Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных сетей и систем

Кафедра электронных вычислительных машин

Дисциплина: Системное программное обеспечение ВМ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Twitter - клиент для платформы Android

БГУИР КП 1-40 02 01 01\* ПЗ

Выполнила:

Студентка гр. 550501 Аль Шейхахмад Г.А.

Проверил: Лавникевич Д.А.

Минск, 2017

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 4](#_Toc484745201)

[**1 ЦЕЛИ И ЗАДАЧА КУРСОВОЙ РАБОТЫ** 5](#_Toc484745202)

[**2 ОБЗОР ЛИТЕРАТУРЫ** 6](#_Toc484745203)

[**2.1 Обзор модуля Fabric (Twitter Kit)** 6](#_Toc484745204)

[**2.2 Обзор аналогов** 6](#_Toc484745205)

[**3**  **СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ** 11](#_Toc484745206)

[**3.1 Структура входных данных** 11](#_Toc484745207)

[**3.2 Структура выходных данных** 11](#_Toc484745208)

[**4 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ** 12](#_Toc484745209)

[**4.1 Класс Const.java** 12](#_Toc484745210)

[**4.2 Класс LoginActivity.java** 13](#_Toc484745211)

[**4.3 Класс MainActivity.java** 13](#_Toc484745212)

[**4.4 Класс SearchActivity.java** 13](#_Toc484745213)

[**4.5 Класс SearchTimeLineFragment.java** 14](#_Toc484745214)

[**4.6 Класс TimeLineFragment.java** 14](#_Toc484745215)

[**4.7 Класс HomeActivity.java** 14](#_Toc484745216)

[**4.8 Файл activity\_login.xml** 14](#_Toc484745217)

[**4.9 Файл activity\_main.xml** 15](#_Toc484745218)

[**4.10 Файл activity\_home\_time\_line.xml** 15](#_Toc484745219)

[**4.11 Файл activity\_search.xml** 15](#_Toc484745220)

[**4.12 Файл timeline\_fragment.xml** 15](#_Toc484745221)

[**5 РАЗРАБОТКА АЛГОРИТМОВ** 16](#_Toc484745222)

[**5.1 Схема алгоритма функции** TweetTimelineListAdapter searchTimeLine() 16](#_Toc484745223)

[**5.2 Схема алгоритма функции** void void getInfo() 16](#_Toc484745224)

[**5.3 Схема алгоритма функции** void putInfo(Result<User> result) 16](#_Toc484745225)

[**6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ** 17](#_Toc484745226)

[**7 РЕЗУЛЬТАТЫ РАБОТЫ** 18](#_Toc484745227)

[**ЗАКЛЮЧЕНИЕ** 24](#_Toc484745228)

[**ЛИТЕРАТУРА** 25](#_Toc484745229)

[**ПРИЛОЖЕНИЕ А** 26](#_Toc484745230)

[**ПРИЛОЖЕНИЕ Б** 27](#_Toc484745231)

[**ПРИЛОЖЕНИЕ В** 28](#_Toc484745232)

# 

# **ВВЕДЕНИЕ**

В данной курсовой работе Twitter-клиент. Для реализации данного клиента использовался язык программирования Java. Вместе с тем использовался модуль Fabric (Twitter Kit).

Twitter - клиент — это клиентское приложение, позволяющее авторизоваться в своем аккаунте социальной сети Twitter, использовать основные функции данной социальной сети.

Социальная сеть Twitter на сегодняшний день является одной из самых известных и пользующихся спросом социальных сетей мирового уровня, таких как Facebook, Telegram, Instagram и т.п.

Twitter создан для того, чтобы люди могли легко и кратко делиться какими-либо новостями, полезными и не очень. Эта социальная сеть очень проста и, наверное, этим и привлекает такое количество людей.

Наряду со стандартными функциями, некоторые веб-серверы имеют дополнительные. Так, к примеру, соответствующее программное обеспечение может фиксировать число обращений пользователей к тому или иному ресурсу, записывать их в отдельный журнал.

# **1 ЦЕЛИ И ЗАДАЧА КУРСОВОЙ РАБОТЫ**

*Цели курсовой работы:* освоение практических навыков проектирования и разработки законченного, отлаженного и протестированного программного продукта; закрепление и углубление теоретических знаний, полученных при изучении курса «Системное программное обеспечение ВМ», изучить и использовать сторонние средства.

*Задача курсовой работы:* проектирование программного обеспечения, a именно Twitter-клиент под платформу Android.

