

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
_____ О.В.
Непомнящий
подпись инициалы, фамилия
« _____ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Сканер предложений для торговых площадок в сети Интернет
тема

Руководитель	_____	<u>ст. преподаватель</u>	<u>И. В. Матковский</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		<u>Ю. В. Аксенов</u>
	подпись, дата		инициалы, фамилия
Консультант	_____	<u>доцент, канд. техн. наук</u>	<u>Л. И. Покидышева</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия
Нормоконтролер	_____	<u>доцент, канд. техн. наук</u>	<u>В. И. Иванов</u>
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2019

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ задания на ВКР.....	5
1.1 Анализ поставленной задачи	5
1.2 Аналоги разрабатываемой системы	5
1.2.1 Autodrom Engine (edrom).....	6
1.2.2 CarsHotSurf.....	6
1.2.3 «Автосканер».....	7
1.2.4 Ads-Tracker.....	7
1.2.5 Глиса.....	8
1.3 Вывод по разделу	9
2 Проектирование программы	11
2.1 Общая архитектура программы.....	11
2.2 Общий алгоритм работы программы	13
2.3 Выбор инструментов разработки	15
2.4 Организация хранения данных	15
2.5 Вывод по разделу	15
3 Разработка программы.....	16
3.1 Модули архитектуры	16
3.1.1 Model	16
3.1.2 Presenter.....	18
3.1.3 View	19
3.1.4 Service.....	21
3.2 Диаграммы последовательностей	22
3.3 Хранение данных	25
3.4 Вывод по разделу	26
4 Руководство пользователя.....	27
4.1 Установка и запуск.....	27

4.2 Демонстрация возможностей программы	27
4.3 Тестирование приложения между разными версиями Android	31
Заключение	32
Список использованных источников	33

ВВЕДЕНИЕ

В последнее время технологии развиваются очень быстро. Это приводит к тому, что предметы, которые были полезными и удовлетворяли требованиям еще вчера, сегодня уже приходят в негодность. Но в том, в чем одни теряют интерес, другие видят пользу для себя. Поэтому формируется вторичный рынок товаров. Приобретение товаров на вторичном рынке имеет целый ряд преимуществ, самым важным из которых является цена. Она ниже, чем цена за тот же или схожий товар на первичном рынке. Однако, если нет возможности или желания проверять доступные предложения слишком часто, то найти выгодное предложение будет очень сложно. Предложение попросту будет выкуплено очень быстро. Таким образом, актуальной представляется задача автоматизации наблюдения за потенциально интересными товарами.

Так как большая часть онлайн сервисов предоставляет Application Programming Interface (API) для работы с их ресурсами, было принято решение использовать эту возможность. Было написано письмо в службу поддержки, с вопросом, возможно ли использовать API бесплатно для дипломного проекта. К сожалению, в ответе пришел отказ, и необходимо было найти другие пути решения задачи.

Альтернативой API является парсинг HTML страниц. Так как программа будет делать точно такие же запросы на сервер, что и браузер, она не будет ничем отличаться от человека, заходящего на ресурс, и просматривающего предложения.

1 Анализ задания на ВКР

1.1 Сравнение и анализ аналогов

Так как в поставленной задаче требуется постоянный мониторинг ресурсов, логичнее будет разрабатывать приложение под мобильные операционные системы. Все же требуется не только постоянно делать запросы на сервер, но и информировать пользователя, в случае появления возможности.

1.2 Аналоги разрабатываемой системы

В Интернете было найдено существенное множество аналогов, среди которых были отобраны несколько вариантов:

- Autodrom Engine (edrom) [1];
- CarsHotSurf [2];
- Автосканер [3];
- Ads-Tracker [4];
- Глиса [5].

Приложения-аналоги будут сравниваться по следующим признакам:

- цена;
- направленность приложения – по каким типам товаров работает приложение;
- поддержка нескольких ресурсов – возможность работать не только с 1 конкретным ресурсом, а так же и с другими;
- поддержка иностранных ресурсов – возможность работать с зарубежными ресурсами.

1.2.1 Autodrom Engine (edrom)

Разработчиками данное приложение описывается как специальная программа для поиска объявлений, размещаемых в сети Интернет разработанное для пользователей, занимающихся куплей-продажей автомобилей [1].

Первый минус приложения состоит в том, что оно узко специализируется только на автомобилях, что не совсем удобно, так как существует множество других товаров, предложения по которым хотелось бы отслеживать.

Вторым минусом является то, что приложение платное. Не исключено, что оно использует API, предоставляемое владельцами площадок, это создает накладные расходы для владельцев приложения.

Плюсом этого приложения является его проработанность и гибкость в конкретной сфере. В него встроены функции фильтрации по конкретным параметрам, которые могут использоваться только в данной сфере.

1.2.2 CarsHotSurf

Разработчики описывают данное приложение как специализированную программу, предназначенную для автоматического поиска авто объявлений, размещаемых на популярных авто сайтах: России, Европы и США [2].

Минусы:

- приложение до сих пор находится в стадии бета-версии, несмотря на то, что существует с 2013 года. Это может говорить о том, что разработка была либо приостановлена, либо отменена;
- приложение специализируется только на поиске автомобилей.

Плюсы:

- приложение является бесплатным;
- поиск объявлений на иностранных ресурсах.

1.2.3 Автосканер

Разработчики описывают данное приложение, как помощник для оперативного реагирования на появляющиеся предложения по продаже автомобилей [3].

Минусы:

- приложение является платным, хотя есть возможность бесплатно использовать его в течение 1 дня;
- приложение специализируется только на поиске автомобилей, что делает его менее гибким.

Ярко выделяющихся плюсов, среди аналогов, найдено не было.

1.2.4 Ads-Tracker

Разработчики описывают данное приложение как сервис, который отслеживает появление свежих объявлений по заданным фильтрам и информирует об этом пользователя. Так же заявлена возможность интеграции с некоторыми мессенджерами и электронной почтой [4].

Минусом является то, что приложение имеет бессрочный бесплатный режим. Но на него накладываются строгие ограничения, которые сильно уменьшают возможности пользователя. Самое важное ограничение – возможность сканировать рынки на предмет новых предложений только 1 раз в час.

Плюсы:

- бесплатный режим;
- работа с большим количеством Интернет-ресурсов, представленных далее:
 - Авито (avito.ru);
 - Авто.ру (auto.ru);
 - Юла (youla.io);

- Юла Авто;
- Циан (cian.ru);
- Локалс (locals.ru);
- Рухі.pro;
- Домофонд (domofond.ru);

- интеграция с популярными мессенджерами и электронной почтой, в описании указано, что приложение может отправлять результаты своих поисков пользователю напрямую, без необходимости проверять приложение Ads-Tracker.

1.2.5 Глиса

Разработчики описывают данное приложение как расширитель возможностей для сайтов с предложениями по продаже товаров на вторичном рынке. Так же заявлены возможности сбора информации для баз данных и борьбы с конкурентами [5].

Минус – бесплатный режим с ограничением до 1 проверки в час, а так же ограничение на количество проверенных предложений, до 3 за раз.

