

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: «РОЗРОБКА МЕСЕНДЖЕРУ «LICHAT» ДЛЯ ANDROID МОВОЮ
JAVA»

Виконав: студент 4 курсу, групи ПД-41

спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Виходцев М.М.

(прізвище та ініціали)

Керівник Трінтіна Н.А.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ–2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки - 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

А.В. Бондаренко

“ _____ ” _____ 2023 року

ЗАВДАННЯ
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

ВИХОДЦЕВ МИКИТА МИКОЛАЙОВИЧ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка месенджера «LyChat» для Android мовою Java»

Керівник роботи Трінтіна Наталія Альбертівна к.т.н, доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “24” лютого 2023 року No26.

2. Строк подання студентом роботи «1» червня 2023 року

3. Вхідні дані до роботи:

3.1. Науково-технічна література, пов'язана з месенджерами

3.2. Android

3.3. Java

3.4. Firebase

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

розробити).

- 4.1. Аналіз галузі месенджерів
- 4.2. Аналіз сучасних месенджерів
- 4.3. Розробка програми
- 4.4. Тестування системи

5. Перелік графічного матеріалу

- 5.1. Мета, об'єкт, предмет дослідження
- 5.2. Аналіз аналогів
- 5.3. Вимоги
- 5.4. Технічні засоби реалізації
- 5.5. Діаграма прецедентів
- 5.6. Діаграми послідовності
- 5.7. Висновки

6. Дата видачі завдання «25» лютого 2023 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	25.02.23 – 27.02.23	Виконано
2	Аналіз сучасних месенджерів	28.02.23 – 05.03.23	Виконано
3	Визначення вимог	06.03.23 – 10.03.23	Виконано
4	Аналіз засобів реалізації	10.03.23 – 15.03.23	Виконано
6	Проектування і розробка месенджера	16.03.23 – 20.04.23	Виконано
7	Тестування месенджера	21.04.23 – 29.04.23	Виконано
8	Вступ, висновки, реферат	30.04.23 – 05.05.23	Виконано
9	Розробка обов'язкових демонстраційних матеріалів	06.05.23 – 15.05.23	Виконано
10	Попередній захист роботи	22.05.2023	
11	Здача роботи	01.05.2023	

Студент _____ Виходцев М.М.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Трінтіна Н.А.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 43 с., 10 табл., 35 рис., 1 дод., 18 джерел.

JAVA, ANDROID, ANDROID STUDIO, ANDROID SDK, ANDROID JETPACK, FIREBASE, RXJAVA, CLOUD FIRESTORE, INSTANT MESSAGING, NOSQL DATABASE, FIREBASE AUTHENTICATION, FIREBASE STORAGE.

Об'єкт дослідження – процес комунікації в інтернеті.

Предмет дослідження – система миттєвих повідомлень.

Мета роботи – покращення якості спілкування в інтернеті за рахунок підвищення рівня безпеки та зручності при використанні месенджера.

Методи дослідження – методи створення мобільних додатків, методи аналізу та роботи з отриманими результатами, методи тестування мобільних додатків.

Наукова новизна: побудова системи миттєвих повідомлень з використанням хмарних технологій, забезпечення безпеки та зручності використання такого рішення.

В роботі було розглянуто та проаналізовано месенджери аналоги, їх переваги та недоліки. Були виявлені проблеми та вимоги до систем миттєвих повідомлень. Були досліджені методи розробки та засоби реалізації таких систем.

Галузь використання: особисте та бізнес спілкування.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Визначення месенджеру	9
1.2 Огляд існуючих месенджерів	11
1.2.1 WhatsApp	12
1.2.2 Viber	13
1.2.3 Telegram	13
1.2.4 Signal	14
1.4 Висновки аналізу	16
2 МОДЕЛЮВАННЯ СИСТЕМИ	17
2.1 Визначення вимог до системи	17
2.2 Огляд архітектури системи	19
2.3 Огляд засобів реалізації	20
2.3.1 Серверна частина	20
2.3.2 Клієнтський додаток	24
2.4. Огляд баз даних системи	27
2.5 Моделювання роботи системи	28
2.5.2 Діаграма послідовності реєстрації та автентифікації	28
2.5.3 Діаграма послідовності пошуку користувачів	29
2.5.4 Діаграма послідовності відображення чатів	30
2.5.5 Діаграма послідовності відправки повідомлення	31
2.5.6 Діаграма послідовності оновлення фотографії	32
3 ОГЛЯД ТА ТЕСТУВАННЯ РОБОТИ СИСТЕМИ	34
3.1 Перевірка роботи системи	34
3.1.1 Перевірка роботи реєстрації та автентифікація	34
3.1.2 Перевірка роботи пошуку	40
3.1.3 Перевірка роботи миттєвих повідомлень	42
3.1.4 Перевірка роботи відображення чатів	43
3.1.5 Перевірка роботи персоналізації профілю чатів	44
3.2 Тестування програми	47
ВИСНОВКИ	52
ПЕРЕЛІК ПОСИЛАНЬ	53

ВСТУП

Останні роки світ має тенденцію збільшення використання цифрових технологій у нашому повсякденному житті. У тому числі люди багато спілкуються через інтернет і потреба в такому спілкуванні тільки збільшується. Одним з варіантів спілкування через інтернет є месенджери. З ними, люди можуть спілкуватися зі своїми друзями, родиною та колегами по роботі будь-коли та в будь-якому місці в режимі реального часу. Наразі майже у кожної людини встановлено месенджер на її мобільному пристрої.

Об'єктом дослідження в бакалаврській роботі є саме процес комунікації в інтернеті, а предметом дослідження – системи миттєвих повідомлень, саме завдяки яким, процес комунікації в інтернеті і досягається.

Метою роботи є покращення якості спілкування в інтернеті за рахунок підвищення рівня безпеки та зручності при використанні месенджера.

Завданнями дослідження, по-перше, є аналіз предметної області систем миттєвих повідомлень, визначення принципу їх роботи та аналіз сучасних месенджерів, виявлення їх переваг та недоліків. Далі завдання полягає у визначенні вимог до месенджерів та огляд технічної бази, завдяки якій, можна створити систему миттєвих повідомлень. Змоделювати систему месенджера та розробити його, відносно вимог, які були визначені. І останнім завданням дослідження є перевірка роботи створеної системи та її тестування. Розроблений месенджер можна використовувати для особистого та бізнес спілкування

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Визначення месенджера

Месенджери або системи миттєвих повідомлень (Instant messaging) – це системи, що надає можливість спілкуватися в інтернеті, використовуючи миттєві текстові повідомлення. Під миттєвістю мається на увазі, що адресат отримує повідомлення одразу, як його відправить відправник. Більшість месенджерів також надає можливості обміну файлами, картинками, відео, а також здійснювати аудіо та відео виклики. Системи миттєвих повідомлень широко використовуються при особистому та бізнес спілкуванні.

На відміну від електронної пошти, спілкування в месенджерах проходить в реальному часі. Все спілкування відбувається в чатах, це спеціальна бесіда для двох або більше людей в яку не можуть інші люди. Чат між двома користувачами називають приватним чатом, між 3 та більше користувачами – груповим чатом. Користувачі можуть бачити мережевий статус співбесідника, зазвичай це онлайн або офлайн. Також відображається інформація про те, що співбесідник пише вам повідомлення. Ну а надісланні повідомлення миттєво отримуються співбесідником що він одразу може відповісти.

Більшість систем миттєвих повідомлень надають схожі функціональні можливості. Зазвичай надають наступні можливості:

- миттєві повідомлення;
- приватні та групові чати;
- мережевий статус користувачів;
- стікери та емоджі;
- пошук інших користувачів;
- оповіщення при отриманні повідомлення;
- передача файлів, картинок тощо;
- налаштування додатка та персоналізація профілю.

Для того, щоб почати користуватися месенджером, потрібно завантажити відповідний додаток на свій пристрій. Надалі потрібно створити персональний акаунт, зазвичай для цього використовується номер мобільного телефону, після чого вам дадуть можливість персоналізувати акаунт – ввести своє ім'я, прізвище, нікнейм та фотографію профілю. Процедура реєстрація відбувається тільки один раз, надалі використовується процедура автентифікації, коли користувач заходить на свій акаунт за допомогою мобільного телефону, який вказано на його акаунті. Після процедури реєстрації або автентифікації користувач входить на свій акаунт і отримує всі функції месенджера.

Системи миттєвих повідомлень існують вже достатньо давно, але стали дійсно популярні коли компанія Mirabilis випустила месенджер під назвою ICQ. Ця система стало дуже популярною в інтернеті та почали з'являтися схожі програмні застосунки, такі як AIM, Yahoo Messenger та Skype. Наразі ж месенджери стали неймовірно популярні, відповідно статистиці (рис 1.1) від Statista [3], найпопулярніший месенджер WhatsApp налічує понад 2 мільярди активних користувачів по всьому світу станом на січень 2023 року.

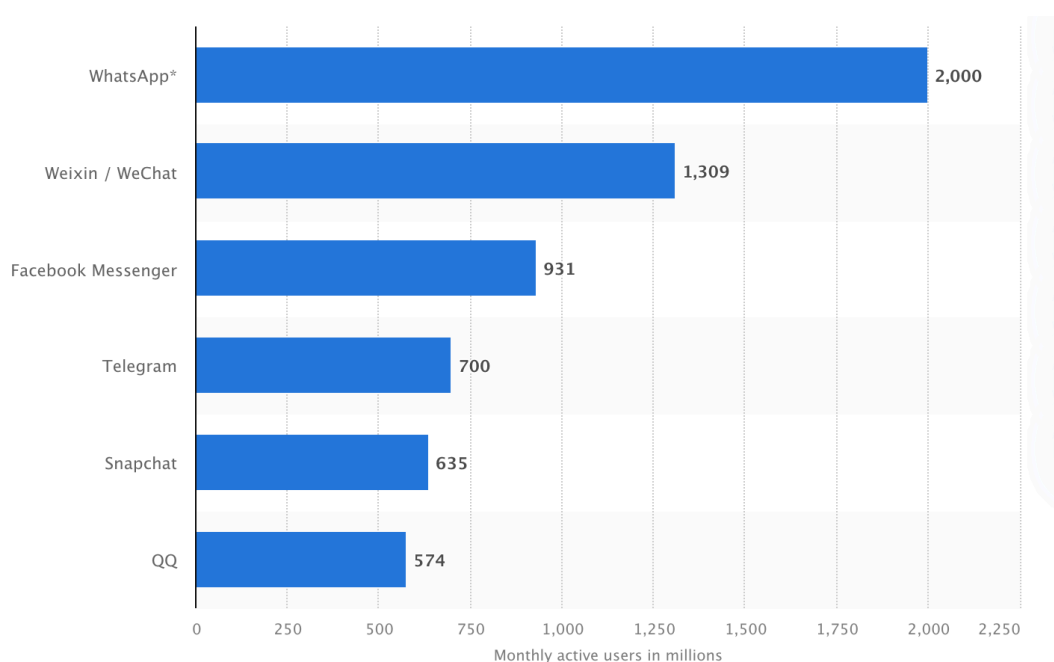


Рисунок 1.1 – Статистика по кількості активних користувачів месенджерів на місяць від Statista [3] станом на січень 2023 року

1.2 Огляд існуючих месенджерів

Як вже було зазначено, найпопулярнішим месенджером у світі є WhatsApp (рис 1.1). Другим за популярністю йде месенджер WeChat, який здебільшого використовується в азійсько-тихоокеанському регіоні та вчасності Китаю. Та і на третьому місці знаходиться система миттєвих повідомлень від Facebook – Messenger.

Щодо популярності месенджерів в Україні, спираючись на дані, які надає опитування Київського міжнародного інституту соціології [4], станом на вересень 2021 року, найпопулярнішим месенджером є Viber. На рисунку 1.2 наведено результати опитування.

Мобільний застосунок	%
Viber (Вайбер чи вібер)	73,6
WhatsApp (Вотсап)	25,3
Месенджер Фейсбуку	42,7
Телеграм	31,6
Твіттер	4,9
Сігнал	3,8
Скайп	11,3
Інстаграм	26,4
Тік Ток	9,1
Інше (ЗАПИШІТЬ)_____	2,6
Нічим з перерахованого	18,7
ВАЖКО СКАЗАТИ/ ВІДМОВА ВІДПОВІДАТИ	0,8

Рисунок 1.2 – Звіт Київського міжнародного інституту соціології, щодо найбільш популярних мобільних додатків [4]

Для детального аналізу було обрано наступні 4 системи миттєвих повідомлень:

- WhatsApp, обрано, бо є найпопулярнішим месенджером у світі;
- Viber, обрано бо, є найпопулярнішим месенджером в Україні;
- Telegram, обрано, бо має великий функціонал;

- Signal, обрано, бо має високий рівень безпеки та конфіденційності.

1.2.1 WhatsApp

WhatsApp (WhatsApp Messenger) [5] – безплатний, багатоплатформовий месенджер. Є найпопулярнішим месенджером у всьому світі й нараховує 2 мільярди активних користувачів. Месенджер створено компанією WhatsApp Inc, що, на цей час, належить Facebook.

Дозволяє користувачам обмінюватися текстовими та аудіо повідомленнями, робити аудіо та відео дзвінки. Надає можливість ділитися картинками, документами, локаціями тощо. З мінусів немає функції редагування надісланих повідомлень, немає само видалення повідомлень та має закритий вихідний код.

WhatsApp використовує кастомізований стандарт Extensible Messaging and Presence Protocol (XMPP), який після встановлення створює акаунт користувача, використовуючи номер телефона як ім'я користувача.

На рисунку 1.3 наведено екранні форми месенджера WhatsApp.