Разрабатываемый клиент должен осуществлять поиск «твитов», вывод новостной и пользовательской ленты, написание новых «твитов». Использовать в программе сторонние библиотеки.

# **2 ОБЗОР ЛИТЕРАТУРЫ**

## **2.1 Обзор модуля Fabric (Twitter Kit)**

При разработке twitter-клиента под платформу Android на языке Java был использован модуль Fabric.   
Данный модель позволяет с легкостью внедрять в свое приложение полезные функции. С его помощью была реализована авторизация в приложении, вывод «твитов», поиск и извлечение информации о пользователе.

Для того, чтобы использовать Fabric, его необходимо установить. А сделать это очень просто в самой Android Studio.  
После установки можно выбрать необходимые вам библиотеки и функции. Одним нажатием кнопки Fabric самостоятельно внедрит в ваше приложение необходимые библиотеки. И после чего еще предложит подробную инструкцию для реализации некоторых функций.

А уже для получения большей информации о возможностях Fabric можно узнать на сайте <https://docs.fabric.io>.

## **2.2 Обзор аналогов**

 На сегодняшний день существует множество Twitter-клиентов. Многие их которых очень похожи, многие из них очень удобны в использовании и пользуются большим спросом, а многие совсем малоизвестны из-за недостатков в функционале и интерфейсе.

В данном разделе будут приведены примеры аналогов. Будут кратко рассмотрены их достоинства и недостатки.

* **Twitter for Android (официальное приложение)**

Удобство чтения ленты. Главным преимуществом официального twitter-клиента всегда была скорость работы, в отличие от многих сторонних приложений, в нем не тормозит пролистывание ленты, даже на большой скорости. В новой версии появилось переключение вкладок с помощью жестов вправо/влево, однако работа с данным обновлением приносит немалые неудобства.

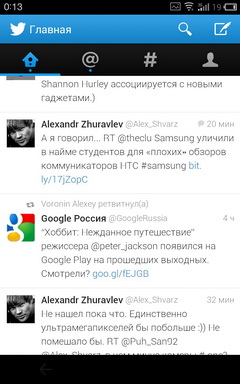


Рисунок 2.2.1 – Twitter-клиент (официальное приложение)

**Работа с отдельным «твитом».** При нажатии на «твит», он открывается в отдельном окне. Отсюда же можно быстро «ретвитнуть» сообщение, ответить автору или добавить его в избранное.

Плюсы:

1. Бесплатный
2. Поддержка изображений с pic.twitter.com
3. Holo-стиль

Минусы:

1. Нет поддержки картинок в ленте
2. Тормоза при переходе с одной вкладки на другую
3. Нет поддержки twitlonger
4. Тонкий плохо читаемый шрифт

* **Plume**

«Плюм» является одним из самых старых и функциональных твиттер-клиентов на Android. Год назад его даже можно было назвать лучшим.  
  
**Удобство чтения ленты.** Первое на что обращаешь внимание – количество информации, поместившееся на экране. В Plume очень грамотно распорядились местом, в результате пользователям нужно гораздо реже листать ленту. Переключение между вкладками осуществляется с помощью горизонтального свайпа.

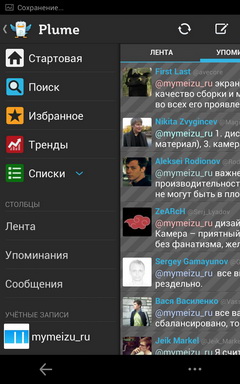


Рисунок 2.2.2 – Plume

**Работа с изображениями.** Клиент поддерживает отображение картинок в ленте, по умолчанию это миниатюры, но в настройках можно включить и большие превью.

**Работа с отдельным «твитом».** При нажатии на «твит», перед вами открывается небольшая панель для работы с ним («ретвит», ответы, добавление в избранное и переход по ссылке). Если «твит» является частью диалога, то он тоже подгружается.

Плюсы:

1. Поддержка встроенного браузера
2. Большие миниатюры в ленте
3. Настройки фотохостинга
4. Чтение и публикация с помощью twitlonger
5. Множество настроек

Минусы:

1. Скорость и плавность работы

* **Robird**

**Удобство чтения ленты.** По умолчанию лента Robird выполнена в темных тонах, но при желании тема меняется в настройках. Разработчик уделил много внимания скорости и плавности работы, поэтому Robird на данный момент можно назвать самым быстрым и плавным twitter-клиентом.

