

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК, ФІЗИКИ ТА  
МАТЕМАТИКИ  
КАФЕДРА ІНФОРМАТИКИ, ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА  
ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

**СТВОРЕННЯ СЕРВІСУ ОБМІНУ ПОВІДОМЛЕННЯМИ  
«GroupMessenger» ФАКУЛЬТЕТУ ФКНФМ ХДУ ДЛЯ  
МОБІЛЬНИХ ПРИСТРОЇВ**

**Кваліфікаційна робота (проект)**

на здобуття ступеня вищої освіти магістр

Виконав: здобувач 2 курсу, 231 М групи  
Спеціальності 122 Комп'ютерні науки  
Освітньо-професійної (наукової)  
програми «Комп'ютерні науки» другого  
(магістерського) рівня вищої освіти  
Кудін Віталій Андрійович  
Керівник доктор педагогічних наук,  
професор Шерман М.І  
Керівник доктор педагогічних наук,  
професор Співаковський О.В.  
Рецензент кандидат педагогічних наук,  
доцент Єрмакова-Черченко Н.О.

## ЗМІСТ

<b>ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ ....</b>	<b>4</b>
<b>ВСТУП.....</b>	<b>5</b>
<b>РОЗДІЛ 1. Дослідження предметної області. Огляд існуючих систем.....</b>	<b>7</b>
1.1. Поява інформаційно-комунікативних технологій в освіті .....	7
1.2. Використання месенджерів для покращення освітніх процесів .....	8
1.3. Огляд існуючих систем комунікації .....	10
1.3.1. Telegram .....	12
1.3.2. Facebook.....	13
1.3.3. Viber .....	14
1.3.4. WhatsApp.....	16
1.4. Функціональне порівняння з аналогами.....	18
1.5. Постановка основної мети кваліфікаційної роботи .....	18
1.6. Висновок до розділу 1 .....	19
<b>РОЗДІЛ 2. Проектування сервісу обміну повідомленнями «GroupMessenger». Визначення основних підходів до розробки.....</b>	<b>21</b>
2.1. Побудова архітектури програми .....	21
2.1.1 Концепція MVC .....	22
2.1.2 Концепція MVP .....	23
2.2. Організація вхідної та вихідної інформації.....	26
2.2.1. Форма як метод збирання інформації.....	26
2.2.2. Форма реєстрації.....	26
2.2.3. Форма входу в систему .....	27
2.2.4. Форма створення чат кімнати.....	28

2.3. Проектування клієнтської підсистеми.....	29
2.3.1. Компонентний підхід .....	29
2.3.2. Впровадження Flux-архітектури в проект.....	29
2.3.3. Ієрархії компонентів проекту .....	31
2.3.4. Вибір технологій для розробки клієнтської підсистеми .....	33
2.4. Проектування серверної підсистеми та бази даних .....	34
2.4.1. REST API сервер.....	34
2.4.2. Моделі та колекції .....	35
2.4.3. Вибір технологій для розробки серверної підсистеми .....	37
2.5. Висновок до розділу 2 .....	39
<b>РОЗДІЛ 3. Розробка програмного продукту та опис його функціонування.....</b>	<b>40</b>
3.1. Розробка інтерфейсу користувача.....	40
3.1.1. Екран «Реєстрація».....	40
3.1.2. Екран «Входу» .....	41
3.1.3. Екран «Чати».....	42
3.1.4. Екран «Чат» .....	43
3.1.5. Екран «Мої викладачі».....	45
3.2. Захист інформації.....	45
3.3. Об'єднання клієнтської та серверної підсистеми.....	48
3.4. Висновок до розділу 3 .....	50
<b>ВИСНОВКИ .....</b>	<b>52</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....</b>	<b>54</b>
<b>ДОДАТОК А.....</b>	<b>58</b>
<b>ДОДАТОК Б.....</b>	<b>60</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

AJAX	від англ. Asynchronous JavaScript and XML
REST	- Representational State Transfer
API	- Application Programming Interface
JWT	- Json Web Token
Фреймворк	- це програмні продукти, які спрощують створення і підтримку технічно складних або навантажених проектів.
БД	- база даних
Серіалізація	- це процес переведення будь-якої структури даних в послідовність бітів.
Десеріалізація	- це відновлення початкового стану структури даних з бітової послідовності.
Flux-архітектура	- це архітектура, яку команда Facebook використовує при роботі з React
ПП	- Програмний Продукт
HTTP	- це протокол передачі даних, що використовується в комп'ютерних мережах.
Інтерфейс	- це засіб зручної взаємодії користувача з інформаційною системою.
СУБД	- система управління базою даних

## ВСТУП

**Актуальність теми.** У зв'язку з реаліями сьогодення, а саме перехід на дистанційне навчання, постає проблема забезпечення студентів та працівників освіти засобами доступу та швидкого обміну інформацією між собою. Саме створення такого сервісу, який буде вирішувати цю проблему, зможе покращити освітній процес, зокрема в режимі дистанційного навчання.

**Зв'язок роботи з науковими програмами, планами, темами:** Основний зміст роботи складають результати досліджень, проектування та розробки, які проводились протягом 2020-2021 років, відповідно до наукового напрямку кафедри інформатики, програмної інженерії, економічної кібернетики за програмою спеціальності комп'ютерні науки.

**Метою наукової роботи** є створення сервісу обміну повідомленнями «GroupMessenger» факультету ФКНФМ ХДУ для мобільних пристроїв. Поставлена мета може бути досягнута вирішенням наступних завдань:

1. Дослідити предметну область об'єкта дослідження;
2. Розглянути існуючі рішення предмета дослідження;
3. Визначити набір програмних інструментів за допомогою яких, буде вестися розробка;
4. Спроекувати архітектуру програмної системи;
5. Розробити на основі отриманих знань сервіс обміну повідомленнями «GroupMessenger» і спроекувати до нього базу даних;

**Об'єктом дослідження** є вплив сучасних програмних засобів обміну інформацією та доступу до неї на покращення освітніх процесів, зокрема під час дистанційного режиму навчання.

**Предметом дослідження** є сучасні програмні засоби швидкого обміну повідомленнями, виявлення їх переваг та недоліків.

**Методи дослідження:** методи емпіричного дослідження (спостереження, порівняння), методи теоретичного дослідження (метод сходження від конкретного, метод формалізації), методи пошуку науково-технічних рішень (метод фундаментального проектування, узагальнений евристичний метод)

**Наукова новизна** полягає в проектуванні та розроблені уніфікованої програмної система обміну повідомленнями навчального призначення, яка підлаштована під особливості навчального процесу в Херсонському державному університеті, зокрема для факультету комп'ютерних наук, фізики та математики. Тепер кожен студент, який зареєструється в цій системі, автоматично попадає в кімнату своєї групи навчання, де йому будуть доступними усі повідомлення, які стосуються виключно його групи. Кожен викладач, який буде зареєстрованим в системі, зможе обрати зі списку груп факультету потрібну йому групу та відправити повідомлення, яке прийде до всіх учасників цієї групи. Також кожен студент, який є учасником групи зможе обрати викладача та створити з ним колективний діалог групи.

**Практичне значення одержаних результатів:** Програмна система моєї кваліфікаційної роботи може бути рекомендована до впровадження в інші вищі заклади освіти, зокрема, такі як університети, інститути, академії зі схожими процесами навчання, а також ознайомить з новими технологічними підходами розробки аналогічних проектів.

