

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ**

Рівень вищої освіти  
Спеціальність  
Освітня програма  
Група

Перший (бакалаврський)  
Інженерія програмного забезпечення  
Інженерія програмного забезпечення  
6.04.121.010.18.1

## **ДИПЛОМНИЙ ПРОЄКТ**

на тему: «Розроблення месенджера з використанням  
технологій стеку MERN»

Виконав: студент Кирило ХІМАН

Керівник: проф., к.е.н., доцент Ірина ЗОЛОТАРЬОВА

Рецензент: завідувачка кафедри програмного  
забезпечення комп'ютерних систем національного  
технічного університету «Дніпровська політехніка»,  
к.т.н., доцент Ірина УДОВИК

Харків – 2022 рік

## РЕФЕРАТ

Пояснювальна записка до дипломного проєкту: 44 с., 23 рис., 15 табл., 16 джерел.

Об'єктами проєктування є функціональні елементи, архітектура, інформаційне і програмне забезпечення веб-застосунку «Месенджер».

Мета проєктування: розробити Single Page Application використовуючи технології стеку MERN.

Методи проєктування: використання застосунків Ramus Educational, Draw.io, Figma, Visual Studio Code, MongoDB Cloud, Google Chrome.

Для успішного виконання обраної задачі, було проведено порівняльний аналіз існуючих аналогів, з метою виявлення недоліків та переваг. На базі отриманих результатів та визначених вимог було розроблено веб-застосунок «Месенджер» використовуючи технології MongoDB, Express, React та Node.js, опис архітектури, діаграми та макет інтерфейсу користувача.

В результаті чого, було створено веб-застосунок, котрий дозволяє вести текстовий діалог в реальному часі, створювати одиночні та групові чати та обмінюватися фотографіями різних форматів.

Рекомендації щодо використання результатів роботи: створений веб-застосунок може використовуватися для комунікації в реальному часі між двома та більшою кількістю людей.

МЕСЕНДЖЕР, ЧАТ, ФУНКЦІОНАЛ, ІНТЕРНЕТ, БАЗА ДАНИХ, SINGLE PAGE APPLICATION, HTML, CSS, JAVASCRIPT, REACT, NODE.JS, MONGODB, ОБМІН ІНФОРМАЦІЄЮ У РЕАЛЬНОМУ ЧАСІ.

## ABSTRACT

Volume of work: 44 pp., 23 figs., 15 tables, 16 sources.

The objects of design are functional elements, architecture, information and software of the web application "Messenger".

Design goal: to develop a Single Page Application using MERN stack technology.

Design methods: use of Ramus Educational, Draw.io, Figma, Visual Studio Code, MongoDB Cloud, Google Chrome.

To successfully complete the selected task, a comparative analysis of existing analogues was conducted in order to identify shortcomings and advantages. Based on the obtained results and defined requirements, the Messenger web application was developed using MongoDB, Express, React and Node.js technologies, description of architecture, diagrams and layout of the user interface.

As a result, a web application has been created that allows you to have real-time text dialogue, create single and group chats, and share photos in a variety of formats.

Recommendations for using the results of the work: the created web application can be used for real-time communication between two or more people.

MESSENGER, CHAT, FUNCTIONAL, INTERNET, DATABASE, SINGLE PAGE APPLICATION, HTML, CSS, JAVASCRIPT, REACT, NODE.JS, MONGODB, EXCHANGE OF INFORMATION IN REAL TIME.

## ЗМІСТ

ВСТУП .....	9
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	11
1.1 Коротка характеристика об'єкту управління ФОП «ТРУБІН Д.В.» .....	11
1.2 Опис предметної області «Розроблення месенджера з використанням технологій стеку MERN» .....	12
1.3 Огляд і аналіз існуючих аналогів, що реалізують функції предметної області.....	17
2. СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-ЗАСТОСУНКУ «МЕСЕНДЖЕР» .....	21
2.1 Глосарій.....	21
2.2 Розроблення варіантів використання .....	21
2.2.1 Діаграма варіантів використання .....	21
2.2.2 Специфікація варіантів використання .....	22
2.3 Специфікація функціональних вимог .....	25
2.4 Проектування інтерфейсу користувача .....	26
3. ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ .....	28
3.1 Логічна постановка .....	28
3.2 Проектування структури бази даних.....	28
3.2.1 Концептуальне інфологічне проектування.....	28
3.2.2 Проектування логічної моделі даних .....	30
3.2.3 Проектування фізичної моделі даних .....	31
3.3 Проектування програмного забезпечення .....	31
3.4 Тестування програмної системи .....	36
3.5 Розгортання програмного продукту .....	41
ВИСНОВКИ.....	42
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	43
ДОДАТКИ.....	44

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

MERN – MongoDB, Express, React, Node.js — набір серверного програмного забезпечення, який використовується для веб-розробки, орієнтований на JavaScript

HTML – Hyper Text Markup Language — стандартизована мова розмітки документів для перегляду веб-сторінок у браузері

CSS – Cascading Style Sheets — формальна мова опису зовнішнього вигляду документа, написаного з використанням мови розмітки (HTML)

СУБД – система управління базами даних

БД – база даних

API – Application Programming Interface — опис способів, якими одна комп'ютерна програма може взаємодіяти з іншою програмою

SPA – Single Page Application — односторінковий додаток

## ВСТУП

У сучасному світі все більше людей надають перевагу спілкуванню на відстані, використовуючи при цьому чати або месенджери. Сучасні месенджери надають можливість не тільки обмінюватись текстовими повідомленнями, але й файлами, голосовими повідомленнями, або навіть відеодзвінками. На сьогоднішній день для багатьох людей такі програмні продукти є і залишаються основним інструментом для спілкування між собою.

Найчастіше чат – це веб-додаток, де можна у реальному часі спілкуватися з іншими відвідувачами чату, створювати приватні або групові діалоги. Кожен присутній при вході вписує свій «нікнейм», тобто прізвисько, за допомогою якого, його зможуть ідентифікувати інші користувачі застосунку.

Варто зазначити, що сучасні месенджери це не тільки засіб спілкування між знайомими та друзями, але й інструмент для спілкування з клієнтами, поряд з електронною поштою та телефоном. Це гнучкий інструмент комунікації, який власники сайтів можуть використовувати на своєму сайті для онлайн технічної підтримки. Якщо клієнтам знадобиться допомога, вони зможуть задати питання онлайн консультанту та швидко отримати відповідь. Грамотно працюючі з подібним сервісом, вони можуть генерувати велику кількість нових лідів, що зрештою призведе до зростання продажів.

Більш того чат може бути джерелом інформації о стані бізнесу, який може підказати у якому напрямку потрібно рухатись. Використовуючи чат виключно як інструмент комунікації, підприємці упускають велику кількість можливостей для розвитку свого бізнесу.

У цій науковій праці представлена програмна реалізація месенджера за допомогою якого користувачі можуть обмінюватись повідомленнями в реальному часі.

У якості програмного забезпечення для розробки цього веб-додатку були обрані технології стеку MERN, а саме:

- MongoDB – база даних, яка підтримує зберігання документів у JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів. Для роботи з цією базою даних було використано бібліотеку Mongoose, яка дозволяє нам визначати схеми зі строго-типізованими даними.

- Express.js, або просто Express – програмний каркас розробки серверної частини веб-застосунків для Node.js, спроектований для створення веб-застосунків і API. Де-факто є стандартним каркасом для Node.js.

- React – JavaScript-бібліотека, яка використовується для розробки SPA або мобільних додатків.

- Node.js – програмна платформа, розробленої на двигуні V8, що трансліює JavaScript в машинний код.

Також у якості технології для обміну даними в реальному часі було обрано JavaScript-бібліотеку Socket.IO. Вона складається з двох частин: клієнтської, яка запускається у браузері та серверної для Node.js. Обидва компоненти мають схожий API. Ця бібліотека головним чином використовує протокол WebSocket.

Завдяки вищеописаному веб-застосунку, користувачі зможуть створювати групові або особисті діалоги з іншими користувачами месенджеру та обмінюватись з ними інформацією у реальному часі.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Коротка характеристика об'єкту управління ФОП «ТРУБІН Д.В.»

Основним напрямком фізичної особи-підприємця Трубін Д.В. є надання послуг в сфері інформаційних технологій. Він зі своїми співробітниками займаються розробкою різноманітних веб, десктоп та мобільних додатків для клієнтів зі всього світу. Комунікація з клієнтами здійснюється здебільшого англійською, тому знання іноземної мови є одним із найважливіших навичок кожного члену команди. Схему організаційної структури управління підприємством наведено на рис. 1.1.

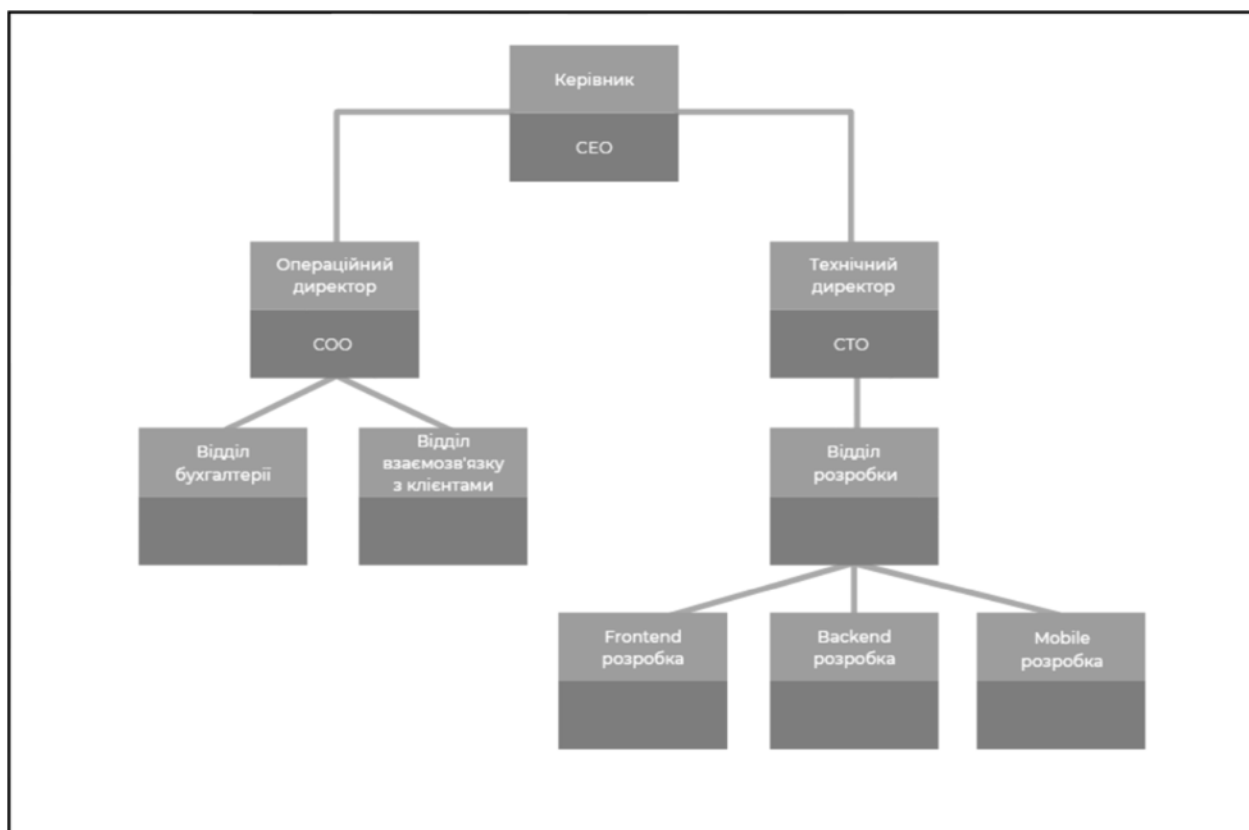


Рис. 1.1. Схема організаційної структури ФОП «ТРУБІН Д.В.»

Відділ розробки в даному підприємстві поділений на три категорії, а саме: frontend-розробники, які займаються розробкою клієнтської частини застосунків, backend-розробники, відповідальних за внутрішню (серверну) частину продукту та mobile-розробники, що працюють над мобільними додатками.



Підприємство переважно спеціалізується на розробці програмних продуктів, використовуючи сучасні веб-технології, а саме набір технологій MERN, за допомогою якого можна створювати всі види додатків, включаючи веб, десктопні або мобільні додатки.

