

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Модели данных и системы управления базами данных

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту
на тему

«RSS Reader»

Студент гр. 753501
Романовский В.Д.
Руководитель
Удовин И. А.

Минск 2020

Содержание

Дисциплина: Модели данных и системы управления базами данных	1
Введение	3
Краткие теоретические сведения	4
Технологии и средства разработки.....	5
Модули программы и их описание	9
Проектирование базы данных	15
Тестирование.....	16
Методика использования	19
Заключение	22
Приложение	24

Введение

Интернет представляет собой идеальный источник для получения информации, однако порой бывает трудно разобраться в огромном потоке информации. С этим могут помочь специальные приложения – агрегаторы новостей.

В рамках данной курсовой работы будет разработано приложение Android. Тема моей работы: RSS reader, приложение, с помощью которого пользователи смогут быть в курсе самых важных и интересных новостей.

Основной целью работы является демонстрация моих навыков разработки и создания баз данных.

Краткие теоретические сведения

RSS (англ. Rich Site Summary или Really Simple Syndication) — семейство XML-форматов, предназначенных для описания лент новостей, анонсов статей, изменений в блогах и т. п. Информация из различных источников, представленная в формате RSS, может быть собрана, обработана и представлена пользователю в удобном для него виде специальными программами-агрегаторами или онлайн-сервисами.

База данных — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов. Базы данных активно используются для динамических сайтов со значительными объемами данных — часто это интернет-магазины, порталы, корпоративные сайты. Такие сайты обычно разработаны с помощью серверного языка программирования (как пример, PHP) или на основе CMS (как пример, WordPress), и не имеют готовых страничек с данными по аналогии с HTML-сайтами. Странички динамических сайтов формируются «на лету» в результате взаимодействия скриптов и баз данных после соответствующего запроса клиента к веб-серверу.

Система управления базами данных — это комплекс программных средств, необходимых для создания структуры новой базы, ее наполнения, редактирования содержимого и отображения информации. Наиболее распространенными СУБД являются MySQL, PostgreSQL, Oracle, Microsoft SQL Server.

Реляционная база данных — это набор данных с предопределенными связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк. В таблицах хранится информация об объектах, представленных в базе данных. В каждом столбце таблицы хранится определенный тип данных, в каждой ячейке — значение атрибута. Каждая строка таблицы представляет собой набор связанных значений, относящихся к одному объекту или сущности. Каждая строка в таблице может быть помечена уникальным идентификатором, называемым первичным ключом, а строки из нескольких таблиц могут быть связаны с помощью внешних ключей. К этим данным можно получить доступ многими способами, и при этом реорганизовывать таблицы БД не требуется.

Технологии и средства разработки

Android Studio, основанная на программном обеспечении IntelliJ IDEA от компании JetBrains, — официальное средство разработки Android приложений. Данная среда разработки доступна для Windows, macOS и Linux.



Рисунок 2.1 Значок Android Studio

Возможности Android Studio:

- Расширенный редактор макетов: способность работать с UI компонентами при помощи Drag-And-Drop, функция предпросмотра макета на нескольких конфигурациях экрана.
- Сборка приложений, основанная на Gradle.
- Различные виды сборок и генерация нескольких .apk файлов
- Рефакторинг кода
- Статический анализатор кода (Lint), позволяющий находить проблемы производительности, несовместимости версий и другое.
- Встроенный ProGuard и утилита для подписывания приложений.
- Шаблоны основных макетов и компонентов Android.
- Поддержка разработки приложений для Android Wear и Android TV.
- Встроенная поддержка Google Cloud Platform, которая включает в себя интеграцию с сервисами Google Cloud Messaging и App Engine.
- Android Studio 2.1 поддерживает Android N Preview SDK, а это значит, что разработчики смогут начать работу по созданию приложения для новой программной платформы.

- Новая версия Android Studio 2.1 способна работать с обновленным компилятором Jack, а также получила улучшенную поддержку Java 8 и усовершенствованную функцию Instant Run.
- В Android Studio 3.0 по стандарту включены инструменты языка Kotlin основанные на JetBrains IDE.

SQLite – компактная встраиваемая СУБД.

Слово «встраиваемый» (embedded) означает, что SQLite не использует парадигму клиент-сервер, то есть движок SQLite не является отдельно работающим процессом, с которым взаимодействует программа, а представляет собой библиотеку, с которой программа компонуется, и движок становится составной частью программы. Таким образом, в качестве протокола обмена используются вызовы функций (API) библиотеки SQLite. Такой подход уменьшает накладные расходы, время отклика и упрощает программу. SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в единственном стандартном файле на том компьютере, на котором исполняется программа. Простота реализации достигается за счёт того, что перед началом исполнения транзакции записи весь файл, хранящий базу данных, блокируется; ACID-функции достигаются в том числе за счёт создания файла журнала.



