Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «Северо-Восточный федеральный университет имени М.К. Аммосова»

Институт математики и информатики

Кафедра информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:  
«Мобильное приложение «Справочник студента» для платформы Android»

Выполнил: студент 4 курса

ИМИ группы ФИИТ-14

Анисимов Анатолий Юрьевич

Научный руководитель:

старший преподаватель каф. ИТ ИМИ

Леверьев Владимир Семенович

Оценка:

Дата:

Якутск, 2018

**Содержание**

[Введение 3](#_Toc515445170)

[Глава 1. Теоретическая 5](#_Toc515445171)

[1.1 Выбор ПО для разработки приложения 5](#_Toc515445172)

[1.2 Выбор языка программирования 7](#_Toc515445173)

[1.3 Django 9](#_Toc515445174)

[1.4 REST 10](#_Toc515445175)

[1.5 Django REST Framework 11](#_Toc515445176)

[1.6 Retrofit 2 12](#_Toc515445177)

[1.7 Требования к приложению 12](#_Toc515445178)

[Выводы по первой главе 13](#_Toc515445179)

[Глава 2. Разработка приложения 14](#_Toc515445180)

[2.1. Описание приложения 14](#_Toc515445181)

[Заключение 20](#_Toc515445182)

[Список использованной литературы 21](#_Toc515445183)

[Приложения 22](#_Toc515445184)

# Введение

Тема данной выпускной квалификационной работы: мобильное приложение «Справочник студента» для платформы Android.

В последние годы платформа Android обрела широкую популярность во всем мире, у всех от мала до велика, имеется по смартфону с этой операционной системой.

**Актуальность:** во время пар бывает необходимо сверяться со справочниками для уточнения нужной информации, например, на сайте yagu.s-vfu.ru, а переключаться с программы в браузер начинает надоедать. В этом случае удобно использовать сторонний девайс для просмотра справочника на нем. Смартфон для этого как раз и подходит.

Приложение разработано и оптимизировано специально для смартфона, отображается лучше, чем сайт Moodle отображается в браузере, где надо увеличивать масштаб и попадать на ссылки. Благодаря кэшированию может работать без подключения к интернету, но для этого обязательно должна быть выполнена предварительная загрузка данных минимум 1 раз.

*Объектом исследования* является разработка мобильного приложения «Справочник студента» и веб-сервера для хранения данных. *Предметом исследования* выступает изучение среды разработки на Android, сопутствующего программного обеспечения, изучение веб-разработки, и изучение предметной области справочника.

**Цель работы**: разработать приложение «Справочник студента», работающее с веб-сервером и предоставляющее полную или достаточную информацию по выбранной теме.

Данная цель будет достигаться путем решения следующих **задач**:

1. Проанализировать и изучить среду разработки для Android, веб-разработку, предметную область выбранной темы.
2. Разработать веб-сервер и приложение по выбранной теме.

Выпускная квалификационная работа содержит введение, две главы, заключение, список использованной литературы и приложения. В первой главе рассмотрен выбор программных средств разработки: Android Studio, язык Java, веб-фреймворк Django, язык Python. Во второй главе описан процесс разработки веб-сервера и приложения.

# Глава 1. Теоретическая

## 1.1 Выбор ПО для разработки приложения

Было рассмотрено следующее программное обеспечение: фреймворки PhoneGap от Adobe, Xamarin от Xamarin, React.Native от Facebook, NativeScript от Progress и IDE Android Studio от Google. Они делятся на кроссплатформенные (код написанный с использованием фреймворка переводится в нативный код для ОС, таким образом экономится время при разработке приложений, один и тот же код подходит для нескольких ОС) и нативные (в которых используются инструменты и оригинальные языки ОС).

К кроссплатформенным относятся:

**PhoneGap** – это бесплатный фреймворк для разработки мобильных приложений. Код пишется на JavaScript, и компилируется в готовые приложения для любой мобильной ОС. Необходимость в знаниях родных языков программирования каждой из мобильных ОС отпадает, но требуется знание JavaScript. Приложение запускает активность с WebView и посредством него вызывает машинные команды устройства.

**Xamarin** – это тоже бесплатный фреймворк для разработки приложений, но не только мобильных, но и для Windows и Mac OS. Используя язык, основанный на базе C#, разработчики могут использовать инструменты Xamarin для написания нативных Android, iOS и Windows приложения с нативным пользовательским интерфейсом и распространять код на нескольких ОС, включая Windows и macOS. В 2016 году куплен Microsoft и полностью интегрирован в Visual Studio.

**React.Native** – это фреймворк, который строит иерархию компонентов пользовательского интерфейса для построения JavaScript кода. Он имеет набор компонентов, для обеих платформ iOS и Android для создания мобильных приложений с оригинальным внешним видом и ощущением. React.Native фокусируется на интерфейсе пользователя, что делает приложения, созданные с использованием его, быстрозагружаемыми и плавными. Для программирования на нем необходимо знание HTML, CSS и JavaScript.

**NativeScript** – это фреймворк с открытым исходным кодом, основанный на использовании языков JavaScript и TypeScript. Позволяет создавать кроссплатформенные приложения с пользовательским интерфейсом, написанным на платформо-независимом языке XML, который с помощью этих абстракций вызывает нативные компоненты интерфейса определенной платформы. Обеспечивает прямой доступ к API платформы, не требует компилирования кода на платформенном языке, он транслируется как есть в WebView на устройстве и запускается на нативном уровне.

