**Дніпропетровський ліцей інформаційних технологій**

**при Дніпропетровському національному університеті**

**імені Олеся Гончара**

**Випускна робота**

**на тему:**

**Створення додатка клієнта на платформі Android для різних соціальних мереж**

**Виконавець:**

**ліцеїст 11-Б-1 класу**

**Долматов Ілля**

**Керівник роботи:**

**Буланий О.П. \_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дніпро**

**2016**

# ЗМІСТ

Зміст

[ЗМІСТ 2](#_Toc471937067)

[ВСТУП 5](#_Toc471937068)

[Цілі 6](#_Toc471937069)

[ТЕОРИТИЧНА ЧАСТИНА 7](#_Toc471937070)

[Загальний опис 7](#_Toc471937071)

[Актівіті 8](#_Toc471937072)

[Фрагменти 8](#_Toc471937073)

[Адаптери 9](#_Toc471937074)

[Діалогові вікна 10](#_Toc471937075)

[Моделі 10](#_Toc471937076)

[Доступ до бази даних 10](#_Toc471937077)

[ОПИС РОБОТИ 11](#_Toc471937078)

[Аудіозаписи 11](#_Toc471937079)

[Особисті повідомлення 12](#_Toc471937080)

[Новини 12](#_Toc471937081)

[ВИКОРИСТАНІ АЛГОРИТМИ 14](#_Toc471937082)

[Аудіозаписи 14](#_Toc471937083)

[Опис 14](#_Toc471937084)

[Використані бібліотеки: 14](#_Toc471937085)

[Зображення 16](#_Toc471937086)

[Особисті повідомлення 17](#_Toc471937087)

[Опис 17](#_Toc471937088)

[Використані бібліотеки 17](#_Toc471937089)

[Зображення 19](#_Toc471937090)

[Новини 19](#_Toc471937091)

[Опис 19](#_Toc471937092)

[Використані бібліотеки 20](#_Toc471937093)

[Профіль 21](#_Toc471937094)

[Опис 21](#_Toc471937095)

[Зображення 22](#_Toc471937096)

[Друзі 22](#_Toc471937097)

[Зображення 23](#_Toc471937098)

[ДОВІДНИК КОРИСТУВАЧА 24](#_Toc471937099)

[Вхід у соціальну мережу «ВКонтакте» 25](#_Toc471937100)

[Вхід у соціальну мережу «Facebook» 26](#_Toc471937101)

[Вхід до Last.fm 27](#_Toc471937102)

[Початок роботи та список новин 28](#_Toc471937103)

[Контекстне меню 29](#_Toc471937104)

[Особисті повідомлення 30](#_Toc471937105)

[Головне меню та профіль 31](#_Toc471937106)

[Друзі 32](#_Toc471937107)

[Аудіозаписи 33](#_Toc471937108)

[СТРУКТУРА ПРОГРАМИ 34](#_Toc471937109)

[Напрямки використання 38](#_Toc471937110)

[Програмно-апаратні вимоги 38](#_Toc471937111)

[Комплектація програми 38](#_Toc471937112)

[Використані програмні засоби 38](#_Toc471937113)

[ВИСНОВКИ 40](#_Toc471937114)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 41](#_Toc471937115)

# ВСТУП

У наш час майже кожна людина має свій власний обліковий запис у різноманітних соціальних мережах. І майже кожен хоча б раз на день заходить у свій обліковий запис за допомогою комп’ютера або смартфона. Зараз телефони все більше і більше займають місце комп’ютерів, тому кількість користувачів різних соціальних мереж, які переглядають свою сторінку з телефона невпинно росте, в той час як кількість користувачів, які роблять теж саме з комп’ютера падає. Цю думку підтверджують соціологічні опитування: з кожним роком все більше людей користуються смартфоном, а не комп’ютером під час переглядання своїх сторінок. Окрім того, майже кожна людина має обліковий запис не в одній соц. мережі, а в двох чи трьох. І кожна з цих мереж має свій окремий додаток-клієнт, тому користувач вимушеній постійно змінювати «вікно» на смартфоні, щоб мати змогу переглядати свою сторінку у іншій мережі. Як раз цю проблему я хочу вирішити. А саме, об’єднати декілька різних соціальних мереж в один клієнт, в якому користувач зможе переглядати свої сторінки з декількох цих мереж одночасно.

## Цілі

* Створення простого аудіоплеера
  + Отримування списка улюблених композицій з «Last.fm».
  + Можливість слухати аудіозаписи з серверу «Last.fm».
  + Можливість зміни позицію часу під час програвання композиції та зміни запису на наступний або попередній.
  + Додавання та видалення записів з улюблених.
  + Сортування композицій за різними ознаками (назва, співак, час та порядок як у звичайному клієнті).
  + Пошук інших композицій у «Last.fm».
  + Збереження інформації про аудіозапис.
* Вході або реєстрації користувача у різних соціальних мережах, таких як «ВКонтакте», «Facebook»та «Last.fm».
* Перегляд новин
  + Перегляд новин від друзів та спільнот, на які користувач підписан з різних соціальних мереж одночасно.
  + Можливість опубліковувати власні записи в одній або декількох соціальних мережах.
  + Перегляд вкладень до записів, таких як зображення та відео.
  + Фільтрація новин за тим, звідки вони з’явились.
* Створення вікна для перегляду особистих повідомлень
  + Можливість попереднього перегляду списка повідомлень, тобто ім’я користувача, який прислав повідомлення, час отримання, чи в мережі зараз цей користувач та перші декілька речень повідомлення.
  + Перегляд повного списку повідомлень
  + Надсилання повідомлень до користувача
* Локалізація, тобто переклад додатку на дві мови, а саме російську та англійську.
* Створення візуальної, інтуїтивно зрозумілої інструкції для користувачів з поясненням які елементи що роблять.
* Перегляд власного профіля та записів зі стіни цього профіля.
* Перегляд списку друзів
  + Пошук нових друзів за ім’ям
  + Перегляд профіля будь-яких друзів з соціальної мережі «ВКонтакте».
  + Можливість додатки знайдених користувачів до друзів.

# ТЕОРИТИЧНА ЧАСТИНА

## Загальний опис

Більшість значних соціальних мереж має так званий API. Це програмний додаток для розробників інших додатків, які хочуть явимось чином інтегрувати с цією мережою. В роботі було використано VK API, Facebook Graph API та Last.fm API, які на даний момент є доступними до завантаження та використання для всіх. Більшість соціальних мереж має схожу архітектуру, а саме: профіль користувача, список друзів, список груп, список новин від друзів та груп, на які користувач підписан. Також більшість соціальних мереж має засіб надсилання та отримання повідомлень від інших користувачів або бесід. Оскільки більшість архітектури схожа, то можна вигадати спосіб об’єднати декілька мереж в один додаток.

Для виконання цієї роботи було обрано середовище програмування Android Studio, яке є найпопулярнішим та безкоштовним для всіх бажаючих, а також підтримує всі версії Android, оскільки було створено самим Google. Мова середовища Java також являє собою одну з найпопулярніших мов у світі, оскільки є об’єктно-орієнтованою. Для виконання цілей було використано декілька сторонніх API від соціальних мереж та ряд алгоритмів для взаємодії з ними за допомогою запитів та слухачів.

Загалом проект розбитий на декілька блоків, кожен з яких має свій фрагмент, або декілька, адаптер, який дозволяє створювати власний тип списку, модель для збереження інформації та запит до бази даних. Окрім цих елементів деякі блоки мають актівіті, або якусь унікальну для даного проекта архітектуру. Взаємодію блоків між собою можна побачити на схемі нижче.

Нажаль, «ВКонтакте» не так давно припинили поставляти можливість роботи з аудіозаписами, через що блок з аудіозаписами був змінений. Тепер користувач має можливість побачити список улюблених композицій з сайту Last.fm, додавати в цей список нові та видаляти старі аудіозаписи. Через це досить велика частина коду біла видалена, оскільки змінена механіка аудіоплеєра.

## Актівіті

Актівіті являє собою елемент на кшталт вікна, як у Windows, але для Android. Загалом у проекті існую 5 різних актівіті. Кожне актівіті виконую одну з двох функцій: функцію вікна або функцію контейнера для фрагментів. Декілька актівіті, а не одне для всього проекта зроблене задля того, щоб не перевантажувати кодом один клас, а розподілити пізні функції з різних блоків на декілька актівіті. В цих класах в основному виконується робота створення фрагментів, та створення і обробки подій меню. Окрім цього деякі актівіті не мають фрагментів, а самі виконують роль фрагмента, оскільки мають лише один вигляд і не потребують зміни контенту. Актівіті основного меню та аудіоплеера мають додатково ряд методів, які дозволяють відкривати інші актівіті, або отримувати фрагменти, які в них знаходяться. Окрім цього вікно аудіоплеера має методи для пошуку пісень у мережі. Всі актівіті мають батька CoreActivity, який реалізує спільний для всіх метод під час натискання кнопки назад та у разі потреби можна змінювати дію цієї кнопки.

## Фрагменти

Фрагмент це елемент, який накладається на актівіті у разі потреби. У проекті основну роль грають саме фрагменти, оскільки саме в них відбувається запит до сервера та обробка відповіді. Всього є 9 різних фрагментів, кожен з яких або реалізую зв’язок з сервером, або є контейнером для інших фрагментів у разі потреби табуляції фрагментів. Всі фрагменти, які роблять запити до сервера мають в собі внутрішній клас Listener, в якому і відбувається обробка запиту та його перетворення на об’єкт, зрозумілий цій програмі. Всі «слухачі», тобто внутрішні класи Listener мають батька CoreListener, який має потрібні для всіх поля та одразу наслідую інтерфейс Runnable, який дозволяє цьому класу працювати окремим потоком. Деякі фрагменти, як і деякі актівіті, виконують роль контейнерів, тобто не мають власного інтерфейсу, а лише деяке функціональне значення. Фрагменти активно приймають участь у взаємодії з іншими класами, оскільки зберігають в собі багато інформації про різні класи та інформацію з самих соціальних мереж. Маю батька CoreFragment, за допомогою якого, як і у актівіті, можна змінювати подію під час натискання кнопки назад. Більшість фрагментів мають інтерфейс у вигляді списка. У цей список інформація заноситься за допомогою так званих адаптерів, яві дозволяють робити власні типи елементів списку.

## Адаптери

Адаптери це класи, які відповідають за перетворення інформації з моделі деякого типу в зрозумілий користувачу вигляд. Кожен адаптер має внутрішній клас, який допомагає при збережені інформації про контейнери, які містять в собі текст, зображення та кнопки. Адаптери також мають спільного батька CoreAdapter, який не тільки реалізує однакові для всіх методи з абстрактного класу, який робить адаптер адаптером, а також додає ряд нових методів, за допомогою яких можна миттєво змінювати данні, які відображаються користувачу та які беруть участь у програмі. А саме: додавання нового елементу в кінець списку, видалення елементу за індексом або за моделлю цього елемента, очистка всіх елементів або заміна існуючих на якийсь новий список та метод, який дозволяє забрати з адаптеру список, з яким він працює. CoreAdapter має можливість «налаштовуватись» на деяку модель, тобто всі методи на поля будуть саме того типу, який потрібно в даному адаптері, наприклад в адаптері з новинами всі методи будуть приймати та повертати модель новини, а не якусь іншу.