Рисунок 2.2 Значок SQLite

Несколько процессов или потоков могут одновременно без каких-либо проблем читать данные из одной базы. Запись в базу можно осуществить только в том случае, если никаких других запросов в данный момент не обслуживается; в противном случае попытка записи оканчивается неудачей, и в программу возвращается код ошибки. Другим вариантом развития событий является автоматическое повторение попыток записи в течение заданного интервала времени.

SQLite поддерживает динамическое типизирование данных. Возможные типы значений: INTEGER, REAL, TEXT и BLOB. Также поддерживается специальное значение NULL.

Размеры значений типа TEXT и BLOB не ограничены ничем, кроме константы SQLITE_MAX_LENGTH в исходниках sqlite, равной миллиарду (10^9).

Каждое значение в любом поле любой записи может быть любого из этих типов, независимо от типа, указанного при объявлении полей таблицы. Указанный при объявлении поля тип хранится для справки в его исходном написании, и используется в качестве основы для выбора предпочтений (так называемое «type affinity»: это подход, редко встречающийся в других СУБД) при выполнении неявных преобразований типов на основании похожеści этого названия типа на что-либо, знакомое SQLite. В этот алгоритм зашит обширный перечень практикуемых в других СУБД вариантов названий типов данных. Если безопасного преобразования записываемого значения в предпочитаемый тип не получается, SQLite записывает значение в его исходном виде. Для получения значений из базы есть ряд функций для каждого из типов, и если тип хранимого значения не соответствует запрашиваемому, оно тоже, по возможности, преобразуется.

Java

Программы на Java транслируются в байт-код Java, выполняемый виртуальной машиной Java (JVM) — программой, обрабатывающей байтовый код и передающей инструкции оборудованию как интерпретатор.

Достоинством подобного способа выполнения программ является полная независимость байт-кода от операционной системы и оборудования, что позволяет выполнять Java-приложения на любом устройстве, для которого существует соответствующая виртуальная машина. Другой важной особенностью технологии Java является гибкая система безопасности, в рамках которой исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером), вызывают немедленное прерывание.

Если достаточно формально подходить к вопросу о типизации языка Java, то можно сказать, что он относится к языкам со статической типизацией.

Явная типизация означает, что при объявлении переменной вам нужно указывать её тип, при неявной — это делать необязательно. В качестве примера языков с явной типизацией можно привести Python, JavaScript.



Рисунок 2.3 Значок Java

Java не поддерживает динамическую типизацию, то есть тип переменной не определяется во время исполнения. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем». В Java имеются встроенные типы: булевый, строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. Из коллекций в язык Java встроены: список, кортеж (неизменяемый список), словарь, множество и другие.

XML (англ. eXtensible Markup Language) — расширяемый язык разметки. Рекомендован Консорциумом Всемирной паутины (W3C). Спецификация XML описывает XML-документы и частично описывает поведение XML-процессоров (программ, читающих XML-документы и обеспечивающих доступ к их содержимому). XML разрабатывался как язык с простым формальным синтаксисом, удобный для создания и обработки документов программами и одновременно удобный для чтения и создания документов человеком, с подчёркиванием нацеленности на использование в Интернете. Язык называется расширяемым, поскольку он не фиксирует разметку, используемую в документах: разработчик волен создать разметку в соответствии с потребностями к конкретной области, будучи ограниченным лишь синтаксическими правилами языка.

В Android-разработке используется для создания макетов.

Модули программы и их описание

В результате проектирования и разработки получился следующий набор классов:

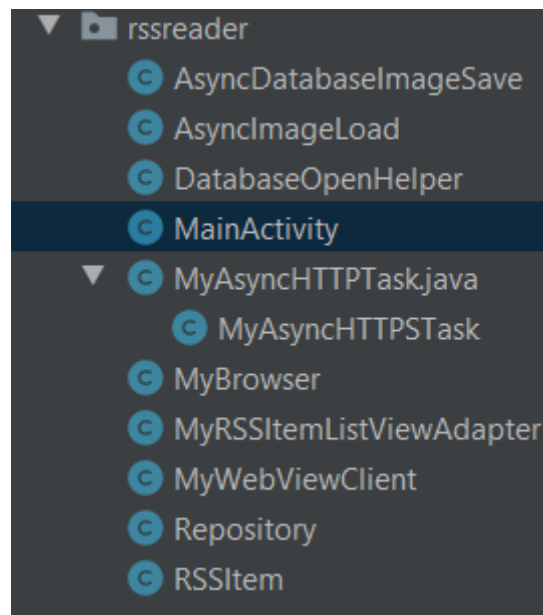


Рисунок 3.1 Классы приложения

Класс **MainActivity** представляет собой главный экран приложения (Activity). В данном классе обрабатываются события, создаются необходимые объекты для работы с базой данных, а также загружаются файлы для корректного отображения пользовательского интерфейса.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    if (!isPermissionGranted(INTERNET_PERMISSION)) {
        requestPermission(INTERNET_PERMISSION, REQUEST_INTERNET_PERMISSION_CODE);
    }
}
```

Рисунок 3.2 Загрузка интерфейса

Кроме того, в классе **MainActivity** производится управление асинхронными задачами для загрузки данных.

Класс **MyBrowser** является еще одним классом Activity. С его помощью отображаются веб страницы в приложении.

Оба класса Activity, как и положено, прописаны в манифесте приложения.

