

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської кваліфікаційної роботи
на ступінь вищої освіти бакалавр

на тему: **«Розробка додатку «Sociable» з використанням
Kotlin для Android»**

Виконала: студентка 4 курсу, групи ПД-44

спеціальності 121 Інженерія програмного
забезпечення

(шифр і назва спеціальності)

Стефанюк Л.Б

(прізвище та ініціали)

Керівник

Гребенюк В.В

(прізвище та ініціали)

Рецензент

(прізвище та ініціали)

Нормоконтроль

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Спеціальність -121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного
забезпечення

_____ О.В. Негоденко

« ____ » _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ
СТЕФАНІУК ЛІЛІ БОГДАНІВНІ

1.Тема роботи: «Розробка додатку «Sociable» з використанням Kotlin для Android».

Керівник роботи Гребенюк Віктор Вікторович, доцент кафедри, доктор філософії (PhD) затверджені наказом вищого навчального закладу від — «12» березня 2021 року №65.

2. Строк подання студентом роботи 01.06.2021

3. Вхідні дані до роботи:

3.1. Середовище розробки Android Studio

3.2. Алгоритм дії додатку

3.3. Науково-технічна література, пов'язана з розробкою додатків за допомогою середовища розробки Android Studio на мові програмування Kotlin

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити).

4.1. Аналіз обов'язків розроблюваного додатку

4.2. Аналіз та порівняння існуючих прототипів

4.3. Дослідження програмних засобів для розробки додатку

4.4. Розробити функціонал створеного додатку

5. Дата видачі завдання 19.05.2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.21 – 22.04.21	
2	Аналіз існуючих прототипів	22.04.21 – 23.04.21	
3	Дослідження програмних засобів	23.04.21 – 01.05.21	
4	Моделювання об'єкту проектування	01.05.21 – 05.05.21	
5	Розробка функціоналу додатку	05.05.21 – 07.05.21	
6	Вступ, висновки, реферат	07.05.21 – 08.05.21	
7	Розробка презентації застосунку	08.05.21 – 24.05.21	
8	Попередній захист роботи	25.05.21	

Студент

Керівник роботи

РЕФЕРАТ

Текстова частина бакалаврської роботи 52с., 3 рис., 51 джерел.

Ключеві слова: Kotlin, Anroid Studio, Android OS, Firebase, соціальна мережа, мобільні додатки, діаграми.

Об'єкт дослідження – поліпшення функціоналу та якості соціальних мереж.

Предмет дослідження – соціальна мережа, для зручного використання із додаванням нового функціоналу.

Мета роботи – розробка додатку «Sociable» з використанням Kotlin для Android.

Методи дослідження – методи теорії інформації, методи структурного аналізу і проектування, методи розробки програмного забезпечення, методи тестування, валідації та верифікації програмного забезпечення.

Наукова новизна даної роботи полягає в наступному:

1. Розроблено алгоритм для покращення роботи соціальної мережі.
2. Встановлено, що зв'язка Firebase з мовою програмування Kotlin, показує найкращі результати по швидкодії та використанню ресурсів сервера.
3. На основі результатів виконаних порівнянь та досліджень, був розроблений авторська соціальна мережа для зручного використання із додавання нового функціоналу.

В роботі виконано аналіз існуючих соціальних мереж і серверів. В результаті аналізу було визначено основні потреби користувачів. Проаналізовано можливості середовища розробки Android Studio. Розроблено логіку практичних завдань та загальну концепцію представлення інформації для користувачів.

Галузь використання – додаток може використовувати будь-яка людина, яка має доступ до інтерне

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	10
1.1. Мобільні додатки	10
1.2. Розробка мобільних додатків.....	13
1.3. Соціальні мережі	15
РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ.....	25
2.1. Мова програмування Kotlin	25
2.2. Середовище розробки Android Studio	29
2.3. Android OS	31
2.4. Firebase	41
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	44
3.1. Діаграма варіантів використання	44
3.2. Діаграма класів програмного продукту	47
3.3. Структурна діаграма ІС	52
ВИСНОВКИ.....	53
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54
ДОДАТКИ.....	59
Додаток А.....	59
Додаток Б.	75

ВСТУП

У зв'язку з надзвичайною швидкістю поширення тенденції сучасного соціуму на діджиталізацію різноманітних процесів, які утворюють життя людини, все більше щоденних занять людей переносяться з реального життя в інтернет.

Подібна тенденція в першу чергу торкнулася механізму спілкування, яке з кожним днем все більше переноситься в медіа простір. На хвилі подібних змін набули актуальності так звані соціальні мережі, онлайн-платформи, які використовуються для спілкування, знайомств, створення соціальних відношення між людьми, які мають схожі інтереси або офлайн-зв'язки, а також розваг і роботи.

Виходячи з актуальності явища соціальних мереж, об'єктом дослідження було обрано соціальні мережі.

В свою чергу, предметом дослідження є створення власної соціальної мережі.

Метою роботи є розроблення мобільного додатку соціальної мережі «Sociable» на базі ОС Android засобами мови програмування Kotlin.

Для досягнення поставленої мети слід виконати наступні завдання:

1. Проаналізувати предметну область.
 - проаналізувати поняття мобільних додатків;
 - провести огляд методів і засобів розробки мобільних додатків;
 - проаналізувати поняття соціальної мережі;
2. Провести огляд обраних засобів реалізації.
 - провести огляд мови програмування Kotlin;
 - провести огляд середовища розробки Android Studio;
 - провести огляд ОС Android;
 - провести огляд хмарного сервісу Firebase

3. Описати процес розробки програмного забезпечення.

- створити діаграму варіантів використання;
- створити діаграму класів програмного продукту;
- створити структурну діаграму.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Мобільні додатки

Мобільний додаток - це комп'ютерна програма або програмне забезпечення, призначене для роботи на мобільному пристрої, такому як телефон, планшет або годинник. Додатки спочатку були призначені для сприяння підвищенню продуктивності, наприклад електронної пошти, календаря та контактних баз даних, але суспільний попит на програми спричинив швидке розширення в інших сферах, таких як мобільні ігри, автоматизація роботи на заводі, послуги GPS та локації, відстеження замовлень та квитки. покупки, завдяки чому зараз доступні мільйони програм. Зазвичай програми завантажуються з платформ розповсюдження програм, якими керує власник мобільної операційної системи, наприклад, App Store (iOS) або Google Play Store. Деякі програми безкоштовні, а інші мають ціну, при цьому прибуток розподіляється між творцем програми та платформою розповсюдження. Мобільні програми часто відрізняються від настільних програм, призначених для роботи на настільних комп'ютерах, та веб-програм, які працюють у мобільних веб-браузерах, а не безпосередньо на мобільному пристрої.

У 2009 році оглядач технологій Девід Поуг заявив, що смартфони можуть отримати прізвисько "телефонні додатки", щоб відрізнити їх від раніше менш складних смартфонів. [1] Термін "додаток", скорочене від "програмне забезпечення", з тих пір став дуже популярним; у 2010 році Американське товариство діалектів визначило його "Словом року". [2]

Більшість мобільних пристроїв продаються з декількома програмами, що постачаються у комплекті з попередньо встановленим програмним забезпеченням, наприклад веб-браузером, поштовим клієнтом, календарем, картографічною програмою та додатком для придбання музики, інших засобів

масової інформації чи інших додатків. Деякі попередньо встановлені програми можна видалити звичайним процесом видалення, залишаючи таким чином більше місця для бажаних. Якщо програмне забезпечення не дозволяє цього, деякі пристрої можуть викорінюватися для усунення небажаних програм.

Програми, які не встановлені попередньо, зазвичай доступні через платформи розповсюдження, які називаються магазинами програм. Вони почали з'являтися в 2008 році, і ними, як правило, керує власник мобільної операційної системи, наприклад, Apple App Store, Google Play, Windows Phone Store та BlackBerry App World. Однак існують незалежні магазини додатків. Деякі програми безкоштовні, а інші потрібно купувати. Зазвичай їх завантажують з платформи на цільовий пристрій, але іноді їх можна завантажити на ноутбуки або настільні комп'ютери. Для додатків із ціною, як правило, відсоток, 20-30%, надходить до постачальника дистрибуції (наприклад, iTunes), а решта надходить до виробника програми. [3] Отже, той самий додаток може коштувати різну ціну залежно від мобільної платформи.

Програми також можна встановлювати вручну, наприклад, запускаючи пакет програм Android на пристроях Android.

Спочатку мобільні програми пропонувались для загальної продуктивності та пошуку інформації, включаючи електронну пошту, календар, контакти, фондовий ринок та інформацію про погоду. Однак суспільний попит та доступність інструментів для розробників спонукали до швидкого розширення на інші категорії, такі як ті, якими обробляються набори програм для настільних програм. Як і у випадку з іншим програмним забезпеченням, збільшення кількості та різноманітності програм зробило відкриття складним завданням, що, в свою чергу, призвело до створення широкого спектру джерел огляду, рекомендацій та курирування, включаючи блоги, журнали та спеціальні послуги з виявлення додатків в Інтернеті. У 2014 році державні регуляторні органи почали намагатися регулювати та керувати програмами, зокрема медичними. [4] Деякі компанії пропонують додатки як

альтернативний метод доставки вмісту з певними перевагами перед офіційним веб-сайтом.

Зі зростанням кількості мобільних додатків, доступних у магазинах додатків, і вдосконаленими можливостями смартфонів, люди завантажують більше додатків на свої пристрої. [5] Використання мобільних додатків стає дедалі поширенішим серед користувачів мобільних телефонів. [6] Дослідження comScore, проведене у травні 2012 року, повідомляло, що протягом попереднього кварталу більше абонентів мобільних пристроїв використовувало додатки, ніж переглядало веб-сторінки на своїх пристроях: 51,1% проти 49,8% відповідно [7]. Дослідники виявили, що використання мобільних додатків суттєво корелює з контекстом користувача та залежить від місцезнаходження користувача та часу доби [8]. Мобільні додатки відіграють дедалі більшу роль у галузі охорони здоров'я, і коли їх правильно розробити та інтегрувати, це може принести багато переваг.

Фірма з досліджень ринку Gartner передбачила, що в 2013 році буде завантажено 102 млрд. Додатків (91% з них безкоштовно), що принесе в США 26 млрд. Дол., Що на 44,4% більше порівняно з 18 млрд. Дол. До другого кварталу 2015 року лише магазини Google Play та Apple заробили 5 мільярдів доларів. Згідно з підрахунками аналітиків, економіка додатків створює дохід на рівні понад 10 млрд. Євро на рік в межах Європейського Союзу, тоді як у 28 штатах ЄС через зростання ринку додатків створено понад 529 000 робочих місць. [12]

Усі програми, націлені на певну мобільну платформу, відомі як власні програми. Тому програма, призначена для пристрою Apple, не працює на пристроях Android. Як результат, більшість підприємств розробляють програми для декількох платформ.

Розробляючи власні програми, професіонали включають найкращі у своєму класі модулі інтерфейсу користувача. Це зумовлює кращу продуктивність, послідовність та хороший досвід роботи з користувачем. Користувачі також отримують вигоду від ширшого доступу до інтерфейсів

програмування програм та безмежно використовують усі програми з конкретного пристрою. Крім того, вони також без особливих зусиль перемикаються з одного додатка на інший.

Основною метою створення таких програм є забезпечення найкращої продуктивності для конкретної мобільної операційної системи.

Концепція гібридної програми - це поєднання власних та веб-програм. До цієї категорії належать програми, розроблені за допомогою Apache Cordova, Xamarin, React Native, Sencha Touch та інших подібних технологій.

Вони створені для підтримки веб-та рідних технологій на багатьох платформах. Більше того, ці програми легше і швидше розробляти. Він передбачає використання однієї кодової бази, яка працює в декількох мобільних операційних системах.

Незважаючи на такі переваги, гібридні програми мають нижчу продуктивність. Часто програми не мають однакового зовнішнього вигляду в різних мобільних операційних системах.

Веб-програма кодується в HTML5, CSS або JavaScript. Доступ до Інтернету необхідний для належної поведінки та користувацького досвіду цієї групи програм.

Ці програми можуть захоплювати мінімальний простір пам'яті на пристроях користувача порівняно із власними та гібридними програмами. Оскільки всі бази персональних даних зберігаються на Інтернет-серверах, користувачі можуть отримати потрібні дані з будь-якого пристрою через Інтернет.

1.2. Розробка мобільних додатків

Розробка мобільних додатків - це процес або процес, за допомогою якого розробляється мобільний додаток для мобільних пристроїв, таких як особисті цифрові помічники, корпоративні цифрові помічники або мобільні телефони. Ці програми можна попередньо встановити на телефони під час виробничих

платформ або доставити як веб-програми, використовуючи обробку на стороні сервера або клієнта (наприклад, JavaScript), щоб забезпечити "подібний до програми" досвід роботи у веб-браузері. Розробники прикладного програмного забезпечення також повинні враховувати широкий спектр розмірів екрану, технічних характеристик та конфігурацій через сильну конкуренцію в мобільному програмному забезпеченні та зміни на кожній з платформ. Розробка мобільних додатків неухильно зростає, збільшуються доходи та створюються робочі місця. Згідно з аналітичним звітом 2013 року, в ЄС налічується 529 000 робочих місць з економікою прямих додатків, тоді як 28 членів (включаючи Великобританію), 60 відсотків з яких є розробниками мобільних додатків.

Як частина процесу розробки, дизайн мобільного користувацького інтерфейсу (UI) також має важливе значення для створення мобільних додатків. Мобільний користувацький інтерфейс розглядає обмеження, контексти, екран, введення та мобільність як начерки дизайну. Користувач часто є центром взаємодії зі своїм пристроєм, а інтерфейс включає компоненти як апаратного, так і програмного забезпечення. Введення користувача дозволяє користувачам маніпулювати системою, а вихід пристрою дозволяє системі вказувати на ефекти маніпуляцій користувачів. Обмеження дизайну мобільного інтерфейсу включають обмежену увагу та форм-фактори, такі як розмір екрану мобільного пристрою для рук (рук) користувача. Контексти мобільного інтерфейсу сигналізують про підказки від активності користувачів, таких як розташування та планування, які можуть відобразитися в результаті взаємодії користувачів у мобільному додатку. Загалом, мета дизайну мобільного інтерфейсу - головним чином зрозумілий, зручний інтерфейс. Інтерфейс мобільних додатків повинен: враховувати обмежену увагу користувачів, мінімізувати натискання клавіш та бути орієнтованим на завдання з мінімальним набором функцій. Цю функціональність підтримують мобільні корпоративні платформи додатків або інтегровані середовища розробки (IDE).

Мобільні інтерфейси або інтерфейсні інтерфейси покладаються на мобільні фонові інтерфейси для підтримки доступу до корпоративних систем. Мобільний сервер сприяє маршрутизації даних, безпеці, автентифікації, авторизації, роботі в автономному режимі та організації служб. Ця функціональність підтримується поєднанням проміжних компонентів, включаючи сервер мобільних додатків, мобільний серверний сервіс як послугу (MBaaS) та інфраструктуру орієнтованої на службу архітектури (SOA).