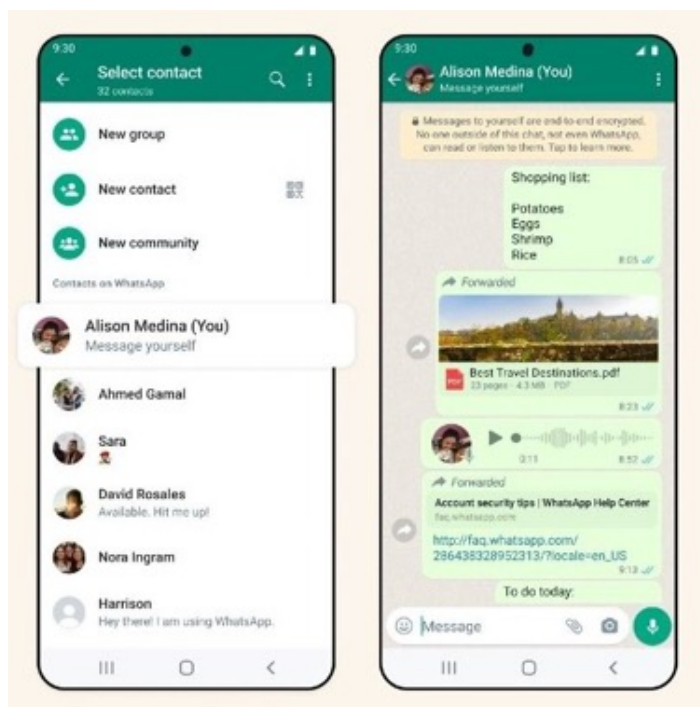


Рисунок 1.3 – Приклад екранних форм месенджера WhatsApp

1.2.2 Viber

Viber, або Rakuten Viber [6] – безплатний месенджер, що належить японській компанії Rakuten. Має додатки на всі популярні платформи: Android, iOS, Windows, macOS та Linux. Має закритий вихідний код.

Користувачі реєструються та ідентифікуються за номером мобільного телефону. Дозволяє користувачам обмінюватися мультимедійними даними, такими як зображення та відеозаписи, а також надає платну послугу міжнародних стаціонарних та мобільних дзвінків під назвою Viber Out.

Одним з найголовніших недоліків месенджера це наявність великої кількості рекламних повідомлень.

На рисунку 1.4 наведено екранні форми месенджера Viber.

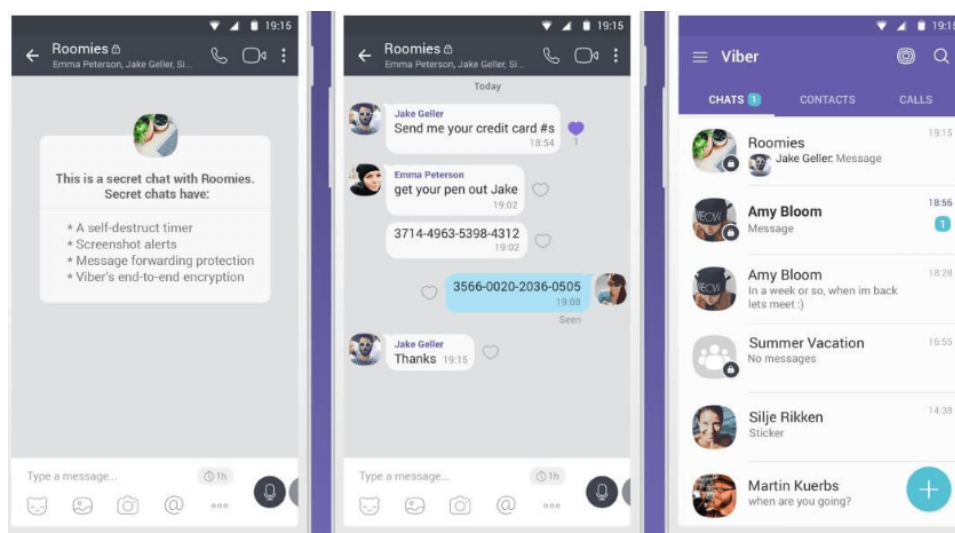


Рисунок 1.4 – Приклад екранних форм месенджера Viber

1.2.3 Telegram

Telegram [7] – відомий багатоплатформовий месенджер. Має користувацьку базу і має багато функцій. Користувачі можуть відправляти текстові та голосові повідомлення, робити аудіо та відео дзвінки. Надає можливість ділитися документами, картинками, локаціями, контактами, аудіофайлами тощо. Також користувачі можуть підписуватися на різні групи.

Telegram має відкритий вихідний код для клієнтських додатків який

ліцензовано за ліцензією GPL-2.0, окрім сервера, що є власністю компанії.

Telegram надає додаткові чати які мають end-to-end шифрування – «секретні чати». Трафік між клієнтським додатком та сервером зашифровані, щоб треті сторони в мережі, такі як провайдери, не могли отримати доступ до даних. Має централізовану архітектуру.

На рисунку 1.5 наведено екранні форми месенджера Telegram.



Рисунок 1.5 – Приклад екранних форм месенджера Telegram

1.2.4 Signal

Signal [8] – централізований, безпечний месенджер, який фокусується на забезпеченні максимальної конфіденційності. Має клієнтські додатки на всі популярні платформи: Android, iOS, Windows, macOS та Linux. Розроблений некомерційною організацією Signal Foundation.

Месенджер включає функції посилання текстових та голосових повідомлень, картинок, відео та інших файлів. З недоліків можна виділити неможливість редагування надісланого повідомлення.

Програмне забезпечення Signal є безплатний із відкритим кодом. Його мобільні додатки випущені за ліцензією GPL-3.0, а настільні додатки та сервери випущені за ліцензією AGPL-3.0.

Signal використовує номери мобільного телефону як ідентифікатор користувача. Месенджер використовує end-to-end шифрування для всіх

комунікацій.

На рисунку 1.6 наведено екранні форми месенджера Signal.



Рисунок 1.6 – Приклад екранних форм месенджера Signal

На таблиці 1.1 наведено результати аналізу розглянутих месенджерів.

Таблиця 1.1 – Результати аналізу месенджерів

	Telegram	WhatsApp	Viber	Signal
Реєстрація за номером телефону	+	+	+	+
Приватні чати	+	+	+	+
Групові чати	+	+	+	+
Двухфакторна авторизація	+	+	-	+
Шифрування за замовченням	-	+	+/-	+
Повідомлення зберігаються на сервері	+	+	-	-
Редагування відправленого повідомлення	+	-	+	-
Має відкритий вихідний код	+/-	-	-	+
Має централізовану архітектуру	+	+	+	+/-

1.4 Висновки аналізу

Після аналізу поняття систем миттєвих повідомлень та розглядання сучасних месенджерів можна виділити переваги й недоліки месенджерів, їх наведено в таблиці 1.2.

Таблиця 1.2 – Переваги та недоліки сучасних месенджерів

Переваги	Недоліки
Миттєва передача повідомлень	Риск втрати персональної інформації
Надає можливості передачі файлів	Можливі рекламні повідомлення
Розмови в реальному часі	Будь-хто може писати вам повідомлення
Використовується для особистого та бізнес спілкування	
Використання на відстані	
Легкі в використанні	

2 МОДЕЛЮВАННЯ СИСТЕМИ

2.1 Визначення вимог до системи

Після аналізу галузі систем миттєвих повідомлень, розглядання та порівняння сучасних месенджерів, можна висунути мінімальний функціонал до системи, що розробляється:

- реєстрація користувача за номером телефону;
- автентифікація користувача за номером телефону;
- персоналізація профілю, можливість встановлення імені, прізвища, нікнейму та фотографії профілю;
- пошук інших користувачів за їхнім номером телефону, ім'ям, прізвищем, нікнеймом або номером телефону;
- відображення чатів користувача;
- відправка повідомлень іншим користувачам в чатах;
- мережевий статус користувача (онлайн або офлайн);

На рисунку 2.1 наведено mind map до функціонала системи, що розробляється.

Mind map – діаграма, що використовується для візуального організації інформації в ієрархію. Вона починається з головної ідеї проекту, після чого ця тема розбивається на підтеми, які відображають функціональність додатка. Mind map допомагає організувати ідеї, візуалізуючи їх, щоб бачити повну картину.

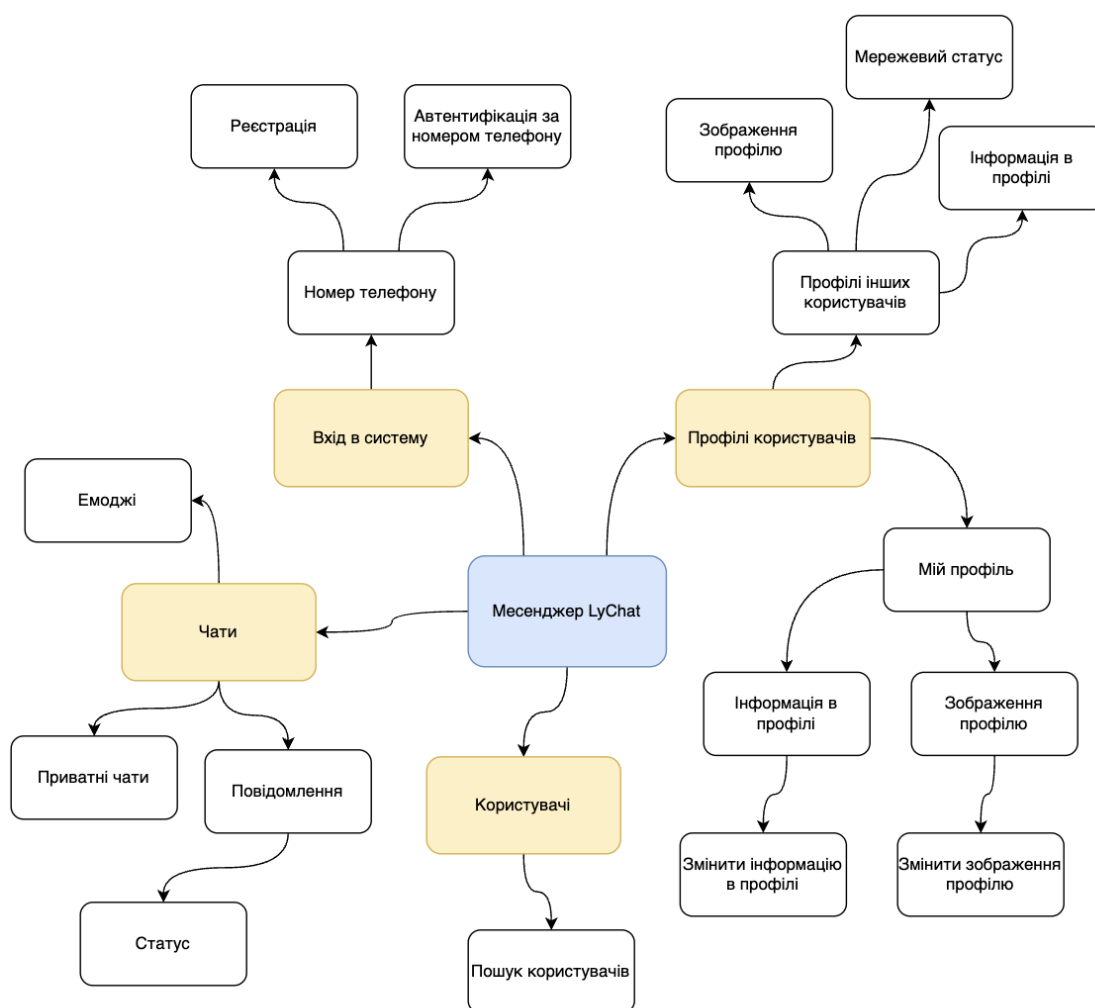


Рисунок 2.1 – Mind map функціоналу месенджера LyChat

Після визначення вимог до програмного забезпечення та його функціоналу, побудуємо діаграму прецедентів додатка (use case diagram).

Діаграма прецедентів – графічне зображення можливих взаємодій користувача (акторів) з системою.

Актор – сутності, що взаємодіють з системою. Актором може бути людина, організація або зовнішні системи, які взаємодіють програмою. Вони мають бути зовнішніми об'єктами що споживають або виробляють інформацію.

На рисунку 2.2 наведено діаграму прецедентів.