## Діалогові вікна

В деяких блоках проекта зустрічається потрібність у діалогових вікнах, за допомогою яких користувачу дається доступ до додаткових функцій додатка. Такі діалоги не містять в собі жодної корисної інформації, а лише є запитами та обробкою відповіді користувача, яка надалі передається в один з фрагментів для подальшої обробки.

## Моделі

Моделі не виконують жодної логіки, але є невід’ємними, оскільки зберігають в собі інформацію про різні типи елементів з мережі. На відміні від інших головних елементів, вони не мають спільного батька, а лише реалізують інтерфейс Parcelable, який дозволяє передавати цей об’єкт між актівіті та між фрагментами. Всі моделі є повністю інкапсульованими, тобто доступ до цього поля відбувається через спеціальний метод. Всі моделі мають конструктор, за допомогою якого і створюються. Проблема створення спільного батька полягає в тому, що кожна модель має унікальні поля, хоча всі і працюють за однаковим принципом.

## Доступ до бази даних

Окрім запитів до серверу програма використовую запити до бази даних, в якій зберігаються моделі. За допомогою класів, які мають доступ до бази, фрагменти можуть записувати в пам’яті різні моделі, які потім можна зчитувати звідти за допомогою цих самих класів. Всі класи цього типу мають спільного батька DBCoreService. В цьому абстрактному класі знову реалізовані всі спільні методи та спільні поля. Як і CoreAdapter має можливість «налаштовуватись» на деяку потрібну для даного класу модель.

# ОПИС РОБОТИ

Загалом проект ділиться на 8 частин: профіль, новини, предперегляд особистих повідомлень, перегляд історії особистих повідомлень, список друзів, список аудіо записів, вхід в різні соціальні мережі та створення нового повідомлення на стіні користувача.

## Аудіозаписи

Для відображення аудіозаписів використовується окремий актівіті, який має два фрагменти: один статичний, який користувач бачить постійно (список) та фрагмент, який дає додаткову інформацію про аудіозапис. Список записів зберігається в об’єкті типу List. Для кожного елемента ListView перевизначений метод натискання на елемент, за допомогою чого можна прочитати додаткову інформацію про деякий запис, а також змінити граючий запис на наступний або попередній. Кожний елемент в ListView має в собі кнопку та три надписи, а саме, назва, співак та час програвання даного запису. При натисканні на кнопку на будь-якому аудіозаписі за допомогою класу android.media.MediaPlayer починається програвання запису. При повторному натисканні запис зупиняється та може бути знову відновлений при наступному натисканні на тому ж місці, де його до цього зупинили. З кнопкою запису, який зараз грає, тісно зв’язана кнопка в окремому фрагменті, яка також відповідає за вмикання/вимикання запису. Для зберігання інформації про запис було створено окремий клас Song. Також є окреме Activity, за допомогою якого можна змінювати налаштування еквалайзеру для даного запису. Реалізована сортування за різними критеріями, а саме: за назвою, за співаком, за часом програвання та у послідовності, яка була визначена користувачем у соціальній мережі «ВКонтакте». За допомогою кнопки угорі актівіті можна шукати за назвою або автором потрібний аудіозапис та додати його до свого плей-листа. З плей-листа в свою чергу можна видаляти аудіозаписи. Список аудіозаписів зберігається в окремій базі даних.

## Особисті повідомлення

Особисті повідомлення мають окремий фрагмент та актівіті. Фрагмент знаходиться в головному меню справа від новин, а актівіті відчиняється у разі потреби відкрити історію повідомлень з користувачем. Фрагмент та актівіті мають лише невеликий зв’язок, який полягає в передаванні моделі при натисканні на якийсь діалог. Обидва елементи використовують список з власним адаптером. Список останніх повідомлень, які знаходяться у фрагменті зберігаються в окремій таблиці у базі даних, а історія повідомлень з кожним користувачем окремо завантажується кожного разу с сервера знову. У предперегляді відображається малюнок аватара користувача, його ім’я та прізвище, останнє повідомлення, чи в мережі зараз користувач та час останнього повідомлення. При довгому натисканні з’являється діалогове вікно, за допомогою якого можна видалити історію повідомлень з обраним користувачем. При натисканні по одному з таких елементів з’являється вікно, де можна побачити останні повідомлення від цього користувача, а при перемотці вниз ці повідомлення будуть завантажуватись все далі. За допомогою спеціального поля EditText користувач може ввести нове повідомлення та відіслати його. У такому режимі користувач бачить список з елементів, які містять в собі аватар співрозмовника, його ім’я, текст повідомлення та дату цього повідомлення.

## Новини

Новини мають власний фрагмент, який можна побачити одразу після переходу з початкового актівіті. Новини відображають записи, які біли опубліковані користувачем, групами та іншими користувачами, на яких підписана людина. Записи збираються одразу з двох соціальних мереж, а саме «ВКонтакте» та Facebook. Кожний запис відображає назву групи або ім’я користувача, аватар, час створення запису, текст, якщо він є, одне чи декілька зображень та три кнопки. Перша «Мені подобається», друга «Опублікувати на моїй сторінці» та третя «Коментарі». Коментарі в свою чергу відкривають окреме актівіті, де можна прочитати, що написали інші користувачі та написати власний коментар. За допомогою контекстного меню можна опублікувати власний запис на одній чи декількох сторінках соціальних мереж у вигляді тексту.

# ВИКОРИСТАНІ АЛГОРИТМИ

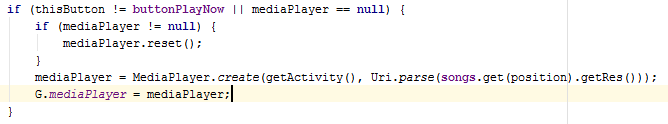
Основним елементом у роботі є ListView, за допомого якого на екран виводяться всі записи, повідомлення та аудіозаписи користувача.

## Аудіозаписи

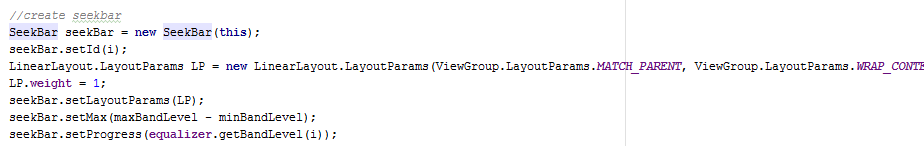
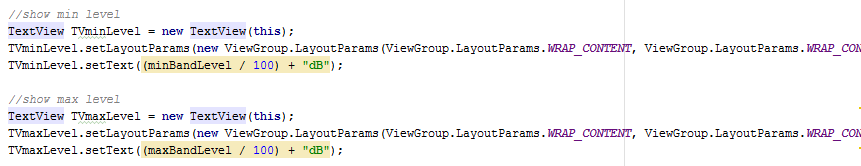
### Опис

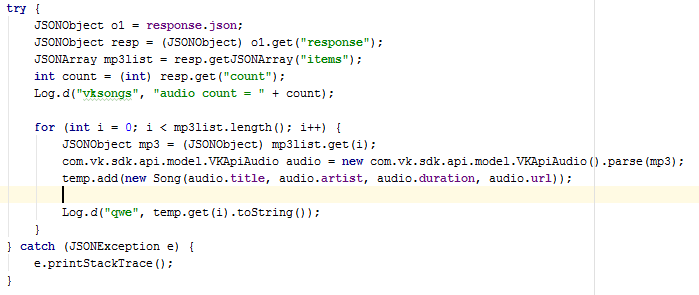
Для відображення аудіозаписів використовується окремий Activity, який має два фрагменти: один статичний, який користувач бачить постійно (список). Список записів зберігається в об’єкті типу List. Для кожного елемента ListView перевизначений метод натискання на елемент, за допомогою чого можна перейти на сайт з даним записом. Кожний елемент в ListView має в собі кнопку та три надписи, а саме, назва, співак та час програвання даного запису. При натисканні на кнопку на будь-якому аудіозаписі відбувається перехід на сайт цього запису, де його можна прослухати. Для зберігання інформації про запис було створено окремий клас Song.

### Використані бібліотеки:

* android.media та android.net.Uri – використовуються для програвання запису та перетворення строки на посилання на пісню. Використовуються парою.
* android.view – використовуються для присвоєння фрагменту потрібний Layout.
* android.widget – використовуються для звернення до різних елементів, які присутні у Layout’і даного фрагменту.

ListView   
ImageButton

  
Toast  
Клас Toast використовується для відображення помилок, які виникають під час звернення в мережу.  
BaseAdapter   
  
SeekBar   
Snipper  
  
TextView  


* org.json – є допоміжним пакетом класів при обробці даних, отриманих слухачем.Під обробкою запиту розуміється перетворення строки, яку отримую «слухач» в окремі дані, які потім зберігаються у класі Song, а елементи цього списку в класі List<Song>. Потім саме цей список відображається користувачу.
* java.util – пакет класів, який дає можливість використовувати списки та масиви.
* android.app.Activity – клас, який зберігає відображає всі інші елементи, наприклад, повідомлення для користувача з вибором.

Окрім вищезазначених класів використовується ряд власних класів, які будуть описані потім.

### Зображення

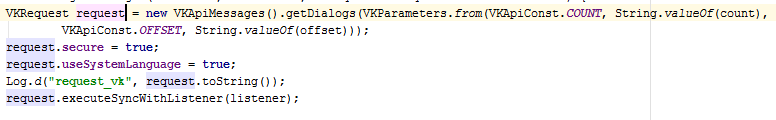
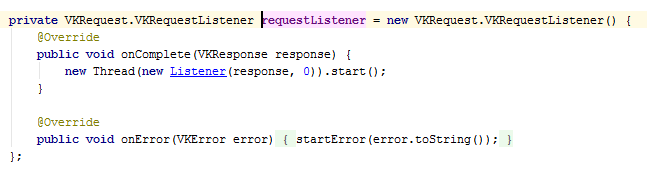
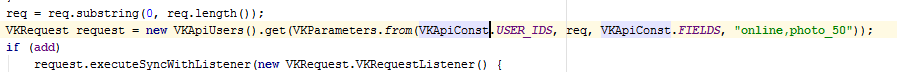


## Особисті повідомлення

### Опис

При переході до вкладки з особистими повідомленнями користувач бачить ListView з елементами, які відображають останні повідомлення у діалогах з різними користувачами. Реалізовано за допомогою власного адаптера MessagePreviewAdapter та MessageHistoryAdapter. Відображається аватар користувача чи бесіди, ім’я користувача чи назва бесіди, останнє повідомлення, час отримання чи відправки цього повідомлення. Також, за допомогою зеленого фону відображаються повідомлення, які не були прочитані. Елементи списку мають перевизначений метод натискання. При натисканні по елементу списка створюється нове актівіті, в якому відображаються історія з останніх 20 повідомлень з користувачем, якого обрав користувач та можливістю написати та відправити нове повідомлення. На зразок предперегляду за допомогою зеленого фону відображається чи прочитане дане повідомлення. Повідомлення складається з аватара користувача, який написав повідомлення, його ім’я, самого повідомлення та часу відправки чи отримання. Написання повідомлення відбувається за допомогою EditText, а відправка за допомогою ImageButton. Кожні 5 секунд відбувається автоматичне оновлення історії повідомлень, що забезпечує можливість швидкого обміну повідомленнями за допомогою соціальної мережі «ВКонтакте».

