



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Чорноморський національний університет імені Петра Могили**

**Факультет комп'ютерних наук**

**Кафедра інженерії програмного забезпечення**

**ЗАТВЕРДЖУЮ**

Зав. кафедри \_\_\_\_\_ Є. О. Давиденко

« \_\_\_\_\_ » \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на виконання кваліфікаційної роботи бакалавра**

Видано студенту групи 409 факультету комп'ютерних наук

Мельникову Андрію Владиславовичу

(прізвище, ім'я, по батькові студента)

1. Тема кваліфікаційної роботи

Застосунок для онлайн-знайомств

Затверджена наказом по ЧНУ від «17» березня 2023 р. № 60

2. Строк представлення кваліфікаційної роботи « \_\_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

3. Очікуваний результат роботи та початкові дані, якщо такі потрібні

Очікуваним результатом є розроблений застосунок для онлайн-знайомств

4. Перелік питань, що підлягають розробці:

- дослідження предметної області та аналіз існуючих аналогів;
- формування специфікації вимог до ПЗ;
- визначення архітектури для проектування ПЗ;
- моделювання та проектування ПЗ;
- розробка ПЗ;
- здійснення тестування роботи ПЗ;

– проведення аналізу результатів розробки ПЗ;

5. Перелік графічних матеріалів

Презентація до КРБ

6. Завдання до спеціальної частини

Аналіз охорони праці на робочих місцях фахівців з інформаційних технологій

7. Консультанти:

Консультант	Кафедра (організація)	Частина роботи
канд. техн. наук, доц. Алексєєва А.О	Кафедра екології Медичного інституту ЧНУ ім. Петра Могили	Спеціальна частина з охорони праці

Керівник роботи PhD, ст. викл. Кафедри ПЗ Антіпова Катерина Олександрівна  
(посада, прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Завдання прийнято до виконання

Мельников Андрій Владиславович

(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(підпис)

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 2023р.

# КАЛЕНДАРНИЙ ПЛАН

## виконання кваліфікаційної роботи

Тема: Застосунок для онлайн-знайомств

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КРБ	13.03.2023	16.03.2023	виконано
2.	Пошук літератури за темою кваліфікаційної роботи	20.04.2023	21.04.2023	виконано
3.	Складання календарного плану роботи	23.04.2023	24.04.2023	виконано
4.	Аналіз предметної області	25.04.2023	25.04.2023	виконано
5.	Розробка проєктних рішень	26.04.2023	30.04.2023	виконано
6.	Моделювання та конструювання ПЗ	01.05.2023	06.05.2023	виконано
7.	Кодування розробленого ПЗ	10.05.2023	25.05.2023	виконано
9.	Розробка частини КРБ з охорони праці	13.05.2023	15.05.2023	виконано
10.	Відгук керівника КРБ	19.06.2023	19.05.2023	виконано
11.	Оформлення КРБ та презентації КРБ	27.04.2023	27.04.2023	виконано
12.	Попередній захист КРБ	07.06.2023	07.06.2023	виконано
13.	Рецензування КРБ	19.06.2023		
14.	Завершення оформлення КРБ та презентації	19.06.2023	19.06.2023	виконано
15.	Захист кваліфікаційної роботи	26.06.2023	26.06.2023	

Розробив студент Мельников Андрій Владиславович

(прізвище, ім'я, по батькові)

(підпис)

«\_\_» \_\_\_\_\_ 2023 р

Керівник роботи PhD,ст.викл. каф. ПЗ Антіпова Катерина Олександрівна

(посада, прізвище, ім'я, по батькові)

(підпис)

«\_\_» \_\_\_\_\_ 2023 р

## АНОТАЦІЯ

до кваліфікаційної роботи бакалавра

«Застосунок для онлайн-знайомств»

Студент 409 групи: Мельников А. В.

Керівник: ст. викладач, PhD Антіпова К. О.

Ця робота присвячена розробці застосунку для онлайн-знайомств.

Об'єкт роботи є процес розробки застосунку для онлайн-знайомств.

Предмет роботи є технології, необхідні для розробки застосунку для знайомств.

Метою роботи є популяризація застосунків, які дозволяють користувачам знайомитися між собою, шляхом розробки продукту, який буде привабливим для користувачів та забезпечуватиме їм зручний та безпечний спосіб знайомств.

У вступі визначається актуальність теми, мета, предмет та об'єкт дослідження.

У першому розділі проведено аналіз існуючих застосунків-аналогів, визначення функціоналу, переваг, недоліків, технології за допомогою яких було створено застосунок. Також проведено формування та опис специфікації вимог застосунку, що розробляється.

У другому розділі проведено моделювання та розробку структури застосунку, що розробляється.

У третьому розділі проводиться огляд мов програмування, їх бібліотек та інших технологій, що використовуються для розробки застосунку.

У четвертому розділі демонструється результат виконаної роботи, тестування застосунку.

У висновках проводиться аналіз виконаних робіт та отриманих результатів.

Кваліфікаційна робота бакалавра викладена на 50 сторінок, містить 4 розділи, 34 ілюстрацій, 4 таблиць, 20 джерел в переліку посилань.

Ключові слова: *застосунок, розробка застосунку для онлайн-знайомств, онлайн-знайомства.*

## **ABSTRACT**

of the Bachelor's Thesis

« Application for Online Dating »

Student: Andrii Melnykov

Supervisor: Senior Lecturer, PhD Kateryna Antipova

This thesis is devoted to the development of an application for online dating.

The object of the study is the process of developing an application for online dating.

The subject of the study is the technologies necessary for the development of such an application.

The aim of the work is to create an application that allows users to meet each other, as well as to develop and implement a product that will be attractive to users and provide them with a convenient and safe way of getting to know each other.

The introduction defines the relevance of the topic, the aim, the subject, and the object of the research.

The first chapter provides an analysis of existing similar applications, defines the functionality, advantages, disadvantages, and the technologies used to develop the application. The chapter also includes the formation and description of the application's requirements specification.

The second chapter involves modeling and developing the structure of the application being created.

The third chapter reviews programming languages, libraries, and other technologies used for application development.

The fourth chapter demonstrates the results of the completed work, including testing of the application.

The conclusions provide an analysis of the completed work and the obtained results.

The Bachelor's thesis consists of 50 pages, including 4 chapters, 34 illustrations, 4 tables, and 20 references in the bibliography.

Keywords: application, development of an application for online dating, online dating.

## ЗМІСТ

<b>ПЕРЕЛІК СКОРОЧЕНЬ</b> .....	3
<b>ВСТУП</b> .....	4
<b>1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ</b> .....	5
1.1 Огляд застосунків-аналогів для онлайн-знайомств .....	5
1.2 Аналіз застосунку, що розробляється.....	10
1.3 Специфікація вимог до ПЗ.....	12
Висновки до розділу 1 .....	15
<b>2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	17
2.1 Діаграма сценаріїв використання (use case).....	17
2.2 Діаграми діяльності .....	18
2.3 Діаграми пакетів .....	20
2.4 Діаграми розгортання.....	22
Висновки до розділу 2.....	23
<b>3 ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ</b> .....	24
3.1 Оглядання технологій .....	24
Висновки до розділу 3.....	36
<b>4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ</b> .....	37
4.1 Завантаження та налаштування технологій .....	37
4.2 Реалізація застосунку для онлайн-знайомств .....	39
Висновки до розділу 4.....	47
<b>ВИСНОВКИ</b> .....	48
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ</b> .....	49

## ПЕРЕЛІК СКОРОЧЕНЬ

ПЗ	–	програмне забезпечення
ЦА	–	цільова аудиторія
ОС	–	операційна система
ТЗ	–	технічне завдання
API	–	application programming interface
UML	–	unified modeling language
UI	–	user interface
UX	–	user experience
PC	–	personal computer



## ВСТУП

Кваліфікаційна робота бакалавра розглядає розробку застосунку для онлайн-знайомств, а саме проектування та моделювання основних складових проекту.

**Актуальність теми** кваліфікаційної роботи бакалавра зумовлена ситуацією в країні. Спочатку наша країна зіткнулась з пандемією Covid-19, а потім повномасштабним вторгненням росіянської федерації. Як наслідок цих подій дуже багато співвітчизників не мають змогу знайомитись на вулиці, в кафе і тд.

З цієї причини багато людей змінило своє місце перебування, де у них немає знайомих або друзів. Даний проєкт має ціль допомогти нашим громадянам знаходити нові знайомства або навіть кохання у різних куточках нашої країни. Також технологічний прогрес не стоїть на місці і тому все більше і більше людей сидять в телефонах не відволікаючись ні на що інше.

**Об'єкт дослідження:** процес розробки застосунку для онлайн-знайомств.

**Предмет дослідження:** технології, необхідні для розробки застосунку для знайомств.

**Метою роботи** є популяризація застосунків, які дозволяють користувачам знайомитися між собою, шляхом розробки продукту, який буде привабливим для користувачів та забезпечуватиме їм зручний та безпечний спосіб знайомств.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- аналіз предметної сфери;
- дизайн та розробка логотипу проєкту;
- дизайн зовнішнього вигляду веб-застосунку;
- створення веб-застосунку;
- тестування та виправлення помилок;
- відправлення проєкту на production server та його запуск в мережу Інтернет.

**Практичне застосування:** даний застосунок можна використовувати для знайомств у мережі Інтернет. Будь-яка людина старше 18-ти років зможе використовувати застосунок, щоб знайти собі друзів, кохання, тощо.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ВИМОГ

## 1.1 Огляд застосунків-аналогів для онлайн-знайомств

Аналіз предметної є важливою складовою для успішного проєктування будь-якого застосунку, тому що це визначити необхідний функціонал, основні переваги та недоліки конкурентів. Для проведення аналізу аналогів обрано наступні вебзастосунки: Tinder (табл. 1.1), Badoo (табл. 1.2), Bumble (табл. 1.3).

### Tinder

Tinder - це мобільний додаток для онлайн-знайомств, де користувачі можуть переглядати фотографії і профілі інших користувачів, вибирати тих, що їм подобаються, та взаємодіяти з ними за допомогою текстових повідомлень, якщо дві людини взаємно підпадають під вимоги одне одного. Tinder є однією з найпопулярніших платформ для онлайн-знайомств, особливо серед молодіжної аудиторії.

Таблиця 1.1 – Опис Tinder

<b>Назва</b>	Tinder
<b>Розробник</b>	Match Group, InterActiveCorp, Tinder Inc.
<b>Архітектура</b>	Mobile application, web application
<b>Мова реалізації</b>	Java, C#, Objective-C
<b>Функції</b>	1. Перегляд профілів; 2. Зацікавленість; 3. Налаштування пошуку; 4. Платні функції; 5. Відправка повідомлень.
<b>Переваги</b>	1. Простота використання; 2. Широкий вибір користувачів; 3. Зручний пошук.

<b>Недоліки</b>	1. Обмеження у кількості співбесідників; 2. Проблеми з приватністю даних.
<b>Вебсайт</b>	<a href="https://tinder.com/">https://tinder.com/</a>

Кінець таблиці 1.1 – Опис Tinder



Рисунок 1.1 – Вигляд інтерфейсу «Tinder»

## Badoo

Badoo - це соціальна мережа для знайомств, яка працює в онлайн-режимі. Badoo дає можливість користувачам створювати свій профіль, завантажувати фотографії, додавати інформацію про себе та переглядати профілі інших користувачів. Крім того, Badoo пропонує різноманітні інструменти для пошуку та знайомства з новими

2023 р. Мельников А. В. 121 – ПРП.01 – 409.21910912

людьми, включаючи функцію "Люди поруч", яка дозволяє знайти користувачів, які перебувають у вашому регіоні, та ігри для знайомств, які допомагають знайти спільну мову з іншими користувачами. Badoo є однією з найбільш популярних платформ для знайомств, яка має більше 470 мільйонів користувачів у світі.