**Апробація результатів дослідження:** Результати досліджень, які проводилися у кваліфікаційній роботі було оприлюднено у збірнику IV Міжнародної науково-практичної конференції: «Наука, технології, інновації: світові тенденції та регіональний аспект» у статті «Використання месенджерів для покращення освітніх процесів». Ознайомитись з матеріалами цієї статті можна за посиланням:

«<https://novaosvita.com/wp-content/uploads/2021/10/ScTechInn-Odesa-Sept2021.pdf>».

## РОЗДІЛ 1

### Дослідження предметної області. Огляд існуючих систем

#### 1.1. Поява інформаційно-комунікативних технологій в освіті

Інформаційно-комунікаційні технології (ІКТ) [1] - це сукупність інформаційних засобів (комп'ютери, програмне забезпечення, системи зберігання та аудіовізуальні системи) та технологій зв'язку (системи дротового та бездротового зв'язку), які дозволяють користувачам створювати, отримувати доступ, зберігати, передавати та змінювати інформацію, включаючи мультимедійні файли. Поява інформаційних технологій сприяла швидкому поширенню в практику більшістю освітніх закладів нових підходів сприйняття інформації. Основна вимога, яка з'явилась до інформаційно-комунікаційних технологій в системі освіти, є забезпечення зворотного зв'язку, відповідність обраної методики загальної стратегії проведення заняття і деякі інші.

Централізований підхід до ІКТ у вищій освіті повністю протистоїть дещо сумнівній ролі, яку вони до теперішнього часу грали в школах. ІКТ настільки поширені в адміністративних та науково-дослідницьких університетів, що вони стали сприйматися як належне, і тепер дивно спостерігати той факт, що ІКТ починають руйнувати навчальну функцію вузів. На Всесвітній конференції ЮНЕСКО в 2009 році [2] була досліджена «нова динаміка» вищої освіти і наукових досліджень і виявлений зв'язок зростаючої ролі ІКТ з поширенням відкритого і дистанційного навчання.

ІКТ є рушійною силою, оскільки працівники освіти розуміють, що поєднання цифрових технологій і ресурсів для їх відтворення, дають більше можливостей для розширення горизонтів і поліпшення якості навчання, викладання і підготовки, ніж всі попередні освітні технології,

які беруть свій початок від шкільної дошки до телебачення. Велика частина викладання та навчання є вербальної, будь це слова, числа, формули або зображення. Цифрові навчальні матеріали якісно відрізняються від традиційних навчальних матеріалів своєю можливістю керувати ними. ІКТ є координатором, так як Інтернет - це унікальний засіб для широкого, доступного поширення освітнього матеріалу. Оскільки, Інтернет також став засобом взаємодії, його потенціал для викладання і навчання став ще більш істотним. Найголовніше, що саме освітяни мають величезний вплив на впровадження ІКТ на всіх рівнях освіти. Шкільні системи розуміють, що повинні пристосуватися до цифрового покоління дітей, а студенти університетів вибирають онлайн навчання в більшій кількості, ніж будь-коли раніше.

При комплексному використанні і застосуванні інформаційно-комунікаційних технологій в середовищі дистанційної освіти, зокрема таких програм як месенджери, важливим аспектом є застосування інформаційних та комунікаційних засобів для розробки навчально-методичних матеріалів і комплексів по всіх темах курсу. Створені таким чином навчально-методичні матеріали дозволяють допомогти студентам правильно зрозуміти сутність проблеми і знайти шляхи її вирішення, а не бути тільки засобом передачі навчальної інформації.

## **1.2. Використання месенджерів для покращення освітніх процесів**

Оскільки останні місяці після пандемії ми переважно залишалися вдома, програми для спілкування - чати, голосові записи та відеодзвінки були важливими інструментами для роботи, а також для спілкування з родиною та друзями.

Месенджер - це програма (додаток) для смартфона або персонального комп'ютера, що дозволяє миттєво обмінюватися з іншими людьми, або групами людей текстовими повідомленнями,



мультимедійними файлами, документами, телефонними дзвінками та іншим контентом за допомогою інтернет-зв'язку. Важко знайти людину, яка б не знала що це таке, або в якої у смартфоні чи на комп'ютері не було встановлено жодної такої програми. Ми навіть не замислюємося наскільки часто користуємося месенджерами у своїх повсякденних справах. Але яке відношення вони мають до процесів навчання? Використання месенджерів не тільки в розважальних, але і в навчальних цілях стає новою соціальною практикою і привертає увагу різних вчених. Як вважають багато дослідників в галузі освіти, на перше місце починає виходити інформатизація навчання, яка відбувається зараз і в зв'язку з цим необхідне оновлення педагогічних підходів і понять в педагогіці.

Саме поняття «Освітні комунікації», як вважає А.І. Шутенко (Корольський В.В., Крамаренко Т.Г., Семеріков С.О., Шокалюк С.В.) [3], має входити в нові педагогічні підходи. Багато фахівців негативно ставляться до застосування месенджерів в освітній діяльності, тому що вважають, що традиційні методи, вивчені роками, можуть дати найбільш довершений результат, ніж інноваційні методи. Але це не так.

Найголовніша перевага застосування месенджерів полягає в тому, що стають можливими нові педагогічні стилі і стратегії. Однією з таких стратегій є спільне створення цифрового контенту (Долінер Л.І.) [4]. Подібна робота може проходити як між викладачем і студентами, так і між учнями. Усі необхідні матеріали викладач публікує в групу, розміщує лабораторні завдання та встановлює терміни їх здачі, а це в свою чергу перетворює групу в основний центр перегляду, зберігання та розповсюдження інформації між обмеженим колом людей. Завдяки функціям сповіщення кожен учасник буде одержувати повідомлення про новий допис або публікацію на свої електронні гаджети де встановлена програма, а тому не пропустить нічого важливого. Також є можливість записати ці відеоматеріали та опублікувати їх в групу. Тепер студент,

який не зміг бути присутнім на занятті, зможе переглянути в записі відеоконференції та опрацювати роботу самостійно, а у випадку незрозумілості задати питання безпосередньо викладачеві.

Месенджер - це новітній спосіб комунікації, який ідеально підходить для впровадження в освітні процеси, які ґрунтуються на швидкому опрацюванні інформації. Але попри суттєві переваги, вони мають також і недоліки. По своїй суті, вони містять значну кількість сторонньої інформації, яка ніяким чином не стосується навчальних процесів. До такої інформації можна віднести як приватні діалоги, так і різні публічні групи інтересів (Дадішов Р., Рахматиллаєва Д.) [5]. Також кожний учасник повинен мати електронний пристрій з виходом в інтернет та наявність інтернет зв'язку.

### 1.3. Огляд існуючих систем комунікації

З кожним роком кількість активних користувачів такими програмними продуктами як месенджер невинно зростає. На рисунку 1.1 зображена статистика популярності месенджерів (за даними порталу [statista.com](https://www.statista.com) [6]).