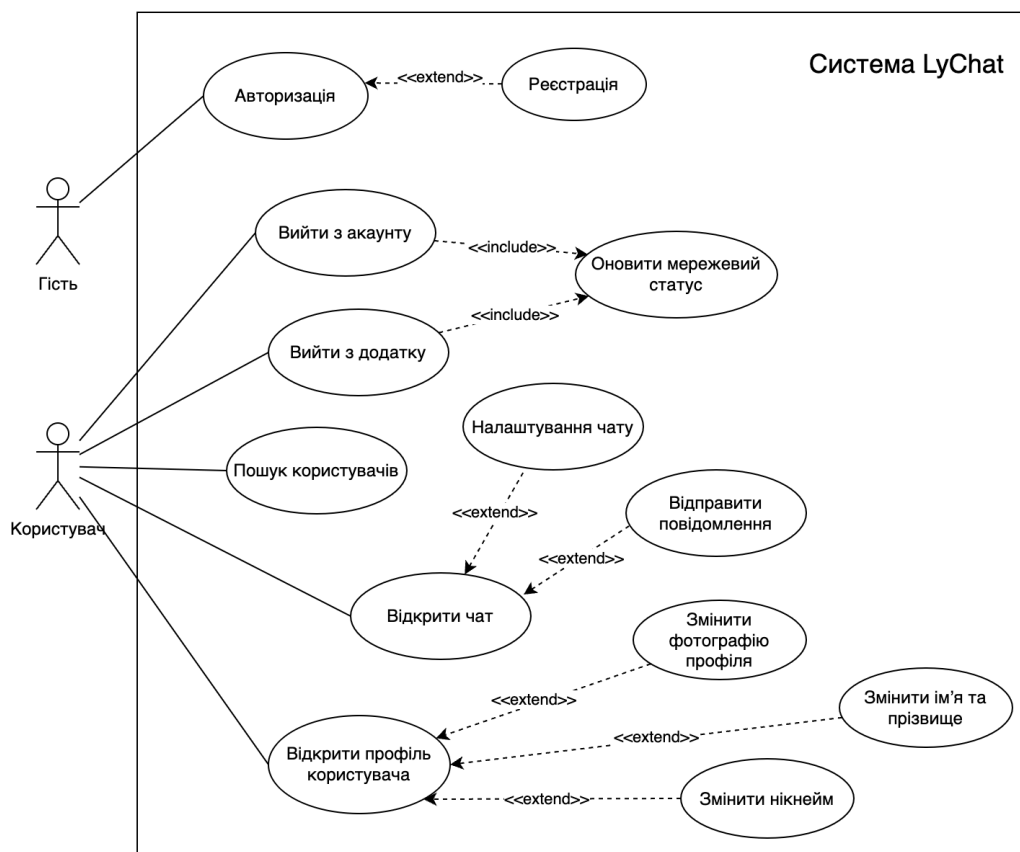


Рисунок 2.2 – Діаграма прецедентів месенджера LyChat

2.2 Огляд архітектури системи

Месенджер буде мати client-server архітектуру (рис 2.3). Це означає, що буда два ключових компоненти: клієнтський додаток та сервер.

Клієнтський додаток – це додаток, що запускається на телефоні користувача. Він включає інтерфейси, з якими буде взаємодіяти користувач: інтерфейс чату, інтерфейс налаштувань. Цей додаток відповідальний за взаємодію з користувачем: відправка та отримання повідомлень, відображення чатів, пошук інших користувачів.

Сервер – програмне забезпечення, що відповідає за бізнес-логіку, зберігання даних та взаємозв'язок з клієнтськими додатками. Він відповідальний за отримання відправлення та зберігання повідомлень, за роботу реєстрації та автентифікації користувачів та за управління акаунтами користувачів.

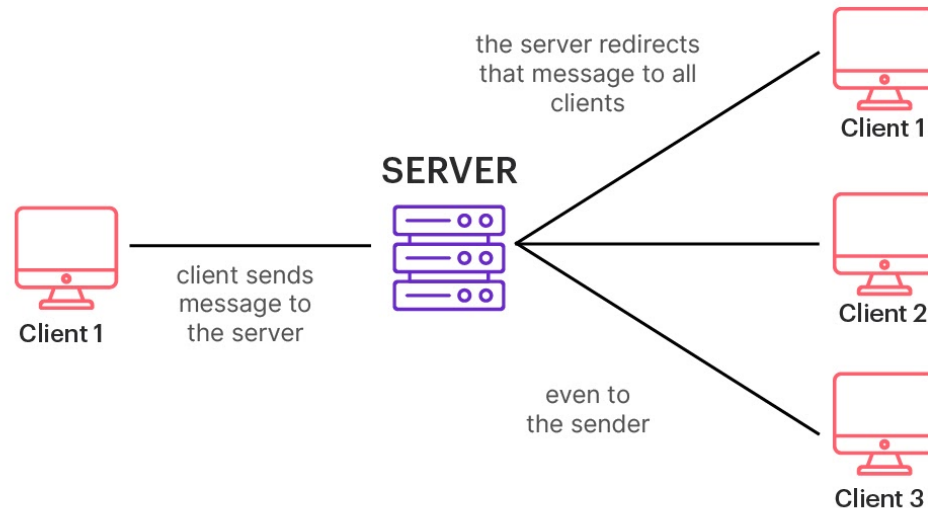


Рисунок 2.3 – Приклад client-server архітектури

2.3 Огляд засобів реалізації

Оскільки месенджер реалізує client-server архітектуру, треба створити сервер та клієнтський додаток.

2.3.1 Серверна частина

Серверною частиною додатка було обрано backend-as-a-service (BaaS) від Google – Firebase. Firebase це платформа для розробки додатків, що допомагає розробникам створювати, покращувати та збільшувати додатки (рис 2.4).

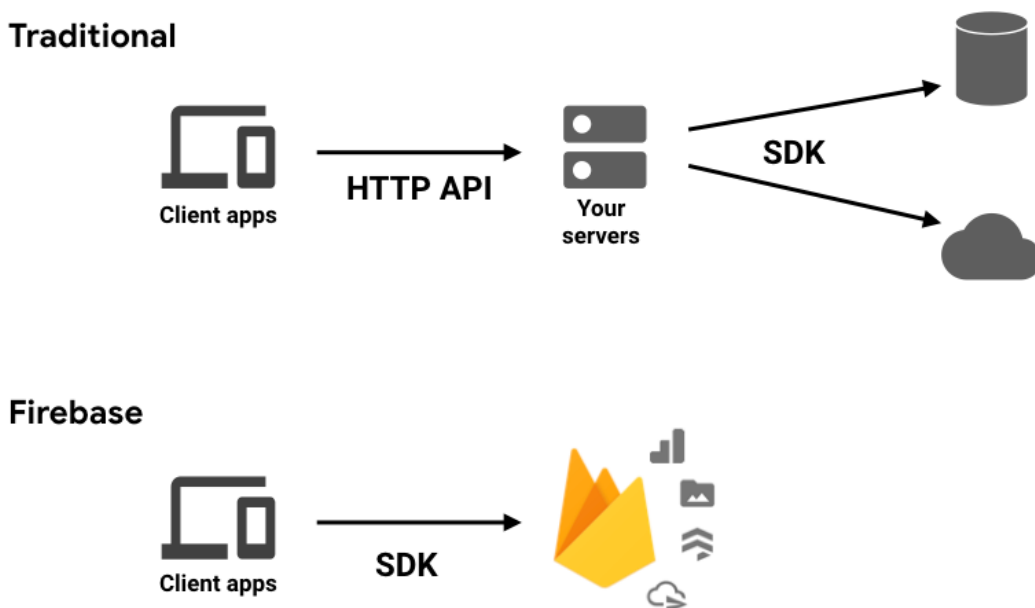


Рисунок 2.4 – Порівняння Firebase з традиційним сервером

Firebase має великий стек технологій, завдяки яким Firebase можна використовувати замість звичного серверу.

Firebase Authentication – сервіс що надає набір засобів розробки для автентифікації користувачів. Він підтримує автентифікацію за паролем, номером телефону та за популярними сервісами ідентифікації, таких як Facebook, Twitter, Google тощо. Використовує галузеві стандарти OpenID Connect та OAuth 2.0. Сервіс буде використовуватися в месенджері для реєстрації та автентифікації користувачів за номером мобільного телефону.

Cloud Firestore – гнучка, NoSQL, хмарна база даних для зберігання синхронізованих даних. Головною особливістю цієї бази даних є те, що вона надає «Realtime» оновленні дані, при їхній змінні в базі даних, тобто синхронізуючи інформацію між підключеними пристроями. Завдяки цьому, додатки завжди відображають актуальну інформацію.

Відповідно до моделі даних Cloud Firestore NoSQL, вся інформація зберігається в документах, що знаходяться в колекціях (рис 2.5). Визначення колекцій та документів:

- колекції це найвищий шар, використовується для зберігання документів. Завдяки йому можна організувати свої дані та писати запити.
- документи використовуються для зберігання даних в полях. Вони підтримують велику кількість різних форматів даних, від простих строк та цифр до складніших вкладених об'єктів.

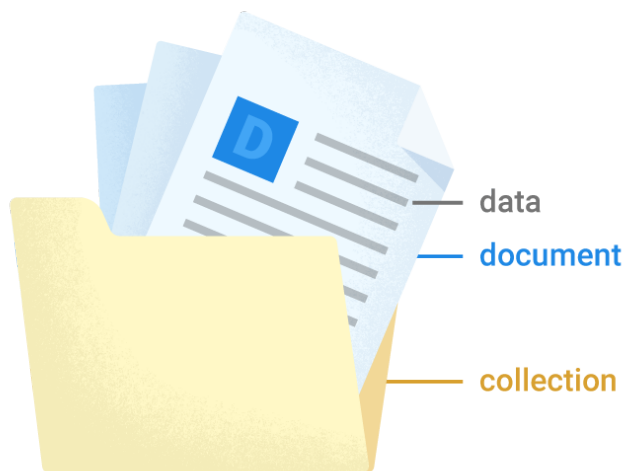


Рисунок 2.5 – Графічний приклад моделі даних Cloud Firestore

Також є можливість створення вкладеної колекції (subcollection) в документі та будувати ієрархічні структури даних.

Cloud Storage – сервіс, що надає сховище для зберігання файлів. Завантаження та вивантаження даних виконується і є надійними незалежно від якості інтернет з'єднання. Cloud Storage інтегрується з Firebase Authentication, що підвищує рівень безпеки збереженої інформації. Сховище буде використано для збереження фото, відео та інших файлів користувачів

Для забезпечення безпеки збережених даних в Cloud Firestore та Cloud Storage, Firebase надає функціонал Security Rules. Вони визначають, хто може читати та записувати інформацію в бази даних, як ці дані структуровані і які індекси існують. Ці правила завжди працюють та кожен запис або отримання даних буде виконано тільки за умови, що правила це дозволяють.

Сервіси Firebase шифрують дані, під час їх передачі, використовуючи HTTPS і логічно відокремлюючи дані користувачів. Також сервіси Cloud Firestore, Cloud Storage та Firebase Authentication шифрують збережені в них дані.

Також сервіси Firebase сертифіковано відповідно до стандартів безпеки та конфіденційності. Сервіси, що використовуються в месенджері: Cloud Firestore, Cloud Storage та Firebase Authentication, мають наступні сертифікати:

- ISO 27001;
- ISO 27017;
- ISO 27018;
- SOC 1;
- SOC 2;
- SOC 3

Плюси та мінуси використання Firebase, як серверного рішення для месенджера наведено в табл. 2.1.

Таблиця 2.1 – Плюси та мінуси використання Firebase

Плюси	Мінуси
Наявність безплатного плану	Доступний не у всьому світі
Прискорення розробки додатка	Обмежене використання в безплатному плані
Підтримка від Google	
Високий рівень безпеки	
Можливості бази даних	
Простота інтеграції та використання	
Можливість створення без серверних платформ	

2.3.2 Клієнтський додаток

Клієнтський додаток LyChat буде розроблятися на мобільні телефони на платформі Android. Мовою програмування буде Java, що є об'єктно-орієнтованою мовою програмування високого рівня, з використанням Android SDK в IDE Android Studio.

Android SDK – це набір інструментів та бібліотек для розробки програмного забезпечення, які призначені для створення Android додатків. Android SDK можна використовувати в будь-якому IDE, але найкраще використовувати Android Studio, в якому за замовченням встановлено Android SDK, яке оптимізовано для роботи в цьому IDE. В Android SDK знаходяться наступні компоненти:

- Android SDK Tools, що містить набір інструментів для розробки, налагодження (debugging), тестування та інших утиліт для створення додатків на Android;
- SDK Build Tools – інструменти, що використовують для побудови актуальних бінарних файлів до Android додатку. Головна їх функція це побудова, запуск та тест додатків на Android;
- SDK Platform tools. Їх використовують для підтримки нових версій Android та їх особливостей. Цей компонент обов'язковий для розробки додатків на Android. Воно містить:
 - 1) Android Debug Bridge (adb) – це командний рядок, який дозволяє комунікувати з пристроєм. Завдяки команді adb є можливість виконувати дії з пристроєм, такі як: завантаження та налагодження додатків;
 - 2) fastboot – дозволяє прошити пристрій новим образом системи
 - 3) systrace – інструмент, що допомагає збирати та перевіряти інформацію про час з всіх робочих процесах на пристрої на системному рівні;
- SDK Platform – для кожної версії Android є одна доступна SDK Platform. Вони пронумеровані за номером версії Android та версією API. Перед

початком розробки, треба обрати версію SDK Platform. Кожні версії розрізняються, новіші підтримують більше можливостей під час розробки, але вони не будуть сумісні з старими пристроями.

– Android Emulator – емулятор, що симулює пристрій на платформі Android на комп'ютері. Завдяки йому, розробники можуть запускати свій додаток на різних пристроях та різних версіях Android прямо на комп'ютері та тестувати його. На ньому можна запустити різні Android телефони, планшети, розумні часи та телевізори. Емулятор має багато можливостей справжнього Android пристрою:

- 1) симульовані телефонні дзвінки та текстові повідомлення;
- 2) симулювання різної швидкості мережі;
- 3) симулювання апаратних датчиків, таких як обертання;
- 4) змінювання геолокацію на пристрої;
- 5) доступ до Google Play тощо;

Також для полегшення розробки буде використовуватися Android Jetpack, що є колекцією програмних компонентів, які допомагають в розробці додатків. Вони допомагають слідувати найкращим практикам (best practices) та писати шаблонний код під час розробки.

Android Jetpack розв'язує наступні проблеми, які виникають під час розробки програмного забезпечення на Android:

- управління життєвим циклом Activity;
- виживання після змін конфігурації;
- запобігання витоку пам'яті;

А використовуватися будуть наступні компоненти:

- Data Binding – зв'язування елементів інтерфейсу до джерел даних додатку. Допомагає легше отримувати елементи з інтерфейсу використовуючи менше коду.
- Lifecycles – допомагає керувати життєвим циклом Activity в додатку;

– ViewModel – допомагає керувати даними, що відображаються на інтерфейсі, з урахуванням життєвого циклу Activity;

В клієнтському додатку буде відбуватися дуже багато подій одночасно: відправка та отримання інформації з сервера, її обробка та відображення на інтерфейсі користувача. Саме тому потрібне рішення для роботи з асинхронним програмуванням, і буде використано бібліотеку RxJava. Ця бібліотека для створення асинхронних програм які працюють, на основі подій.