### Використані бібліотеки

* android.content.Intent – клас, який дозволяє здійснювати перехід між основним актівіті та актівіті з історією повідомлень.
* android.graphics.Bitmap та android.graphics.BitmapFactory – здійснюють завантаження з інтернету та зберігання зображень.
* android.view.View – використовується для обробки основних подій фрагменту для предперегляду повідомлень.
* com.vk.sdk.api (класи з окремого додатку від соціальної мережі «ВКонтакте») – пакет класів, який відповідає за запити, отримання та часткову обробку даних з соціальної мережі «ВКонтакте».  
  Створення запиту:   
  «Слухач» запиту:   
  Константи та параметри 

Всі інші класи були описані вище загалом, а окремі їх частини не вимагають докладного опису. Власні класи також будуть описані нижче.

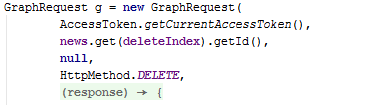
### Зображення

## Новини

### Опис

Новини уявляють з себе список ListView з власним адаптером NewsAdapter, відображаються списки новин від користувачів та груп, на які підписан користувач та, власне, записи самого ж користувача. Відображаються текст новини, картинки та відео. Кожна новина має такі елементи: аватар користувача чи групи, якими була опублікована новина, назва групи чи ім’я користувача, позначка звідки цей запис отримано, це може бути «ВКонтакте» або «Facebook», дата публікації, кнопки «Мені подобається», яка додає дану новини в список улюблених, кількість людей, які також додали цей запис до списку улюблених, кнопка «Опублікувати в мене», яка публіку запис на стіні користувача та кількість інших користувачів, які це зробили, кнопка коментарів. Окрім цього, залежно від змісту новини, може бути текст, одне або декілька зображень (до 10 зображень) та відео. При довгому натиску по новині відображається невелике повідомлення, при натисканні по якому можна видалити новину з списку, який Ви бачите, тобто додати до списку небажаних новин. Окрім цього, за допомогою контекстного меню Android можна обрати пункт створення нового запису, після чого стане доступне актівіті для створення запису. У ньому можна написати будь-який текст, який підтримує телефон та обрати за допомогою CheckBox в яку соціальну мережу потрібно опублікувати цей запис. Після натиснення кнопки «Викласти» запис буде опубліковано в тих соціальних мережах, які обрав користувач за допомогою CheckBox`ів.

### Використані бібліотеки

* com.facebook – пакет класів, які дозволяють родити запити до соціальної мережі «Facebook» та обробляти відповіді з сервера.  
* java.util – пакет класів, який окрім можливість використовувати списки та масиви надає методи для виведення власної інформації для розробників.
* java.io.IOException – клас, який забезпечую обробку помилки програмою, не заважаючи роботі інших частин. Тобто, при виникненні помилки в новинах, не вимикається весь додаток, а відбувається обробка помилки в даній частині.
* java.text.ParseException та java.text.SimpleDateFormat – другий клас дозволяє перетворювати дату будь-якого формату, вказавши цей формат, перетворити в дату, зрозумілу комп’ютеру. Перший клас обробляє помилку, якщо дата була не того формату, який вказано у другому класі. 
* java.net.URL – забезпечує доступ до посилання на картинку, тобто перетворю текст посилання на гіперпосилання, зрозуміле комп’ютеру.

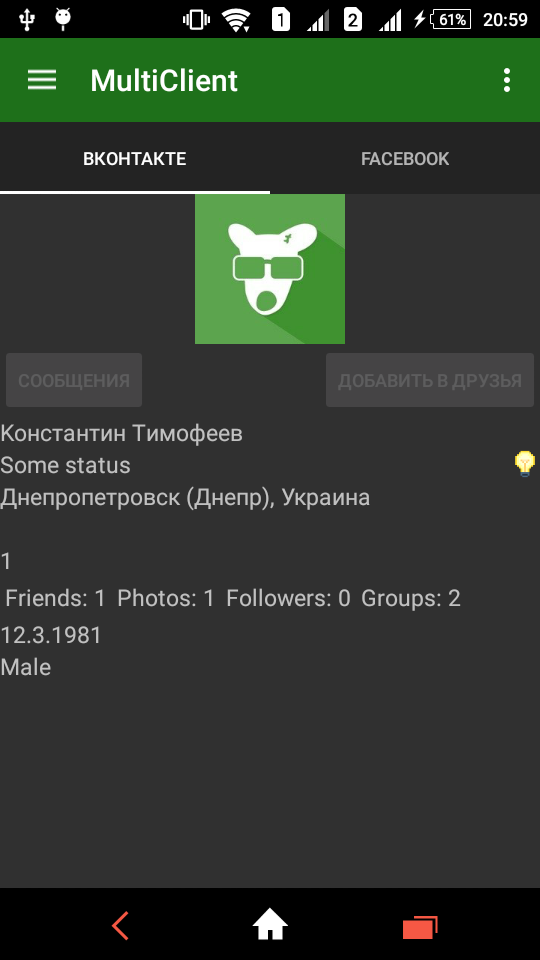
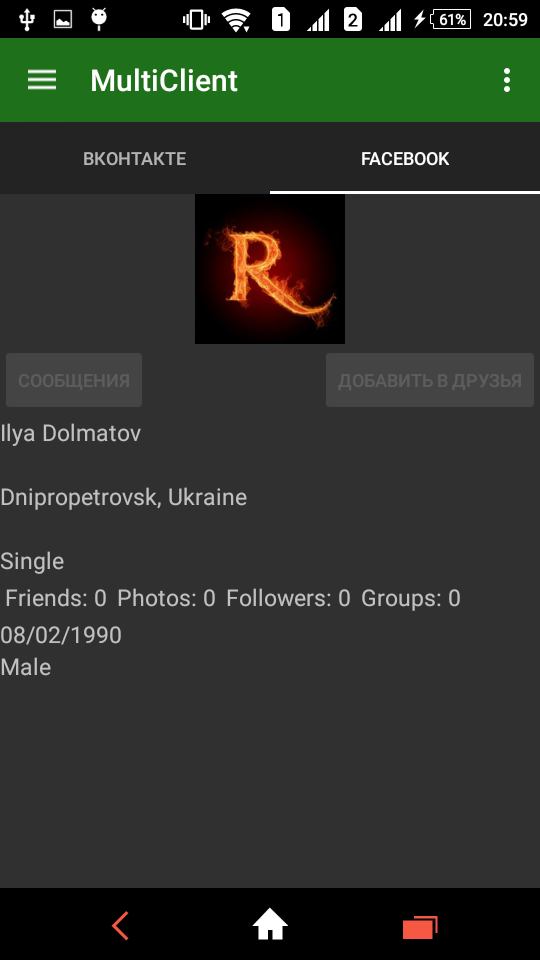
## Профіль

### Опис

За допомогою головного меню можна перейти до власного профілю. Власний профіль уявляє з себе фрагмент з двома вкладками, а сам профіль «ВКонтакте» та «Facebook». У кожній вкладці відображається основна інформація, яку користувач виклав про себе в кожній соціальній мережі, до якої він надав додатку доступ. Сторінка профіля складається з фотографії користувача, тобто аватара, його ім’я, статусу, місця народження чи проживання, залежить від користувача, інформація про навчальний заклад, статус відношень, дата народження, стать, чи онлайн користувач (якщо так, то э зображення лампочки біля статусу) та кількість друзів, фотографій, груп, на які підписан користувач, та кількість людей, які підписані на користувача, але не є йому друзями. Окрім того є список записів, які були опубліковані користувачем за весь час. Цей список майже не відмінний від звичайних новин за реалізацією, тож розглядатися окремо не буде. Всі написи відображаються за допомогою TextView, а зображення аватара та лампочка, яка повідомляє про онлайн містяться у ImageView.

Нових використаних бібліотек чи пакетів класів немає, тому розділ про використані бібліотеки опускається.

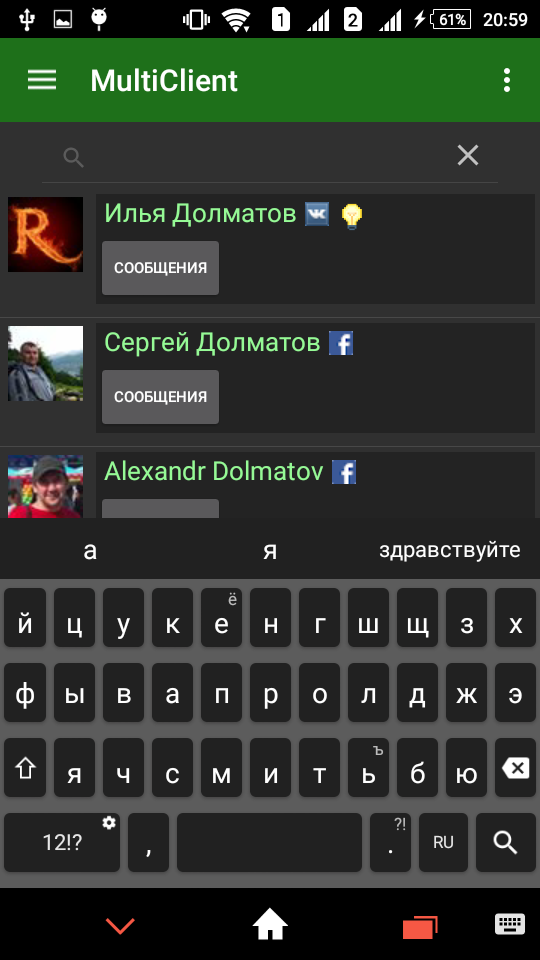
### Зображення



### Друзі

Друзі – це список користувачів, які мають статус друзів з користувачем, який використовує додаток. Реалізовано за допомогою ListView, має власний адаптер FriendsAdapter. Кожний елемент списку відображає аватар іншого користувача, його ім’я, позначку з якої соціальної мережі цей користувач, позначка, чи онлайн користувач та кнопка переходу до особистих повідомлень з цим користувачем. При натисканні по користувачам з соціальної мережі «ВКонтакте», не по кнопці, з’явиться нове актівіті. Це актівіті дуже схоже на фрагмент з профілем користувача, але є дві допоміжні кнопки, а саме перехід до історії особистих повідомлень з цим користувачем та, якцо користувач не є другом, можливість одати його до друзів. Окрім цього, зверху є строчка пошуку. Поки користувач вводить в неї ім’я деякого іншого користувача, додаток шукає 20 користувачів, які підходять під текст, який є у пошуку. Пошук відбувається лише у мережі «ВКонтакте», оскільки «Facebook» можливості пошуку друзів для зовнішніх розробників не надає. Відображення користувачів, яких знайшов пошук так саме, як і відображення друзів, а при натисканні відбувається та сама дія.