Розробка мобільних додатків стає все більш важливою для багатьох підприємств, де понад 3 мільярди людей у всьому світі користуються смартфонами, понад 1,5 мільярда - планшетами станом на 2019 рік. Користувачі в середньому проводять 90% свого мобільного часу в додатках, а їх більше 700 мільйони завантажень програм із різних магазинів програм. [2]

1.3. Соціальні мережі

Послуга соціальних мереж (також веб-сайт соціальних мереж або соціальні медіа) - це онлайн-платформа, яку люди використовують для побудови соціальних мереж або соціальних відносин з іншими людьми, які поділяють подібні особисті чи кар'єрні інтереси, діяльність, досвід чи реальні зв'язки. [1] [2]

Послуги соціальних мереж відрізняються за форматом та кількістю функцій. Вони можуть включати цілий ряд нових інформаційних та комунікаційних засобів, що працюють на настільних ПК та ноутбуках, на мобільних пристроях, таких як планшетні комп'ютери та смартфони. Вони можуть мати цифрові фото / відео / спільний доступ та щоденникові записи в Інтернеті (ведення блогу). [2] Інтернет-послуги спільноти іноді розглядаються розробниками та користувачами як послуги соціальних мереж, хоча в більш широкому розумінні служба соціальних мереж, як правило, надає послугу, орієнтовану на індивіда, тоді як послуги Інтернет-спільноти орієнтовані на

групу. Визначені як "веб-сайти, що сприяють побудові мережі контактів з метою обміну різними типами вмісту в Інтернеті", сайти соціальних мереж надають простір для взаємодії, що триває не лише в особистих взаємодіях. Ці комп'ютерно-посередницькі взаємодії пов'язують членів різних мереж і можуть допомогти як підтримувати, так і розвивати нові соціальні та професійні стосунки. [3]

Сайти соціальних мереж дозволяють користувачам ділитися ідеями, цифровими фотографіями та відео, публікаціями та інформувати інших про діяльність та події в Інтернеті чи реальному світі з людьми в їх мережі. Хоча особисті соціальні мережі - наприклад, зібрання на сільському ринку, щоб поговорити про події - існували з самого раннього розвитку міст [4], Інтернет дозволяє людям спілкуватися з іншими людьми, які проживають в різних місцях, починаючи від міста. по всьому світу. Залежно від платформи соціальних медіа, учасники можуть мати можливість зв'язатися з будь-яким іншим учасником. В інших випадках учасники можуть зв'язатися з ким-небудь, з ким вони мають зв'язок, і згодом з кожним, хто має зв'язок, тощо. Успіх служб соціальних мереж можна побачити в їх домінуванні в суспільстві сьогодні: Facebook має масових 2,13 млрд активних щомісячних користувачів і в середньому 1,4 млрд активних щоденних користувачів у 2017 р. [5] LinkedIn, орієнтована на кар'єру послуга соціальних мереж, зазвичай вимагає, щоб учасник особисто знав іншого учасника в реальному житті, перш ніж він зв'яжеться з ними в Інтернеті. Деякі послуги вимагають від членів попередньо встановленого зв'язку для зв'язку з іншими учасниками.

Основні типи послуг соціальних мереж містять категорії місць (наприклад, вік чи рід занять чи релігія), засоби спілкування з друзями (як правило, сторінки самоопису) та систему рекомендацій, пов'язану з довірою. Можна поділити соціальні мережі на чотири типи: [6]

- послуги соціалізації соціальних мереж, що використовуються в основному для спілкування з існуючими друзями (наприклад, Facebook)

- Інтернет-соціальні мережі - це децентралізовані та розподілені комп'ютерні мережі, де користувачі спілкуються між собою через Інтернет-послуги.
- мережеві послуги соціальних мереж, що використовуються в основному для несоціальних міжособистісних комунікацій (наприклад, LinkedIn, сайт, орієнтований на кар'єру та працевлаштування)
- послуги соціальної навігації соціальні мережі, що використовуються в основному для допомоги користувачам у пошуку конкретної інформації або ресурсів (наприклад, Goodreads для книг)

Були спроби стандартизувати ці послуги, щоб уникнути необхідності дублювати записи друзів та інтересів (див. Стандарт FOAF). Дослідження показує, що Індія зафіксувала найбільший у світі ріст користувачів соціальних мереж у 2013 році. [7] Опитування 2013 року показало, що 73% дорослих людей США використовують веб-сайти соціальних мереж. [8]

В Інтернеті доступні різноманітні послуги соціальних мереж. Однак більшість включають загальні ознаки: [1]

- послугами соціальних мереж є Інтернет-програми [1] [9]
- створений користувачами вміст (UGC) - це життєва сила соціальних мереж. [1] [9]
- користувачі створюють спеціальні профілі для веб-сайту або програми, які розробляються та підтримуються організацією SNS [1] [2]
- послуги соціальних мереж сприяють розвитку соціальних мереж в Інтернеті, пов'язуючи профіль користувача з профілем інших людей або груп. [1] [2] [10]

Різноманітність та зміна асортименту самостійних та вбудованих послуг соціальних мереж в Інтернет-просторі створює складність у визначенні. [1] Крім того, ідея того, що ці послуги визначаються їх здатністю об'єднувати людей і надає занадто широке визначення. Таке широке визначення може

припустити, що телеграф і телефон були послугами соціальних мереж - а не Інтернет-технологій, які мають намір описати вчені [11]. Термінологія також незрозуміла, де деякі посилаються на послуги соціальних мереж як соціальні медіа. [2]

Нещодавня спроба [1] дати чітке визначення переглянула визначну літературу в цьому районі та визначила чотири спільні риси, унікальні для сучасних служб соціальних мереж:

1. послуги соціальних мереж - це інтерактивні програми на базі Інтернету Web 2.0, [1] [9]
2. створений користувачами вміст (UGC), такий як цифрові фотографії, що надсилаються користувачами, текстові повідомлення, "додавання тегів", коментарі в Інтернеті та "веб-журнали" (блоги) у стилі щоденника, - це життєва сила організму SNS, [1] [9]
3. користувачі створюють спеціальні профілі для сайту або програми, які розробляються та підтримуються організацією SNS, [1] [2] та
4. послуги соціальних мереж полегшують розвиток соціальних мереж в Інтернеті, пов'язуючи профіль користувача з профілем інших людей чи груп. [1] [2]

Потенціал комп'ютерних мереж сприяти нещодавно вдосконаленим формам комп'ютерної опосередкованої соціальної взаємодії був запропонований ще на початку [33]. Зусилля для підтримки соціальних мереж за допомогою комп'ютерної комунікації були докладені в багатьох ранніх онлайн-сервісах, включаючи Usenet, [34] ARPANET, LISTSERV та послуги дошки оголошень (BBS). Багато прототипових функцій сайтів соціальних мереж також були присутні в таких онлайн-сервісах, як America Online, Prodigy, CompuServe, ChatNet та The WELL. [35]

Ранні соціальні мережі у Всесвітній павутині розпочались у формі узагальнених інтернет-спільнот, таких як Theglobe.com (1995), [36] Geocities

(1994) та Tripod.com (1995). Багато з цих раних спільнот зосереджувались на об'єднанні людей для взаємодії один з одним за допомогою чатових кімнат та заохочували користувачів ділитися особистою інформацією та ідеями через особисті веб-сторінки, надаючи прості у користуванні засоби публікації та безкоштовний або недорогий веб-простір. Деякі спільноти - наприклад, Classmates.com - застосували інший підхід, просто запросивши людей посилатися один на одного через електронні адреси. PlanetAll стартував у 1996 році.

Наприкінці 90-х років профілі користувачів стали центральною особливістю сайтів соціальних мереж, що дозволило користувачам складати списки "друзів" та шукати інших користувачів зі схожими інтересами. Нові методи соціальних мереж були розроблені до кінця 1990-х років, і багато сайтів почали розробляти більш розширені функції, щоб користувачі могли знаходити друзів та керувати ними [37]. Open Diary, спільнота онлайн-діалогів, винайшла вміст, призначений лише для друзів, і коментар читача - дві особливості соціальних мереж, важливих для взаємодії з користувачем. [9]

Це нове покоління веб-сайтів соціальних мереж почало процвітати з появою SixDegrees у 1997 р. [2], після чого відкритим щоденником у 1998 р. [38] Mixi у 1999 р. [39] Makeoutclub у 2000 р. [40] [41] Hub Culture у 2002 р. - перша соціальна мережа Friendster та Канада Nexoria у 2003 р. [42], яка незабаром стала частиною масової мережі Інтернет. Однак, завдяки високому рівню проникнення Інтернету в країну, першим масовим сайтом соціальних мереж став південнокорейський сервіс Cyworld, заснований як веб-сайт у блогах у 1999 році, а функції соціальних мереж додані в 2001 році [43] [2]. Вона також стала однією з перших компаній, які отримали прибуток від продажу віртуальних товарів. [44] [45] Того ж року за Фрестертером пішли MySpace та LinkedIn, а зрештою і Бебо. Френдстер став дуже популярним на Тихоокеанських островах. Orkut став першим популярним сервісом соціальних мереж у Бразилії (хоча більшість його перших користувачів були зі Сполучених Штатів) і швидко зростав популярністю в Індії (Madhavan,

2007). [2] Підтверджуючи стрімке зростання популярності сайтів соціальних мереж, до 2005 року повідомлялося, що Myspace отримує більше переглядів сторінок, ніж Google. Facebook, [46], запущений в 2004 році, став найбільшим сайтом соціальних мереж у світі [47] на початку 2009 року. [48] Facebook вперше був представлений як Гарвардський сайт соціальних мереж [2], розширившись до інших університетів і, врешті-решт, як хтось. Термін соціальні медіа був введений і незабаром набув поширення. [49]

Веб-сервіси соціальних мереж дають змогу зв'язати людей, які поділяють інтереси та діяльність через політичні, економічні та географічні кордони. [50] За допомогою електронної пошти та обміну миттєвими повідомленнями створюються інтернет-спільноти, де заохочується економія подарунків та взаємний альтруїзм завдяки співпраці. Інформація підходить для подарункової економіки, оскільки інформація є нерівноцінним товаром і може бути подарована практично безкоштовно [51] [52]. Вчені відзначають, що термін "соціальний" не може враховувати лише технологічні особливості платформ соціальних мереж. [53] Отже, рівень комунікабельності мережі повинен визначатися фактичними показниками її користувачів. Відповідно до теорії комунікацій використання та задоволення, все більше людей звертаються до Інтернету та соціальних медіа для задоволення когнітивних, афективних, особистих інтегративних, соціальних інтеграційних та вільних від напруги потреб. Інтернет-технології, як доповнення для задоволення потреб, у свою чергу впливають на повсякденне життя, включаючи стосунки, школу, церкву, розваги та сім'ю [54]. Компанії використовують соціальні медіа як спосіб дізнатись про особистості та поведінку потенційних працівників. У багатьох ситуаціях кандидата, який міг би бути прийнятий на роботу, відхиляли через образливі або непристойні фотографії чи коментарі, розміщені в соціальних мережах або через стрічку новин.

Facebook та інші інструменти соціальних мереж все частіше стають цілями наукових досліджень. Вчені в багатьох галузях почали досліджувати вплив соціальних мереж, досліджуючи, як такі сайти можуть впливати на

питання особистості, політики, конфіденційності, [55] соціального капіталу, молодіжної культури та освіти [56]. Дослідження також припускають, що люди додають друзів у режимі офлайн на Facebook, щоб підтримувати контакт, і це часто стирає межі між роботою та домашнім життям. [57] Користувачі з усього світу також використовують сайти соціальних мереж як альтернативне джерело новин. [58] Хоча сайти соціальних мереж, можливо, змінили спосіб доступу до новин, [59] користувачі, як правило, мають неоднозначні думки щодо надійності вмісту, доступного через ці веб-сайти. [60]

Згідно з дослідженням, проведеним у 2015 році, 63% користувачів Facebook або Twitter у США вважають ці мережі своїм головним джерелом новин, при цьому найпопулярнішими є розважальні новини. У часи останніх новин користувачі Twitter частіше залишаються вкладеними в цю історію. У деяких випадках, коли новина є більш політичною, користувачі можуть з більшою ймовірністю висловити свою думку щодо пов'язаної історії Facebook коментарем чи лайком, тоді як користувачі Twitter просто стежать за стрічкою сайту та ретвітнуть статтю. [61] В соціальних мережах в Інтернеті правдивість та надійність новин може зменшитися через відсутність традиційних медіа-воротарів. [62]

Згідно з дослідженням, проведеним у 2015 році, 63% користувачів Facebook або Twitter у США вважають ці мережі своїм головним джерелом новин, при цьому найпопулярнішими є розважальні новини. У часи останніх новин користувачі Twitter частіше залишаються вкладеними в цю історію. У деяких випадках, коли новина є більш політичною, користувачі можуть з більшою ймовірністю висловити свою думку щодо пов'язаної історії Facebook коментарем чи лайком, тоді як користувачі Twitter просто стежать за стрічкою сайту та ретвітують статтю. [61] В соціальних мережах в Інтернеті достовірність та надійність новин може зменшитися через відсутність традиційних медіа-воротарів. [62]

Дослідження 2015 року показує, що 85% людей у віці від 18 до 34 років використовують сайти соціальних мереж для прийняття рішення про придбання. У той час як понад 65% людей у віці 55 років і надмірно покладаються на усну передачу [63]. Кілька веб-сайтів починають застосовувати силу моделі соціальних мереж для благодійності. Такі моделі забезпечують спосіб зв'язку роздроблених галузей та малих організацій без ресурсів для охоплення ширшої аудиторії із зацікавленими користувачами. [64] Соціальні мережі забезпечують інший спосіб спілкування людей у цифровій формі. Ці спільноти гіпертекстів дозволяють обмінюватися інформацією та ідеями, що є старою концепцією, розміщеною в цифровому середовищі. У 2011 році HCL Technologies провела дослідження, яке показало, що 50% британських роботодавців заборонили використовувати сайти / послуги соціальних мереж у робочий час. [65] [66]

Дослідження дали нам неоднозначні результати щодо того, чи може участь людини у соціальних мережах вплинути на її почуття самотності. Дослідження показали, що те, як людина обирає використовувати соціальні мережі, може змінити її почуття самотності як негативним, так і позитивним чином. Деякі компанії із мобільними працівниками заохочують своїх працівників використовувати соціальні мережі, щоб відчувати зв'язок. Педагоги використовують соціальні мережі, щоб залишатися на зв'язку зі своїми учнями, тоді як люди використовують їх, щоб підтримувати зв'язок зі своїми близькими стосунками. [67] Кожен користувач соціальних мереж здатний створити спільноту, яка зосереджується навколо особистої ідентичності, яку вони вирішили створити в Інтернеті. [68] У своїй книзі «Цифрові ідентичності: Створення та спілкування з Інтернетом» [69] Роб Кавер стверджує, що основою соціальних мереж у Web 2.0 є високошвидкісна мережа, яка зміщує представлення в Інтернеті до візуального та відносного до інших людей, ускладнюючи процес ідентичності для молодих людей та створення нових форм тривоги. [69] У 2016 році в повідомленнях новин зазначалося, що надмірне використання сайтів SNS може бути пов'язане зі