**LibGDX** – это бесплатный open source фреймворк разработки игр, написанный на Java c некоторыми С и С++ компонентами для производительного кода. Он позволяет разрабатывать десктопные и мобильные игры, используя одну и ту же основу кода. Кроссплатформенный, поддерживает Windows, Linux, Mac OS X, Android, iOS, Blackberry и веб-браузеры с поддержкой OpenGL. LibGDX дает разработчикам возможность писать, тестировать и отлаживать приложение на их ПК и использовать тот же код на Android. Он абстрагирует отличия между Windows, Linux приложениями и Android приложениями.

К нативным относятся:

**Android Studio** (созданная на базе IntelliJ IDEA Community Edition) в настоящее время является основной интегрированной средой, рекомендуемой для разработки приложений Android (исходные средства разработки Android работали на базе Eclipse IDE). Среда Android Studio в сочетании с бесплатным пакетом Android Software Development Kit (SDK) и бесплатным пакетом Java Development Kit (JDK) предоставляет все необходимое для создания, запуска и отладки приложений Android, поддержки их распространения (например, отправки в магазин Google Play) и т. д.

**Android NDK** – это набор инструментов, позволяющий включать код на нативных языках, таких как C и C++, и предоставляющий библиотеки платформ, которые помогают использовать нативные активности и получать доступ к компонентам устройства. Google не советует прибегать к NDK, если не требуется совершать сложных вычислений, но он может быть полезен для обеспечения быстроты и плавности работы приложения, либо повторном использовании библиотек, написанных на C или С++. При работе в Android Studio код, написанный на С или С++ компилируется в нативные библиотеки с помощью NDK, затем код Java вызывает функции библиотек NDK через Java Native Interface фреймворк

**Unreal engine** – это +Unity

Для разработки была выбрана среда Android Studio. Она была выбрана потому, что знакомство с этой средой уже было налажено, было написано несколько тестовых приложений, в общем, имелся опыт работы на ней. В других средах надо было бы изучить язык С# либо JavaScript. Если первый еще знаком, на парах изучен более-менее, то второй надо было бы изучать почти с нуля. По языку Java достаточные знания есть, и плюс он является нативным языком, значит программирование будет вестись так, как оно подразумевается разработчиками Androida. Еще отчасти потому, что было решено довериться лицензионному ПО от Google, кто, как не они, знают, как правильно работать со своим детищем, да и они вроде все что делают, делают хорошо. Среда основательная, дизайн уютный, весь нужный функционал прикручен: присутствуют понятные и своевременные подсказки, и, например, программа может сама находить устаревший вариант кода и преобразовывать его в актуальный, можно оформлять внешний вид, окна, формы, тут же начать их программировать на языке Java, и сразу построить и запустить получившееся приложение на создаваемом через менеджер виртуальных устройств виртуальном устройстве, либо на своем смартфоне, предварительно подключив его через USB кабель в режиме отладки.

## 1.2 Выбор языка программирования

В Android Studio поддерживаются 2 языка программирования: Java и Kotlin.

Kotlin – это статический типизированный язык программирования. Работающий на Java Virtual Machine и также может быть скомпилирован в код JavaScript. Разработан команда программистов JetBrains из Санкт-Петербурга. Хотя его синтакс не совместим с Java, JVM-ная интерпретация стандартной библиотеки Kotlin предназначена для взаимодействия и совместимости с Java. Kotlin использует автоматическое определение типа для определения типа значений и выражений, для которых тип был оставлен неустановленным. Это снижает его похожесть с Java, который часто использует полностью избыточные спецификации типов. Начиная с Android Studio 3.0 Kotlin полностью поддерживаемый Google’ом язык программирования на Android ОС, включен в пакет Android Studio IDE как альтернативна стандартному компилятору Java. Первая версия вышла в 2016-м году, т.е. этот язык еще совсем молод. В основе его лежат простота, малое количество кода, и полная совместимость с Java. Это сделано для безболезненного и постепенного перехода разработчиков с Java на Kotlin.

Java — сильно типизированный объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems (в последующем приобретённой компанией Oracle). Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, с помощью виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года.

Java — это программная платформа, включающая в себя мощную библиотеку, большой объем кода, пригодного для повторного использования, а также среду для выполнения программ, которая обеспечивает безопасность, независимость от операционной системы и автоматическую сборку "мусора".

Создатели Java составили официальное техническое описание, в котором объяснялись цели и достоинства нового языка. В этом документе приведено одиннадцать характерных особенностей Java. Этот язык:

* простой;
* объектно-ориентированный;
* распределенный;
* надежный;
* безопасный;
* не зависящий от архитектуры компьютера;
* переносимый;
* интерпретируемый;
* высокопроизводительный;
* многопоточный;
* динамичный.

Синтаксис Java, по существу, представляет собой упрощенный вариант синтаксиса C++. В этом языке отсутствует потребность в файлах заголовков, арифметике (и даже в синтаксисе) указателей, структурах, объединениях, перегрузке операций, виртуальных базовых классах и т.п.

Языком программирования был выбран Java, т.к. есть опыт работы на нем и на форумах и документации должно быть больше информации по возникающим вопросам.