Таблиця 1.2 – Опис Badoo

<b>Назва</b>	Badoo
<b>Розробник</b>	Bumble Inc.
<b>Архітектура</b>	Web application, social network
<b>Мова реалізації</b>	Java, Scala, Python, Objective-C, Swift, JavaScript, HTML, CSS
<b>Функції</b>	<ol style="list-style-type: none"> <li>1. Вікторини у чатах, щоб краще пізнати один одного.</li> <li>2. Фільтрація користувачів.</li> <li>3. Платні послуги.</li> <li>4. Створення профілю.</li> <li>5. Система відгуків.</li> </ol>
<b>Переваги</b>	<ol style="list-style-type: none"> <li>1. Функції приватності.</li> <li>2. Велика база користувачів.</li> <li>3. Можливість змінювати місце розташування</li> </ol>
<b>Недоліки</b>	<ol style="list-style-type: none"> <li>1. Обмежені функції пошуку.</li> <li>2. Багато спаму.</li> <li>3. Велика кількість фейків</li> </ol>
<b>Вебсайт</b>	<a href="https://badoo.com/">https://badoo.com/</a>

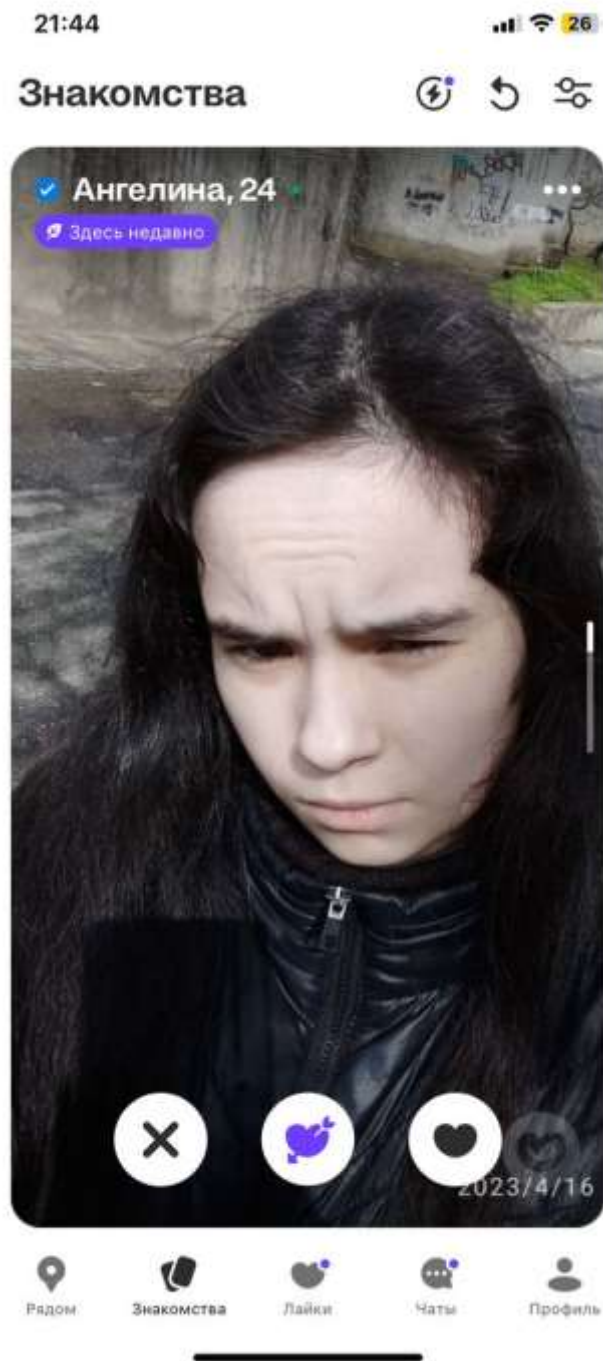


Рисунок 1.2 – Вигляд інтерфейсу «Badoo»

## Bumble

Bumble - це додаток для знайомств, який дозволяє користувачам шукати нових знайомих в онлайні. Bumble пропонує різноманітні функції, включаючи можливість створення профілю з фотографіями та інформацією про себе, пошук потенційних

партнерів на основі віку, місцезнаходження та інших параметрів, а також спілкування з іншими користувачами через повідомлення або відео-чат.

Таблиця 1.3 – Опис Bumble

<b>Назва</b>	Bumble
<b>Розробник</b>	Bumble Inc.
<b>Архітектура</b>	Web application, mobile application
<b>Мова реалізації</b>	Java, Kotlin, Swift, Python, JavaScript
<b>Функції</b>	1. Bumble BFF. 2. Метчі. 3. Фільтри пошуку. 4. Функція «Жінки першими». 5. Пошук.
<b>Переваги</b>	1. Безпека. 2. Верифікація. 3. Різні способи зв'язку. 4. Bumble Bizz.
<b>Недоліки</b>	1. Менша база користувачів. 2. Відсутність детальної інформації. 3. Обмеження у безкоштовній версії.
<b>Вебсайт</b>	<a href="https://bumble.com/">https://bumble.com/</a>



Рисунок 1.3 – Вигляд інтерфейсу «Bumble»

## 1.2 Аналіз застосунку, що розробляється

Основні функції застосунків для знайомств включають можливість перегляду профілів інших користувачів, відправлення повідомлень, встановлення фільтрів для пошуку потенційних партнерів тощо. Важливим аспектом є також алгоритм підбору партнерів, який може ґрунтуватися на взаємних інтересах, геолокації, віку та інших факторах.

Інтерфейс застосунків для знайомств зазвичай має бути простим та зрозумілим для користувачів. Меню повинно бути легко доступним, а профілі користувачів мають бути зручними для перегляду та редагування.

Таблиця 1.4 – Опис системи, що проєктується

<b>Основні задачі</b>	<ol style="list-style-type: none"><li>1. Авторизація через соціальну мережу Facebook;</li><li>2. Створення фільтрації пошуку людей;</li><li>3. Створення приємного інтерфейсу для застосунку;</li><li>4. Створення карток для свайпу, коли подобається людина чи ні;</li><li>5. Створення функції для відображення інформації про людини;</li></ol>
<b>Користувачі системи</b>	<ol style="list-style-type: none"><li>1. Користувач застосунку</li><li>2. Адміністратор</li></ol>
<b>Сценарії роботи</b>	<ol style="list-style-type: none"><li>1. Як користувач застосунку, я хочу мати змогу бачити інформацію про іншого співбесідника.</li><li>2. Як користувач застосунку, я хочу конфідесійність своїх даних.</li><li>3. Як адміністратор застосунку, я хочу мати змогу блокувати користувачів, які порушують правила політики застосунку.</li><li>4. Як користувач застосунку, я хочу мати змогу обирати хто мені подобається, а хто ні.</li></ol>
<b>Засоби апаратної та програмної реалізації</b>	<ol style="list-style-type: none"><li>1. Sublime Text;</li><li>2. Ionic framework;</li><li>3. AngularJS;</li><li>4. Firebase;</li><li>5. NodeJS;</li><li>6. HTML;</li></ol>



	7. CSS; 8. NoSQL;
<b>Вихідні дані</b>	1. представлення користувачам інформації щодо проєкту через сторінки вебсайту; 2. представлення користувачам інформації про учасників проєкту.

Кінець таблиці 1.4 – Опис системи, що проєктується

### **1.3 Специфікація вимог до ПЗ**

#### **Призначення застосунку, для якого розробляється програмне забезпечення**

Призначенням є впровадження застосунку, який був би привабливим для користувачів та забезпечував зручний та безпечний спосіб для знайомств завдяки розробці ПЗ.

#### **Погодження, що ухвалені в програмній документації**

Було узгоджено, що для створення ПЗ та його роботи буде використано такі технології: текстовий редактор Sublime Text 3, Java-Script фреймворк AngularJS, середа розробки гібридних кроссплатформених мобільних застосунків, веб-застосунків Ionic Framework, платформа для розробки мобільних та веб-додатків, яка надає широкий спектр інструментів для зберігання та синхронізації даних, аутентифікації користувачів, аналітики, тестування, розгортання та багато іншого Firebase, мова розмітки HTML, мова стилів CSS, база даних NoSQL.

#### **ЗАГАЛЬНИЙ ОПИС**

##### **Сфера застосування**

Це ПЗ не має обмежень у сфері споживання, його можна застосовувати у звичайному житті будь де.

##### **Характеристика користувачів**

Основні характеристики користувачів: наявність смартфона з доступом в мережу Інтернет, бути старшим за 18 років.

## **Загальна структура і склад системи**

Основна структура: веб-застосунок, застосунок для телефону

## **Загальні обмеження**

Обмеження для роботи: відсутність смартфона, який не підключен до мережі Інтернет.

## **ОСНОВНІ ФУНКЦІЇ ЗАСТОСУНКУ ДЛЯ ОНЛАЙН-ЗНАЙОМСТВ**

*Наявність підключення до мережі Інтернет*

### **Опис функції**

Ця функція дає змогу користувачу користуватися застосунком.

### **Вхідна і вихідна інформація**

Вхідна – особисті дані, які будуть витягуватися з анкети соціальної мережі Facebook.

Вихідна – анкета у застосунку для онлайн-знайомств.

### **Функціональні вимоги**

Підключення до мережі Інтернет; авторизація в застосунку через соціальну мережу Facebook.

*Функція взаємних лайків та метчів з іншими користувачами*

### **Опис функції**

Дана функція включає в себе основну логіку цього застосунку. Завдяки цієї функції люди будуть розуміти чи сподобались вони один одному візуально.

### **Вхідна і вихідна інформація**

Вхідна – ваші профілі у застосунку

Вихідна – відображення метчу

### **Функціональні вимоги**

Використання Firebase в якому будуть зберігатися айдішники кожного користувача, якого ви лайкнете; користувач лайкає вас взаємно; отримання метчу.

## **ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ**

### **Джерела і зміст вхідної інформації (даних)**

В даному ПЗ джерелами вхідної інформації є користувач ( авторизація через соціальну мережу Facebook ) та інтернет з'єднання ( дає доступ до користування застосунком ).

### **Нормативно-довідкова інформація**

Вимоги відсутні

### **Вимоги до способів організації, збереження та ведення інформації**

Отримання даних з інтерфейсу програмування застосунків Facebook Graph API, який дозволяє розробникам взаємодіяти з різними функціями Facebook, такими як отримання профілю користувача.

### **ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

Вимоги до технічного забезпечення мають деякі обмеження у вигляді підтримки смартфоном ОС не нижче Android 5.0 та IOS 12.0 відповідно.

### **ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

#### **Архітектура програмної системи**

Архітектура включає: клієнтська частина, серверна частина, база даних.

#### **Системне програмне забезпечення**

Застосунок має бути побудований з використанням текстового редактору Sublime Text 3 з використанням фреймворку для розробки мобільних додатків з використанням веб-технологій Ionic Framework. Процес отримання даних для профілю має відбуватись Facebook Graph API, який використовує протокол HTTP. Для зберігання користувачів, лайків, метчів має використовуватись NoSQL Database та Firebase.