```
<activity android:name=".MyBrowser"></activity>
<activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

Рисунок 3.3 AndroidManifest

Класс **MyWebViewClient** помогает системе решить, стоит ли открывать веб-страницы во встроенном в приложение браузере или в стандартном браузере устройства.

```
@Override
public boolean shouldOverrideUrlLoading(WebView view, WebResourceRequest request) {
    view.loadUrl(request.getUrl().toString());
    return true;
}
```

Рисунок 3.4 Класс MyWebViewClient

Класс **DatabaseOpenHelper** помогает работать с базой данных. С его помощью можно получить объект SQLiteDatabase для работы с базой данных, а также позволяет удобно управлять версиями базы данных.

```
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("drop table if exists rssitem");
    onCreate(db);
}
```

Рисунок 3.5 Метод, вызываемый при увеличении версии базы данных

Класс **Repository** получает объект SQLiteDatabase для работы с базой данных с помощью **DatabaseOpenHelper**, а также содержит в методах запросы к базе данных. Перед работой требуется вызвать метод open(), а после вызвать метод close().

```

public void insertItems(ArrayList<RSSItem> list)
{
    for (RSSItem item: list)
        insertItem(item);
}

```

Рисунок 3.6 Класс Repository

Класс **RSSItem** используется для хранения новостей в памяти и для их отображения.



Рисунок 3.7 Представление для элемента в списке новостей

Класс **MyRSSItemListViewHolder** используется для отображения списка RSSItem. Адаптер проходит по списку и для каждого элемента генерирует представление с нужной информацией, а затем добавляет это представление (объект View) в список.

```

@Override
public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

    LayoutInflater inflater = (LayoutInflater)context.getSystemService(context.LAYOUT_INFLATER_SERVICE);
    View view = inflater.inflate(R.layout.rss_item, parent, attachToRoot: false);
    RSSItem item = items.get(position);
    //TODO: image
    TextView title = view.findViewById(R.id.title_view);
    TextView date = view.findViewById(R.id.date_view);
    TextView preview = view.findViewById(R.id.preview_view);
    ImageView image = view.findViewById(R.id.image_view);
    title.setText(item.getTitle());
    date.setText(item.getDate());
    preview.setText(item.getPreview());
}

```

Рисунок 3.8 Класс MyRSSItemListViewHolder

Класс **MyAsyncHTTPSTask** используется для асинхронной загрузки списка **RSSItem** с сайтов интернет ресурсов. В случае ошибки загрузки выводится сообщение об ошибке. Класс получает rss файл и с его помощью формирует список **RSSItem**.

```
<item>
  <title>Xiaomi обогнала Amazon, а Facebook оказался в конце рейтинга инновационных компаний</title>
  <id>tag:dev.by,2020-12-14:26408305</id>
  <pubDate>Mon, 14 Dec 2020 12:54:12 GMT</pubDate>
  <description>
    Журнал Fortune опубликовал ежегодный рейтинг наиболее привлекательных для инвестиций компаний. Результаты оказались неоднозначными.
  </description>
  <author>Николай Чикишев</author>
  <link>https://dev.by/news/xiaomi-obognala-amazon-a-facebook-okazalsya-v-kontse-reitinga-innovatsionnyh-kompanii</link>
  <guid>https://dev.by/news/xiaomi-obognala-amazon-a-facebook-okazalsya-v-kontse-reitinga-innovatsionnyh-kompanii</guid>
  <enclosure url="https://dev.by/storage/images/46/83/83/76/derived/0455332734875309edff81c3d1fc2614.jpg" type="image/jpeg"/>
</item>
```

Рисунок 3.9 RSS файл

Обработка rss файла производится в методе **getRSSItems()** с помощью регулярных выражений и DOM.

```
if (list.getLength() > 0) {
    Element descriptionElement = (Element) list.item( index: 0);
    description = descriptionElement.getTextContent();

    Pattern p = Pattern.compile("src=\".+?\"");
    Matcher m = p.matcher(description);
    if (m.find()) {
        String str = m.group();
        extraURL = str.substring(5, str.length() - 1);
        Log.d(TAG, extraURL);
    }

    Pattern pattern = Pattern.compile("<.+?>");
    Matcher matcher = pattern.matcher(description);
    String newStr = matcher.replaceAll( replacement: "");
    description = newStr;
}
```

Рисунок 3.10 Обработка RSS

Класс **AsyncImageLoad** проходит по списку **RSSItem** и выполняет асинхронную загрузку изображений.