Плюсы:

- возможность работать с большим количеством ресурсов, среди которых:

- Авито (avito.ru);
- ЮЛА (youla.io);
- ebay.com;
- AliExpress (aliexpress.com);
- Дром(drom.ru);
- Циан (cian.ru);
- Домофонд (domofond.ru);
- Госзакупки (zakupki.gov.ru);
- Из рук в руки (irr.ru);

- ТИУ(tiu.ru);
- ОЛХ(olx.ua,olx.kz,olx.by);
- функции для борьбы с конкурентами, такие как мониторинг конкурирующих предложений на предмет поднятия и понижения цены;
- функции для рекламы своих предложений и для сбора информации о пользователях.

1.3 Вывод по разделу

Итоги сравнения будут представлены в таблице 1, для удобства сравнения.

Таблица 1 – сравнение аналогов

№	Название	Цена	Направленность	Поддержка нескольких ресурсов	Поддержка иностранных ресурсов
1	Autodrom Engine	2500р/мес	Только автомобили	-	-
2	CarsHot Surf	Бесплатно	Только автомобили	+	+
3	Автосканер	Пробный период, 400р/нед	Только автомобили	-	-
4	Ads-Tracker	Бесплатный режим, 2р/день	Любые товары	+	-
5	Глиса	Бесплатный режим, 5р/день	Любые товары	+	-

Как можно увидеть в таблице, большинство приложений предоставляют либо пробный период, либо бесплатный режим. Это сильно ограничивает возможность использования приложения. Так же большинство приложений направлены на определенную сферу товаров.

Основными плюсами разрабатываемого приложения будут:

- возможность сделать приложение полностью бесплатным, так как нет необходимости в покупке API;
- возможность поддержки большинства Интернет-ресурсов;
- возможность отслеживать предложения в любой отрасли.

2 Проектирование программы

2.1 Общая архитектура программы

Паттерном разработки был выбран Model-View-Presenter (MVP). Основная идея его состоит в том, чтобы отделить реализацию интерфейса от бизнес-логики. Это позволит сделать приложение модульным и простым для понимания [6].

Общая схема взаимодействия модулей изображена на рисунке 1.

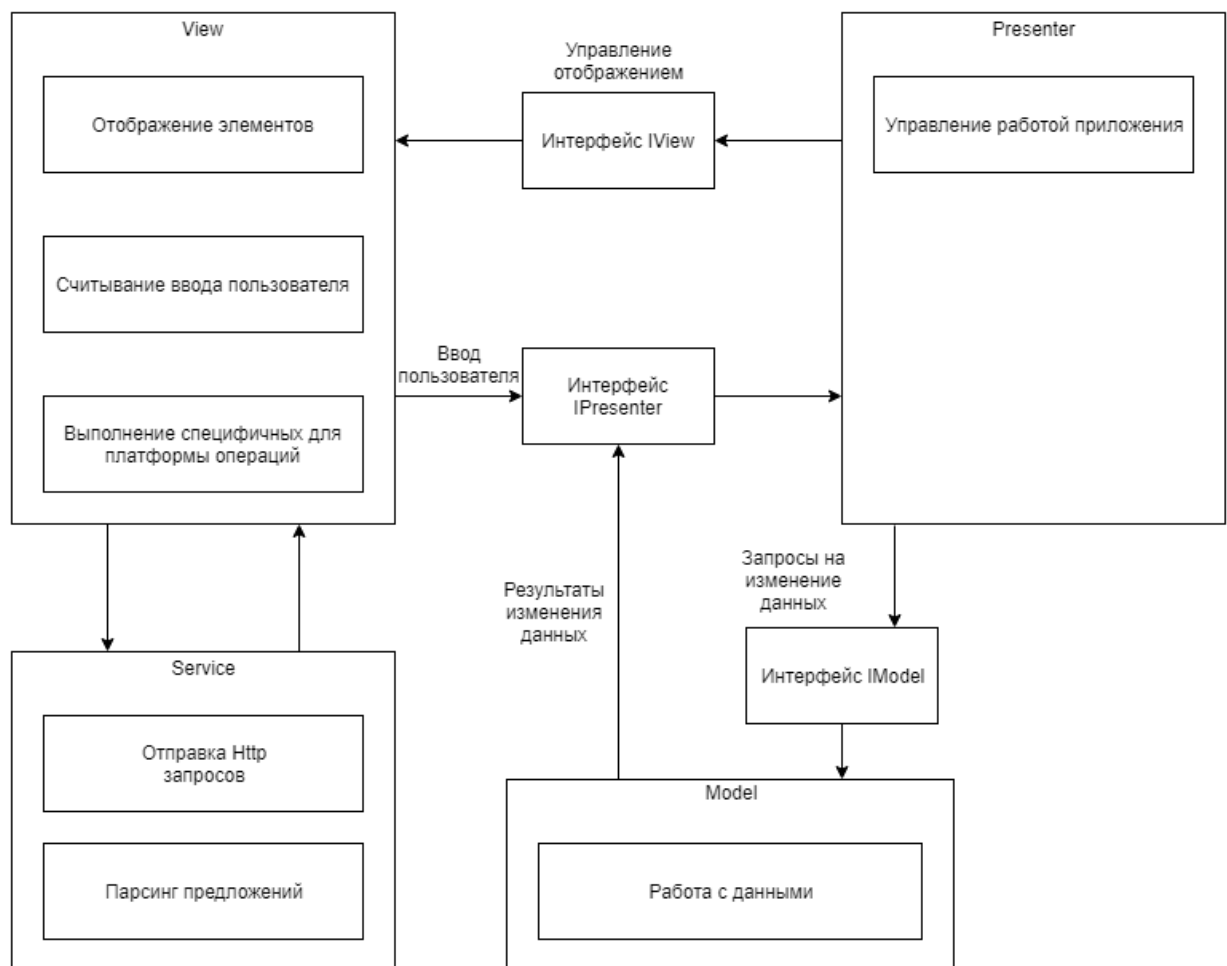


Рисунок 1 – Общая схема взаимодействия модулей приложения

Программа состоит из четырех основных категорий компонентов:

- View – отвечает за отображение элементов приложения, считывание ввода пользователя и выполнение специфических для платформы операций, например – запуск сервиса, для работы приложения в фоне;
- Presenter – управляет работой приложения, маршрутизирует запросы между модулями View и Model. А также решает, как и когда View будет отображать изменения, происходящие в Model;
- Model – хранит и обрабатывает всю информацию в приложении, такую как: текущее состояние системы, данные для фильтров, текущие загруженные предложения;
- Service – выполняет HTTP-запросы по заданному URL, парсит возвращаемый текст-HTML, на основе полученных данных создает объекты объявлений [7].

Взаимодействие между четырьмя модулями реализовано через интерфейсы. Сделано это для упрощенного дополнения приложения функционалом, и ограничения возможности отступления от архитектуры приложения. Например, View будет иметь в своем распоряжении только интерфейс IPresenter, что не позволит ей каким-либо образом влиять напрямую на модель.

