

Ив –Вычислительная техника и программное обеспечение Конференция – 3G

**Российская научно-социальная программа для молодежи и школьников «Шаг в
будущее»**

**ЮЖНО-УРАЛЬСКАЯ ИНТЕЛЛЕКТУАЛЬНО-СОЦИАЛЬНАЯ ПРОГРАММА
ДЛЯ МОЛОДЕЖИ И ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ - СОЗВЕЗДИЕ – НТТМ»**

**КООРДИНАЦИОННЫЙ ЦЕНТР Г. ВЕРХНИЙ УФАЛЕЙ
«ИНТЕЛЛЕКТУАЛЫ XXI ВЕКА» НТТМ**

Разработка мобильного приложения – гида по историческим достопримечательностям «Малахитовая линия»

**Исследовательская (творческая) работа на Московский
молодежный интеллектуальный форум «Шаг в будущее-Созвездие-
НТТМ»**

(3G) Информационные технологии, автоматизация, энергосбережение

Автор:

Альжанов Максим Булатович,

г. Челябинск, «Лицей № 97 г.

Челябинска», класс 9

Научные руководители:

Деникина Наталья Владимировна,

**учитель информатики МАОУ «Лицей №
97 г. Челябинска»;**

Деникин Антон Витальевич,

**учитель информатики МАОУ «Лицей №
97 г. Челябинска»;**

Москва - 2018

Оглавление

Что такое зеленая линия	2
Проблема	2
Цели и задачи:	2
Анализ рынка	3
Зарождение приложения	4
Как написать приложение на Android	5
Что делать дальше?	5
Создание проекта	5
Содержание проекта	7
Manifest	7
Java	8
Res	8
drawable	8
layout	8
mipmap	8
values	8
Работа с проектом	8
Layout-ы	9
Добавление картинки	9
Архитектура решения	10
Алгоритм:	10
Android приложение	11
Server	12
Admin приложение	12
Заключение	12
Список литературы:	13

Что такое зеленая линия

Я много путешествую. В этом году мне довелось посетить Пермь. У нас было время побродить по городу, и я узнал, что в Перми для туристов проложена Зеленая линия – пешеходная тропа по городу, следуя которой, ты можешь осмотреть основные достопримечательности города. Линия нанесена на асфальт. На достопримечательностях стоят стенды с подробной информацией. Таким образом, любой заинтересовавшийся историей города может получить информацию о его исторических объектах. Правда, мне **показалось**, что в современном мире можно иметь более удобный способ передачи информации об истории города, ведь существуют же мобильные приложения и электронные гиды. В Перми же такого Мобильного **приложения нет**.

Проблема

Приехав домой, я решил разузнать есть ли такая услуга у нас в городе. Поисковики выдают ссылку на Красную линию в Челябинске. Но сервис не работает. <http://www.red74.ru> Проект создавался к юбилею города, но так и не был завершен. Организаторы этого ресурса пошли по пути коммерциализации своего продукта. Сейчас они продают «Квест для детей по парку» за 349 рублей.

Есть также сайт <http://nashural.ru/progulki-po-stolicam/chelyabinsk/redline.htm> «Наш Урал», на нем подробно расписаны все исторические объекты в городе. Но нет не проложенного маршрута, не мобильной версии для телефона.

Таким образом у туриста или жителя моего родного города нет удобного способа посетить все достопримечательности, получить о них информацию, тем более нет возможности информирования на иностранном языке.

Цели и задачи:

Цель работы: Написать приложение-путеводитель для Android

Задачи:

- Провести анализ рынка
- Посмотреть на готовые решения
- Провести опрос
- Написать приветствующий экран
- Написать главную активность

- Написать класс для работы с API
- Написать класс для асинхронной загрузки данных
- Составить зеленую линию и наполнить приложение
- Выложить приложение в Play Market

Анализ рынка

Анализ рынка был проведен через поисковые системы и магазины мобильных приложений. Доступны из работающих три мобильных приложения экскурсовода:

1. **izi.TRAVEL** гид-путеводитель. Разработчик - Informap Technology Center, приложение для Android с автоматическим проигрыванием историй тура. Реализованы возможности служб определения местоположения (GPS), возможность поделиться уникальными фактами или понравившимися историями с друзьями, проверки знаний с помощью развлекательных викторин, возможность скачать аудиогид на телефон до поездки, **но:**
 - a. это платный контент: Стоимось туров от 55рублей.
 - b. Ограниченное количество туров по России (Москва, Санкт-Петербург и Стрельна)
 - c. В отзывах офлайн карты не работают
 - d. Не все экскурсии нравятся.
2. Путеводитель по городам офлайн **iniCall.com** – мобильное приложение в котором - путеводитель работает без интернета, много статей о достопримечательностях с фото, офлайн прокладка маршрута позволит Вам добраться до нужной точки, маршруты на машине или пешком., достопримечательности, отели, рестораны, ночные клубы на карте города, встроенный планировщик маршрута позволит составить план обхода достопримечательностей, офлайн поиск адресов. **Но:**
 - a. Отзывы плохие, что карты не работают,
 - b. Из наших городов только Москва и Санкт Петербург,
 - c. Весь контент платный стоимость от 99,00Р до 199,00Р за единицу
3. **Developer-blog.ru** путеводитель по отдельным городам Москва, Санкт Петербург, Сочи - Путеводитель в который заложены многие возможности, поддержка русского языка; справочная информация о множестве достопримечательностей: адреса, стоимость, график работы, номер телефона и др.; сохранение самых интересных объектов в списке избранных; отслеживание текущего местоположения по координатам GPS или сотовой сети; вычисление ближайших

достопримечательностей, расстояния и направления до них; актуальная схема метро; расчёт количества станций и времени движения между остановками; поиск ближайшей остановки и расстояния до нее. После покупки карты Вы сможете: использовать подробную оффлайн карту Москвы с более чем 20000 объектов на ней; прокладывать маршрут между любыми 2 точками на карте; сохранять выбранные места в списке Избранного. **Но:**

- a. Отзывы, что платные услуги не работают, офлайн карта не определяет адрес,
- b. Информация по объектам не актуальная (стоимость и часы работы)
- c. Аудио сопровождения нет и перевода тоже.

Таким образом бесплатных мобильных приложений путеводителей не хватает, из отзывов понятно, что туристам нужны интересные исторические путеводители, а всю остальную информацию можно подчерпнуть в уже существующих GOOGLE maps, 2Gis и Yandex.maps

Зарождение приложения

Я решил создать свое приложение-путеводитель для телефонов. Я задумался, а как было бы удобно? Какая мобильная версия экскурсии по Челябинску устроила бы и детей, и взрослых и гостей и Челябинцев. Мы провели опрос 44 человек на улице Кирова. Оказалось, что большинству людей понравились идеи снабдить приложение навигатором и поддержкой несколькими языками. Большинство также проголосовало за отсутствие платы за мобильное приложение.

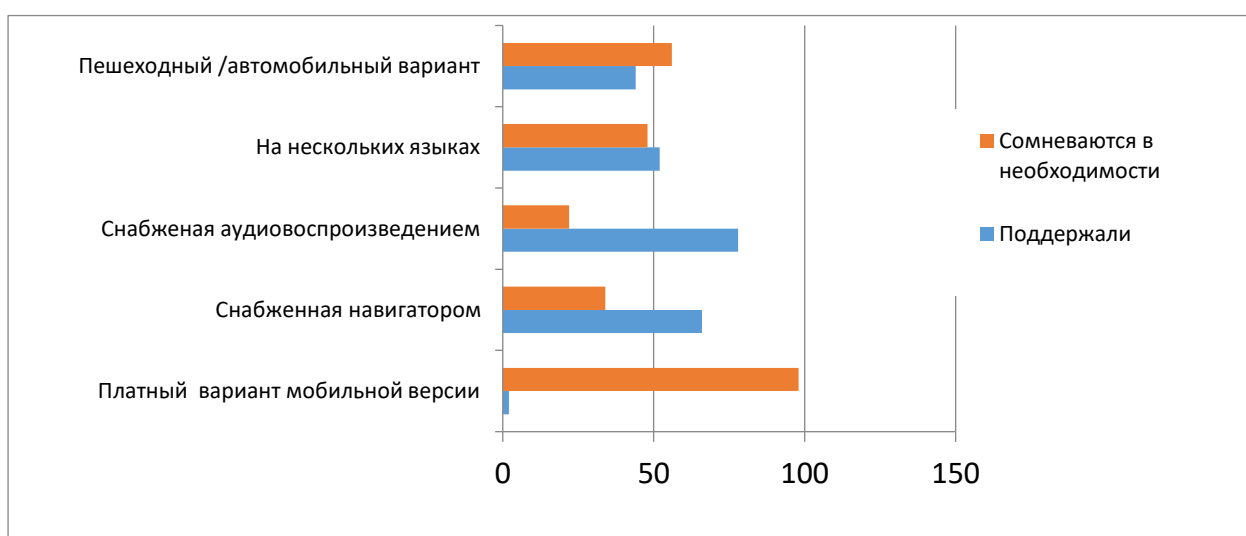


Таблица 1: результаты опроса

В результате родился алгоритм будущей программы для мобильной версии. Мы ее назвали Малахитовой линией (с целью просто выделиться из всех остальных линий уральского региона) кстати не в одном городе нет полноценной мобильной версии.

