

        TextView textViewtitle8 = (TextView)
view.findViewById(R.id.textViewTitle8);

        textViewtitle8.setText("Сатлер Ольга
Николаевна");

        TextView textView8 = (TextView)
view.findViewById(R.id.textViewT8);

        textView8.setText("Преподаватель
информатики");

        TextView textView8 = (TextView)
view.findViewById(R.id.textView8);

        textView8.setText("Образовательное
учреждение: \n" +

                "Карагандинский государственный \n" +
                "технический университет, 2002\n" +

                "Специальность: системы
автоматизированного \n" +

                "проектирования\n" +

                "Квалификация: инженер-
системотехник\n" +

                "кандидат технических наук\n" +

                "Должность: доцент кафедры
информатики,\n" +

                "естественнонаучных дисциплин и \n" +

                "методик преподавания \n" +

                "Опыт преподавания: 15 лет");

        ImageView image8 = (ImageView)
view.findViewById(R.id.image8);

        image8.setImageResource(R.drawable.image8);

        TextView textViewtitle9 = (TextView)
view.findViewById(R.id.textViewTitle9);

        textViewtitle9.setText("Старовойтов Александр
Степанович");

        TextView textView9 = (TextView)
view.findViewById(R.id.textViewT9);

        textView9.setText("Преподаватель
информатики");

        TextView textView9 = (TextView)
view.findViewById(R.id.textView9);

        textView9.setText("Образовательное
учреждение: \n" +

                "Белгородский Государственный \n" +
                "университет, 2000\n" +

                "Специальность: математика\n" +

```



```

+
        "Квалификация: учитель математики и \n"
        "информатики, кандидат физико-
математических \n" +
        "наук, доцент\n" +
        "Должность: доцент кафедры
информатики, \n" +
        "естественнонаучных дисциплин и \n" +
        "методик преподавания\n" +
        "Опыт преподавания: 17 лет");

        ImageView image9 = (ImageView)
view.findViewById(R.id.image9);
        image9.setImageResource(R.drawable.image9);

        TextView textviewtitle10 = (TextView)
view.findViewById(R.id.textviewTitle10);
        textviewtitle10.setText("Флоринский Владимир
Вячеславович");
        TextView textviewt10 = (TextView)
view.findViewById(R.id.textviewT10);
        textviewt10.setText("Преподаватель
математики");
        TextView textview10 = (TextView)
view.findViewById(R.id.textview10);
        textview10.setText("Образовательное
учреждение: \n" +
        "Харьковский государственный \n" +
        "университет им. А.М.Горького, 1977\n"
+
        "Специальность: математика\n" +
        "Квалификация: математик\n" +
        "кандидат физико-математических \n" +
        "наук, доцент\n" +
        "Должность: доцент кафедры общей \n" +
        "математики \n" +
        "Опыт преподавания: 41 год");
        ImageView image10 = (ImageView)
view.findViewById(R.id.image10);
        image10.setImageResource(R.drawable.image10);

        TextView textviewtitle11 = (TextView)
view.findViewById(R.id.textviewTitle11);
        textviewtitle11.setText("Чернявских Светлана
Дмитриевна");
        TextView textviewt11 = (TextView)
view.findViewById(R.id.textviewT11);
        textviewt11.setText("Преподаватель биологии");

```

```

        TextView textview11 = (TextView)
view.findViewById(R.id.textview11);
        textview11.setText("Образовательное
учреждение: \n" +
        "Белгородский государственный \n" +
        "педагогический институт \n" +
        "им. М.С. Ольминского, 1996\n" +
        "Специальность: биология и психология\n"
+
        "Квалификация: учитель биологии, \n" +
        "педагог-психолог и звание учитель \n" +
        "средней школы\n" +
        "кандидат биологических наук, доцент\n"
+
        "Должность: декан факультета математики
и естественнонаучного образования,
доцент\n" +
        "Опыт преподавания: 20 лет");
        ImageView image11 = (ImageView)
view.findViewById(R.id.image11);
        image11.setImageResource(R.drawable.image11);