В модулях Presenter и Model, их соответствующих интерфейсах, а также в интерфейсе IView нет привязки к определенной ОС. Это позволяет переносить приложение на другие платформы. При переносе приложения на другую платформу, это позволит использовать те же модули, к которым необходимо будет дополнить реализацию интерфейса IView.

2.2 Общий алгоритм работы программы

Далее будет описан общий принцип работы программы.

1. Инициализация – после запуска приложения начнется парсинг файлов JSON со статичными данными и выделение памяти под переменные [8].

2. Настройка – приложение не может работать, пока пользователь не выберет параметры поиска и фильтрации. До того, как это случится, приложение будет ожидать ввода. После ввода приложение сформирует строку URL, по которой далее будет обращаться сервис.

3. Работа сервиса – после настройки сервис начинает обращаться по сформированной для него ссылке, и парсить полученные по ней данные. Происходить это будет с некоторой задержкой, это нужно для того, чтобы не использовать все ресурсы устройства.

4. Оповещение пользователя – в ходе работы, после того, как сервис найдет какое либо предложение, сформируется push-уведомление, которое должно оповестить пользователя.

5. Вывод данных – независимо от результатов поиска сервиса, приложение ожидает ввода от пользователя, и выводит на экран последние предложения, которые были найдены.

На рисунке 2 представлена блок-схема работы программы.

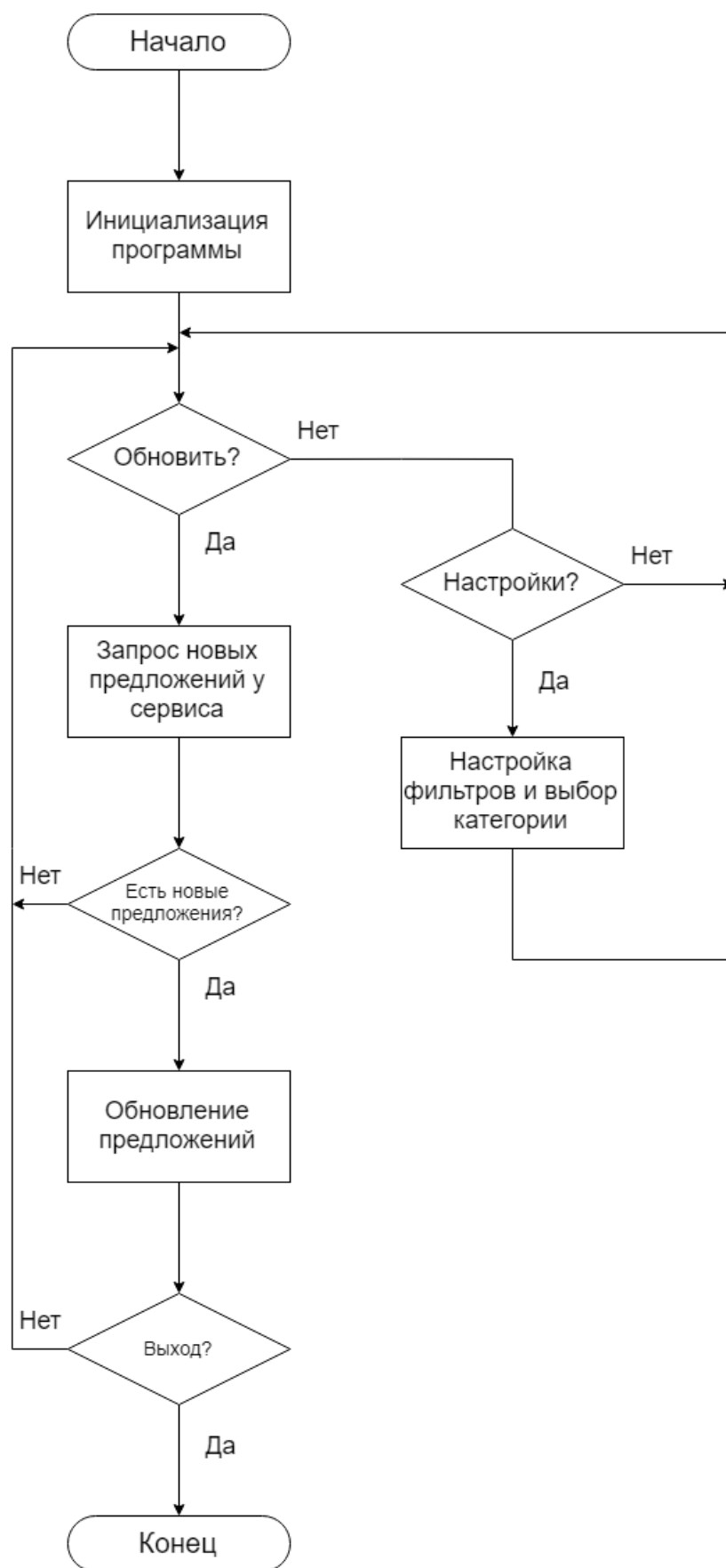


Рисунок 2 – Блок-схема работы программы

2.3 Выбор инструментов для разработки

Средой разработки был выбран Xamarin.Android, так как он предоставляет широкий спектр инструментов для создания приложений на мобильной ОС Android.

При разработке в этой среде используется язык C#, этот язык имеет множество официальных и неофициальных библиотек, которые упрощают процесс разработки. Из этих библиотек были использованы следующие:

- системные библиотеки для работы с HTTP (HttpClient) [9];
- системные библиотеки стандартных контейнеров (List) [10];
- расширение для стандартных контейнеров с поддержкой LINQ [11];
- библиотека Newtonsoft.Json – упрощает взаимодействие с файлами JSON [12].

2.4 Хранение данных

Так как программа не требует хранения больших объемов данных, и, во время выполнения программы, меняться будут только данные, полученные из сети, использование базы данных не требуется.

В данном случае хранить статичные данные, такие как варианты выбора для фильтров и доменные имена сайтов, будет целесообразно в файлах с форматированием JSON. Это позволит расширять или обновлять данные для работы программы, при этом, не редактируя исходный код.

2.5 Вывод по разделу

При проектировании приложения был выбран паттерн разработки MVP, формат JSON для организации хранения данных, выбраны необходимые инструменты для разработки, описан общий алгоритм работы программы.

3 Разработка программы

3.1 Модули архитектуры

3.1.1 Model

На рисунке 3 представлена Unified Modeling Language (UML) - диаграмма классов модуля Model [13].