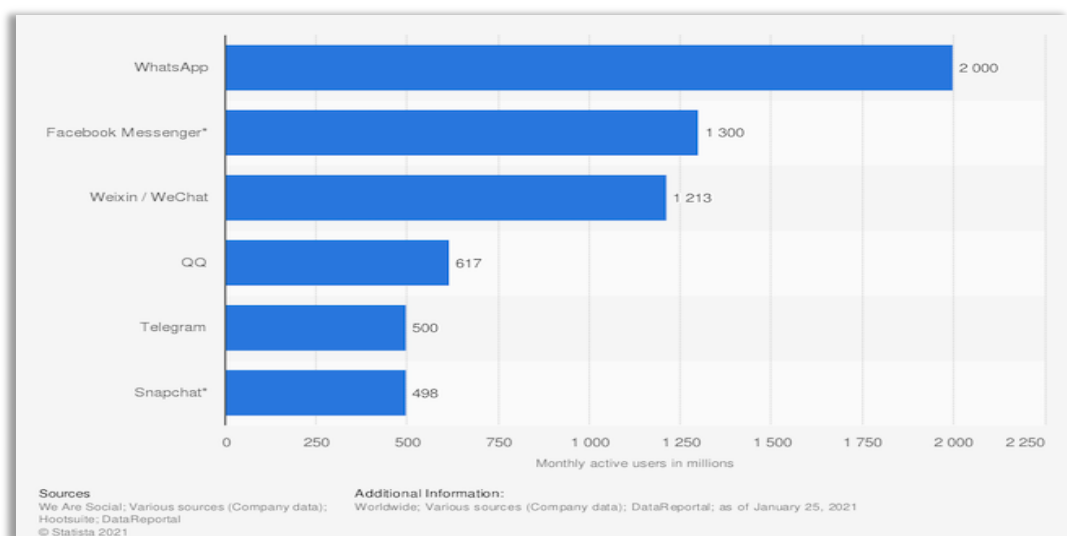


Рисунок 1 – Популярність месенджерів

В анонімному опитуванні, яку проводила редакція AIN.UA [7], було з'ясовано, які месенджери популярні в Україні. Всього в опитуванні взяли участь 1229 осіб. Вони відповідали на задані AIN.UA питання. Нижче ви побачите результати дослідження.

Як показало опитування, найбільш популярним месенджерів в Україні є Telegram [7]. Для спілкування його вибирають 50,6% респондентів. Крім того, 32% опитаних користуються Telegram для робочої комунікації. На другій позиції - Viber. Як основний месенджер вибирають 18,5% опитаних. За ним слідує Facebook Messenger - 8,1%. При цьому, за допомогою WhatsApp і Signal спілкуються тільки 6% і 1,8% респондентів відповідно.

Крім цього в опитуванні були і інші питання, зокрема: «Який месенджер Ви застосовуєте як основний» та «Який месенджер обираєте для роботи». На рисунку 1.2 та рисунку 1.3 зображено результат опитування [7].

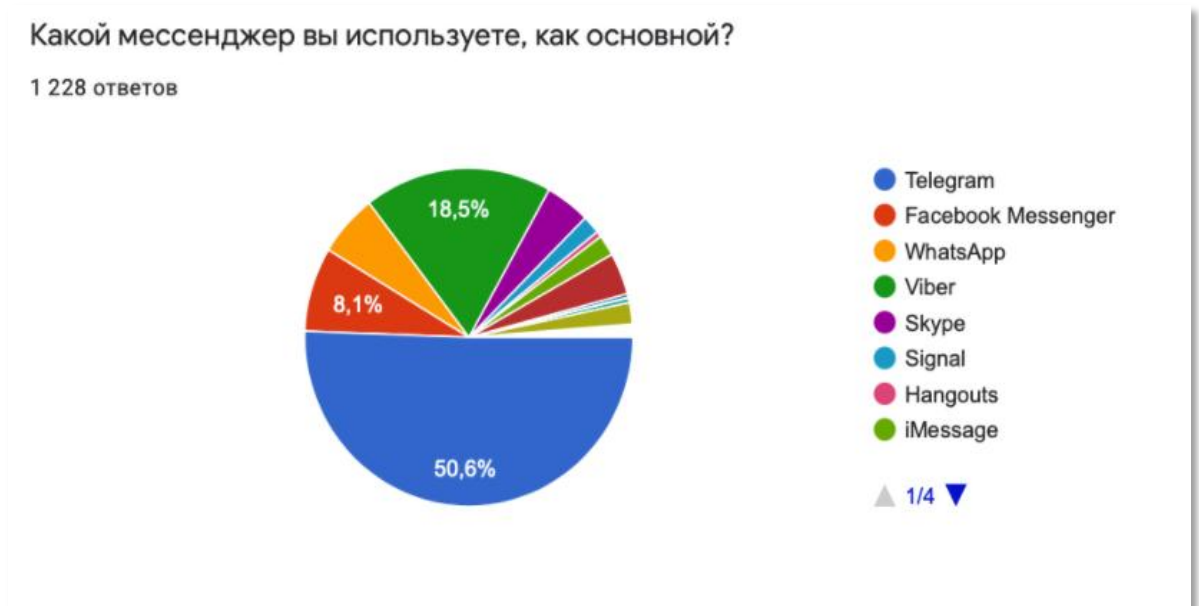


Рисунок 1.2 – Опитування. Питання №1

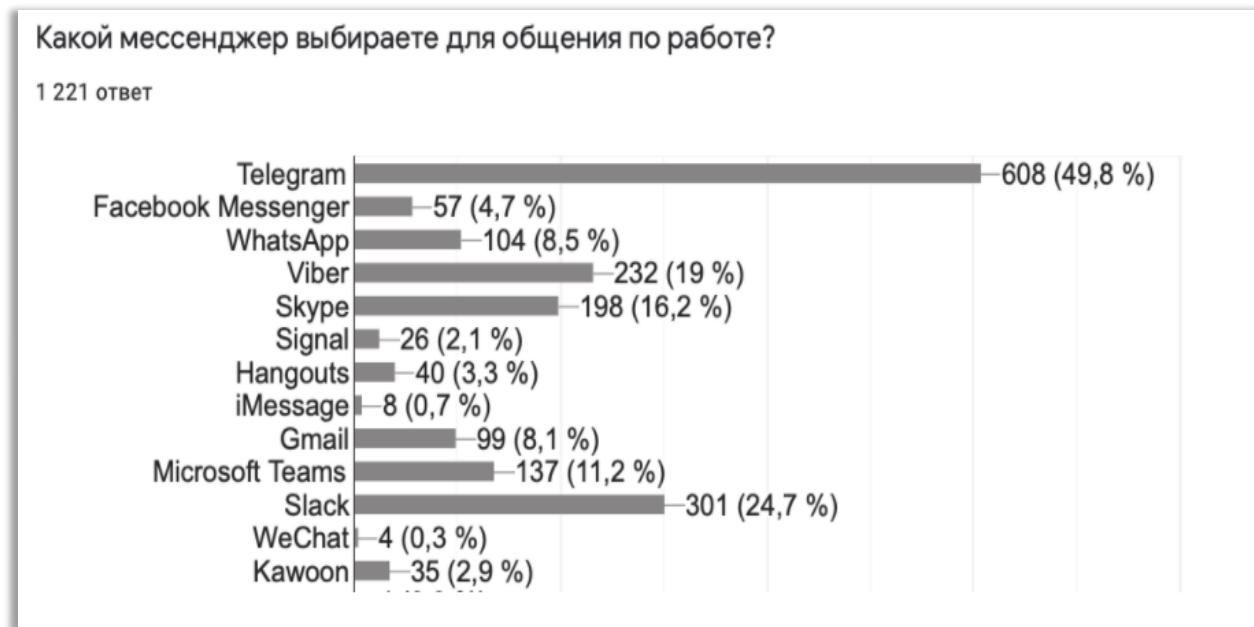


Рисунок 1.3 – Опитування. Питання №2

Як бачимо з результату опитування, найбільш популярними програмами в Україні є Telegram, Facebook, Viber та WhatsApp. Тому надалі огляд існуючих рішень буде проводитись на їх основі.