## 1.3 Django

Согласно книге Владимира Дронова «Django: Практика создания веб-сайтов на Python», Django – это библиотека, созданная специально для разработки сайтов, ставшая самой популярной библиотекой такого рода из всех, написанных для использования с Python. Она реализует базовую функциональность веб-сайта и включает в себя все необходимое для программирования сайта, не занимаясь созданием всей инфраструктуры. Django содержит также средства для взаимодействия с базами данных, инструментами отладки, административный сайт, который можно использовать при работе с хранящимися в базе данными, и даже полнофункциональный веб-сервер, нужный для отладки сайта.

В возможности Django входит:

* реализация архитектуры «Model-View-Controller»;
* унифицированные средства для работы с БД любых поддерживаемых форматов;
* мощный шаблонизатор, основанный на специальных тегах, с возможностью наследования шаблонов;
* средства для работы с формами;
* инструменты для реализации разграничения доступа;
* встроенный административный сайт с возможностями настройки, который можно использовать для работы с данными;
* простая и ясная структура создаваемых сайтов: каждый раздел сайта представляет собой отдельное приложение, которое можно использовать в другом сайте, а их совокупность – сам сайт.

## 1.4 REST

REST - (сокр. от англ. REpresentational State Transfer - «передача состояния представления») – это стиль архитектурного взаимодействия элементов распределенного приложения в сети. Основан на использовании ресурсов, а не действий посредством GET или POST запросов (Rest-запрос). Принимает запрос и посылает представление состояния ресурса в формате JSON или XML.

Требует соблюдения шести ограничений, направленных на общее улучшение работоспособности системы:

1. Унифицированный интерфейс.

Это основное требование, описывает интерфейсы клиента и сервера. Подразумевает использование HTTP запросов (GET, PUT, POST, DELETE), унифицированный идентификатор ресурса и прием ответов HTTP(status, body).

1. Отсутствие состояния.

Сервер не должен иметь информации о состоянии клиента, каждое сообщение должно быть самоописываемым и иметь достаточную информацию для ее обработки на сервере. Каждое состояние должно хранится на клиенте.

1. «Клиент-серверная» модель.

Клиент должен быть отделен от сервера, не иметь прямой доступ к ресурсам сервера, и наоборот. Мостом, связующим их, служит унифицированный интерфейс, использующий HTTP запросы.

1. Возможность кэширования.

Все, что приходит с сервера, должно быть кэшируемым, явно, неявно, либо по договоренности

1. Расслоенная система.

Клиент не должен знать, напрямую он обращается к серверу, либо к какому-то промежуточным элементом. Например, клиент не должен знать, принимает он данные с сервера, или с кэша. Это улучшает масштабируемость системы, обеспечивает многоуровневость.

1. Код по требованию.

Уникальное требование, означает возможность расширения клиента, путем отправки некоторой логики сервера на клиент как сценарий или апплет. Это единственное выборочное требование.

Соблюдение REST требований придает системе следующие свойства:

* масштабируемость;
* простоту;
* модифицируемость;
* прозрачность;
* мобильность компонентов;
* надежность.

## 1.5 Django REST Framework

Django REST Framework – это мощный и гибкий фреймворк, использующий подход REST (Representational State Transfer, передача состояния представления), для построения веб-API (от англ. application programming interface - интерфейс прикладного программирования), т.е. API для сервера.

Имеет возможности:

* просматриваемый в веб браузере API, что очень полезно для разработчиков;
* политика аутентификации, включающая пакеты OAuth1 и OAuth2;
* поддержка сериализации ORM и не-ORM источников данных;
* полностю настраиваемый – просто используйте обычные views, основанные на функциях, если не нуждаетесь в более мощных фичах;
* исчерпывающая документация и отличное сообщество, которое поможет вам советом;
* используется и доверен такими известными международными компаниями, как Mozilla, Red Hat и Eventbrite.

## 1.6 Retrofit 2

Документация описания Retrofit 2 крайне скупа, она гласит, что Retrofit 2 – это типобезопасный HTTP клиент для Android и Java, разработанный компанией Square. В связке с библиотекой Okhttp, предоставляющей возможность кэширования ответов сервера и управления header’ами, тоже, кстати, их детищем, - это необходимое приспособление для написания клиентского API для приложения, сообщающегося с сервером. Следует заветам REST архитектуры, поэтому его можно назвать RESTful API.

## 1.7 Требования к приложению

Требования к приложению, учитывая специфику, следующие:

* быстрота;
* простота;
* получать данные с сервера и управлять принятыми данными;
* кэшировать полученные данные;
* показывать список дисциплин;
* показывать список лекций;
* четко и разборчиво отображать текст лекции;

## Выводы по первой главе

Глава 2. Разработка приложения

## 2.1. Описание приложения

Темой справочника был выбран предмет из учебной программы первого курса, Основы программирования, поэтому приложение называется «Основы программирования». Оно основано на теоретической базе лекций Эверстова Владимира Васильевича с сайта «yagu.s-vfu.ru», так же известном как «мудл». Приложение выводит список имеющихся лекций по предмету Основы программирования, при нажатии на кнопку «Перейти на мудл» открывает на смартфоне браузер и переходит на мудл. При нажатии на элемент списка открывает новую активность с картинкой С++ и полным текстом лекции. По клавишам «Пред. лекция» и «След. лекция» можно перейти к предыдущей или следующей лекции из списка. При нажатии на клавишу «Перейти на мудл» открывает браузер и переходит по ссылке мудла, и, по идее, должен был открывать страницу с текущей открытой в приложении лекцией, но почему-то переходит на страницу предмета, наверное, сайт так устроен. В целях удобности и быстроты было решено не делать ненужных функций, нагромождений на экране. В итоге приложение вышло понятным, удобным, быстрым насколько возможно, с простором для дальнейших улучшений.