RxJava розширює патерн програмування Observer (рис 2.6) для підтримки потоків даних (streams) та додання операторів. Завдяки цим рішенням, більше не треба працювати з низькорівневою багатопоточністю, синхронізацією, thread-safety, структурами даних для багатопоточного програмування та вводом та виводом, що не блокують додаток.

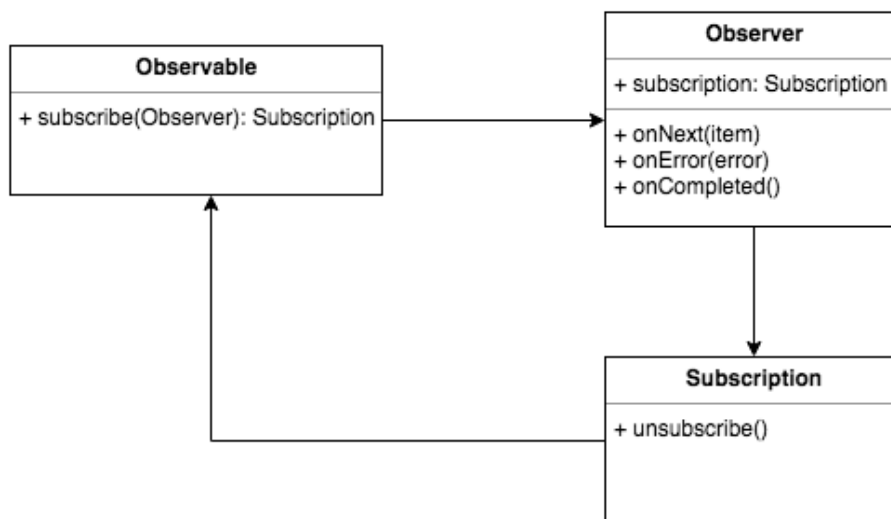


Рисунок 2.6 – UML діаграма класів патерну Observer

Для реалізації користувацького інтерфейсу в Android використовується XML (Extensible Markup Language). XML це легка мова розмітки, яка не ускладнює верстку. В ній присутні тільки теги, які потрібно тільки написати.

Користувацький інтерфейс для Android додатків будується як ієрархія макетів(layouts) та віджетів(widgets). Макети це контейнери, які визначають, як віджети будуть розташовані на екрані. При розробці буде використовуватися макет ConstraintLayout. А віджети це спеціальні об'єкти, такі як кнопки, поля тощо.

2.4. Огляд баз даних системи

В проєкті використовується хмарна NoSQL база даних від Firebase – Cloud Firestore. В ній немає таблиць, тільки колекції та документи, де кожен документ це JSON-подібна структура. Кожен документ в колекції може бути не схожим на інших. І також документи можуть містити інші вкладені колекції.

Перша колекція – users, в якій зберігаються документи користувачів, один документ на одного користувача. В документі зберігаються:

- ім'я;
- прізвище;
- нікнейм;
- мобільний номер користувача;
- id користувача;
- мережевий статус;
- посилання на фотографію профілю

Кожне поле в документах це string, тобто строкове значення. Назви документів відповідають до id користувача, який зберігається в цьому документі.

Друга колекція – chats, в ній зберігаються документи, з інформацією про чати. Один документ на один чат. Документ в собі містить наступну інформацію:

- id чату;
- останнє повідомлення в чаті;
- id першого користувача;
- id другого користувача

Всі поля в документах – string. Назви документів відповідають до id чату, який зберігається в цьому документі. Окрім того, в кожному документі є вкладена колекція (subcollection) – messages. В цих вкладених колекціях зберігаються документи з інформацією про повідомлення, один документ на одне повідомлення. Ці документи містять в собі наступну інформацію:

- id користувача, що відправив повідомлення;

- статус повідомлення;
- текст повідомлення;
- дата і час відправки

Поле, де зберігається дата і час відправки має тип даних `timestamp`, всі інші – `string`. Назви документів випадково генеруються самим `Firestore`.

2.5 Моделювання роботи системи

2.5.2 Діаграма послідовності реєстрації та автентифікації

В додатку, для спрощення навігації, процедури автентифікації та реєстрації об'єднанні. Для того, щоб почати, користувач повинен ввести свій номер мобільного телефону та верифікувати його. Після цього, системи дивитися, чи існує користувач з таким номером телефона, якщо існує, то відбувається вхід в акаунт, якщо ні, то користувач потрапляє на екран реєстрації. Діаграму наведено на рисунку 2.7, детальний опис кроків:

- 1) користувач вводить номера мобільного телефону у відповідне поле;
- 2) додаток відправляє номер телефона до `Firebase Authentication`;
- 3) `Firebase Authentication` відправляє смс на номер телефона з кодом підтвердження;
- 4) користувач вводить код підтвердження у поле;
- 5) додаток формує спеціальний `PhoneAuthCredential` та відправляє його до `Firebase Authentication`;
 - А) якщо було неправильно введено код підтвердження, то `Firebase Authentication` повертає помилку, яка відображається користувачу;
 - Б) якщо код підтвердження правильний і номер телефона користувача вже зареєстрований, відбувається вхід в акаунт;
 - В) якщо код підтвердження правильний і номер користувача НЕ зареєстрований, відбувається перехід на екран реєстрації;
 - а) користувач вводить дані у формі реєстрації;

- б) додаток відправляє форму до Cloud Firestore;
- в) Cloud Firestore зберігає нового користувача;
- г) користувач входить у свій новий акаунт;

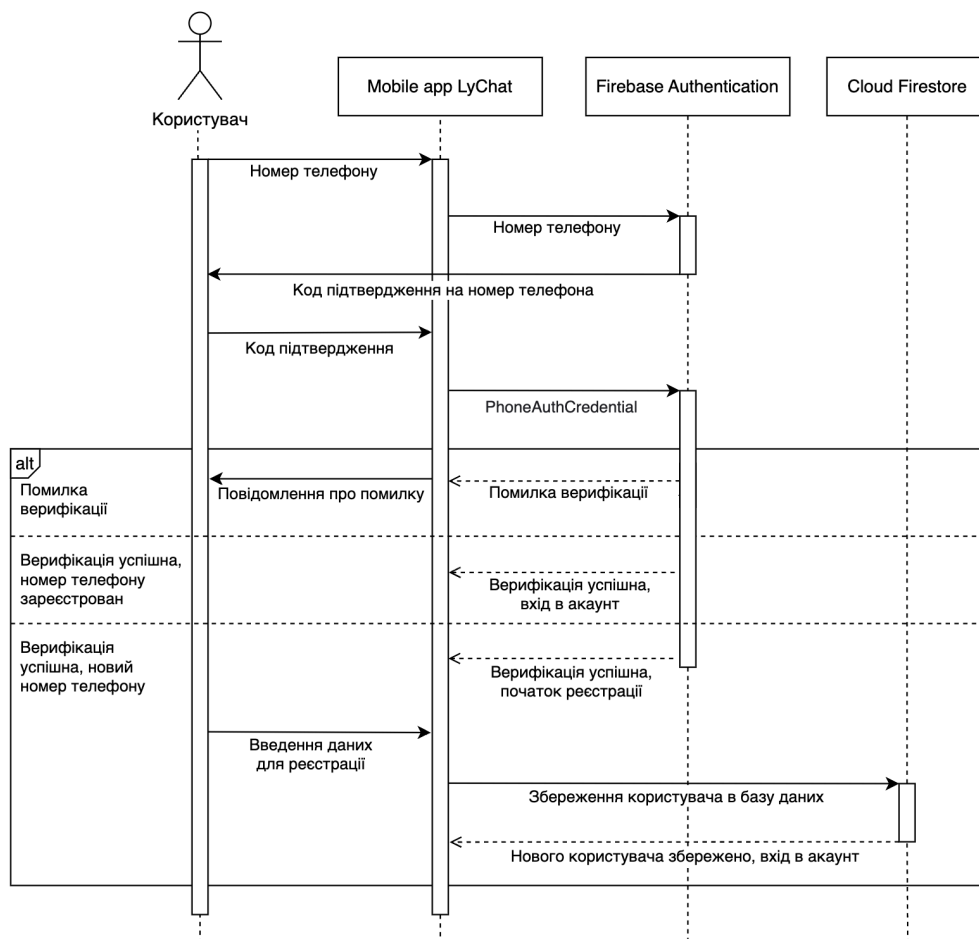


Рисунок 2.7 – UML діаграма послідовності автентифікації та реєстрації

2.5.3 Діаграма послідовності пошуку користувачів

Для того, щоб почати спілкування, спочатку потрібно знайти іншого користувача. Це робиться через функціонал пошуку. Пошук шукає користувачів в базі даних по наступним полям:

- ім'я;
- прізвище;
- нікнейм;
- номер мобільного телефону;

Cloud Firestore повертає список користувачів, що були знайдені, після чого список відображається в додатку. Діаграма зображена на рисунку 2.8.

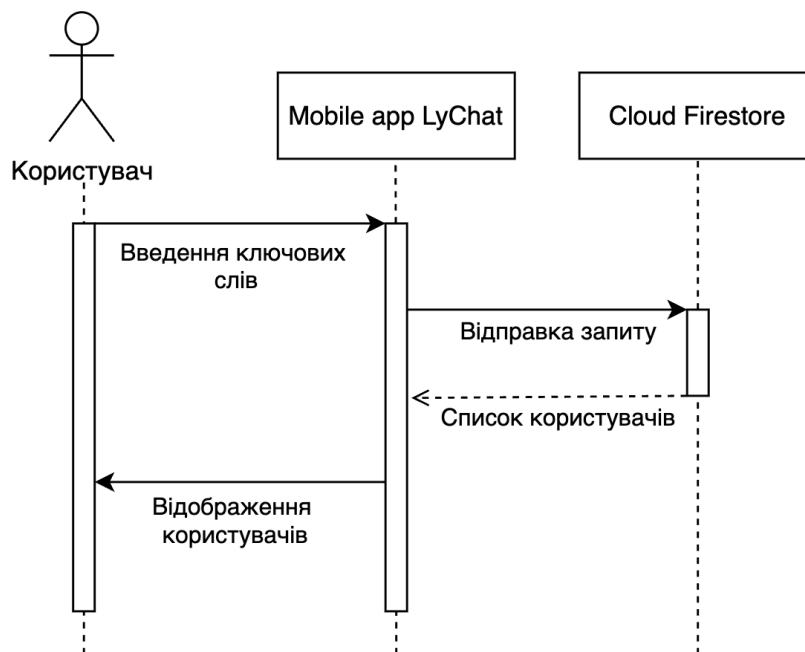


Рисунок 2.8 – UML діаграма послідовності пошуку користувачів

2.5.4 Діаграма послідовності відображення чатів

Коли користувач переходить на головну сторінку застосунку. Для того, щоб відобразити чати, додатку відправляє два запити до бази даних – перший на отримання списку чатів і другий для отримання співбесідників з цих чатів. Діаграму наведено на рисунку 2.9.

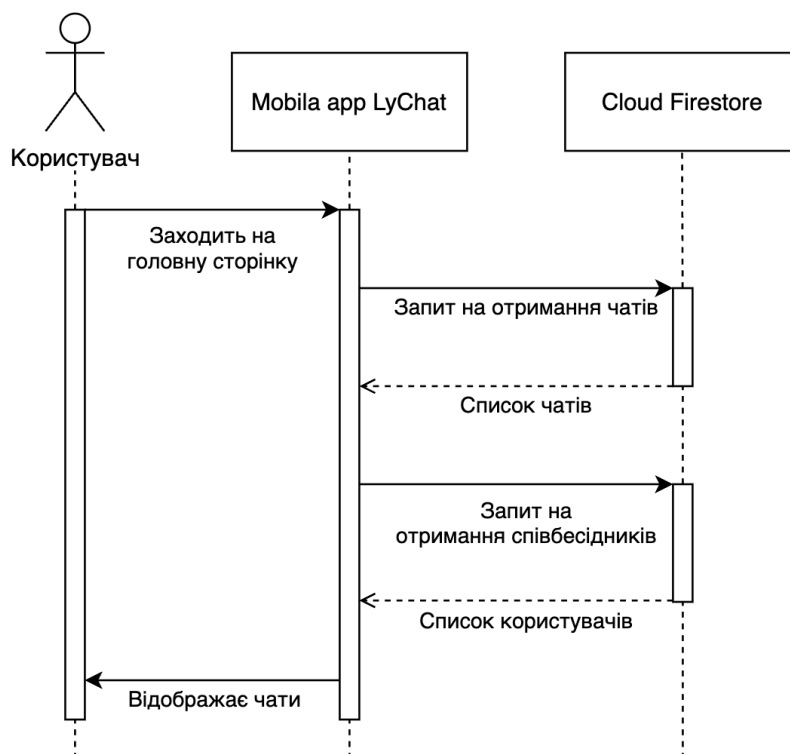


Рисунок 2.9 – UML діаграма послідовності відображення чатів

2.5.5 Діаграма послідовності відправки повідомлення

Коли адресант відправляє повідомлення, додаток відображає для нього це повідомлення як «надіслане» і відправляє модель цього повідомлення в базу даних, яка, своєю чергою повертає копію цього повідомлення на додатки адресанта та адресату. У адресата оновлюється чат та з'являється нове повідомлення, а в адресанта повідомлення оновлюється і стає «отриманим». Діаграму зображено на рисунку 2.10.