## 1.2 Опис предметної області «Розроблення месенджера з використанням технологій стеку MERN»

В ході аналізу поставленої задачі «Розроблення месенджера з використанням технологій стеку MERN», включно з урахуванням вдосконалення й автоматизації деяких робіт, була розроблена контекстна діаграма та діаграма декомпозиції бізнес-процесів у стандарті IDEF0.

**ВХІД:** Дані користувача застосунку.

**ВИХІД:** Відправка повідомлення.

**МЕХАНІЗМИ:** Користувач, модуль перевірки.

**УПРАВЛІННЯ:** Правила використання веб-застосунку.

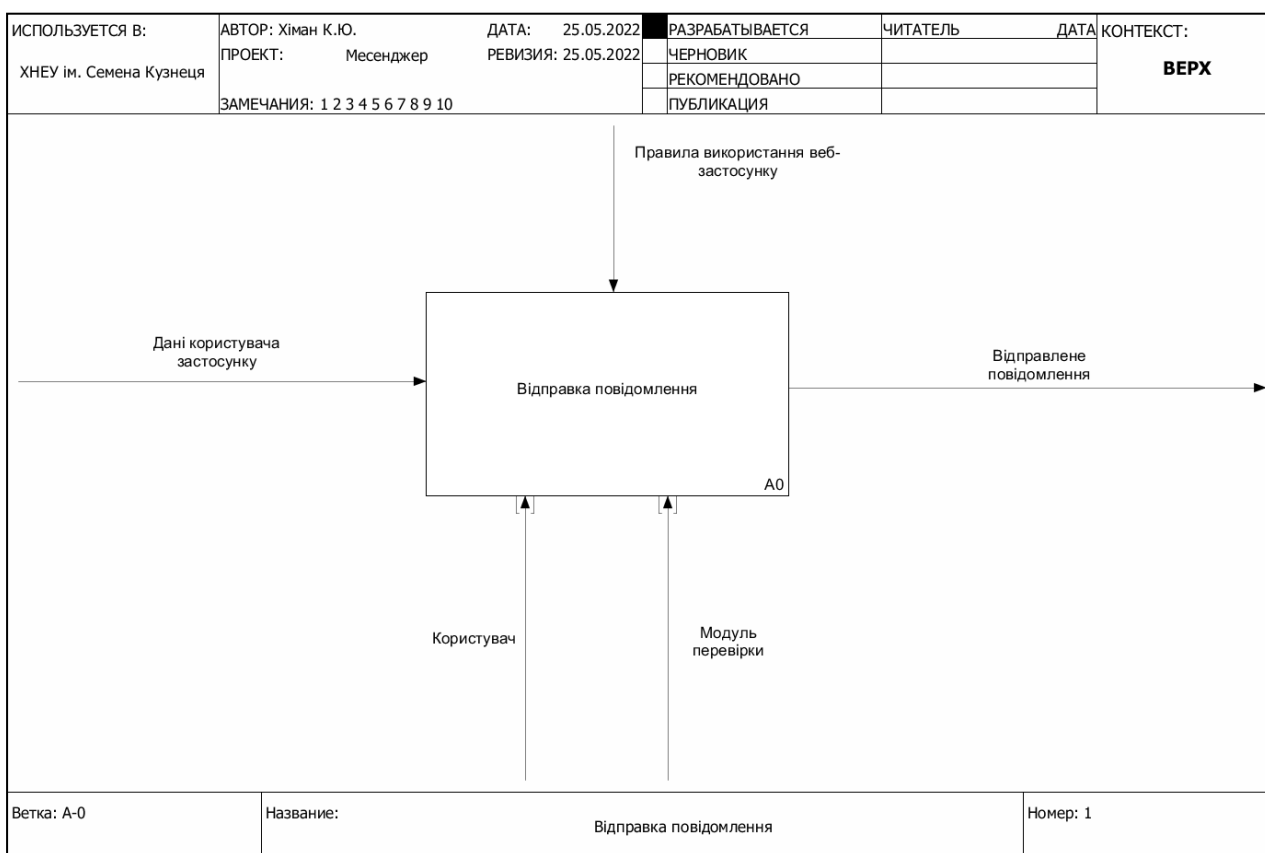


Рис. 2.1. Контекстна діаграма задачі в стандарті IDEF0

На рисунку 2.1 зображено нульовий рівень діаграми. У функціональному блоці описана головна функція, в якості вхідних даних приймаються ім'я користувача та пароль. Вихідні дані – це саме повідомлення. В якості механізмів виступають модуль перевірки на коректність введених даних та їх наявність у базі, а також сам користувач, так як він безпосередньо впливає на систему.

Декомпозиція контекстної діаграми (рис. 2.2) включає в себе наступні рівні:

**ВХІД:** Дані користувача застосунку.

**ВИХІД:** Відправлене повідомлення.

**МЕХАНІЗМИ:** Користувач, модуль перевірки.

**УПРАВЛІННЯ:** Правила використання веб-застосунку.

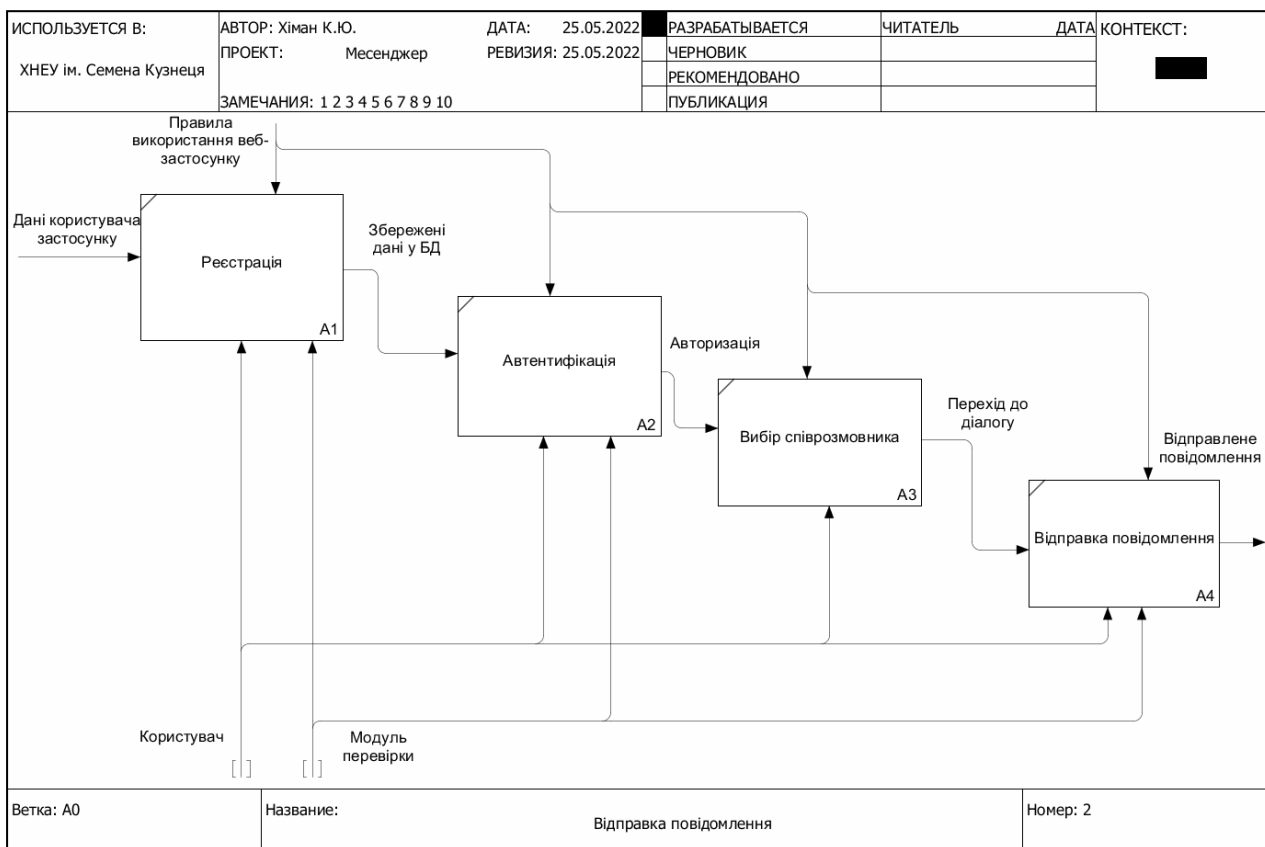


Рис. 2.2. Рівень A0

Декомпозиція блоку А1 «Реєстрація» (рис 2.3) складається з наступних блоків:

ВХІД: Електронна пошта, пароль

ВИХІД: Збережені дані у БД

МЕХАНІЗМИ: Користувач, модуль перевірки

УПРАВЛІННЯ: Правила використання веб-застосунку

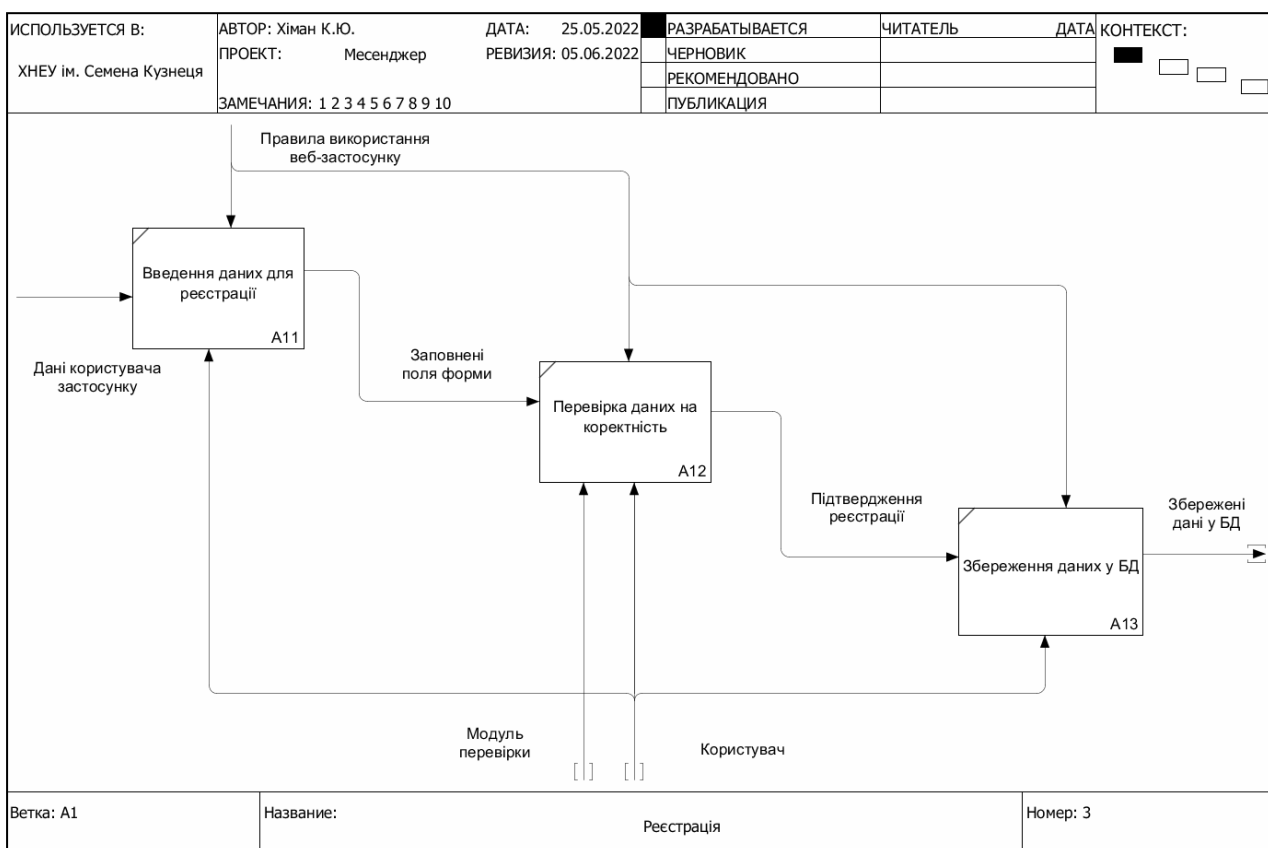


Рис. 2.3. Рівень А1

Наступний блок А2 «Авторизація» (рис 2.4) складається з таких інтерфейсних дуг:

ВХІД: Збережені дані у БД.