Разработка веб-сервера

Структура базы данных

Структура приложения

Модель классов

Разработка приложения Android

Программа состоит из трех активностей и их классов, класса СУБД. Первая активность - общий список статей, сделана с помощью компонентов ListView, кнопки Button и TextView. Кнопка «Перейти на мудл» передает число 1000 третьей активности и запускает ее. Число 1000 нужно для того, чтобы активность поняла, что лекция еще не была выбрана и следует перейти на страницу курса. Код которой будет приведен в приложении (приложение 1).

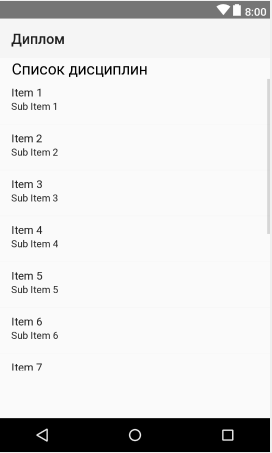


Рис. 1 Главная активность - общий список статей.

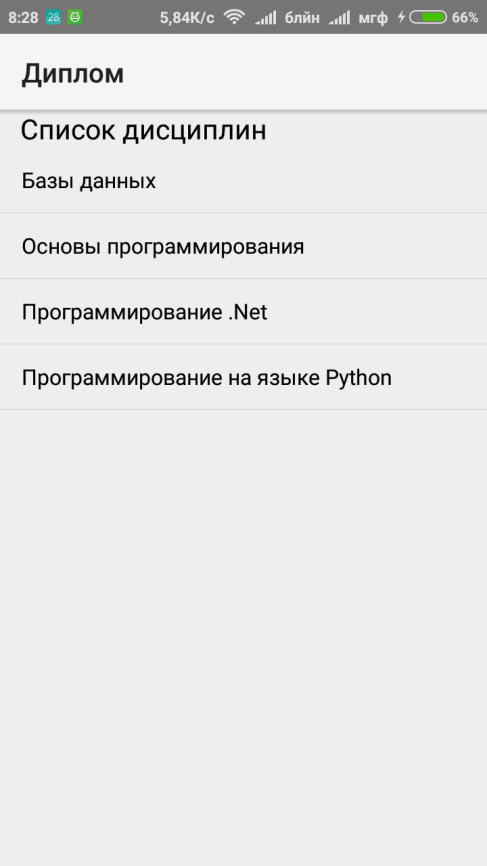


Рис. 2 Главная активность – общий список статей, пример.

Код класса первой активности будет представлен в приложении (приложение 2).

Переход из первой активности ко второй происходит при нажатии по элементу из списка. Когда программа регистрирует клик, то через встроенный класс intent передается информация о позиции нажатого элемента в списке, которая принимается второй активностью.

Вторая статья сделана с помощью компонентов WebView и 2 кнопок Button. Она открывает в себе загруженные данные описания одной статьи. Т.к. данные представлены в текстовом виде в txt-файле, для красоты было принято использовать WebView. Весь текст может быть размечен HTML тэгами, имеется поддержка картинок, которая нужна для отображения таблиц и строк кода в первичном знакомом виде, которые заскриниваются и, по надобности, делятся. Из первой активности она принимает номер выбранного элемента из списка и, рассматривая его как ID нужной нам статьи, находит из базы данных при помощи класса DatabaseHelper описание выбранной в первой активности статьи. Нажимая на кнопки «Пред. лекция» и «След. лекция», можно перейти к предыдущей или следующей лекции. Код второй активности, написанный на xml, будет представлен в приложении (приложение 3). Код класса второй активности, написанный на Java, будет представлен в приложении (приложение 4).

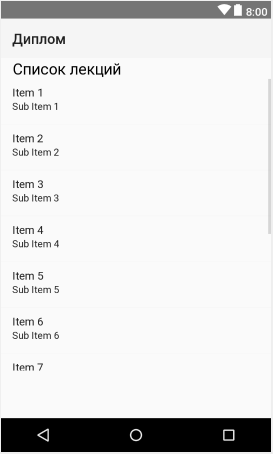


Рис. 3 Вторая активность – описание статей.

