

Министерство образования и науки Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра дискретной математики и
информационных технологий

**Разработка мобильного приложения для размещения онлайн-объявлений
о товарах и услугах от частных лиц для ОС Android**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

Студента 4 курса 421 группы

направления подготовки 09.03.01 — Информатика и вычислительная техника
факультета компьютерных наук и информационных технологий

Муковозова Андрея Андреевича

Научный руководитель

ассистент

М. В. Белоконь

Зав. кафедрой

к.ф.- м.н., доцент

Л.Б. Тяпаев

Саратов 2018 год

ВВЕДЕНИЕ

Android представляет собой операционную систему, предназначенную для планшетов, смартфонов, наручных часов, телевизоров и многих других устройств и основанную на ядре Linux и собственной реализации виртуальной машины Java от Google. Историю разработки системы Android связывают прежде всего с компанией Android Inc., которая позднее была куплена корпорацией Google. Впоследствии был создан бизнес-альянс Open Handset Alliance (ОНА), куда вошло до 84 компаний. Одним из инициаторов создания этого альянса стала корпорация Google. На данный момент именно она осуществляет всю необходимую деятельность для поддержки и дальнейшего развития платформы [1].

Популярность системы Android обеспечивается множеством факторов. Android является операционной системой с открытым исходным кодом, что позволяет другим производителям модифицировать ее и выпускать девайсы с уникальными возможностями. Так, каждый человек может выбрать конфигурацию системы, которая будет соответствовать всем необходимым требованиям для максимально комфортного и плодотворного использования смартфона. Также Android является кастомизируемой системой [3]. Она дает возможность полностью изменить интерфейс с помощью сторонних приложений. Для данной работы наиболее важна возможность создавать Java/Kotlin-приложения, управляющие устройством через разработанные Google библиотеки, которую также предоставляет система Android [2].

В современном мире каждый пользуется смартфоном. Большое количество людей пользуется мобильными устройствами с ОС Android. Особую популярность набирают приложения с возможностью реализации покупок и обмена различных товаров. К ним относятся: AliExpress, Avito, Joom.

Сегодняшняя экономика всерьез осваивает способы реализации товаров с помощью криптовалют. Однако ни одно из рассмотренных приложений не позволяет осуществлять покупку и обмен товара с использованием цифровой валюты, базирующейся на криптографических методах.

В дипломной работе разработано мобильное приложение для ОС Android для размещения объявлений о товарах и услугах от частных лиц. Но в данном приложении пользователю будет предоставлена возможность использовать различные виды криптовалют для оценки товаров.

Феномен криптовалют привлекает внимание исследователей разных сфер последние 5 лет. Криптовалюта — это цифровая валюта, в которой методы шифрования используются для регулирования генерации единиц валюты и проверки перевода средств, действующих независимо от центрального банка. Ее важной особенностью является то, что создание и контроль основывается на криптографических методах, а также то, что учёт криптовалют децентрализован. Как правило, функционирование систем основано на таких технологиях как блокчейн, направленный ациклический граф, консенсусный реестр (ledger) и др. Инновационность этой технологии также связана с тем, что информация о транзакциях обычно не шифруется и доступна в открытом виде. Для обеспечения неизменности базы цепочки блоков транзакций используются элементы криптографии (цифровая подпись на основе системы с открытым ключом, последовательное хеширование) [4].

Теоретической основой работы являются электронные ресурсы с документацией к используемым библиотекам. А также официальный сайт для разработчиков Android, на котором опубликованы уроки по базовым сценариям в мобильных приложениях. Кроме того, опубликована полная документация для Android SDK.

Актуальность работы заключается в том, что мобильных приложений, использующих криптовалюту, как основную для продажи товаров, в Play Маркет не нашлось. Так как тема криптовалют сейчас актуальна, было решено реализовать данное мобильное приложение для самой популярной ОС в мире.

