СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 5](#_Toc10639033)

[1 ОБЗОР ИСТОЧНИКОВ 6](#_Toc10639034)

[2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ 7](#_Toc10639035)

[3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ 9](#_Toc10639036)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 15](#_Toc10639037)

[5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 16](#_Toc10639038)

[6 ТЕСТИРОВАНИЕ 21](#_Toc10639039)

[ЗАКЛЮЧЕНИЕ 23](#_Toc10639040)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 24](#_Toc10639041)

[ПРИЛОЖЕНИЕ А 25](#_Toc10639042)

[ПРИЛОЖЕНИЕ Б 26](#_Toc10639043)

[ПРИЛОЖЕНИЕ В 27](#_Toc10639044)

[ПРИЛОЖЕНИЕ Г 28](#_Toc10639045)

# ВВЕДЕНИЕ

По данных Counterpoint Research, более половины пользователей проводят больше пяти часов в сутки за смартфоном. Можно сказать, что приложения для мобильных платформ актуальны теперь как никогда раньше.

Сегодня абсолютное большинство людей имеют аккаунты в нескольких социальных сетях. Twitter – популярная социальная сеть, характерной особенностью которой является особый стиль записей («твитов», от англ. «tweet» – щебетать). Ограничение каждого «твита» в 280 символов обуславливает особый лаконичный характер публикуемых пользователями сообщений.

В Twitter можно читать политических и общественных деятелей, пользователями Twitter являются большинство крупных организаций и брендов.

Целью данной работы является разработка Twitter-клиента для ОС Android. Twitter-клиент – приложение, осуществляющее обмен информацией с сервером Twitter посредством API. Минимальная функциональность должна включать в себя возможность просматривать новостную ленту, добавлять новые «твиты», просматривать аккаунты пользователей.

Заинтересованность в создании данного программного продукта обусловлена возможностью изучить разработку ПО для платформы Android, получить представление об основах Material Design, а также получить опыт работы с сетью.

# 1 ОБЗОР ИСТОЧНИКОВ

Среди языков программирования, популярных среди Android-разработчиков выделяются языки Java и Kotlin. В качестве языка, используемого для написания приложения, выбран язык Java, так как синтаксис и некоторые средства Java изучались в курсе Кроссплатформенного Программирования, а также существует большое количество источников для изучения опыта написания программ под платформу Android.

В отличие от языка С++ (от которого произошёл язык Java), многие аспекты Java созданы для удобства разработчика. Объектно-ориентированный подход, используемый в Java, предоставляет программисту возможность писать более читаемый код, а также легко адаптироваться к специфике любой задачи.

Разметка выполнена с помощью набора инструментов XML, по умолчанию предоставляемого Android. Такой вид разметки позволяет разделять графический пользовательский интерфейс от логики программы. Таким образом, внешний вид приложения можно менять независимо от кода.

В качестве рекомендуемой среды разработки под Android Google предлагает Android Studio. Android Studio предоставляет возможности разработки, отладки и сборки многофайловых проектов, а также позволяет эмулировать конфигурации различных устройств с целью тестирования приложений.

Для взаимодействия с сервером Twitter использована Twitter SDK. Отрицательной стороной использования данной библиотеки является практически полное отсутствие структурированной документации, а также неработоспособность некоторых модулей. Положительной – существующие готовые решения типовых проблем, возникающих в ходе разработки клиентских приложений для Twitter.

Дополнительно для асинхронной загрузки изображений из сети была использована библиотека Picasso, также используемая авторами Twitter SDK.

# 2 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ

Наиболее существенная задача приложения-клиента – организовать коммуникацию между клиентм и сервером, а также взаимодействие пользователя с клиентом (косвенно, с сервером). Соответственно, приложение должно предоставлять удобный интерфейс пользователя, в зависимости от состояния которого вызываются необходимые участки кода и выполняется логика программы.