#### **Мова і технологія розробки ПЗ**

ПЗ має розроблятись за допомогою мови JavaScript, фреймворку AngularJS, HTML, CSS, NodeJS.

### **ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ**

Інтерфейс застосунку повинен відповідати принципам UI/UX дизайну, щоб користувачі могли легко орієнтуватися та користуватися ним. Він має включати налаштування пошуку, такі як вік та стаття, а також можливість вийти з облікового

запису в застосунку. Для поліпшення користувацького досвіду, інтерфейс має бути адаптивним та містити приємні оку анімації. Крім того, для взаємодії з іншими користувачами, в ньому мають бути кнопки для дій, такі як лайк, дізлайк та перегляд інформації про користувача.

### **Апаратний інтерфейс**

Мобільний телефон ( смартфон ) на базі ОС Android, IOS.

### **Комунікаційний протокол**

Застосунок передбачає використання протоколу HTTP – протоколу для комунікації між клієнтом та сервером.

## **ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

### **Доступність**

ПЗ є доступним для користувачів, які самотні у цьому житті і прагнуть знайти собі друзів або навіть кохання.

### **Супроводженість**

Застосунок не потребує супроводженості.

### **Переносимість**

ПЗ може працювати на Windows. Для перенесення коду на ОС Android або IOS потрібно буде внести деякі зміни незважаючи на те, що ionic framework платформа для розробки гібридних застосунків. Якщо з ОС Android проблем не виникне, то з перенесенням на ОС IOS можуть виникати певні проблеми, оскільки IOS має свої особливості. Наприклад, IOS має ряд обмежень, пов'язаних з безпекою та приватністю користувачів, тому можуть бути необхідні додаткові налаштування.

### **Продуктивність**

Залежить від версії операційної системи.

## **Висновки до розділу 1**

У першому розділі кваліфікаційної роботи бакалавра було проведено огляд та аналіз аналогів програмного забезпечення у сфері онлайн-знайомств, що дозволило покращити навички аналізу функціоналу подібних систем. Було виділено переваги та

недоліки досліджених систем, проведено детальний аналіз розроблюваного застосунку, визначено його функціональні можливості, сценарії використання та складено технічне завдання (ТЗ) до програмного продукту.

Розділ містить специфікацію вимог до програмного забезпечення, що розробляється. В даній частині розділу представлено повний опис поведінки ПЗ. До опису поведінки включено множину як функціональних, так і нефункціональних вимог для опису взаємодії роботи користувачів з ПЗ.

## 2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Діаграма сценаріїв використання (use case)

Use Case або сценарій використання - це опис послідовності дій користувача або системи, які ілюструють те, як користувачі взаємодіють з системою для досягнення певної мети. У такому сценарії описується, як користувачі виконують дії, які дані вони вводять та який результат вони очікують від системи. Це дає можливість зрозуміти та сформулювати вимоги до функціоналу системи та визначити, як вона повинна працювати.

Даний застосунок розробляється у сфері dating, що є дуже популярним на сьогоднішній день. Застосунки для онлайн-знайомств мають велику ЦА, що дає користувачу широкий вибір для знайомств.

Основна задумка застосунку, що проєктується – дати можливість без перешкод знайомитись з новими людьми по всьому світу. Важливою складовою є наявність підключення смартфона користувача до мережі Інтернет, яке буде давати змогу користуватись застосунком для онлайн-знайомств.

#### *Короткий Use Case*

Користувач встановлює застосунок для онлайн-знайомств на свій пристрій. Пристрій користувача запускає застосунок та пропонує зареєструватись через соціальну мережу Facebook. Користувач логінується у застосунку та починає ним користуватися.

#### *Поверхневий Use Case*

##### *Головний сценарій (успішний)*

Користувач встановлює застосунок на свій пристрій. Гаджет запускає застосунок та його перекидає на сторінку, де будуть відображатися інші користувачі. Далі користувач застосунку починає шукати собі друзів або пару ставлячи лайки в профілі. Якщо симпатія двох людей співпаде, то на екрані користувачів буде виводитися повідомлення, що їх симпатія один до одного взаємна. Також у подальшому буде введена система репортів.

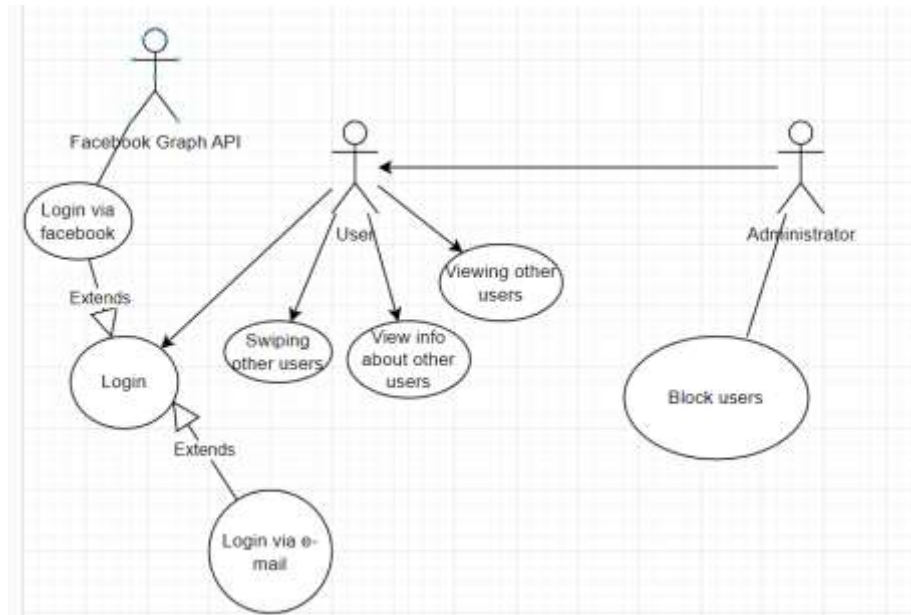


Рисунок 2.1 – Діаграма UseCase

Якщо користувач буде погрожувати іншим або буде намагатись отримати сексуальний контент від інших, то інші люди зможуть відправити скаргу на недопустиму поведінку іншого користувача по відношенню до них. Далі адміністратор будет залишати право за собою блокувати в разі порушення чи ні.

## 2.2 Діаграми діяльності

Діаграма діяльності є типом діаграми UML (Unified Modeling Language), який описує послідовність дій, які виконуються в процесі взаємодії між об'єктами або користувачами в системі. Головною метою діаграм діяльності є візуалізація робочих процесів або операцій.

Діаграма діяльності складається з блоків, які зображують дії, та стрілок, які вказують порядок виконання цих дій. Кожен блок може мати один або більше входів та виходів.

Застосунок для онлайн-знайомств повинен мати зручний і зрозумілий інтерфейс, оскільки люди зараз звикли до інтуїтивно-зрозумілого інтерфейсу. Щоб відобразити основні дії, які можуть здійснити користувачі зазвичай використовують діаграми діяльності, що є аналогом блок-схем і візуально описує логічну

послідовність дій користувача та можливих розгалуджень системи. Створено кілька діаграм діяльності, що наведені нижче:

*Користувачі зловили метч.* (рис. 2.3). Дана діаграма відображає подію, коли два користувачі зійшлися у своїй симпатії. Якщо обидва користувачі поставили один одному лайк, то це створює між ними event, який називається метч.

*Відсутнє інтернет з'єднання.* Ця діаграма відображає діяльність, коли у пристроя користувача не має з'єднання з мережею Інтернет. Якщо у користувача нема доступу до Інтернету, то йому покаже, що інтернет з'єднання відсутнє.

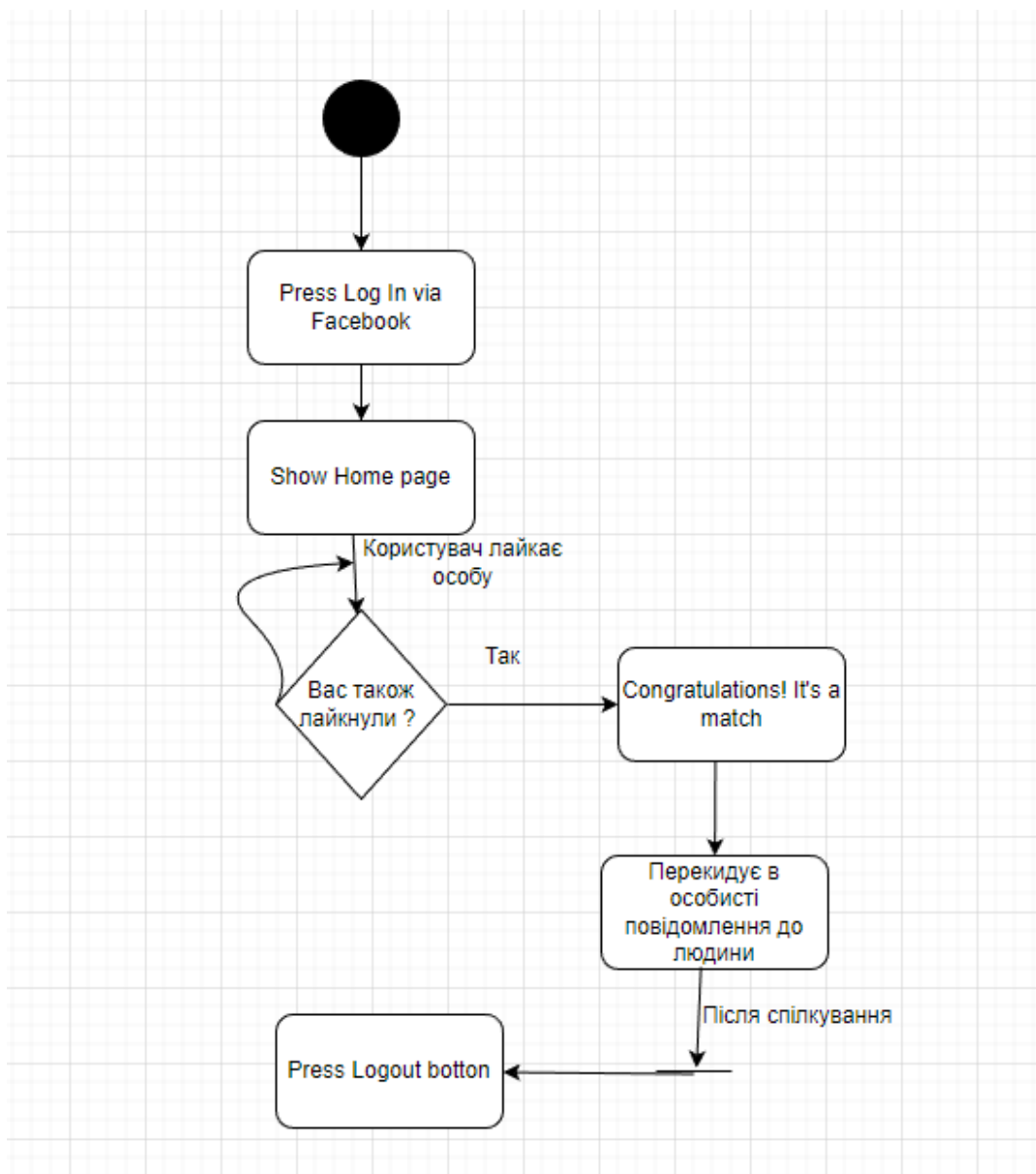


Рисунок 2.3 – Діаграма діяльності «Користувачі зловили метч»