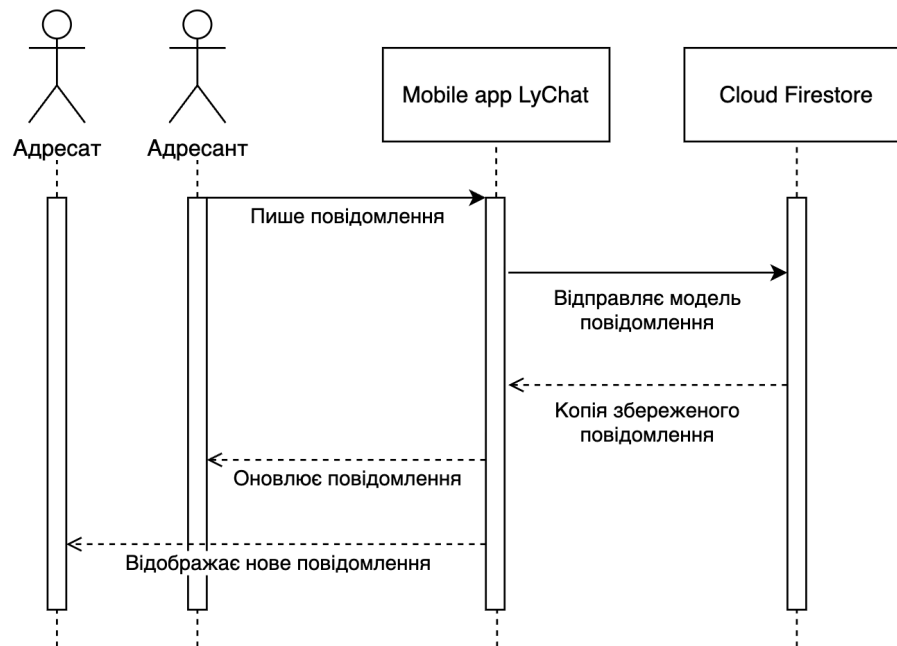


Рисунок 2.10 – UML діаграма послідовності відправки повідомлення

2.5.6 Діаграма послідовності оновлення фотографії

У користувача є можливість персоналізувати профіль і саме змінювати фотографію профілю. Для цього користувач обирає фотографію і звантажує її в додаток. Своєю чергою, додаток відправляє її в хмарне сховище Firebase Storage, де вона зберігається. Після чого Firebase Storage повертає посилання на цю фотографію, яка надалі зберігається у документі користувача в Cloud Firestore. Діаграму наведено на рисунку 2.11.

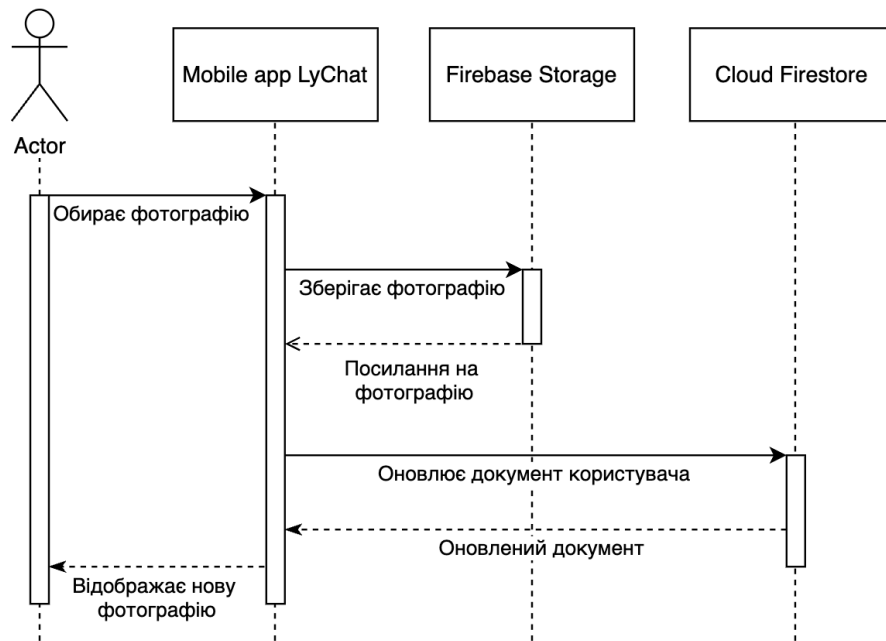


Рисунок 2.11 – UML діаграма послідовності оновлення фотографії

3 ОГЛЯД ТА ТЕСТУВАННЯ РОБОТИ СИСТЕМИ

3.1 Перевірка робота системи

Систему миттєвих повідомлень LyChat, яку було реалізовано, треба перевірити на відповідність вимогам, роботоздатність та провести тестування.

3.1.1 Перевірка роботи реєстрації та автентифікація

Екран вводу номера телефону, це перше, що бачить користувач після завантаження месенджера (рис 3.1). Спочатку користувачу треба підібрати код своєї країни, для цього в інтерфейсі передбачено спеціальний елемент – вибирач коду країни, який можна побачити зліва від поля вводу телефонного номера на рисунку 3.1.

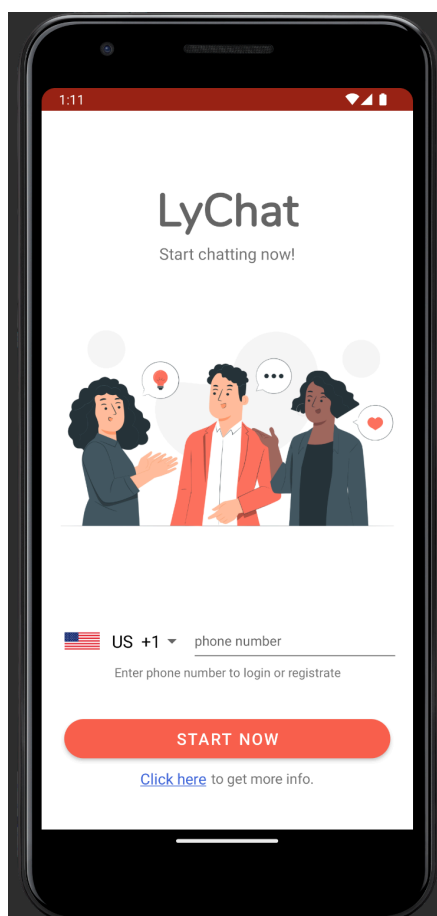


Рисунок 3.1 – Екран вводу номера телефону месенджера LyChat

Примітка. На всіх ілюстраціях додатка, месенджер LyChat запущено на емуляторі в IDE Android Studio.

Додаток автоматично визначає код країни користувача, використовуючи наступні методи:

- визначення країни вставленої SIM карти;
- визначення країни з інформації про підключену мережу;
- визначення країни з інформації про місцевість пристрою.

Методи їдуть зверху вниз за перевагою використання додатком. Якщо код країни користувача обрано неправильно, користувач може сам його обрати. Для цього потрібно один раз натиснути на елемент вибору країни. Після натискання відкривається вікно вибору країни (рис 3.2).

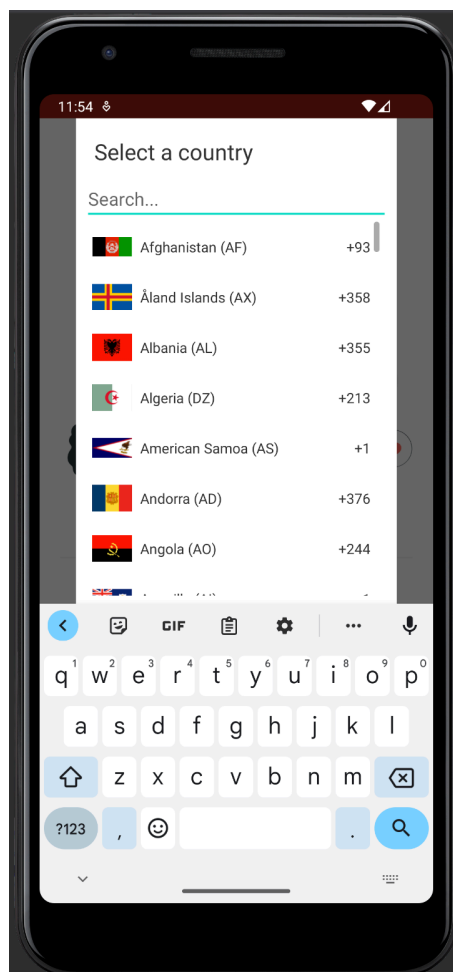


Рисунок 3.2 – Вікно вибору країни

Вибираємо у вікні Україну та вводимо номер телефона у поле вводу (рис 3.3) та тиснемо на кнопку «Почати зараз».

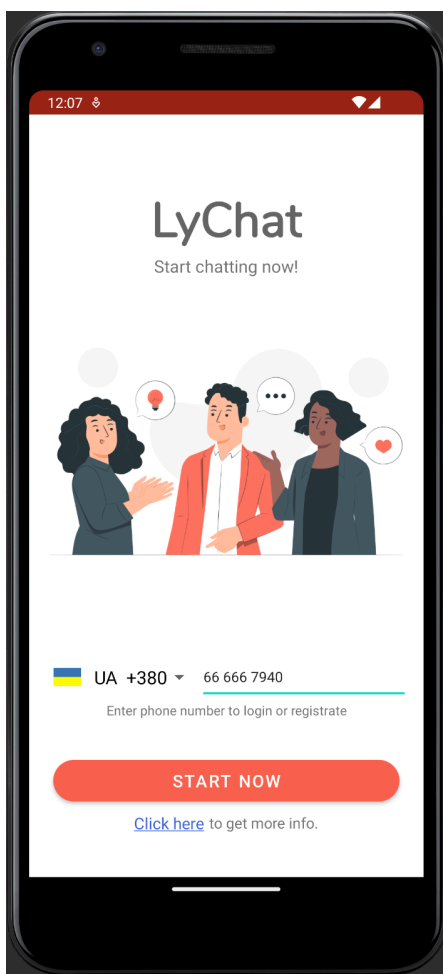


Рисунок 3.3 – Екран вводу номера телефона з заповненими полями

Після натискання на кнопку відривається новий екран – верифікація номера мобільного телефону. На номер телефона, що було вказано, приходять СМС з верифікаційним кодом (рис 3.4). Вводимо верифікаційний код у відповідне поле на екрані верифікації (рис 3.5).

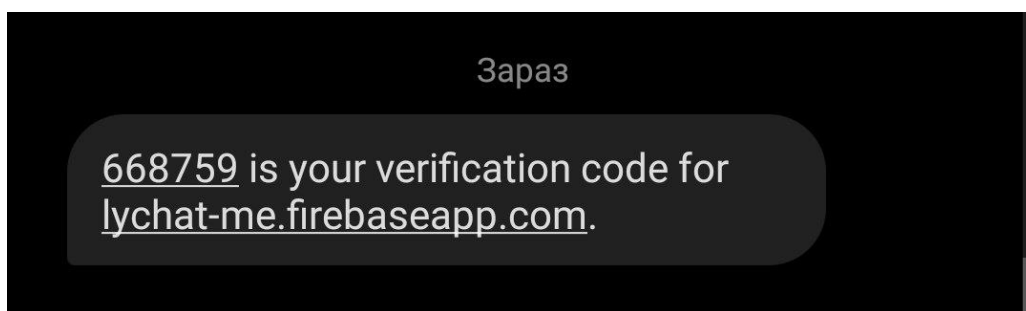


Рисунок 3.4 – СМС з верифікаційним кодом

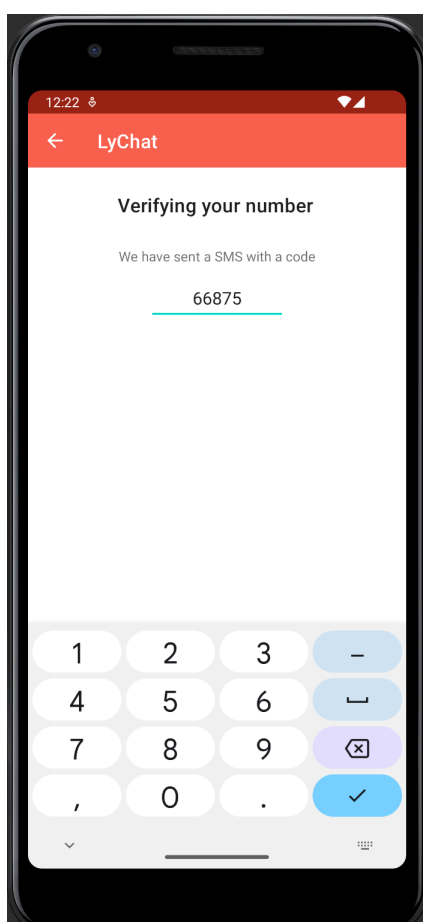


Рисунок 3.5 – Екран верифікації номера телефону

Після вводу всіх 6 цифр, додаток одразу починає верифікувати номер телефону. На цьому етапі після успішної верифікації номера, перевіряється умова: чи вже зареєстровано цей номер телефону в Firebase Authentication? Якщо так, то користувач входить у свій акаунт і потрапляє на екран чатів, тобто відбувається процедура автентифікації. В іншому ж випадку, коли цей номер не зареєстровано Firebase Authentication, відкривається екран реєстрації нового акаунту. Наразі,

номер телефону, який був ведений не зареєстровано, то відкрилось екран реєстрації (рис 3.6).