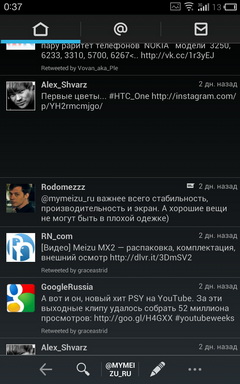


Рисунок 2.2.3 – Robird

**Работа с изображениями.** Как и другие twitter-клиенты, Robird поддерживает почти все популярные фотохостинги, от официального pic.twitter.com до Instagram. Интересно, что Robird корректно работает и несколькими вложенными картинками, кроме того с его помощью вы и сами можете опубликовать сразу несколько изображений. В ленте Robird отображает картинки в виде прямоугольных миниатюр, при нажатии на них картинка открывается полностью.

**Работа с отдельным «твитом».** При нажатии на «твит» он выделяется, а кнопки на SmartBar меняются. Теперь там появляются иконки для работы с «твитом»: быстрый ответ, «ретвит», добавление в избранное. Если вы хотите посмотреть всю переписку, достаточно долгого нажатия на «твит».

Плюсы:

1. Скорость и плавность работы
2. Полная поддержка SmartBar
3. Удобная навигация
4. Поддержка практически всех фотохостингов
5. Поддержка нескольких изображений в ленте

Минусы:

1. Отсутствие встроенного браузер
2. Отсутствие поддержки нескольких аккаунтов

# **3 СТРУКТУРНОЕ ПРОЕКТИРОВАНИЕ**

## **3.1 Структура входных данных**

Используемые файлы: «ic\_account\_balance.png», «ic\_search.png», «login\_activity.jpg», «border\_bottom.xml», «border\_et.xml», «border\_layout.xml», «button\_style.xml» .

Файлы с расширением .png и .jpg в качестве иконок некоторых кнопок, а также в качестве фонового изображения Activity. Файлы с расширением .xml служат для описания стилей некоторых виджетов.

## **3.2 Структура выходных данных**

Структура данных не изменилась.

# **4 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ**

## **4.1 Класс Const.java**

В данном классе хранятся константы, необходимые для входа в аккаунт через приложение, а также поля, кторые будут использоваться в качестве ключей для передачи некоторых параметров, присущих пользователю.

Класс Const имеет следующие поля:

String USER\_NAME – ключ имени пользователя.

String USER\_ID – ключ идентификатора пользователя.

String SCREEN\_NAME – ключ имени пользователя.

String LOCATION – ключ местоположения пользователя.

String FOLLOWING\_COUNT – ключ количества подписок пользователя.

String FOLLOWERS\_COUNT – ключ количества подписчиков пользователя.

String TWEET\_COUNT – ключ количества «твитов» пользователя.

String PROFILE\_IMG\_URL – ключ URL фото пользователя.

String PROFILE\_IMG\_BG\_URL – ключ URL фонового фото пользователя.

String SEARCH\_TEXT – ключ текста для поиска пользователя.

String CONSUMER\_KEY – ключ, который дается при регистрации приложения в сети Twitter.

String CONSUMER\_SECRET – секретный пароль, который также дается при регистрации приложения, соответствующий CONSUMER\_KEY.

## **4.2 Класс LoginActivity.java**

LoginActivity – это Activity, которая отображается при запуске приложения. Она реализует вход пользователя в аккаунт.

Она включает в себя один элемент – кнопку входа:

TwitterLoginButton loginButton;

Класс, реализующий данную кнопку принадлежит Twitter библиотеке, подключить которую можно в Android Studio (com.twitter.sdk.android.core).

При нажатии на кнопку происходит перенаправление на сайт Twitter, где пользователю предлагают разрешить авторизацию через Twitter в данном приложении.

После авторизации данные о пользователе передаются в MainActivity для дальнейшей работы с ними.

## **4.3 Класс MainActivity.java**

В MainActivity создаются кнопки FloatingActionButton, которые располагаются в Activity поверх всех слоев и виджетов.

Таких кнопок три:

1. add\_tweet\_btn – кнопка для написания нового твита.
2. search\_btn – кнопка для поиска среди «твитов».
3. home\_btn – кнопка для просмотра новостной ленты.

После вызывается функция, которая принимает данные, переданные ранее из LoginActivity.

Затем создается экземпляр TimeLineFragment, в котором будет отображаться лента «твитов» пользователя.

## **4.4 Класс SearchActivity.java**

В SearchActivity присутсвует поле для ввода текста, по которому будет осуществляться поиск, а также две кнопки:

1. search\_btn – кнопка для поиска.
2. clear\_btn – кнопка для очистки поля для ввода.