### Зображення



# ДОВІДНИК КОРИСТУВАЧА

Довідник має стиль інструкції з картинок. Кожна сторінка має деякий зміст, підписаний вгорі, а за допомогою умовних позначень та блоків з коментарями пояснюється як користуватися додатком.

Умовні позначення:

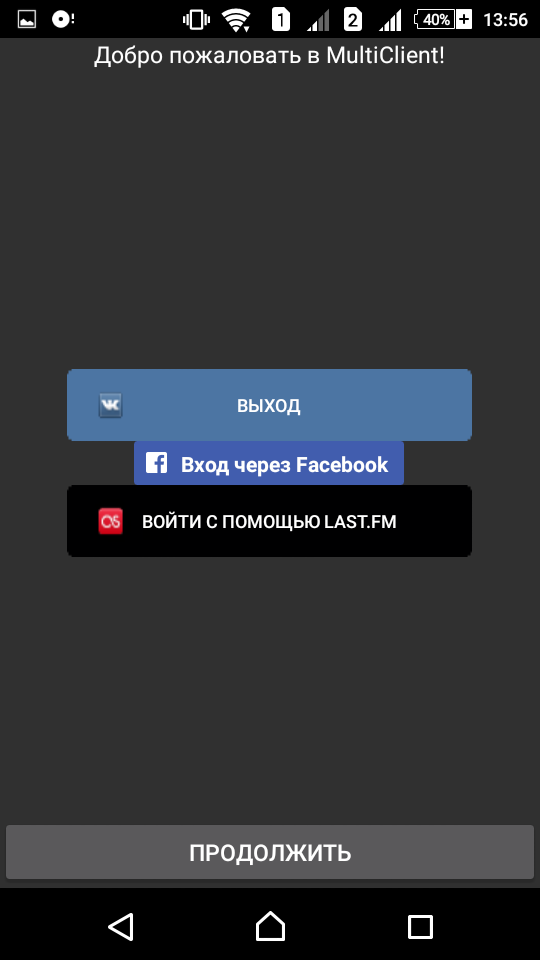
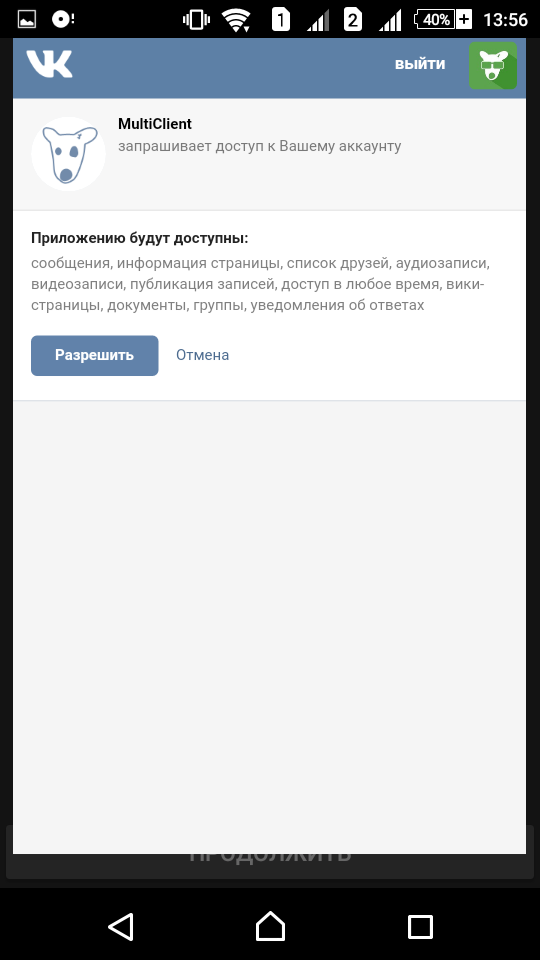
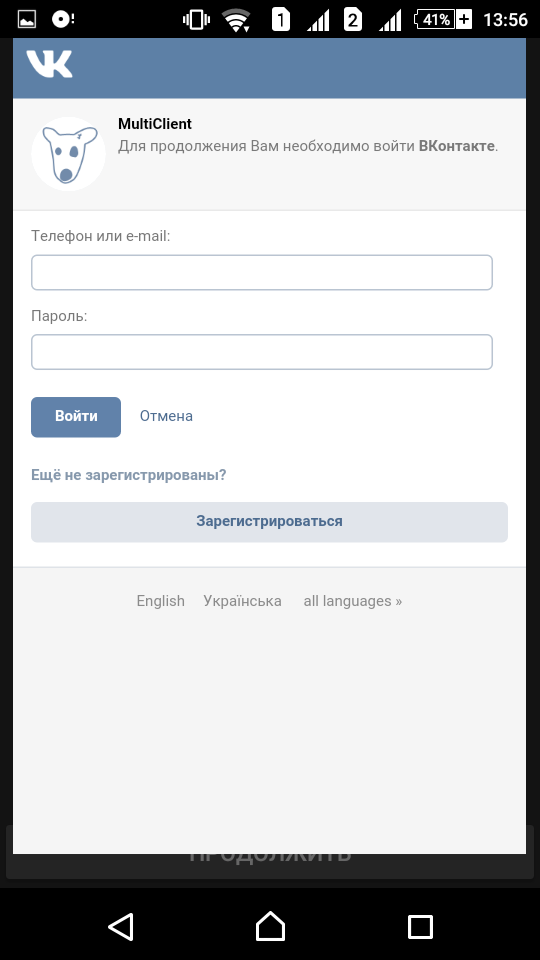
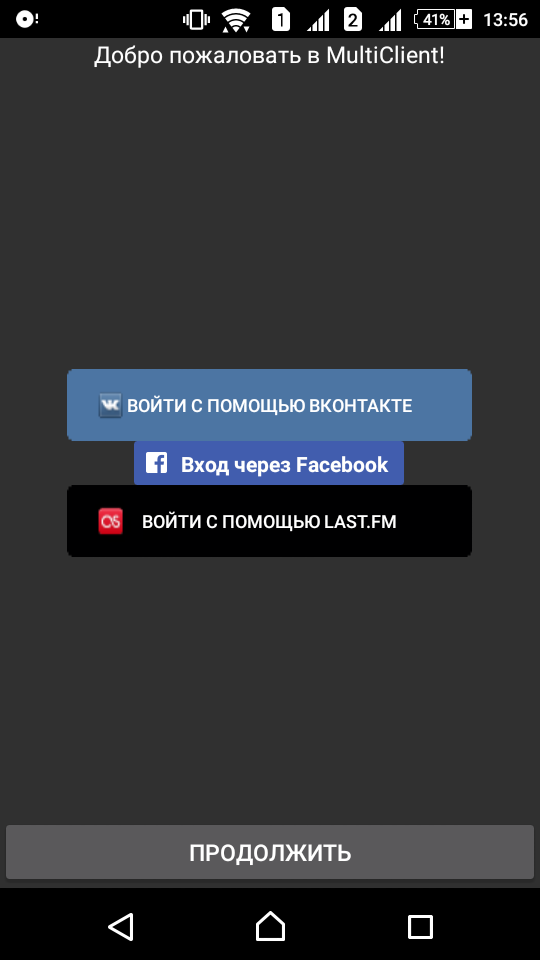


При натисканні у зоні прямокутника, відбувається перехід до стадії, на яку веде лінія від прямокутника.



При проводженні пальцем за стрілкою, відбувається перехід до стадії, на яку веде лінія від кола. Тобто, у даному прикладі при проводженні пальцем вправо відбудеться перехід до стадії, зображеної ліворуч від даної.

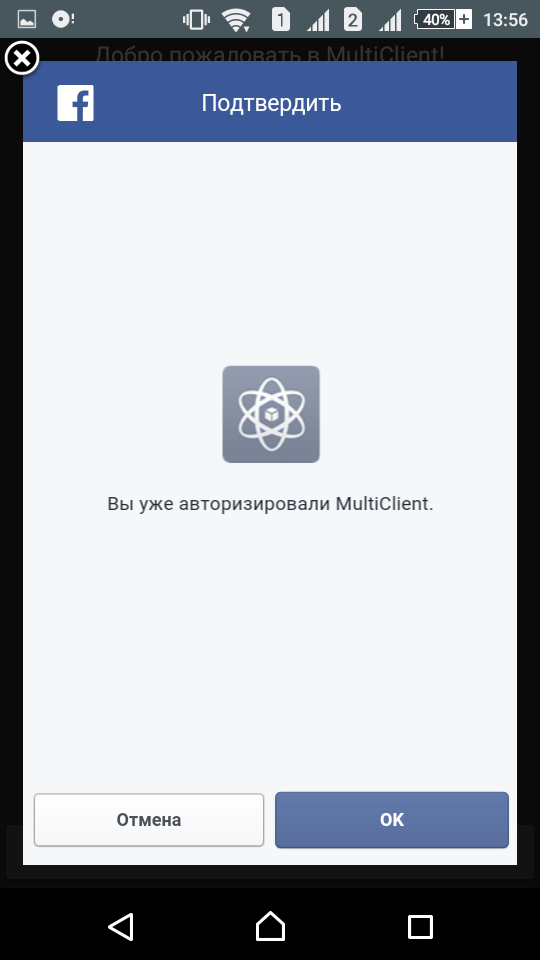
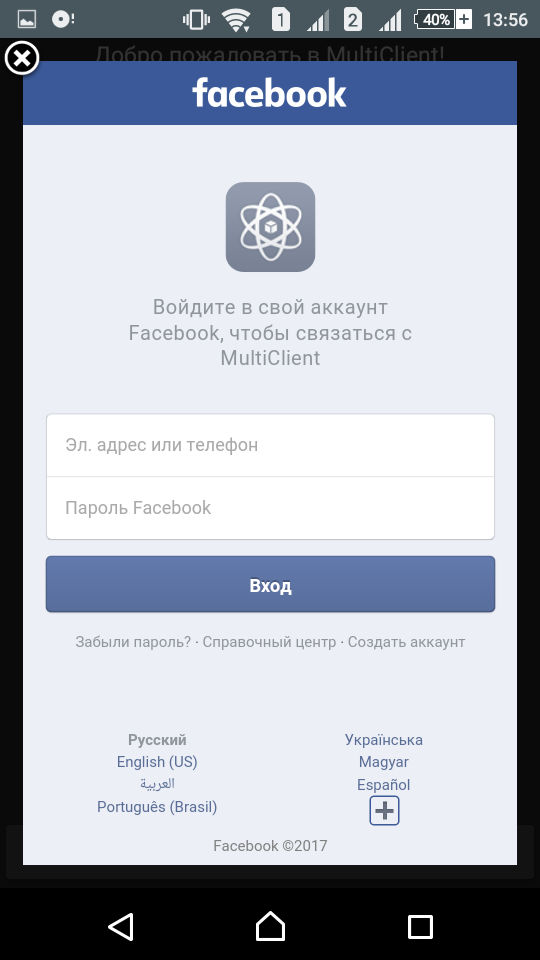
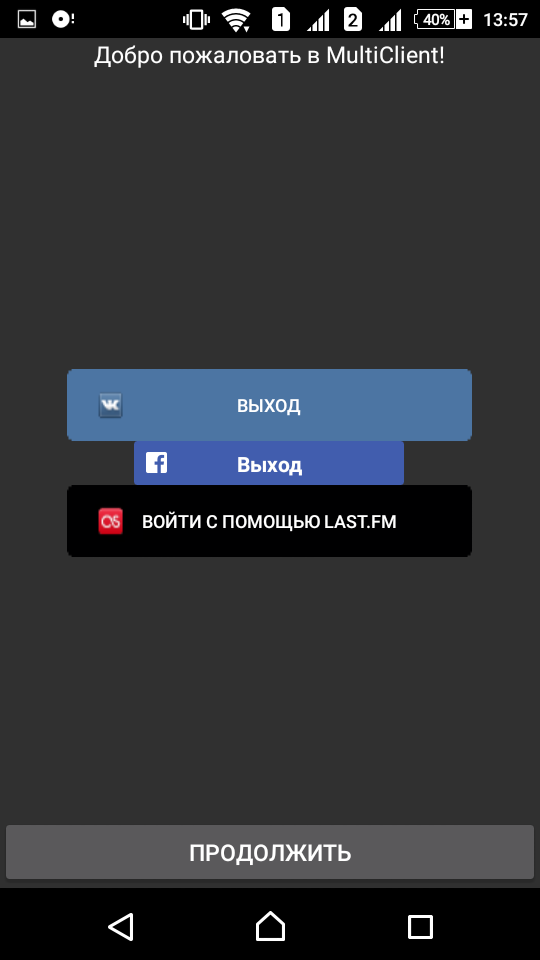
## Вхід у соціальну мережу «ВКонтакте»