В зависимости от манипуляций пользователя, механизм клиента отправляет запросы серверу на получение некоторой информации и\или запрашивает изменения данных, хранящихся на сервере. Затем клиент ожидает ответа, обрабатывает его и меняет собственные данные, которые в изменённом виде необходимо отобразить в интерфейсе.

Графический интерфейс пользователя данного приложения объединяет несколько xml-файлов, соответствующих главному экрану, странице информации о пользователе, страницу поиска и различные вспомогательные элементы: фрагменты ленты твитов, меню тулбара и др.

Такие затратные с точки зрения процессорного времени операции, как, например, сетевые запросы, целесообразно выполнять асинхронно по отношению к основному потоку. В качестве инструмента для решения этой задачи выбран AsyncTask.



Рисунок 2.1 – Структурная схема приложения

# 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ

В объектно-ориентированном подходе к программированию функциональным блокам соответствуют классы. Рассмотрим наиболее существенные классы. Диаграмма классов продемонстрирована в приложении А.

**3.1 Класс LoginActivity**

Класс, реализующий авторизацию и хранение информации о ней. Страница, соответствующая этому классу, не отображается, если пользователь уже авторизован.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| loginButton | TwitterLoginButton | Механизм авторизации из Twitter SDK | |
| token | String | Информация о сессии | |
| secret | String |
| session | TwitterSession | Актуальная Twitter-сессия | |
| isUserAuthorized | Boolean | Информация о том, авторизован ли пользователь | |
| mShared | SharedPreferences | Механизм постоянного хранения информации об авторизации | |
| Private методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| saveSessionInfo | String, String | Void | метод, используемый для записи информации о сессии в SharedPreferences |
| getSessionInfo | Void | Void | метод, используемый для получения информации о сессии из SharedPreferences |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity в Android, составляющая паттерна «Шаблонный метод» |
| onActivityResult | Int, int, Intent | Void | Передаёт результат авторизации в LoginButton |

**3.2 Класс TimelineActivity**

Данный абстрактный класс соответствует любому Activity-классу, содержащему в себе ленту «твитов».

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| tweetsRecyclerView | RecyclerView | Элемент View, используемый для отображения «твитов» | |
| tweetAdapter | TweetAdapter | Объект, формирующий фрагмент RecyclerView из информации о «твите» | |
| task | AsyncTask<Void, Void, Void> | Средство асинхронного выполнения запроса к серверу | |
| Package-private методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| initToolbar | Void | Void | Метод инициализации toolbar |
| initTask | Void | Void | Метод инициализации AsyncTask. Абстрактен, т.к. каждый класс Activity отображает особый вид ленты, а значит, информация о «твитах» запрашивается по-разному |
| loadTweets | Void | Void | Загружает «твиты» |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity |

**3.3 Класс StandardTimelineActivity**

Наследник TimelineActivity, особенность которого – стандартный тулбар с меню приложения.

|  |  |  |  |
| --- | --- | --- | --- |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreateOptionsMenu | Menu | Boolean | Метод инициализации меню в тулбаре |
| onOptionsItemSelected | MenuItem | Boolean | Метод обработки нажатия на иконки в тулбаре |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity. Инициализирует все поля и вызывает метод loadTweets() родительского класса |

**3.4 Класс MainActivity**

MainActivity – «минимальный» класс-наследник StandardTimelineActivity, переопределяет методы OnCreate и initTask (инициализирует поле task родительского класса через TimelineTask).

**3.5 Класс UserInfoActivity**

Класс – клас-наследник StandartTimelineActivity, имеющий, соответственно, тулбар с меню, RecyclerView для отображения твитов. Так как класс представляет собой логику страницы информации о пользователе, особенность класса – наличие блока набора View для отображения основных данных о пользователе.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| userId | Long | ID пользователя | |
| nameTextView, nickTextView,  userImageView,  descriptionTextView,  followingCountTextView,  followersCountTextView | View | Объекты блока данных о пользователи | |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| displayUserInfo | User | Void | Заполняет поля информации о пользователе актуальной информацией |
| Package-private методы класса | | | |
| initTask | Void | Void | Метод инициализации AsyncTask (UserInfoTask) |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity, инициализирует все элементы интерфейса по значению ID пользователя, получаемого из extras |