### 1.3.1. Telegram

Telegram - це безкоштовний месенджер, за допомогою якого можна спілкуватися з іншими користувачами, обмінюватися файлами різних форматів та різних розмірів. Програма розроблена Павлом Дуровим - російським программістом та одним із авторів популярної мережі ВКонтакте [8]. У повідомленнях застосовується спеціальний протокол шифрування MTProto - власна розробка Павла Дурова. Захист застосовується і для передачі повідомлень, а також для авторизації користувача.

Безпеку на високому рівні підтримує також те, що сервери, де зберігаються дані користувача, знаходяться в різних місцях на планеті: Лондоні, Сингапурі та в Сан-Франциско [8]. Кожен день додається близько 600 тисяч учасників. Найбільшу популярність месенджер

отримав в країнах близькосхідного регіону, Південної Америки, Німеччини, Італії, Іспанії. Додаток працює на 13 мовах. Це конкурент таких сервісів, як Viber, WhatsApp і т.п. За допомогою Telegram можна:

- Спілкуватися онлайн з користувачами по всьому світу;
- Пересилати та отримувати файли різних форматів (фото, аудіо, відео та інші);
- Створювати групові чати з кількістю учасників до 200, в тому числі засекречені. Останнє можливе завдяки шифруванню.
- Організувати спілкування в супер групах - до 10 тисяч осіб;
- Використовувати «username» - можливість, яка дозволяє знайти людину, не знаючи його номера телефону;
- Знаходити групи за інтересами і брати участь в їх обговореннях;
- Записати і переслати звукове повідомлення;
- Слухати музику - безкоштовно і навіть без інтернету. Для цього розроблений медіаплеєр з зрозумілим інтерфейсом і великим набором функцій.
- Пересилати, створювати та безкоштовно завантажувати стікери;
- Використовувати автоматизованих помічників – ботів;

### **1.3.2. Facebook**

Facebook почав свою роботу ще у лютому 2004 року як внутрішня соціальна мережа в Гарвардському університеті. Він був створений Марком Цукербергом разом з Едвардом Саверіном, обидва студенти коледжу [9]. Тільки в 2006 році Facebook став доступний для будь-якої людини 13 років і старше і швидко завоював популярність, перевершивши MySpace як найпопулярнішу соціальну мережу в світі. Серед основних можливостей можна виділити наступні:

- Facebook дозволяє заводити список друзів та обирати настройки конфіденційності. Це дає змогу налаштувати хто може бачити контент в вашому профілі;

- Facebook дозволяє завантажувати фотографії і підтримувати фотоальбоми, якими можна ділитися з друзями;
- Facebook підтримує інтерактивний онлайн-чат і можливість коментувати сторінки профілю вашого друга, щоб залишатися на зв'язку;
- Facebook підтримує групові сторінки, фан-сторінки і бізнес-сторінки, які дозволяють компаніям використовувати Facebook в якості засобу для маркетингу в соціальних мережах;
- Мережа розробників Facebook забезпечує розширену функціональність і можливості монетизації;
- Ви можете транслювати відео в прямому ефірі, використовуючи Facebook Live;

Серед основних недоліків можна виділити наступні [10]:

- Можливість підробляти профілі. Відкрити обліковий запис та створити підроблений профіль зможе будь-хто у Facebook;
- Проблеми конфіденційності даних. Facebook надає вам надійні та безпечні налаштування конфіденційності, але у більшості випадків цих налаштувань недостатньо для забезпечення належної безпеки даних;

### **1.3.3. Viber**

Маючи понад мільярд користувачів, Viber є одним з ключових гравців на ринку. Він використовує ваш номер мобільного телефону для ідентифікації у мережі та дозволяє безкоштовно спілкуватися за допомогою VoIP, оминаючи оператора мобільного зв'язку. Viber Out дозволяє здійснювати дзвінки на стаціонарні або мобільні номери, що не належать до Viber, за більш низькими тарифами VoIP. Месенджер доступний для більшості платформ, включаючи iOS, Android, Windows Phone та BlackBerry. У месенджера також є веб версія, яка дозволяє

залишатися на зв'язку, перебуваючи за комп'ютером. Серед основних можливостей можна виділити наступні:

- Безкоштовні голосові дзвінки. Користувач може здійснювати VoIP-дзвінки з Viber на Viber по Wi-Fi і 3G-мережі оператора;
- Відправлення миттєвих безкоштовних текстових повідомлень;
- Відправка файлів різного формату - фото, відео, документи, аудіо;
- Синхронізація девайсів, підключених до облікового запису;
- Вхідний дзвінок в Viber надходить як і звичайний дзвінок, попередньо запускати для цього програму не потрібно. Виклик також можна прийняти або скинути;
- Незважаючи на наявність в додатку цифрової клавіатури, безкоштовні дзвінки можна здійснювати тільки на номери прив'язані до Viber. Якщо зателефонувати на будь-який інший номер, він просто піде через мережу оператора і рахуватися за його тарифами;
- Прийняті і пропущені дзвінки в Viber зберігаються в окремому журналі;
- Наскрізне шифрування - ключі діалогу знаходяться на пристроях користувачів і не передаються третім особам;

Недавно Viber отримав масштабне оновлення. Щоб зробити додаток цікавішим у використанні, Rakuten Viber представив нову функцію для чатів під назвою «Об'єктив Viber» [11]. Він об'єднався з Snap, Inc., творцем Snapchat, щоб додати доповнену реальність (AR) до своєї вбудованої камери. Це дає дійсно хорошу платформу для перетворення камери у повсякденний засіб спілкування. Об'єктиви Viber від Snap вперше запропонували користувачам Viber відеозв'язок та фотографії з підтримкою AR. Старт включає 30 нових об'єктивів, включаючи маски тварин та персонажів, підводний об'єктив та багато іншого. До кінця 2021 року компанія планує додати ще понад 60 додаткових об'єктивів [11]. На підтримку проекту було виділено суму

щонайменше 300 мільйонів. Користувачі також матимуть можливість створити власні ексклюзивні об'єктиви у Viber. ФК Барселона, Всесвітній фонд дикої природи, є одними з перших партнерів Viber, які запропонували нові об'єктиви у додатку для обміну повідомленнями.

#### **1.3.4. WhatsApp**

Брайан Ектон та Ян Кум заснували WhatsApp у 2009 році [12]. Історія запуску почалася, коли вони залишили роботу в Yahoo, щоб подорожувати по всьому світу. Але незабаром їхні заощадження почали закінчуватися, і їм довелося подати заявку на роботу у Facebook, що теж не вдалося за планом. Вони були дуже розчаровані, але ця невдача змусила їх розпочати нову життєву подорож - WhatsApp.

Ян Кум придбав собі смартфон у січні 2009 року та незабаром усвідомив потенціал індустрії мобільних додатків [12]. Він хотів створити додаток, який відображатиме статуси поруч з окремими іменами користувачів. Початок подорожі не був гладким, але вони проклали собі дорогу, як і багато інших успішних підприємців. Куму вдалося розробити додаток для iOS і 24 лютого 2009 року зареєструвати бренд "WhatsApp inc.". Ян назвав його "WhatsApp", тому що він більше нагадував те, що відбувається, що узгоджується з ідеєю статусів.

Детальніше розглянемо можливості програми WhatsApp. У статті: «12 of the best hidden WhatsApp features you need to know (Jason Cipriani) [13]» описуються можливості цього додатку. Серед основних можливостей можна виділити наступні:

- Відформуйте текст, щоб додати деякої активності до ваших повідомлень. WhatsApp дозволяє вам наживо виділяти жирний шрифт, наводити курсивом або додавати закреслення до тексту. Для цього вам потрібно додати пару спеціальних символів до, та після тексту, який потрібно відформувати.