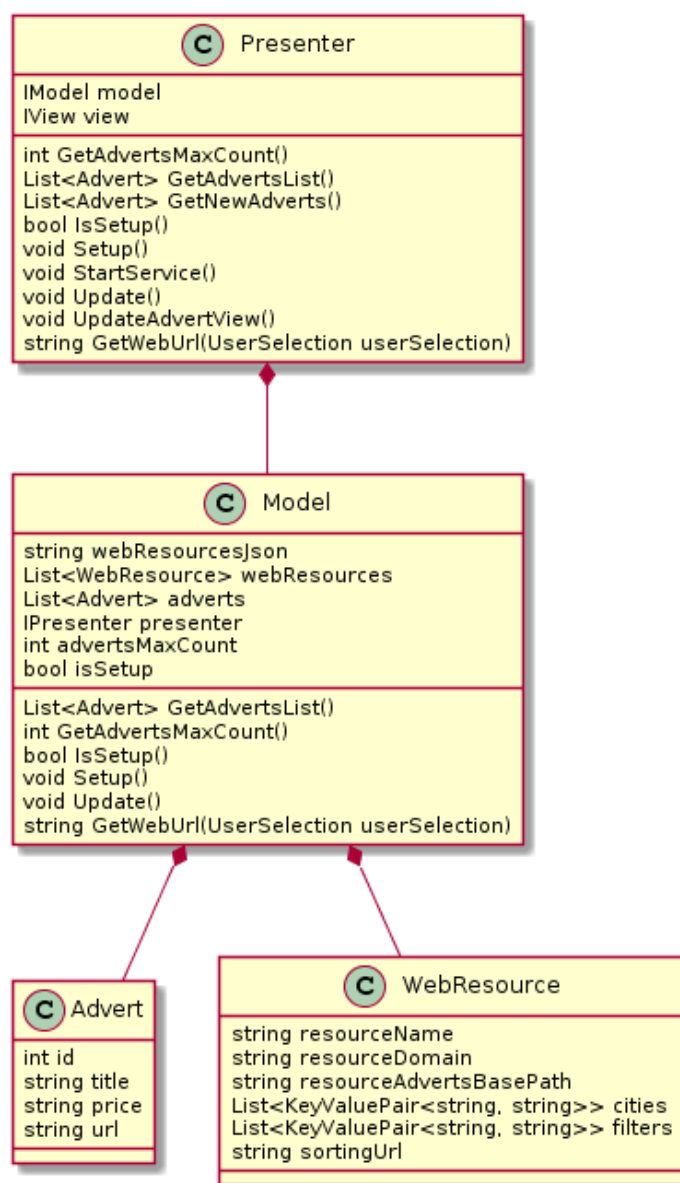


Рисунок 3 – Диаграмма классов модуля Model

Модуль Model состоит из 3 классов:

- Model – отвечает за хранение и изменение данных в виде найденных предложений, а так же за инициализацию приложения и парсинг статичных данных. Имеет в своем распоряжении интерфейс класса Presenter;
- Advert – оболочка для хранения данных по предложениям;
- WebResource – оболочка для хранения статичных данных для выбора пользователя. Хранит в себе название ресурса, его домен, соответствие URL-строки и городов и фильтров, а так же URL для сортировки.

3.1.2 Presenter

На рисунке 4 представлена диаграмма классов модуля Presenter.

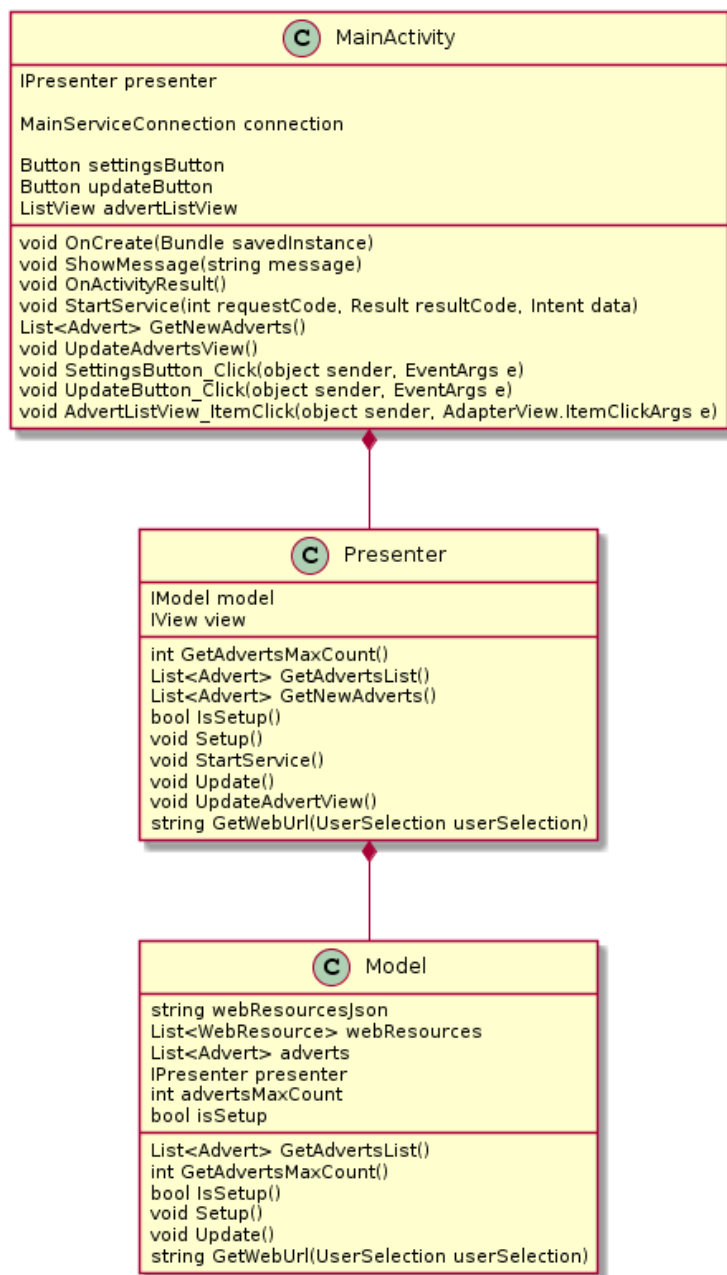


Рисунок 4 – Диаграмма классов модуля Presenter

Модуль Presenter состоит из одного класса, который отвечает за маршрутизацию команд в приложении и связывание всех модулей вместе.

Модуль View состоит из следующих классов:

- MainActivity – наследуется от стандартного класса активности Android.OS.AppCompatActivity. Является точкой входа в приложение. Создает объект класса Presenter и, после соответствующей команды, сервис. Описывает главный экран приложения;

- MainServiceConnection – наследуется от Java.Lang.Object. Выполняет роль связующего между приложением и сервисом;

- SettingsActivity – наследуется от стандартного класса активности Android.OS.AppCompatActivity. Описывает экран с настройками, после ввода извлекает введенные данные и отправляет на обработку;

- UserSelection – оболочка для хранения выбранных пользователем данных.

3.1.4 Service

На рисунке 6 представлена диаграмма классов модуля Service.