        TextView textviewtitle12 = (TextView)
view.findViewById(R.id.textviewTitle12);
        textviewtitle12.setText("Шевченко Олеся
Александровна");
        TextView textviewt12 = (TextView)
view.findViewById(R.id.textviewT12);
        textviewt12.setText("Position");
        TextView textview12 = (TextView)
view.findViewById(R.id.textview12);
        textview12.setText("Образовательное
учреждение: \n" +
        "Белгородский государственный \n" +
        "университет, 2018\n" +
        "Направление подготовки: педагогическое
\n" +
        "образование\n" +
        "Профиль: информатика и иностранный
\n" +
        "язык (английский)\n" +
        "Должность: магистрант кафедры
математики \n" +
        "Опыт преподавания: 1 год" +
        "\n" +
        "\n" +

```

```

        "\n");

        ImageView image12 = (ImageView)
view.findViewById(R.id.image12);

        image12.setImageResource(R.drawable.image12);

        return view;
    }
}
package com.deatheek.school.fragment;

import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;

import com.deatheek.school.R;

public class FragmentFour extends Fragment {

    public FragmentFour() {
        // Required empty public constructor
    }

    public static FragmentOne newInstance(String
param1, String param2) {
        FragmentOne fragment = new FragmentOne();
        Bundle args = new Bundle();
        fragment.setArguments(args);
        return fragment;
    }

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

```

```

@Override
    public View onCreateView(LayoutInflater inflater,
ViewGroup container,
        Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        View view =
inflater.inflate(R.layout.fragment_four,
        container, false);

        TextView textviewtitle1 = (TextView)
view.findViewById(R.id.textviewTitle1);
        textviewtitle1.setText("Аверин Геннадий
Викторович");
        TextView textview1 = (TextView)
view.findViewById(R.id.textviewT1);
        textview1.setText("Преподаватель
математики");
        ImageView image1 = (ImageView)
view.findViewById(R.id.image1);
        image1.setImageResource(R.drawable.image1);
        TextView textview1 = (TextView)
view.findViewById(R.id.textview1);
        textview1.setText("Образовательное
учреждение: \n" +
        "Николаевский ордена Трудового
Красного \n" +
        "\n" +
        "Знамени кораблестроительный институт
        \n" +
        "им. адм. С.О. Макарова, 1980\n" +
        "Специальность: холодильные и \n" +
        "компрессорные машины и установки\n" +
        "Квалификация: инженер-механик\n" +
        "доктор технических наук, профессор\n" +
        "\n" +
        "Должность: заведующий кафедрой общей
        \n" +
        "математики, профессор\n" +
        "Опыт преподавания: 19 лет");

        TextView textviewtitle2 = (TextView)
view.findViewById(R.id.textviewTitle2);
        textviewtitle2.setText("Борисовский Иван
Петрович");
        TextView textview2 = (TextView)
view.findViewById(R.id.textviewT2);

```

```

        textView2.setText("Преподаватель
математики");

        ImageView image2 = (ImageView)
view.findViewById(R.id.image2);

        image2.setImageResource(R.drawable.image2);

        TextView textView2 = (TextView)
view.findViewById(R.id.textView2);

        textView2.setText("Образовательное
учреждение: \n" +

                "Белгородский государственный \n" +
                "педагогический институт \n" +
                "им. М.С. Ольминского, 1992\n" +
                "Специальность: математика\n" +
                "Квалификация: учитель математики и \n"
+
                "\n" +
                "информатики и вычислительной техники
\n" +
                "и звание учитель средней школы\n" +
                "\n" +
                "кандидат физико-математических наук,
\n" +
                "доцент\n" +
                "Должность: заведующий кафедрой\n" +
                "математики, доцент\n" +
                "Опыт преподавания: 23 года");

        TextView textViewtitle3 = (TextView)
view.findViewById(R.id.textViewTitle3);

        textViewtitle3.setText("Витохина Наталья
Николаевна");

        TextView textView3 = (TextView)
view.findViewById(R.id.textViewT3);

        textView3.setText("Преподаватель
математики");

        TextView textView3 = (TextView)
view.findViewById(R.id.textView3);

        textView3.setText("Образовательное
учреждение: \n" +

                "Белгородский государственный \n" +
                "университет, 2002\n" +
                "Специальность: математика\n" +
                "Квалификация: учитель математики \n" +
                "и информатики\n" +
                "кандидат физико-математических наук\n"
+
                "\n" +
                "Должность: доцент кафедры математики
\n" +

```