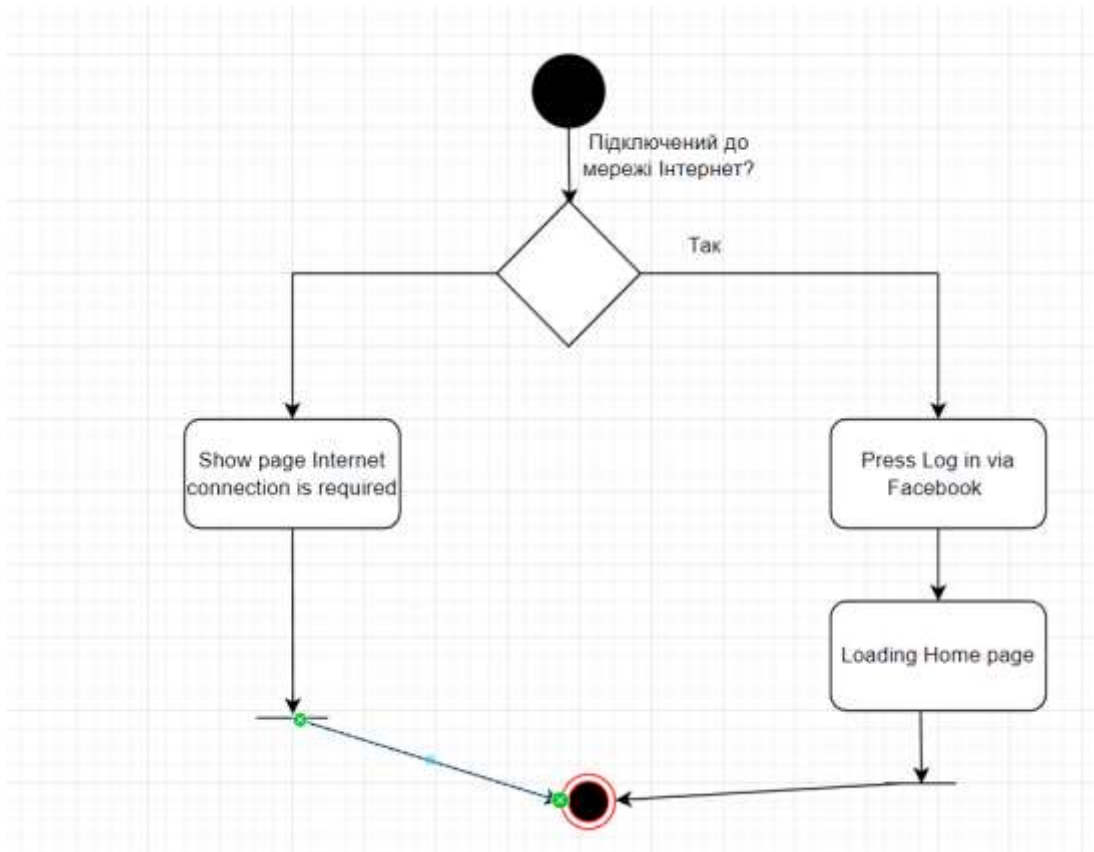


Рисунок 2.4 – Діаграма діяльності «пристрій користувача не підключен до мережі Інтернет»

Діаграми було створені за допомогою спеціалізованого програмного забезпечення моделювання та конструювання Star UML для демонстрації деяких процесів дії користувача застосунку.

### 2.3 Діаграми пакетів

Діаграма пакетів (Package diagram) - це тип структурної діаграми в UML, яка відображає структуру програмної системи на основі пакетів та залежностей між ними. Нижче показано діаграму пакетів для усього застосунку (рис. 2.5).

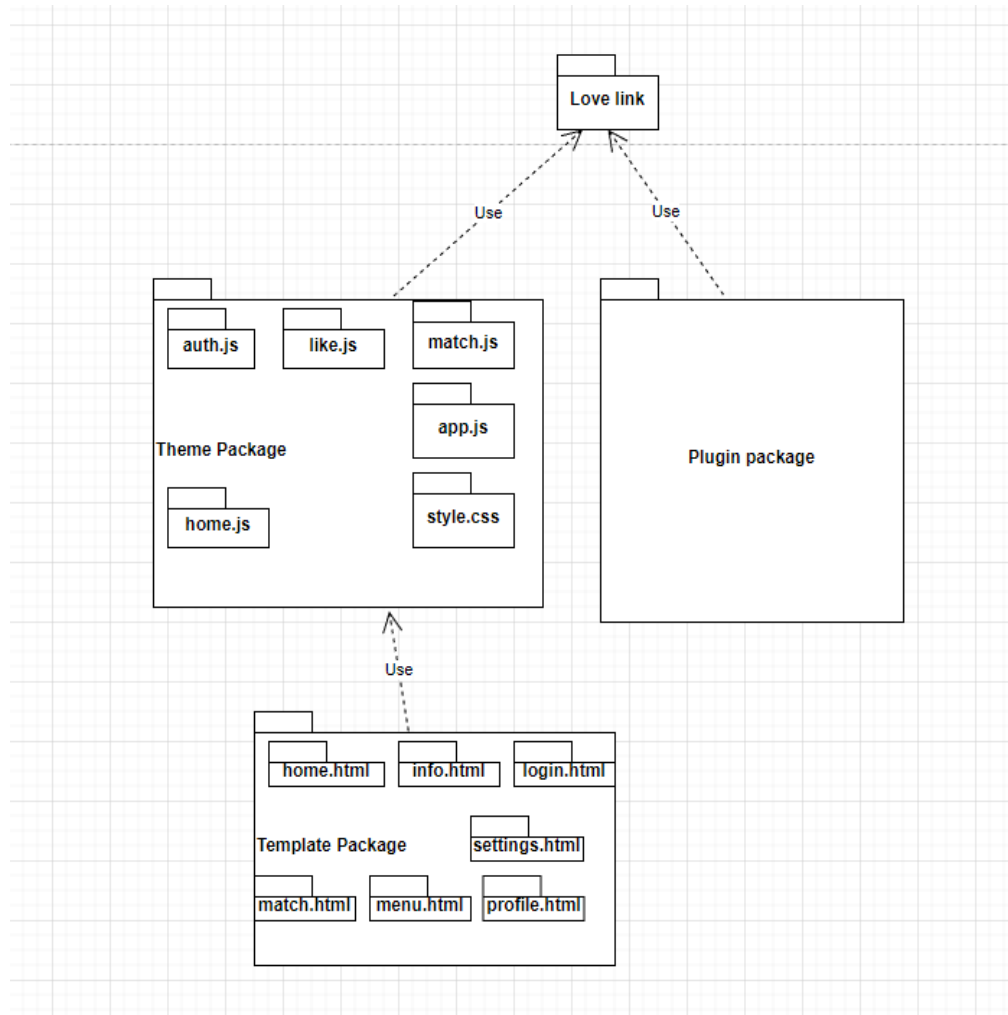


Рисунок 2.5 – Діаграма пакетів

Plugin Package – це набір файлів плагінів для використання в Ionic Framework. В цьому пакеті зазвичай містяться файли, необхідні для підключення додаткових функціональних можливостей до застосунку. Theme Package – це пакет, що містить файли теми Ionic Framework. В ньому зазвичай містяться функції, скрипти, зображення та інші частини зовнішнього вигляду застосунку. Використання готових тем може значно спростити розробку додатку та підвищити його зовнішній вигляд. Template Package – це пакет, що містить набір шаблонів для різних типів сторінок в Ionic Framework.

## 2.4 Діаграми розгортання

Діаграма розгортання - це тип діаграми, що відображає фізичну конфігурацію системи, тобто розташування та взаємодію між апаратними та програмними компонентами системи, які відображаються у вигляді вузлів та з'єднань між ними. Ця діаграма може бути використана для опису розташування серверів, комп'ютерів, з'єднань між ними, а також компонентів програмного забезпечення, які запускаються на кожному вузлі. При роботі застосунку задіяні три вузли – база даних, вебсервер застосунку та пристрій користувача. На пристрої користувача виконуються скрипти, на сервері обробляються запити користувача та надсилаються запити до бази даних. База даних в свою чергу повертає результати запитів (рис. 2.6).