Після вводу даних про аккаунт користувач може продовжити, натиснувши на кнопку «Войти» або відмовитись, натиснувши на кнопку «Отмена»

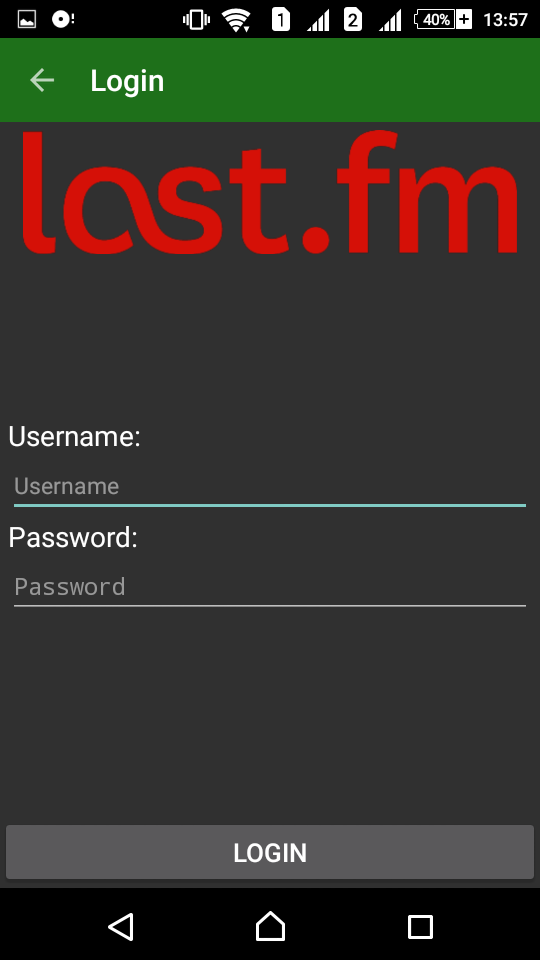
Для продовження використання додатка користувач має дати згоду на використання власних даних натиснувши на кнопку «Разрешить» або відмовитись, натиснувши на кнопку «Отмена»

## C:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-13-56-34.pngВхід у соціальну мережу «Facebook»



## C:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-13-57-12.pngВхід до Last.fm

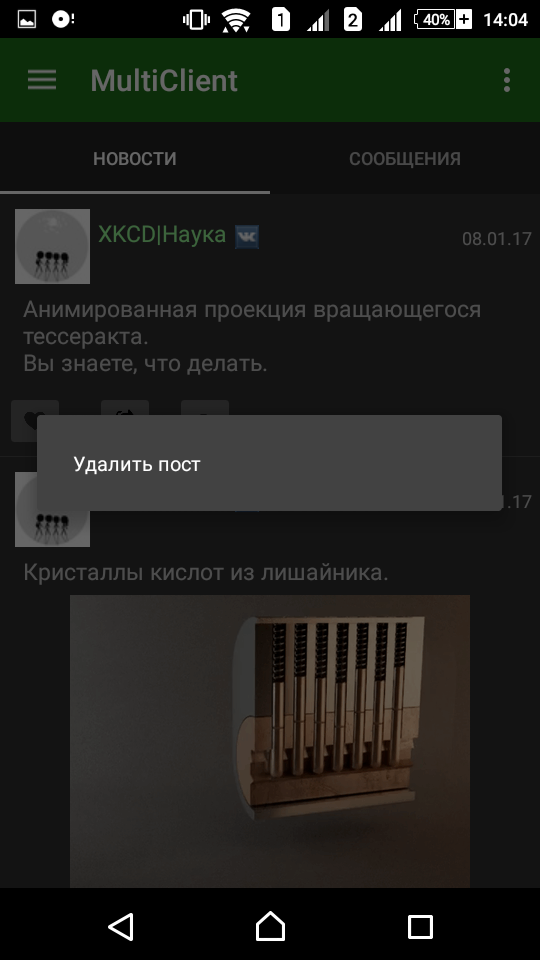
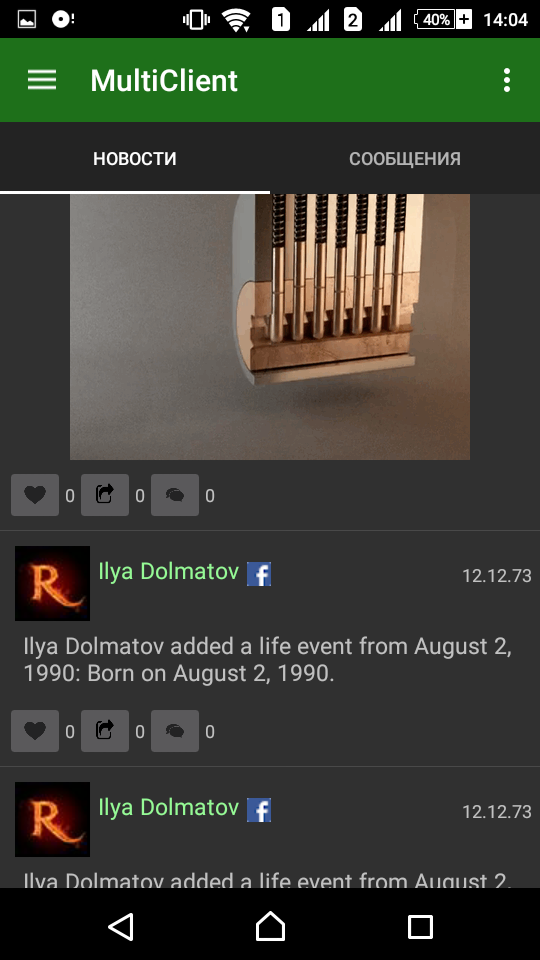
У випадку з Last.fm підтвердження доступу не потрібне, оскільки все, що надає цей сайт, це музика.



## C:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-14-04-04.pngC:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-14-03-45.pngПочаток роботи та список новин