збільшенням рівня депресії, що майже втричі перевищує показник для користувачів, які не належать до SNS. Експерти у всьому світі [які?] Заявили, що 2030 людей, які більше використовують СНС, мають вищий рівень депресії, ніж ті, хто користується СНС менше. [70] Принаймні одне дослідження дійшло до того, що дійшло висновку, що негативні наслідки використання Facebook рівні або більші, ніж позитивні ефекти віч-на-віч. [71]

Згідно з недавньою статтею "Комп'ютери в поведінці людини", також було доведено, що Facebook веде до питань соціального порівняння. Користувачі можуть вибирати, які фотографії та оновлення стану публікувати, дозволяючи їм зображати своє життя звинувачувальними манерами. [72] Ці оновлення можуть призвести до того, що інші користувачі відчуватимуть, що їхнє життя поступається в порівнянні. [73] Користувачі можуть відчувати особливу схильність порівнювати себе з іншими користувачами, з якими вони поділяють подібні характеристики чи спосіб життя, що призводить до більш справедливого порівняння. [72] Мотиви цих порівнянь можуть бути пов'язані з цілями вдосконалення себе, дивлячись на профілі людей, які відчувають себе вищими, особливо коли їх спосіб життя подібний і можливий [72]. Можна також здійснити самопорівняння, щоб зробити себе почуттям вищого за інших, переглянувши профілі користувачів, котрі вважають гіршими. [72] Однак дослідження Harvard Business Review показує, що ці цілі часто призводять до негативних наслідків, оскільки використання Facebook пов'язане з нижчим рівнем добробуту; Показано, що психічне здоров'я знижується через використання Facebook. [73] «Комп'ютери в поведінці людини» наголошує, що ці відчуття погіршення психічного здоров'я пропонували людям призупинити перерву у своїх облікових записах у Facebook; ця акція називається "Втома від Facebook" і є поширеною в останні роки. [72]

Використання соціальних мереж сприяло новій формі жорстокого спілкування, і академічні дослідження висвітлили низку соціально-технологічних пояснень цієї поведінки. Сюди входить анонімність, яку

надають міжособистісні комунікації, [74] фактори, що включають нудьгу чи пошук уваги, [75] або результат більш поляризованих онлайн-дебатів [76]. Вплив цього зловживання виявив наслідки поширеності Інтернет-знуцань та Інтернет-тролінгу. Також помітно зросло політичне насильство та зловживання через платформи соціальних медіа. Наприклад, одне дослідження Уорда та Маклафліна показало, що 2,57% усіх повідомлень, надісланих британським депутатам у Twitter, містять образливі повідомлення. [76]

РОЗДІЛ 2. ВИБІР ЗАСОБІВ РЕАЛІЗАЦІЇ

2.1. Мова програмування Kotlin

Kotlin [6] є крос-платформною, статично набраною мовою програмування загального призначення з висновками про тип. Kotlin розроблений для повноцінної взаємодії з Java, і версія JVM стандартної бібліотеки Kotlin залежить від бібліотеки класів Java [7], але умовивід типу дозволяє його синтаксису бути більш стислим. Kotlin в основному націлений на JVM, але також компілюється в JavaScript (наприклад, для веб-програм, що використовують інтерфейс, що використовує React [8]), або власного коду (через LLVM); наприклад, для власних додатків iOS, що діляться бізнес-логікою з додатками Android. [9] Витрати на розвиток мови несе JetBrains, тоді як Фонд Котліна захищає торгову марку Котлін. [10]

7 травня 2019 року Google оголосив, що мова програмування Kotlin тепер є найкращою мовою для розробників додатків для Android. В результаті багато розробників перейшли на Kotlin. [11] З моменту виходу Android Studio 3.0 у жовтні 2017 року Kotlin був включений як альтернатива стандартному компілятору Java. Компілятор Android Kotlin за замовчуванням виробляє байт-код Java 6 (який працює у будь-якому наступному JVM), але дозволяє програмісту вибрати націлити Java 8 до 15 для оптимізації [12] або дозволяє додаткові функції, наприклад Java 8 пов'язана з Kotlin 1.4, [13] і має експериментальну підтримку класу записів для сумісності Java 16. [14]

Підтримка Kotlin для JavaScript (тобто класичний бек-енд) розробниками вважається стабільною в Kotlin 1.3, тоді як Kotlin / JS (на базі ІЧ) у версії 1.4 - альфа-версією. Kotlin / Native Runtime (наприклад, для підтримки Apple) вважається бета-версією. [15]

У липні 2011 року JetBrains представила проект Kotlin, нову мову для JVM, який розроблявся рік. [16] Дмитро Ємеров, керівник JetBrains, заявив, що

більшість мов не мають тих функцій, які шукали, за винятком Scala. Однак він назвав повільний час складання Scala як недолік. [16] Однією з заявлених цілей Kotlin є компіляція так само швидко, як Java. У лютому 2012 року JetBrains відкрила проект за ліцензією Apache 2. [17]

Назва походить від острова Котлін, поблизу Санкт-Петербурга. Андрій Бреслав згадав, що команда вирішила назвати його на честь острова, подібно до того, як Java була названа на честь індонезійського острова Ява [18] (хоча мова програмування Java, можливо, була названа на честь кави) [19].

JetBrains сподівається, що нова мова буде стимулювати продажі IntelliJ IDEA. [20]

Kotlin v1.0 вийшов 15 лютого 2016 р. [21] Це вважається першим офіційно стабільним випуском, і JetBrains взяв на себе зобов'язання щодо довгострокової зворотної сумісності, починаючи з цієї версії.

На Google I / O 2017 Google оголосив про першокласну підтримку Kotlin для Android. [22]

Kotlin v1.2 вийшов 28 листопада 2017 р. [23] До цього випуску нещодавно додано функцію спільного використання коду між платформами JVM та JavaScript (станом на версію 1.4 багатоплатформене програмування є альфа-функцією [24], оновленою з "експериментальної"). З новим плагіном Kotlin / JS Gradle було зроблено демонстрацію з повним стеком. [25] [26]

Kotlin v1.3 був випущений 29 жовтня 2018 року, пропонуючи програми для асинхронного програмування.

7 травня 2019 року Google оголосив, що мова програмування Kotlin тепер є найкращою мовою для розробників додатків для Android. [11]

Kotlin v1.4 вийшов у серпні 2020 р., Напр. деякі незначні зміни щодо підтримки платформ Apple, тобто взаємодії Objective-C / Swift. [27]

Керівник розвитку Андрій Бреслав заявив, що Kotlin розроблений як об'єктно-орієнтована мова промислової потужності та "краща мова", ніж Java, але при цьому повністю взаємодіє з кодом Java, що дозволяє компаніям здійснювати поступову міграцію з Java на Kotlin . [28]

Крапка з комою необов'язкова як термінатор твердження; у більшості випадків нового рядка достатньо для того, щоб компілятор зміг зробити висновок про те, що оператор закінчився.

У оголошеннях змінних Kotlin та списках параметрів тип даних постає після імені змінної (і з роздільником двокрапки), подібно до BASIC, Pascal та TypeScript.

Змінні в Kotlin можуть бути лише для читання, оголошені за допомогою ключового слова `val` або змінні, оголошені за допомогою ключового слова `var`. [30]

Учасники класу за замовчуванням є загальнодоступними, а самі класи за замовчуванням остаточно, що означає, що створення похідного класу вимкнено, якщо базовий клас не оголошено за допомогою ключового слова `open`.

На додаток до класів та функцій-членів (еквівалентних методам) об'єктно-орієнтованого програмування, Kotlin також підтримує процедурне програмування з використанням функцій. [31] Функції Kotlin (і конструктори) підтримують аргументи за замовчуванням, списки аргументів змінної довжини, іменовані аргументи та перевантаження унікальним підписом. Функції-члени класу є віртуальними, тобто відправляються на основі типу середовища виконання об'єкта, до якого вони викликаються.

Kotlin 1.3 додає (стабільну в `stdlib`; експерименти, що визначаються користувачем), підтримку контрактів [32] (натхненний парадигмою програмування Ейфеля за контрактом [33])

На думку розробників Kotlin, ви можете викликати код JavaScript з Kotlin, напр. писати повні, безпечні для використання програми React або писати та підтримувати повнофункціональні веб-програми, що використовують спільну логіку перевірки з інтерфейсом, або ви можете "генерувати бібліотеки з коду Kotlin, які можна використовувати як модулі з будь-якої бази коду, написаної на JavaScript або TypeScript ". [34]

Котлін послаблює обмеження Java щодо дозволу статичним методам і змінним існувати лише в тілі класу. Статичні об'єкти та функції можна визначити на верхньому рівні пакету, не потребуючи надмірного рівня класу. Для сумісності з Java Kotlin надає анотацію `JvmName`, яка визначає ім'я класу, що використовується під час перегляду пакета з проекту Java. Наприклад, `@file: JvmName("JavaClassName")`.

Коли Kotlin було оголошено офіційною мовою розробки Android на Google I / O у травні 2017 року, вона стала третьою мовою, яка повністю підтримується для Android, крім Java та C ++. [47] Станом на 2020 рік Kotlin все ще найбільш широко використовується на Android, за оцінками Google, 70% з 1000 найкращих додатків у Play Store написані в Kotlin. Сама Google має 60 програм, написаних на Kotlin, включаючи Карти та Диск. Багато програм для Android, наприклад, Google Home, перебувають у процесі міграції до Kotlin, тому використовують і Kotlin, і Java. Kotlin на Android розглядається як вигідний завдяки своїй безпеці нульових покажчиків, а також завдяки своїм функціям, які роблять коротший і читабельніший код. [48]

На додаток до свого видатного використання на Android, Kotlin набирає популярності у розробці на стороні сервера. Spring Framework офіційно додав підтримку Kotlin версією 5.4 січня 2017 р. [49] Для подальшої підтримки Kotlin, Spring переклав всю свою документацію на Kotlin і додав вбудовану підтримку для багатьох специфічних для Kotlin функцій, таких як програми. [50] На додаток до Spring, JetBrains випустив перший для Kotlin фреймворк під назвою Ktor для створення веб-додатків. [51]

У 2020 році JetBrains в опитуванні розробників, які використовують Kotlin, виявив, що 56% використовують Kotlin для мобільних додатків, тоді як 47% використовують його для веб-сервісу. Трохи більше третини всіх розробників Kotlin сказали, що вони мігрують до Kotlin з іншої мови. Більшість користувачів Kotlin націлювались на Android (або іншим чином на JVM), лише 6% використовували Kotlin Native. [52]

2.2. Середовище розробки Android Studio

Android Studio є офіційним [7] інтегрованим середовищем розробки (IDE) для операційної системи Android від Google, побудованою на програмному забезпеченні IntelliJ IDEA від JetBrains і розробленою спеціально для розробки Android. [8] Він доступний для завантаження в операційних системах на базі Windows, macOS та Linux або як послуга на основі підписки в 2020 році. [9] [10] Це заміна Eclipse Android Development Tools (E-ADT) як основної IDE для розробки власних додатків Android.

Android Studio було оголошено 16 травня 2013 р. На конференції Google I/O. Це було на етапі попереднього перегляду, починаючи з версії 0.1 у травні 2013 року, потім увійшло в бета-версію, починаючи з версії 0.8, яка вийшла в червні 2014 року [11] Перша стабільна збірка була випущена в грудні 2014 року, починаючи з версії 1.0. [12]

7 травня 2019 року Котлін замінив Java як переважну мову Google для розробки додатків для Android. [13] Java як і раніше підтримується, як і C++. [14]

Специфічною особливістю Android Studio є відсутність можливості вимкнення функції автоматичного збереження. [15]

У поточній стабільній версії представлені такі функції: [16] [17]

- Підтримка збірки на основі Gradle
- Спеціальний рефакторинг для Android та швидкі виправлення
- Інструменти Lint для виявлення продуктивності, зручності використання, сумісності версій та інших проблем
- Можливості інтеграції та підписання програм ProGuard
- Майстри на основі шаблонів для створення загальних конструкцій та компонентів Android

- Розширений редактор макетів, що дозволяє користувачам перетягувати компоненти інтерфейсу, можливість попереднього перегляду макетів на декількох конфігураціях екрана [18]
- Підтримка для створення додатків Android Wear
- Вбудована підтримка Google Cloud Platform, що забезпечує інтеграцію з Firebase Cloud Messaging (раніше "Google Cloud Messaging") та Google App Engine [19]
- Віртуальний пристрій Android (емулятор) для запуску та налагодження програм у студії Android.

Android Studio підтримує ті самі мови програмування IntelliJ (і CLion), наприклад Java, C ++ та інші з розширеннями, такими як Go; Android Studio 3.0 або пізнішої версії, підтримує Kotlin [21] та "всі функції мови Java 7 та підмножини функцій мови Java 8, які залежать від версії платформи." [22] Зовнішні проекти підтримують деякі функції Java 9. [23] Хоча IntelliJ заявляє, що Android Studio підтримує всі випущені версії Java і Java 12, незрозуміло, до якого рівня Android Studio підтримує версії Java до Java 12 (в документації згадується часткова підтримка Java 8). Принаймні деякі нові мовні функції до Java 12 можна використовувати в Android. [24]

Після того, як додаток скомпільовано з Android Studio, його можна опублікувати в Google Play Store. Додаток має відповідати політиці щодо вмісту розробника Google Play Store.

2.3. Android OS

Android - це мобільна операційна система, заснована на модифікованій версії ядра Linux та іншого програмного забезпечення з відкритим кодом, призначена в основному для сенсорних мобільних пристроїв, таких як смартфони та планшети. Android розробляється консорціумом розробників, відомим як Open Handset Alliance, і комерційно фінансується Google. Він був представлений у листопаді 2007 року, а перший комерційний пристрій Android був запущений у вересні 2008 року.

Це безкоштовне програмне забезпечення з відкритим кодом; його вихідний код відомий як Android Open Source Project (AOSP), який в основному ліцензується за ліцензією Apache. Однак більшість пристроїв Android постачаються з попередньо встановленим запатентованим програмним забезпеченням [11], особливо Google Mobile Services (GMS) [12], яке включає такі основні програми, як Google Chrome, платформа цифрового розповсюдження Google Play та пов'язана з ними платформа розробки служб Google Play. Близько 70 відсотків смартфонів Android управляють екосистемою Google; [13] конкуруючі екосистеми та форки Android включають Fire OS (розроблена Amazon) або LineageOS. Однак назва та логотип "Android" є товарними знаками Google, які встановлюють стандарти щодо обмеження використання "несертифікованих" пристроїв за межами їх екосистеми для використання бренду Android.

Вихідний код був використаний для розробки варіантів Android для цілого ряду іншої електроніки, таких як ігрові консолі, цифрові камери, портативні медіаплеєри, ПК та інші, кожен зі спеціалізованим інтерфейсом користувача. Деякі відомі похідні версії включають Android TV для телевізорів та Wear OS для переносних пристроїв, розроблені Google. Пакети програм на Android, які використовують формат APK, зазвичай розповсюджуються через власні магазини додатків, такі як Google Play Store,

Samsung Galaxy Store, Huawei AppGallery, Cafe Bazaar та GetJar, або платформи з відкритим кодом, такі як Aptoide або F-Droid.

Android є найбільш продаваною ОС у світі на смартфонах з 2011 року, а на планшетах - з 2013 року. Станом на травень 2017 року вона має понад два мільярди активних користувачів щомісяця, найбільшу встановлену базу будь-якої операційної системи, а станом на січень 2021 року Google Play Store має понад 3 мільйони програм. [16] Поточна стабільна версія - Android 11, випущена 8 вересня 2020 року.