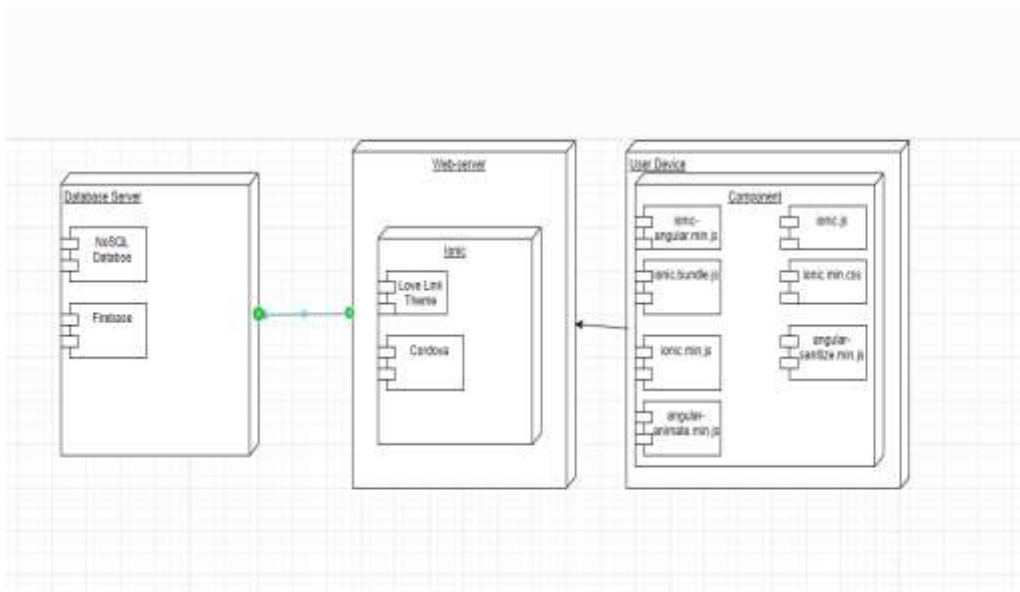


Рисунок 2.6 – Діаграма розгортання

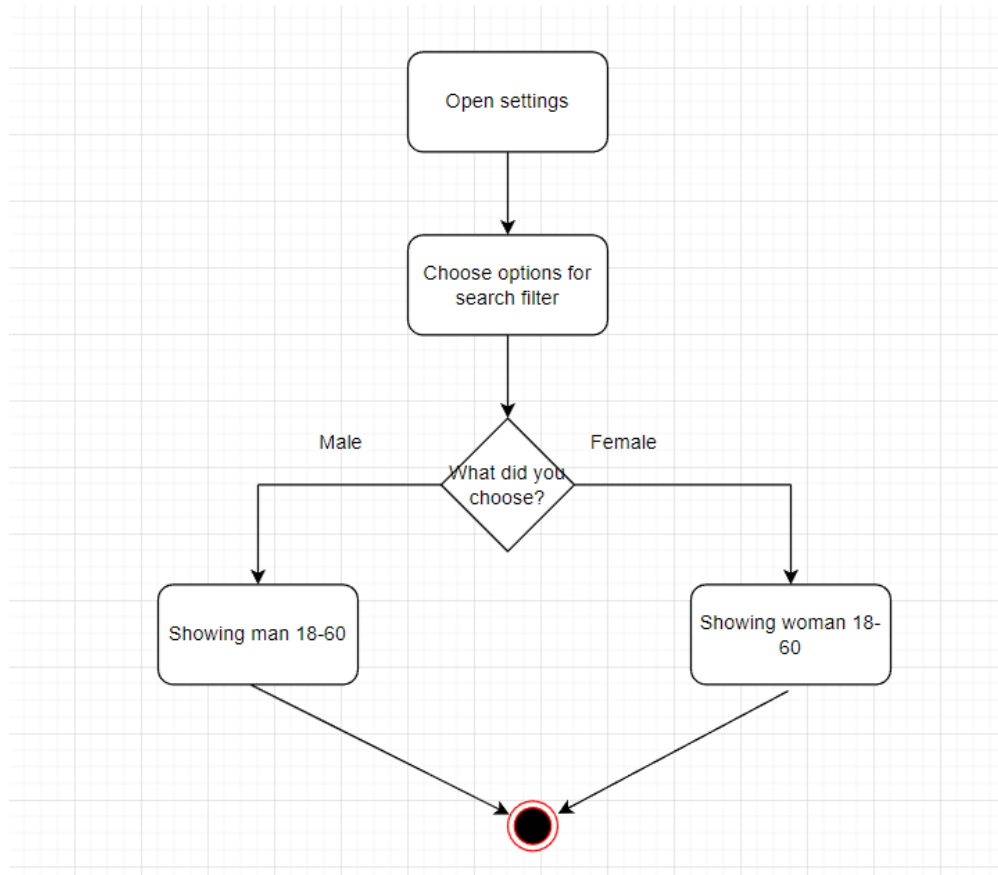


Рисунок 2.7 – Діаграма діяльності «Фільтрація»

На рисунку 2.7 зображено діаграму діяльності фільтрації пошуку для користувачів. Діаграма показує, що демонстрація людей буде згідно заданим параметрам фільтра.

## Висновки до розділу 2

В другому розділі було описано основні етапи, сценарії використання та алгоритми роботи застосунку для онлайн-знайомств. Розроблено UML-діаграми, а саме діаграму пакетів, діаграму розгортання, діаграму UseCase та інші, завдяки чому закріплено навички описувати алгоритми та архітектуру застосунку за допомогою UML. На основі діаграми використання складено сценарії використання, після розробки короткого та поверхневого сценарія використання, складено повного usecase, що містить в собі всі деталі використання ПЗ.

## 3 ОГЛЯД СТЕКУ ЗАДІЯНИХ ТЕХНОЛОГІЙ

### 3.1 Оглядання технологій

Для створення застосунку обрано наступний стек технологій:

- Мова програмування JavaScript, а саме використання фреймворку AngularJS ;
- Фреймворк для розробки мобільних додатків з використанням веб-технологій Ionic Framework;
- NoSQL та Firebase для зберігання даних користувачів;
- Sublime Text 3;
- Node.js для виконання коду JavaScript на стороні сервера;



Рисунок 3.1 – Основний стек технологій

### **Ionic Framework**

Ionic Framework - це популярний інструментарій для розробки мобільних додатків з використанням веб-технологій, таких як HTML, CSS та JavaScript. Ось деякі переваги використання Ionic для проєктів:

- Кросплатформеність: Ionic дозволяє розробляти мобільні додатки для різних платформ, таких як iOS та Android, використовуючи один і той же код. Це

значно знижує час і зусилля, потрібні для розробки та підтримки додатків на кількох платформах.

– Візуальна привабливість: Ionic надає доступ до бібліотеки стилів та компонентів, які дозволяють створювати візуально привабливі додатки. Бібліотека містить різні стилі та компоненти, такі як кнопки, форми, списки та інші, що спрощують розробку додатків з красивим та сучасним інтерфейсом.

– Простота використання: Ionic має простий та зрозумілий API, що дозволяє розробникам швидко та ефективно створювати функціональні додатки.

– Підтримка плагінів Cordova: Ionic має вбудовану підтримку плагінів Cordova, що дозволяє додаткам отримувати доступ до різних функцій пристрою, таких як камера, геолокація та інші.

– Розширюваність: Ionic дозволяє розширювати функціональність додатків за допомогою різних модулів та компонентів.

Через це було і обрано використання фреймворку Ionic Framework.



Рисунок 3.2 – Логотип Ionic Framework

## JavaScript

JavaScript - це мова програмування, яку широко використовують для створення веб-додатків. Вона призначена для взаємодії з користувачами на стороні клієнта, тобто у браузері.

JavaScript має кілька особливостей, які роблять її потужним інструментом для розробки веб-додатків. За допомогою JavaScript можна створювати динамічні веб-сторінки, які реагують на дії користувача. Вона надає можливість змінювати вміст сторінки, валідувати дані та створювати захоплюючі ефекти.

Синтаксис JavaScript схожий на інші мови програмування, що полегшує його вивчення. Вона підтримує різні типи даних, такі як числа, рядки, масиви та об'єкти. Крім того, JavaScript є динамічною мовою, що дозволяє змінювати типи даних під час виконання програми.

JavaScript підтримує об'єктно-орієнтований підхід до програмування, що дозволяє створювати класи, об'єкти та використовувати принципи спадкування та поліморфізму. Це сприяє структуруванню коду та полегшує повторне використання.

JavaScript також має велику кількість бібліотек та фреймворків, таких як React, Angular та Vue.js, які допомагають розробникам створювати складні веб-додатки. Ці інструменти надають розширені можливості, такі як створення компонентів, управління станом та маршрутизація.

За допомогою платформи Node.js, JavaScript може використовуватись і на стороні сервера. Це дозволяє розробникам писати серверний код з використанням JavaScript, що забезпечує єдність мови програмування в усьому стеку додатка.

JavaScript є однією з найпопулярніших мов програмування у світі веб-розробки. Вона дозволяє створювати потужні, інтерактивні та захоплюючі веб-додатки, і її використовують мільйони розробників по всьому світу.



Рисунок 3.2 – Логотип JavaScript

## AngularJS

AngularJS - це високопопулярний фреймворк JavaScript, створений компанією Google. Його випущено в 2010 році і він швидко став улюбленим інструментом для розробки потужних односторінкових веб-додатків.

Основна перевага AngularJS полягає в його здатності розширювати HTML та надавати декларативні можливості програмування. Це досягається завдяки концепції двостороннього зв'язку даних, яка автоматично синхронізує дані між моделлю (дані додатка) та представленням (відображенням на сторінці).

AngularJS має низку важливих особливостей, які варто відзначити:

- Модульність: Фреймворк дозволяє розбити додаток на невеликі, самостійні модулі, що спрощує організацію та керування кодом.
- Керування залежностями: Завдяки механізму впровадження залежностей, AngularJS дозволяє легко підключати та використовувати різні компоненти.
- Директиви: Цей фреймворк має вбудовані та можливість створювати власні директиви, які розширюють HTML функціональністю. Це дає змогу створювати перевикористовувані компоненти та контролювати поведінку додатка.
- Фільтри: AngularJS надає можливість використовувати фільтри для обробки та форматування даних перед їх відображенням. Ви можете використовувати вбудовані фільтри або створити власні.
- Тестування: Цей фреймворк надає розширені можливості для тестування додатків, включаючи юніт-тести, інтеграційні тести та автоматизовані тести, що допомагають перевірити правильність функціонування додатку.
- Роутинг: AngularJS має вбудований механізм маршрутизації, який дозволяє створювати посилання, що відкривають різні сторінки додатка без перезавантаження.



- Підтримка Ajax: Зручні засоби для роботи з асинхронними запитами дозволяють взаємодіяти з сервером та оновлювати дані на сторінці без перезавантаження її.
- Варто відзначити, що AngularJS перетворився на нову ітерацію - Angular, що пропонує ще більше можливостей та покращень. Проте, AngularJS все ще підтримується, особливо для існуючих проєктів.

AngularJS є правильним вибором для використання його разом з Ionic Framework і ось чому:

- Кросплатформовість: Ionic Framework підтримує платформи Android, iOS та Windows. AngularJS, з іншого боку, може бути використаний для розробки веб-додатків на будь-якій платформі.
- Простота: AngularJS є дуже простим та легким у використанні фреймворком, що дозволяє легко створювати і підтримувати код.
- Швидкість розробки: використання AngularJS в Ionic Framework дозволяє розробникам швидко створювати мобільні додатки без необхідності вивчення нових технологій.
- Масштабованість: AngularJS дозволяє створювати масштабовані додатки зі складним функціоналом, що дозволяє розширювати додатки в майбутньому без необхідності переписувати весь код.
- Спільнота: AngularJS та Ionic Framework мають велику спільноту розробників, яка надає безкоштовну та швидку підтримку та розвиток цих технологій.



Рисунок 3.3 – Логотип AngularJS

## NoSQL

Звичайні бази даних, які ми звикли бачити, мають таблиці з рядками та стовпцями. Але є інший тип баз даних, які використовуються для зберігання і отримання даних - їх називають NoSQL. NoSQL - це не тільки SQL, альтернативний підхід до управління даними, який надає більшу гнучкість.

### Особливості NoSQL:

- Гнучкість схеми: На відміну від традиційних баз даних, NoSQL не вимагає жорсткого задання схеми даних перед зберіганням. Ви можете додавати, змінювати або видаляти поля в документах або ключі та значення в асоціативних масивах.
- Масштабованість: NoSQL дозволяє розподіляти дані на кластери серверів, що дозволяє легко масштабувати систему для обробки великих обсягів даних.
- Швидкодія: NoSQL забезпечує швидке зберігання та отримання даних завдяки оптимізованим алгоритмам доступу та індексації.

- Різноманітність моделей даних: NoSQL підтримує різні моделі даних, такі як документи, ключі та значення, колонки або графи, що дає більшу свободу при моделюванні даних.
- Географічна розподіленість: Деякі NoSQL-системи дозволяють розподілити дані на різних серверах у різних регіонах світу, що дозволяє створювати глобально розподілені додатки з швидким доступом до даних.
- Простота використання: NoSQL мають простий і зрозумілий інтерфейс, що полегшує роботу з базами даних і програмування на них.

NoSQL - це потужний інструмент, який дозволяє зберігати та опрацьовувати дані у зручній та гнучкій спосіб.

## **Firestore**

Firestore - це платформа розробки мобільних та веб-додатків, яка надає набір інструментів та сервісів для полегшення роботи з розробкою додатків. Вона створена командою Google з метою полегшити життя розробників та надати їм потужні інструменти для розробки додатків з широким функціоналом.

Firestore пропонує різноманітні компоненти та сервіси:

- База даних в реальному часі: Ви можете використовувати Firestore для зберігання та синхронізації даних у режимі реального часу. Це означає, що зміни, внесені на одному пристрої, негайно відображаються на інших пристроях.
- Аутентифікація користувачів: Firestore надає можливість налаштувати аутентифікацію користувачів у вашому додатку. Ви можете використовувати різні методи, такі як електронна пошта, соціальні мережі та номери телефонів, для авторизації користувачів.
- Документна база даних: Firestore дозволяє зберігати структуровані дані за допомогою гнучкої документної бази даних. Ви можете легко створювати, оновлювати та отримувати дані з Firestore Cloud Firestore.