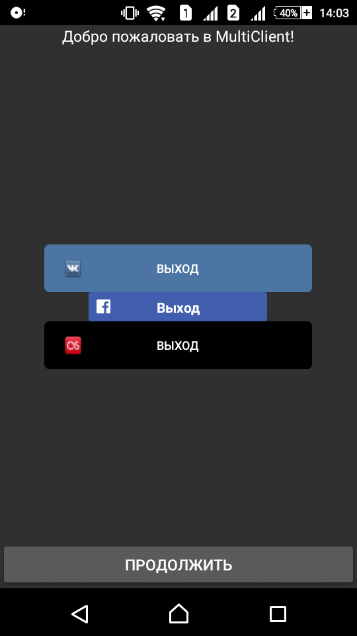
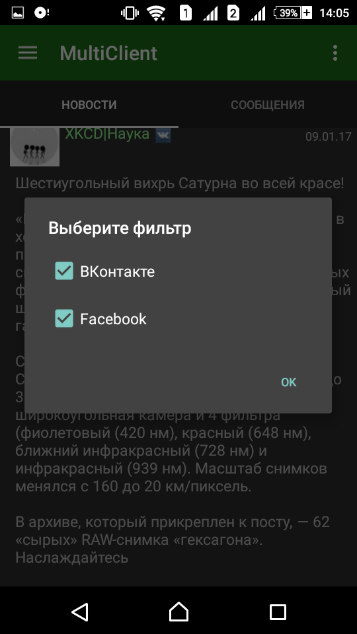
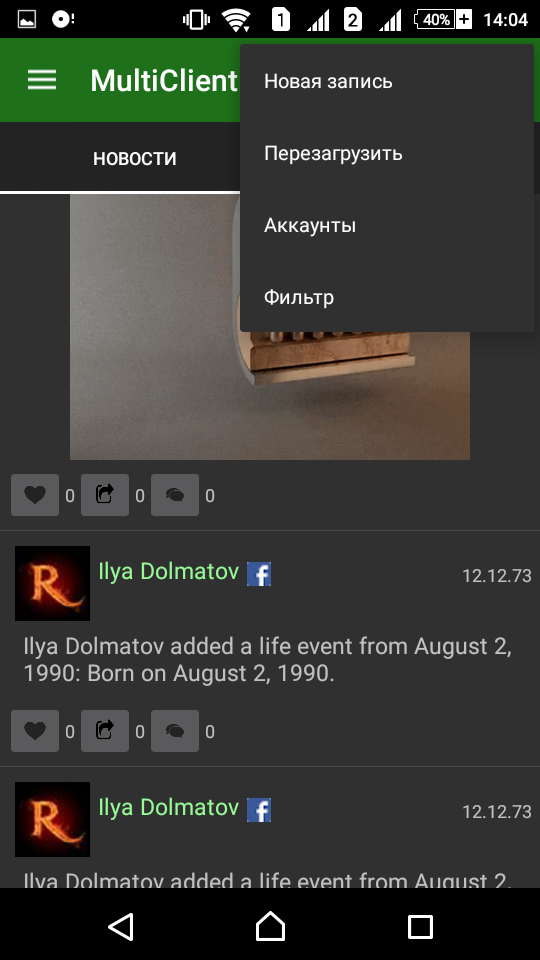
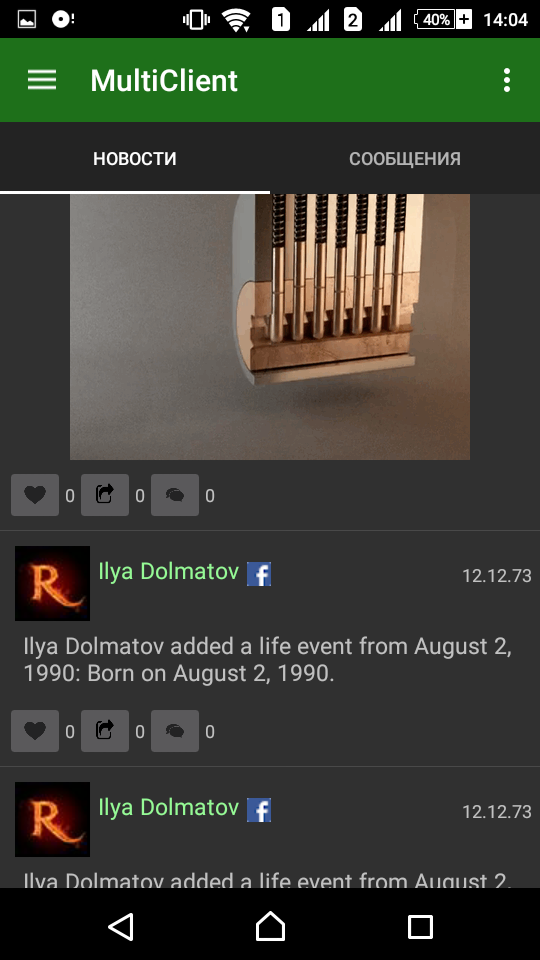
При пролистувані можна побачити інші новини, які були викладені раніш з різних соціальних мереж.

При довгому натисканні по новині з’являється доступ до її видалення, тобто, можна додати новину до небажаних.



Після переходу з початкового екрану користувач бачить головне меню та список новин.

## Контекстне меню



Обравши пункт «Перезагрузить» користувач завантажує всі новини знову, в тому числі й нові.

Початковий екран додатку.

Можливість фільтрувати новини: показуються лише ті, які обрані.

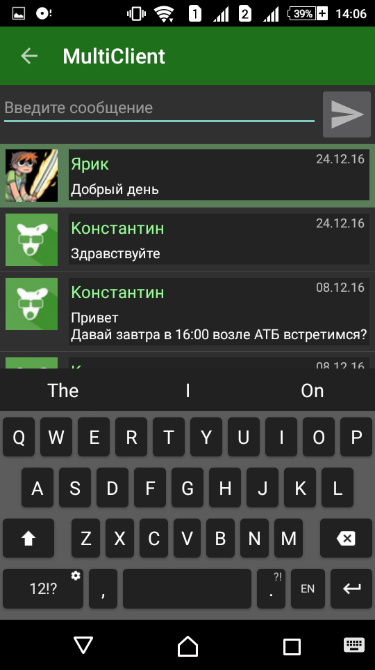
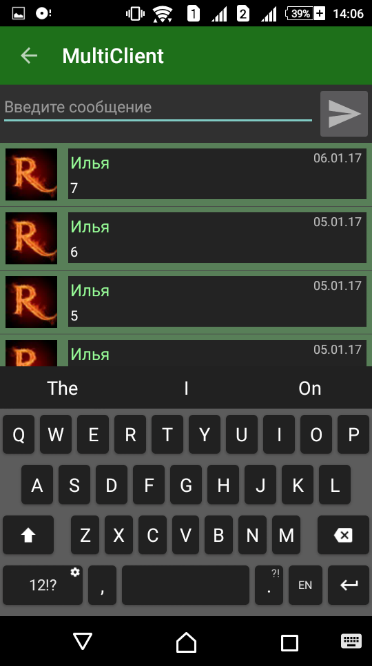
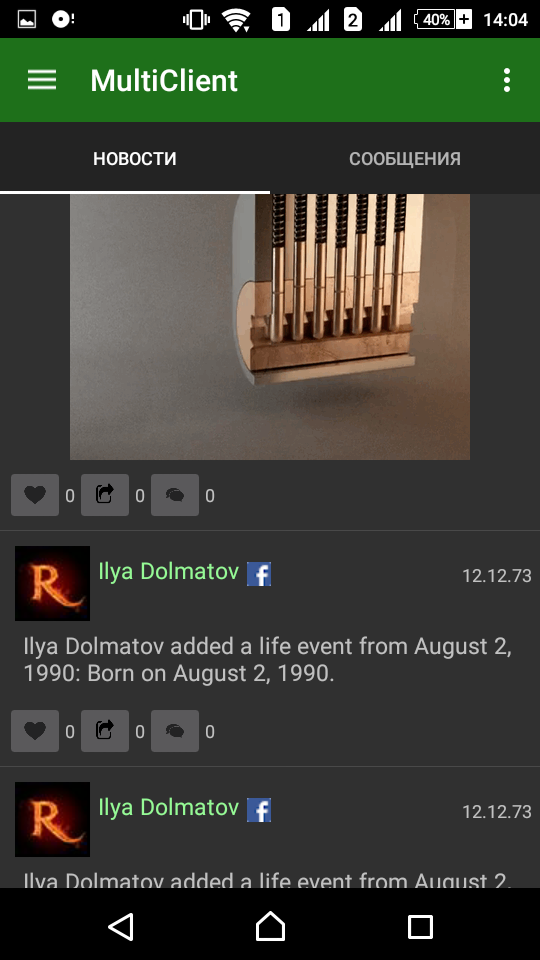
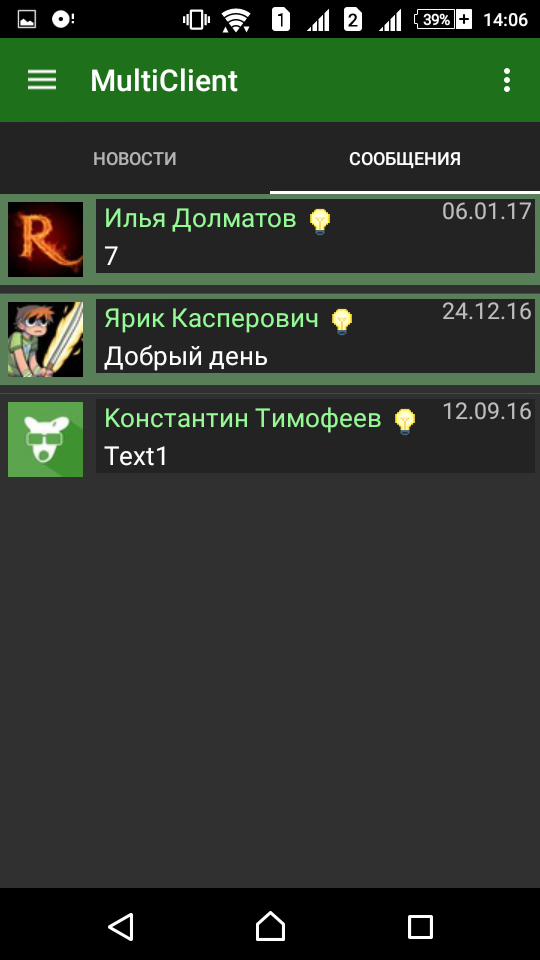
Створення нового запису на власній сторінці та вибір куди його публікувати.

## C:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-11-20-48-30.pngОсобисті повідомлення

При довгому натисканні по елементу зі списку діалогів, з’явиться можливість видалити історію повідомлень з обраним користувачем.

При натисканні по будь-якому діалогу, з’являється можливість написати нове повідомлення та прочитати більше повідомлень, які були раніше.

Перейти до особистих повідомлень можна декількома способами: натиснути на кнопку вгорі «Сообщения» або провести пальцем справа наліво Перехід назад здійснюється аналогічно.



## C:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-14-09-13.pngC:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-14-09-07.pngC:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-14-08-48.pngC:\Users\Ilya\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_2017-01-09-14-04-42.pngГоловне меню та профіль