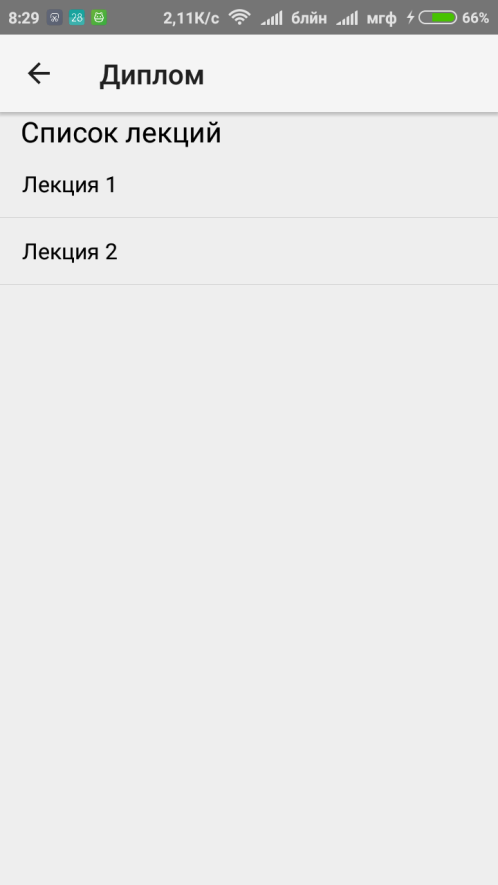


Рис. 4 Вторая активность – описание статей, пример

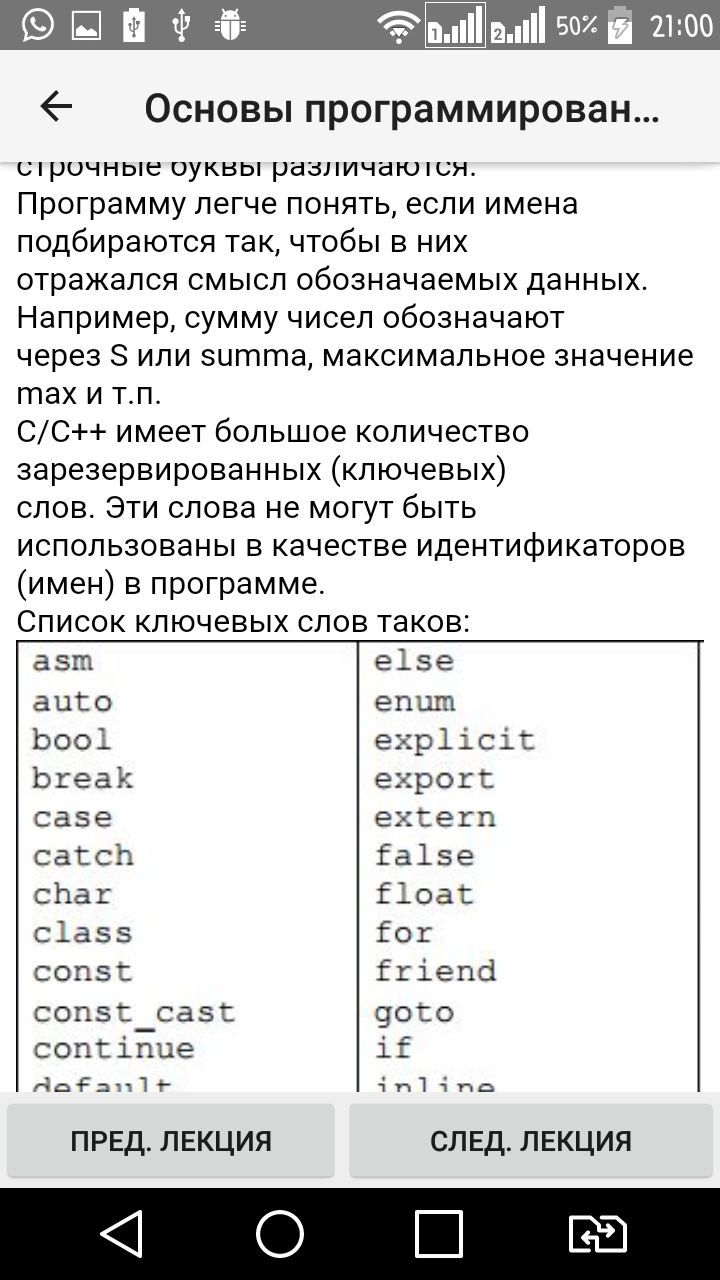


Рис. 5 Вторая активность – описание статей, пример 2.

Третья активность – предназначена для перехода по нажатию кнопки «Перейти на мудл» открывать браузер и переходить на страницу либо курса, либо выбранной лекции. Он реализован с помощью компоненты WebView. Код класса активности мудла будет приведен в приложении (приложение 6). Код этой активности, написанный на xml, короткий, поэтому привожу код здесь:

<**WebView xmlns:android="http://schemas.android.com/apk/res/android"  
 android:id="@+id/webview"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** />

Класс DatabaseHelper использует возможности встроенной в Android Studio СУБД SQLite. Код класса DatabaseHelper, написанный на Java, будет представлен в приложении (приложение 5).

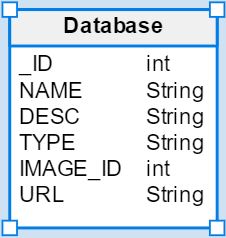


Рис. 6 Модель базы данных.