**3.6 Класс SearchActivity**

Данный класс реализует поиск «твитов» по запросу пользователя.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| searchText | EditText | Элемент View, используемый для ввода пользователем поискового запроса | |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| initToolBar | Void | Void | Метод инициализации поля ввода запроса в тулбаре |
| initTask | Void | Void | Метод инициализации AsyncTask (SearchTask) |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| onCreate | Bundle savedInstanceState | Void | основной метод Activity. |

**3.7 Класс TweetAdapter**

Данный класс является адаптером, предназначенных для создания специальных элементов, отображающих информацию о «твите», и заполнения ими RecyclerView. TweetAdapter имеет в себе вложенный класс TweetViewHolder, реализующий паттерн ViewHolder, суть которого в создании объекта, хранящего ссылки на все элементы View, для каждого фрагмента RecyclerView с целью минимизировать количество дорогостоящего в смысле затрат процессорного времени метода findViewById.

**3.8 Класс TimelineTask**

Класс, реализующий асинхронную загрузку информации с сервера.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| tweetAdapter | TweetAdapter | Адаптер, инициализируемый в конструкторе. Заполняется элементами в ходе выполнения основного метода | |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| TimelineTask | TweetAdapter | Void | Конструктор класса |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| doInBackground | Void | Void | основной метод AsyncTask, аналогичный методу run() в Thread. |

**3.8 Класс UserInfoTask**

Класс, реализующий асинхронную загрузку информации с сервера.

|  |  |  |  |
| --- | --- | --- | --- |
| Private поля класса | | | |
| Имя | Тип | Описание | |
| tweetAdapter | TweetAdapter | Адаптер, инициализируемый в конструкторе. Заполняется элементами в ходе выполнения основного метода | |
| userId | Long | ID пользователя | |
| Context | UserInfoActivity | Activity, из которого вызывается выполнение AsyncTask. Хранится, чтобы заполнить не только RecyclerView, но и блок информации о пользователе посредством вызова метода displayUserInfo класса UserInfoActivity | |
| Public методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| UserInfoTask | TweetAdapter, Long, UserInfoActivity | Void | Конструктор класса |
| Protected методы класса | | | |
| Имя | Принимаемые параметры | Возвращаемое значение | Описание |
| doInBackground | Void | Void | Переопределяет метод родительского класса. |

# 4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

В данном разделе будут рассмотрены алгоритмы, используемые в разрабатываемом ПО.

**4.1. Авторизация пользователя**

Блок-схема алгоритма приведена в приложении Б.

Шаг 1. Создать Twitter-сессию, используя CONSUMER\_KEY и CONSUMER\_SECRET из ресурсов.

Шаг 2. Получить информацию об авторизации из SharedPreferences.

Шаг 3. Если пользователен авторизован, перейти в MainActivity. Если нет, установить на TwitterLoginButton callback.

Шаг 4. В случае успешной авторизации, сохранить информацию о сессии в SharedPreferences. В обратном случае – кинуть исключение и вернуться в LoginActivity.

Шаг 5. Завершить алгоритм.

**4.2. Загрузка и демонстрация информации о пользователе**

Блок-схема алгоритма приведена в приложении В.

Шаг 1. Получить ID пользователя из extras.

Шаг 2. Инициализировать UserInfoTask и вызвать execute().

Шаг 3. Осуществить запрос.

Шаг 4. В случае успеха – передать информацию о «твитах» в TweetAdapter, заполнить блок информации о пользователе. В случае неудачи – вывести сообщение об ошибке.

Шаг 5. Завершить алгоритм.

# 5 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

Минимальные системные требования:

* ОС Android, минимальная версия 4.2;
* Мобильный телефон на базе Android;
* Веб-браузер.

Комплект поставляемого ПО:

* исполняемый .APK файл «Twitty».