При нажатии на кнопку поиска создается экземпляр SearchTimeLineFragment, в котором будет отображаться список «твитов», удовлетворяющих поисковому запросу.

### **4.5 Класс SearchTimeLineFragment.java**

SearchTimeLineFragment – это фрагмент типа ListFragment, который располагается в SearchActivity. В нем создается экземпляр SearchTimeLine в который передается текст, по которому будет осуществляться поиск.

Итоговый список и будет отображать данный фрагмент.

### **4.6 Класс TimeLineFragment.java**

TimeLineFragment – это фрагмент типа ListFragment, который располагается в MainActivity. В нем создается экземпляр UserTimeLine в который передается имя пользователя, чей список твитов и будет отображаться в данном фрагменте.

## **4.7 Класс HomeActivity.java**

В HomeActivity отображается новостная лента пользователя, т.е. новые «твиты» подписок пользователя.

В ней создается экземпляр FixedTweetTimeLine и адаптер TweetTimeLineListAdapter, в который передается имя пользователя, для которого будет отображаться новостная лента в данном фрагменте.

## **4.8 Файл activity\_login.xml**

В данном файле прописана xml разметка LoginActivity.

Она включает в себя фоновое изображение Activity и кнопку для в хода в аккаунт с помощью Twitter.

## **4.9 Файл activity\_main.xml**

В данном файле прописана xml разметка MainActivity.

Она включает в себя:

1. Фоновое изображение страницы пользователя.
2. Главное изображение аккаунта.
3. Имя пользователя.
4. Логин пользователя.
5. Количество «твитов» пользователя.
6. Количество подписок пользователя.
7. Количество подписчиков пользователя.
8. TimeLineFragment, в котором отображается список «твитов» пользователя.

## **4.10 Файл activity\_home\_time\_line.xml**

В данном файле прописана xml разметка HomeActivity.

В ней содержится фрагмент, в котором отображается новостная лента.

**4.11 Файл activity\_search.xml**

В данном файле прописана xml разметка SearchActivity.

В ней содержится фрагмент, в котором отображается лента «твитов», удовлетворяющая поиску.

## **4.12 Файл timeline\_fragment.xml**

В данном файле прописана xml разметка TimeLineFragment.

В ней содержится фрагмент, в котором отображается лента пользователя.

# **5 РАЗРАБОТКА АЛГОРИТМОВ**

С точки зрения алгоритмов, интерес представляют алгоритмы реализующие:

* Создание списка результатов поиска searchTimeLine()
* Получение данных из другой Activity getInfo()
* Передачу данных в другую Activity putInfo(Result<User> result)

## **5.1 Схема алгоритма функции** TweetTimelineListAdapter searchTimeLine()

Данная функция реализует создание списка, являющегося результатом поиска среди «твитов» в системе. Разработанная схема представлена на чертеже в Приложении А.

## **5.2 Схема алгоритма функции** void void getInfo()

Данная функция принимает данные о пользователе из LoginActivity при входе. Разработанная схема представлена на чертеже в Приложении Б.

## **5.3 Схема алгоритма функции** void putInfo(Result<User> result)

Данная функция передает данные о пользователе в MainActivity при входе в аккаунт. Разработанная схема представлена на чертеже в Приложении В.

# **6 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

Минимальные системные требования:

* Смартфон на платформе Android версии не менее 4.0
* Доступ в интернет
* Доступ к учетной записи пользователя в социальной сети Twitter.

Приложение сохраняет текущую сессию, что позволяет пользователю проходить авторизацию лишь раз. При необходимости, из учетной записи можно выйти и войти в другой аккаунт.

# **7 РЕЗУЛЬТАТЫ РАБОТЫ**

В данном разделе проведем небольшой обзор разработанного приложения.

После запуска приложения на главном экране, представленном на рисунке 7.1, будет располагаться кнопка, позволяющая пользователю войти в приложение с помощью Twitter.

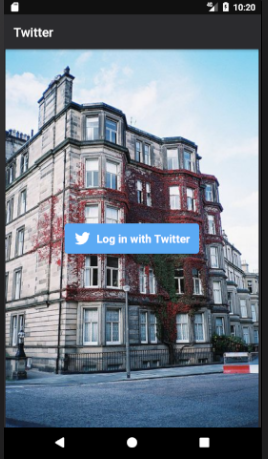


Рисунок 7.1 – Стартовая страница приложения

После нажатия кнопки, пользователь будет перенаправлен на сайт для в хода в учетную запись Twitter ( рисунок 7.2.).