Итак, она должна быть простой, удобной, бесплатной. Вот такой прототип для Android я и создавал.

Как написать приложение на Android

Первое что нужно сделать для начала разработки приложение – скачать IDE¹ и SDK². Ссылки на скачивание легко найти в интернете. Достаточно просто написать “Android Studio” в поисковом запросе любого поискового движка. Android Studio — это интегрированная среда разработки для работы с платформой Android официально поддерживаемая создателями Android, Google.

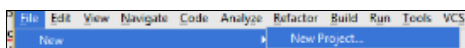
Что делать дальше?

После установки Android Studio можно приступить к созданию своей первой программы.

В качестве языка программирования для Android используется Java. Для создания пользовательского интерфейса используется XML.

Создание проекта

Запускаем Студию и выбираем File -> New -> New Project....

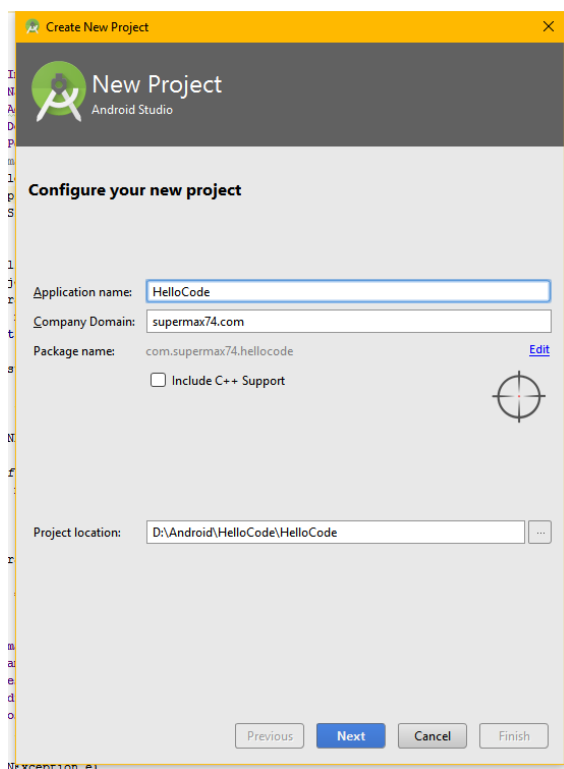


Появится диалоговое окно.

Поле Application name: - понятное имя для приложения, которое будет отображаться в заголовке приложения. Заменим на FlappyBird. В принципе здесь можно написать на русском языке, но у Android есть замечательная возможность выводить нужные строки на телефонах с разными языками. Скажем, у американца на телефоне появится надпись на английском, а у русского - на русском. Поэтому в первоначальных настройках всегда используются английские варианты, а локализованные строки можно подготовить позже.

¹ IDE - среда разработки

² SDK – англ. Software Development Kit (Пакет инструментов разработчика). Необходим для работы с Android



Поле **Company Domain**: служит для указания вашего сайта. По умолчанию там может появиться ваше имя как пользователя компьютера. Если сайт у вас есть, то можете ввести его адрес, либо придумайте какое-нибудь название. Введённое имя запоминается и будет автоматически подставляться в следующих новых проектах. Экономия, однако.

Поле **Package name**: формирует специальный Java-пакет на основе вашего имени из предыдущего поля. В Java используется перевёрнутый вариант для наименования пакетов, поэтому сначала идёт `com`, а потом уже название сайта. Пакет служит для

уникальной идентификации вашего приложения, когда вы будете его распространять. Если сто человек напишет сто приложений с названием "Flappy Bird", то будет непонятно, где приложение, написанное официальным разработчиком. А приложение с именем пакета `com.official.FlappyBirdGame` проще найти. Кнопка **Edit** позволяет отредактировать подготовленный вариант. Например, вы пишете приложение на заказ и вам нужно использовать имя пакета, утверждённое заказчиком, а не ваш вариант по умолчанию.