Для запуска программы:

1. Получить готовый .APK файл.
2. Запустить .APK файл «Twitty», нажав на соответствующую иконку в меню (см. рисунок 5.1). Дождаться появления стартового окна (см. рисунок 5.2).

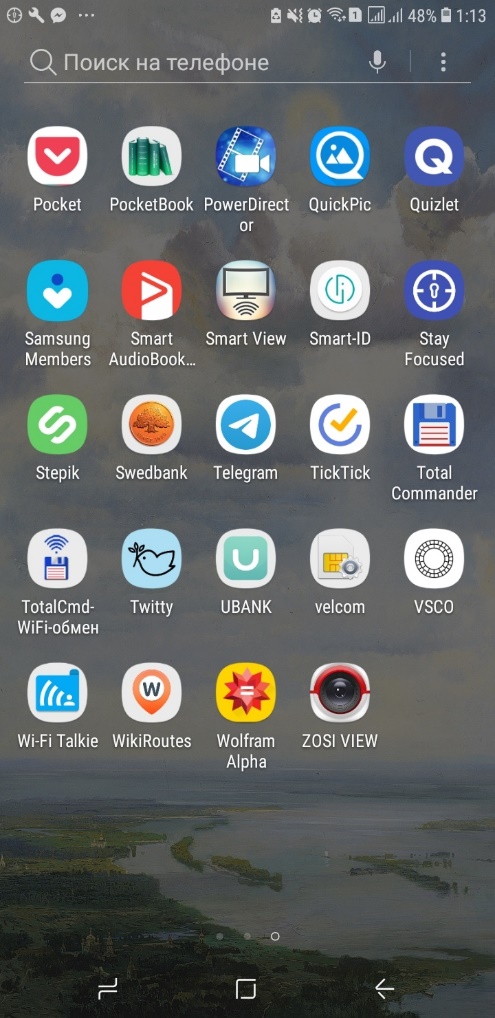


Рисунок 5.1 – Иконка приложения в меню



Рисунок 5.2 – Стартовая страница приложения

Как пользоваться программой:

1. Для того, чтобы использовать приложение, необходимо пройти процедуру авторизации, ввести логин и пароль Twitter.
2. После успешной авторизации пользователь попадает на главную страницу приложения с лентой «твитов» тех пользователей, на которых он подписан.
3. Взаимодействовать с «твитами» можно посредством иконок. В наличии две функции: добавить в избранное и ретвитнуть. Также можно перейти на страницу пользователя, кликнув на его аватар или логин.
4. Для того, чтобы воспользоваться поиском по твитам, необходимо кликнуть на иконку «лупа» на тулбаре. Результат поиска – лента «твитов» – обновляется после каждого изменения запроса, в т.ч. по мере ввода слова поиска.
5. Чтобы добавить «твит», следует кликнуть на иконку «плюс», расположенную в меню тулбара, затем ввести текст «твита» и подтвердить публикацию. Необходимо помнить, что максимальный объём «твита» – 280 символов.
6. При возникновении вопросов по эксплуатации следует связаться с автором продукта.