Android Inc. була заснована в Пало-Альто, штат Каліфорнія, в жовтні 2003 року Енді Рубіном, Річ Майнером, Ніком Сірсом та Крісом Уайтом. [17] [18] Рубін описав проект Android як "величезний потенціал у розробці розумніших мобільних пристроїв, які більше знають про місцезнаходження та уподобання власника" [18]. Першими намірами компанії була розробка вдосконаленої операційної системи для цифрових камер, і це було основою її презентації для інвесторів у квітні 2004 р. [19] Потім компанія вирішила, що ринок камер недостатньо великий для своїх цілей, і через п'ять місяців вона перенаправила свої зусилля і випустила Android як операційну систему для мобільних телефонів, яка буде конкурувати з Symbian та Microsoft Windows Mobile [19] [20].

Рубін рано відчував труднощі із залученням інвесторів, і Android зіткнувся з виселенням з офісних приміщень. Стів Перлман, близький друг Рубіна, приніс йому 10000 доларів готівкою в конверті і незабаром після цього передав нерозголошену суму як насіння. Перлман відмовився від частки в компанії і заявив: "Я зробив це, тому що вірив у цю справу і хотів допомогти Енді" [21] [22].

У липні 2005 року [18] Google придбав Android Inc. щонайменше на 50 мільйонів доларів. [23] Ключові співробітники, включаючи Рубін, Майнер, Сірс та Уайт, приєдналися до Google як частина придбання. [18] На той час про секретну компанію Android Inc. було відомо небагато, оскільки компанія надала лише кілька деталей, крім того, що вона робить програмне

забезпечення для мобільних телефонів. [18] В Google команда під керівництвом Рубіна розробила платформу мобільних пристроїв на базі ядра Linux. Google продавав платформу для виробників мобільних телефонів та операторів, обіцяючи забезпечити гнучку та модернізовану систему. [24] Google "вибудував ряд апаратних компонентів та програмних партнерів і проінформував перевізників про те, що він відкритий для різного рівня співпраці"

Спекуляції щодо наміру Google вийти на ринок мобільного зв'язку продовжували наростати до грудня 2006 р. [26] Ранній прототип був дуже схожий на телефон BlackBerry, без сенсорного екрану і фізичної клавіатури QWERTY, але прихід Apple iPhone 2007 року означав, що Android "повинен був повернутися до креслення" [27] [28]. Пізніше Google змінив свої специфікаційні документи для Android, зазначивши, що "сенсорні екрани будуть підтримуватися", хоча "Продукт був розроблений з наявністю дискретних фізичних кнопок як припущення, тому сенсорний екран не може повністю замінити фізичні кнопки". [29] До 2008 року як Nokia, так і BlackBerry оголосили про сенсорні смартфони, щоб конкурувати з iPhone 3G, і фокус Android з часом перейшов лише на сенсорні екрани. Першим комерційно доступним смартфоном під управлінням Android став HTC Dream, також відомий як T-Mobile G1, оголошений 23 вересня 2008 р. [30] [31]

5 листопада 2007 року було відкрито Open Handset Alliance - консорціум технологічних компаній, включаючи Google, виробників пристроїв, таких як HTC, Motorola і Samsung, операторів бездротового зв'язку, таких як Sprint і T-Mobile, і виробників чіпсетів, таких як Qualcomm і Texas Instruments. з метою розробки "першої по-справжньому відкритої та всеосяжної платформи для мобільних пристроїв". [32] [33] [34] Протягом року Open Handset Alliance зіткнувся з двома іншими конкурентами з відкритим кодом, Symbian Foundation та LiMo Foundation, останні також розробили мобільну операційну систему на базі Linux, таку як Google. У вересні 2007 року Інформаційний

тиждень висвітлив дослідження Evaluateserve, в якому повідомлялося, що Google подав кілька заявок на патенти в галузі мобільної телефонії. [35] [36]

Починаючи з 2008 року, Android спостерігав численні оновлення, які поступово вдосконалювали операційну систему, додаючи нові функції та виправляючи помилки в попередніх випусках. Кожен основний випуск названий в алфавітному порядку після десерту чи цукристого ласощі, причому перші кілька версій Android називаються "Кекс", "Пончик", "Еклер" та "Фройо", у такому порядку. Під час анонсу Android KitKat у 2013 році Google пояснив, що "оскільки ці пристрої роблять наше життя таким солодким, кожна версія Android названа на честь десерту", хоча представник Google заявив CNN в інтерв'ю, що "Це як внутрішня команда річ, і ми воліємо бути трішки - як би я сказав - трохи невпевненими в цьому питанні, я скажу ". [37]

У 2010 році Google випустив свою серію пристроїв Nexus - лінійку, в якій Google співпрацював з різними виробниками пристроїв для виробництва нових пристроїв та представлення нових версій Android. Серія була описана як "відіграла ключову роль в історії Android, впроваджуючи нові ітерації програмного забезпечення та апаратні стандарти", і стала відомою завдяки своєму програмному забезпеченню "без роздуття" із "своєчасними ... оновленнями". [38] На своїй конференції розробників у травні 2013 року Google анонсував спеціальну версію Samsung Galaxy S4, де замість використання власної налаштування Android від Samsung телефон працював «на складі Android» і обіцяли швидко отримувати нові оновлення системи [39]. Пристрій стане початком програми Google Play Edition, а за ним підуть інші пристрої, включаючи HTC One Google Play Edition [40] та Moto G Google Play [41]. У 2015 році Ars Technica писав, що "На початку цього тижня останній з телефонів Android у версії Google Play в Інтернет-магазині Google був перерахований як" більше не доступний для продажу "і що" Тепер їх немає, і це виглядає цілим подібно до завершення програми ". [42] [43]

З 2008 по 2013 рік Уго Барра працював представником продукту, представляючи Android на прес-конференціях та Google I / O, щорічній

конференції Google, орієнтованій на розробників. У серпні 2013 року він покинув Google, щоб приєднатися до китайського виробника телефонів Xiaomi. [44] [45] Менш ніж за півроку до того часу тодішній генеральний директор Google Ларрі Пейдж оголосив у своєму повідомленні в блозі, що Енді Рубін перейшов з підрозділу Android, щоб брати участь у нових проектах у Google, і що Сундар Пічай стане новим керівником Android. [46] [47] Зрештою сам Пічай змінив посаду, ставши новим генеральним директором Google у серпні 2015 року після реструктуризації компанії в конгломерат Alphabet [48] [49], зробивши Хіросі Локхаймера новим керівником Android. [50] [51]

В Android 4.4 Kit Kat спільний доступ до записів на картки пам'яті MicroSD заблоковано для встановлених користувачем програм, до яких залишаються доступними лише спеціальні каталоги з відповідними іменами пакетів, що знаходяться всередині Android / data /. Доступ до запису було відновлено за допомогою Android 5 Lollipop через несумісний назад інтерфейс Google Storage Access Framework. [52]

У червні 2014 року Google оголосив про Android One, набір "апаратних еталонних моделей", який "дозволить [виробникам пристроїв] легко створювати високоякісні телефони за низькі витрати", призначений для споживачів у країнах, що розвиваються. [53] [54] [55] У вересні Google оголосив про випуск першого набору телефонів Android One в Індії. [56] [57] Однак у червні 2015 року Recode повідомив, що проект "розчарував", посиляючись на "неохочих споживачів та виробничих партнерів" та "осічки пошукової компанії, яка ніколи не зламала обладнання" [58]. Плани відновити Android One з'явилися в серпні 2015 року [59], а Африка оголосила наступним місцем для програми через тиждень. [60] [61] У звіті видання The Information у січні 2017 року зазначається, що Google розширює свою недорогу програму Android One на США, хоча The Verge зазначає, що компанія, ймовірно, не буде виробляти фактичні пристрої сама. [62] [63]

Google представив смартфони Pixel і Pixel XL у жовтні 2016 року, що продаються як перші телефони, виготовлені Google, [64] [65], і до випуску

ширшого випуску пропонуються виключно деякі програмні функції, такі як Google Assistant [66] [67].] Телефони Pixel замінили серію Nexus [68] новим поколінням телефонів Pixel, випущених у жовтні 2017 року. [69]

У травні 2019 року операційна система заплуталась у торговій війні між Китаєм та США за участю компанії Huawei, яка, як і багато інших технологічних фірм, потрапила в залежність від доступу до платформи Android. [70] [71] Влітку 2019 року Huawei оголосила, що створить альтернативну операційну систему Android [72] [73], відому як Harmony OS, [74], і подала заявку на права інтелектуальної власності на великих світових ринках. [75] [76] В даний час Huawei не планує замінювати Android найближчим часом, оскільки Harmony OS розроблена для пристроїв Інтернету речей, а не для смартфонів. [77]

22 серпня 2019 року було оголошено, що Android "Q" буде офіційно затаврований як Android 10, закінчуючи історичну практику називання основних версій десертами. Google заявив, що ці назви не є "всеохоплюючими" для міжнародних користувачів (через те, що вищезазначені продукти не є міжнародно відомими, або їх важко вимовити на деяких мовах). Того ж дня поліція Android повідомила, що Google замовив встановлення статуї гігантського числа "10" у вестибюлі нового офісу розробників. [80] Android 10 був випущений 3 вересня 2019 року для телефонів Google Pixel.

З обсягом сховища звичайний доступ до письма до спільної внутрішньої пам'яті користувача заблоковано, і доступними, як завжди, залишаються лише конкретні каталоги програм. Файли та каталоги зовні залишаються доступними лише через несумісну назад фреймворк Storage Access. Незважаючи на те, що ці обмеження, як стверджується, покращують конфіденційність користувачів, приватні каталоги для конкретних додатків вже існували в / data / з ранніх версій операційної системи. [81]

Користувальницький інтерфейс Android за замовчуванням в основному заснований на прямих маніпуляціях, використовуючи сенсорні входи, які

вільно відповідають дійсним діям, таким як проведення, натискання, стискання та зворотне стискання для маніпулювання об'єктами на екрані, разом з віртуальною клавіатурою [82]. Ігрові контролери та повнорозмірні фізичні клавіатури підтримуються через Bluetooth або USB. [83] [84] Відповідь на введення користувачем розроблена негайно і забезпечує плавний сенсорний інтерфейс, часто використовуючи вібраційні можливості пристрою для забезпечення тактильного зворотного зв'язку з користувачем. Внутрішнє обладнання, таке як акселерометри, гіроскопи та датчики наближення, використовується деякими програмами для реагування на додаткові дії користувача, наприклад, регулювання екрану від книжкової до альбомної залежно від того, як орієнтований пристрій, [85] або дозволяючи користувачеві керувати транспортною засобу в гоночній грі, обертаючи пристрій, імітуючи управління кермом. [86]

Пристрої Android завантажуються на головний екран, основний навігаційно-інформаційний «хаб» на пристроях Android, аналогічно робочому столу, що знаходиться на персональних комп'ютерах. Домашні екрани Android зазвичай складаються з піктограм додатків та віджетів; піктограми додатків запускають відповідний додаток, тоді як віджети відображають оновлюваний вміст, що автоматично оновлюється, наприклад прогноз погоди, поштову скриньку користувача чи індикатор новин безпосередньо на робочому столі. Домашній екран може складатися з декількох сторінок, між якими користувач може проводити пальцем вперед і назад [88]. Додатки сторонніх розробників, доступні в Google Play та інших магазинах додатків, можуть широко переформатувати тему головного екрана [89] і навіть імітувати зовнішній вигляд інших операційних систем, таких як Windows Phone. [90] Більшість виробників пристосовують зовнішній вигляд та функції своїх пристроїв Android, щоб відрізнитись від своїх конкурентів. [91]

У верхній частині екрана знаходиться рядок стану, що відображає інформацію про пристрій та його зв'язок. Цей рядок стану можна потягнути (проведіть) вниз, щоб відкрити екран сповіщень, де програми відображають

важливу інформацію або оновлення, а також швидкий доступ до системних елементів керування та перемикань, таких як яскравість дисплея, налаштування підключення (WiFi, Bluetooth, стільникові дані), аудіорежим та ліхтарик. [88] Постачальники можуть застосувати розширені налаштування, такі як можливість регулювання яскравості ліхтарика. [92]

Програми ("програми"), які розширюють функціональність пристроїв (і повинні бути 64-розрядними [105]), створюються за допомогою набору для розробки програмного забезпечення Android (SDK) [106] і, часто, мови програмування Kotlin, яка замінила Java як бажана мова Google для розробки додатків для Android у травні 2019 р. [107], і спочатку була оголошена в травні 2017 р. [108] [109] Досі підтримується Java (спочатку єдина опція для програм простору користувача, і її часто змішують з Kotlin), як і C ++. [110] Java та / або інші мови JVM, такі як Kotlin, можуть поєднуватися з C / C ++, [111] разом із вибором стандартних середовищ виконання, які забезпечують кращу підтримку C ++. [112] Мова програмування Go також підтримується, хоча з обмеженим набором інтерфейсів прикладного програмування (API). [113]

SDK включає повний набір засобів розробки [114], включаючи налагоджувач, бібліотеки програмного забезпечення, емулятор слухавки на основі QEMU, документацію, зразок коду та навчальні посібники. Спочатку підтримуваним Google інтегрованим середовищем розробки (IDE) було Eclipse з використанням плагіна Android Development Tools (ADT); У грудні 2014 року Google випустив Android Studio, засновану на IntelliJ IDEA, як основну IDE для розробки додатків для Android. Доступні інші засоби розробки, включаючи власний набір для розробки (NDK) для додатків або розширень на C або C ++, Google App Inventor, візуальне середовище для початківців програмістів та різні платформи для мобільних веб-додатків. У січні 2014 року Google представив основу на основі Apache Cordova для перенесення веб-додатків Chrome HTML 5 на Android, загорнуті в оболонку власного додатка. [115] Крім того, Google придбав Firebase у 2014 році, що надає корисні інструменти для розробників програм та веб-розробників. [116]

Android має все більший вибір сторонніх додатків, які користувачі можуть придбати, завантаживши та встановивши файл APK програми (пакет додатків Android), або завантаживши їх за допомогою програми зберігання програм, яка дозволяє користувачам встановлювати, оновлювати та видаляти програми зі своїх пристроїв. Google Play Store - це основний магазин додатків, встановлений на пристроях Android, які відповідають вимогам сумісності Google і ліцензують програмне забезпечення Google Mobile Services. [117] [118] Google Play Store дозволяє користувачам переглядати, завантажувати та оновлювати програми, опубліковані Google та сторонніми розробниками; станом на січень 2021 року в Play Store доступно понад три мільйони додатків для Android. [16] [119] Станом на липень 2013 року було встановлено 50 мільярдів додатків. [120] [121] Деякі перевізники пропонують пряму оплату через оператора за покупки додатків Google Play, де вартість програми додається до щомісячного рахунку користувача. [122] Станом на травень 2017 року щомісяця зареєстровано понад мільярд активних користувачів Gmail, Android, Chrome, Google Play і Maps.