```

        "Опыт преподавания: 15 лет");

        ImageView image3 = (ImageView)
view.findViewById(R.id.image3);

        image3.setImageResource(R.drawable.image3);

        TextView textViewtitle4 = (TextView)
view.findViewById(R.id.textViewTitle4);

        textViewtitle4.setText("Красовская Людмила
Владимировна");

        TextView textView4 = (TextView)
view.findViewById(R.id.textViewT4);

        textView4.setText("Преподаватель
информатики");

        TextView textView4 = (TextView)
view.findViewById(R.id.textView4);

        textView4.setText("Образовательное
учреждение: \n" +

                "\n" +
                "Дагестанский государственный
\n" +
                "технический" +
                "университет, 2000\n" +
                "Специальность: вычислительные
машины, \n" +
                "комплексы, системы и сети\n" +
                "Квалификация: инженер\n" +
                "кандидат технических наук, доцент\n" +
                "Должность: заведующий кафедрой\n" +
                "информатики, естественнонаучных \n" +
                "дисциплин и методик \n" +
                "преподавания, доцент \n" +
                "Опыт преподавания: 18 лет");

        ImageView image4 = (ImageView)
view.findViewById(R.id.image4);

        image4.setImageResource(R.drawable.image4);

        TextView textViewtitle5 = (TextView)
view.findViewById(R.id.textViewTitle5);

        textViewtitle5.setText("Некрасова Ирина
Викторовна");

        TextView textView5 = (TextView)
view.findViewById(R.id.textViewT5);

        textView5.setText("Преподаватель
математики");

        TextView textView5 = (TextView)
view.findViewById(R.id.textView5);

        textView5.setText("Образовательное
учреждение: \n" +

                "Белгородский государственный \n" +

```

```

"университет, 2001\n" +
"Специальность: математика\n" +
"Квалификация: учитель математики \n" +
"и информатики\n" +
"кандидат физико-математических наук\n"
+
"Должность: доцент кафедры общей \n" +
"математики \n" +
"Опыт преподавания: 17 лет");

ImageView image5 = (ImageView)
view.findViewById(R.id.image5);

image5.setImageResource(R.drawable.image5);

TextView textviewtitle6 = (TextView)
view.findViewById(R.id.textviewTitle6);

textviewtitle6.setText("Остапенко Светлана
Ивановна");

TextView textviewt6 = (TextView)
view.findViewById(R.id.textviewT6);

textviewt6.setText("Преподаватель
математики");

TextView textview6 = (TextView)
view.findViewById(R.id.textview6);

textview6.setText("Образовательное
учреждение: \n" +
"Белгородский государственный \n" +
"университет, 2007\n" +
"Специальность: математика\n" +
"Квалификация: учитель математики и \n"
+
"информатики" +
"кандидат педагогических наук,\n" +
\n" +
"Должность: доцент кафедры математики
\n" +
"Опыт преподавания: 11 лет");

ImageView image6 = (ImageView)
view.findViewById(R.id.image6);

image6.setImageResource(R.drawable.image6);

TextView textviewtitle7 = (TextView)
view.findViewById(R.id.textviewTitle7);

textviewtitle7.setText("Полунин Виктор
Александрович");

TextView textview7 = (TextView)
view.findViewById(R.id.textviewT7);

textview7.setText("Преподаватель
математики");

```

```

TextView textview7 = (TextView)
view.findViewById(R.id.textview7);

textview7.setText("Образовательное
учреждение: \n" +
"Белгородский государственный \n" +
"университет, 2006\n" +
"Специальность: математика\n" +
"Квалификация: бакалавр\n" +
"кандидат физико-математических наук\n"
+
"Должность: доцент кафедры \n" +
"дифференциальных уравнений \n" +
"Опыт преподавания: 8 года");

ImageView image7 = (ImageView)
view.findViewById(R.id.image7);

image7.setImageResource(R.drawable.image7);

TextView textviewtitle8 = (TextView)
view.findViewById(R.id.textviewTitle8);

textviewtitle8.setText("Сатлер Ольга
Николаевна");

TextView textviewt8 = (TextView)
view.findViewById(R.id.textviewT8);

textviewt8.setText("Преподаватель
информатики");