Третье поле **Project location**: позволяет выбрать место на диске для создаваемого проекта.

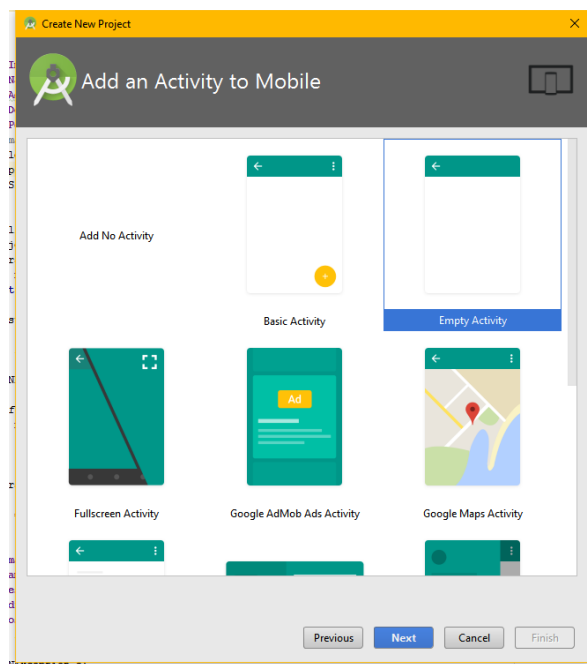
В последней версии Android Studio появилась возможность добавлять коды на языке C++. Я не знаю, как это работает и не рекомендую нажимать на галочку без особых знаний и понимания что вы делаете.

Нажимаем на кнопку **Next** для перехода к выбору целевых устройств.

Кроме выбора устройств нужно указать целевую версию Android под которую будет разрабатываться приложение.

Если щёлкнуть по ссылке **Help me choose**, то откроется окно с графиком. Если вам интересно, можете посмотреть.

Идём дальше и снова нажимаем кнопку **Next**.



Здесь следует выбрать внешний вид экрана приложения.

Предложенные шаблоны позволяют сэкономить время на написание стандартного кода для типичных ситуаций. Опытный разработчик может вручную написать любой из предложенных вариантов, используя вариант Add No Activity, где никаких заготовок не будет.

Пару лет назад был только один шаблон. Список шаблонов постоянно пополняется и

теперь их уже 12. Перечислю часть из них.

- Basic Activity
- Empty Activity
- Fullscreen Activity
- Google Maps Activity
- Google AdMod Ads Activity

Кстати в написании своей работы я использовал Google Maps Activity.

Итак, мы выбрали вариант Empty Activity и переходим к следующему окну.

Ну вот, мы закончили с первоначальной настройкой. Нажимаем кнопку Finish.

Содержание проекта

Вкладка Android содержит две основные папки: app и Gradle Scripts. Первая папка app является отдельным модулем для приложения и содержит все необходимые файлы приложения - код, ресурсы картинок и т.п. Вторая папка служит для различных настроек, управления проектом и многих других вещей.

Manifest

Папка manifest содержит единственный файл манифеста AndroidManifest.xml. В этом файле должны быть объявлены все активности, службы, приёмники и контент-провайдеры приложения. Также он должен содержать требуемые приложению разрешения. Например, если приложению требуется доступ к сети, это должно быть

определено здесь. «AndroidManifest.xml» можно рассматривать, как описание для развертывания Android-приложения.

Java

Папка java содержит три подпапки - рабочую и для тестов. Рабочая папка имеет название вашего пакета и содержит файлы классов. Сейчас там один класс MainActivity.

Res

Папка res содержит файлы ресурсов, разбитых на отдельные подпапки.


drawable — в этих папках хранят графические ресурсы - картинки и xml-файлы, описывающие цвет и фигуры.

layout — в данной папке содержатся xml-файлы, описывающие внешний вид форм и различных элементов форм. После создания проекта там уже имеется файл activity_main.xml, который отвечает за внешний вид главного окна приложения.

miptar - здесь хранят значки приложения под разные разрешения экрана

values — тут размещаются строковые ресурсы, ресурсы цветов, тем, стилей и измерений, которые мы можем использовать в нашем проекте. Здесь находятся файлы colors.xml, dimens.xml, strings.xml, styles.xml

Работа с проектом

Нажмём на зелёный треугольник  Run (Shift+F10) на панели инструментов в верхней части студии для запуска приложения.