- Позначте повідомлення зірочкою, щоб його було легше знайти пізніше. Позначення зірочкою повідомлення створює зручний спосіб повернутися до значущого повідомлення, номера телефону, адреси чи посилання на статтю, яку надіслав вам друг. Натисніть і утримуйте повідомлення, потім торкніться значка зірки.
- Приховайте свій останній вхід у додаток. Іноді знаючи про те, що люди бачать, коли ви востаннє були в Інтернеті, викликає надмірний тиск, і ви відчуваєте, що вам потрібно відповідати на повідомлення, що очікують на розгляд.

Коли мова йде про повідомлення, WhatsApp приходить в голову першим, адже він один з перших таких програмних продуктів на ринку. Існує ряд переваг і недоліків WhatsApp у порівнянні з конкурентами [14]. Серед переваг можна назвати, зокрема, такі:

- Групові дзвінки: WhatsApp має функцію групових дзвінків. Ви можете легко здійснювати групові дзвінки своїм друзям, родичам чи будь-кому іншому. Це одна з переваг використання WhatsApp для спілкування;
- Підтримка фільтрів: Ви можете налаштувати своє зображення, перш ніж надіслати. Вибравши фотографію на екрані редагування, потрібно провести пальцем угору по цьому екрану, щоб відкрився ящик з фільтрами;
- Без реклами: багато месенджерів та додатків сервісів чату поставляються з рекламою, їх можна уникнути лише у версії Pro, яка є платним варіантом. WhatsApp - це абсолютно безкоштовний додаток для обміну повідомленнями без реклами.

Серед основних недоліків можна виділити наступні [14]:

- Без цензури вмісту: WhatsApp не має цензури вмісту;
- Конфіденційність: Деякі люди можуть додати ваш номер до групи без вашого переднього дозволу.

#### **1.4. Функціональне порівняння з аналогами**

Сервіс обміну повідомленнями «GroupMessenger» - це уніфікований програмний продукт, який функціонально підлаштований під освітні процеси факультету ФКНФМ ХДУ, а саме способом отримання актуальної інформації. Це означає, що в порівнянні з аналогами, не буде можливості використовувати його у розважальних цілях або цілях, які не пов'язаних з навчанням, а саме:

- Створювати приватні чати;
- Додавляти учасників не з групи навчання;
- Видаляти чати с викладачами;
- Впливати на кількість учасників;

Серед функціональних переваг, які стосуються навчальних процесів можна виділити наступні:

- Не потрібно самотужки створювати групу та наповнювати її учасниками;
- Для того щоб створити колективний чат з викладачем достатньо просто вибрати його зі списку. Система сама створить чат-кімнату, згенерує їй назву, додасть туди усіх учасників з групи навчання та викладача;
- Для того щоб створити колективний чат зі студентами, окремої групи навчання, викладачу достатньо просто вибрати їх зі списку груп;
- Усі дані будут зберігатися на окремій бази даних, та не залежати від сховища інших програмних продуктів та від їхньої політики конфіденційності;

#### **1.5. Постановка основної мети кваліфікаційної роботи**

Основна мета кваліфікаційної роботи - це розробити зручний сервіс обміну повідомленнями «GroupMessenger» факультету ФКНФМ ХДУ. Програмна система буде складатися з двох програмних підсистем:

клієнтської підсистеми та серверної підсистеми. Клієнтська підсистема представляє собою мобільних додаток. Додаток буде працювати під мобільні операційні системи. Серверна підсистема представляє собою REST API сервер, який взаємодіє з клієнтською підсистемою за допомогою HTTP-запитів. На основі отриманих знань першого розділу дипломного проекту буде вестися проектування та розробка програмного продукту, більш детально з цими процесами можна ознайомитися у розділі 2: «Проектування сервісу обміну повідомленнями «GroupMessenger». Визначення основних підходів до розробки» та у розділі 3: «Розробка програмного продукту та опис його функціонування»

## **1.6. Висновок до розділу 1**

У розділі 1 «Дослідження предметної області. Огляд існуючих систем» було виконано такі поставлені завдання:

- 1) Дослідити предметну область об'єкта розробки, а саме було:
  - досліджено вплив інформаційно-комунікативних технологій на освіту;
  - досліджено вплив використання сучасних месенджерів для покращення освітніх процесів;
- 2) Розглянути існуючі рішення об'єкта розробки, а саме такі програми, як:
  - Telegram;
  - Facebook;
  - Viber;
  - WhatsApp;
- 3) Сформулювати постановку основної мети кваліфікаційного проекту.

Також було розглянуто появу інформаційно-комунікативних технологій в освіті, їх вплив на освітні процеси та стратегії проведення занять з їх використанням, зокрема під час дистанційного навчання. Усі

завдання, які були поставлені в цьому розділі, виконані успішно та у визначені календарні терміни.

## РОЗДІЛ 2

### Проектування сервісу обміну повідомленнями «GroupMessenger». Визначення основних підходів до розробки

#### 2.1. Побудова архітектури програми

Як це видно з рисунку 2.1, програмна система складається з клієнтської і серверної частини, тим самим реалізуючи мережеву архітектуру «клієнт-сервер».

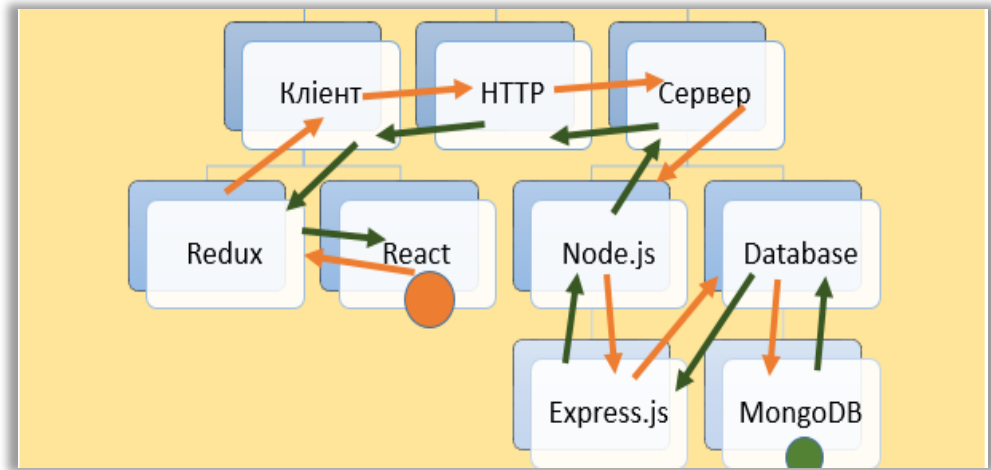


Рисунок 2.1 – «Архітектура веб-додатку»

Клієнтська частина, реалізована у мобільний додаток, формує запити до сервера і обробляє відповіді від нього. Серверна частина приймає запити від користувача, виконує необхідні обчислення, зберігає або видає потрібну інформацію з бази даних і відправляє її клієнту по мережі з використанням протоколу HTTP або HTTPS.

Логіка обробки даних розподілена між сервером і клієнтом, так, наприклад, зберігання даних здійснюється, переважно, на сервері, обмін інформацією відбувається по мережі. Одним з переваг такого підходу є той факт, що клієнту не потрібно мати пристрій з великою швидкістю миттєвих обчислень аби комфортно користуватись додатком. Основою

для реалізації такого підходу у різних програмних додатках став AJAX (від англ. Asynchronous JavaScript and XML) та WebSocket.