Через відкритий характер Android, для Android існує також низка сторонніх ринків програм, або для того, щоб замінити пристрої, яким заборонено поставлятися із Google Play Store, або пропонувати програми, які не можуть бути запропоновані в Google Play Store через до порушень політики або з інших причин. Прикладами цих сторонніх магазинів є Amazon Appstore, GetJar та SlideMe. F-Droid, інший альтернативний ринок, прагне надавати лише програми, які розповсюджуються за безкоштовними та відкритими ліцензіями. [117] [123] [124] [125]

У жовтні 2020 року Google видалив із додатка Play Store кілька програм для Android, оскільки було виявлено, що вони порушують правила збору даних. Міжнародна рада з питань цифрової звітності (IDAC) повідомила фірму, що додатки для дітей, такі як Number Coloring, Princess Salon та Cats & Cosplay, із загальним завантаженням 20 мільйонів, порушують політику Google. [126]

Основною апаратною платформою для Android є ARM (архітектури ARMv7 та ARMv8-A), архітектури x86 та x86-64 також офіційно підтримуються в пізніших версіях Android. [135] [136] [137] Неофіційний проект Android-x86 забезпечив підтримку архітектур x86 напередодні офіційної підтримки. [138] [139] З 2012 року почали з'являтися пристрої Android з процесорами Intel, включаючи телефони [140] та планшети. Отримавши підтримку 64-розрядних платформ, Android спочатку працював на 64-розрядному x86, а потім на ARM64. Оскільки Android 5.0 "Lollipop", на додаток до 32-розрядних варіантів підтримуються 64-розрядні варіанти всіх платформ. [135] Неофіційний експериментальний порт операційної системи до архітектури RISC-V був випущений в 2021 році [141].

Вимоги до мінімального обсягу оперативної пам'яті для пристроїв під управлінням Android 7.1 коливаються на практиці від 2 ГБ для найкращого обладнання та до 1 ГБ для найбільш поширеного екрану. Android підтримує всі версії OpenGL ES та Vulkan (і версія 1.1 доступна для деяких пристроїв [142]).

Пристрої Android включають безліч додаткових апаратних компонентів, включаючи нерухомі або відеокамери, GPS, датчики орієнтації, спеціальні ігрові елементи управління, акселерометри, гіроскопи, барометри, магнітометри, датчики наближення, датчики тиску, термометри та сенсорні екрани. Деякі апаратні компоненти не потрібні, але стали стандартними для певних класів пристроїв, таких як смартфони, і додаткові вимоги застосовуються, якщо вони є. Спочатку потрібно було деяке інше обладнання, але ці вимоги були послаблені або взагалі ліквідовані. Наприклад, оскільки Android спочатку розроблявся як ОС телефону, потрібно було таке обладнання, як мікрофони, тоді як з часом функція телефону стала не обов'язковою. Раніше Android вимагав камери з автофокусом, яка була спокійно встановлена на камеру з фіксованим фокусом [121], якщо вона взагалі була присутня, оскільки камера відмовлялася як вимога повністю, коли Android починав використовуватися на приставках.

Окрім того, що кілька смартфонів та планшетів працюють на смартфонах і планшетах, кілька постачальників працюють під управлінням Android на звичайному обладнанні ПК за допомогою клавіатури та миші. [143] [144] [145] [146] На додаток до їх доступності на комерційному обладнанні, подібні версії Android, зручні для апаратного забезпечення для ПК, є у вільному доступі з проекту Android-x86, включаючи налаштований Android 4.4. [147] Використовуючи емулятор Android, який є частиною Android SDK, або сторонні емулятори, Android також може працювати не нативно на архітектурах x86. [148] [149] Китайські компанії будують ПК та мобільну операційну систему на базі Android, щоб "безпосередньо конкурувати з Microsoft Windows та Google Android". [150] Китайська інженерна академія зазначила, що "більше десятка" компаній налаштовували Android після китайської заборони на використання Windows 8 на державних ПК. [151] [152] [153]

2.4. Firebase

Firebase - це платформа, розроблена Google для створення мобільних та веб-додатків. Спочатку це була незалежна компанія, заснована в 2011 році. [1] У 2014 році Google придбав платформу [2], і тепер вона є їхньою флагманською пропозицією для розробки додатків.

Firebase перетворився на Envolve, попередній стартап, заснований Джеймсом Тампліном та Ендрю Лі у 2011 році. Envolve надав розробникам API, що дозволяє інтегрувати функціональність онлайн-чату на їх веб-сайти. Випустивши службу чату, Тамплін і Лі виявили, що вона використовується для передачі даних програми, які не були повідомленнями чату. Розробники використовували Envolve для синхронізації даних користувачів, таких як стан гри, у режимі реального часу між своїми користувачами. Тамплін і Лі вирішили розділити систему чату та архітектуру в режимі реального часу, яка

її використовувала. [3] Вони заснували Firebase як окрему компанію у вересні 2011 р. [4], а вона була відкрита для громадськості у квітні 2012 р. [5]

Першим продуктом Firebase стала Firebase Realtime Database, API, який синхронізує дані додатків на iOS, Android та веб-пристроях та зберігає їх у хмарі Firebase. Продукт допомагає розробникам програмного забезпечення у створенні спільних додатків у реальному часі.

У травні 2012 року, через місяць після запуску бета-версії, Firebase залучив 1,1 млн. Доларів на початкове фінансування від венчурних капіталістів Flybridge Capital Partners, Greylock Partners, Founder Collective та New Enterprise Associates. [6] У червні 2013 року компанія додатково залучила 5,6 мільйона доларів на фінансування серії А від Union Square Ventures та Flybridge Capital Partners. [7]

У 2014 році Firebase випустила два продукти. Хостинг Firebase [8] та автентифікація Firebase. [9] Це позиціонувало компанію як мобільний сервер як послугу.

У жовтні 2014 року Google придбав Firebase [10]. Через рік, у жовтні 2015 року, Google придбав Divshot, платформу веб-хостингу HTML5, щоб об'єднати її з командою Firebase. [11]

У травні 2016 року на Google I / O, щорічній конференції розробників компанії, Firebase представив Firebase Analytics і оголосив, що розширює свої послуги, щоб стати уніфікованою платформою серверного сервісу (BaaS) для розробників мобільних пристроїв. Зараз Firebase інтегрується з різними іншими сервісами Google, зокрема Google Cloud Platform, AdMob та Google Ads, пропонуючи ширші продукти та масштаби для розробників. [12] Google Cloud Messaging, сервіс Google для надсилання push-сповіщень на пристрої Android, був замінений продуктом Firebase, Firebase Cloud Messaging, який додав функціональність доставки push-повідомлень як на iOS, так і на веб-пристрої. У січні 2017 року Google придбав Fabric and Crashlytics у Twitter, щоб додати ці служби до Firebase. [13] [14]

У жовтні 2017 року Firebase запустила Cloud Firestore, базу даних у режимі реального часу, як продукт-наступник оригінальної бази даних Firebase Realtime. [15] [16] [17] [18]

Стверджується, що Firebase використовує Google для відстеження користувачів без їх відома. 14 липня 2020 року було подано позов, в якому звинувачували Google у порушенні федерального закону про провідну мережу та закону про конфіденційність у Каліфорнії. У ній зазначалося, що за допомогою Firebase Google збирав та зберігав дані користувачів, реєструючи те, що користувач переглядав у багатьох типах програм, незважаючи на те, що користувач виконував власні вказівки Google щодо відключення роботи в Інтернеті та додатках, зібраних компанією. [20]

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Діаграма варіантів використання

Діаграма використання найпростіша - це представлення взаємодії користувача із системою, яка показує взаємозв'язок між користувачем та різними випадками використання, в яких користувач бере участь. Діаграма випадків використання може ідентифікувати різні типи користувачів системи та різні випадки використання, і часто вона супроводжується також іншими типами діаграм. Варіанти використання представлені кругами або еліпсами.

Незважаючи на те, що сам випадок використання може детально вивчити кожну можливість, діаграма прикладів використання може допомогти забезпечити огляд системи на більш високому рівні. Раніше вже було сказано, що "схеми використання - це принципи вашої системи".

Через їх спрощений характер, схеми використання можуть бути хорошим інструментом комунікації для зацікавлених сторін. Креслення намагаються імітувати реальний світ і дають зацікавленій стороні уявлення про те, як буде розроблена система. Сіау та Лі провели дослідження, щоб визначити, чи взагалі існувала дійсна ситуація для схем використання або вони були непотрібними. Було виявлено, що діаграми випадків використання передають намір системи більш спрощеним чином зацікавленим сторонам і що вони "інтерпретуються більш повно, ніж діаграми класів".

Метою діаграми використання є відображення динамічного аспекту системи. Додаткові схеми та документація можуть бути використані для забезпечення повного функціонального та технічного уявлення про систему. Вони забезпечують спрощене та графічне представлення того, що система насправді повинна робити.

Елементи:

- рамки системи (англ. system border) - прямокутник із назвою у верхніх частинах та еліпсами (прецедентами) всередині. Часто може бути опущено без корисної інформації про полезну інформацію,
- актор (англ. actor) - стилізований людський персонаж, обзначаючий набір ролей користувача (розуміється в широкому змісті: людина, зовнішня сутність, клас, інша система), взаємодіючого з деякою сутністю (системною, підсистемою, класом). Актори не можуть бути пов'язані між собою з іншим (за вимкнення відносин щодо обробки / дослідження),
- прецедент - еліпс із надписом, що означає виконувану систематичну дію (може включати можливі варіанти), що призводить до спостережуваних акторами результатів. Надпис може бути ім'ям або описом (з точки зору актора) того, "що" робить система (а не "як"). Ім прецедента зв'язано з неперервним (атомарним) сценарієм - конкретною послідовністю дій, ілюструючою поведінку. Під час сценарію актори обмінюються із систематичними повідомленнями. Сценарій може бути приведений на діаграмі прецедентів у відео UML-коментарі. З одним прецедентом може бути пов'язано кілька різних сценаріїв

На рисунку 3.1 зображено діаграму варіантів використання, яка описує можливі дії користувача в системі.

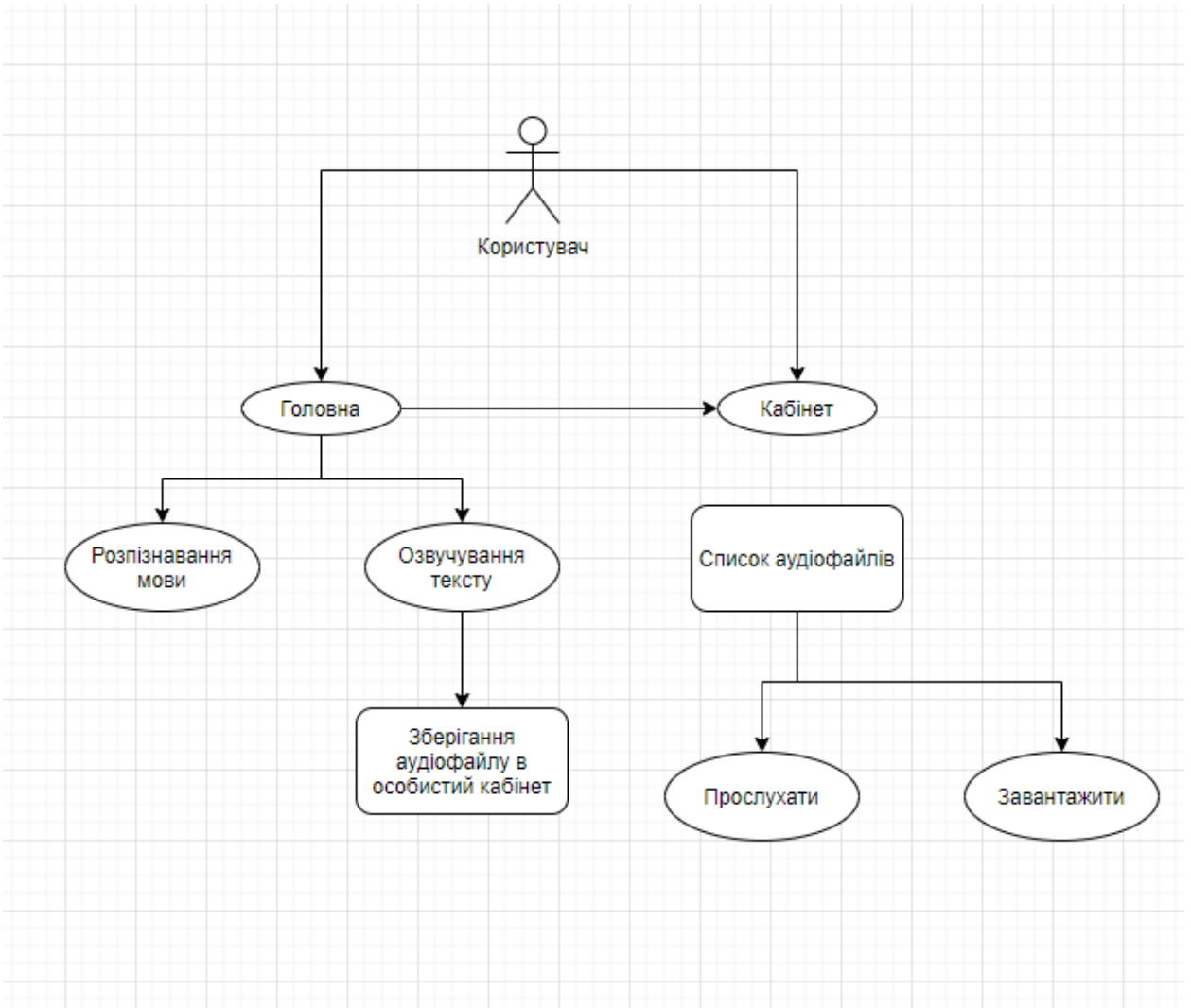


Рисунок 3.1 — Діаграма варіантів використання

3.2. Діаграма класів програмного продукту

У програмній інженерії діаграма класів в Уніфікованій мові моделювання (UML) - це тип статичної структурної діаграми, що описує структуру системи, показуючи класи системи, їх атрибути, операції (або методи) та взаємозв'язки між об'єктами.

Діаграма класів є основним будівельним елементом об'єктно-орієнтованого моделювання. Він використовується для загального концептуального моделювання структури програми та для детального моделювання переведення моделей у програмовий код. Діаграми класів також можуть бути використані для моделювання даних. Класи на діаграмі класів представляють як основні елементи, взаємодії в програмі, так і класи, що програмуються.

На схемі класи представлені вікнами, які містять три відділення:

- У верхньому відділенні міститься назва класу. Надруковано жирним шрифтом і відцентровано, а перша літера написана великими літерами.
- Середній відсік містить атрибути класу. Вони вирівняні за лівим краєм, а перша буква мала.
- У нижньому відділенні містяться операції, які може виконувати клас. Вони також вирівняні за лівим краєм, а перша буква - мала.

При проектуванні системи ряд класів ідентифікується та згруповується у схему класів, яка допомагає визначити статичні відносини між ними. При детальному моделюванні класи концептуального проекту часто поділяються на ряд підкласів.

Залежність - це семантичний зв'язок між залежними та незалежними елементами моделі. Він існує між двома елементами, якщо зміни у визначенні одного елемента (сервера або цілі) можуть спричинити зміни для іншого (клієнта або джерела). Ця асоціація є односпрямованою. Залежність

відображається у вигляді штрихової лінії з відкритою стрілкою, яка вказує від клієнта до постачальника.