Целью нашей работы является разработка мобильного приложения для размещения объявлений онлайн и продажи личных вещей как с помощью

обычной так и цифровой валюты. Для достижения цели были поставлены следующие **задачи**:

1. Изучить теоретический материал, связанный с разработкой мобильных приложений под ОС Android.
2. Найти и изучить инструменты, необходимые для разработки приложения.
3. Спроектировать логику работы приложения.
4. Спроектировать дизайн приложения.
5. Разработать и реализовать мобильное приложения для публикации онлайн-объявлений и возможностью продажи личных вещей.

Цели и задачи определили **структуру** работы, которая состоит из введения, двух разделов, заключения и списка использованных источников, включающего 43 страницы.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

1 «Описание использованных технологий»

Раздел 1 включает в себя обзор всех основных библиотек, архитектурных подходов, задействованных при разработке мобильного приложения.

Библиотеки:

1. Retrofit
2. RxJava

Архитектурный подход, используемый в реализации мобильного приложения, называется Clean Architecture.

1.1 Retrofit

Эта библиотека используется для облегчения клиент-серверного взаимодействия. Она позволяет с легкостью конфигурировать запросы. Есть возможность добавлять Query параметры, Part-параметры. Также есть возможность отправлять в теле запроса объекты (Body), а также Multipart/form-data для загрузки файлов на

сервер. Для, например, авторизации некоторых запросов есть возможность добавлять параметры в header, такие как token. Для использования библиотеки необходимо сконфигурировать клиент Retrofit. В конфигурации необходимо задать базовый URL, к которому будет обращаться библиотека при выполнении запросов, также нужно указать средства для парсинга данных из JSON-объекта, который приходит с сервера в объект нужного нам типа для дальнейшего использования в приложении [5].

1.2 RxJava

RxJava — "Реактивное расширение" для JVM — библиотека для составления асинхронных и основанных на событиях программ с использованием наблюдаемых последовательностей для виртуальной машины Java [9].

RxJava обладает следующими преимуществами:

1. Интуитивность (Действия в Rx описываются в таком же стиле, как и в других библиотеках, вдохновленных функциональным программированием, например, Java Streams. Rx дает возможность использовать функциональные трансформации над потоками событий).
2. Расширяемость (RxJava может быть расширена пользовательскими операторами. И хотя Java не позволяет сделать это элегантно, RxJava предлагает всю расширяемость доступную в реализациях Rx на любом другом языке).
3. Декларативность (Функциональные трансформации объявлены декларативно, то есть не содержат переменных).
4. Компонуемость (Операторы в Rx легко комбинируются, чтобы проводить сложные операции).
5. Преобразуемость (Операторы в Rx могут трансформировать типы данных, фильтруя, обрабатывая и расширяя потоки данных при необходимости) [12].

1.2.1 Ключевые типы

Rx базируется на двух фундаментальных типах, в то время как некоторые другие расширяют их функциональность. Этими базовыми типами являются Observable и Observer, которые мы и рассмотрим в этом разделе. Мы также рассмотрим Subject'ы – они помогут в понимании основных концепций Rx [11].

Rx построена на паттерне Observer. В этом нет ничего нового, обработчики событий уже существуют в Java (например, JavaFX EventHandler), однако они проигрывают в сравнении с Rx по следующим причинам:

1. Обработку событий в них сложно компоновать.
2. Их вызов нельзя отложить.
3. Могут привести к утечке памяти.
4. Не существует простого способа сообщить об окончании потока событий.
5. Требуют ручного управления многопоточностью [15].

1.3 Архитектура приложения. Clean Architecture

1.3.1 Введение

Clean Achitecture — способ построения архитектуры приложения, предложенный Робертом Мартином (который также известен как дядюшка Боб --- Uncle Bob) в 2012 году [20].

Clean Architecture включает в себя два основных принципа:

1. Разделение на слои.
2. Инверсия зависимостей.

1.3.1.1 Разделение зависимостей

Суть принципа заключается в разделении всего кода приложения на слои. Всего имеются три слоя:

1. Слой отображения.
2. Слой бизнес-логики.