ADB попросит нас выбрать устройство. Дабы не тратить времени, подключим физическое устройство и выберем его.

После того как Gradle соберет проект он откроется на телефоне.

Если все сделано правильно, то на экране откроется программа с заголовком “Hello World” и таким же текстом сверху слева.

Откроем activity_main.xml и перейдем в режим правки текста снизу.

Найдем строки описания Text View.

В этих строках поменяем значение `android:text` на “Hello Code”.

Студия наругается на нас т.к. мы использовали `HardCode`³.

Исправим это нажав `Alt+Enter`. Большинство ошибок в студии можно исправить этим сочетанием клавиш. В появившемся меню выберем `Extract string resource`.

Зададим название ресурса “HelloCode” и нажмем `Enter`.

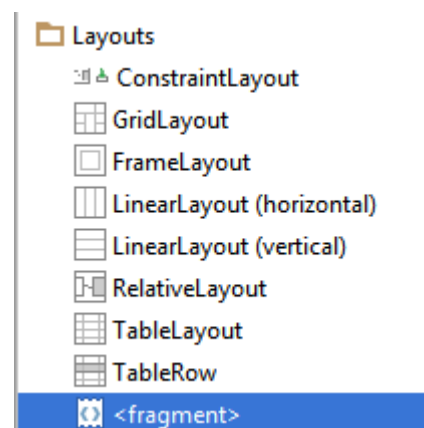
Если запустить приложение еще раз, увидим, что вместо `Hello World`, теперь стоит `Hello Code`.

Layout-ы

Layout-ы нужны для размещения элементов в `Activity`

На данный момент для `Android 7.0 Nougat` существует 8 типов layout-ов. Основные из них:

- `GridLayout` размещение по таблице с возможностью объединять ячейки.
- `FrameLayout`
- `LinearLayout`
- `RelativeLayout`
- `TableLayout`



Добавление картинки



Допустим нам в приложении потребовалась картинка.

Вот как ее добавить:

Первое что следует сделать: скопировать картинку и вставить ее в папку `drawable`. Студия сразу предложит задать название файла. Оставим как есть.

³ Хардкод (`hardcode`) это распространенная ошибка программистов, которая заключается в «принудительном» присвоении переменной какого либо значения, вместо того чтобы присваивать его динамически, в зависимости от ситуации. Слово `hard` в переводе с английского означает «твёрдый», а `code` — «программа, программный код». Хардкод — одна из главных причин появления в программах разного рода глюков и багов.

Второе, добавить элемент `ImageView` в `layout` перетаскив его из списка элементов на `layout`.

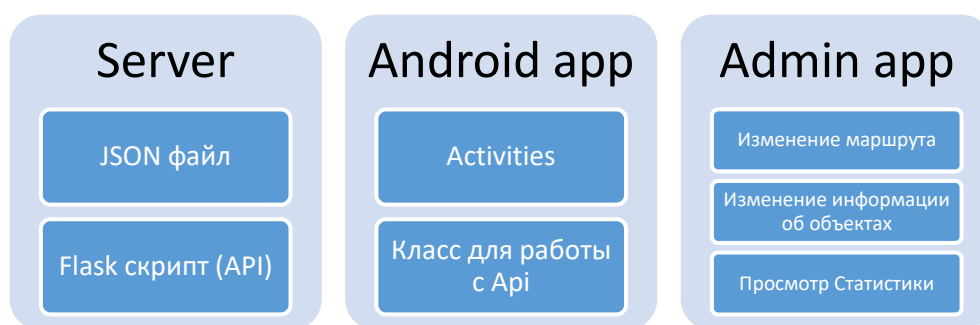
И третье – задать ресурс файла в свойствах `ImageView`. Нажав на элемент, в окне `Properties` следует найти пункт `src`⁴, выбрать его и нажать троеточие.

В появившемся окне выбрать нашу фотографию.

Готово.

Архитектура решения

Моё решение состоит из 3-х частей: серверная часть, Андроид приложение и программа администратии.



Общение между сервером происходит с помощью открытого API по протоколу `http` в формате `JSON` файла.

Алгоритм:

Исходя из моего многолетнего (с 8 лет) опыта пользователя `Android` я с уверенностью могу сказать, как будет удобно и практично пользователю моего приложения.

Приложение должно занимать мало места на устройстве. Привлекать пользователя интуитивно-понятным интерфейсом и количеством показываемой информации на экране.