- Зберігання файлів: Firebase також надає можливість зберігати файли, такі як зображення чи відео, за допомогою Firebase Storage. Це допомагає вам зберігати та керувати мультимедійними ресурсами вашого додатку.
- Функції в хмарі: Firebase Cloud Functions дозволяє вам створювати власні функції, які виконуються на серверах Firebase. Ви можете використовувати їх для обробки подій, відправки сповіщень та виконання складних обчислень.
- Хостинг: Firebase Hosting - це сервіс, який дозволяє розгорнути ваш веб-сайт або веб-додаток на хмарних серверах. Він надає швидке та просте розгортання, щоб ваш додаток був доступний для користувачів усього світу.
- Сповіщення: Firebase Cloud Messaging дозволяє надсилати сповіщення на мобільні пристрої. Ви можете надсилати користувачам повідомлення про оновлення, нові події та іншу важливу інформацію.



Рисунок 3.4 – Логотип Firebase

### **Переваги NoSQL і Firebase**

NoSQL і Firebase мають кілька переваг в порівнянні з традиційними реляційними базами даних, такими як MySQL або PostgreSQL. Ось кілька переваг використання NoSQL та Firebase:

- Гнучкість: NoSQL дозволяє зберігати дані без схеми даних, що дає більшу гнучкість в зміні структури даних. Firebase також забезпечує гнучкість шляхом динамічного зберігання даних у форматі JSON, що дозволяє швидко і просто змінювати структуру даних.

- Швидкість: NoSQL зазвичай працює швидше, оскільки використовує простіші запити та не має складних операцій з'єднання таблиць. Firebase також має дуже швидку роботу завдяки використанню реального часу та миттєвій синхронізації даних між пристроями.
- Розшарування та масштабованість: NoSQL зазвичай має кращу масштабованість та розшарування, оскільки дані можуть зберігатися на кількох серверах. Firebase також має хорошу масштабованість та розшарування, оскільки може легко працювати з кількома пристроями та різними платформами.
- Простота використання: NoSQL та Firebase дуже прості у використанні та дозволяють швидко розгорнути базу даних та налаштувати її на потреби проекту. Firebase, зокрема, має дуже простий API та дуже зручний інтерфейс для роботи з даними.

## **Node.js**

Node.js - це відкрите серверне середовище виконання JavaScript. Node.js наймовірно корисна річ для розробників. Воно дозволяє нам використовувати JavaScript не лише у браузері, але й на сервері. І це чудово! За допомогою Node.js ми можемо створювати потужні серверні додатки та веб-сервери, які надають швидкодію та надійність.

Особливості Node.js прості та чудові. Воно використовує двигун V8, той самий, який використовується в Google Chrome, що робить виконання JavaScript-коду швидким і ефективним. Це означає, що наші додатки будуть працювати швидко і зможуть обслуговувати багатокористувацькі запити без проблем.

Ще одна крута річ про Node.js - це його здатність до асинхронного програмування. Він використовує події та зворотні виклики, щоб обробляти багато запитів одночасно, не блокуючи виконання інших операцій. Це дозволяє нам писати додатки, які можуть обробляти багато різних речей одночасно, забезпечуючи нам швидку та плавну роботу.

Ще одна приємна риса Node.js - це його модульність. Ми можемо використовувати різні модулі, щоб розширити функціональність наших додатків. Node.js має велику кількість пакетів та бібліотек, які ми можемо використовувати за допомогою npm (пакетний менеджер Node.js). Це забезпечує нам легкість у встановленні та управлінні залежностями наших проектів.

Node.js є надзвичайно популярним серед розробників, і зараз ми бачимо все більше і більше веб-додатків, API та сервісів, побудованих на цій технології. Вона дає нам можливість розробляти швидкі, потужні та масштабовані додатки, які задовольняють потреби наших користувачів.

Таким чином, Node.js - це потужний інструмент, який дозволяє нам розробляти захоплюючі веб-додатки та сервіси. Він спрощує нашу роботу, забезпечує швидкість та надійність, і допомагає нам досягти успіху у світі веб-розробки.

Node.js має кілька переваг для розробки програмного забезпечення:

- Швидкість виконання: Node.js розроблений на базі двигуна V8 JavaScript від Google, що забезпечує високу швидкість виконання коду.
- Спрощений розробка серверної частини: Node.js пропонує простий та зручний інтерфейс для розробки серверної частини додатків, що зменшує час та зусилля, необхідні для розробки.
- Однаковість мов програмування: JavaScript використовується як мова програмування на стороні клієнта та сервера, що дозволяє зменшити зусилля та час, необхідні для навчання та розуміння коду.
- Масштабованість: Node.js пропонує пакети та бібліотеки для масштабування додатків, що дозволяє збільшувати об'єм роботи, без зниження продуктивності.
- Доступність: Node.js є відкритим джерелом та безкоштовним для використання, що зменшує витрати на розробку програмного забезпечення.

- Розширюваність: Node.js має велику кількість розширень та модулів, що дозволяє виконувати різні завдання, такі як робота з базами даних, відправка електронних листів, розробка API та багато іншого.



Рисунок 3.5 – Логотип Node.js

### **Facebook Graph API**

Facebook Graph API - це інтерфейс програмування застосунків (API), який надає розробникам доступ до різноманітних даних та функцій платформи Facebook. Цей API дозволяє розробникам створювати додатки, які взаємодіють зі соціальною мережею Facebook і отримують доступ до користувачів, їх даних та діяльності.

Особливості Facebook Graph API:

Отримання доступу до користувачів: За допомогою Graph API розробники можуть отримувати доступ до публічної інформації про користувачів, такої як ім'я, профільна картинка, друзі, статус та інше. Це дозволяє створювати персоналізовані додатки та взаємодіяти з користувачами на платформі Facebook.

Розміщення повідомлень із додатків: Розробники можуть використовувати Graph API для розміщення повідомлень, фотографій, відео та іншого контенту від імені користувачів. Це дає можливість додаткам публікувати актуальну інформацію та взаємодіяти з аудиторією.

Управління сторінками та групами: Graph API дозволяє розробникам отримувати доступ до сторінок та груп на Facebook. Це включає можливість публікувати, взаємодіяти та аналізувати діяльність сторінок та груп.

Аналітика та статистика: За допомогою Graph API розробники можуть отримувати доступ до аналітичних даних та статистики, пов'язаних зі своїми додатками та сторінками. Це дозволяє вимірювати ефективність додатків та розуміти поведінку користувачів.

Інтеграція з іншими сервісами: Facebook Graph API дозволяє розробникам інтегрувати свої додатки з іншими сервісами та платформами. Наприклад, це може бути інтеграція з мобільними додатками, веб-сервісами чи сторонніми інструментами.

Facebook Graph API надає розробникам потужний інструментарій для взаємодії з платформою Facebook. З його допомогою можна створювати різноманітні додатки, які відповідають потребам користувачів і дозволяють взаємодіяти з ними на рівні соціальної мережі Facebook.

Facebook Graph API використовує HTTP протокол для комунікації між клієнтами та сервером. Клієнтські додатки можуть здійснювати запити до API за допомогою HTTP методів, таких як GET, POST, PUT, DELETE і PATCH. Кожен запит має URL, який містить інформацію про ресурс, який необхідно отримати або змінити, а також параметри запиту, такі як фільтри, поля, сортування тощо.

Прийнятий формат відповіді в Facebook Graph API - JSON, що є стандартом в Інтернеті для обміну даними. Отримана відповідь може містити інформацію про ресурс, який був запитаний, а також додаткову інформацію, таку як пейджінг, метадані та інші корисні дані.



# Graph API



Рисунок 3.6 – Логотип Graph API

Для взаємодії з Facebook Graph API, розробники можуть використовувати різні бібліотеки, які надають зручний інтерфейс для взаємодії з API, такі як axios, request, fetch тощо.

## Висновки до розділу 3

В третьому розділі було проаналізовано стек задіяних технологій, а саме: мову програмування JavaScript та її фреймворк AngularJS, платформу розробки мобільних та веб-додатків Firebase, популярний інструментарій для розробки мобільних додатків IonicFramework, відкрите серверне середовище виконання JavaScript Node.js, інтерфейс програмування застосунків (API) – Facebook GraphAPI.

Під час огляду технологічного стеку та проведення аналізу актуальних мов програмування, фреймворків та баз даних для веб-застосунків, було визначено підходящі технології для розробки як front-end, так і back-end частини. Основано на отриманих результатах, був здійснений обґрунтований вибір програмних засобів для реалізації онлайн-застосунку для знайомств.

## 4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Застосунок для онлайн знайомств «LoveLink» містить в собі 3 сторінки ( головна сторінка, налаштування для фільтрації та перегляд свого профілю) для користувача, який залогінувся. Застосунок, після реалізації, можна буде розмістити для роботи в мережі Інтернет, після чого з’явиться змога у користувачів користуватися застосунком. Застосунок створено для одного типу користувачів, а саме клієнтів. Роль клієнта відповідає створеним діаграмам use cases. Згідно сформованим вимогам та функціоналу системи, користувач має можливість виконувати такі дії: логінитися, лайкати або дізлайкати інших користувачів, дивитися інформацію про інших клієнтів, ставити фільтрацію для пошуку.

### 4.1 Завантаження та налаштування технологій

Спочатку потрібно завантажити AngularJS. Для цього перейдемо на офіційний сайт AngularJs.

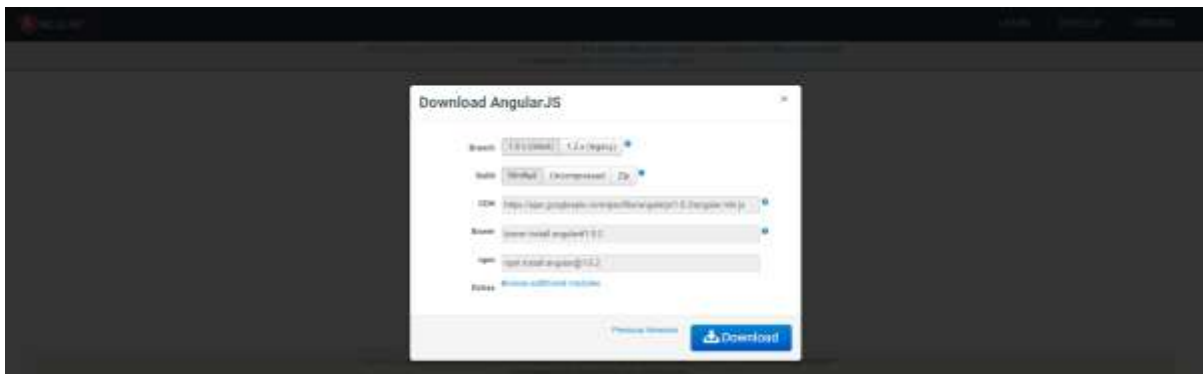


Рисунок 4.1 – Завантаження AngularJS

Після цього завантажуюмо Node.js також з офіційного сайту.



Рисунок 4.2 – Завантаження Node.js

Далі потрібно встановити Ionic Framework



Рисунок 4.3 – Офіційний сайт Ionic Framework



Рисунок 4.4 – Завантаження Ionic Framework на PC

Також нам потрібен текстовий редактор для реалізації проєкту. Для цього зайдемо на офіційний сайт Sublime Text та завантажимо його на PC.