Класс **AsyncDatabaseImageSave** сохраняет полученные изображения в базу данных.

```

@Override
protected Integer doInBackground(ArrayList<RSSItem>... arrayList) {
    ArrayList<RSSItem> list = arrayList[0];
    repository.clearImages();
    for (RSSItem item: list)
    {
        if (item.getBitmap() != null)
        {
            repository.insertBitmap(item.getBitmap(), item.getImageUrl());
        }
    }
    return 0;
}

```

Рисунок 3.11 Сохранение изображений в базу данных

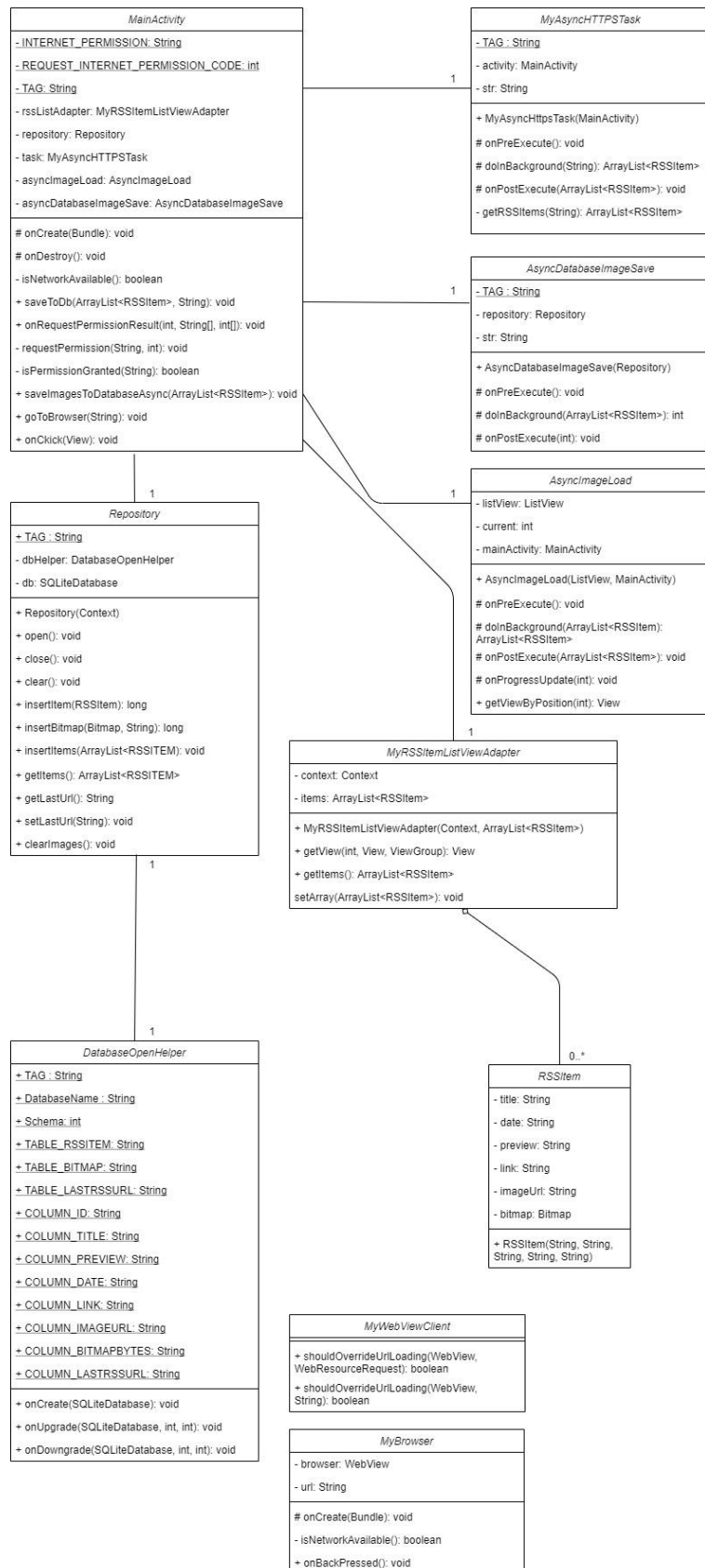


Рисунок 3.12 Диаграмма классов приложения

Проектирование базы данных

В результате проектирования базы данных была получена следующая схема данных:

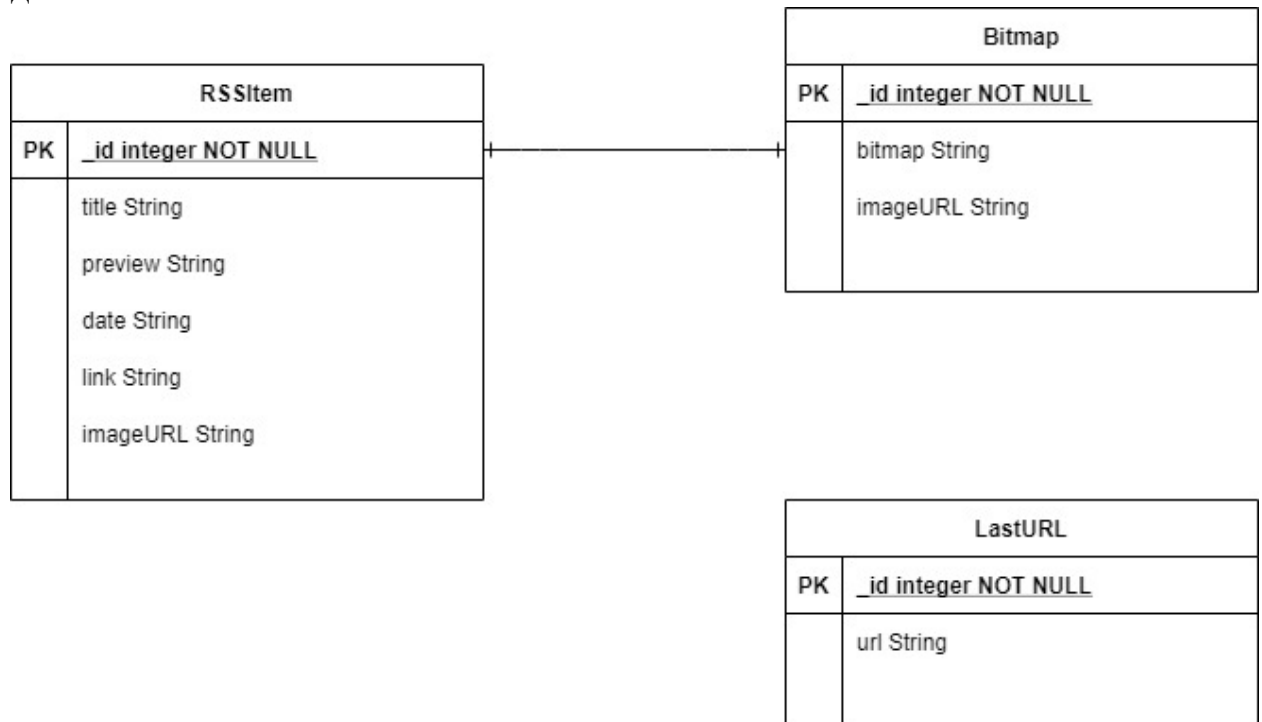


Рисунок 4.1 Схема базы данных

В таблице RSSItem хранятся объекты класса RSSItem.

Таблица Bitmap используется для хранения картинок.

В таблице LastURL хранится последний url, к которому был выполнен запрос на загрузку.

```
@Override
public void onCreate(SQLiteDatabase db) {
    db.execSQL("create table if not exists rssitem (_id integer primary key autoincrement, " +
        "title String, preview String, date String, link String, imageURL String);");
    db.execSQL("create table if not exists bitmap (_id integer primary key autoincrement, imageURL String, bitmapbytes String);");
    db.execSQL("create table if not exists lastrssurl (_id integer primary key autoincrement, last String);");
}
```

Рисунок 4.2 Создание таблиц в базе данных

Тестирование

Запустим приложение. Приложение запустилось без ошибок. Введем в поле source адрес dev.by/rss. Нажмем на кнопку load.