Для подальшого опису поведінки систем ці діаграми класів можуть бути доповнені діаграмою стану або машиною стану UML.

Асоціація представляє родину посилань. Двійкова асоціація (з двома кінцями) зазвичай представляється у вигляді рядка. Асоціація може пов'язувати будь-яку кількість класів. Асоціація з трьома ланками називається потрійною асоціацією. Асоціацію можна назвати, а кінці асоціації можна прикрасити іменами ролей, показниками власності, кратністю, видимістю та іншими властивостями.

Існує чотири різні типи асоціацій: двонаправлена, односпрямована, агрегаційна (включає агрегацію композиції) та рефлексивна. Двонаправлені та односпрямовані асоціації є найбільш поширеними.

Наприклад, клас польоту асоціюється з класом літака двонаправлено. Асоціація представляє статичне відношення, яке ділиться між об'єктами двох класів.

Агрегація є варіантом взаємозв'язку "має"; агрегація є більш конкретною, ніж асоціація. Це асоціація, яка представляє частково цілі або часткові стосунки. Як показано на зображенні, професор "має" клас для викладання. Як тип асоціації, агрегація може бути названа та мати ті самі прикраси, що і асоціація. Однак агрегація не може включати більше двох класів; це має бути бінарна асоціація. Крім того, навряд чи існує різниця між агрегаціями та асоціаціями під час реалізації, і діаграма може взагалі пропустити відносини агрегування. [7]

Агрегація може відбуватися, коли клас є колекцією або контейнером інших класів, але вміщені класи не мають сильної залежності життєвого циклу від контейнера. Вміст контейнера все ще існує, коли контейнер знищений.

В UML він графічно представлений у вигляді порожнистої форми ромба на вміщуючому класі одним рядком, що зв'язує його із вміщеним класом. Сукупність - це семантично розширений об'єкт, який у багатьох операціях

тракується як одиниця, хоча фізично він складається з декількох менших об'єктів.

Приклад: Бібліотека та студенти. Тут студент може існувати без бібліотеки, зв'язок між студентом і бібліотекою є агрегацією.

Це вказує на те, що один із двох пов'язаних класів (підклас) вважається спеціалізованою формою іншого (супер тип), а суперклас - узагальненням підкласу. На практиці це означає, що будь-який екземпляр підтипу є також екземпляром суперкласу. Зразкове дерево узагальнень цієї форми зустрічається в біологічній класифікації: людина - це підклас маймуни, який є підкласом ссавців тощо. Зв'язок найлегше зрозуміти за допомогою фрази „А - це В” (людина - це ссавець, ссавець - тварина).

Графічне представлення UML узагальнення - це форма порожнистого трикутника на кінці суперкласу рядка (або дерева рядків), що зв'язує його з одним або кількома підтипами.

Відносини узагальнення також відомі як спадщина або відносини "є".

Суперклас (базовий клас) у відносинах узагальнення також відомий як "батьківський", суперклас, базовий клас або базовий тип.

Підтип у відносинах спеціалізації також відомий як "дочірній", підклас, похідний клас, похідний тип, клас успадкування або тип успадкування.

Зверніть увагу, що ці стосунки нічим не схожі на біологічні стосунки батьків та дітей: використання цих термінів надзвичайно поширене, але може ввести в оману.

А - це тип В

Наприклад, "дуб - це тип дерева", "автомобіль - це тип транспортного засобу"

Узагальнення може бути показано лише на діаграмах класів та на діаграмах використання.

При моделюванні UML взаємозв'язок реалізації - це взаємозв'язок між двома елементами моделі, в яких один елемент моделі (клієнт) реалізує

(реалізує або виконує) поведінку, яку вказує інший елемент моделі (постачальник).

Графічне представлення UML реалізації - це порожниста форма трикутника на кінці інтерфейсу штрихової лінії (або дерева рядків), яка з'єднує її з одним або кількома реалізаторами. Проста головка стрілки використовується на кінці інтерфейсу штрихової лінії, що з'єднує її з користувачами. У діаграмах компонентів використовується графічна умова «м'яч і сокет» (реалізатори виставляють кульку або льодяник, тоді як користувачі показують сокет). Реалізації можна показати лише на діаграмах класів або компонентів. Реалізація - це взаємозв'язок між класами, інтерфейсами, компонентами та пакетами, що з'єднує елемент клієнта з елементом постачальника. Зв'язок реалізації між класами / компонентами та інтерфейсами показує, що клас / компонент реалізує операції, пропонувані інтерфейсом.

Залежність - це слабша форма зв'язку, яка вказує на те, що один клас залежить від іншого, оскільки він використовує його в певний момент часу. Один клас залежить від іншого, якщо незалежний клас є змінною параметра або локальною змінною методу залежного класу. Це відрізняється від асоціації, де атрибут залежного класу є екземпляром незалежного класу. Іноді відносини між двома класами дуже слабкі. Вони взагалі не реалізовані зі змінними-членами. Швидше вони можуть бути реалізовані як аргументи функції-члена.

інший. Ці відносини зазвичай описуються як "А має В" (у матері-кота є кошенята, у кошенят - мати-кішка).

Представлення UML асоціації - це лінія, що з'єднує два пов'язані класи. На кожному кінці рядка є додаткові позначення. Наприклад, ми можемо вказати, використовуючи наконечник стрілки, що загострений кінець видно з хвоста стрілки. Ми можемо вказати власність шляхом розміщення кульки, ролі, яку відіграють елементи цього кінця, вказавши ім'я ролі та множинність

екземплярів цієї сутності (діапазон кількості об'єктів, які беруть участь в асоціації з точки зору іншого кінця).

Класи сутності моделюють довгоживучу інформацію, якою обробляє система, а іноді і поведінку, пов'язану з цією інформацією. Їх не слід ідентифікувати як таблиці баз даних чи інших сховищ даних.

Вони намальовані як кола з короткою лінією, прикріпленою до нижньої частини кола. Як варіант, їх можна намалювати як звичайні класи із позначенням стереотипу «сутність» над назвою класу.

На рисунку 3.2 зображена діаграма класів програмного продукту, яка повністю описує внутрішню структуру проекту.

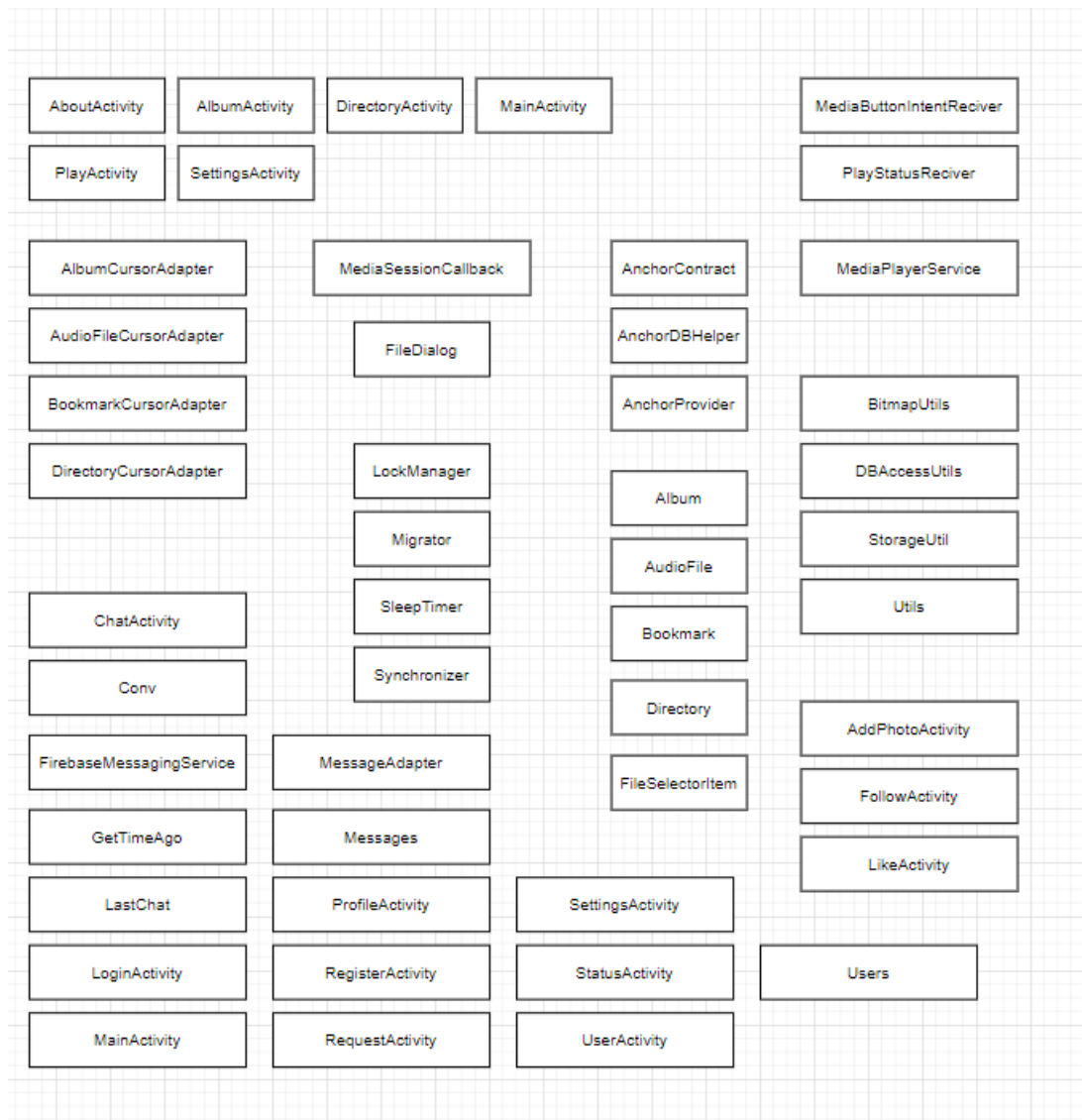


Рисунок 3.2 — Діаграма класів

3.3. Структурна діаграма ІС

На рисунку 3.3 зображена структурна діаграма проекту, яка описує взаємодію між його модулями.

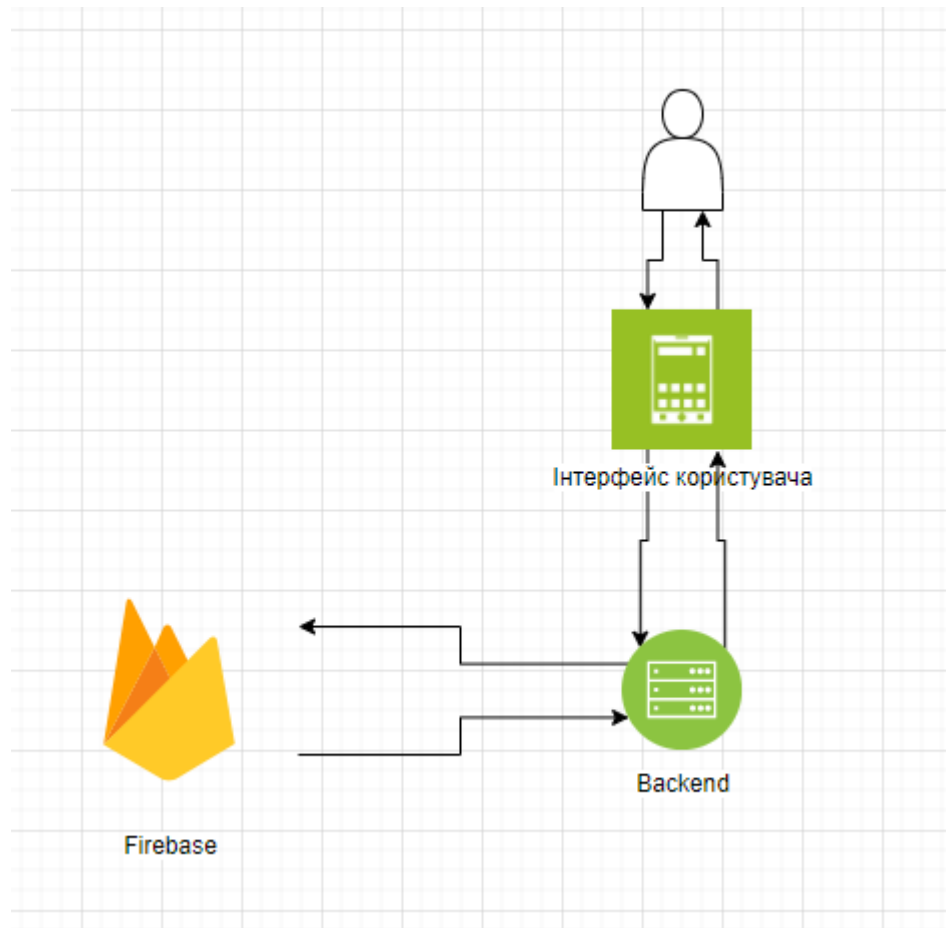


Рисунок 3.3 — Структурна діаграма

ВИСНОВКИ

В ході роботи над даним проектом було виконано наступні завдання:

1. Проаналізувати предметну область.
 - проаналізувати поняття мобільних додатків;
 - провести огляд методів і засобів розробки мобільних додатків;
 - проаналізувати поняття соціальної мережі;
2. Провести огляд обраних засобів реалізації.
 - провести огляд мови програмування Kotlin;
 - провести огляд середовища розробки Android Studio;
 - провести огляд ОС Android;
 - провести огляд хмарного сервісу Firebase;
3. Описати процес розробки програмного забезпечення.
 - створити діаграму варіантів використання;
 - створити діаграму класів програмного продукту;
 - створити структурну діаграму.