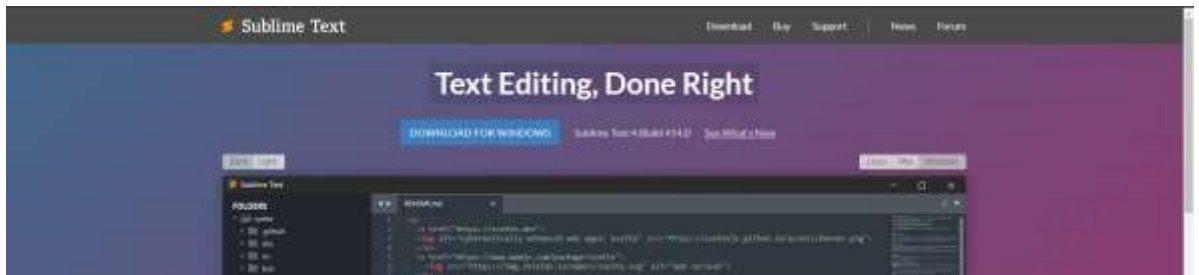


Рисунок 4.5 – Офіційний сайт Sublime Text

І наостанок потрібно створити проєкт у Firebase, який буде використано для зберігання даних користувачів.



Рисунок 4.6 – Проєкт у Firebase

## 4.2 Реалізація застосунку для онлайн-знайомств

### Реєстрація та авторизація

Реєстрація є необхідною складовою функціоналу веб-сайту, і більшість веброзробників включають можливість реєстрації та авторизації. Авторизовані користувачі мають привілеї, які недоступні неавторизованим користувачам, оскільки останні можуть лише переглядати сторінки без можливості внесення змін або додавання контенту на сайті. Це зумовлено тим, що авторизація є процесом ідентифікації користувача та надання йому спеціальних привілеїв або доступу до певної інформації чи функцій.

Застосунок для онлайн знайомств також використовує авторизацію через соціальну мережу Facebook, оскільки інакше користувач не зможе використовувати даний застосунок.

За можливість авторизації користувача відповідають наступні файли проєкту:

1) За візуальну частину ( оформлення та відображення ):

- style.css;
- login.html;

2) За авторизацію

- auth.js (services);
- auth.js (controllers);

```
login.html x
1 <ion-view view-title="Login">
2   <ion-content class="landing">
3     <div class="facebook">
4       <button class="button button-positive facebook-button" ng-click="auth.login()">
5         <i class="ion-social-facebook big"></i> Login with Facebook
6       </button>
7     </div>
8   </ion-content>
9 </ion-view>
```

Рисунок 4.7 – Фрагмент коду login.html

```
auth.js x
1 'use strict';
2
3 app.factory('Auth', function(FURL, $firebaseAuth, $firebaseObject, $state) {
4
5   var ref = new Firebase(FURL);
6   var auth = $firebaseAuth(ref);
7
8   var Auth = {
9     login: function() {
10      return auth.$authWithOAuthPopup('facebook', {
11        remember: "sessionOnly",
12        scope: "public_profile, email, user_location, user_birthday, user_photos, user_about_me"
13      });
14    },
15
16    logout: function() {
17      return auth.$unauth();
18    }
19  };
20
21  auth.$onAuth(function(authData) {
22    if(authData) {
23      console.log('Logged in!');
24    } else {
25      $state.go('login');
26      console.log('You need to login.');
```

Рисунок 4.8 – Фрагмент коду auth.js (services)

```
auth.js
1 'use strict';
2
3 app.controller('AuthCtrl', function(Auth) {
4
5     var auth = this;
6
7     auth.login = function() {
8         console.log('Login clicked');
9
10        return Auth.login().then(function(user) {
11            console.log(user.facebook);
12        });
13
14    };
15
16    auth.logout = function() {
17        Auth.logout();
18    };
19 });
```

Рисунок 4.9 – Фрагмент коду auth.js (controllers)

Адаптивний інтерфейс впливає перш за все на зовнішній вигляд сторінки авторизації користувача. Коли браузер розгорнутий на повний екран, сторінка авторизації виглядає як на зображенні (4.10). Проте, якщо вікно браузера стиснуте і для звичайного відображення сторінки недостатньо місця, сторінка стає компактнішою, як показано на зображенні (4.11).

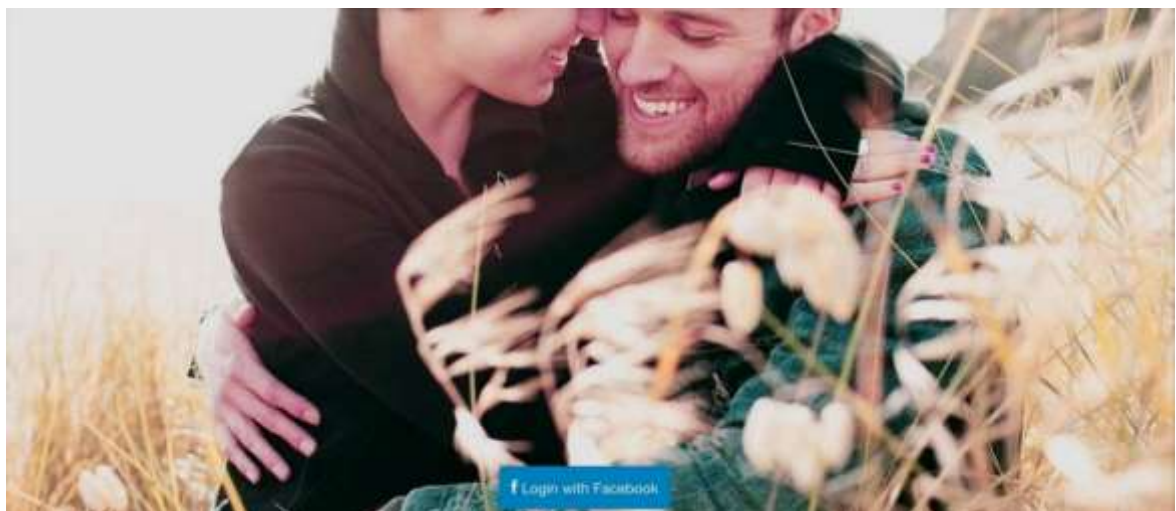


Рисунок 4.10 – Вигляд сторінки авторизації при повному вікні

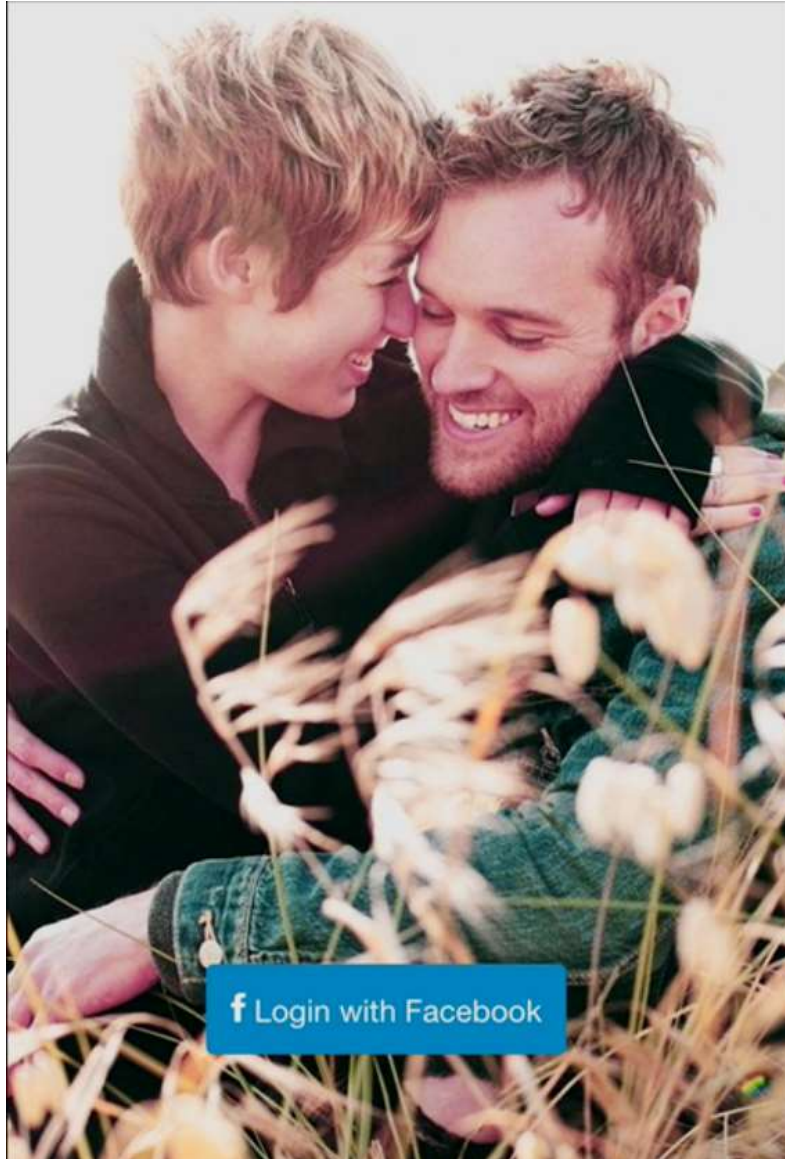


Рисунок 4.11 – Вигляд сторінки авторизації при зменшеному вікні

### **Функція лайків**

Функціонал лайків є необхідною складовою багатьох застосунків для знайомств, що дозволяє користувачам виражати свої інтереси та виразити сподобання до інших користувачів. Тому застосунок для онлайн знайомств не є виключенням.

```
like.js
1 'use strict';
2
3 app.factory('Like', function(FURL, $firebaseArray) {
4
5     var ref = new Firebase(FURL);
6
7     var Like = {
8
9         allLikesByUser: function(uid) {
10             return $firebaseArray(ref.child('likes').child(uid));
11         },
12
13         addLike: function(uid1, uid2) {
14             return ref.child('likes').child(uid1).child(uid2).set(true);
15         },
16
17         removeLike: function(uid1, uid2) {
18             return ref.child('likes').child(uid1).child(uid2).remove();
19         }
20     };
21
22     return Like;
23
24 });
```

Рисунок 4.12 – Фрагмент коду like.js

```
<script src="js/services/like.js"></script>
```

Рисунок 4.13 – Підключення like.js в index.html

### Функція фільтрування пошуку

Функція фільтрації пошуку у застосунках для знайомств має значний вагомий вплив. Вона дозволяє користувачам більш точно визначити свої вподобання та встановити параметри пошуку, щоб знайти потенційних партнерів, які відповідають їх вимогам і критеріям. Також фільтрація допомагає зекономити час користувачів, який витрачається на перегляд великої кількості профілів.



```
settings.html
1 <ion-view view-title="Settings">
2   <ion-content>
3
4     <div class="list padding">
5       <div class="item item-divider">Show Age (18 - {{sett.maxAge}})</div>
6
7       <div class="item range range-positive">
8         18<input type="range" name="age" min="18" max="60" value="25"
9           ng-model="sett.maxAge"
10          on-release="sett.changeMaxAge()">60
11       </div>
12
13       <div class="item item-divider">Show Me</div>
14
15       <ion-toggle toggle-class="toggle-balanced"
16         ng-model="sett.men"
17         ng-checked="{{sett.men}}"
18         ng-change="sett.selectMen()"
19       >Men
20     </ion-toggle>
21     <ion-toggle toggle-class="toggle-balanced"
22       ng-model="sett.women"
23       ng-checked="{{sett.women}}"
24       ng-change="sett.selectWomen()"
25     >Women
26   </ion-toggle>
27
28   <button class="button button-block button-assertive" ng-click="sett.logout()">Logout</button>
29
30 </div>
31 </ion-content>
32 </ion-view>
```

Рисунок 4.14 – Фрагмент коду settings.js

Вони можуть встановити конкретні критерії, що дозволяють звузити вибірку до більш обмеженої групи профілів, які відповідають їхнім потребам і уподобанням.