3. Слой работы с данными.

Самым главным слоем является слой бизнес-логики. Особенность данного слоя заключается в том, что он не зависит ни от каких внешних библиотек или фреймворков. Это достигается за счет инверсии зависимостей [17].

1.3.1.2 Инверсия зависимостей

Согласно данному принципу, слой бизнес-логики не должен зависеть от внешних. То есть классы из внешних слоев не должны использоваться в классах бизнес-логики. Взаимодействие с внешними слоями происходит через интерфейсы, которые реализуют классы внешних слоев.

Благодаря разделению ответственности между классами легко можно изменять код приложения, а также добавлять новый функционал, затрагивая при этом минимальное количество классов [18]. Помимо этого, получается легко тестируемый код. Стоит заметить, что построение правильной архитектуры целиком и полностью зависит от самого разработчика и его опыта [20].

Преимущества чистой архитектуры:

1. Независимость от UI, БД и фреймворков.
2. Позволяет быстрее добавлять новые функции.
3. Более высокий процент покрытия кода тестами.
4. Повышенная простота навигации по структуре пакетов.

Недостатки чистой архитектуры:

1. Большое количество классов.
2. Довольно высокий порог вхождения.

1.3.2 Слои и инверсия зависимостей

Архитектуру приложения, построенную по принципу Clean Architecture можно разделить на три слоя:

1. Слой отображения (presentation).

2. Слой бизнес-логики (domain).
3. Слой работы с данными (data).

1.3.3 Слой бизнес-логики (Domain)

Бизнес-логика — это правила, описывающие, как работает бизнес (например, пользователь не может совершить покупку на сумму больше, чем есть на его счёте). Бизнес-логика не зависит от реализации базы данных или интерфейса пользователя. Бизнес-логика меняется только тогда, когда меняются требования бизнеса, и не зависит от используемой СУБД или интерфейса пользователя [19].

1.3.4 Слой работы с данными (Data)

В данном слое содержится всё, что связано с хранением данных и управлением ими. Это может быть работа с базой данных, SharedPreferences, сетью или файловой системой, а также логика кеширования, если она имеется.

1.3.5 Слой отображения (Presentation)

Слой представления содержит все компоненты, которые связаны с UI, такие как View-элементы, Activity, Fragment'ы и т. д. Помимо этого здесь содержатся Presenter'ы и View (или ViewModel'и при использовании MVVM). В данном tutorialе для реализации слоя presentation будет использован шаблон MVP, но вы можете выбрать любой другой (MVVM, MVI) [17].

1.3.6 Presenter

Согласно концепции MVP, View не может напрямую взаимодействовать с Model, поэтому связующим звеном между ними является Presenter. Presenter реагирует на действия пользователя, о которых ему сообщила View (такие как нажатие на кнопку, пункт списка или ввод текста), после чего принимает решения о том, что делать дальше. Например, это может быть запрос данных у модели и отображение их во View.

2 «Логическая схема приложения»

В разделе 2 описывается работа и логика приложения. Каждый экран приложения имеет конкретную логику и для чего-то нужен. В мобильном приложении пользователь имеет возможность выполнить следующие действия:

1. Просмотреть список объявлений.
2. Отфильтровать список объявлений по категориям.
3. Выполнить поиск объявлений по имени.
4. Посмотреть детальную информацию об объявлении.
5. Зарегистрироваться в приложение.
6. Выполнить вход в приложение через электронную почту.
7. Создать объявление.
8. Просмотреть «Мои объявления».
9. Просмотреть настройки.

Ниже будет описан сокращенный основной функционал приложения.

2.1 Экран 1. Список товаров

При запуске пользователь попадает на главную страницу. На ней присутствует список товаров, список категорий, кнопка создания товара. При ошибке загрузки данных перед пользователем открывается Экран 2. При открытии бокового меню пользователь может переключиться на следующие экраны:

1. Страница авторизации.
2. Страница объявлений.
3. Страница объявлений пользователя.
4. Страница избранных объявлений.
5. Страница настроек.