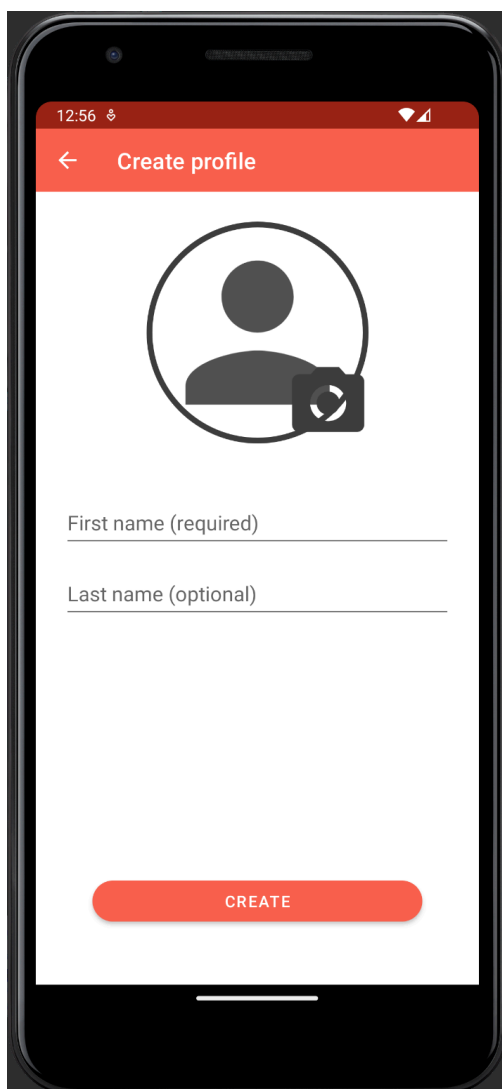


Рисунок 3.6 – Екран реєстрації нового акаунту

Ім'я є обов'язковим полем, прізвище – необов'язковим. Також опціонально можна обрати фотографію профілю. Якщо не обрати фотографію, буде встановлена стандартна картинка.

Введемо ім'я та прізвище у відповідні поля. При натисканні на картинку відкривається локальне сховище з останніми зображеннями (рис 3.7). Обираємо потрібне зображення та потрапляємо назад на екран реєстрації (рис 3.8).

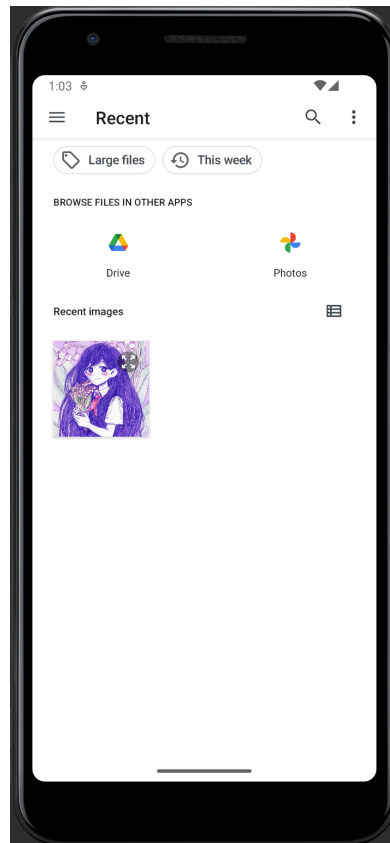


Рисунок 3.7 – Локальне сховище фотографій

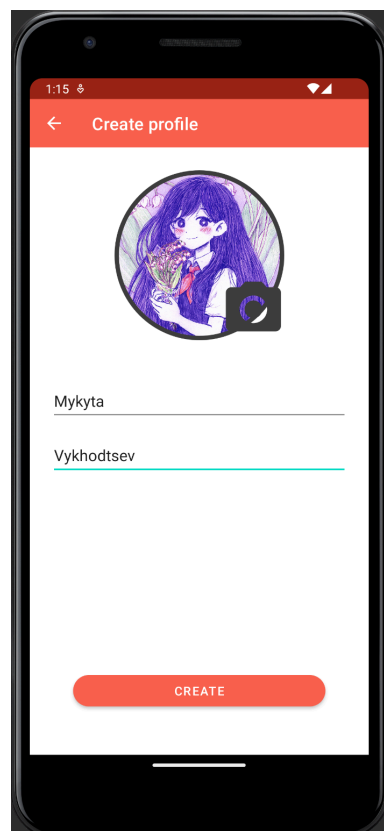


Рисунок 3.8 – Екран реєстрації нового акаунту з заповненими полями

При натисканні на кнопку «Створити», акаунт зберігається в базу даних та відкривається екран чатів месенджера. Вийдемо з акаунту та спробуємо до нього зайти, через процедуру автентифікації. Введемо той самий номер телефона (рис 3.3) та верифікуємо його. Після вводу верифікаційного коду, відбувається вхід в акаунт та перенаправлення на екран чатів месенджера.

3.1.2 Перевірка роботи пошуку

Після створення нового акаунту, користувач потрапляє в пустий екран чатів (рис 3.9). Він пустий, бо у користувача ще немає чатів, які повинні тут відображатися.

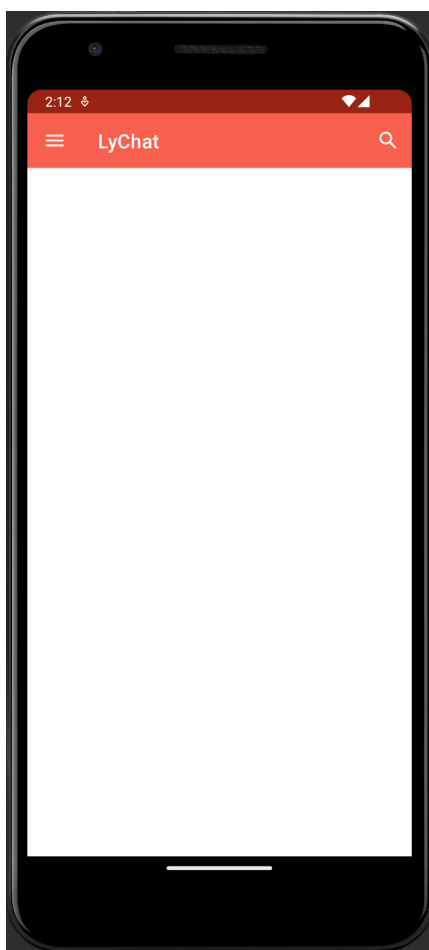


Рисунок 3.9 – Пустий екран чатів

Для того, щоб почати спілкуватися, спочатку треба знайти іншого користувача. Для цього потрібно клацнути на лупу, що знаходиться в правому верхньому куті головного екрану (рис 3.9). При кліку відкривається екран пошуку (рис 3.10). Введемо в пошукове поле нікнейм іншого користувача (рис 3.11). При натисканні на користувача відкривається чат з користувачем.



Рисунок 3.10 – Екран пошуку

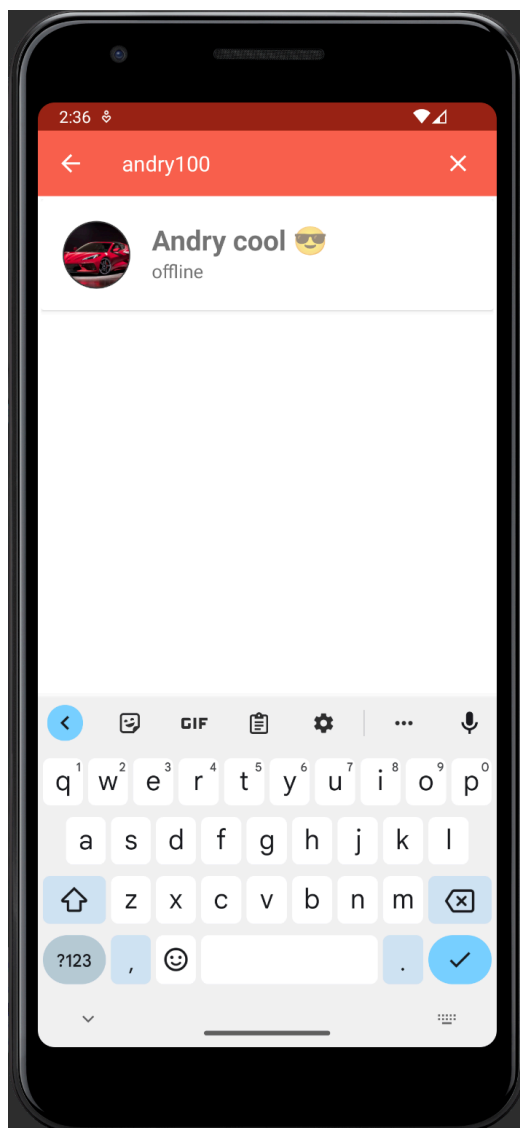


Рисунок 3.11 – Екран пошуку з знайденим користувачем

3.1.3 Перевірка роботи миттєвих повідомлень

Після того, як чат відкрився, можна спокійно надсилати повідомлення співбесіднику. Для цього потрібно написати повідомлення та натиснути на кнопку відправки. Повідомлення миттєво відправляється та з'являється на екрані адресанта та адресата. На рисунку 3.12 наведено екран чату з листуванням.

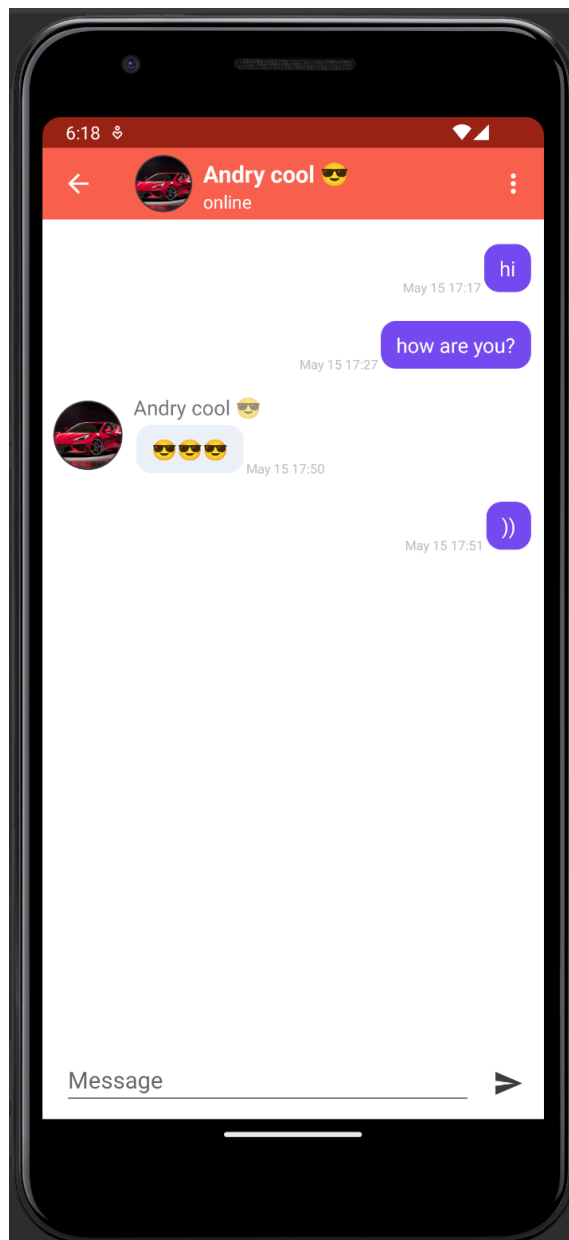


Рисунок 3.12 – Листування в чаті

3.1.4 Перевірка роботи відображення чатів

Після того, як на акаунті з'явився перший чат, він почав відображатися на екрані чатів (рис 3.13). З появою ще нових чатів, всі вони будуть відображатися на цьому екрані.



Рисунок 3.13 – Екран чатів

3.1.5 Перевірка роботи персоналізації профілю чатів

Тепер перевіримо роботу персоналізації профілю. Для початку треба натиснути на іконку меню, що знаходиться у верхньому лівому куті екрану чатів, для того, щоб відкрити навігаційну панель (рис 3.14). На цьому екрані натискаємо на пункт «Settings», відкривається екран налаштування профілю (рис 3.15).

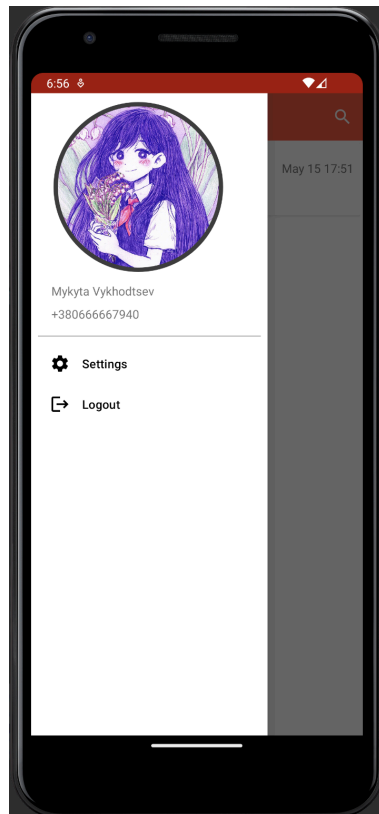


Рисунок 3.14 – Навігаційна панель додатка

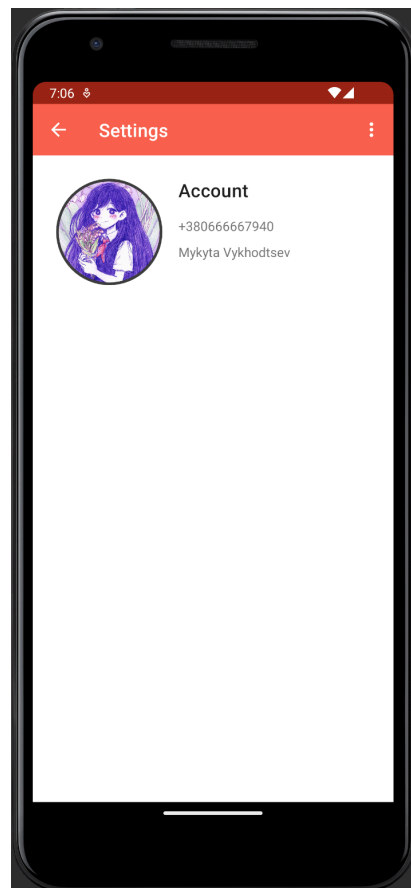


Рисунок 3.15 – Екран налаштування профілю

На цьому екрані натискаємо на іконку трьох крапок, що знаходиться в правому верхньому куті, з'являється випадаюче меню (рис 3.16).