Використання AJAX призвело до появи «односторінкових додатків» (англ. SPA або Single page applications), де з сервера веб-сторінка завантажується всього один раз на початку сесії, а вся подальша робота з обміну даними здійснюється у фоновому режимі. Такий підхід дозволяє бути веб-додатком більш інтерактивним і продуктивним. При використанні AJAX клієнт і сервер у фоновому режимі обмінюються даними без перезавантаження сторінки.

Існують два види SPA додатків: додаток, який вбудовує контент декількох сторінок в одну сторінку-оболонку, і додаток, який використовує Ajax-завантаження контенту, коли користувач натискає по посиланню. Таким чином, на сторінці оновлюється тільки та частина контенту, яка змінюється, а інша частина сторінки залишається без змін і перезавантажень.

Таким чином, сервіс обміну повідомленнями «GroupMessenger» реалізовано на архітектурних концепсіях програмування MVC MVP.

### **2.1.1. Концепція MVC**

Model-View-Controller (MVC) - архітектурний підхід, який розділяє додаток на три основні структурні елементи: модель (Model), перегляд (View) та контролер (Controller) [15]. Кожен із цих компонентів створений для обробки конкретних аспектів розробки програми. Нижче наведені структурні елементи MVC:

- **Модель.** Компонент Model відповідає всій логіці даних, з якою працює користувач. Це можуть бути дані, які передаються між компонентами View та Controller, або будь-які інші дані, пов'язані з бізнес-логікою. Наприклад, об'єкт «Банк» буде отримувати інформацію про клієнта з бази даних, обробляти її та оновлювати

дані назад до бази даних або використовувати її для обробки даних.

- Перегляд. Компонент View використовується для всієї логіки інтерфейсу програми. Наприклад, представлення даних «Банк» включатиме всі компоненти інтерфейсу користувача, такі як текстові поля, випадаючі списки тощо, з якими взаємодіє кінцевий користувач.
- Контролер. Компонент Controller діє як інтерфейс між компонентами Model та View для обробки всієї бізнес-логіки та вхідних запитів, маніпулювання даними за допомогою компонента Model та взаємодії з View для відображення кінцевого результату. Наприклад, контролер «Банк» оброблятиме всі взаємодії та вхідні дані з представлення даних клієнта та оновлюватиме базу даних за допомогою моделі клієнта. Цей же контролер буде використовуватися для перегляду даних.

### **2.1.2. Концепція MVP**

Мінімально життєздатний продукт (MVP) - це концепція, яка підкреслює вплив навчання при розробці нових продуктів. Ключовою передумовою ідеї MVP є те, що ви виробляєте фактичний продукт, який можете запропонувати користувачам і спостерігати за їх взаємодією з програмним продуктом (ПП), а потім в процесі випускати нові функції, залежно від вподобань користувачів або необхідності додати нову функціональність. Основна мета MVP - це завжди мінімізувати витрачені час та зусилля, перевіряючи, як ринок реагує на вашу ідею, перед створенням повного продукту, надання голих функціональних можливостей, які задовольняють суттєві потреби користувачів та ефективно вирішують конкретну проблему. Такий підхід має численні переваги, включаючи економічність, мінімальні ризики, чіткість

продукту та інші. Ця концепція використовується як провідний компонент у процесі визначення пріоритетності створення функціоналу вашого ПП, щоб допомогти приймати рішення на основі даних.

У статті: «8 Main Benefits of Building an MVP [16]» вказані такі переваги використання концепції:

- Зосередьтеся на основних функціях. Підхід MVP допомагає знайти ясність та зосередитися на основній функціональності ПП. Це дозволяє перевірити бізнес-концепцію з мінімальними витратами та часом. Більшість власників продуктів схильні до додавання зайвих функцій ще до того, як продукт навіть вийде на ринок. Додаючи величезну кількість функціональних можливостей, дуже легко втратити концентрацію на конкретній проблемі, яку ви прагнете вирішити;
- Ясність бачення. На початковому етапі розробки ПП слід вказати основні функції та цінності програмного забезпечення для клієнтів та все задокументувати. Після створення контрольного списку потрібно поділитися ним із командою. Це першочергове бачення, безумовно, допоможе вам залишатися в курсі і приймати кращі рішення в довгостроковій перспективі;
- Розвиток ранніх відносин з клієнтами. Час відіграє важливу роль у бізнесі. MVP допомагає залучити нових користувачів та зацікавлених сторін на початковому етапі розробки. Перші користувачі розповсюдять інформацію про ваш продукт і дадуть безцінний відгук. Такі дії зроблять ваш продукт на 100 % орієнтованим на користувача;
- Краще розуміння потреб клієнта. Зібрані дані та детальне дослідження цільової аудиторії неможливо переоцінити. Відгуки перших користувачів, які скористалися ПП, набагато цінніші, ніж найкращі припущення бізнес-аналітики та досвідчених радників. Адже, чим раніше клієнт зможе протестувати продукт, тим



ефективніше вийде розробка. Ваші користувачі повідомлять, яку функціональність вони цінують найбільше, яку - найменше та які функції потрібно додати в наступному випуску;

- Очищення інтерфейсу користувача. Підхід MVP запобігає появі у вашого стартового продукту небажаних елементів інтерфейсу. В результаті продукт буде з легкістю сприйматися і не ускладнювати використання;
- Швидший випуск. Рішення розробити лише основну функціональність значно прискорює випуск продукту. Після випуску першої версії продукту ви можете швидко перевірити ключові гіпотези та зібрати відгуки користувачів як про основні, так і про бажані одиниці функціоналу. Зволікання із датою виходу може призвести до розвитку марних сподівань та втрати часу на дорогі виправлення помилок. Завжди знайдеться той, хто може випустити той самий додаток перед вами. Потрібно робити все можливе, щоб ваш початковий випуск був мінімальним, а наступні - поступовими. Це зробить вас більше адаптованими до ринку;
- Гнучкість та постійне оновлення. Наступна перевага застосування MVP - це можливість чудово реагувати на потреби сучасного швидкозмінного ринку. Підхід MVP створює простір для оновлень та нових функцій, які просять клієнти. Крім того, ваш продукт зможе отримати вигоду від випуску новіших технологій та інструментів, коли вони стануть доступними. Це допоможе вам зберегти програмний продукт актуальним на конкурентному ринку.
- Розробка з мінімальними ризиками. Дуже важливо мати на увазі, що великомасштабні програмні продукти займають роки для створення та вимагають багато грошей, часу та зусиль. Усі найпопулярніші ПП починалися з малого та впроваджували широкі можливості з роками. Facebook, Instagram, Snapchat,

Spotify, Airbnb, Uber, Zappos, Dropbox [16] та багато інших були спочатку розроблені як MVP.

## **2.2. Організація вхідної та вихідної інформації**

### **2.2.1. Форма як метод збирання інформації**

Форма є одним з основних елементів взаємодії між клієнською та серверною підсистемами програмної системи. За допомогою форм користувачі «спілкуються» з програмою, роблять цільові дії в ній, авторизуються і реєструються, шукають необхідну інформацію. Також форми є елементами управління, які застосовуються для збору інформації від відвідувачів ресурсів.

Форму можна використовувати для того щоб отримувати відомості. Наприклад, за допомогою форми можна здійснювати збір персональних даних, відомості про замовлення, відомості, які необхідні для виставлення рахунку, способи доставки і так далі. Від відвідувачів потрібно внести різного роду інформацію в поле форми.

Форми складаються з набору текстових полів, кнопок, списків і інших елементів управління, які активізуються клацанням миші. Технічно форми передають дані в структури об'єкта від клієнта до сервера. Сервер переводить їх в JSON та зберігає в базі даних. При проектуванні системи були розроблені такі форми: форма реєстрації користувача, форма входу в систему, форма створення чат кімнати.