Рисунок 5.3 – Процесс авторизации

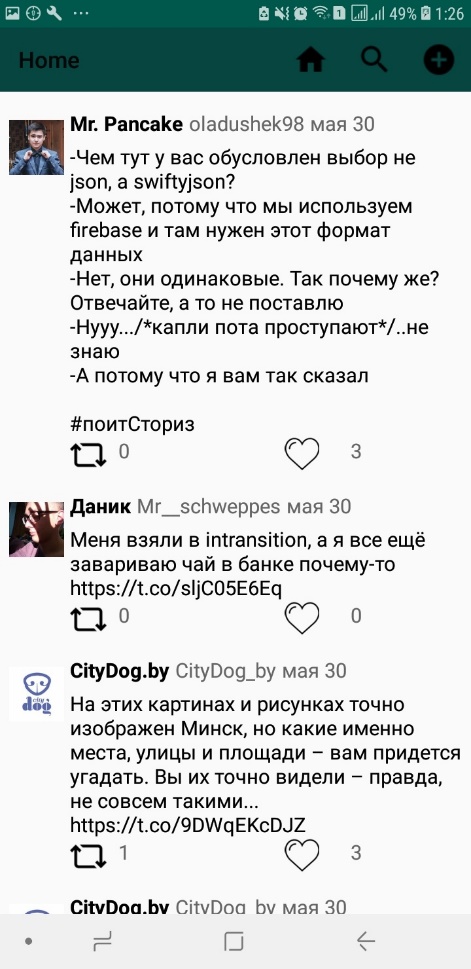


Рисунок 5.4 – Главная страница приложения



Рисунок 5.5 – Страница пользователя



Рисунок 5.6 – Страница поиска

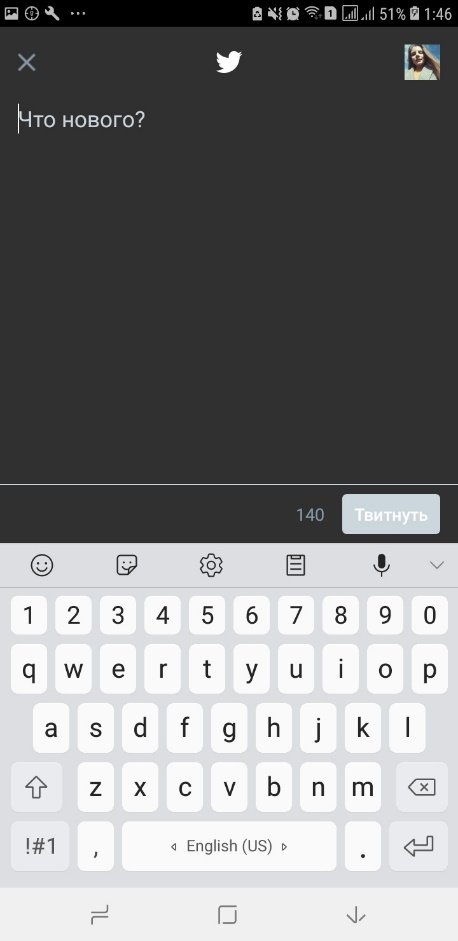


Рисунок 5.7 – Страница добавления «твита»

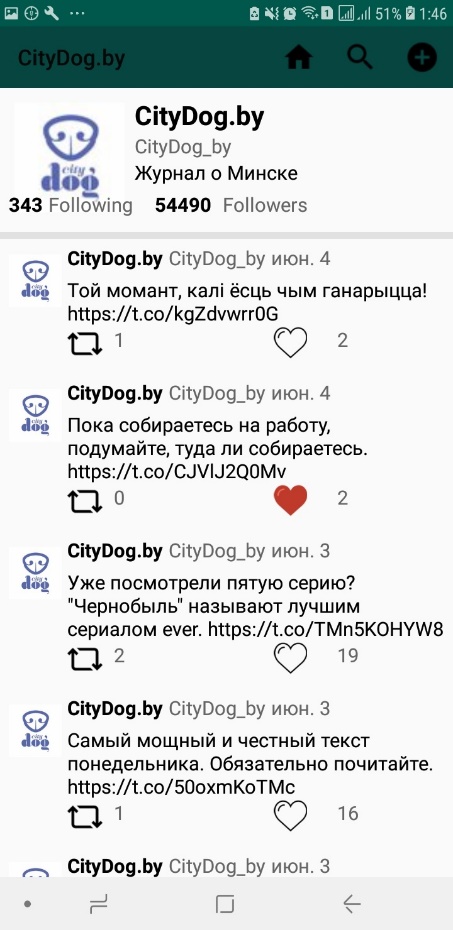


Рисунок 5.8 – Демонстрация опции «Добавить в избранное»

# 6 ТЕСТИРОВАНИЕ

Для тестирования были выбраны следующие устройства: Xiomi Redmi Note 5, Xiomi Redmi 6A и Samsung Galaxy J8 (скриншоты тестирования на устройстве Samsung приведены выше).