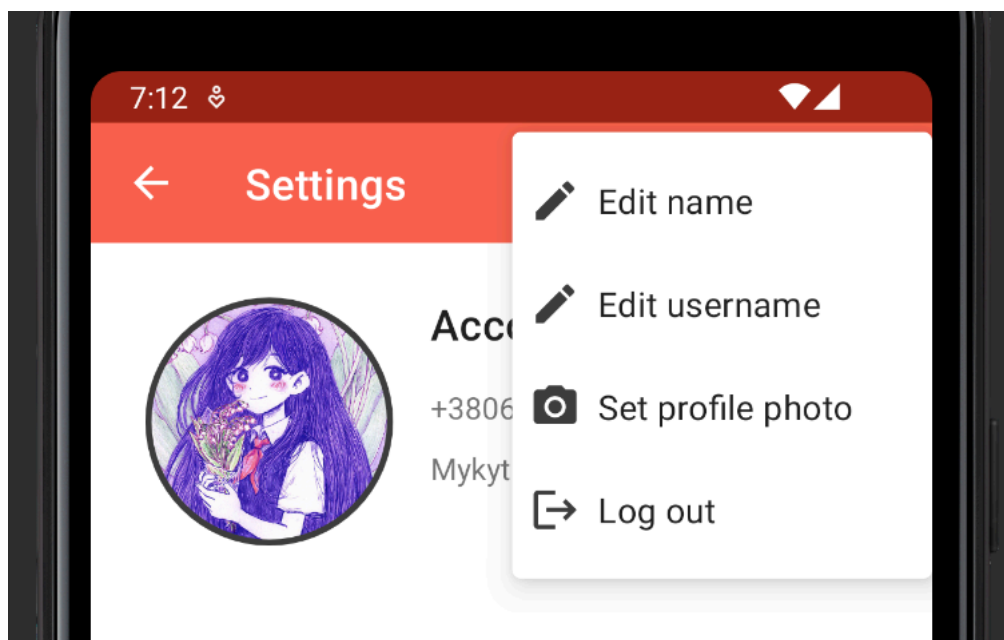


Рисунок 3.16 – Випадаюче меню

Оберемо пункт «Edit username». Відкривається нове вікно, де можна встановити нікнейм (рис 3.17), впишемо і збережемо його.

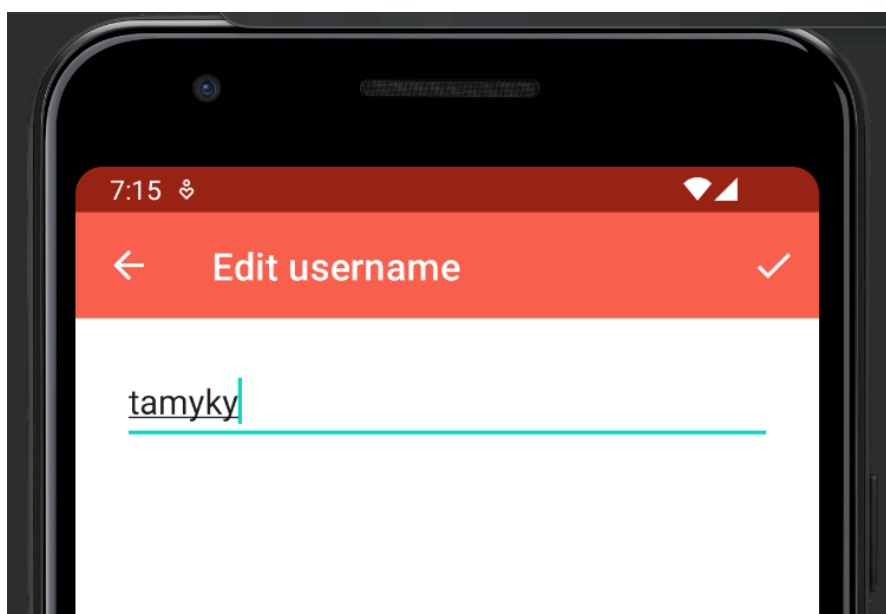


Рисунок 3.17 – Екран оновлення нікнейму

Тепер повернемося до випадаючого меню (рис 3.16) та оберемо пункт «Set profile photo». Відкривається локальне сховище фотографій (рис 3.7), оберемо нову фотографію. Після натискання на фотографію знов з'являється екран налаштувань, вже з оновленою інформацією (рис 3.18).

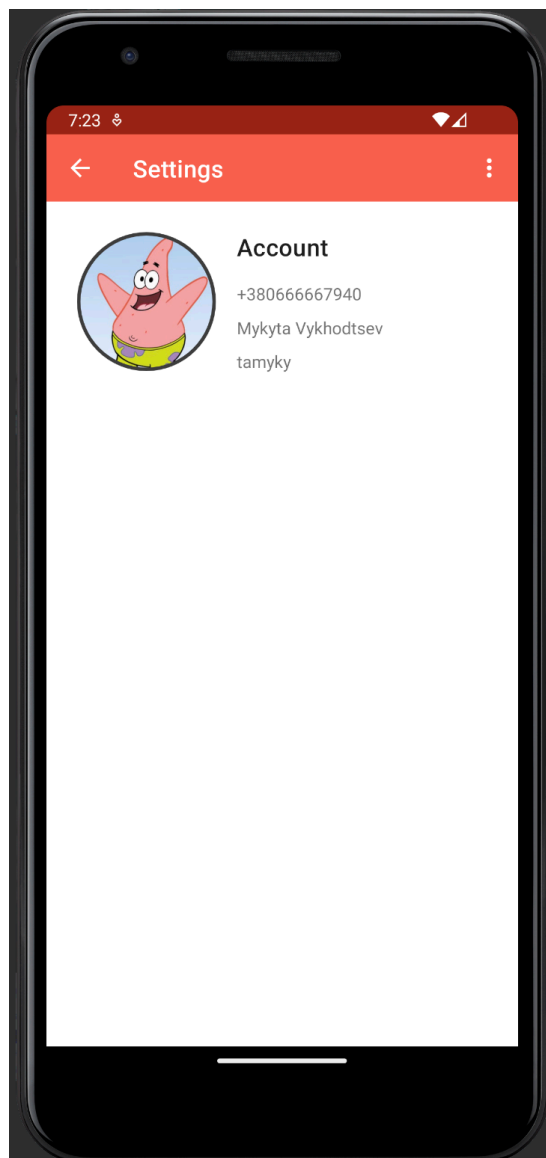


Рисунок 3.18 – Екран налаштування профілю з оновленою інформацією

3.2 Тестування програми

Система має працювати правильно без перебоїв та проблем. Для цього потрібно провести тестування програмного продукту. Для цього напишемо тест кейси для тестування функціонала.

Тест кейс – документ, що є інструкцією тестування. Він містить певні передумови та кроки, для виконання тестової ситуації й також очікуваний результат.

Таблиця 3.1 – Тест кейс позитивної автентифікації

Тест кейс 01. Позитивна автентифікація користувача		
Передумови: відкритий додаток на екрані вводу номера телефону та наявність акаунту		
Кроки	Дані тесту	Очікуваний результат
Введення номера телефону в поле	Валідний номер телефону	Відкривається вікно верифікації
Введення верифікаційного коду	Верифікаційний код, що прийшов на введений номер телефон	Верифікація успішна, виконано вхід в акаунт, відкривається вікно чатів

Таблиця 3.2 – Тест кейс негативної автентифікації

Тест кейс 02. Негативна автентифікація користувача		
Передумови: відкритий додаток на екрані вводу номера телефону та наявність акаунту		
Кроки	Дані тесту	Очікуваний результат
Введення номера телефону в поле	Не валідний номер телефону	Повідомлення, що номер телефону не валідний

Таблиця 3.3 – Тест кейс негативної автентифікації

Тест кейс 03. Негативна автентифікація користувача		
Передумови: відкритий додаток на екрані вводу номера телефону та наявність акаунту		
Кроки	Дані тесту	Очікуваний результат
Введення номера телефону в поле	Валідний номер телефону	Відкривається вікно верифікації
Введення верифікаційного коду	Випадкові 6 цифр	Повідомлення користувача, що верифікація не пройшла

Таблиця 3.4 – Тест кейс позитивної реєстрації

Тест кейс 04. Позитивна реєстрація користувача		
Передумова: відкритий додаток на екрані вводу номера телефону		
Кроки	Дані тесту	Очікуваний результат
Введення номера телефону в поле	Валідний номер телефону	Відкривається вікно верифікації
Введення верифікаційного коду	Верифікаційний код, що прийшов на введений номер телефону	Верифікація успішна, відкривається вікно реєстрації
Введення даних в поля реєстрації	Ім'я – «Микита»	Успішна реєстрація, виконано вхід в акаунт, відкривається вікно чатів.

Таблиця 3.5 – Тест кейс пошуку

Тест кейс 05. Пошук користувача		
Передумова: відкритий додаток з виконаним входом в акаунт на екрані чатів		
Кроки	Дані тесту	Очікуваний результат
Нажати на лупу, у правому верхньому куті екрану		Відкривається вікно пошуку
Введення номера телефону у поле пошуку	Номер телефону користувача, якого шукаємо	Користувача знайдено та відображено

Таблиця 3.6 – Тест кейс відправлення повідомлень

Тест кейс 06. Відправлення повідомлення		
Передумова: відкритий додаток з виконаним входом в акаунт на екрані чатів, та наявність хоча б одного чату		
Кроки	Дані тесту	Очікуваний результат
Нажати на чат в списку на екрані		Відкривається чат з користувачем, який було обрано
Відправка повідомлення в чат	Довільне текстове повідомлення	Повідомлення відправлено в чат та відображається адресанту та адресату

Таблиця 3.7 – Тест кейс оновлення фотографії профілю

Тест кейс 07. Відправлення повідомлення		
Передумова: відкритий додаток з виконаним входом в акаунт на екрані чатів		
Кроки	Дані тесту	Очікуваний результат
Нажати на іконку меню в верхньому лівому куті екрану		Відкривається навігаційна панель
Нажати на пункт «Налаштування»		Відкривається екран налаштувань профілю
Натиснути на три крапки у правому куті екрану		З'являється випадаюче меню
У випадаючому меню обрати пункт «Встановити фото профілю»		Відкривається локальне сховище пристрою
Натиснути на потрібну фотографію	Фотографії, що збережені на пристрої користувача	Повертається екран налаштування профілю, фото профілю оновлюється

ВИСНОВКИ

У ході виконання випускної кваліфікаційної роботи бакалавра був розроблений месенджер LyChat. Було проведено аналіз систем миттєвих повідомлень, розглянуто сучасні месенджери та сформовано вимоги до системи.

Месенджер LyChat було розроблено на client-server архітектурі з використанням хмарних технологій, які надає Firebase. Клієнтський додаток було створено мовою програмування Java на мобільні пристрої Android.

Фінальна версія месенджера відповідає всім поставленим вимогам та має всі найважливіші функції:

- реєстрація користувача за номером телефону;
- автентифікація користувача за номером телефону;
- персоналізація профілю, можливість встановлення імені, прізвища, нікнейму та фотографії профілю;
- пошук інших користувачів за їхнім номером телефону, ім'ям, прізвищем, нікнеймом або номером телефону;
- відображення чатів користувача;
- відправка повідомлень іншим користувачам в чатах;
- мережевий статус користувача (онлайн або офлайн);

У майбутньому планується розробити додаткові функції месенджера та покращити вже існуючі. Також розробити клієнтську версію месенджера на операційну систему IOS.

ПЕРЕЛІК ПОСИЛАНЬ

1. Instant messaging [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Instant_messaging.
2. Messaging Apps & Communication Platforms Comparative Analysis [Електронний ресурс] // Deep Knowledge Analytics. – 2021. – Режим доступу до ресурсу: <https://analytics.dkv.global/messengers-report.pdf>.
3. Most popular global mobile messenger apps as of January 2023, based on number of monthly active users [Електронний ресурс]. – 2023. – Режим доступу до ресурсу: <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>.
4. ЯКІ МОБІЛЬНІ ДОДАТКИ Є НАЙБІЛЬШ ПОПУЛЯРНИМИ? [Електронний ресурс] // Київський міжнародний інститут соціології. – 2021. – Режим доступу до ресурсу: <https://www.kiis.com.ua/?lang=ukr&cat=reports&id=1072&page=1>.
5. WhatsApp [Електронний ресурс] – Режим доступу до ресурсу: <https://www.whatsapp.com/>.
6. Viber: Home [Електронний ресурс] – Режим доступу до ресурсу: <https://www.viber.com/ua/>.
7. Telegram Messenger [Електронний ресурс] – Режим доступу до ресурсу: <https://telegram.org/>.
8. Signal >> Home [Електронний ресурс] – Режим доступу до ресурсу: <https://signal.org/en/>.
9. Android Developers: Android Mobile App Developer Tools [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/>.
10. Firebase [Електронний ресурс] – Режим доступу до ресурсу: <https://firebase.google.com/>.
11. ReactiveX [Електронний ресурс] – Режим доступу до ресурсу: <https://reactivex.io/>
12. Material Design Components [Електронний ресурс] – Режим доступу до

ресурсы: <https://m2.material.io/components?platform=android>.