TextView textview8 = (TextView)
view.findViewById(R.id.textview8);

textview8.setText("Образовательное
учреждение: \n" +
"Карагандинский государственный \n" +
"технический университет, 2002\n" +
"Специальность: системы
автоматизированного \n" +
"проектирования\n" +
"Квалификация: инженер-
системотехник\n" +
"кандидат технических наук\n" +
"Должность: доцент кафедры
информатики, \n" +
"естественнонаучных дисциплин и \n" +
"методик преподавания \n" +
"Опыт преподавания: 15 лет");

ImageView image8 = (ImageView)
view.findViewById(R.id.image8);

image8.setImageResource(R.drawable.image8);

```

```

        TextView textViewTitle9 = (TextView)
view.findViewById(R.id.textViewTitle9);
        textViewTitle9.setText("Старовойтов Александр
Степанович");
        TextView textView9 = (TextView)
view.findViewById(R.id.textView9);
        textView9.setText("Преподаватель
информатики");
        TextView textView9 = (TextView)
view.findViewById(R.id.textView9);
        textView9.setText("Образовательное
учреждение: \n" +
        "Белгородский Государственный \n" +
        "университет, 2000\n" +
        "Специальность: математика\n" +
        "Квалификация: учитель математики и \n"
+
        "информатики, кандидат физико-
математических \n" +
        "наук, доцент\n" +
        "Должность: доцент кафедры
информатики, \n" +
        "естественнонаучных дисциплин и \n" +
        "методик преподавания\n" +
        "Опыт преподавания: 17 лет");
        ImageView image9 = (ImageView)
view.findViewById(R.id.image9);
        image9.setImageResource(R.drawable.image9);
        TextView textViewTitle10 = (TextView)
view.findViewById(R.id.textViewTitle10);
        textViewTitle10.setText("Флоринский Владимир
Вячеславович");
        TextView textView10 = (TextView)
view.findViewById(R.id.textView10);
        textView10.setText("Преподаватель
математики");
        TextView textView10 = (TextView)
view.findViewById(R.id.textView10);
        textView10.setText("Образовательное
учреждение: \n" +
        "Харьковский государственный \n" +
        "университет им. А.М.Горького, 1977\n"
+
        "Специальность: математика\n" +
        "Квалификация: математик\n" +
        "кандидат физико-математических \n" +
        "наук, доцент\n" +
        "Должность: доцент кафедры общей \n" +
        "математики \n" +
        "Опыт преподавания: 41 год");
        ImageView image10 = (ImageView)
view.findViewById(R.id.image10);
        image10.setImageResource(R.drawable.image10);
        TextView textViewTitle11 = (TextView)
view.findViewById(R.id.textViewTitle11);
        textViewTitle11.setText("Чернявских Светлана
Дмитриевна");
        TextView textView11 = (TextView)
view.findViewById(R.id.textView11);
        textView11.setText("Преподаватель биологии");
        TextView textView11 = (TextView)
view.findViewById(R.id.textView11);
        textView11.setText("Образовательное
учреждение: \n" +
        "Белгородский государственный \n" +
        "педагогический институт \n" +
        "им. М.С. Ольминского, 1996\n" +
        "Специальность: биология и психология\n"
+
        "Квалификация: учитель биологии, \n" +
        "педагог-психолог и звание учитель \n" +
        "средней школы\n" +
        "кандидат биологических наук, доцент\n"
+
        "Должность: декан факультета математики
\n" +
        "и естественнонаучного образования,
доцент\n" +
        "Опыт преподавания: 20 лет");
        ImageView image11 = (ImageView)
view.findViewById(R.id.image11);
        image11.setImageResource(R.drawable.image11);
        TextView textViewTitle12 = (TextView)
view.findViewById(R.id.textViewTitle12);
        textViewTitle12.setText("Шевченко Олеся
Александровна");
        TextView textView12 = (TextView)
view.findViewById(R.id.textView12);
        textView12.setText("Position");
        TextView textView12 = (TextView)
view.findViewById(R.id.textView12);
        textView12.setText("Образовательное
учреждение: \n" +
        "Харьковский государственный \n" +
        "университет им. А.М.Горького, 1977\n"
+
        "Специальность: математика\n" +
        "Квалификация: математик\n" +
        "кандидат физико-математических \n" +
        "наук, доцент\n" +
        "Должность: доцент кафедры общей \n" +
        "математики \n" +
        "Опыт преподавания: 41 год");
        ImageView image10 = (ImageView)
view.findViewById(R.id.image10);
        image10.setImageResource(R.drawable.image10);
        TextView textViewTitle11 = (TextView)
view.findViewById(R.id.textViewTitle11);
        textViewTitle11.setText("Чернявских Светлана
Дмитриевна");
        TextView textView11 = (TextView)
view.findViewById(R.id.textView11);
        textView11.setText("Преподаватель биологии");
        TextView textView11 = (TextView)
view.findViewById(R.id.textView11);
        textView11.setText("Образовательное
учреждение: \n" +
        "Белгородский государственный \n" +
        "педагогический институт \n" +
        "им. М.С. Ольминского, 1996\n" +
        "Специальность: биология и психология\n"
+
        "Квалификация: учитель биологии, \n" +
        "педагог-психолог и звание учитель \n" +
        "средней школы\n" +
        "кандидат биологических наук, доцент\n"
+
        "Должность: декан факультета математики
\n" +
        "и естественнонаучного образования,
доцент\n" +
        "Опыт преподавания: 20 лет");
        ImageView image11 = (ImageView)
view.findViewById(R.id.image11);
        image11.setImageResource(R.drawable.image11);
        TextView textViewTitle12 = (TextView)
view.findViewById(R.id.textViewTitle12);
        textViewTitle12.setText("Шевченко Олеся
Александровна");
        TextView textView12 = (TextView)
view.findViewById(R.id.textView12);
        textView12.setText("Position");
        TextView textView12 = (TextView)
view.findViewById(R.id.textView12);
        textView12.setText("Образовательное
учреждение: \n" +
        "Харьковский государственный \n" +
        "университет им. А.М.Горького, 1977\n"
+
        "Специальность: математика\n" +
        "Квалификация: математик\n" +
        "кандидат физико-математических \n" +
        "наук, доцент\n" +
        "Должность: доцент кафедры общей \n" +
        "математики \n" +
        "Опыт преподавания: 41 год");

```