ВИХІД: Автентифікація.

МЕХАНІЗМИ: Користувач, модуль перевірки.

УПРАВЛІННЯ: Правила використання веб-застосунку.

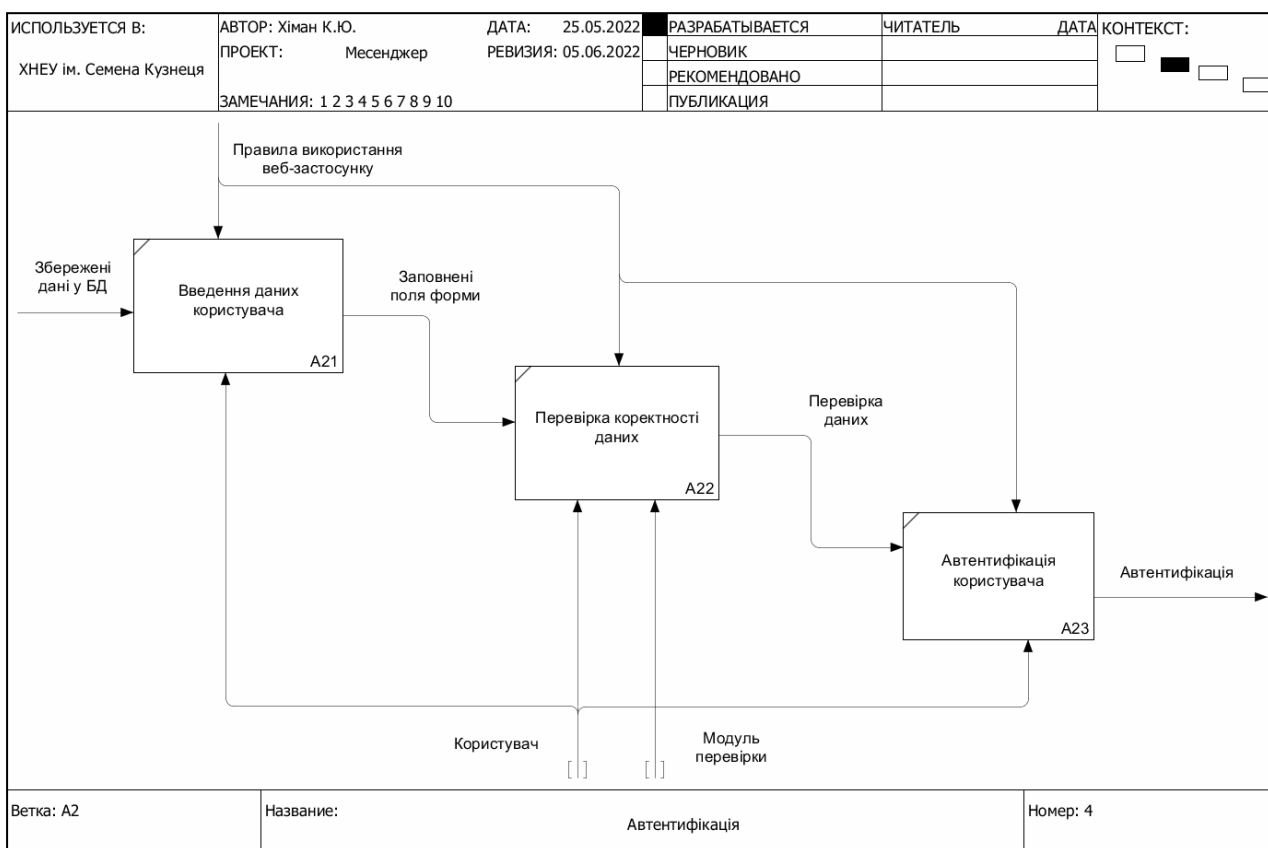


Рис. 2.4. Рівень А2

Рівень А3 «Вибір співрозмовника» складається з зображених на рис. 2.5 блоків:

ВХІД: Автентифікація.

ВИХІД: Перехід до діалогу.

МЕХАНІЗМИ: Користувач.

УПРАВЛІННЯ: Правила використання веб-застосунку.

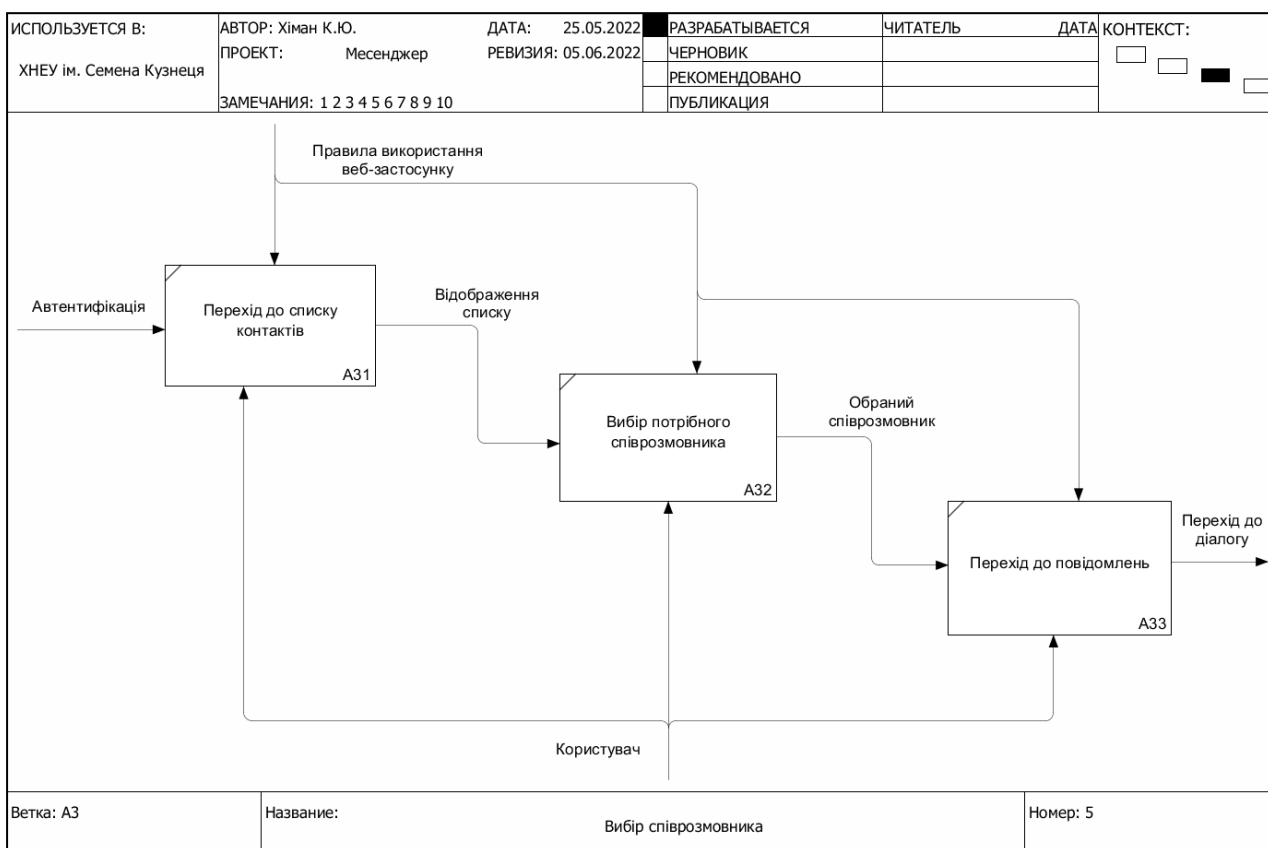


Рис. 2.5. Рівень А3

Останній рівень декомпозиції А4 «Відправка повідомлення» (рис. 2.6) складається з наступних блоків:

ВХІД: Перехід до діалогу.

ВИХІД: Відправлене повідомлення.

МЕХАНІЗМИ: Користувач, модуль перевірки.

УПРАВЛІННЯ: Правила використання веб-застосунку.

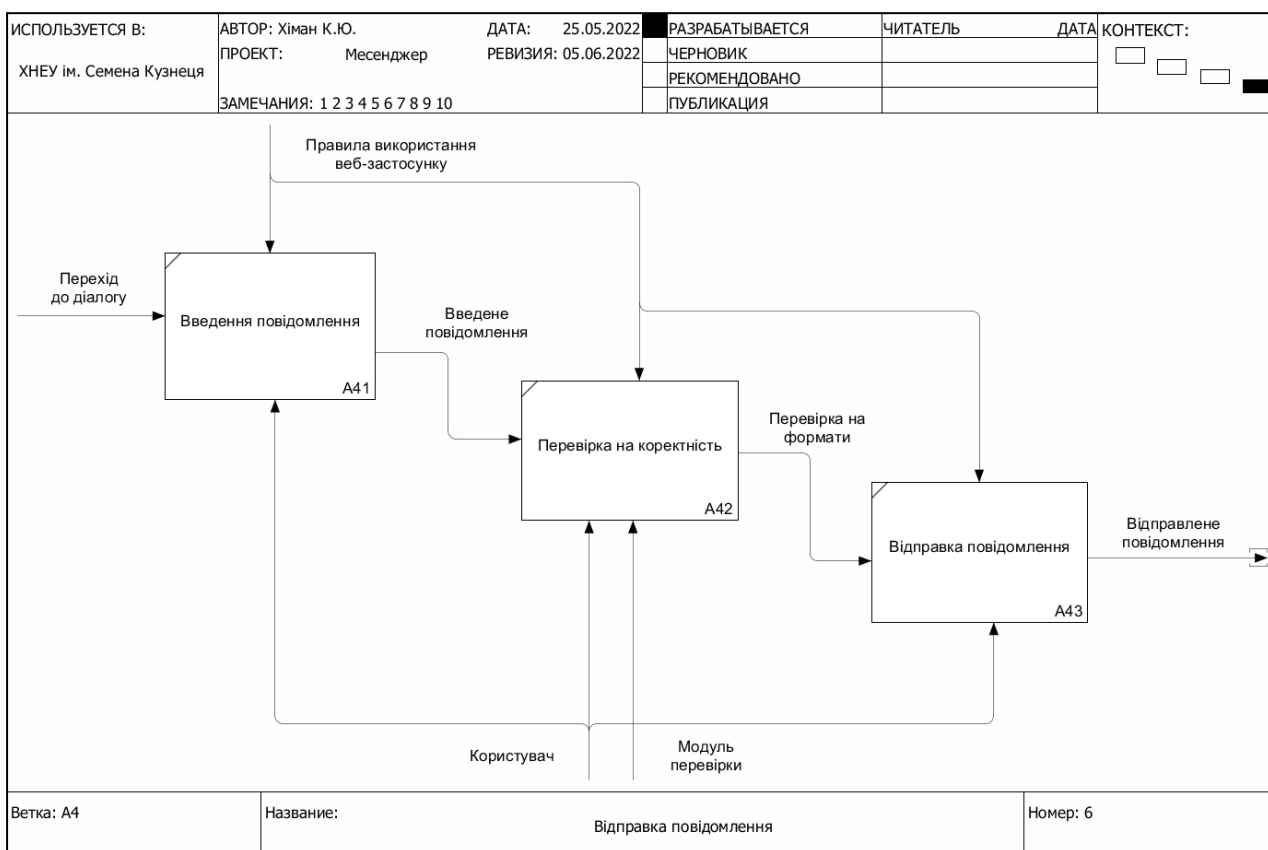


Рис. 2.6. Рівень А4

### 1.3 Огляд і аналіз існуючих аналогів, що реалізують функції предметної області

В ході огляду і аналізу існуючих аналогів, що реалізують функції предметної області, було розглянуті три програмних продукти: Telegram, WhatsApp та Viber, а саме десктопні та веб версії цих додатків.

Найголовнішою перевагами Telegram вважаються анонімність та інтуїтивно зрозумілий та сучасний інтерфейс. В Telegram реалізована не тільки передача повідомлень, але й дзвінки, боти та канали.

Особливість додатку в тому, що при авторизації на новому пристрої не потрібно завантажувати бекап-файли, все це відбувається автоматично, уся історія завантажується сама по собі, а файли зберігаються на сторонньому сервері й доступні у будь-який момент.