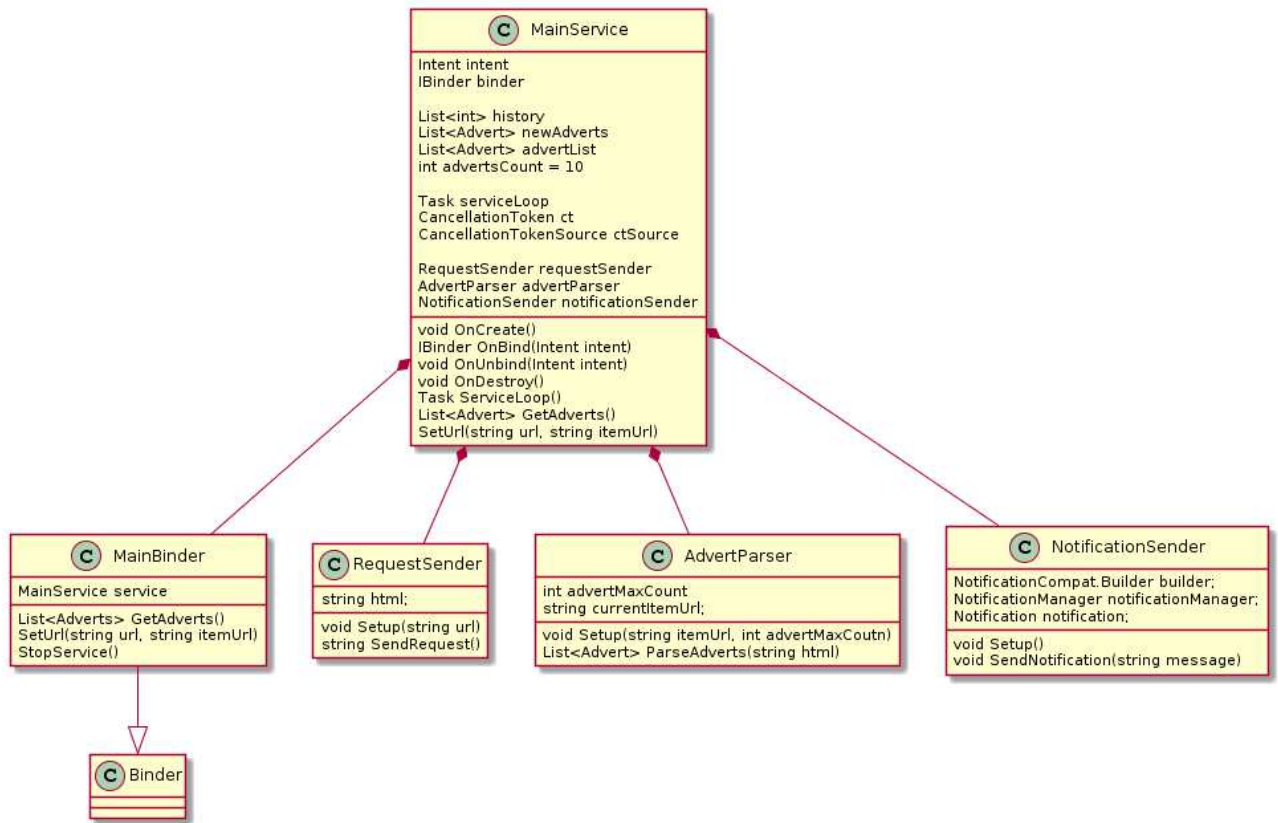


Рисунок 6 – Диаграмма классов модуля Service

Модуль Service состоит из следующих классов:

- MainService – наследуется от стандартного класса сервиса Service. С помощью классов-помощников отправляет HTTP-запросы по заданной URL-строке, и парсит содержимое ответов;
- MainBinder – наследуется от класса Android.OS.Binder, позволяет вызывать методы из запущенного сервиса;
- RequestSender – вспомогательный класс, создает соединение и асинхронно отправляет HTTP-запросы по указанному URL;

- AdvertParser – вспомогательный класс, получает HTML от объекта класса RequestSender, парсит содержимое HTML-строки и возвращает список объектов Advert;

- NotificationSender – вспомогательный класс, способен формировать и отправлять push-уведомления.

3.2 Диаграммы последовательностей

На рисунке 7 представлена реализация события «Update» в виде диаграммы последовательностей. Как видно на диаграмме, все запросы состояния и обработка информации происходит через Model, согласно паттерну MVP.

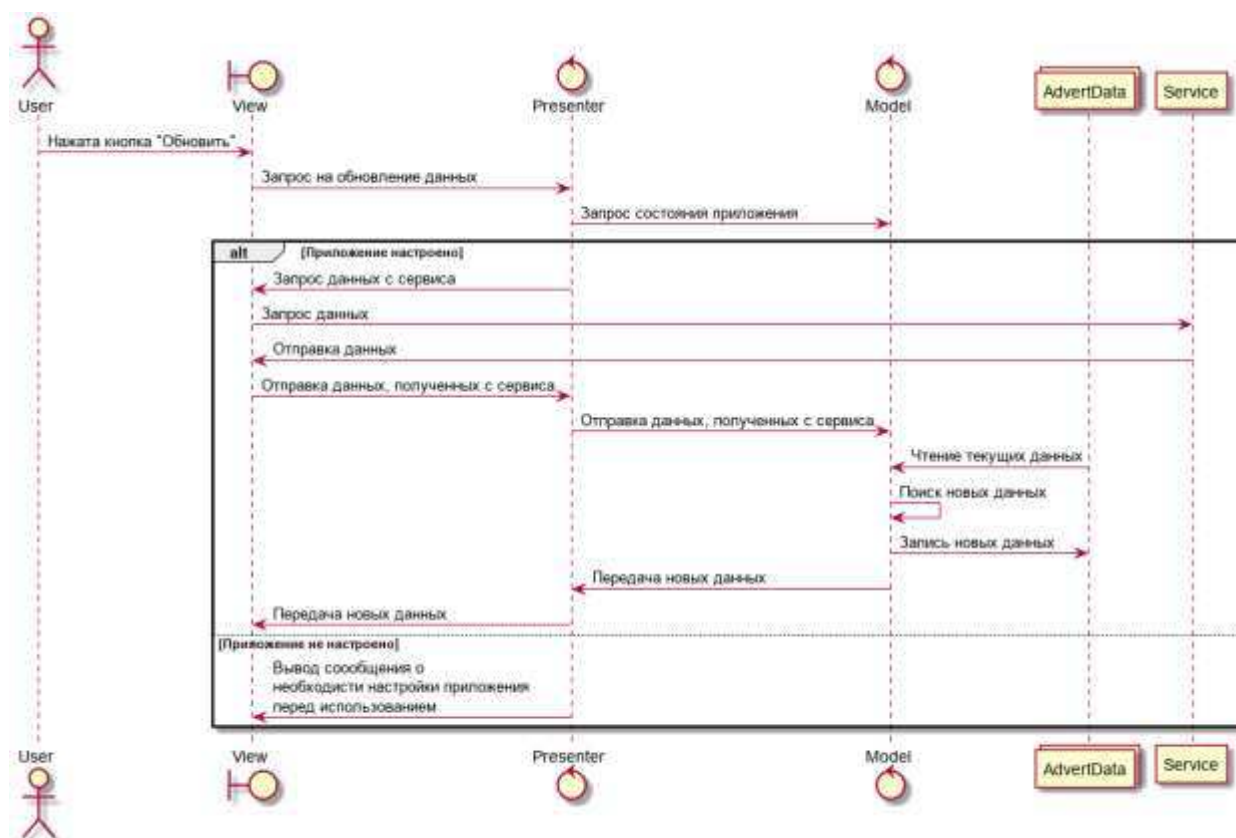


Рисунок 7 – Диаграмма последовательностей события «Update»

На рисунке 8 представлена реализация настройки приложения в виде диаграммы последовательностей. Так же взаимодействие с данными через Model, но настройка сервиса происходит через View, потому что сервис является ОС-зависимым модулем.