```

<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:style="http://schemas.android.com/tools"
    android:id="@+id/scroll"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">
<HorizontalScrollView
    android:id="@+id/hscroll"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
<Text View

    android:id="@+id/textviewTitle1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft="15sp"
    android:layout_marginTop="27sp"/>
<Text View
    android:id="@+id/textviewT1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>
<ImageView
    android:id="@+id/image1"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft="80sp"
    android:layout_marginTop="14sp"
    android:src="@drawable/image1" />

```

```

<Text View
    android:id="@+id/textview1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft="15sp"/>

```

```

<Text View
    android:id="@+id/textviewTitle2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft="15sp"
    android:layout_marginTop="27sp"/>

```

```

<Text View
    android:id="@+id/textviewT2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>

```

```

<ImageView
    android:id="@+id/image2"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft="80sp"
    android:layout_marginTop="14sp"
    android:src="@drawable/image1" />

```

```

<Text View
    android:id="@+id/textview2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"

```

```

        android:layout_marginLeft = "15sp"/>

<TextView
    android:id="@+id/textviewTitle3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft = "15sp"
    android:layout_marginTop = "27sp"/>

<TextView
    android:id="@+id/textviewT3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft = "15sp"/>

<ImageView
    android:id="@+id/image3"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft = "80sp"
    android:layout_marginTop = "14sp"
    android:src="@drawable/image1" />

<TextView
    android:id="@+id/textview3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft = "15sp"/>

<TextView
    android:id="@+id/textviewTitle4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft = "15sp"
    android:layout_marginTop = "27sp"/>

        android:layout_height="wrap_content"
        android:textSize="26sp"
        android:isScrollContainer="true"
        android:textStyle="bold"
        android:layout_marginLeft = "15sp"
        android:layout_marginTop = "27sp"/>

<TextView
    android:id="@+id/textviewT4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft = "15sp"/>

<ImageView
    android:id="@+id/image4"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft = "80sp"
    android:layout_marginTop = "14sp"
    android:src="@drawable/image1" />

<TextView
    android:id="@+id/textview4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft = "15sp"/>

<TextView
    android:id="@+id/textviewTitle5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft = "15sp"
    android:layout_marginTop = "27sp"/>

```