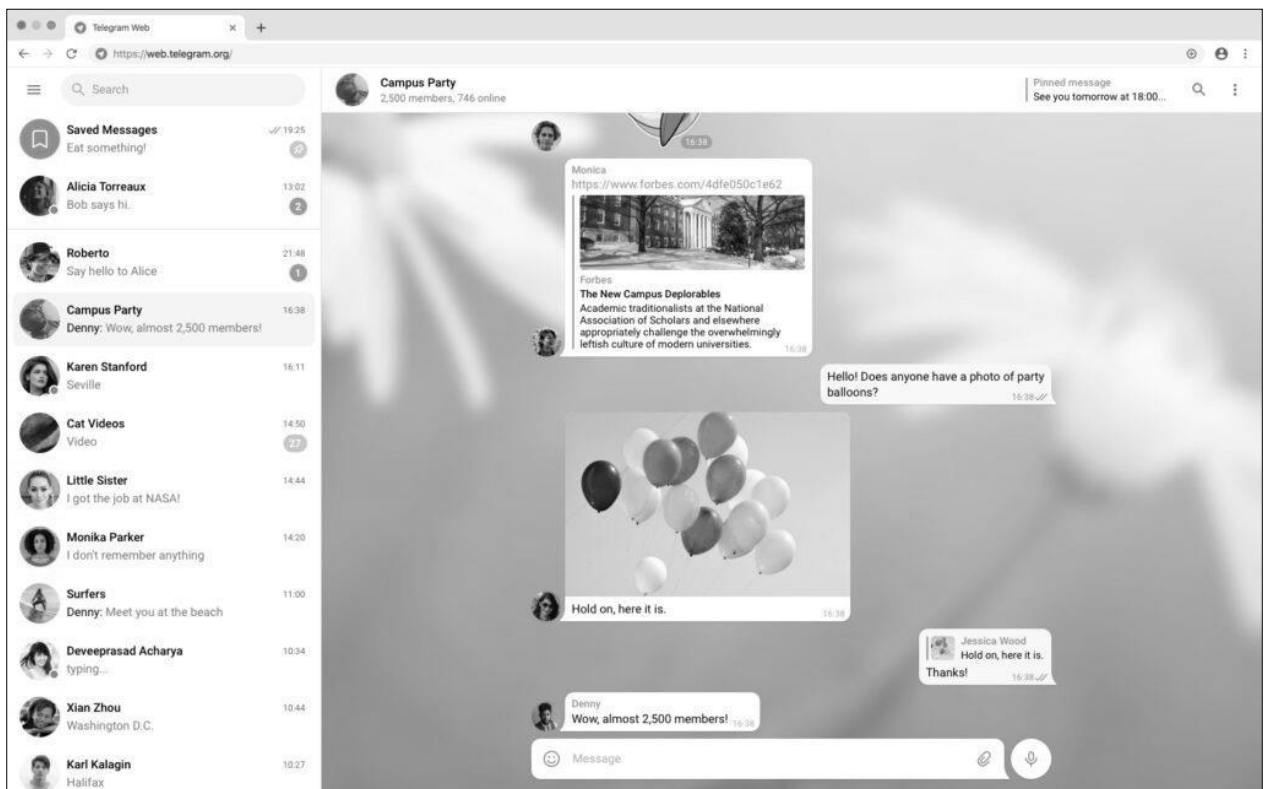


Рис. 3.1. Інтерфейс Telegram Web

WhatsApp – найбільш масовий месенджер. На початку додаток був платним: перший рік – без оплати, всі наступні роки – приблизно 1 долар в рік. Однак в версіях для ПК та веб-версії є особливість, яка не дозволяє використовувати систему, якщо на мобільному пристрої нема зв'язку з

інтернетом. Також не можна бути активним одночасно в веб-версії та версії для персонального комп'ютеру.

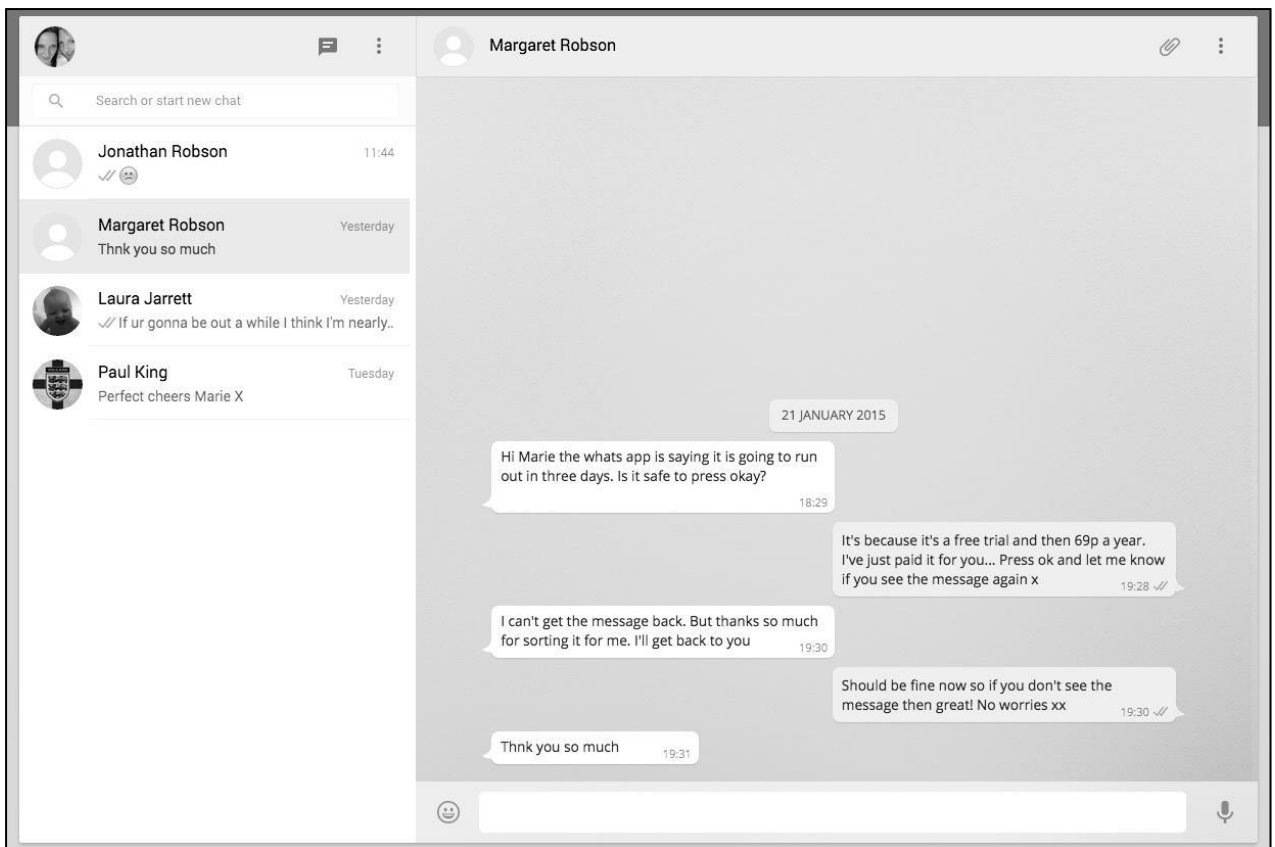


Рис. 3.2. Інтерфейс WhatsApp Web

Viber має шифрування, секретні бесіди, дзвінки та відео дзвінки, а також велику кількість стікерів, як платних, так і безкоштовних. На відміну від Telegram та WhatsApp, у нього немає веб-версії, що є доволі значним недоліком.

Також у Viber'і наявна велика кількість реклами. Додаток збирає дані користувачів для підбору спеціальних рекламних повідомлень.

Опрацювавши аналоги, було прийняте рішення розробити схожий додаток, а саме веб-версію месенджеру з сучасним дизайном, використовуючи набір JavaScript-технологій MERN.



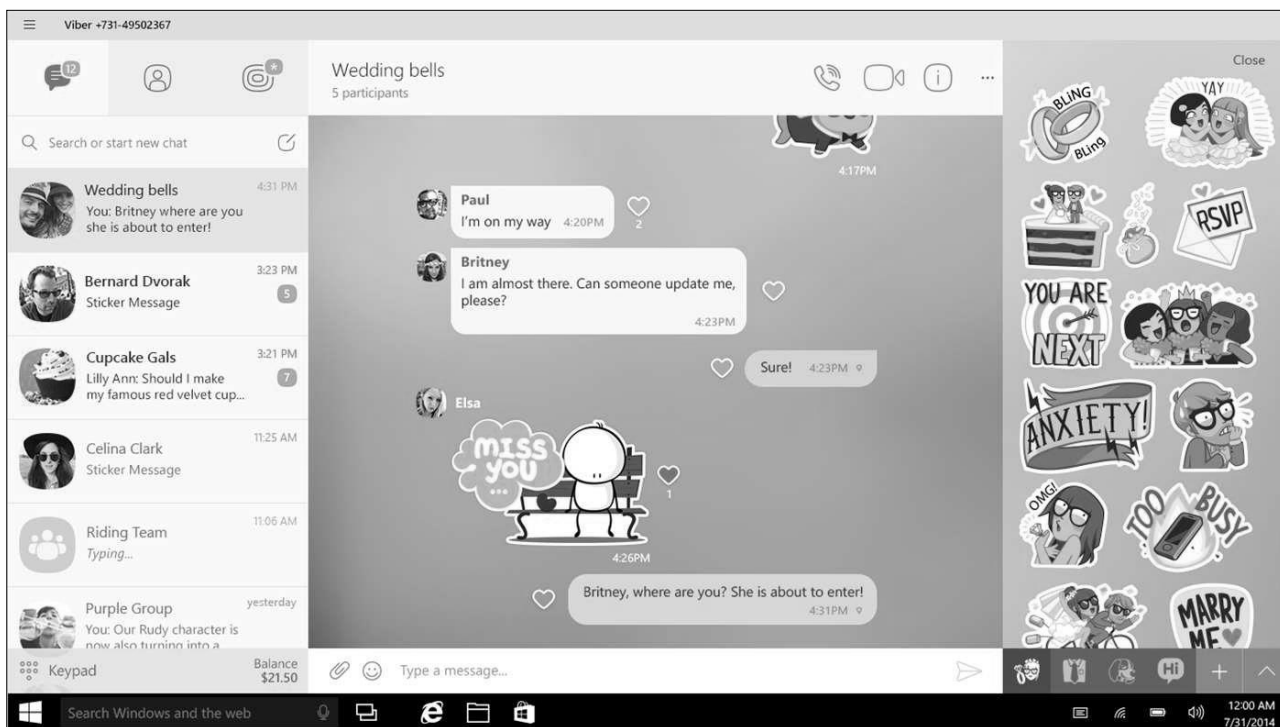


Рис. 3.3. Інтерфейс Viber Desktop

### Висновок до розділу 1

У ході проходження переддипломної практики було проведено аналіз предметної області, розгляд та опрацювання бізнес процесів. Була складена документація для проекту, а також побудовані IDEF0-діаграми для формалізації та опису бізнес процесів.

Також були розглянуті існуючі аналоги програмних продуктів для виявлення найбільш поширеного функціоналу сучасних месенджерів.

## РОЗДІЛ 2. СПЕЦИФІКАЦІЯ ВИМОГ ДО ВЕБ-ЗАСТОСУНКУ «МЕСЕНДЖЕР»

### 2.1 Глосарій

Глосарій – це список понять в специфічній області знання з їх визначеннями. Після аналізу предметної області, було виявлено ряд термінів, які надалі будуть використовуватися під час розроблення веб-застосунку «месенджер».

Таблиця 2.1

Термін	Опис терміну
1. Основні поняття та категорії предметної області та проекту	
Веб-застосунок	Розподілений застосунок, в якому клієнтом виступає браузер, а сервером – вебсервер.
Діалог	Діалог між двома або більше користувачами веб-застосунку «месенджер».
Автентифікація	Процедура встановлення належності користувачеві інформації в системі пред'явленого ним ідентифікатора.
2. Користувачі системи	
Користувач	Будь яка людина, що використовує програмний продукт для одержання необхідної їй інформації.
3. Вхідні та вихідні документи	
Дані для авторизації користувача	Набір унікальних текстових символів (пошта, логін, пароль), які попередньо додані до БД, що надають можливість авторизації у веб-застосунку.
Персональні дані користувача	Дані для авторизації користувача інформаційної системи

Даний глосарій містить ряд термінів та їх визначення, які використовує розробник в ході виконання роботи.

### 2.2 Розроблення варіантів використання

#### 2.2.1 Діаграма варіантів використання

Діаграма варіантів використання описує послідовність дій користувача при використанні програмного продукту для більш детального визначення

функціональності, яка буде реалізована. Основною дійовою особою є користувач.