### **2.2.2. Форма реєстрації**

За допомогою цієї форми, користувач має змогу увійти в систему та користуватися усіма можливостями, які надаються йому по його ролі у цій системі. Ролей може бути два: студент або викладач. В таблиці 2.1 представлена характеристика отриманої інформації від цієї форми.

Таблиця 2.1

## Форма реєстрації користувача

Поле	Елемент	Тип
Fullname	<input>	String
Email	<input>	String
Password	<input>	String
Роль	<checkbox>	Student   Teacher

**2.2.3. Форма входу в систему**

За допомогою цієї форми, користувач зможе повторно увійти в систему. Для цього знадобиться лише ввести свій Email і пароль, який він указав при реєстрації. Після цього введення інформація відправляється на сервер. Там вона перевіряється і, якщо такий користувач уже був зареєстрований, генерує access токен та відправляє його клієнту. Access токени використовуються для автентифікації на основі переданої інформації, щоб дозволити додаткам отримати доступ до API. Тепер кожен раз, коли користувач буде надсилати запит за даними, сервер буде знати хто це робить та, відповідно, буде відправляти чи не відправляти інформацію. В таблиці 2.2, представлена характеристика отриманої інформації від цієї форми.

Таблиця 2.2

## Форма входу в систему

Поле	Елемент	Тип
Password	<input>	String
Email	<input>	String

**2.2.4. Форма створення чат кімнати**

За допомогою цієї форми, користувач програмної системи зможе створити чат-групу для обміну повідомленнями між усіма учасниками які знаходяться в ній. Якщо користувач це студент, то він обирає між списком викладачів, якщо користувач це викладач, то він обирає між списком груп різних спеціальностей. Після вибору автоматично генерується назва групи із контексту та створюється чат кімната. В таблиці 2.3 представлена характеристика отриманої інформації від цієї форми.

Таблиця 2.3

## Форма створення чат кімнати

Поле	Елемент	Тип
Students	<checkbox[]>	Boolean
Techers	<checkbox[]>	Boolean

## **2.3. Проектування клієнтської підсистеми**

### **2.3.1. Компонентний підхід**

При безпосередній розробці було використано компонентний підхід. Вперше за багато років головною тенденцією в розвитку JavaScript-екосистеми стає стандартизація. Компоненти вступили в гру, і тепер всі грають за їхніми правилами.

Всі елементи системи, від стилю до тестування, - навіть управління станом, перейшли на компонентний підхід. І далі тенденція збережеться і посилиться (Banks A.) [17]. Компонентний підхід - парадигма програмування, істотно спирається на поняття компонента - незалежного модуля програмного коду, призначеного для конструкції. Концепція компонентного підходу у фронтенді відкриває неймовірні можливості для повторного використання коду багато разів. Цей патерн проектування дає можливість повторного використання одного і того ж коду у різних місцях для різних вхідних параметрів, що дозволяє не дублювати фрагменти однакового коду. Для цих цілей ідеально підходить бібліотека React.js, що підтримує компонентний підхід. React.js — JavaScript-бібліотека для створення користувацьких інтерфейсів [18]. Сам React працює на принципах Flux-архітектури.

### **2.3.2. Впровадження Flux-архітектури в проект**

Flux - це архітектура, яку Facebook використовує внутрішньо під час роботи з React в своїх проектах. Це не фреймворк чи бібліотека, це просто новий тип архітектури, що доповнює React та концепцію односпрямованого потоку даних. Flux - це новий спосіб створення складних інтерфейсів користувача, які добре масштабуються. Ця архітектура має 4 головних структурних блоків (Boduch A.) [19]:

- Диспетчер - (Dispatcher);
- Сховище - (Stores);

- Controller - (Views) - (React компонент);
- Дія - (Action) (рис. 2.2);

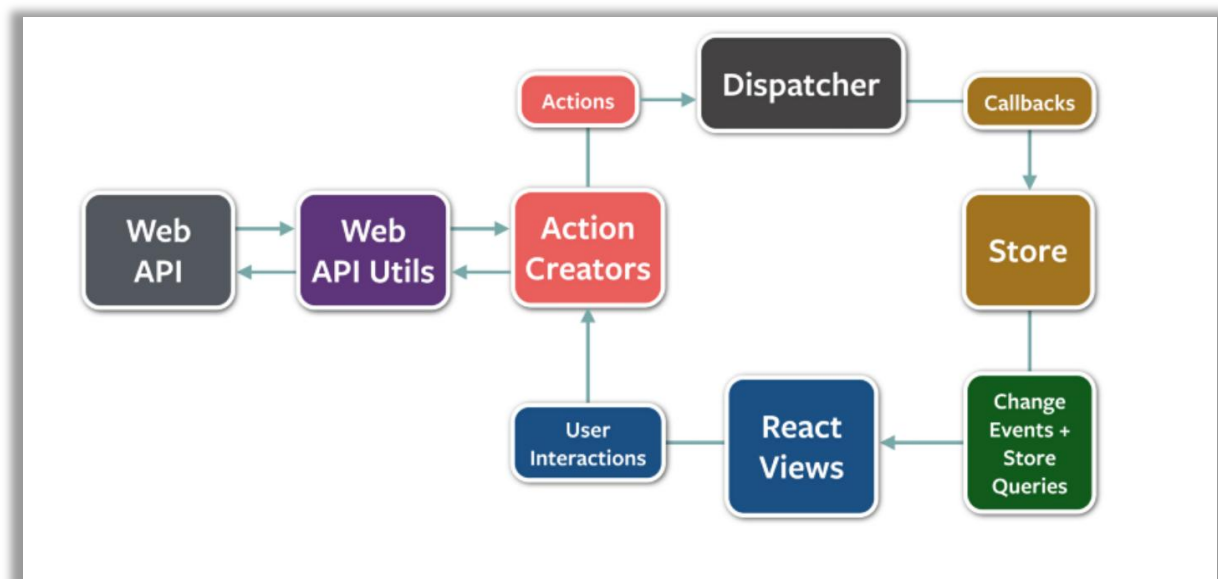


Рисунок 2.2 – «Потік даних Flux»

В таблиці 2.5 описано роль кожного з них у функціонуванні клієнської частини.

Таблиця 2.5

### Елементи Flux-архітектури

Елемент	Роль
Actions	Помічники, які передають дані в Dispatcher
Dispatcher	Отримує ці дії і передає корисне навантаження зареєстрованим callback-ом.
Controller Views	React компоненти захоплюють стан з Stores, а потім передають дочірнім компонентів.

Елемент	Роль
Stores	Діють як контейнери для стану програми та логіки. Реальна робота додатка відбувається в Stores. Stores, зареєстровані для прослуховування дій Dispatcher, будуть відповідно і оновлювати View.

Основна ідея, яка стоїть за використанням Flux-архітектури у проекті - спростити структуру програми. Це полегшує підтримку і розуміння, коли додаток стає більш складним. Оскільки немає ніякої двозначності у взаємини між різними компонентами, робота стає активною.

### 2.3.3. Ієрархії компонентів проекту

Бібліотека React дозволяє розбити частини програми на різні компоненти. Ми можемо їх вбудовувати і побудувати цікаві ієрархії, які формують робочі елементи створюваного проекту. Нижче представлено візуальне уявлення цього процесу для створюваної програмної системи (рис. 2.3).