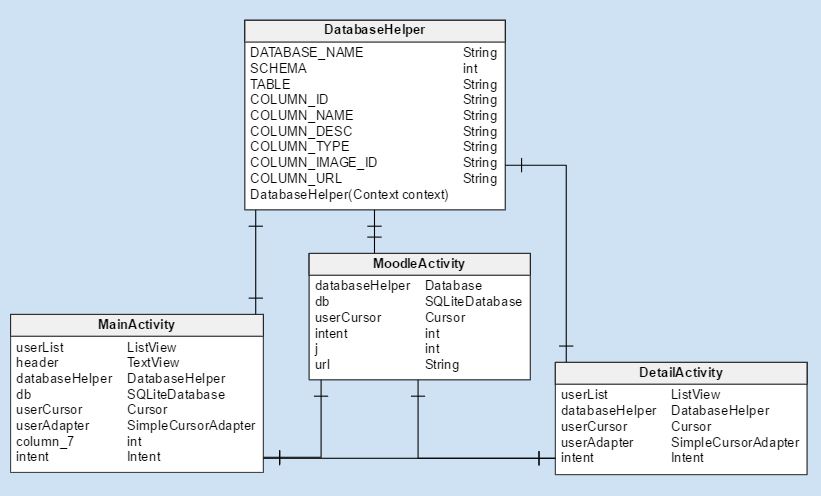


Рис. 7 Диаграмма классов

В дальнейшем планируется создание главной формы, откуда можно перейти в другие формы, создание активностей для новых функций, реализация вставки картинок. Из функциональных возможностей в планах общий список, сортировка по критериям, разбиение по видам, картинки. Воплощен в коде пока только общий список и статья.

Заключение

Первая часть данной работы посвящена обзору технологий разработки мобильных приложений, а также методик создания серверной части.

Вторая часть работы является практической и описывает процесс создания мобильного приложения и серверной части.

В ходе работы создано мобильное приложение, позволяющее студенту просматривать справочные, учебные и иные материалы по изучаемым дисциплинам. Отличительной особенностью является автоматическое кеширование передаваемых данных, что позволяет использовать приложение в режиме офлайн.

Таким образом, поставленные задачи были решены, но работа еще не окончена.

В работе представлено описание черновой версии программы приложение-справочник «Основы программирования» для Android’a в процессе разработки.

На самом деле писать приложение оказалось делом не простым, несмотря на то, что Android Studio понятный в использовании, и он максимально пытается облегчить жизнь программисту. Язык Java, благодаря предмету ООП и сходству с языком С#, более-менее нормально используется.

Справочники на платформе Android имеют будущее, но, на мой взгляд, они не будут так популярны, как интернет сайты с тем же внутренним составом, что и у них.

Список использованной литературы

1. Android Studio // Википедия. [2017—2017]. Дата обновления: 18.05.2017. URL: http://ru.wikipedia.org/?oldid=85473651 (дата обращения: 18.05.2017).
2. Герберт Шилдт – Java 8. Полное руководство, 9-е издание = Java 8. The Complete Reference, 9th Edition. — М.: «Вильямс», 2015. — 1376 с. — ISBN 978-5-8459-1918-2.
3. Гриффитс Дон, Гриффитс Дэвид – Head First. Программирование для Android. — СПб.: Питер, 2016. — 704 с.: ил. — (Серия «Head First O’Reilly»).
4. Дейтел П., Дейтел Х., Уолд А. – Android для разработчиков. 3-е изд. — СПб.: Питер, 2016. — 512 с.: ил. — (Серия «Библиотека программиста»)
5. Хорстманн, Кей С. – Java. Библиотека профессионала, том 1. Основы. 10-е изд.: Пер. с англ. — М .: ООО "И.Д. Вильямс", 2016. — 864 с .: ил. — Парал. тит. англ.
6. Дронов В. – Django: Практика создания Web-сайтов на Python. – СПб.: БХВ-Петербург, 2016. – 528 с.: ил. – (Профессиональное программирование)

Приложения

Приложение 1 – код главной активности, написанный на xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** >  
 <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:weightSum="1"**>  
  
 <**TextView  
 android:id="@+id/header"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text=" Список лекций"  
 android:textSize="10pt"  
 android:textColor="#000000"**/>  
  
 <**ListView  
 android:id="@+id/list"  
 android:layout\_width="match\_parent"  
 android:layout\_height="389dp"  
 android:layout\_weight="0.86"** />  
  
 <**Button  
 android:id="@+id/moodle\_button"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="10dp"  
 android:onClick="onClickMoodle\_button1"  
 android:text="Перейти на мудл"** />  
  
 </**LinearLayout**>  
</**RelativeLayout**>

Приложение 2 – код класса первой активности, написанный на Java

**package** com.example.anisi;  
  
**import** android.content.Intent;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.view.View;  
**import** android.widget.AdapterView;  
**import** android.widget.ListView;  
**import** android.widget.SimpleCursorAdapter;  
**import** android.widget.TextView;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 ListView **userList**;  
 DatabaseHelper **databaseHelper**;  
 SQLiteDatabase **db**;  
 Cursor **userCursor**;  
 SimpleCursorAdapter **userAdapter**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 **userList** = (ListView)findViewById(R.id.***list***);  
 **databaseHelper** = **new** DatabaseHelper(getApplicationContext());  
 **userList**.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 **public void** onItemClick(AdapterView<?> a, View v, **int** position, **long** id) {  
 Intent intent = **new** Intent();  
 intent.setClass(MainActivity.**this**, DetailsActivity.**class**);  
  
 intent.putExtra(**"id"**, position);  
 *//запускаем вторую активность* startActivity(intent);  
 }  
 });  
 }  
 @Override  
 **public void** onResume() {  
 **super**.onResume();  
 *// открываем подключение* **db** = **databaseHelper**.getReadableDatabase();  
  