В ходе тестирования на смартфоне Xiomi Redmi Note 5 была выявлена следующая неполадка: TwitterLoginButton не работала. Как выяснилось, проблема решилась, после того, как был установлен официальный Twitter-клиент. Первопричина крылась в изначально неверной регистрации приложения на https://developer.twitter.com/en/apps. Недостаток был исправлен.

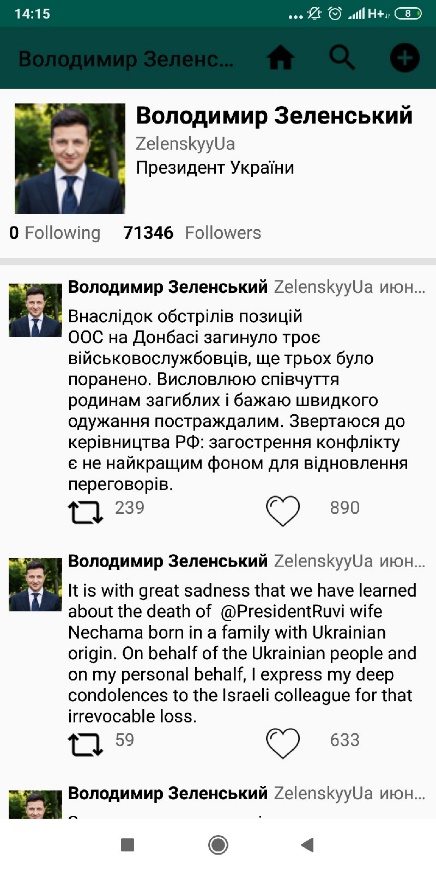
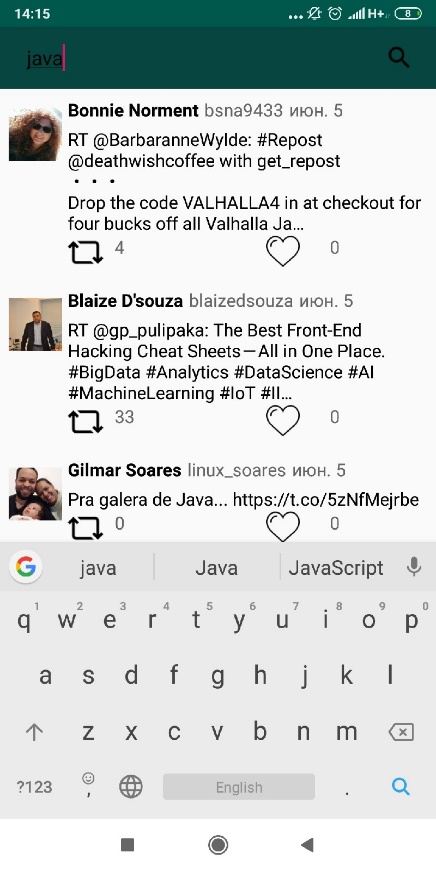
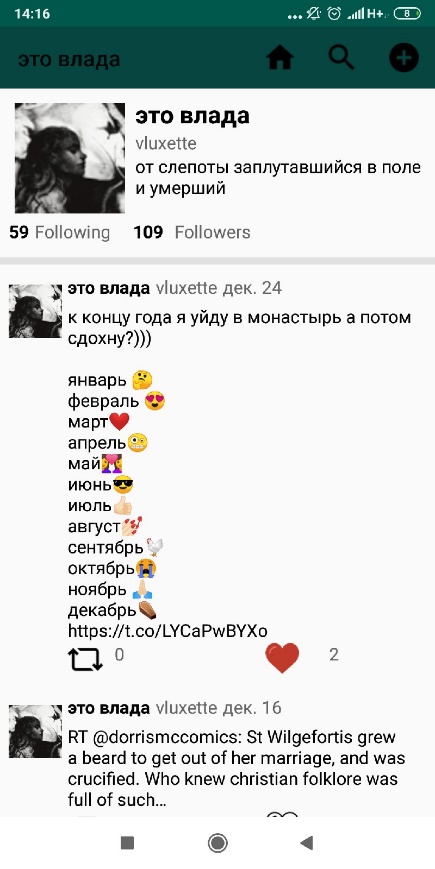