```

<TextView
    android:id="@+id/textviewT5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>
<ImageView
    android:id="@+id/image5"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft="80sp"
    android:layout_marginTop="14sp"
    android:src="@drawable/image1" />
<TextView
    android:id="@+id/textview5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft="15sp"/>
<TextView
    android:id="@+id/textviewTitle6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft="15sp"
    android:layout_marginTop="27sp"/>
<TextView
    android:id="@+id/textviewT6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>
<TextView
    android:id="@+id/textviewT5"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>
<ImageView
    android:id="@+id/image6"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft="80sp"
    android:layout_marginTop="14sp"
    android:src="@drawable/image1" />
<TextView
    android:id="@+id/textview6"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft="15sp"/>
<TextView
    android:id="@+id/textviewTitle7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft="15sp"
    android:layout_marginTop="27sp"/>
<TextView
    android:id="@+id/textviewT7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>
<ImageView
    android:id="@+id/image7"

```



```

        android:layout_width="250sp"
        android:layout_height="360sp"
        android:scaleType="fitStart"
        android:layout_marginLeft = "80sp"
        android:layout_marginTop = "14sp"
        android:src="@drawable/image1" />
<TextView
    android:id="@+id/textview7"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft = "15sp"/>

<TextView
    android:id="@+id/textviewTitle8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft = "15sp"
    android:layout_marginTop = "27sp"/>

<TextView
    android:id="@+id/textviewT8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft = "15sp"/>

<ImageView
    android:id="@+id/image8"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft = "80sp"
    android:layout_marginTop = "14sp"
    android:src="@drawable/image1" />

        android:src="@drawable/image1" />
<TextView
    android:id="@+id/textview8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft = "15sp"/>

<TextView
    android:id="@+id/textviewTitle9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft = "15sp"
    android:layout_marginTop = "27sp"/>

<TextView
    android:id="@+id/textviewT9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft = "15sp"/>

<ImageView
    android:id="@+id/image9"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft = "80sp"
    android:layout_marginTop = "14sp"
    android:src="@drawable/image1" />

<TextView
    android:id="@+id/textview9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

        android:textSize="16sp"
        android:isScrollContainer="true"
        android:typeface="serif"
        android:layout_marginLeft="15sp"/>

<TextView
    android:id="@+id/textviewTitle10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft="15sp"
    android:layout_marginTop="27sp"/>

<TextView
    android:id="@+id/textviewT10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>

<ImageView
    android:id="@+id/image10"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft="80sp"
    android:layout_marginTop="14sp"
    android:src="@drawable/image1" />

<TextView
    android:id="@+id/textview10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft="15sp"/>

        android:textSize="16sp"
        android:isScrollContainer="true"
        android:typeface="serif"
        android:layout_marginLeft="15sp"/>

<TextView
    android:id="@+id/textviewTitle11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="26sp"
    android:isScrollContainer="true"
    android:textStyle="bold"
    android:layout_marginLeft="15sp"
    android:layout_marginTop="27sp"/>

<TextView
    android:id="@+id/textviewT11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft="15sp"/>

<ImageView
    android:id="@+id/image11"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft="80sp"
    android:layout_marginTop="14sp"
    android:src="@drawable/image1" />

<TextView
    android:id="@+id/textview11"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft="15sp"/>

<TextView
    android:id="@+id/textviewTitle12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

        android:textSize="26sp"
        android:isScrollContainer="true"
        android:textStyle="bold"
        android:layout_marginLeft = "15sp"
        android:layout_marginTop = "27sp"/>
<TextView
    android:id="@+id/textviewT12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:isScrollContainer="true"
    android:textStyle="italic"
    android:layout_marginLeft = "15sp"/>
<ImageView
    android:id="@+id/image12"
    android:layout_width="250sp"
    android:layout_height="360sp"
    android:scaleType="fitStart"
    android:layout_marginLeft = "80sp"

        android:layout_marginTop = "14sp"
        android:src="@drawable/image1" />
<TextView
    android:id="@+id/textview12"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="16sp"
    android:isScrollContainer="true"
    android:typeface="serif"
    android:layout_marginLeft = "15sp"/>

</LinearLayout>
</HorizontalScrollView>
</ScrollView>

```