13. The Reactive Manifesto. [Электронный ресурс]. – Режим доступа: <https://www.reactivemanifesto.org/>

14. Carty D. What Is a Test Case? Examples, Types, Format and Tips [Электронный ресурс] / David Carty // APPLAUSE. – 2022. – Режим доступа до ресурсу: <https://www.applause.com/blog/what-is-a-test-case-examples-types-format>.

15. Google Cloud documentation [Электронный ресурс] – Режим доступа до ресурсу: <https://cloud.google.com/docs>.

16. Java Developers [Электронный ресурс] – Режим доступа до ресурсу: <https://dev.java/>.

17. Desmond M. What Is the Client-Server Model? (Components and Benefits) [Электронный ресурс] / Motiso Desmond // Indeed. – 2022. – Режим доступа до ресурсу: <https://www.indeed.com/career-advice/career-development/what-is-client-server-model>.

18. Introduction to Material Design in Android [Электронный ресурс] // GeeksforGeeks. – 2022. – Режим доступа до ресурсу: <https://www.geeksforgeeks.org/introduction-to-material-design-in-android/>.

ДОДАТОК А



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



РОЗРОБКА МЕСЕНДЖЕРУ «LYCHAT» ДЛЯ ANDROID МОВОЮ JAVA

Виконав студент 4 курсу
Групи ПД-41
Виходцев Микита Миколайович
Керівник роботи
К.т.н, доц, доцент кафедри ІПЗ Трінтіна Наталія Альбертівна

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи – покращення якості спілкування в інтернеті за рахунок підвищення рівня безпеки та зручності при використанні месенджера.

Об'єкт дослідження – процес комунікації в інтернеті.

Предмет дослідження – система миттєвих повідомлень.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

1. Проаналізувати сучасні месенджери, визначити їх переваги та недоліки.
2. Визначити вимоги до месенджерів з урахуванням недоліків сучасних месенджерів.
3. Провести огляд технічних засобів реалізації для розробки системи.
4. Спроектувати та розробити месенджер.
5. Провести тестування месенджеру.

3

АНАЛІЗ АНАЛОГІВ

	Telegram	WhatsApp	Viber	Signal	LyChat
Реєстрація за номером телефону	+	+	+	+	+
Чати	+	+	+	+	+
Шифрування за замовченням	-	+	+	+	+
Має централізовану архітектуру	+	+	+	+	+
Повідомлення зберігаються на сервері	+	+	-	-	+

4

ВИМОГИ ДО МЕСЕНДЖЕРУ

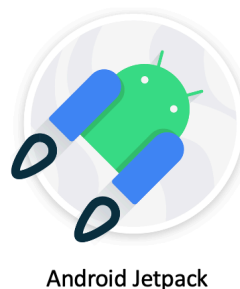
1. Реєстрація користувача за номером телефону.
2. Автентифікація користувача за номером телефону.
3. Персоналізація профілю, можливість встановлення імені, прізвища, нікнейму та фотографії профілю.
4. Пошук інших користувачів за їхнім номером телефону, ім'ям, прізвищем, нікнеймом або номером телефону.
5. Відображення чатів користувача.
6. Відправка повідомлень іншим користувачам в чатах.
7. Відображення мережевого статусу користувача (онлайн або офлайн).

5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Android Studio



Android Jetpack



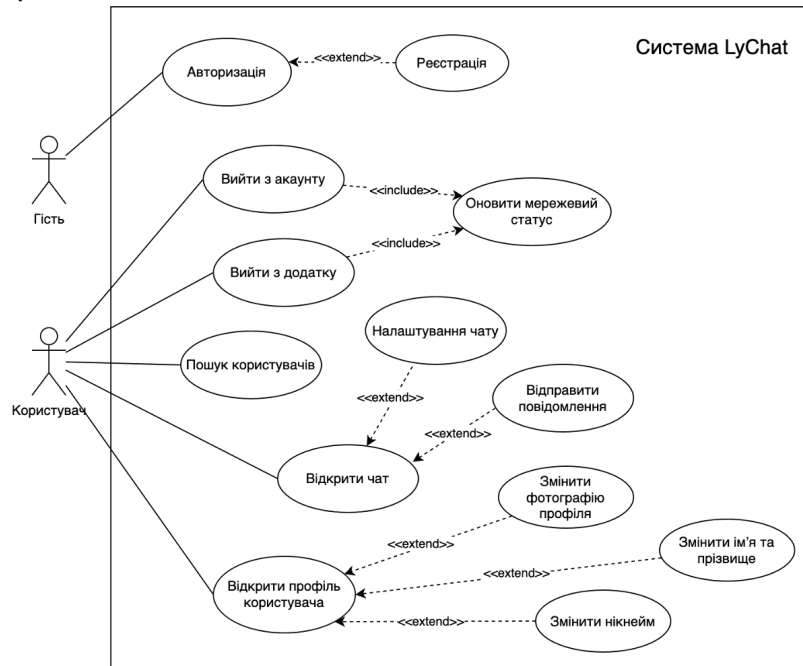
RxJava



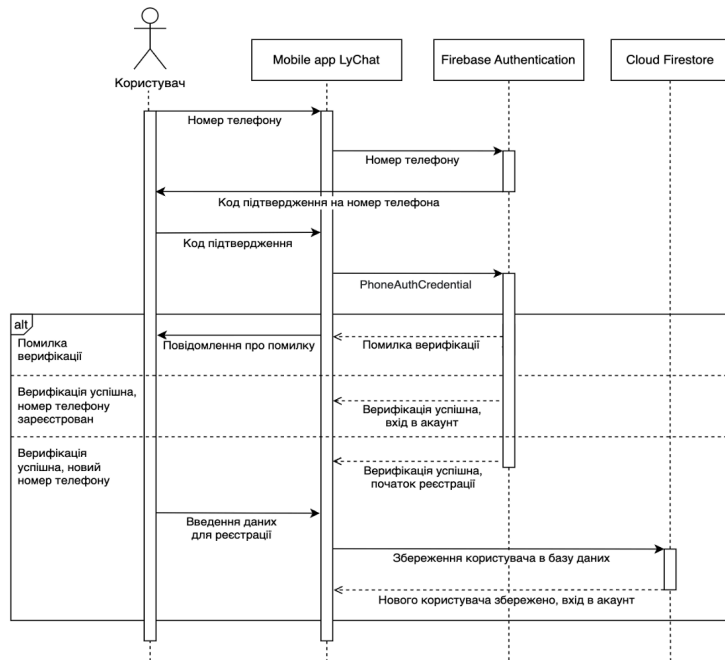
Firebase

6

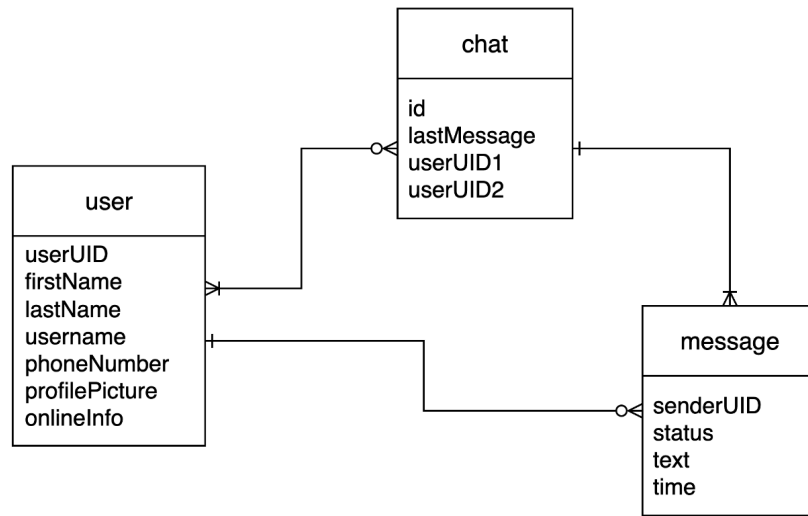
ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ



ДІАГРАМА ПОСЛІДОВНОСТІ РЕЄСТРАЦІЇ ТА АВТЕНТИФІКАЦІЇ

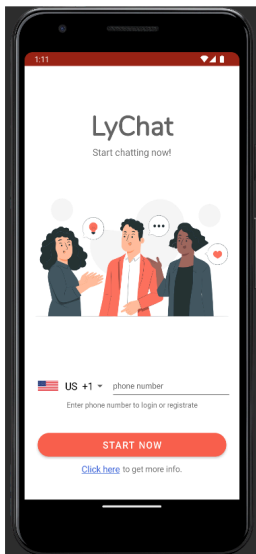


ER-ДІАГРАМА БАЗИ ДАНИХ

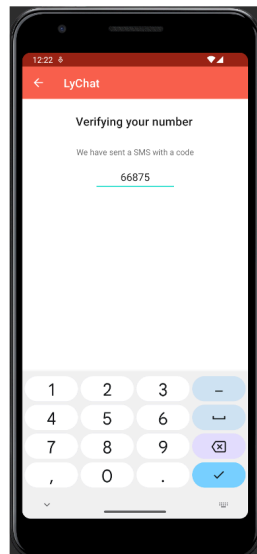


9

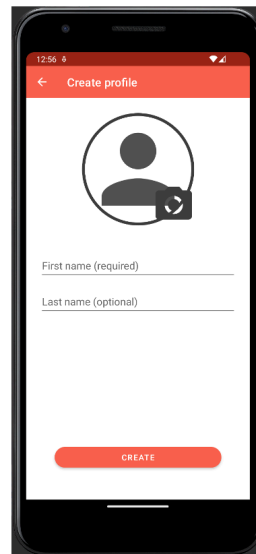
ЕКРАННІ ФОРМИ



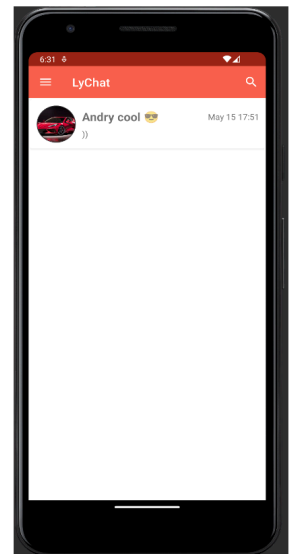
Екран вводу номера телефону



Екран верифікації номера телефону



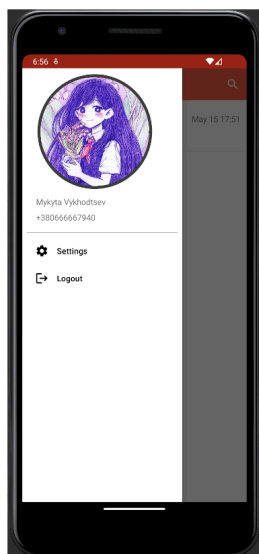
Екран реєстрації нового акаунту



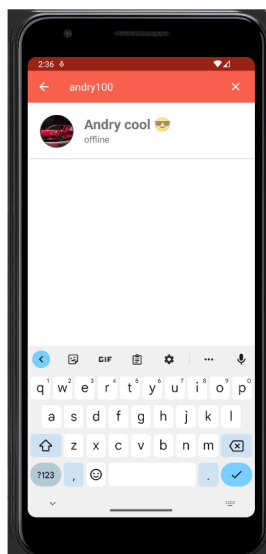
Головний екран – екран чатів

10

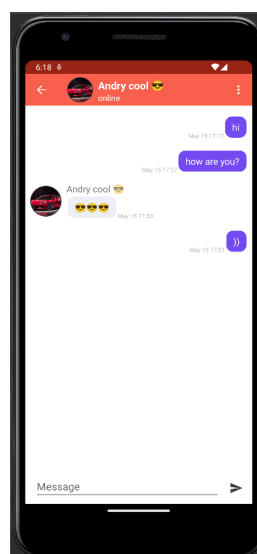
ЕКРАННІ ФОРМИ



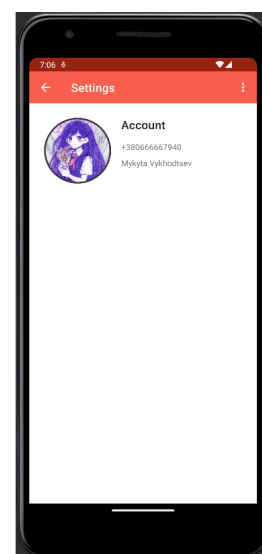
Навігаційна панель



Екран пошуку користувачів



Екран чату



Екран персоналізації профілю

11

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

1. Виходцев М.М. Реактивне програмування / IV Всеукраїнській науково-технічній конференції «Сучасний стан та перспективи розвитку IoT». Збірник тез. 07.04.2023, ДУТ, м. Київ — К.: ДУТ, 2023. — С. 38-40.
2. Виходцев М.М. Розробка месенджера на Android / Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». Збірник тез. 20.04.2023, ДУТ, м. Київ — К.: ДУТ, 2023. — С. 67-68.

12

ВИСНОВКИ

1. Проаналізовано сучасні месенджери: WhatsApp, Viber, Telegram та Signal. Визначено, що головним недоліком є проблеми з безпекою даних користувачів.
2. Спроектовано та розроблено, відповідно до визначених вимог, централізовану систему миттєвих повідомлень на Android мовою програмування Java з використанням хмарних технологій, які надає Firebase.
3. Забезпечено високий рівень безпеки месенджера, завдяки функціям Firebase: захист від хакерських атак, шифрування трафіку між додатком та сервером, шифрування збережених даних користувача в базі даних і налаштування доступу до них.
4. Розроблено тест кейси та проведено тестування месенджера.