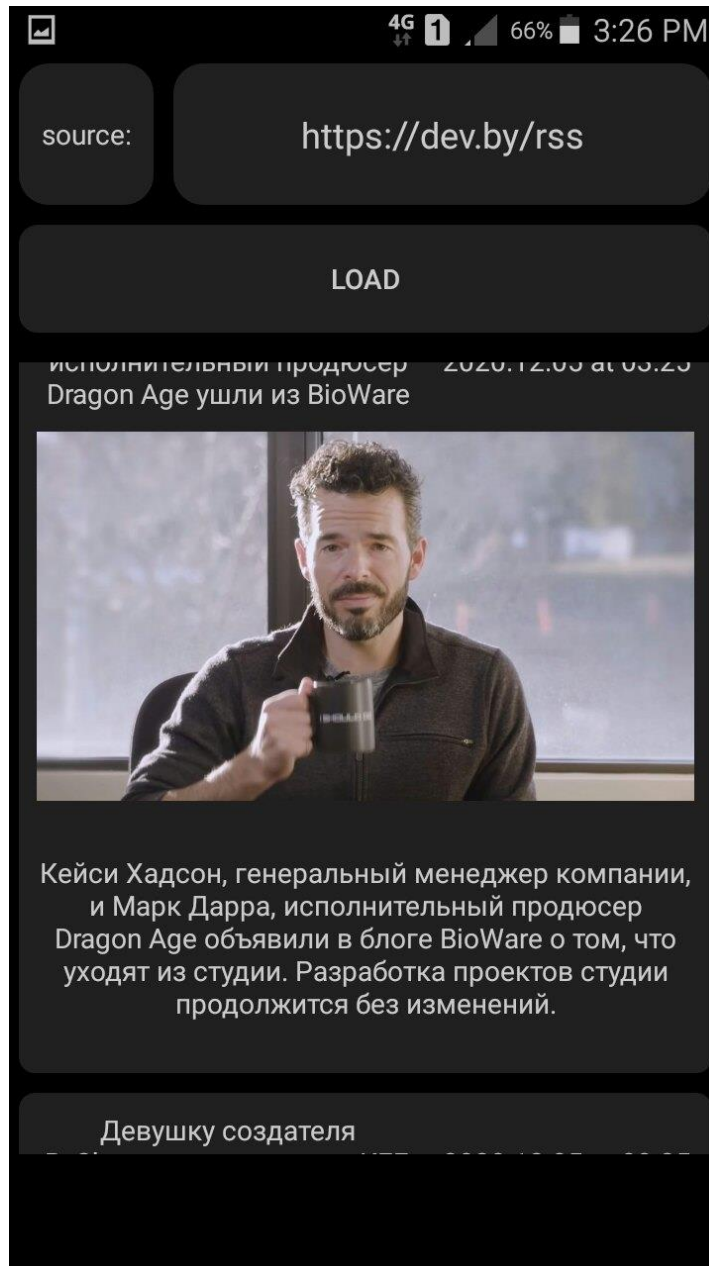


Рисунок 5.1 Список актуальных новостей сайта dev.by
Новости корректно отображаются, список можно листать.

Нажмем на один из элементов списка. Открылся встроенный браузер с полной версией новости.

Теперь введем ссылку на какой-нибудь другой rss ресурс. Загрузим новости. Их список корректно отобразился.

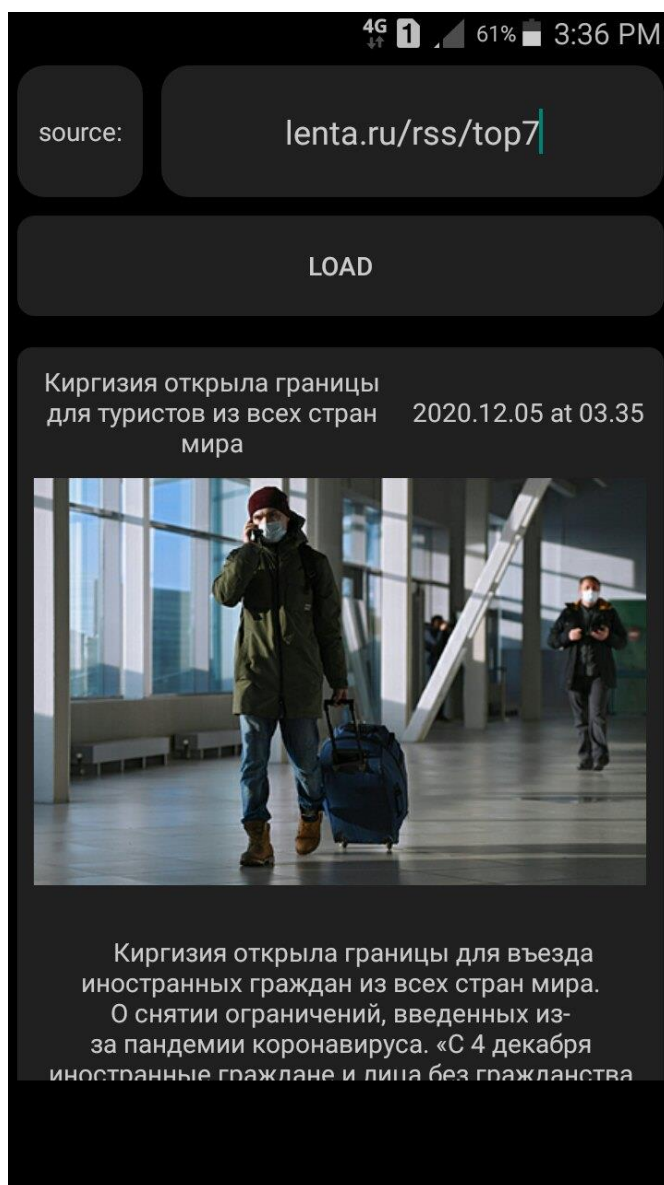


Рисунок 5.2 Список новостей с сайта lenta.ru

Попробуем ввести некорректный адрес. Отобразилась ошибка загрузки.