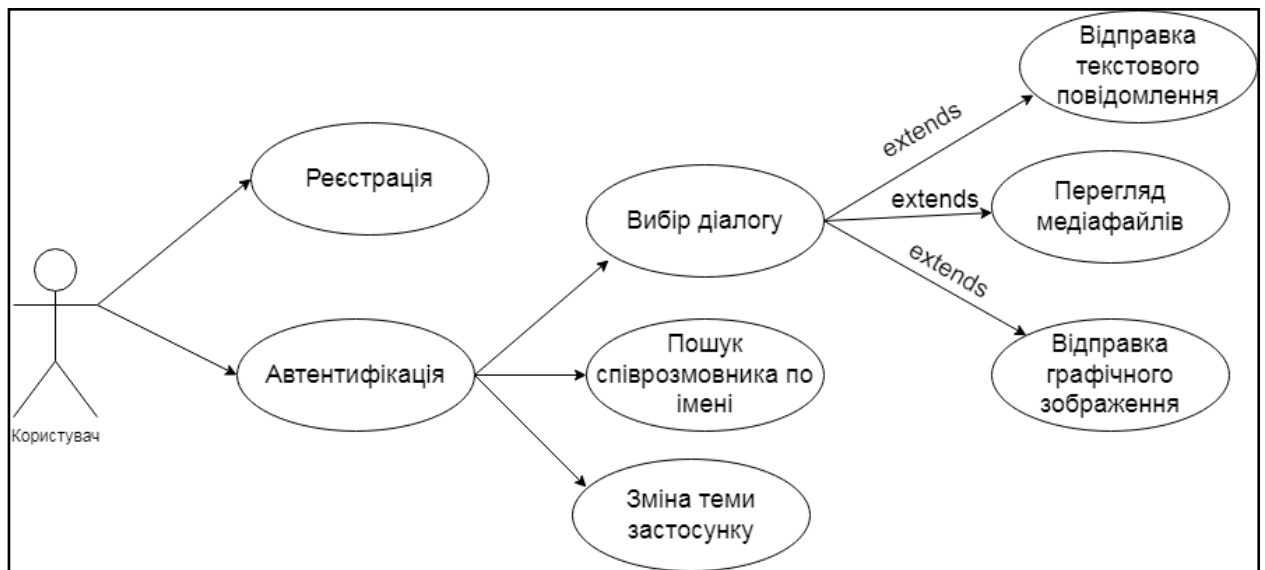


Рис. 2.1. Діаграма варіантів використання

У розроблюваному застосунку передбачені наступні варіанти використання:

- 1) Реєстрація
- 2) Автентифікація
- 3) Вибір діалогу
- 4) Пошук співрозмовника по імені
- 5) Зміна теми застосунку
- 6) Відправка текстового повідомлення
- 7) Перегляд медіа файлів
- 8) Відправка графічного зображення

### 2.2.2 Специфікація варіантів використання

В даному розділі описуються варіанти використання, які реалізують основну функціональність предметної області.

Специфікація варіантів використання веб-застосунку «месенджер» наведена у таблицях 2.2-2.5.

## Варіант використання «Реєстрація»

Таблиця 2.2

Характеристика	Значення
Контекст використання	US-01 Реєстрація нового користувача
Дійові особи	Користувач
Передумова	Користувач не зареєстрований
Тригер	Користувач хоче зареєструватися
Сценарій	<ol style="list-style-type: none"> <li>1. Натиснути на кнопку «Don't have any account»</li> <li>2. Заповнити поля реєстраційної форми</li> </ol>
Постумова	Якщо варіант використання виконався успішно, то буде створено акаунт нового користувача. В іншому разі відобразиться повідомлення про невірні введені дані.

## Варіант використання «Автентифікація»

Таблиця 2.3

Характеристика	Значення
Контекст використання	US-02 Автентифікація користувача
Дійові особи	Користувач
Передумова	Користувач не автентифікований
Тригер	Користувач хоче увійти до свого облікового запису
Сценарій	<ol style="list-style-type: none"> <li>1. Перейти на головну сторінку веб-застосунку</li> <li>2. Заповнити поля форми входу</li> </ol>
Постумова	Якщо варіант використання виконався успішно, користувача буде автентифікована та перенаправлено до списку його діалогів. В іншому разі відобразиться повідомлення про невірно введені дані.

## Варіант використання «Відправка повідомлення»

Таблиця 2.4

Характеристика	Значення
Контекст використання	US-03 Відправка повідомлення
Дійові особи	Користувач №1, Користувач №2
Передумова	Обидва користувача автентифіковані
Тригер	Користувач №1 хоче відправити повідомлення користувачу №2
Сценарій	<ol style="list-style-type: none"> <li>1. Обрати потрібного користувача із списку діалогів</li> <li>2. Набрати текстове повідомлення</li> <li>3. Натиснути кнопку відправити</li> </ol>
Постумова	Якщо варіант використання виконався успішно, користувач №2 отримає звукове сповіщення про надходження нового повідомлення.

## Варіант використання «Відправка повідомлення»

Таблиця 2.5

Характеристика	Значення
Контекст використання	US-04 Зміна теми веб-застосунку
Дійові особи	Користувач
Передумова	Користувач автентифікований
Тригер	Користувач хоче змінити кольорове забарвлення веб-застосунку
Сценарій	<ol style="list-style-type: none"> <li>1. Натиснути кнопку рядом з ім'ям користувача</li> <li>2. Вибрати відповідну тему</li> </ol>
Постумова	Якщо варіант використання виконався успішно, інтерфейс користувача змінить колір на обраний.

### 2.3 Специфікація функціональних вимог

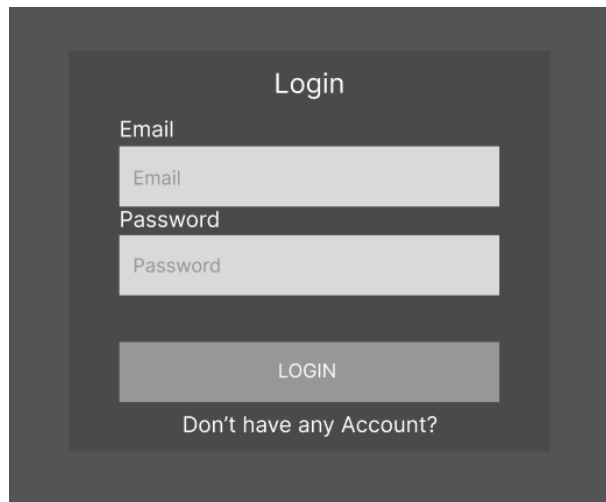
В даному підрозділі описані вимоги для програмного продукту, які описують внутрішню роботу системи та її поведінку. Специфікація функціональних вимог наведена у таблиці 2.6

Таблиця 2.6

Ідентифікатор вимог	Назва вимоги (варіанту використання)	Атрибути вимог	
		Пріоритет	Трудність
FR-01	Реєстрація	Обов'язкове	Висока
FR-02	Автентифікація	Обов'язкове	Висока
FR-03	Перегляд списку доступних користувачів	Обов'язкове	Легка
FR-04	Пошук користувача по імені	Обов'язкове	Легка
FR-05	Відправка текстового повідомлення	Обов'язкове	Середня
FR-06	Відправка графічного зображення	Обов'язкове	Середня
FR-07	Перегляд медіа файлів діалогу	Рекомендоване	Середня
FR-08	Зміна теми месенджеру	Рекомендоване	Легка
FR-09	Звукові сповіщення при отриманні та відправці повідомлень	Рекомендоване	Легка
FR-10	Індикатори статусу користувача	Рекомендоване	Середня

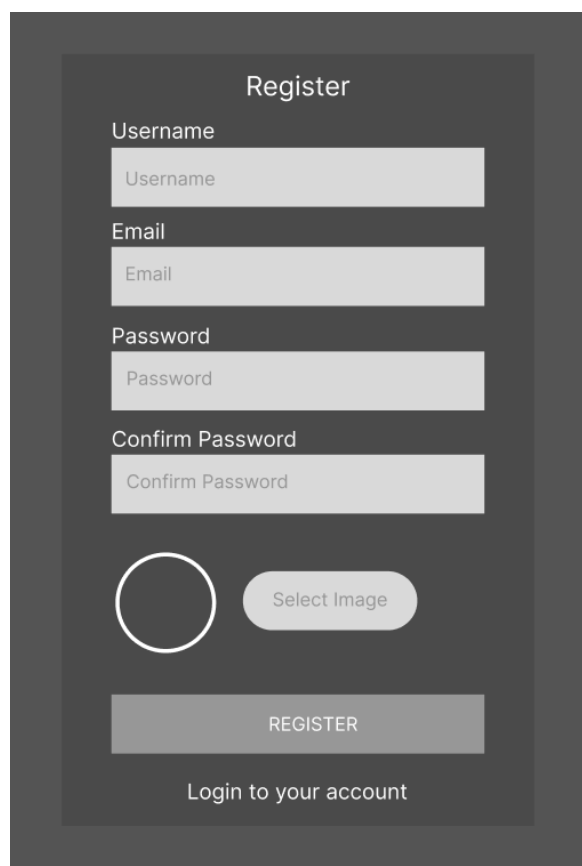
## 2.4 Проектування інтерфейсу користувача

Метою даного підрозділу є опис ходу виконання проектування вайрфреймів для основних елементів користувацького інтерфейсу. Далі наведений приблизний вигляд майбутніх елементів веб-месенджера (рис. 2.2 – 2.4).



The image shows a wireframe for a login form. It is titled "Login" at the top center. Below the title, there are two input fields: "Email" and "Password", each with a corresponding label above it. Below these fields is a "LOGIN" button. At the bottom of the form, there is a link that says "Don't have any Account?".

Рис 2.2. Форма автентифікації користувача



The image shows a wireframe for a register form. It is titled "Register" at the top center. Below the title, there are four input fields: "Username", "Email", "Password", and "Confirm Password", each with a corresponding label above it. Below these fields, there is a circular profile picture placeholder and a "Select Image" button. At the bottom of the form, there is a "REGISTER" button and a link that says "Login to your account".

Рис 2.3. Форма реєстрації користувача

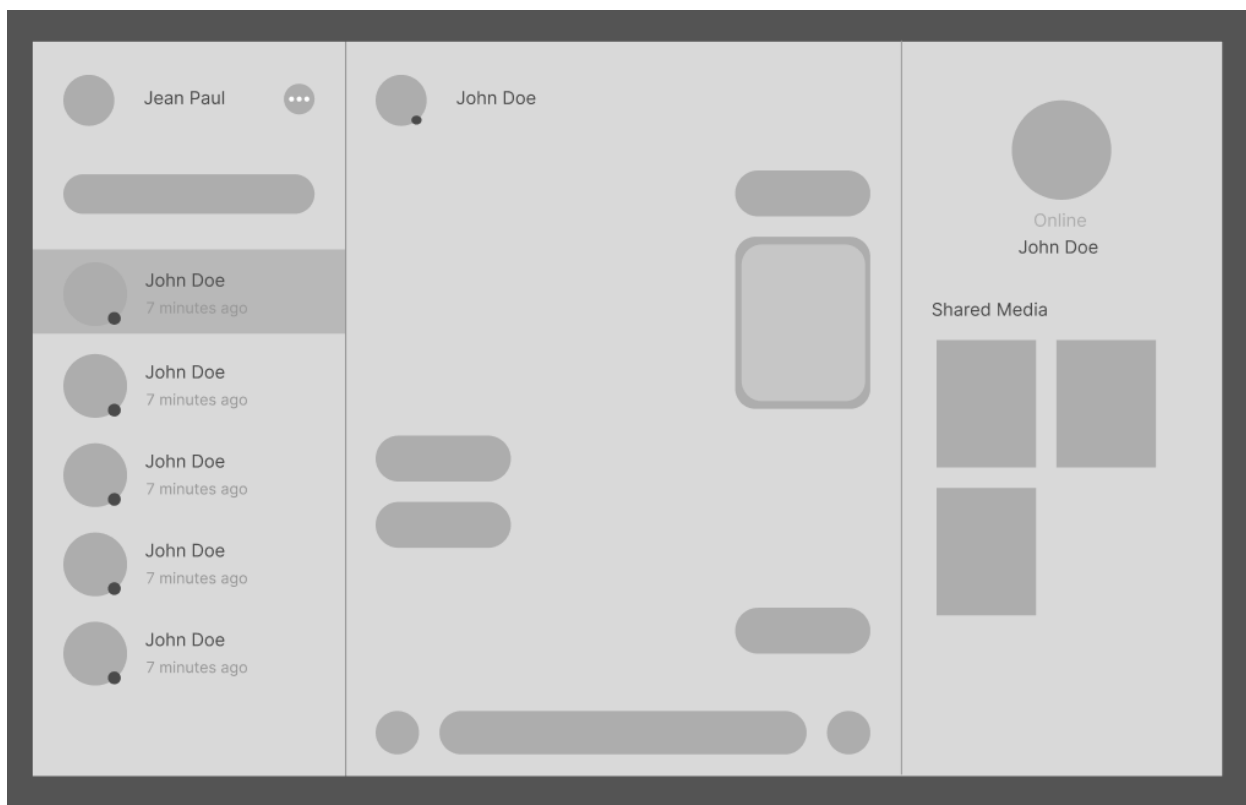


Рис. 2.4. Головний екран месенджера

## Висновки до розділу 2

В даному розділі було розроблено глосарій проекту, діаграма варіантів використання, специфікації варіантів використання, функціональних вимог, а також створено вайрфрейми користувацького інтерфейсу.