Рисунок 6.1 – Демонстрация работы приложения на устройстве Xiomi Redmi Note 5

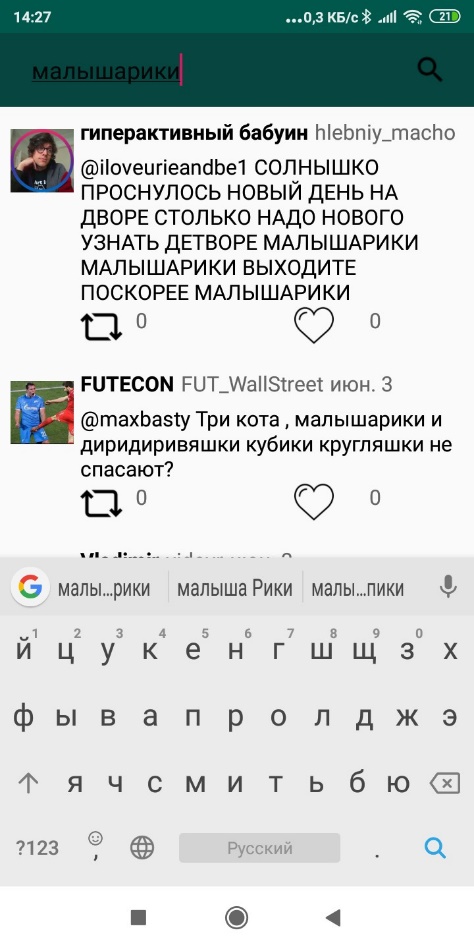
  

Рисунок 6.1 – Демонстрация работы приложения на устройстве Xiomi Redmi 6A

# ЗАКЛЮЧЕНИЕ

В ходе работы над курсовым проектом было сконструировано приложение-клиент для социальной сети Twitter. Процесс разработки задокументирован в данной пояснительной записке.

Данный продукт имеет ряд преимуществ: приятный интуитивно понятный интерфейс, малый размер исходного .APK файла. Однако есть и недостатки: отсутствие обновления страницы по жесту «свайп вниз», отсутствие возможности просматривать видео и комментировать «твиты» прямым образом.

В дальнейшем планируется исправление указанных выше выявленных недостатков, доработка графического интерфейса и расширение функциональности приложения.

Хочется отметить, что процесс разработки принёс автору массу нового опыта, а также провоцировал автора искать решения проблем в различных источниках, разбираться в исходном коде Twitter SDK ввиду отсутствия исчерпывающей документации и даже исправлять отдельные модули. Изучена реализация клиент-серверной архитектуры приложения, а также способы проведения асинхронных операций в ОС Android.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Twitter Documentation – Twitter Documentation For Developers [Электронный ресурс]. – Электронные данные. – Режим доступа: https://developer.twitter.com/en/docs.

[2] Habr – Habr [Электронный ресурс]. – Электронные данные. – Режим доступа: https://habr.com/.

[3] Дейтел П., Android для разработчиков / П. Дейтел, Х. Дейтел, О. Уолд. – Спб.: Питер, 2016. – 512 с.

[4] Start Android – Учебник по Android для начинающих и продвинутых (электронный ресурс). – Электронные данные. – Режим доступа: http://startandroid.ru/ru/.

# ПРИЛОЖЕНИЕ А

Диаграмма классов

# ПРИЛОЖЕНИЕ Б

Схема алгоритма авторизации пользователя

# ПРИЛОЖЕНИЕ В

Схема алгоритма запроса и демонстрации информации о пользователе

# ПРИЛОЖЕНИЕ Г

Листинг программы