При нажатии на любую категорию товары фильтруются по принадлежности к выбранной категории. При нажатии на

кнопку "Выбрать категорию" пользователь переходит на Экран 3. При нажатии на кнопку создать товар пользователь переходит на Экран 4. Элементы списка товаров представляют собой карточки, на которых изображена фотография товара, наименование, стоимость. В верхней части страницы присутствует кнопка поиска, при нажатии на которую можно ввести наименование товара, по которому отсортируется список. При нажатии на любой элемент списка пользователь переходит на Экран 5.

2.2 Фильтр по категориям

На странице фильтра по категориям пользователь видит список родительских категорий, при нажатии на элементы которого разворачиваются списки дочерних категорий для каждой родительской. Кроме того, первым элементом списка дочерних категорий является кнопка "Все категории". При нажатии на дочернюю категорию или на кнопку "Все категории" пользователь переходит на главную страницу и фильтрует товары по дочерней категории или по родительской соответственно.

2.3 Создание товара

Страница создания товара содержит поля для заполнения пользователем. Обязательными полями являются:

1. Наименование товара.
2. Описание товара.
3. Стоимость товара.
4. Наименование валюты.
5. Категория

Необязательными полями являются:

1. Фотография товара.
2. Адрес продавца.

2.4 Авторизация

На странице авторизации присутствуют кнопка для входа в

приложение по почте и кнопка регистрации. При нажатии на кнопку "войти через электронную почту" пользователь попадает на Экран 10. При нажатии на кнопку "Зарегистрироваться" пользователь попадает на Экран 11.

2.5 Вход через электронную почту

На странице логина "Вход через электронную почту" есть два поля (электронная почта и пароль), которые необходимо заполнить для входа в приложение. После заполнения полей кнопка "Войти" становится активной. При нажатии на нее, отправляется запрос на сервер, который проверяет наличие такого пользователя в базе данных. Если авторизация прошла успешно, пользователь попадает на Экран 1.

2.6 Регистрация

При открытии страницы показывается страница загрузки до того момента, пока не скачаются данные для выпадающих списков страны, региона и города. На странице регистрации присутствуют обязательные поля:

1. Имя.
2. Фамилия.
3. Электронная почта.
4. Пароль.

Также имеются необязательные для заполнения поля:

1. Дата рождения.
2. Страна.
3. Регион.
4. Город.
5. Фотография.

ЗАКЛЮЧЕНИЕ

В бакалаврской работе рассмотрены принципы разработки современного Android-приложения, использованы новейшие принципы и подходы к мобильной разработке мобильных приложений. Было реализовано мобильное приложение под ОС Android, в котором пользователь может размещать

онлайн-объявления, смотреть объявления других пользователей и связываться с ними для совершения сделок с помощью криптовалюты.

В итоге цель была достигнута и выполнены все задачи, поставленные во введении. Таким образом, был изучен теоретический материал, связанный с разработкой современных мобильных приложений под ОС Android, освоены инструменты для проектирования архитектуры и разработки приложения, а также реализовано само мобильное приложение.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 ОС Android // Официальный сайт Android [Электронный ресурс] : сайт. URL: <https://www.android.com> (дата обращения 30.04.2018). Загл. с экрана. Яз. рус.
- 2 Google I/O: Android O, нейросети, виртуальная реальность и другие новинки // Hi-tech@mail.ru [Электронный ресурс] : сайт. URL: <https://hi-tech.mail.ru/news/google-io-2017-android> (дата обращения 30.04.2018). Загл. с экрана. Яз. рус.
- 3 Android: A visual history // The Verge [Электронный ресурс] : сайт. URL: <https://www.theverge.com/2011/12/7/2585779/android-history> (дата обращения 30.04.2018). Загл. с экрана. Яз. англ.
- 4 Usman W. Chohan. Cryptocurrencies: A Brief Thematic Review. -- Wayback Machine.. Economics of Networks Journal, 2017, 3 с.
- 5 Retrofit // Square Open Source [Электронный ресурс] : сайт. URL: <http://square.github.io/retrofit/> (дата обращения 30.04.2018). Загл. с экрана. Яз. англ.
- 6 Изучаем Retrofit 2 // Хабрахабр [Электронный ресурс] : сайт. URL: <https://habr.com/post/314028> (дата обращения 01.05.2018). Загл. с экрана. Яз. рус.
- 7 Consuming APIs with Retrofit // Codepath [Электронный ресурс] : сайт. URL: <https://guides.codepath.com/android/consuming-apis-with-retrofit> (дата обращения 01.05.2018). Загл. с экрана. Яз. англ.