Завдяки чіткому виконанню поставлених задач в результаті роботи отримано повноцінний додаток соціальної мережі, які виконує заявлений функціонал і готовий до використання в реальних умовах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Pogue, David (November 4, 2009). "A Place to Put Your Apps". New York Times. Retrieved January 22, 2013.
2. "'App' voted 2010 word of the year by the American Dialect Society (UPDATED) American Dialect Society". Americandialect.org. 2011-01-08. Retrieved 2012-01-28.
3. Siegler, MG (June 11, 2008). "Analyst: There's a great future in iPhone apps". Venture Beat.
4. Yetisen, Ali Kemal; Martinez-Hurtado, J. L; Da Cruz Vasconcellos, Fernando; Simsekler, M. C. Emre; Akram, Muhammad Safwan; Lowe, Christopher R (2014). "The regulation of mobile medical applications". *Lab on a Chip*. 14 (5): 833–40. doi:10.1039/C3LC51235E. PMID 24425070.
5. Pham, Xuan Lam; Nguyen, Thi Huyen; Chen, Gwo Dong (2018). "Research Through the App Store: Understanding Participant Behavior on a Mobile English Learning App". *Journal of Educational Computing Research*. 56 (7): 1076–1098. doi:10.1177/0735633117727599. S2CID 64678404.
6. Ludwig, Sean. December 5, 2012. venturebeat.com, study: "Mobile app usage grows 35%, TV & web not so much"
7. Perez, Sarah. July 2, 2012. "comScore: In U.S. Mobile Market, Samsung, Android Top The Charts; Apps Overtake Web Browsing." techcrunch.com
8. Böhmer, Matthias; Hecht, Brent; Schöning, Johannes; Krüger, Antonio; Bauer, Gernot (2011). "Falling asleep with Angry Birds, Facebook and Kindle". *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11*. pp. 47–56. doi:10.1145/2037373.2037383. ISBN 978-1-4503-0541-9. S2CID 8654592.
9. Marcano-Belisario, José S; Gupta, Ajay K; O'Donoghue, John; Morrison, Cecily; Car, Josip (2016). "Tablet computers for implementing NICE antenatal mental

10. health guidelines: Protocol of a feasibility study". *BMJ Open*. 6 (1): e009930. doi:10.1136/bmjopen-2015-009930. PMC 4735209. PMID 26801468.
11. Ventola, CL (2014). "Mobile devices and apps for health care professionals: uses and benefits". *P T*. 39 (5): 356–64. PMC 4029126. PMID 24883008.
12. "Mobile apps revenues tipped to reach \$26bn in 2013". *The Guardian*. 11 October 2013. Retrieved 19 September 2013.
13. VisionMobile, Plum Consulting, "European App Economy" analyst report, September 2013
14. Obar, Jonathan A.; Wildman, Steve (October 2015). "Social media definition and the governance challenge: An introduction to the special issue". *Telecommunications Policy*. 39 (9): 745–750. doi:10.1016/j.telpol.2015.07.014. SSRN 2647377.
15. Boyd, Danah M.; Ellison, Nicole B. (October 2007). "Social Network Sites: Definition, History, and Scholarship". *Journal of Computer-Mediated Communication*. 13 (1): 210–230. doi:10.1111/j.1083-6101.2007.00393.x. S2CID 52810295.
16. Mlaïki, Alya; Walsh, Isabelle; Kalika, Michel (February 17, 2017). "Why Do We Continue Using Social Networking Sites? The Giving Loop that feeds Computer-Mediated Social Ties". *Systèmes d'Information et Management*. 22 (2). doi:10.9876/sim.v22i2.715 (inactive January 14, 2021). Archived from the original on November 17, 2019. Retrieved November 17, 2019.
17. Joseph, R. (1993), "Touch Me—Feel Me—Feed Me— Kiss Me!", *The Naked Neuron*, Springer US, pp. 71–98, doi:10.1007/978-1-4899-6008-5_4, ISBN 978-0-306-44510-1
18. "Company Info". Facebook Newsroom. Archived from the original on November 15, 2015. Retrieved April 12, 2018.
19. Thelwall, Mike (2009). "Chapter 2 Social Network Sites". *Social Networking and the Web. Advances in Computers*. 76. pp. 19–73. doi:10.1016/S0065-2458(09)01002-X. ISBN 97801237481

20. "India records highest social networking growth Rate: Study". news.biharprabha.com. IANS. July 26, 2014. Archived from the original on August 3, 2014. Retrieved July 26, 2014.
21. Lunden, Ingrid (December 30, 2013). "73% Of U.S. Adults Use Social Networks, Pinterest Passes Twitter In Popularity, Facebook Stays On Top". TechCrunch. Archived from the original on October 27, 2015. Retrieved October 17, 2015.
22. Kaplan, Andreas M.; Haenlein, Michael (January 2010). "Users of the world, unite! The challenges and opportunities of Social Media". *Business Horizons*. 53 (1): 59–68. doi:10.1016/j.bushor.2009.09.003.
23. Amichai-Hamburger, Yair; Hayat, Tsahi (2017). "Social Networking". *The International Encyclopedia of Media Effects*. pp. 1–12. doi:10.1002/9781118783764.wbieme0170. ISBN 9781118784044.
24. Schejter, Amit M.; Tirosh, Noam (October 2015). "'Seek the meek, seek the just': Social media and social justice". *Telecommunications Policy*. 39 (9): 796–803. doi:10.1016/j.telpol.2015.08.002.
25. Buettner, Ricardo (2017). "Getting a job via career-oriented social networking markets: The weakness of too many ties". *Electronic Markets*. 27 (4): 371–385. doi:10.1007/s12525-017-0248-3. S2CID 21257828.
26. Dunbar, R. (November 14, 2003). "PSYCHOLOGY: Evolution of the Social Brain". *Science*. 302 (5648): 1160–1161. doi:10.1126/science.1092116. PMID 14615522. S2CID 144329128.
27. Granovetter, Mark (1983). "The Strength of Weak Ties: A Network Theory Revisited". *Sociological Theory*. 1: 201–233. doi:10.2307/202051. JSTOR 202051. S2CID 11450758.
28. Christakis, Nicholas A; Fowler, James H (2009). *Connected: the surprising power of our social networks and how they shape our lives*. Little, Brown and Co. ISBN 9780316036146. OCLC 318879253. Archived from the original on March 5, 2020. Retrieved December 17, 2019.[page needed]

29. Tong, Stephanie Tom; Van Der Heide, Brandon; Langwell, Lindsey; Walther, Joseph B. (April 2008). "Too Much of a Good Thing? The Relationship Between Number of Friends and Interpersonal Impressions on Facebook". *Journal of Computer-Mediated Communication*. 13 (3): 531–549. doi:10.1111/j.1083-6101.2008.00409.x.
30. Utz, Sonja (July 10, 2016). "Is LinkedIn making you more successful? The informational benefits derived from public social media". *New Media & Society*. 18 (11): 2685–2702. doi:10.1177/1461444815604143. S2CID 34629543.
31. Buettner, Ricardo (June 27, 2016). "Innovative Personality-based Digital Services". PACIS 2016 Proceedings. Archived from the original on November 17, 2019. Retrieved November 17, 2019.
32. Gonçalves, Bruno; Perra, Nicola; Vespignani, Alessandro; Perc, Matjaz (August 3, 2011). "Modeling Users' Activity on Twitter Networks: Validation of Dunbar's Number". *PLOS ONE*. 6 (8): e22656. Bibcode:2011PLoSO...622656G. doi:10.1371/journal.pone.0022656. PMC 3149601. PMID 21826200.
33. "Firebase - Crunchbase". CrunchBase. Retrieved August 1, 2020.
34. Tamplin, James. "Firebase is Joining Google!". Firebase, Inc. Retrieved October 22, 2014.
35. Melendez, Steven (May 27, 2014). "Sometimes You're Just One Hop From Something Huge". *Fast Company*. Retrieved June 11, 2014.
36. "Firebase - CrunchBase". CrunchBase. Retrieved June 11, 2014.
37. Metz, Cade (April 12, 2012). "Firebase Does for Apps What Dropbox Did for Docs". *Wired*. Retrieved February 1, 2019.
38. Ha, Anthony (May 22, 2012). "Firebase Raises \$1.1M For Real-Time App Infrastructure". *TechCrunch*. Retrieved Feb 1, 2019.
39. Darrow, Barb (June 6, 2013). "Firebase gets \$5.6M to launch its paid product and fire up its base". *Gigaom*. Retrieved June 11, 2014.
40. Lardonis, Frederic (May 13, 2014). "Firebase Adds Web Hosting To Its Database Platform". Retrieved 2019-02-01

41. "Firebase Auth". Firebase, Inc. Retrieved May 19, 2016.
42. Tamplin, James. "Firebase is Joining Google!". Firebase, Inc. Retrieved October 22, 2014.
43. Olanoff, Drew. "Google Acquires Divshot To Join Its Firebase Team, Will Shut Down In December". TechCrunch. Retrieved Feb 1, 2019.
44. Tamplin, James (May 18, 2016). "Firebase expands to become a unified app platform". Firebase, Inc. Retrieved Feb 1, 2019.
45. Paret, Rich (January 18, 2017). "Fabric is Joining Google". Retrieved 2017-01-18.
46. Ma, Francis (January 18, 2017). "Welcoming Fabric to Google". Retrieved 2017-01-18.
47. "Google launches Cloud Firestore, a new document database for app developers". TechCrunch. Retrieved 2018-07-16.
48. "Google Announces Firestore, a Document Database". InfoQ. Retrieved 2017-10-19.
49. "Firebase is launching Cloud Firestore, a new document database featuring realtime sync, no-hassle scaling, and offline support". Android Police. 2017-10-03. Retrieved 2017-10-19.
50. "Google's Cloud Firestore Lets You Focus On App Development | Androidheadlines.com". AndroidHeadlines.com|. 2017-10-05. Retrieved 2017-10-19.
51. "Firebase Products". Google. Retrieved 2020-08-01.
52. "Google faces lawsuit over tracking in apps even when users opted out". Reuters. 2020-07-14. Retrieved 2020-07-1

ДОДАТКИ

Додаток А

```
package com.sociable.Login
import android.content.Context
import androidx.appcompat.app.AppCompatActivity
import com.google.firebase.auth.FirebaseAuth
import com.google.firebase.auth.FirebaseAuth.AuthStateListener
import android.os.Bundle
import com.sociabledgdjnnh.R
import com.sociable.Login.LoginActivity
import com.google.android.gms.tasks.OnCompleteListener
import com.google.firebase.auth.AuthResult
import com.google.firebase.auth.FirebaseUser
import android.content.Intent
import android.media.MediaPlayer
import android.net.Uri
import android.util.Log
import android.view.View
import android.widget.*
import com.sociable.Home.HomeActivity
import com.sociable.Login.RegisterActivity

class LoginActivity : AppCompatActivity() {

    private var videoBG: VideoView? = null
    var mMediaPlayer: MediaPlayer? = null
    var mCurrentVideoPosition = 0
```

//firebase

private var mAuth: FirebaseAuth? = null

private var mAuthListener: AuthStateListener? = null

private var mContext: Context? = null

private var mProgressBar: ProgressBar? = null

private var mEmail: EditText? = null

private var mPassword: EditText? = null

private var mPleaseWait: TextView? = null

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity_login)

mProgressBar = findViewById<View>(R.id.progressBar) as ProgressBar

mPleaseWait = findViewById<View>(R.id.pleaseWait) as TextView

mEmail = findViewById<View>(R.id.input_email) as EditText

mPassword = findViewById<View>(R.id.input_password) as EditText

mContext = this@LoginActivity

Log.d(TAG, "onCreate: started.")

mPleaseWait!!.visibility = View.GONE

mProgressBar!!.visibility = View.GONE

setupFirebaseAuth()

init()

// Hook up the VideoView to our UI.

videoBG = findViewById<View>(R.id.videoView) as VideoView

// Build your video Uri

```

val uri = Uri.parse("android.resource://" // First start with this,
    + packageName // then retrieve your package name,
    + "/" // add a slash,
    + R.raw.video4) // and then finally add your video resource. Make sure it
is stored
    // in the raw folder.

    // Set the new Uri to our VideoView
videoBG!!.setVideoURI(uri)
    // Start the VideoView
videoBG!!.start()

    // Set an OnPreparedListener for our VideoView. For more information about
VideoViews,
    //      check      out      the      Android      Docs:
https://developer.android.com/reference/android/widget/VideoView.html
videoBG!!.setOnPreparedListener { mediaPlayer ->
    mMediaPlayer = mediaPlayer
    // We want our video to play over and over so we set looping to true.
    mMediaPlayer!!.isLooping = true
    // We then seek to the current position if it has been set and play the video.
    if (mCurrentVideoPosition != 0) {
        mMediaPlayer!!.seekTo(mCurrentVideoPosition)
        mMediaPlayer!!.start()
    }
}
}
}
VideoView.
*/
override fun onPause() {

```

```

super.onPause()
    // Capture the current video position and pause the video.
    mCurrentVideoPosition = mMediaPlayer!!.currentPosition
    videoBG!!.pause()
}

override fun onResume() {
    super.onResume()
    // Restart the video when resuming the Activity
    videoBG!!.start()
}

override fun onDestroy() {
    super.onDestroy()
    // When the Activity is destroyed, release our MediaPlayer and set it to null.
    mMediaPlayer!!.release()
    mMediaPlayer = null
}

private fun isStringNull(string: String): Boolean {
    Log.d(TAG, "isStringNull: checking string if null.")
    return if (string == "") {
        true
    } else {
        false
    }
}

```

*/**

----- Firebase -----

**/*

private fun init() {

//initialize the button for logging in

val btnLogin = findViewById<View>(R.id.btn_login) as Button

btnLogin.setOnClickListener {

Log.d(TAG, "onClick: attempting to log in.")

val email = mEmail!!.text.toString()

val password = mPassword!!.text.toString()

if (isStringNull(email) && isStringNull(password)) {

*Toast.makeText(mContext, "You must fill out all the fields",
Toast.LENGTH_SHORT).show()*

} else {

mProgressBar!!.visibility = View.VISIBLE

mPleaseWait!!.visibility = View.VISIBLE

mAuth!!.signInWithEmailAndPassword(email, password)

.addOnCompleteListener(this@loginActivity) { task ->

Log.d(TAG, "signInWithEmail:onComplete:" + task.isSuccessful)

val user = mAuth!!.currentUser

// If sign in fails, display a message to the user. If sign in succeeds

// the auth state listener will be notified and logic to handle the

// signed in user can be handled in the listener.

if (!task.isSuccessful) {

Log.w(TAG, "signInWithEmail:failed", task.exception)

*Toast.makeText(this@loginActivity,
getString(R.string.auth_failed),*

Toast.LENGTH_SHORT).show()

```

        mProgressBar!!.visibility = View.GONE
        mPleaseWait!!.visibility = View.GONE
    } else {
        val intent = Intent(this@LoginActivity,
HomeActivity::class.java)
        startActivity(intent)
        // try{
// if(user.isEmailVerified()){
// Log.d(TAG, "onComplete: success. email is verified.");
// Intent intent = new Intent(LoginActivity.this,
HomeActivity.class);
// startActivity(intent);
// }else{
// Toast.makeText(mContext, "Email is not verified \n
check your email inbox.", Toast.LENGTH_SHORT).show();
// mProgressBar.setVisibility(View.GONE);
// mPleaseWait.setVisibility(View.GONE);
// mAuth.signOut();
// }
// }catch (NullPointerException e){
// Log.e(TAG, "onComplete: NullPointerException: " +
e.getMessage() );
// }
    }

    // ...
}
}
}
}

```



```

val linkSignUp = findViewById<View>(R.id.link_signup) as TextView
linkSignUp.setOnClickListener {
    Log.d(TAG, "onClick: navigating to register screen")
    val intent = Intent(this@LoginActivity, RegisterActivity::class.java)
    startActivity(intent)
}

/*
If the user is logged in then navigate to HomeActivity and call 'finish()'
*/if ( mAuth!!.currentUser != null) {
    val intent = Intent(this@LoginActivity, HomeActivity::class.java)
    startActivity(intent)
    finish()
}

}

/**
 * Setup the firebase auth object
 */
private fun setupFirebaseAuth() {
    Log.d(TAG, "setupFirebaseAuth: setting up firebase auth.")
    mAuth = FirebaseAuth.getInstance()
    mAuthListener = AuthStateListener { firebaseAuth ->
        val user = firebaseAuth.currentUser
        if (user != null) {
            // User is signed in
            Log.d(TAG, "onAuthStateChanged:signed_in:" + user.uid)
        } else {

```

```
        // User is signed out
        Log.d(TAG, "onAuthStateChanged:signed_out")
    }
    // ...
}
}
```

```
public override fun onStart() {
    super.onStart()
    mAuth!!.addAuthStateListener(mAuthListener)
}
```

```
public override fun onStop() {
    super.onStop()
    if (mAuthListener != null) {
        mAuth!!.removeAuthStateListener(mAuthListener)
    }
}
```

```
companion object {
    private const val TAG = "LoginActivity"
}
```

package com.sociable.Login

import android.content.Context

import android.media.MediaPlayer

import android.net.Uri

import android.os.Bundle

import android.util.Log

import android.view.View

*import android.widget.**

import androidx.appcompat.app.AppCompatActivity

import com.google.firebase.auth.FirebaseAuth

import com.google.firebase.auth.FirebaseAuth.AuthStateListener

*import com.google.firebase.database.**

import com.sociable.Login.RegisterActivity

import com.sociable.models.User

import com.sociable.R

import com.sociable.Utils.FirebaseMethods

class RegisterActivity : AppCompatActivity() {

private var mContext: Context? = null

private var email: String? = null

private var username: String? = null

private var password: String? = null

private var mEmail: EditText? = null

private var mPassword: EditText? = null

private var mUsername: EditText? = null

private var loadingPleaseWait: TextView? = null

private var btnRegister: Button? = nul

```

private var mProgressBar: ProgressBar? = null
private var videoBG: VideoView? = null
var mMediaPlayer: MediaPlayer? = null
var mCurrentVideoPosition = 0

//firebase
private var mAuth: FirebaseAuth? = null
private var mAuthListener: AuthStateListener? = null
private var firebaseMethods: FirebaseMethods? = null
private var mFirebaseDatabase: FirebaseDatabase? = null
private var myRef: DatabaseReference? = null
private var append = ""
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_register)
    mContext = this@RegisterActivity
    firebaseMethods = FirebaseMethods(mContext)
    Log.d(TAG, "onCreate: started.")
    initWidgets()
    setupFirebaseAuth()
    init()
    // Hook up the VideoView to our UI.
    videoBG = findViewById<View>(R.id.videoView) as VideoView

    // Build your video Uri
    val uri = Uri.parse("android.resource://" // First start with this,
        + packageName // then retrieve your package name,
        + "/" // add a slash,
        + R.raw.video4) // and then finally add your video resource. Make sure it
is stored

```

// in the raw folder.