 *//получаем данные из бд в виде курсора* **userCursor** = **db**.rawQuery(**"select \* from "**+ DatabaseHelper.***TABLE***, **null**);  
 *// определяем, какие столбцы из курсора будут выводиться в ListView* String[] headers = **new** String[] {DatabaseHelper.***COLUMN\_NAME***};  
 *// создаем адаптер, передаем в него курсор* **userAdapter** = **new** SimpleCursorAdapter(**this**, android.R.layout.***simple\_list\_item\_1***,  
 **userCursor**, headers, **new int**[]{android.R.id.***text1***, android.R.id.***text2***}, 0);  
 **userList**.setAdapter(**userAdapter**);  
 }  
  
 @Override  
 **public void** onDestroy(){  
 **super**.onDestroy();  
 *// Закрываем подключение и курсор* **db**.close();  
 **userCursor**.close();  
 }  
  
 **public void** onClickMoodle\_button1(View view) {  
 Intent intent = **new** Intent();  
 intent.setClass(MainActivity.**this**, MoodleActivity.**class**);  
 intent.putExtra(**"id"**, 1000);  
 startActivity(intent);  
 }  
}

Приложение 3 - код второй активности, написанный на xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"** >  
 <**WebView  
 android:id="@+id/webView"  
 android:layout\_width="match\_parent"  
 android:layout\_height="478dp"  
 android:layout\_above="@+id/next\_button"**>  
  
 </**WebView**>  
 <**Button  
 android:id="@+id/prev\_button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="10dp"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentLeft="true"  
 android:layout\_toLeftOf="@+id/next\_button"  
 android:onClick="onClickPrev\_button"  
 android:text="Пред. лекция"** />  
  
 <**Button  
 android:id="@+id/next\_button"  
 android:layout\_width="189dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_x="260dp"  
 android:layout\_y="469dp"  
 android:onClick="onClickNext\_button"  
 android:text="След. лекция"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentRight="true"  
 android:layout\_alignParentEnd="true"** />  
  
</**RelativeLayout**>

Приложение 4 - код класса второй активности, написанный на Java

**package** com.example.anisi;  
  
**import** java.io.BufferedReader;  
**import** java.io.IOException;  
**import** java.io.InputStream;  
**import** java.io.InputStreamReader;  
  
**import** android.app.AlertDialog;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.view.View;  
**import** android.util.Log;  
**import** android.webkit.WebView;  
**import** android.widget.AdapterView;  
**import** android.widget.ImageView;  
**import** android.widget.ListView;  
**import** android.widget.SimpleCursorAdapter;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**public class** DetailsActivity **extends** AppCompatActivity {  
 Context **context**;  
 DatabaseHelper **databaseHelper**;  
 SQLiteDatabase **db**;  
 Cursor **userCursor**;  
 **int k**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_details***);  
 getSupportActionBar().setDisplayHomeAsUpEnabled(**true**);  
 WebView webView = (WebView) findViewById(R.id.***webView***);  
 **databaseHelper** = **new** DatabaseHelper(getApplicationContext());  
 **db** = **databaseHelper**.getReadableDatabase();  
 Intent intent = getIntent();  
 **k** = intent.getIntExtra(**"id"**, 0);  
 **context** = getBaseContext();  
 **userCursor** = **db**.query(**"REFERENCE\_BOOK"**,  
 **new** String[] {**"DESC"**},  
 **"\_id = ?"**, **new** String[] {Integer.*toString*(**k**+1)},  
 **null**, **null**, **null**);  
 String resName = **""**;  
 **if**(**userCursor**.moveToFirst()) {  
 resName = **userCursor**.getString(0);  
 String text = *readRawTextFile*(**context**, getResources().getIdentifier(resName, **"raw"**, **"com.example.anisi.metanit"**));  
 webView.loadDataWithBaseURL(**null**, text, **"text/html"**, **"en\_US"**, **null**);  
 }  
 }  
  
 **public static** String readRawTextFile(Context context, **int** resId) {  
 InputStream inputStream = context.getResources().openRawResource(resId);  
  
 InputStreamReader inputReader = **new** InputStreamReader(inputStream);  
 BufferedReader buffReader = **new** BufferedReader(inputReader);  
 String line;  
 StringBuilder builder = **new** StringBuilder();  
  
 **try** {  
 **while** ((line = buffReader.readLine()) != **null**) {  
 builder.append(line);  
 builder.append(**"<br />"**);  
 }  
 } **catch** (IOException e) {  
 **return null**;  
 }  
 **return** builder.toString();  
 }  
  
 **public void** onClickPrev\_button(View view) {  
 **if**(**k** < 1){  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(DetailsActivity.**this**);  
 builder.setTitle(**"Это первая лекция"**)  
 .setIcon(R.drawable.***n4***)  
 .setCancelable(**false**)  
 .setNegativeButton(**"ОК"**,  
 **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 dialog.cancel();  
 }  
 });  
 AlertDialog alert = builder.create();  
 alert.show();  
 *//this.finish();* }  
 **else** {  
 Intent intent = **new** Intent();  
 intent.setClass(DetailsActivity.**this**, DetailsActivity.**class**);  
 intent.putExtra(**"id"**, **k** - 1);  
 startActivity(intent);  
 **this**.finish();  
 }  
 }  
  