- 8 Using Retrofit 2.x as REST client - Tutorial // Vogella - Free Android courses [Электронный ресурс] : сайт. URL: <http://www.vogella.com/tutorials/Retrofit/article.html> (дата обращения 01.05.2018). Загл. с экрана. Яз. англ.
- 9 RxJava // Github - официальная документация RxJava [Электронный ресурс] : сайт. URL: <https://github.com/ReactiveX/RxJava> (дата обращения 01.05.2018). Загл. с экрана. Яз. англ.
- 10 Введение в RxJava: Почему Rx? // Хабрахабр [Электронный ресурс] : сайт. URL: <https://habr.com/post/269417/> (дата обращения 01.05.2018). Загл. с экрана. Яз. рус.
- 11 Using RxJava 2 - Tutorial // Vogella - Free Android courses [Электронный ресурс] : сайт. URL: <http://www.vogella.com/tutorials/RxJava/article.html> (дата обращения 03.05.2018). Загл. с экрана. Яз. англ.
- 12 Using RxJava 2 - Tutorial // Vogella - Free Android courses [Электронный ресурс] : сайт. URL: <http://reactivex.io/documentation> (дата обращения 03.05.2018). Загл. с экрана. Яз. англ.
- 13 Грокаем RxJava, часть первая: основы // Хабрахабр [Электронный ресурс] : сайт. URL: <https://habr.com/post/265269/> (дата обращения 03.05.2018). Загл. с экрана. Яз. рус.
- 14 Грокаем RxJava, часть вторая: Операторы // Хабрахабр [Электронный ресурс] : сайт. URL: <https://habr.com/post/265583/> (дата обращения 03.05.2018). Загл. с экрана. Яз. рус.
- 15 Грокаем RxJava, часть третья: Реактивность с пользой // Хабрахабр [Электронный ресурс] : сайт. URL: <https://habr.com/post/265727/> (дата обращения 03.05.2018). Загл. с экрана. Яз. рус.
- 16 Understanding Types Of Observables In RxJava // Medium [Электронный ресурс] : сайт. URL: <https://medium.com/mindorks/understanding-types-of-observables-in-rxjava-6c3a2d0819c8> (дата обращения 03.05.2018). Загл. с экрана. Яз. англ.

- 17 A detailed guide on developing Android apps using the Clean Architecture pattern // Medium [Электронный ресурс] : сайт. URL: <https://medium.com/@dmilicic/a-detailed-guide-on-developing-android-apps-using-the-clean-architecture-pattern-d38d71e94029> (дата обращения 07.05.2018). Загл. с экрана. Яз. англ.
- 18 Architecting Android...Reloaded // Fernando Cejas Blog [Электронный ресурс] : сайт. URL: <https://fernandocejas.com/2018/05/07/architecting-android-reloaded> (дата обращения 11.05.2018). Загл. с экрана. Яз. англ.
- 19 Заблуждения Clean Architecture // Хабрахабр [Электронный ресурс] : сайт. URL: <https://habr.com/company/mobileup/blog/335382/> (дата обращения 11.05.2018). Загл. с экрана. Яз. рус.
- 20 Clean Architecture Manifest // Github [Электронный ресурс] : сайт. URL: <https://github.com/ImangazalievM/CleanArchitectureManifest> (дата обращения 11.05.2018). Загл. с экрана. Яз. англ.