// Set the new Uri to our VideoView

videoBG!!.setVideoURI(uri)

// Start the VideoView

videoBG!!.start()

// Set an OnPreparedListener for our VideoView. For more information about VideoViews,

// check out the Android Docs:

<https://developer.android.com/reference/android/widget/VideoView.html>

videoBG!!.setOnPreparedListener { mediaPlayer ->

mMediaPlayer = mediaPlayer

// We want our video to play over and over so we set looping to true.

mMediaPlayer!!.isLooping = true

// We then seek to the current position if it has been set and play the video.

if (mCurrentVideoPosition != 0) {

mMediaPlayer!!.seekTo(mCurrentVideoPosition)

mMediaPlayer!!.start()

}

}

}

/===== Important Section!*

=====

We must override onPause(), onResume(), and onDestroy() to properly handle our

VideoView.

**/*

```
override fun onPause() {  
    super.onPause()  
    // Capture the current video position and pause the video.  
    mCurrentVideoPosition = mMediaPlayer!!.currentPosition  
    videoBG!!.pause()  
}
```

```
override fun onResume() {  
    super.onResume()  
    // Restart the video when resuming the Activity  
    videoBG!!.start()  
}
```

```
override fun onDestroy() {  
    super.onDestroy()  
    // When the Activity is destroyed, release our MediaPlayer and set it to null.  
    mMediaPlayer!!.release()  
    mMediaPlayer = null  
  
}
```

```
private fun init() {  
    btnRegister!!.setOnClickListener {  
        email = mEmail!!.text.toString()  
        username = mUsername!!.text.toString()  
        password = mPassword!!.text.toString()  
        if (checkInputs(email!!, username!!, password!!)) {  
            mProgressBar!!.visibility = View.VISIBLE  
            loadingPleaseWait!!.visibility = View.VISIBLE  
        }  
    }  
}
```

```

        firebaseMethods!!.registerNewEmail(email, password, username)
    }
}
}

```

```

private fun checkInputs(email: String, username: String, password: String):
Boolean {
    Log.d(TAG, "checkInputs: checking inputs for null values.")
    if (email == "" || username == "" || password == "") {
        Toast.makeText(mContext, "All fields must be filled out.",
Toast.LENGTH_SHORT).show()
        return false
    }
    return true
}

```

```

/**

```

```

 * Initialize the activity widgets

```

```

 */

```

```

private fun initWidgets() {
    Log.d(TAG, "initWidgets: Initializing Widgets.")
    mEmail = findViewById<View>(R.id.input_email) as EditText
    mUsername = findViewById<View>(R.id.input_username) as EditText
    btnRegister = findViewById<View>(R.id.btn_register) as Button
    mProgressBar = findViewById<View>(R.id.progressBar) as ProgressBar
    loadingPleaseWait = findViewById<View>(R.id.loadingPleaseWait) as
TextView
    mPassword = findViewById<View>(R.id.input_password) as EditText
    mContext = this@RegisterActivity
}

```

```
mProgressBar!!.visibility = View.GONE  
loadingPleaseWait!!.visibility = View.GONE  
}
```

```
private fun isStringNull(string: String): Boolean {  
    Log.d(TAG, "isStringNull: checking string if null.")  
    return if (string == "") {  
        true  
    } else {  
        false  
    }  
}  
/*
```

```
----- Firebase -----  
*/  
/**
```

```
* Check if @param username already exists in database  
* @param username  
*/
```

```
private fun checkIfUsernameExists(username: String?) {  
    Log.d(TAG, "checkIfUsernameExists: Checking if $username already exists")  
    val reference = FirebaseDatabase.getInstance().reference  
    val query = reference  
        .child(getString(R.string.dbname_users))  
        .orderByChild(getString(R.string.field_username))  
        .equalTo(username)  
    query.addListenerForSingleValueEvent(object : ValueEventListener {  
        override fun onDataChange(dataSnapshot: DataSnapshot) {  
            for (singleSnapshot in dataSnapshot.children) {
```



```

        if (singleSnapshot.exists()) {
            Log.d(TAG, "checkIfUsernameExists: FOUND A MATCH" +
singleSnapshot.getValue(User::class.java)!!.username)
            append = myRef!!.push().key!!.substring(3, 10)
            Log.d(TAG, "onDataChange: username already exists. Appending
random string to name: $append")
        }
    }

    //1st check: Make sure the username is not already in use
    val mUsername: String
    mUsername = username + append

    //add new user to the database
    firebaseMethods!!.addNewUser(email, mUsername, "", "", "")
    Toast.makeText(mContext, "Signup successful. Sending verification
email.", Toast.LENGTH_SHORT).show()
    mAuth!!.signOut()
}

    override fun onCancelled(databaseError: DatabaseError) {}
})
}

/**
 * Setup the firebase auth object
 */
private fun setupFirebaseAuth() {
    Log.d(TAG, "setupFirebaseAuth: setting up firebase auth.")
}

```

```

mAuth = FirebaseAuth.getInstance()
mFirebaseDatabase = FirebaseDatabase.getInstance()
myRef = mFirebaseDatabase!!.reference
mAuthListener = AuthStateListener { firebaseAuth ->
    val user = firebaseAuth.currentUser
    if (user != null) {
        // User is signed in
        Log.d(TAG, "onAuthStateChanged:signed_in:" + user.uid)
        myRef!!.addListenerForSingleValueEvent(object : ValueEventListener {
            override fun onDataChange(dataSnapshot: DataSnapshot) {
                checkIfUsernameExists(username)
            }
        )
        override fun onCancelled(databaseError: DatabaseError) {}
    })
    finish()
    } else {
        // User is signed out
        Log.d(TAG, "onAuthStateChanged:signed_out")
    }
    // ...
    }
    }
}

public override fun onStart() {
    super.onStart()
    mAuth!!.addAuthStateListener(mAuthListener)
    }

```

```
public override fun onStop() {  
    super.onStop()  
    if ( mAuthListener != null) {  
        mAuth!!.removeAuthStateListener(mAuthListener)  
    }  
}  
  
companion object {  
    private const val TAG = "RegisterActivity"  
}  
}
```

Додаток В



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка додатку «Sociable» з використанням Kotlin для Android

Виконала студентка 4 курсу
групи ПД-44
Стефанюк Лілія Богданівна
Керівник роботи
Гребенюк Віктор Вікторович

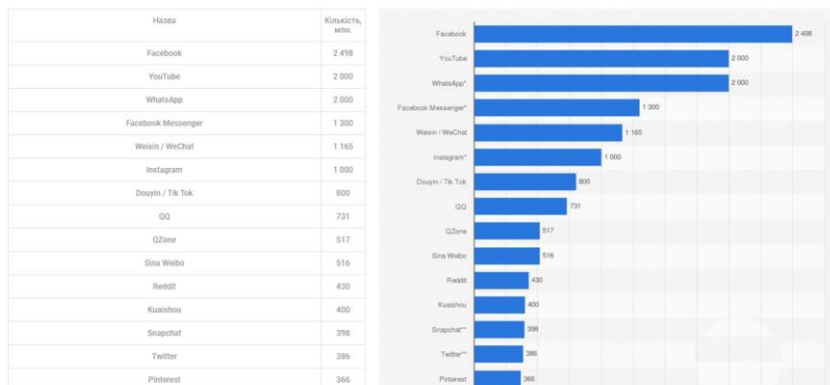
Київ – 2021

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи-** розробка додатку «Sociable» з використанням Kotlin для Android.
- **Об'єкт дослідження-** поліпшення функціоналу та якості соціальних мереж.
- **Предмет дослідження-** соціальна мережа, для зручного використання із додаванням нового функціоналу.

2

ПОПУЛЯРНІСТЬ СОЦІАЛЬНИХ МЕРЕЖ



3

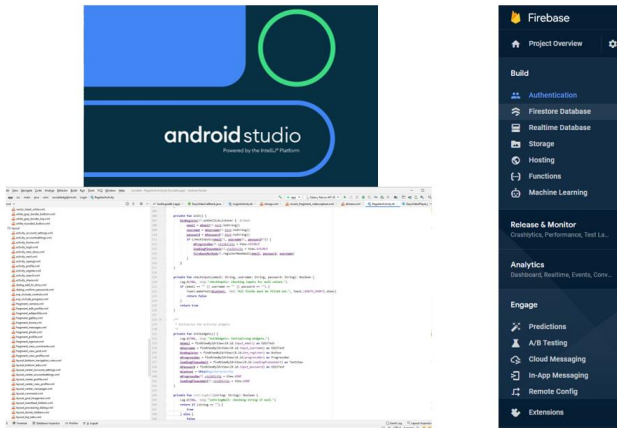
ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



- Спеціальний рефакторинг для Android та швидкі виправлення
- Інструменти Lint для виявлення продуктивності, зручності використання, сумісності версій та інших проблем
- Можливості інтеграції та підписання програм ProGuard
- Майстри на основі шаблонів для створення загальних конструкцій та компонентів Android
- Розширений редактор макетів, що дозволяє користувачам перетягувати компоненти інтерфейсу, можливість попереднього перегляду макетів на декількох конфігураціях екрана
- Віртуальний пристрій Android (емулятор) для запуску та налагодження програм у студії Android

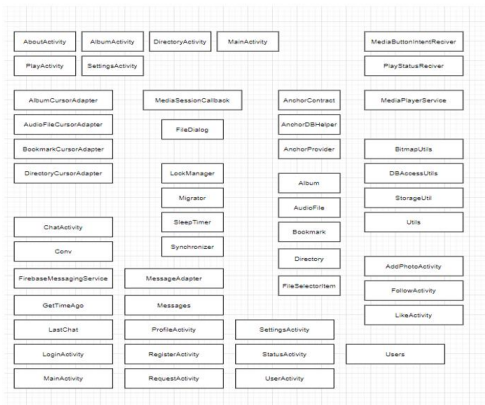
4

СЕРЕДОВИЩЕ РОЗРОБКИ



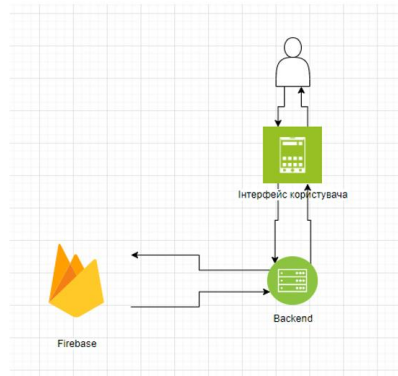
5

ДІАГРАМА КЛАСІВ



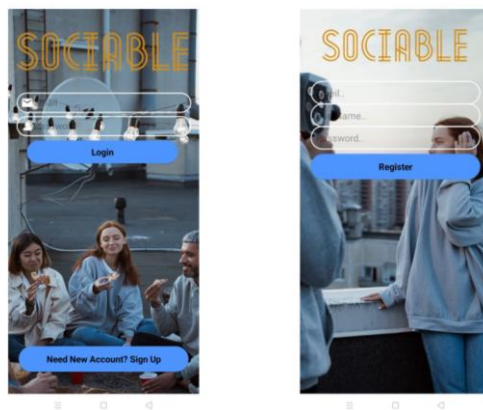
6

СТРУКТУРНА ДІАГРАМА ПРОЕКТУ



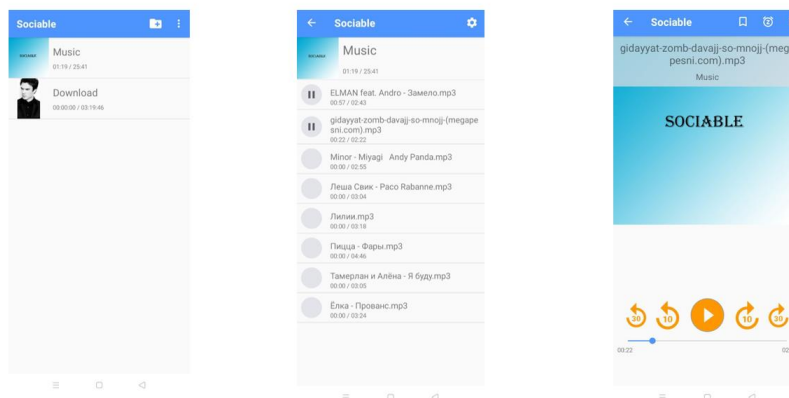
7

ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА



8

ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА



9

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Стефанюк Л.Б Розробка соціальної мережі в середовищі розробки Android Studio для ОС Android на мові Kotlin: XII науково-технічна конференція сучасні інформаційні технології

10

ВИСНОВКИ

У процесі виконання випускної роботи, були поставлені та виконанні наступні задачі:

- Проаналізувати та порівняти популярні соціальні мережі
- Створена модель об'єкту програмування
- Розроблений функціонал соціальної мережі

Для подальшого розвитку, необхідно розширити функціональність соціальної мережі, реалізувавши наступні дії:

- Сповіщення про знайомих користувачів
- Пошук музики в інтернеті
- Можливість зміни стандартного шаблону вигляду соціальної мережі

11

ДЯКУЮ ЗА УВАГУ!