Для того чтобы оптимизировать количество расходуемого трафика я решил кешировать актуальный маршрут на устройство пользователя. Это, во-первых, позволит пользоваться приложением без интернета, а во-вторых ускорит запуск приложения. Для того чтобы у программы всегда была актуальный вариант маршрута, при каждой необходимости приложение будет сверять текущий номер версии с актуальной на сервере и при различии приложение скачает новый маршрут на устройство.

⁴ Аббревиатура `source` широко применяемая разработчиками всей планеты.

Так как контент полностью зависит от того что вернет сервер приложение можно легко переделать под другой город всего лишь запустив ещё один сервер, изменив название приложение, иконку и адрес сервера. Остается только составить маршруты.

Android приложение

```
{
  "l":0,
  "rn":0
  "v":"3",
  "r":[
    {
      "imgurl":"...",
      "name":"Точка1",
      "desc":"Описание точки №1",
      "address":"Адрес точки 1",
      "posx":0.0,
      "posy":0.0
    },
    {
      "imgurl":"...",
      "name":"Точка2",
      "desc":"Описание точки №2",
      "address":"Адрес точки 2",
      "posx":0.1,
      "posy":0.1
    },
  ],
}
```

Приложение на андроид состоит из пяти активностей:

- welcomeActivity – активность приветствия выбора языка и маршрута
- mainActivity – активность с картой и визуальным маршрутом
- listActivity – активность со списком достопримечательностей и пошаговым маршрутом
- infoActivity – активность с подробной информацией об объекте
- settingsActivity – активность с настройками в которой пользователь при ошибке мог бы поменять язык.

В своем приложении я написал класс Sight для хранения информации об одиночном объекте.

Объект Sight содержит такие поля:

- imgurl (String) – ссылка на thumbnail объекта
- name (String) – название достопримечательности
- desc (String) – описание достопримечательности
- address (String) – физический адрес достопримечательности
- pos (LatLng) – Относительные координаты
- pic (Bitmap) – Скачанный thumbnail

Для работы с api при составлении http запроса к северу используются аргументы l, r и v

Аргумент l (Integer) задает запрашиваемый язык. Обязательный аргумент.

Аргумент *г* (Integer) задает запрашиваемый маршрут. При отсутствии данного аргумента сервер вернет массив названий маршрутов для языка *l*.

Аргумент *v* задает желание получить номер версии маршрута. При наличии данного аргумента и аргумента *г* сервер вернет номер актуальной версии маршрута

Когда пользователь заходит в приложение: *mainActivity* проверяет, есть ли обновления маршрута с помощью аргумента *v*. Если обновления есть, *mainActivity* загружает его и сохраняет на устройство. Далее программа преобразовывает JSON файлы в массив класса *Sight*.

Server

Сервер состоит из главного JSON файла и питон-скрипта обрабатывающий http запросы и возвращает JSON строку содержащую информацию об объектах с информацией из главного JSON файла.

Admin приложение

Когда нам нужно обновить маршрут администратор открывает приложение, вводит пароль и изменяет маршрут. В приложении так же можно изменить информацию об объектах или изменить картинку. Но прежде всего в приложении можно изменить как выглядит страница с подробной информацией об объекте. Общение с сервером так же происходит по http протоколу, но с передачей ключа OAuth2

Заключение

Данное решение поможет гостям и жителям Челябинска лучше узнать историю города. При написании приложения были получены навыки написания приложений под ОС Android. Был проведен анализ рынка: подобных решений/альтернатив приложению нет. Был проведен опрос с целью охватить большую часть рынка. Были написаны ключевые активности. Был написан класс для работы с API который так же являлся классом для асинхронной загрузки данных. В будущем планируется дописать периферийные активности, а также выложить приложение в Play Market

Список литературы:

<https://www.youtube.com/watch?v=ac8y518CIng&index=0&list=PLyfVjOYzujugap6Rf3ETNKkx4v9ePIlNK> (Плейлист YouTube о разработке Android приложений. Написал своё первое приложение с помощью него.)

<http://developer.alexanderklimov.ru/android/> (Сайт очень хорошего человека который рассказывает об android разработке с энтузиастом, удовольствием и котиками)

<http://startandroid.ru/ru/uroki/vse-uroki-spiskom.html> (Сайт с уроками и примерами Android приложений)

<https://developer.android.com/reference/packages.html> (Официальный справочник Android от Google. Большинство кода просмотрено оттуда)

<http://myitschool.ru/book/> (Только для учеников IT школы Samsung которым я являюсь)