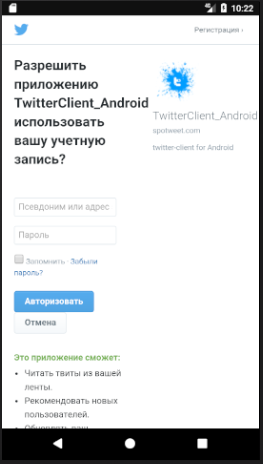


Рисунок 7.2 – Страница браузера для входа в аккаунт

После входа в аккаунт и подтверждения авторизации (разрешения приложению выполнять доступные ему операции) пользователю откроется новая Activity, в которой будет находится информация о пользователе и список его «твитов», как показано на рисунке 7.3. Также будут доступны три кнопки, которые осуществляют написание новых «твитов», поиск среди «твитов» и просмотр новостной ленты.

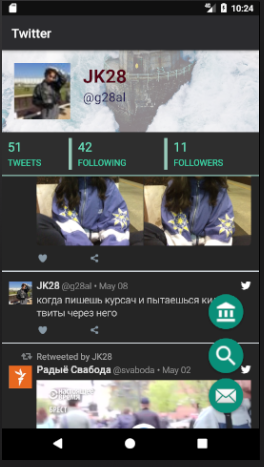


Рисунок 7.3 – Страница профиля

При нажатии на кнопку с иконкой сообщения, пользователь будет перенаправлен в новую Activity, в которой реализована возможность написания нового «твита». Изображение данной Activity представлено на рисунке 7.4.

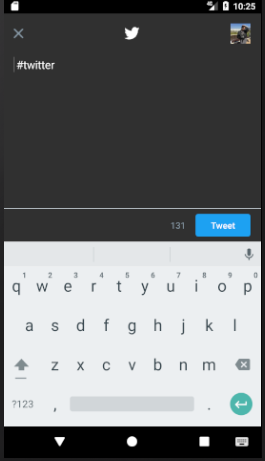


Рисунок 7.4 – Activity для написания нового «твита»

При нажатии на странице профиля кнопки поиска, пользователю откроется Activity для осуществления поиска. Результат поиска среди «твитов» по хэш-тэгу «#twitter» представлен на рисунке 7.5.



Рисунок 7.5 – Результат поиска по хэш-тэгу «#twitter»

При нажатии на странице профиля кнопки Home, пользователю откроется Activity в которой будет отображаться новостная лента, как показано на рисунке 7.6.

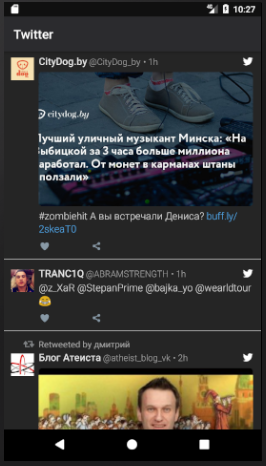


Рисунок 7.6 – Новостная лента

# **ЗАКЛЮЧЕНИЕ**

В данном курсовом проекте был разработан «Twitter-клиент под платформу Android», позволяющий просматривать новостную ленту, ленту собственных «твитов», писать новые «твиты», а также осуществлять поиск среди «твитов». К достоинствам данной программы относится простота использования приложения, а также приятный, с эстетической точки зрения, интерфейс. Конечно, по своему функционалу это приложение уступает официальному приложению Twitter, но в дальнейших планах лежит усовершенствование данного приложения, добавления новых необходимых функций и экспериментов в плане графического интерфейса.

В ходе разработки данного клиентского приложения были получены глубокие знания в разработке ПО под платформу Android, также приобретены новые навыки в языке программирования Java.

# **ЛИТЕРАТУРА**

1. Интернет-ресурс: <http://developer.android.com>.
2. Интернет-ресурс: <http://startandroid.ru>.
3. Интернет-ресурс (документация Fabric): <https://docs.fabric.io>.
4. Интернет-ресурс: <http://developer.alexanderklimov.ru>.

# **ПРИЛОЖЕНИЕ А**

(Обязательное)

Схема алгоритма функции searchTimeLine()

# **ПРИЛОЖЕНИЕ Б**

(Обязательное)

Схема алгоритма функции getInfo()

# **ПРИЛОЖЕНИЕ В**

(Обязательное)

Схема алгоритма функции putInfo(Result<User> result)