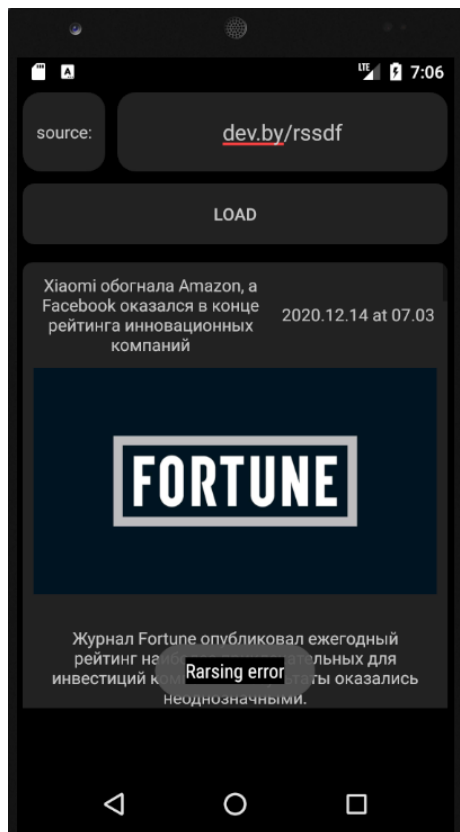


Рисунок 5.3 Ошибка загрузки при введении некорректного адреса. Теперь проверим способность программы сохранять данные между запусками программы. Для этого закроем приложение и запустим заново.

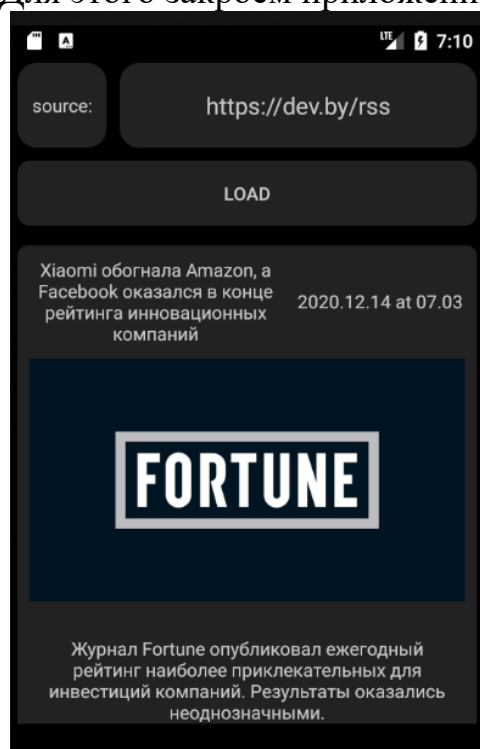


Рисунок 5.4 Приложение после перезапуска. Приложение сохранило и корректно отобразило данные.

Методика использования

Для пользования приложением нужно сначала его установить. Для этого нужно:

1. Скачать .apk файл на устройство Android (версия ОС ≥ 6)
2. Разрешить установку приложений не из Play Market
3. Открыть .apk файл и установить приложение

Далее нужно запустить приложение.