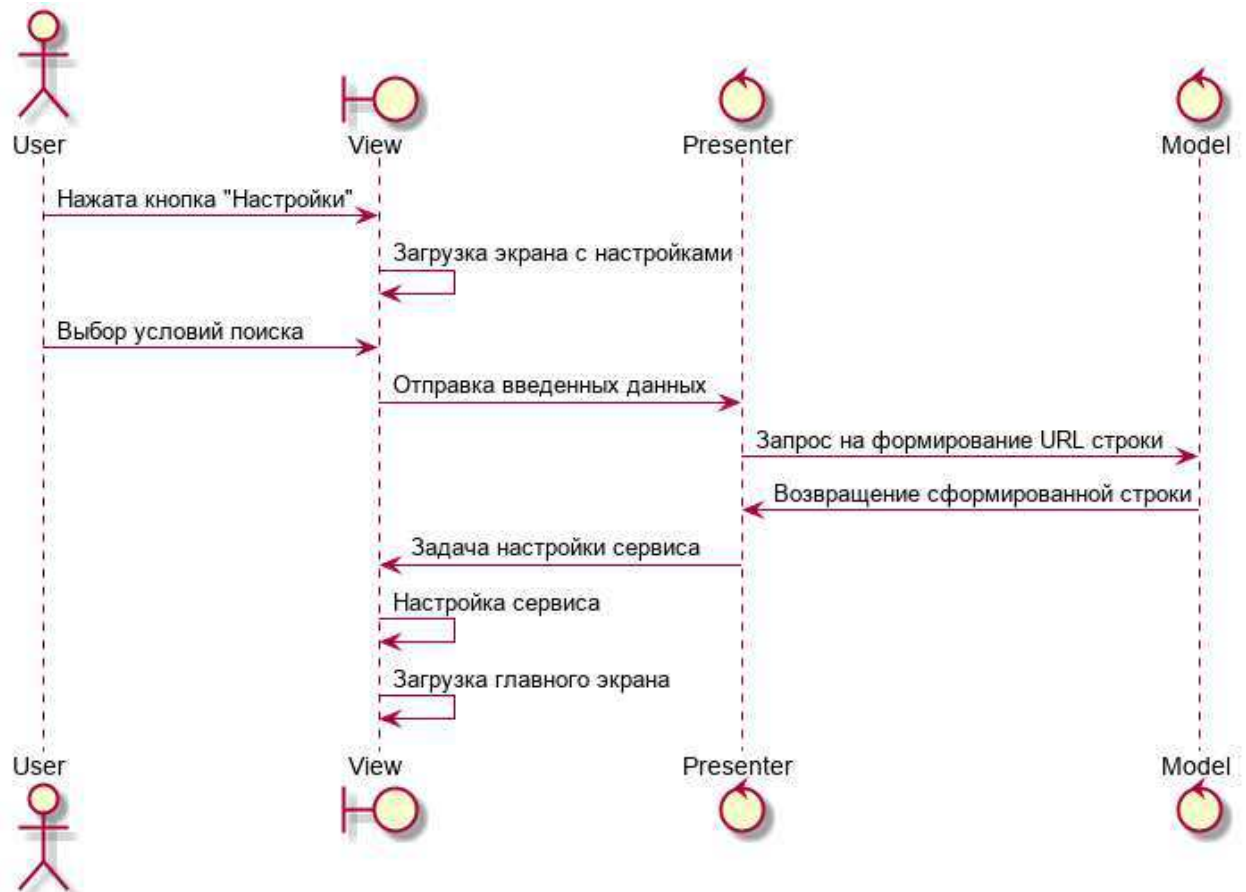


Рисунок 8 – Диаграмма последовательностей события настройки приложения

На рисунке 9 представлена реализация работы сервиса в виде диаграммы последовательностей. Перед запуском сервис ожидает URL-строку. После этого в цикле начинает отправлять запросы и парсить содержимое. Выполняется это с задержкой, для того, чтобы не использовать все ресурсы устройства.