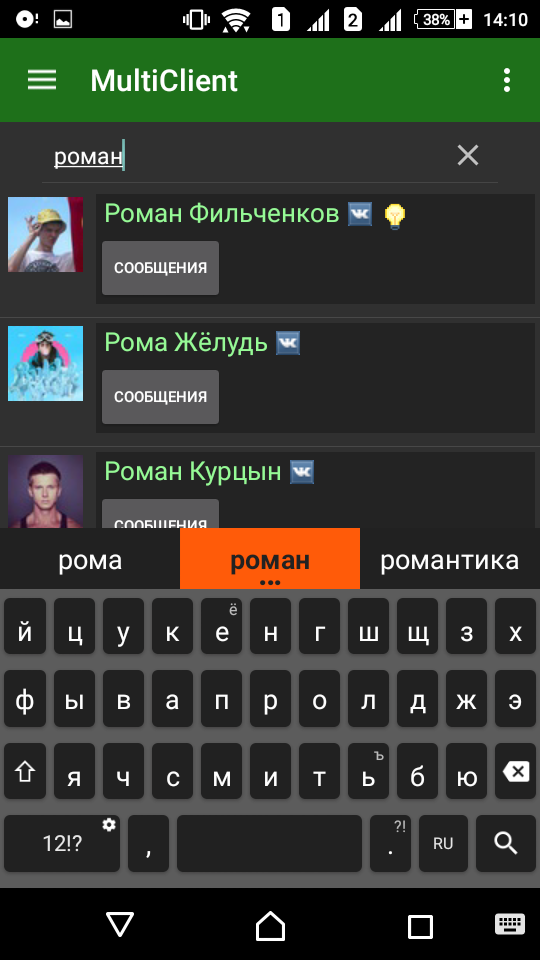
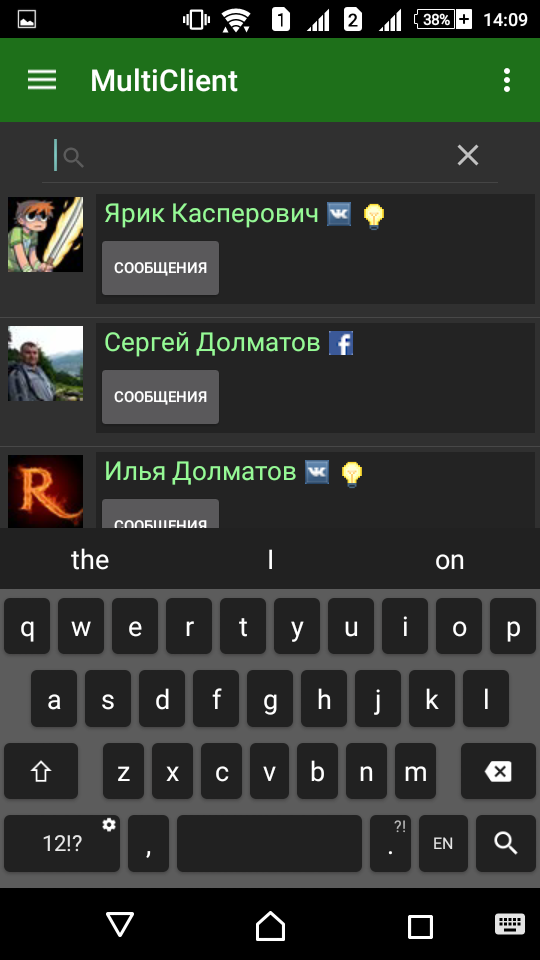
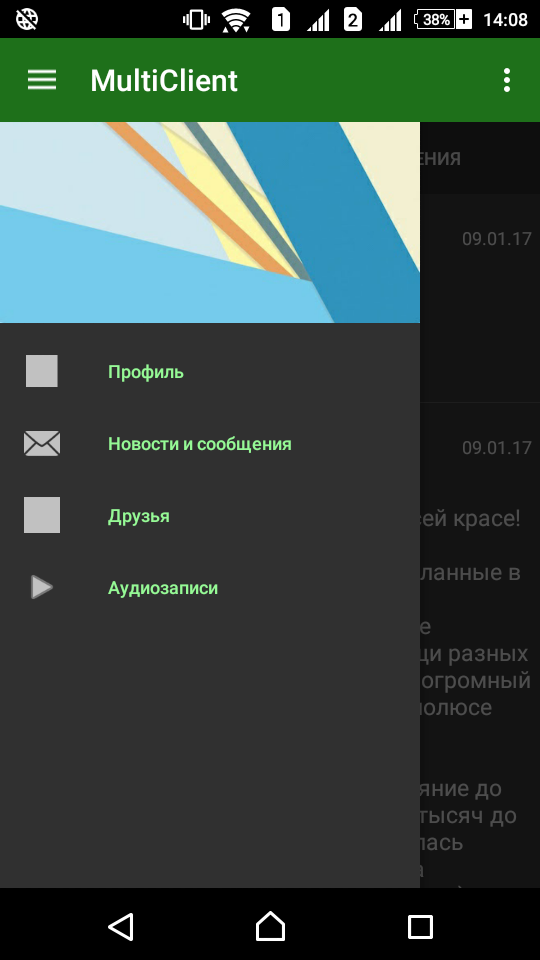
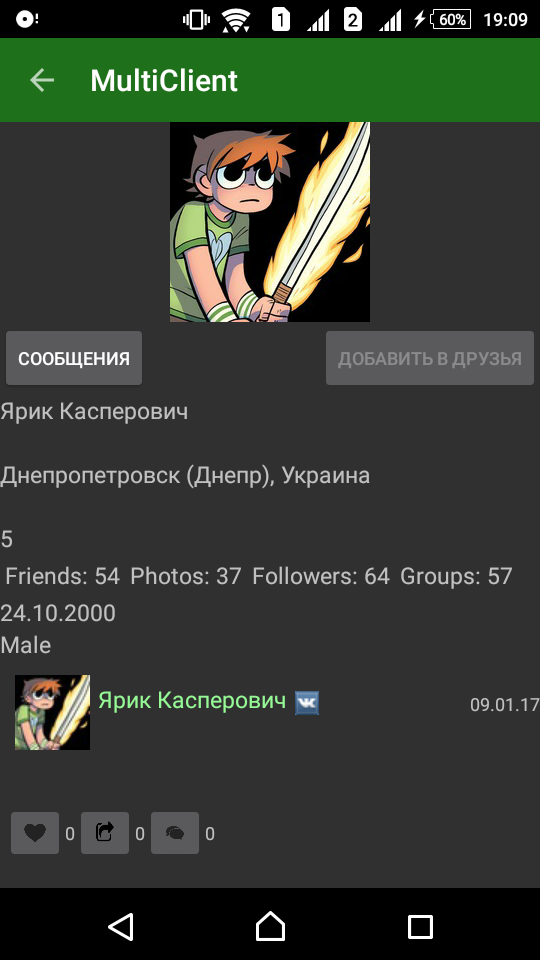
Переміщення між сторінками користувача здійснюється так само, як і між новинами і особистими повідомленнями.

Головне меню дозволяє переміщатись між частинами додатка.

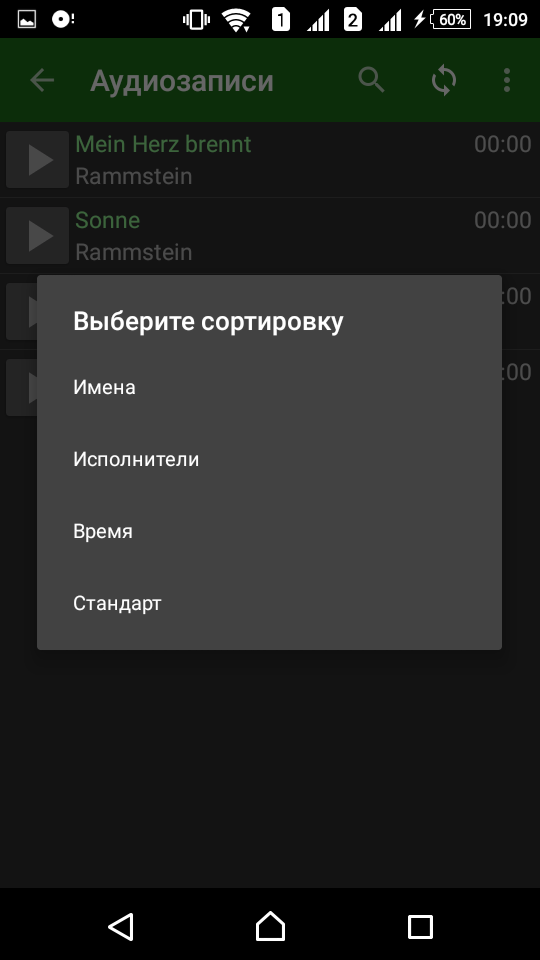
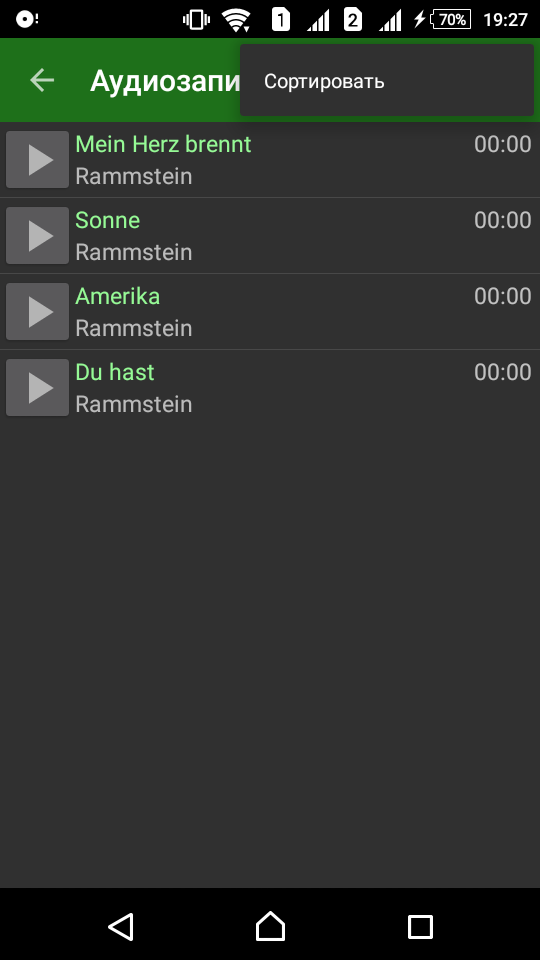
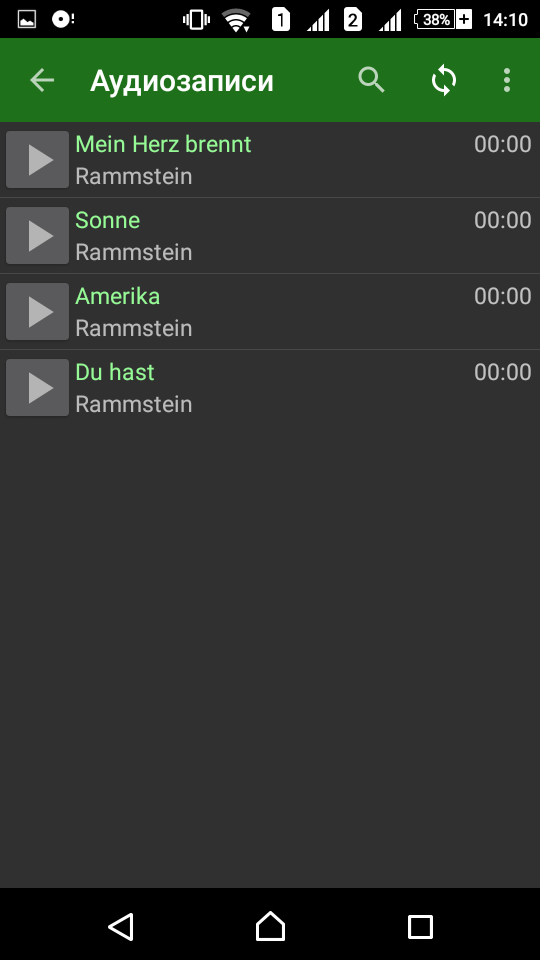
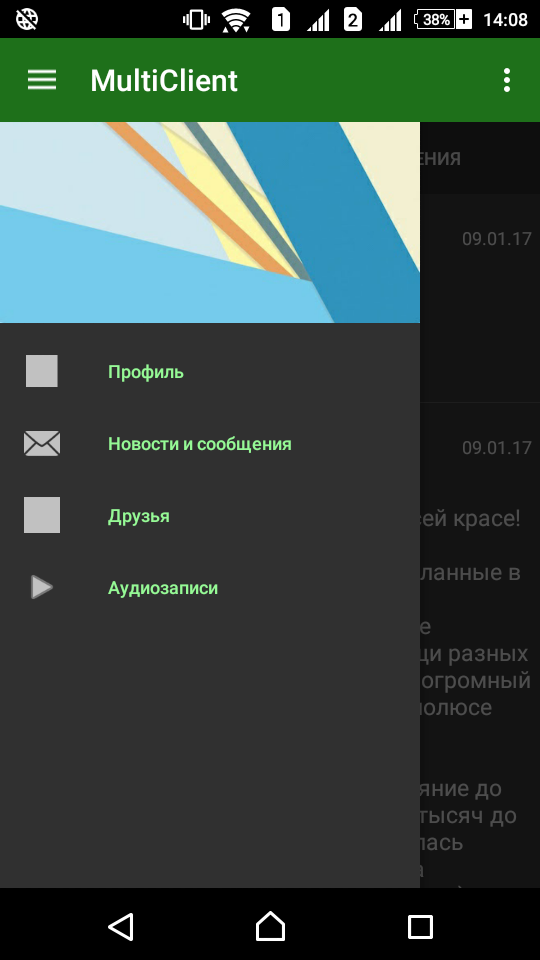
## Друзі

При натисканні по користувачу, який відображається с мережі «ВКонтакте» здійснюється перехід до його профілю.