 **public void** onClickNext\_button(View view) {  
 Intent intent = **new** Intent();  
 **if**(**k**>14){  
 AlertDialog.Builder builder = **new** AlertDialog.Builder(DetailsActivity.**this**);  
 builder.setTitle(**"Это последняя лекция"**)  
 .setIcon(R.drawable.***n4***)  
 .setCancelable(**false**)  
 .setNegativeButton(**"ОК"**,  
 **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
 dialog.cancel();  
 }  
 });  
 AlertDialog alert = builder.create();  
 alert.show();  
 *//this.finish();* }  
 **else** {  
 intent.setClass(DetailsActivity.**this**, DetailsActivity.**class**);  
 intent.putExtra(**"id"**, **k** + 1);  
 startActivity(intent);  
 **this**.finish();  
 }  
 }  
}

Приложение 5 - код класса активности СУБД, написанный на Java

**package** com.example.anisi;  
  
**import** android.database.sqlite.SQLiteOpenHelper;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.content.Context;  
**import** android.content.ContentValues;  
  
**public class** DatabaseHelper **extends** SQLiteOpenHelper {  
 **private static final** String ***DATABASE\_NAME*** = **"reference\_book.db"**; *// название бд* **private static final int *SCHEMA*** = 27; *// версия базы данных* **static final** String ***TABLE*** = **"REFERENCE\_BOOK"**; *// название таблицы в бд  
 // названия столбцов* **public static final** String ***COLUMN\_ID*** = **"\_id"**;  
 **public static final** String ***COLUMN\_NAME*** = **"name"**;  
 **public static final** String ***COLUMN\_DESC*** = **"desc"**;  
 **public static final** String ***COLUMN\_TYPE*** = **"type"**;  
 **public static final** String ***COLUMN\_IMAGE\_ID*** = **"image\_id"**;  
 **public static final** String ***COLUMN\_URL*** = **"url"**;  
  
 **public** DatabaseHelper(Context context) {  
 **super**(context, ***DATABASE\_NAME***, **null**, ***SCHEMA***);  
 }  
  
 @Override  
 **public void** onCreate(SQLiteDatabase db) {  
  
 db.execSQL(**"CREATE TABLE REFERENCE\_BOOK ("** + ***COLUMN\_ID*** + **" INTEGER PRIMARY KEY AUTOINCREMENT,"** + ***COLUMN\_NAME*** + **" TEXT, "** + ***COLUMN\_DESC*** + **" TEXT,"** + ***COLUMN\_TYPE*** + **" TEXT, "** + ***COLUMN\_IMAGE\_ID*** + **" INTEGER, "** + ***COLUMN\_URL*** + **" TEXT);"**);  
  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 1. Построение программ', 'lect1.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37007/');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 2. Базовые типы данных', 'lect2.txt', 'Лекция', "** + R.drawable.***n4*** + **",'http://yagu.s-vfu.ru/mod/resource/view.php?id=37009');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 3. Программы линейной структуры', 'lect3.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37011');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 4-5. Ветвления', 'lect45.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37014');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 6-7. Циклы', 'lect67.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37017');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 8-12. Одномерные массивы', 'lect812.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37024');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 13-15. Двумерные массивы', 'lect1315.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37030');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Лекция 16-17. Строки', 'lect1617.txt', 'Лекция', "** + R.drawable.***n4*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 db.execSQL(**"INSERT INTO "**+ ***TABLE*** +**" ("** + ***COLUMN\_NAME*** + **", "** + ***COLUMN\_DESC*** + **", "** + ***COLUMN\_TYPE*** + **", "** + ***COLUMN\_IMAGE\_ID*** + **", "** + ***COLUMN\_URL*** + **") VALUES ('Пустой элемент для скролла', 'prey.txt', 'Игра', "** + R.drawable.***n1*** + **", 'http://yagu.s-vfu.ru/mod/resource/view.php?id=37037');"**);  
 }  
 @Override  
 **public void** onUpgrade(SQLiteDatabase db, **int** oldVersion, **int** newVersion) {  
 db.execSQL(**"DROP TABLE IF EXISTS "**+***TABLE***);  
 onCreate(db);  
 }  
}

Приложение 6 – код класса третьей активности, написанный на Java

**package** com.example.anisi;  
  
**import** android.database.Cursor;  
**import** android.database.sqlite.SQLiteDatabase;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.webkit.WebView;  
**import** android.content.Intent;  
  
**public class** MoodleActivity **extends** AppCompatActivity {  
 DatabaseHelper **databaseHelper**;  
 SQLiteDatabase **db**;  
 Cursor **userCursor**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **databaseHelper** = **new** DatabaseHelper(getApplicationContext());  
 **db** = **databaseHelper**.getReadableDatabase();  
  
 Intent intent = **new** Intent();  
 **int** j = intent.getIntExtra(**"id"**,0);  
 **if** (j > 999) {  
 String url = **"http://yagu.s-vfu.ru/course/view.php?id=5124"**;  
 setContentView(R.layout.***activity\_moodle***);  
 WebView webView = (WebView) findViewById(R.id.***webview***);  
 webView.loadUrl(url);  
 **this**.finish();  
 }  
 **else**{  
 **userCursor** = **db**.query(**"REFERENCE\_BOOK"**,  
 **new** String[]{**"URL"**},  
 **"\_id = ?"**, **new** String[]{Integer.*toString*(j + 1)},  
 **null**, **null**, **null**);  
 **if** (**userCursor**.moveToFirst()) {  
 String url = **userCursor**.getString(0);  
 setContentView(R.layout.***activity\_moodle***);  
 WebView webView = (WebView) findViewById(R.id.***webview***);  
 webView.loadUrl(url);  
 **this**.finish();  
 }  
 }  
 }  
}