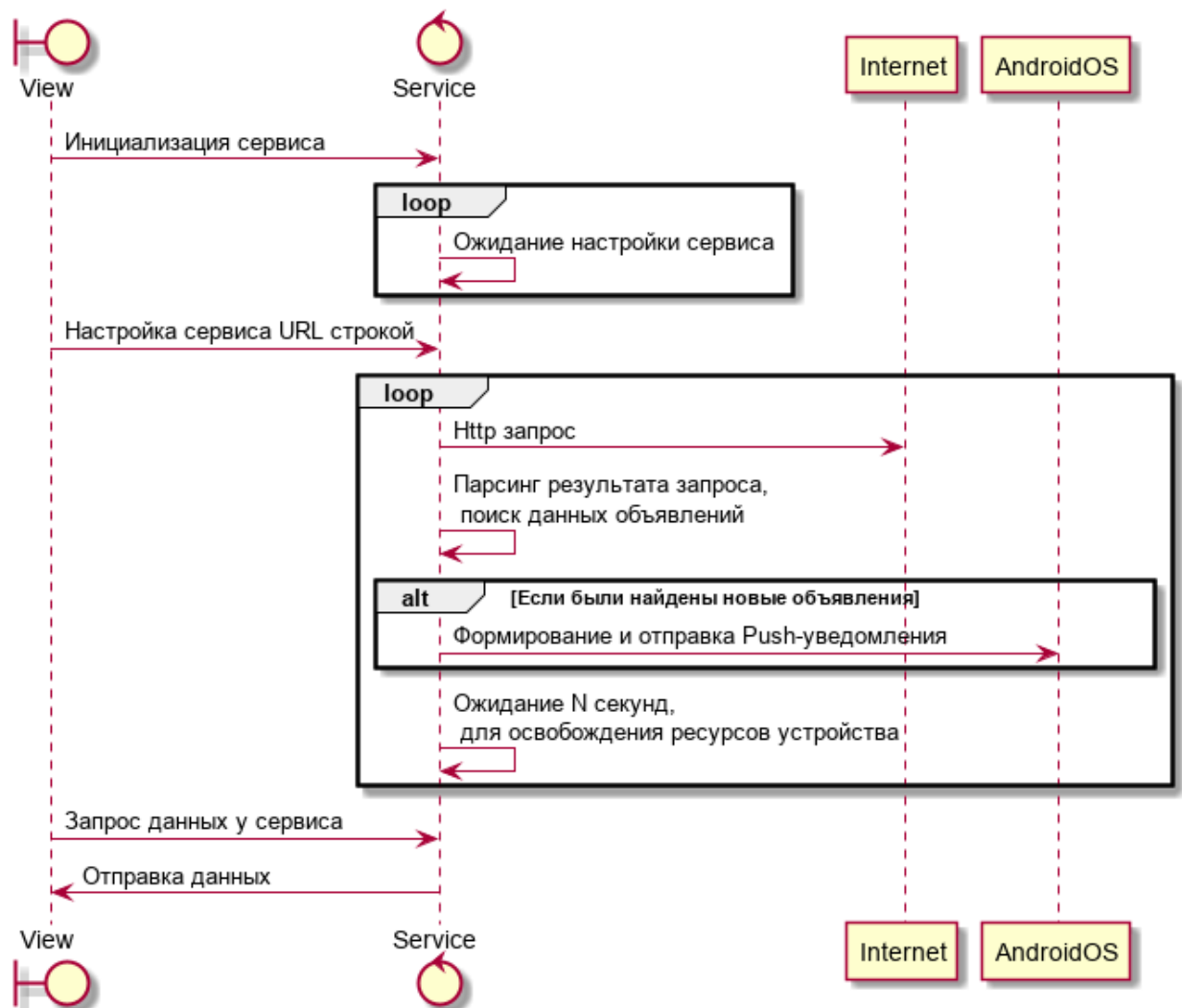


Рисунок 9 – Диаграмма последовательностей работы сервиса

3.3 Хранение данных

Статические данные, такие как домены ресурсов и части URL-строк для поиска, хранятся в файле JSON. Файл содержит в себе массив из объектов, содержащих информацию, как показано на рисунке 10.

```
{
  "resourceName" : "Avito",
  "resourceDomain" : "https://www.avito.ru",
  "resourceAdvertsBasePath" : "",
  "cities" : [
    {
      "Key" : "Красноярск",
      "Value" : "krasnoyarsk"
    }
  ],
  "filters" : [
    {
      "Key" : "Минимальная цена",
      "Value" : "rmin={0}"
    },
    {
      "Key" : "Максимальная цена",
      "Value" : "rmax={0}"
    }
  ],
  "sortingUrl" : "s=104"
}
```

Рисунок 10 – Пример объекта в формате JSON

Как видно на рисунке 10, объект состоит из следующих полей:

- resourceName – название ресурса, требуется для вывода ресурсов в список для реализации возможности выбора ресурса;
- resourceDomain – домен ресурса, необходим в качестве компонента для сборки URL-строки;
- resourceAdvertsBasePath – зарезервированное поле на случай, если после домена сайта потребуется еще одна вставка;

- cities – массив объектов типа KeyValuePair, объекты представляют из себя название города на русском языке, и его соответствие в адресной строке;
- filters – массив объектов типа KeyValuePair, объекты представляют из себя название фильтра и соответствующая ему URL-строка, с возможностью подстановки значения;
- sortingURL – часть URL-строки, необходимая для сортировки предложений по времени добавления.

3.4 Вывод по разделу

Реализовано приложение, обладающее следующим функционалом:

- сканирование предложений на определенных площадках в сети Интернет;
- использование фильтров для сужения объема найденных предложений;
- оповещение пользователя в случае обнаружения новых предложений.

4 Руководство пользователя

4.1 Установка и запуск

Для использования приложения необходимо либо скачать его из Google Play Market, на поддерживающем его устройстве, либо использовать прикрепленный к дипломной работе файл формата apk. После установки необходимо нажать на иконку установленного приложения для его запуска.

4.2 Демонстрация возможностей программы

После запуска перед пользователем появляется главный экран приложения и подсказка, говорящая о том, что для работы приложение необходимо настроить (рисунок 11).

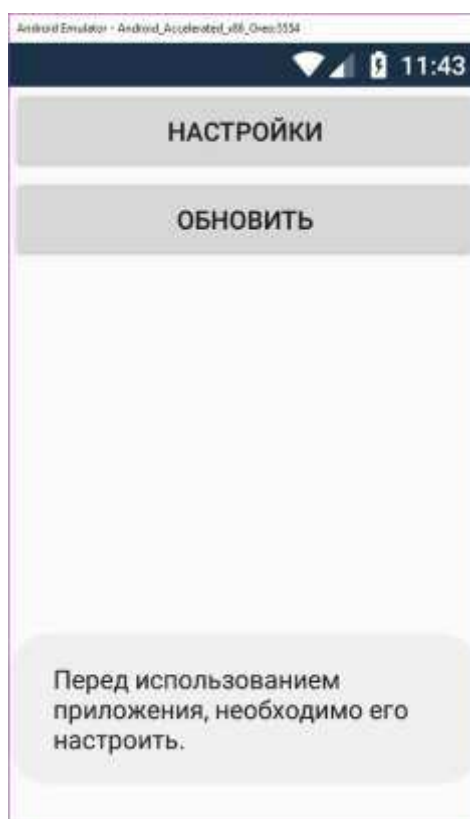


Рисунок 11 – Первый запуск

На главном экране перед пользователем представлены 2 кнопки – «Обновить» и «Настройки» (рисунок 12).

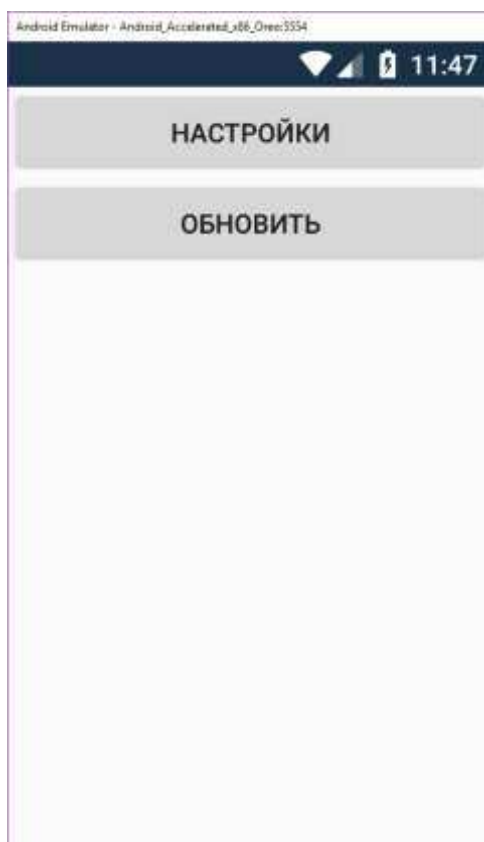


Рисунок 12 – Главный экран до настройки приложения

При нажатии на кнопку «Обновить», до настройки приложения, будет снова появляться подсказка о необходимости настроить приложение перед его использованием.

При нажатии на кнопку «Настройки» откроется экран настроек (рисунок 13).