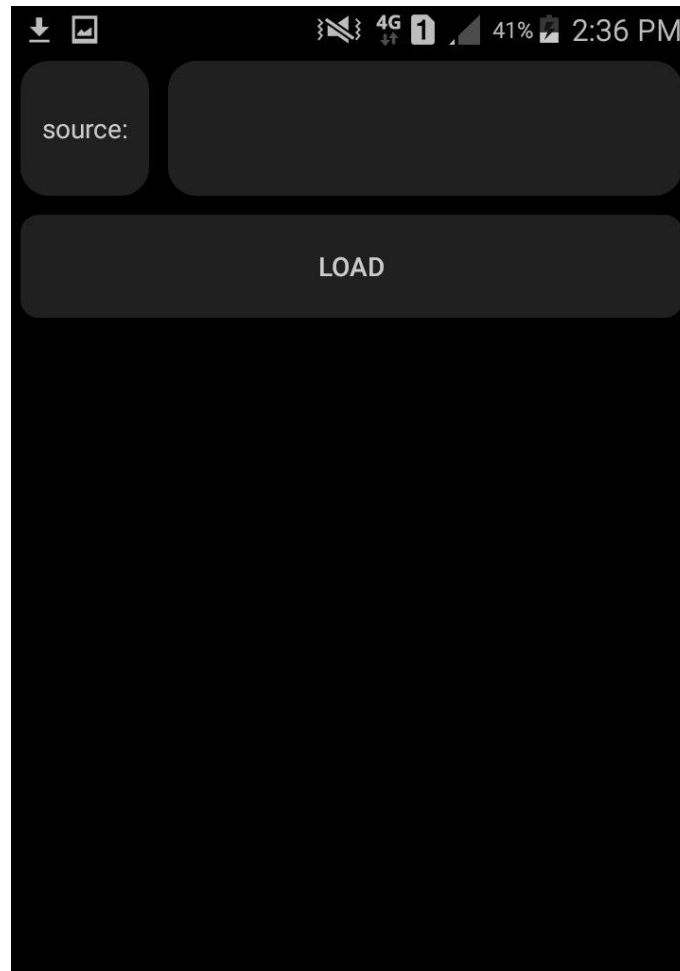


Рисунок 6.1 Экран приложения после установки

Затем требуется найти ссылку на rss ресурс и ввести в поле для ввода.

Например, можно взять dev.by/rss. Ссылки можно найти на сайтах множества интернет изданий, а также на сайтах блогов.

Обычно ссылки помечаются такой иконкой.



Рисунок 6.2 Иконка Rss

После ввода ссылки на ресурс требуется нажать на кнопку load и дождаться асинхронной загрузки.

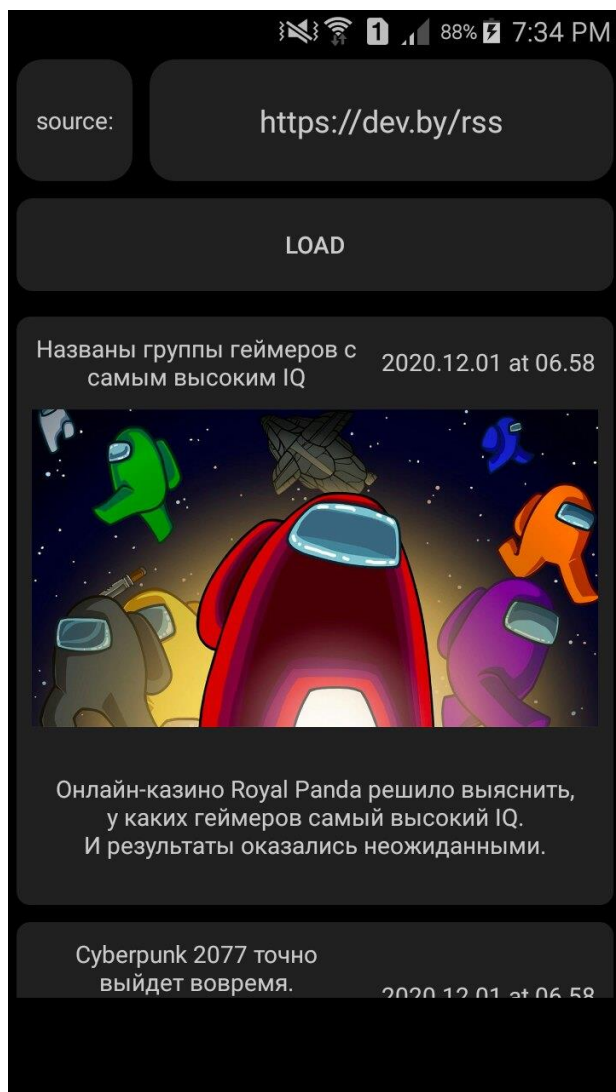


Рисунок 6.3 Экран приложения после загрузки новостей

Далее можно просмотреть список новостей и нажать на понравившуюся. Откроется новость в оригинале во встроенном браузере(WebView).



Рисунок 6.4 Встроенный браузер

Для возврата ко списку новостей требуется нажать кнопку 'Back'. Чтобы обновить список новостей, нужно нажать load еще раз.

Заключение

В рамках данной курсовой работы я показал работу мобильного Android-приложения со встроенной SQL базой данных. Приложение корректно работало и пользовалось системными ресурсами. Как можно было видеть, СУБД SQLite хорошо работала и взаимодействовала с приложением. Основные плюсы SQLite – простота использования в приложениях Android, отсутствие необходимости настройки и высокая надежность базы данных. Среди недостатков можно отметить довольно ограниченный набор поддерживаемых типов по сравнению с другими СУБД, а также отсутствие системы управления пользователями и отсутствие возможности увеличить производительность базы данных.

Список использованной литературы

1. Базы данных – Что это такое? [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://hostiq.ua/wiki/database/>);
2. What is a Relational Database? [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://aws.amazon.com/relational-database/>);
3. SQLite [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://ru.wikipedia.org/wiki/SQLite>);
4. Java (Programming language) [Электронный ресурс]. - Электронные данные. - Режим доступа: ([https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language)));
5. Android Documentation [Электронный ресурс]. - Электронные данные. - Режим доступа: (<https://developer.android.com/docs>);

Приложение

1. Ссылка на исходные файлы программы:
github.com/Valentin2508247/Course-work