## РОЗДІЛ 3. ПРОЄКТНІ ТА ТЕХНІЧНІ РІШЕННЯ

### 3.1 Логічна постановка завдання

Користування веб-застосунком починається з автентифікації. Для доступу до месенджеру користувачу необхідно мати обліковий запис, або зареєструвати новий.

Після успішної авторизації, користувач має доступ безпосередньо самого месенджеру. Він має змогу переглянути інших список інших користувачів додатку, обмінюватись повідомленнями в реальному часі з будь-яким з них. Користувач може відправляти як текстові повідомлення, так і графічні зображення різних форматів (png, jpeg, gif, тощо), а також емодзі. Відправка, як і отримання повідомлень супроводжується звуковим сповіщенням, тож користувач перебуваючи на іншій вкладці браузера буде проінформований про надходження нового повідомлення.

При необхідності користувач може змінити кольорове забарвлення веб-застосунку на більш темний, або навпаки світлий варіант.

### 3.2 Проєктування структури бази даних

#### 3.2.1 Концептуальне інфологічне проєктування

На основі результатів аналізу вхідних та вихідних документів було побудовано модель відображення реквізитів вихідних та вхідних документів на множину елементів даних, що підлягають збереженню у БД, було виконано приведення зібраної інформації до вигляду, зручного для проєктування.

Для зручності проєктування було сформовано словник даних, представлений в таблиці 3.1.

#### Словник даних

Таблиця 3.1

Найменування елемента	Ідентифікатор	Тип	Призначення елемента
1	2	3	4
Електронна пошта	email	String	Електронна пошта користувача
Пароль	password	String	Пароль користувача
Зображення	image	String	Найменування файлу із зображенням користувача
Нікнейм	username	String	Ім'я користувача

Продовження табл. 3.1

Дата створення користувача	created_at	Date	Фактична дата створення користувача
Дата оновлення користувача	updated_at	Date	Фактична дата оновлення користувача
ID користувача	user_id	ObjectId	Унікальний ідентифікатор користувача
Текст повідомлення	message_text	String	Текст відправленого повідомлення
Графічне зображення повідомлення	message_image	String	Назва файлу з графічним зображенням
Статус	message_status	String	Статус повідомлення
Ідентифікатор відправника	sender_id	ObjectId	Унікальний ідентифікатор відправника
Ідентифікатор одержувача	receiver_id	ObjectId	Унікальний ідентифікатор одержувача
Дата створення повідомлення	message.created_at	Date	Фактична дата створення користувача

Всі сутності наведені в таблиці 3.1 було створено на основі базової сутності, саме тому вони мають стандартні поля `_id`, `created_at` та `updated_at`.

Для сутностей розроблюваної системи було створено специфікації обмежень цілісності, а саме – обмеження атрибутів сутностей наведені в таблиці 3.2.

### Обмеження атрибутів сутностей

Таблиця 3.2

Найменування елемента	Допустимі значення	Формат	Умова	Значення за замовчуванням
1	2	3	4	5
Електронна пошта	Набір символів формату «jsmith@domain.com»	Рядковий	Загальні	NOT NULL
Пароль	$6 < n < 20$	Рядковий	Загальні	NOT NULL
Зображення	$0 < n$	Рядковий	Загальні	NOT NULL
Нікнейм	$0 < n$	Рядковий	Загальні	NOT NULL
Дата створення користувача	dd/mm/yyyy	Спеціальний тип «Date»	Загальні	NOT NULL
Дата оновлення користувача	dd/mm/yyyy	Спеціальний тип «Date»	Загальні	NOT NULL
ID користувача	$0 < n$	Спеціальний тип «ObjectId»	Загальні	NOT NULL

Продовження табл. 3.2

Текст повідомлення	0 < n	Рядковий	Загальні	NOT NULL
Графічне зображення повідомлення	0 < n	Рядковий	Загальні	NOT NULL
Статус	0 < n	Рядковий	Загальні	NOT NULL
Ідентифікатор відправника	0 < n	Спеціальний тип «ObjectId»	Загальні	NOT NULL
Ідентифікатор одержувача	0 < n	Спеціальний тип «ObjectId»	Загальні	NOT NULL
Дата створення повідомлення	dd/mm/yyyy	Спеціальний тип «Date»	Загальні	NOT NULL

### 3.2.2 Проектування логічної моделі даних

Логічна модель – це модель даних конкретної предметної області, виражена незалежно від конкретного продукту керування базами даних або технології зберігання, але в термінах структур даних.

Логічна модель предметної області наведена на рисунку 3.1.

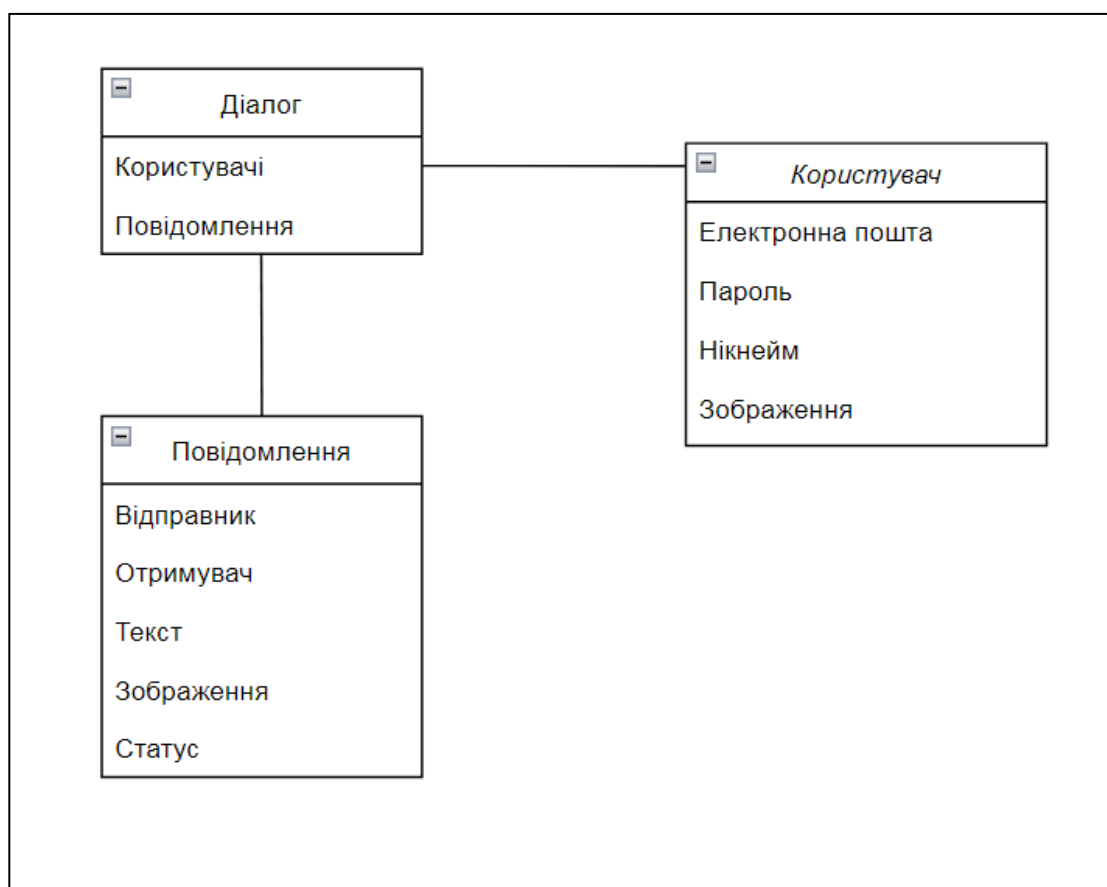


Рис. 3.1. Діаграма «Сутність - Зв'язок» для логічної моделі

### 3.2.3 Проектування фізичної моделі даних

Фізична модель – це подання дизайну даних як реалізованого чи призначеного для реалізації у СУБД. У життєвому циклі проекту вона типово походить від логічної моделі даних, хоча може бути зворотно розроблена з даної реалізації БД.

Фізична модель предметної області зображена на рисунку 3.2.

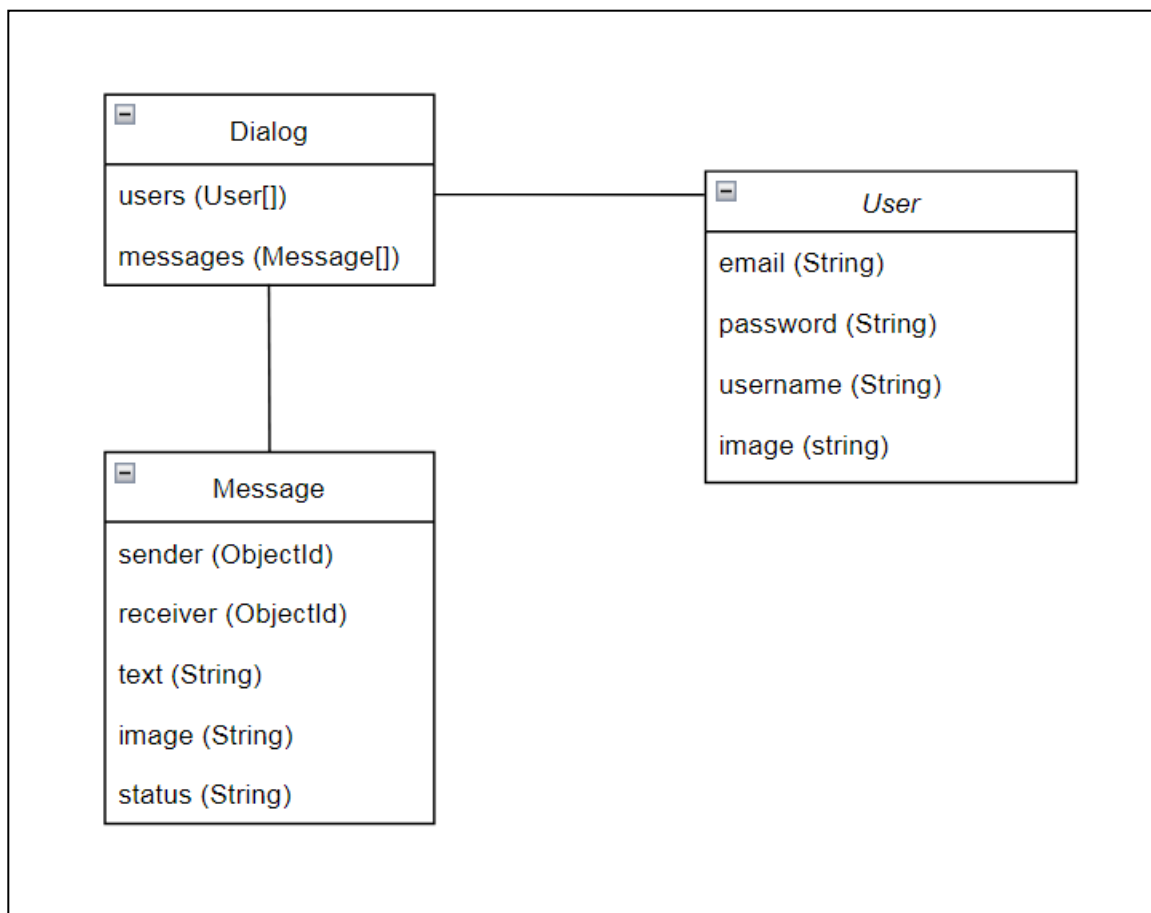


Рис. 3.2. Діаграма «Сутність - Зв'язок» для фізичної моделі

### 3.3 Проектування програмного забезпечення

Для створення веб-застосунку були обрані технології стеку MERN, а саме React – для клієнтської частини, Node.js та Express – для серверної, і MongoDB в якості не реляційної бази даних. Технології цього стеку об'єднують те, що для розробки обох частин додатку (клієнтської та серверної) використовуються одна мова програмування – JavaScript. Цей аспект суттєво полегшує та пришвидшує процес розроблення програмного продукту.