Рисунок 4.15 – Налаштування фільтрації пошуку

## Функція “Match” між користувачами

Функція "match" є надзвичайно важною в застосунках для знайомств, оскільки вона дозволяє знайти потенційних партнерів, що відповідають вподобанням і інтересам користувачів.

```
match.js
1 'use strict';
2
3 app.factory('Match', function(FURL, $firebaseArray, $ionicPopup) {
4
5     var ref = new Firebase(FURL);
6
7     var Match = {
8
9         allMatchesByUser: function(uid) {
10             return $firebaseArray(ref.child('matches').child(uid));
11         },
12
13         checkMatch: function(uid1, uid2) {
14             var check = ref.child('likes').child(uid2).child(uid1);
15
16             check.on('value', function(snap) {
17                 if (snap.val() != null) {
18                     ref.child('matches').child(uid1).child(uid2).set(true);
19                     ref.child('matches').child(uid2).child(uid1).set(true);
20
21                     $ionicPopup.alert({
22                         title: 'Matched',
23                         template: 'Yay, you guys are matched!'
24                     });
25                 }
26             })
27         },
28
29         removeMatch: function(uid1, uid2) {
30             ref.child('matches').child(uid1).child(uid2).remove();
31             ref.child('matches').child(uid2).child(uid1).remove();
32         }
33     };
34 };
35
36 return Match;
37
38 });
```

Рисунок 4.16 – Фрагмент коду match.js

Вона сприяє підвищенню сумісності, ефективному пошуку, психологічному комфорту та уникненню небажаних зв'язків. Також без цієї функції клієнт не зрозуміє чи симпатія взаємна.

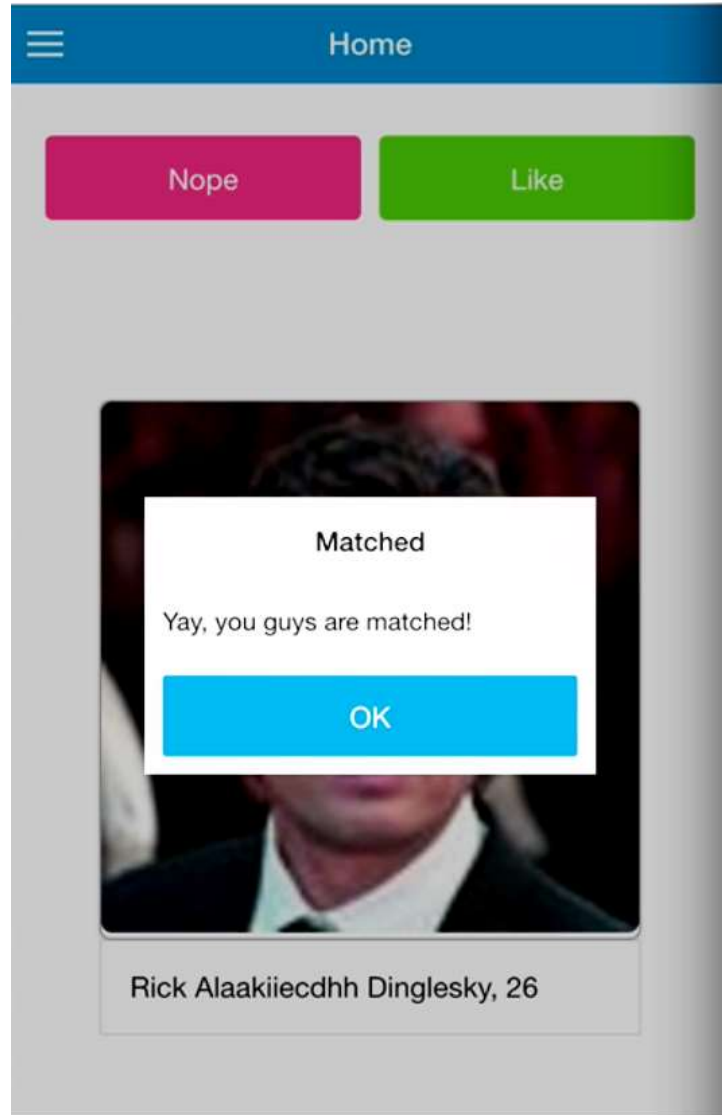


Рисунок 4.17 – Демонстрація роботи функції match

**Функція перегляду інформації про інших користувачів.**

Однією з функцій застосунку є перегляд інформації інших користувачів. Ця функція дозволяє користувачам вирішити для себе чи підходить людина по інтересам, віку, вподобанням і тд.

```
1 <ion-modal-view>
2   <ion-header-bar>
3     <h1 class="title">Profile</h1>
4     <div class="buttons">
5       <button class="button button-clear" ng-click="home.closeModal()">Close</button>
6     </div>
7   </ion-header-bar>
8
9   <ion-content>
10    <div class="item item-body">
11      <b2>{{home.info.name}}</b2>
12      <p>Age: {{home.info.age}}</p>
13      <p>Gender: {{home.info.gender}}</p>
14      <p>Location: {{home.info.location}}</p>
15    </div>
16  </ion-content>
17 </ion-modal-view>
```

Рисунок 4.18 – Фрагмент коду info.html

В результаті цього коду з'являється модальне вікно, яке містить усю інформацію про користувача, а саме: ім'я, вік, стать та локацію.

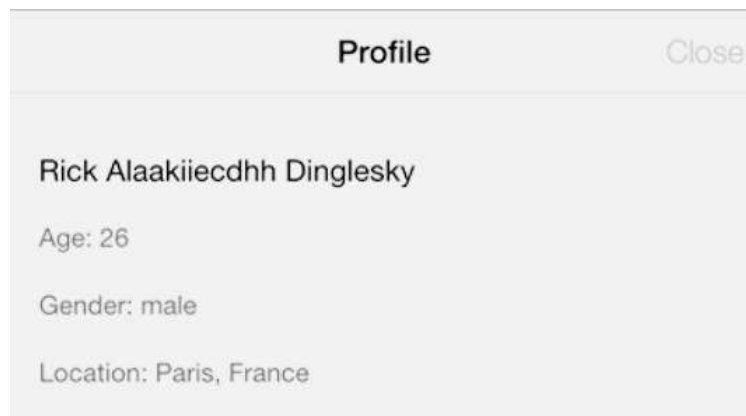


Рисунок 4.19 – Демонстрація роботи функції перегляду інформації

## Висновки до розділу 4

В четвертому розділі було проведено роботу з розробки front-end і back-end частин застосунку для онлайн знайомств. В результаті чого було закріплено навички роботи з мовою програмування JavaScript, його фреймворком AngularJS, а також створення і в подальшому впровадження в систему БД – Firebase. Було реалізовано весь запланований функціонал застосунку

## ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра було створено застосунок для онлайн-знайомств за рахунок розробки програмного забезпечення.

- для досягнення визначеної мети вирішено поставленні завдання:
- проаналізовано предметну область та застосунки-аналоги;
- визначено функціонал застосунку;
- створено необхідні UML-діаграми;
- розроблено дизайн та макет застосунку;
- розроблено front-end частину вебзастосунку на базі технології Angular.js.

Завдяки проведеному дослідженню предметної області визначено актуальність застосунку для онлайн-знайомств за рахунок розробки застосунку «LoveLink» та сферу застосування застосунку. За результатами аналізу застосунків-аналогів та визначенню їх переваг та недоліків сформовано вимоги до ПЗ, що розроблялося.

Для моделювання та проектування системи розроблено 4 види UMLдіаграм, задля найбільш зрозумілої демонстрації роботи застосунку. Відповідно до технічного завдання обрано найбільш придатні для розробки вебзастосунку технології, розроблено front-end та back-end частини ПЗ. Проведено тестування застосунку та усунуено знайдені помилки. Результатом проведеної роботи є вебзастосунок для онлайн-знайомств.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Diogo M., Bernardino J., Cabral B. Consistency models of NoSQL databases. Future Internet, 2019. 19p.
2. Jin B., Sahni S., Shevat A. Designing Web APIs: Building APIs That Developers Love. O'Reilly Media, 2018. 232 p.
3. Chhetri N., A Comparative Analysis of Node . js ( Server-Side JavaScript ). St. Cloud State University, 2016. 118p.
4. Khawas C., Shah P. Application of Firebase in Android App Development-A Study. International Journal of Computer Applications, 2018. 179p.
5. Hulliyah K., Shudhuashar M., Santoso W. Whatsapp Chatbot Implementation Using Node JS for a Da'wah Media Digitalization. IEEE, 2021. 320p.
6. Sheha A., Pushpanjali M. Angular Js. International Journal of Scientific & Engineering Research, 2016. 11 p.
7. Sihotang J., Richel Y., Pakpahan A. Membership Information System Using Node JS. Abstract Proceedings International Scholars Conference, 2019. 287 p.
8. Atzeni P., Bugiotti F., Cabibbo L., Torlone R. Data Modeling in the NoSQL World. Università Roma Tre, Italy, 2020. 35p.
9. BUCEA-MANEA-TONIS. Angular JS – The Newest Technology in Creating Web Applications. EndNote, 2016. 112 p.
10. Corbellini A., Cristian M., Alejandro Z., Daniela G., Schiaffino S. Persisting big-data: The NoSQL landscape, 2017. 23 p.
11. UML для бізнес моделювання. URL: <https://evergreens.com.ua/ru/articles/uml-diagrams.html> (дата звернення: 05.05.2023).
12. AngularJS Tutorial with practical coding exercises. URL: <https://www.guru99.com/angularjs-tutorial.html> (дата звернення: 29.05.2023).
13. Criss Griffith. Mobile App Development with Ionic, Revised Edition: Cross-Platform Apps with Ionic, Angular, and Cordova. O'Reilly Media, 2017. 292p.

14. I Code Academy. Angular JS for Beginners: Your Guide to Easily Learn Angular JS In 7 Days Paperback. WhiteFlower Publishing, 2020. 196p.
15. Laurence Moroney. The Definitive Guide to Firebase: Build Android Apps on Google's Mobile Platform. Apress, 2017. 288p.
16. Ionic Framework. URL: <https://ionicframework.com/> (дата звернення: 30.05.2023).
17. Ionic Framework: Pros and Cons Of Building Apps With Ionic. URL: <https://markovate.com/blog/ionic-framework-with-its-pros-and-cons/> (дата звернення: 26.04.2023)
18. AngularJS: Pros and Cons. URL: <https://data-flair.training/blogs/angularjs-advantages-and-disadvantages/> (дата звернення: 26.04.2023)
19. Greg Lim. Beginning Node.js Development. Independently published, 2019. 155p.
20. Node.js Tutorial. URL: <https://www.w3schools.com/nodejs/> (дата звернення: 01.06.2023)