Окрім можливості перегляду списку друзів, можна робити пошук інших людей у мережі «ВКонтакте».



## Аудіозаписи



Сортування може бути за декількома критеріями, які обирає користувач власноруч.

За допомогою контекстного меню можна робити пошук нових записів, перезавантаження та сортування.

# СТРУКТУРА ПРОГРАМИ



На малюнку представлена взаємодія пакетів додатка. Слово «call» означає, що цей пакет визиває до роботи інший пакет, слово «containe» означає, що класи даного пакета містять в собі класи з іншого пакета, слово «use», що класи цього пакету використовую можливості класів іншого пакету.

Надалі описано ряд класі з пакетів activities, fragments, lastFm та news. Крім того для більш детального розуміння є описання декількох класів з пакету cores.

Список класів з поясненням їх призначення:

* **class Application extends android.app.Application** – перевизначає метод **onCreate()**, додаючи в нього ініціалізацію соціальних мереж «ВКонтакте» та «Facebook».
* **class MainActivity extends CoreActivity** – містить в собі **private static MainFragment mainFragment**, перевизначає метод **onCreate(Bundle savedInstanceState)**, надаючи початкові значення змінним **G.loginInVK**, **G.loginInFB** та **G.loginInLFM**. Перевизначає метод **public void onBackPressed()** для закінчення роботи програми після натиснення кнопки назад та має метод для отримання екзепляру **MainFragment** ішими класами.
* **class CoreActivity extends AppCompatActivity** – містить поле **private OnBackPressed onBackPressed,** метод **public void setOnBackPressed(OnBackPressed onBackPressed)**, який дає можливість іншим класам встановлювати поле **onBackPressed** та сам метод **public void onBackPressed()**, який визначає стандартні налаштування, якщо їх не треба змінювати.
* **interface OnBackPressed** – має єдиний метод **void onBack()**, який потрібен для перевизначення його в майбутньому іншими класами.
* **class MainFragment extends Fragment** – має поля **private CallbackManager callbackManager**, **private LFMLoginButton lfmLoginButton** та **private static final String[] myScope**. Перші два поля потрібні для ініціалізації Facebook’а, а останнє для визначення прав, які має надати користувач додатку у мережі ВКонтакте. Перевизначає метод **public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)**, реалізуючи в ньому роботу кнопок входу до всіх соціальних мереж, які є у додатку. Метод **private void startMenu()** робить перехід до **MenuActivity**, а метод **public LFMLoginButton getLfmLoginButton()** надає іншим класам доступ до кнокпи входу Last.fm. Саме цей клас забезпечую вхід у мережі та контролює, до яких зайшов користувач. Містить у собі крім того кнопки, які дозволяють робити вхід.
* **class CoreFragment extends Fragment implements OnBackPressed** реалізує метод **public void OnBack()**, який надає стандартну дію при натисканні кнопки назад.
* **class LFMLoginButton extends Button** -- надає можливість входу та виходу з аккаунта Last.fm у додатку.
* **class LFMLoginActivity extends CoreActivity** – якщо користувач хочу увійти до аккаунта Last.fm, з’являється актівіті, яке надає йому змогу це зробити. Актівіті має поля для вводу логіна та пароля, а також кнопку входу. Після входу статус кнопки на початковому екрані змінюється. Ключ до сесії зберігається за допомого класу **SessionSaver**.
* **class SessionSaver** –має ряд полів, які допомогають в зберіганні ключа доступу до сайту Last.fm. Має два методи **public static Session load(SharedPreferences sharPref)** та **public static void reset(SharedPreferences sharPref)**.
* **class MenuActivity extends CoreActivity** – містить у собі та реалізує головне меню та основні вкладки, а саме **private static MenuFragment menuFragment**, **private ProfileTabsFragment ptf** та **private static FriendsFragment ff**. Також реалізує методи контекстного меню та надає доступ до фрагменту з друзями.
* **class MenuFragment extends CoreFragment** – фрагмент, який містить у **MenuActivity**, має в собі два інших фрагменти, а саме **NewsFragment** та **PreviewListFragment**. Забезпечую перехід між цими фрагментами та надає до них доступ іншим класам.
* **class NewsFragment extends CoreFragment** – клас, який контрулює роботу та відображення новин. Має ряд полів, які містять у собі інформацію про новини загалом та деяку додаткову, наприклад, на яку останню новину натискав користувач. У методі **public View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)** здійснюється створення та иконання запиту до двох серверів соціальних мереж, а також ініціалізація деяких полів. Має в собі внутрішній клас **private class Listener extends CoreListener**, який виконує обробку відповіді від серверів та перетворює відповідь з JSON формату в модель, яка зберігає данні у цьому додатку. Тут відбувається завантаження зображень та створення екземплярів новин **News**. Окрім цього клас має методи для перезавантаження новин, їх видалення з різних соціальних мереж, фільтрацію та методи для додавання новини у список подобаючись записів та копіювання запису на власну сторінку. Визиває у собі три інших класи, які є діалоговими вікнами **DeleteDialogFB**, **DeleteDialogVK** та **FilterDialog**. Список **ListView** використовує власний адаптер для відображення новин **NewsAdapter**.
* **abstract class CoreListener implements Runnable** – клас, який надає поля та конструктори для наслідуючих класів. Має в собі поля для відповідей «ВКонтакте», «Facebook» та «Last.fm». Його наслідують всі внутрішні класи-слухачі.
* **class News implements Parcelable** – модель новин, яка зберігає в собі id новини, зображення того, хто виклав новину, його ім’я, текст до новини, прикріплені зображення, якщо вони є, дату, коли викладена новина, кількість людей, які додали новину до списку «Мені подобається», які скопіювали новини на свою стіну та скільки коментарів до цього запису, відмітка, звідки було отримано запис, id користувача, який опублікував, список документів **Docs**, прикріплених до новини та поле, яке позначає, чи додав користувач цю новину до списку, які подобаються та чи скопіював він її до себе на стіну. Має в собі 3 різних конструктори для різних цілей, методи для отримання іншими класами значення полів моделі та методи для запису і відновлення моделі з пам’яті, за допомогою інтерфейсу **Parcelable**.
* **class NewsAdapter extends CoreAdapter<News>** – клас, який реалізує відображення елементів, які зберігає модель новини. Відображає всі поля, які містить новина, окрім id новини та користувача. Створює у елементі новин нові **ImageView**, в які ставиться зображення, яке було завантажене раніше. Перевизначає метод для кнопок «Мені подобається» та «Додати на мою стіну».

## Напрямки використання

Додаток може використовуватися будь-ким у будь-якій сфері, яка потребує спілкування між людьми на відстані. Окрім того, додаток дозволяє читати новини одразу з декількох мереж. Це та інші можливості додатка актуальні для молодих людей і всіх, хто користується соціальними мережами.

## Програмно-апаратні вимоги

Для роботи додаток потребує Android 4.0.3 або вище, мінімум 11 Мб вільної пам’яті на пристрої, сенсорний екран та підтримку Wi-fi, оскільки додаток краще за все працює саме за допомогою швидкого інтернету.

## Комплектація програми

Програма має в собі єдиний файл – MultiClient.apk. Це стиснений файл встановлення додатка, який підтримує Android.

## Використані програмні засоби

Написання додатку відбувалось за допомогою середовища Android Studio на мові програмування Java. Також було використано декілька графічних редакторів, таких як Gimp та Inkscape для створення та редагування деяких іконок, використаних у роботі. Для створення карти з пакетів, а також карти класів для розуміння зв’язку між ними біло використано програму StarUML. Для використання функцій та можливостей зовнішніх мереж було використано VKApi, FacebookApi та LastFmApi, які наявні у вільному доступі для всіх. Для створення документації, анотації та презентації використано продукти з Microsoft Office .

# ВИСНОВКИ

У роботі виконано більшу частину поставлених цілей, було використано багато різноманітних алгоритмів, створено декілька різноманітних блоків, об’єднаних в один, створено зрозумілий для користувача інтерфейс. За допомогою зображень екрану та графічних об’єктів Word було створено зрозумілу інструкцію з коментарями, яка показує основні можливості додатка. На жаль, велику чаcтину роботи було втрачено через зміну умов соціальної мережі «ВКонтакте», через що повністю працюючий аудіоплеєр з аудіозаписами з цієї мережі було видалено з програми та замінено на композиції з сайту Last.fm. Через ці зміни було змінено досить велику частину коду і втрачено ряд можливостей, які мали бути спочатку, наприклад, еквалайзер та прослуховування музики не виходячи з додатку. Було вирішено дуже багато проблем в різних сферах роботи з додатком для різних цілей та потреб, починаючи з запитів до сервера, перетворенням відповідей на моделі, які зрозумілі додатку, до їх зберігання та відображення на екрані. Було вивчено 3 різних документації для зовнішніх розробників від «ВКонтакте», «Facebook» та «Last.fm». Встановлено 3 модулі від цих API, що забезпечило більш простий доступ до ресурсів цих сайтів та спростило роботу. У роботі є декілька класів, які доповняють зовнішні API, хоча написані власноруч. Це кнопки входу да вікно входу до Last.fm. Загалом використано багато зовнішніх алгоритмів, які тісно зв’язані з власними алгоритмами. Було виконано деяке спрощення роботи за допомогою створення абстрактних класів, за допомогою яких надалі буде простіше доповнювати та модифікувати додаток. На даний момент додаток не зовсім готовий до повноцінного використання, оскільки не підтримує деякі популярні функції, а тому потребує подальшої розробки та покращення.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

* Head First. Программирование для Android
* Шилдт Г. - Java 8. Полное руководство – 2015
* <https://habrahabr.ru>
* <http://stackoverflow.com/>
* <https://developer.android.com>
* <http://startandroid.ru/>
* <http://developer.alexanderklimov.ru/>