Для реалізації користувачького інтерфейсу була використана бібліотека React, бібліотека з відкритим вихідним кодом розроблена компанією Meta Platforms. Застосовуючи цю бібліотеку можна створювати односторінкові веб-застосунки (SPA). Для даної бібліотеки існує велику кількість написаних модулів, що також впливає на швидкість та якість розробки.

Серверна частина реалізована на Node.js – програмній платформі побудованій на JavaScript-рушієві Chrome V8. Найчастіше ця платформа використовується для побудови таких застосунків як потокові веб-сайти, згадані вище односторінкові сайти та інших веб-додатків. Node.js використовує керовану модель введення-виводу, яка робить її гарним вибором для побудови так званих real-time applications (застосунки, які працюють з великим об'ємом даних у реальному часі), як наприклад месенджер чи стримінговий сервіс.

Фреймворком для побудови безпосередньо веб-серверу виступає Express. Сам фреймворк не нав'язує архітектуру та структуру директорій серверу, однак серед розробників найбільш популярним варіантом є патерн MVC (Model-View-Controller). Цей шаблон передбачає поділ системи на три взаємопов'язані складові: модель, вигляд та модуль керування. Такий підхід застосовується для відокремлення даних від інтерфейсу, для того щоб зміни користувачького інтерфейсу мінімально впливали на роботу з даними, а зміни в моделі даних здійснювалися без змін інтерфейсу користувача.

MongoDB була обрана як СУБД для даного проекту. Це документноорієнтована БД, яка не потребує опису схем таблиць. Була використана хмарна версія цієї СУБД, а саме MongoDB Atlas.

Для детального опису станів системи були створені UML-діаграми станів елементів користувачького інтерфейсу (рис. 3.3 – 3.5).

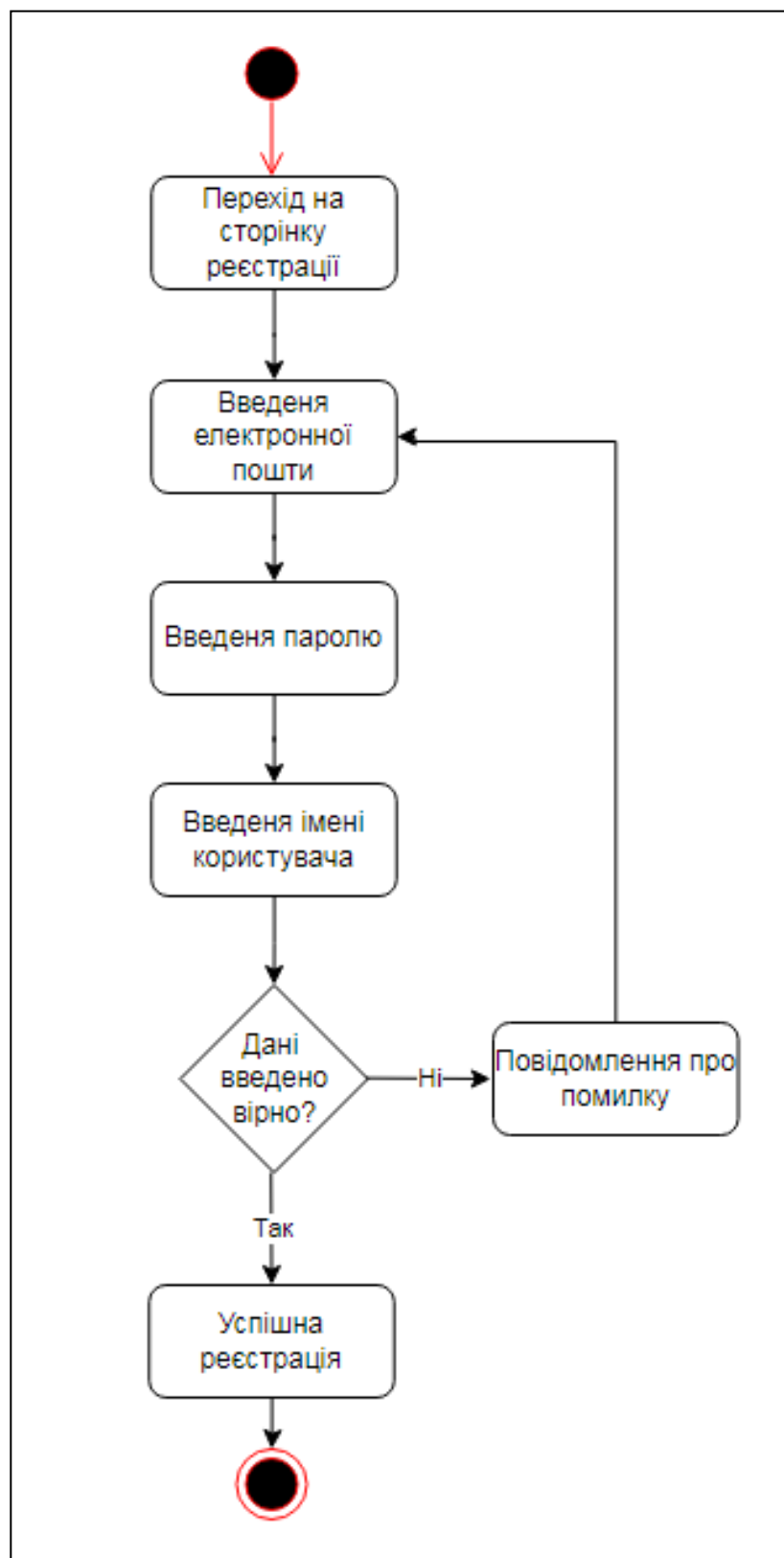


Рис. 3.3. Діаграма станів варіанту використання «Реєстрація користувача»

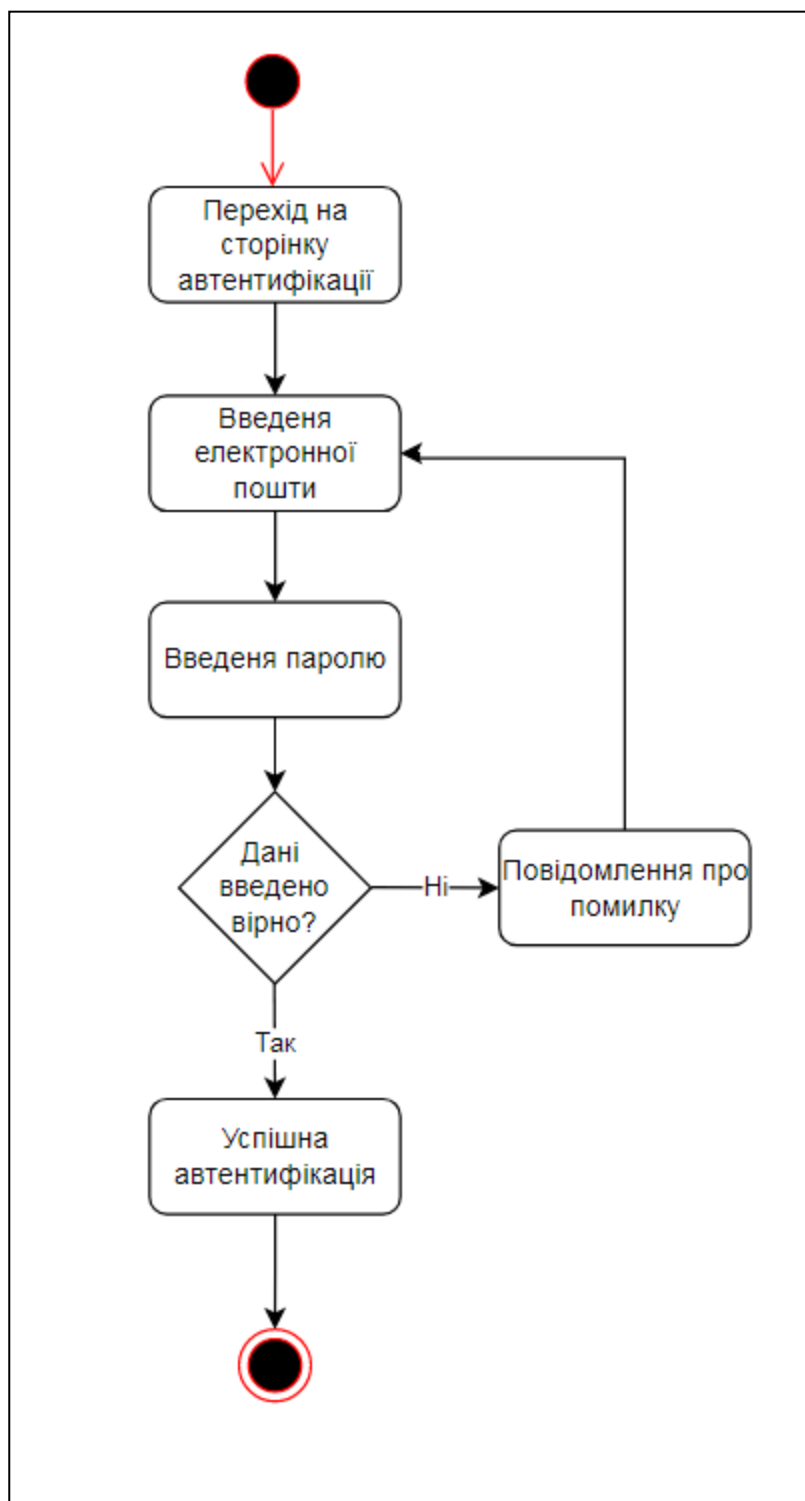


Рис. 3.4. Діаграма станів варіанту використання «Автентифікація користувача»

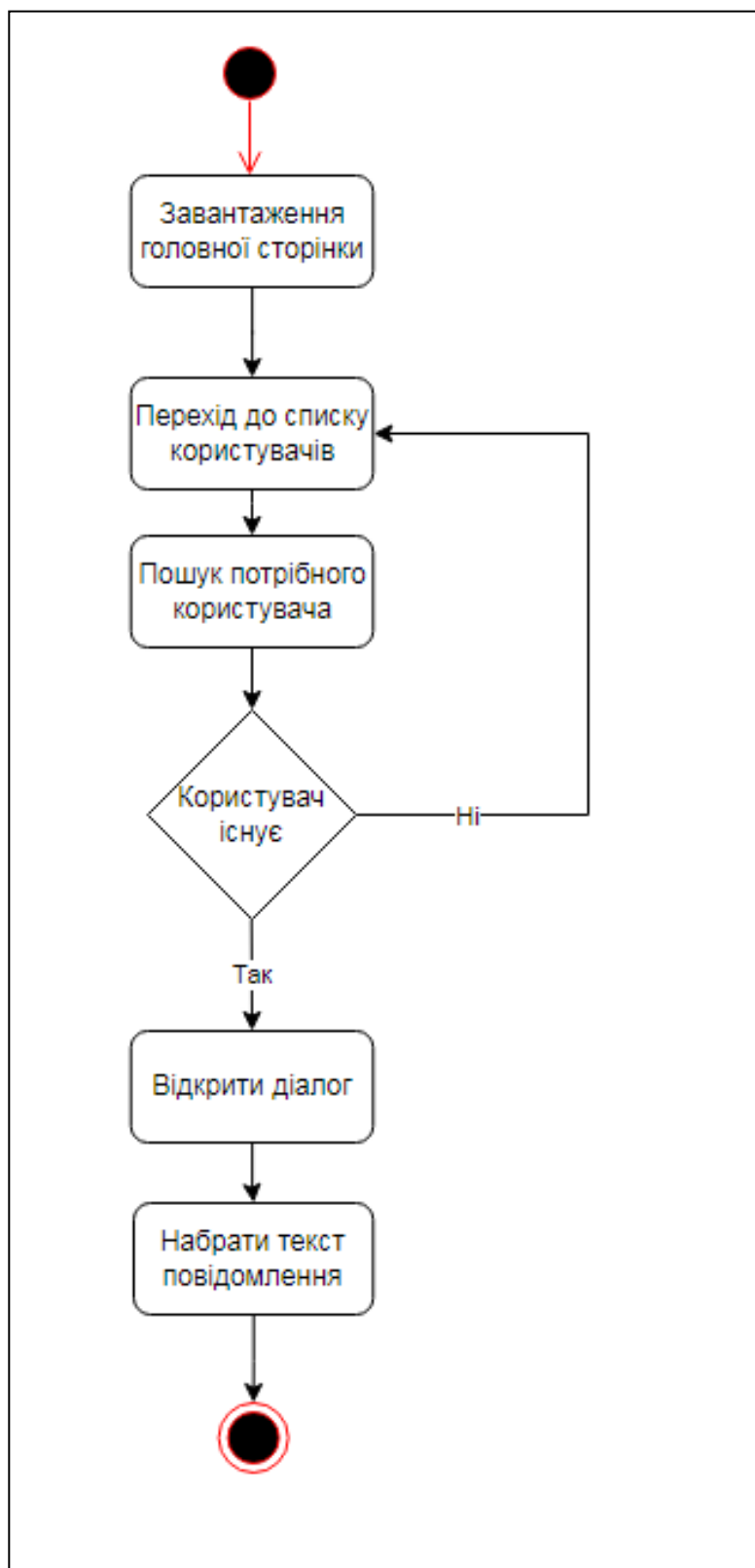


Рис. 3.5. Діаграма станів варіанту використання «Відправка повідомлення»