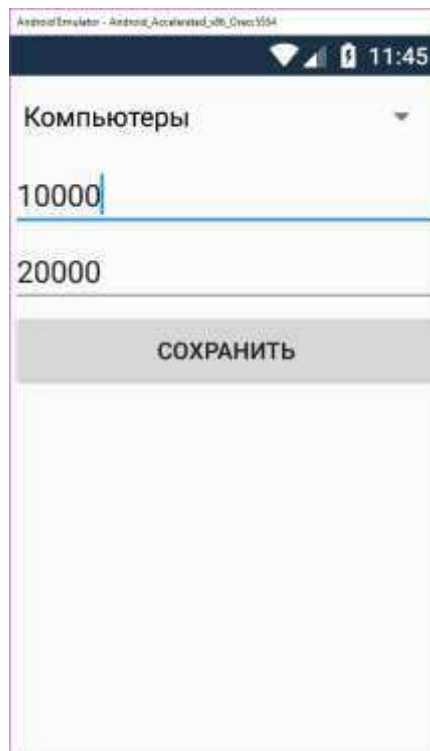


Рисунок 13 – Экран настроек

На этом экране необходимо выбрать фильтры поиска, такие как:

- цена, минимальная и максимальная;
- категория, в которой необходимо производить поиск.

Этот функционал создан для сужения круга объявлений, что, в свою очередь, увеличивает удобство использования программы.

После нажатия на кнопку «Сохранить» на экране настроек, приложение вернет пользователя к главному экрану. На этот раз программа уже настроена и готова к работе. После нажатия на кнопку «Обновить» на главном экране высветятся последние 10 объявлений в указанной категории с указанными фильтрами (рисунок 14).



Рисунок 14 – Главный экран после настройки приложения

Далее пользователь может оставить приложение в фоновом режиме и пользоваться своим телефоном как обычно. Если новое предложение будет найдено, приложение оповестит пользователя об этом с помощью push-уведомления (рисунок 15).

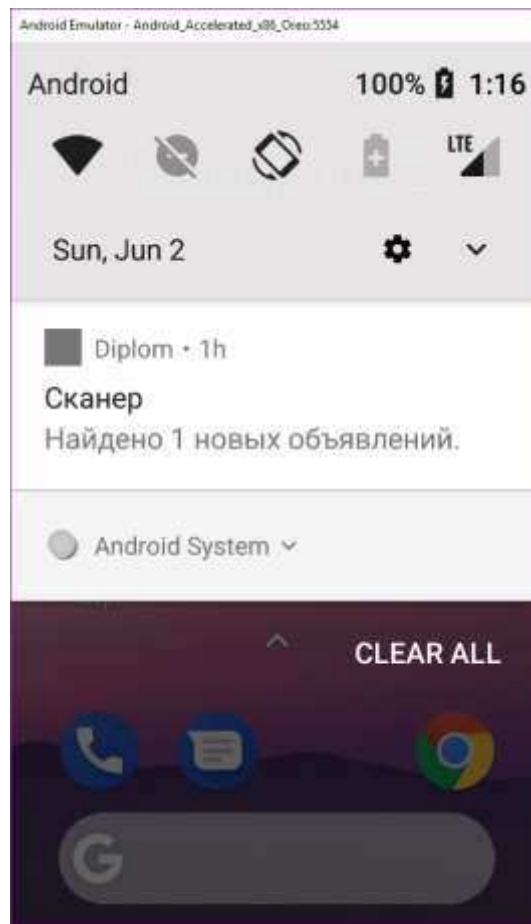


Рисунок 15 – Push-уведомление

4.3 Тестирование приложения под разными версиями Android

Работоспособность программы проверена с помощью различных средств, таких как смартфон «Fly Memory PLUS» с операционной системой Android 7.0 и на эмуляторе операционной системы Android 8.0. Для обеих систем программа работает корректно согласно техническому заданию.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы было разработано приложение под операционную систему Android, которое посредством автоматизации процесса поиска предложений, помогает пользователям вторичных торговых площадок.

Были реализованы все поставленные задачи, а именно: были изучены и проанализированы аналоги данного приложения, выявлены их особенности и недостатки, с помощью этого были поставлены требования к разрабатываемому продукту.

Для разработки приложения под ОС Android был использован оптимальный набор инструментов: Язык программирования C# и среда разработки Xamarin.Android.

Разработанное приложение будет запущено в Google Play Market, и будет развиваться в соответствии и при наличии пожеланий пользователей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Автоматический поиск объявлений в Интернет. Autodrom Engine (edrom) [Электронный ресурс] : – Режим доступа: <http://edrom.ru/>.
2. Программа для поиска автомобилей в Интернете. Cars HotSurf [Электронный ресурс] : – Режим доступа: <http://hotsurf.ru/glavnaya.aspx>.
3. Ваш помощник в автомобильном бизнесе. «Автосканер» [Электронный ресурс] : – Режим доступа: <http://www.autoscanner.biz/>.
4. Сервис Ads-Tracker [Электронный ресурс] : – Режим доступа: <https://ads-tracker.ru/page/about>.
5. Сервис Глиса [Электронный ресурс] : – Режим доступа: <http://глиса.рф/>.
6. Паттерн MVP [Электронный ресурс] : – Режим доступа: <http://rsdn.org/article/patterns/modelviewpresenter.xml>.
7. Парсинг html-сайтов [Электронный ресурс] : – Режим доступа: <http://parsing.valemak.com/ru/what-why-how/stages-of-parsing/>.
8. Введение в Json [Электронный ресурс] : – Режим доступа: <https://www.json.org/json-ru.html>.
9. HttpClient Class Microsoft Docs [Электронный ресурс] : – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.net.http.httpclient>.
10. System.Collections.Generic Namespace Microsoft Docs [Электронный ресурс] : – Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.collections.generic>.
11. Основы LINQ [Электронный ресурс] : – Режим доступа: <https://metanit.com/sharp/tutorial/15.1.php>.
12. Newtonsoft Json.NET [Электронный ресурс] : – Режим доступа: <https://www.newtonsoft.com/json>.
13. UML - диаграммы классов [Электронный ресурс] : – Режим доступа: <https://prog-cpp.ru/uml-classes/>.

Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт

Вычислительная техника
кафедра

УТВЕРЖДАЮ
Заведующий кафедрой ВТ
О.В. Непомнящий
подпись инициалы, фамилия
«27» 06 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
код и наименование направления

Сканер предложений для торговых площадок в сети Интернет
тема

Руководитель	<u>Мамы</u> <u>25.06.19</u> подпись, дата	ст. преподаватель должность, ученая степень	<u>И. В. Матковский</u> инициалы, фамилия
Выпускник	<u>Анб</u> <u>25.06.19</u> подпись, дата		<u>Ю. В. Аксенов</u> инициалы, фамилия
Консультант	<u>Л. И.</u> <u>25.06.19</u> подпись, дата	доцент, канд. техн. наук должность, ученая степень	<u>Л. И. Покидьшева</u> инициалы, фамилия
Нормоконтролер	<u>Иванов</u> <u>26.06.19</u> подпись, дата	доцент, канд. техн. наук должность, ученая степень	<u>В. И. Иванов</u> инициалы, фамилия

Красноярск 2019