### 3.4 Тестування програмної системи

Головною метою тестування програмної системи є завчасне виявлення та мінімізація дефектів у кінцевій програмі. Тестування порівнює стан і поведінку продукту з функціональною специфікацією.

Для тестування створений перелік тест-вимог, який охоплює найважливіші функції застосунку (табл. 3.3).

Таблиця 3.3

№ з/п	Тест-вимога
1	Можливість реєстрації користувача
2	Можливість автентифікації користувача
3	Наявність адаптивного веб-дизайну
4	Можливість виходу з облікового запису
5	Можливість пошуку діалогів за ім'ям співрозмовника
6	Можливість відправки текстового повідомлення
7	Можливість відправки графічного зображення
8	Можливість зміни теми веб-застосунку
9	Наявність відображення індикатору набору тексту
10	Наявність відображення індикатору «онлайн» користувача

Далі наведені тест-плани для тест-вимог для більш детального опису тестування функціоналу системи (табл. 3.4 – 3.9).

#### Тестовий приклад 1: Реєстрація користувача

Призначення: перевірка можливості створення нового користувача в системі.

Передумова для тесту: натиснута кнопка «Don't Have Any Account».

Критерії проходження тесту: всі наведені кроки тест-плану виконані успішно.

Таблиця 3.4

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	2	3	4	5
1	Перейти на сторінку автентифікації	На екрані відобразиться форму для автентифікації	На екрані відобразилась форма для автентифікації	Так
2	Натиснути кнопку «Don't Have Any Account»	Форма зміниться на форму реєстрації	Форма змінилась на форму реєстрації	Так
3	Заповнити всі необхідні поля форми	Коректно заповнені поля	Поля реєстрації заповнені згідно формату	Так
4	Натиснути кнопку «REGISTER»	Обліковий запис створено успішно. Перехід на головний екран	Обліковий запис створився успішно. Маємо доступ до діалогів з користувачами месенджеру	Так

### Тестовий приклад 2: Автентифікація користувача

Призначення: перевірка можливості входу до існуючого облікового запису користувача.

Передумова для тесту: користувач на сторінці з формою входу.

Критерії проходження тесту: всі наведені кроки тест-плану виконані успішно.

Таблиця 3.5

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	2	3	4	5
1	Перейти на сторінку автентифікації	На екрані з'явиться форма для входу	На екрані відобразилась форма для автентифікації	Так

Продовження табл. 3.5

2	Заповнити поля «Email» та «Password»	Поля форми заповнені	Поля форми заповнені згідно формату	Так
3	Натиснути кнопку «LOGIN»	Перехід до головної сторінки месенджера	Маємо доступ до діалогів з користувачами месенджера	Так

## Тестовий приклад 3: Пошук діалогу

Призначення: перевірити можливість пошуку діалогу за ім'ям користувача

Передумова для тесту: користувач увійшов до свого облікового запису

Критерії проходження тесту: всі наведені кроки тест-плану виконані успішно.

Таблиця 3.6

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	2	3	4	5
1	Перейти на головну сторінку месенджера	На екрані відобразиться головна сторінка	На екрані головна сторінка веб-застосунку	Так
2	Ввести до поля пошуку ім'я шуканого співрозмовника	Під полем пошуку з'являться діалоги співрозмовників з введеним ім'ям	На екрані відобразився список діалогів шуканих співрозмовників	Так

## Тестовий приклад 4: Відправка текстового повідомлення

Призначення: перевірити можливість відправляти текстові повідомлення в реальному часі

Передумова для тесту: користувач відкрив діалог з обраним співрозмовником

Критерії проходження тесту: всі наведені кроки тест-плану виконані успішно.

Таблиця 3.7

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	2	3	4	5
1	Відкрити діалог з обраним співрозмовником	На екрані відобразиться вікно діалогу	На екрані відображено діалог з відповідним інтерфейсом	Так
2	Ввести текстове повідомлення до відповідного поля	Текст введений у поле повідомлення	Поле заповнено текстом повідомлення	Так
3	Натиснути кнопку для відправки повідомлення	Відправлене повідомлення відобразиться на екрані	Повідомлення з обраним текстом відобразилось у вікні діалогу	Так

Тестовий приклад 5: Відправка графічного зображення

Призначення: перевірити можливість відправляти графічні зображення

Передумова для тесту: користувач відкрив діалог з обраним співрозмовником

Критерії проходження тесту: всі наведені кроки тест-плану виконані успішно.

Таблиця 3.8

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	2	3	4	5
1	Відкрити діалог з обраним співрозмовником	На екрані відобразиться вікно діалогу	На екрані відображено діалог з відповідним інтерфейсом	Так

Продовження табл. 3.8

2	Натиснути кнопку для додавання зображення	Відкриється вікно для вибору графічного зображення	Вікно для вибору медіафайлу відкрилось	Так
3	Натиснути кнопку для відправки повідомлення	Відправлене графічне зображення відобразиться на екрані	Повідомлення з обраним зображенням відобразилось у вікні діалогу	Так

Тестовий приклад 6: Зміна теми веб-застосунку

Призначення: перевірити можливість змінювати тему месенджеру

Передумова для тесту: користувач знаходиться на головній сторінці месенджеру

Критерії проходження тесту: всі наведені кроки тест-плану виконані успішно.

Таблиця 3.9

№ з/п	Крок сценарію	Очікуваний результат	Отриманий результат	Відмітка про проходження кроку сценарію
1	2	3	4	5
1	Натиснути на кнопку з трьома точками поруч з ім'ям користувача	Відкриється вікно для зміни теми та виходу з облікового запису	Відповідне вікно зв'явилося на екрані	Так
2	Натиснути на одну з кнопок зміни теми	Кольорове забарвлення застосунку зміниться на обране	Тема месенджеру змінилась	Так

### 3.5 Розгортання програмного продукту

Месенджер являє собою клієнт-серверний застосунок, а отже поділений на дві частини клієнт та веб-сервер, відповідно. Застосунок може бути запущений на будь-яких операційних системах при наступних умовах.

Вимоги до апаратних засобів:

- 1) Процесор – не нижче Pentium 2 266 МГц
- 2) Вільний дисковий простір – не менше 124 Мб
- 3) Доступний простір ОЗУ – не менше 128 Мб

Вимоги до програмних засобів:

- 1) Операційна система: Windows 7 та вище, MacOS, Linux
- 2) Наявність встановленої платформи Node.js (версії 16 та вище) та пакетного менеджера npm.

Для запуску застосунку необхідно виконати ряд зазначених кроків:

- 1) Перейти до кореневої директорії застосунку
- 2) Встановити необхідні пакети для всіх частин (client, server, socket) застосунку командою `npm install`
- 3) Заповнити `.env` файл на серверній частині
- 4) Запустити клієнтську частину командою – `npm start`
- 5) Запустити серверну частину командою – `npm run server`
- 6) Запустити сокети командою – `npm run socket`
- 7) Ввести в пошукову строчку браузера адресу додатку та зареєструватися

Висновки до розділу 3

В третьому розділі дипломного проєкту було описано логічну постановку задачі, спроектовано логічну та фізичну модель БД, проведено тестування програмного продукту за функціональною специфікацією, а також описано послідовність дій при розгортанні розробленого веб-застосунку.

## ВИСНОВКИ

У результаті виконання дипломного проекту був спроектований веб-застосунок «Месенджер». Даний програмний продукт дозволяє обмінюватись текстовими та графічними повідомленнями в реальному часі з усіма користувачами додатку.

При виконанні завдання було проведено аналіз предметної області «Розроблення месенджера з використанням технологій стеку MERN» та описано його результати. Було проведено характеристику об'єкту управління «ФОП ТРУБІН Д.В.» та створені діаграми організаційної структури. Бізнес процеси були описані за допомогою проектування діаграм IDEF0. Були розглянуті програмні продукти-аналоги для виявлення найбільш важливих елементів функціональності серед застосунків даної предметної області.

Для більш зручного проектування були розроблені специфікації функціональних вимог, діаграми використання, спроектовані вайрфрейми користувацького інтерфейсу та опис архітектури програмного продукту.

Завдяки вищенаведеним етапам проектування, було описано логічну постановку задачі, архітектуру та технічні рішення застосунку, проведено тестування розробленої системи для виявлення дефектів, а також описано процес розгортання програмного продукту..

Підводячи підсумок, можна сказати, що поставлена задача дипломного проекту була виконана успішно. В результаті було створено веб-застосунок «Месенджер» з використанням технологій стеку MERN, що відповідає основним цілям цього дипломного проекту.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ушакова І.О. Проектування інформаційних систем : практикум /І.О. Ушакова. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2015. – 250 с.
2. Методичні рекомендації до виконання кваліфікаційних робіт для студентів спеціальностей 121 "Інженерія програмного забезпечення", 122 Комп'ютерні науки", 126 "Інформаційні системи та технології" першого (бакалаврського) рівня [Електронний ресурс] / укл. Ю.Е. Парфьонов, І.О. Ушакова; ХНЕУ ім. С. Кузнеця. – Харків : ХНЕУ ім. С. Кузнеця, 2020. – 61 с.
3. Методичні рекомендації до оформлення звітів, курсових проєктів та дипломних робіт (проєктів) / Ушакова І.О., Плеханова Г.О., Беседовський О.М : ХНЕУ ім. С. Кузнеця 2021. – 48 с.
4. Ушакова І. О. Проектування інформаційних систем. Практикум. / І. О. Ушакова– Х : Вид. ХНЕУ ім. С. Кузнеця, 2015. – 251 с.
5. IDEF0 – Вікіпедія [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/IDEF0>
6. Introduction to the MERN stack [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/mern-stack>
7. JavaScript-бібліотека React [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/React>
8. Документо-орієнтована база даних MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/MongoDB>
9. Документація платформи Node.js [Електронний ресурс] – Режим доступу до ресурсу: <https://nodejs.org/uk/docs/>
10. Документація бібліотеки React. [Електронний ресурс] – Режим доступу до ресурсу: <https://uk.reactjs.org/>
11. Документація бібліотеки Express [Електронний ресурс] – Режим доступу до ресурсу: <https://expressjs.com/en/guide/routing.html>
12. Документація MongoDB [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mongodb.com/docs/>
13. Документація бібліотеки Socket.IO [Електронний ресурс] – Режим доступу до ресурсу: <https://socket.io/docs/v4/>
14. JavaScript | MDN [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.mozilla.org/ru/docs/Web/JavaScript>
15. JSON Web Token Introduction [Електронний ресурс] – Режим доступу до ресурсу: <https://jwt.io/introduction>



16. Як використовувати JSON Web Token (JWT) для автентифікації  
[Електронний ресурс] – Режим доступу до ресурсу:  
<https://codeguida.com/post/1567>