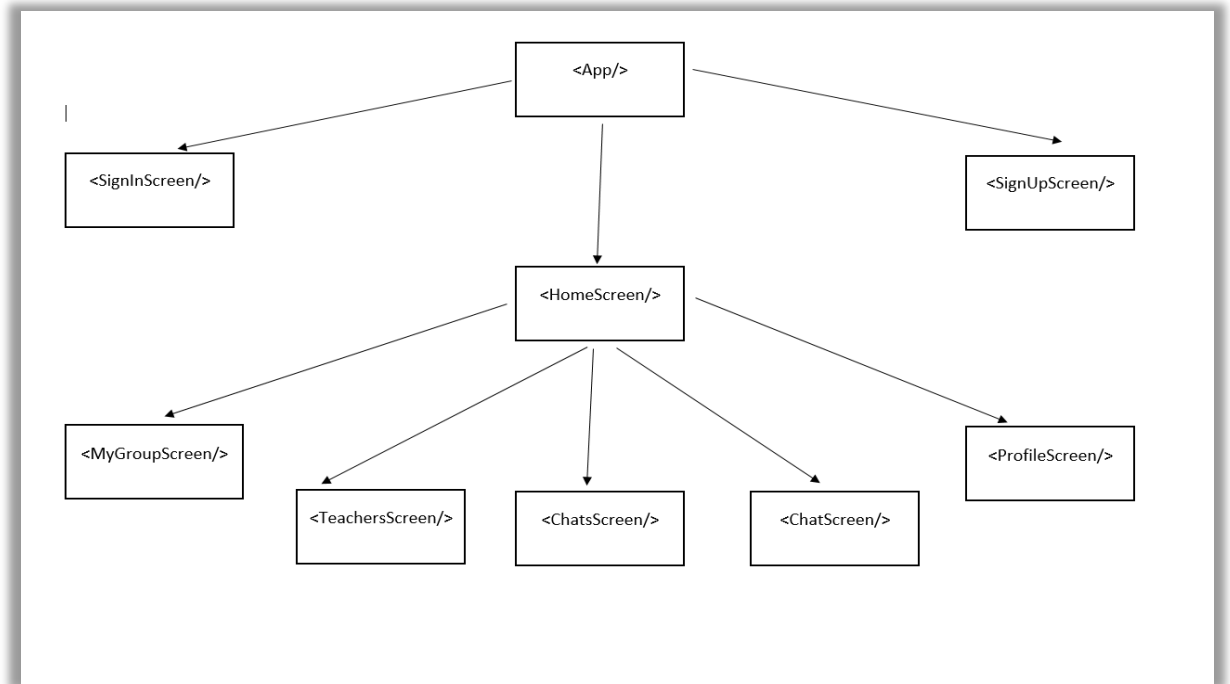


Рисунок 2.3 – «Схема ієрархії компонентів проекту»

Як бачимо зі схеми, структура такої ієрархії досить складна. Тому потрібно більш детально розглянути складові частини, які були використані при проектуванні мобільного додатку «GroupMessenger»:

- <App/> - Головний компонент, який вбудовується у вхідну точку програми. З нього починається рендеринг всього проекту;
- <SignUpScreen/> - Контейнерний компонент, який відповідає за реєстрацію користувача у мобільному додатку;
- <SignInScreen/> - Контейнерний компонент, який відповідає за вхід та вихід користувача в систему;
- <HomeScreen/> - Контейнерний компонент, який відповідає за навігацію користувача по екранам мобільного додатку після проходження аутентифікації користувача у системі;
- <MyGroupScreen/> - Контейнерний компонент, який містить список усіх учасників цієї групи навчання. Кожен студент може перебувати тільки у своїй групі навчання;



- `<TeachersScreen/>` - Контейнерний компонент, який містить список усіх викладачів, які зареєстровані у програмній системі;
- `<ProfileScreen/>` - Контейнерний компонент, який містить відомості про конкретного студента або викладача;
- `<ChatsScreen/>` - Контейнерний компонент, який відповідає за відображення усіх чат кімнат у яких перебуває користувач;
- `<ChatScreen/>` - Контейнерний компонент, який відповідає за показ усієї інформації, яка міститься у чат кімнаті конкретної групи.

#### **2.3.4. Вибір технологій для розробки клієнтської підсистеми**

Для розробки клієнтської частини сервісу обміну повідомленнями «GroupMessenger», який представляє собою мобільний додаток, було використано популярну бібліотеку створення мобільних інтерфейсів React Native, яка працює разом з бібліотекою управління станом Redux та фреймворк для взаємодії з пристроєм Expo.

React Native з'явився в 2015 році як продовження React.js для мобільних пристроїв. Спочатку цей фреймворк був тільки для iOS, через що довгий час був трохи краще оптимізований для цієї операційної системи. React Native - це open-source розробка компанії Facebook <sup>20</sup>. Основна ідея цього рішення - отрисовка UI нативними елементами платформ, а також управління і бізнес-логіка на стороні JS. Трохи детальніше про архітектуру: в основі лежить взаємодія JS з нативним UI і API за допомогою моста (bridge). JS частина працює на архітектурі React, що робить розробку React.js і React Native дуже схожою. Основний мінус, що всі дані проходять конвертацію. Також для побудови інтерфейсів використовується бібліотека Yoga - розробка Facebook, яка спрощує взаємодію з UI. З версії 0.60 RN почав використовувати свій JS engine (Hermes) для Android, з 0.64 також для iOS. З 2018 роки команда React Native працює над новою архітектурою,

яка прибере прошарок-міст, але за станом на квітень 2021 року немає дати релізу.

Ехро є як фреймворком, так і платформою, яка спрощує створення та розгортання мобільних додатків, розроблених за допомогою React Native. Ехро є домом для багатьох корисних інструментів та рідних бібліотек для React Native. Тож це інструмент, який дозволяє легко та швидко розпочати свій проект. Перевага Ехро в тому, що його дуже просто налаштувати, і це значно спрощує життя розробникам на всіх етапах проекту:

- Розробка: ви просто пишете код, Ехро піклується про створення додатків.
- Тестування: Ехро дозволяє протестувати та перевірити свої програми, не запускаючи їх на пристрої. Просто встановіть програму Ехро Go та відскануйте QR -код, щоб протестувати додаток, який розробляється.
- Публікація та оновлення: Ехро спрощує публікацію програм на платформах Apple та Google, а також дозволяє їх оновлювати без проводів.

## **2.4. Проектування серверної підсистеми та бази даних**

### **2.4.1. REST API сервер**

REST (скор. від англ. Representational State Transfer- «передача стану уявлення») - архітектурний стиль взаємодії компонентів розподіленого додатка в мережі. REST є узгоджений набір обмежень, що враховуються при проектуванні розподіленої гіпермедіа-системи. У певних випадках (інтернет-магазини, пошукові системи, інші системи, засновані на даних) це призводить до підвищення продуктивності і спрощення архітектури.

Існує п'ять обов'язкових обмежень для побудови розподілених REST API додатків: модель «клієнт-сервер», відсутність стану,

кешування, одноманітність інтерфейсу, проміжні вузли (Masse M.) [21]. Виконання цих обмежувальних вимог є обов'язковим для REST-систем.

Ці обмеження визначають роботу сервера в тому, як він може обробляти і відповідати на запити клієнтів. Діючи в рамках цих обмежень, система набуває такі бажані властивості як продуктивність, масштабованість, простота, здатність до змін, переносимість, відстеження та надійність.

#### **2.4.2. Моделі та колекції**

Ключовою проблемою моделювання даних є збалансування потреб програми, характеристики продуктивності механізму баз даних та шаблонів пошуку даних. При розробці моделей даних завжди враховується використання даних додатком (тобто запити, оновлення та обробку даних), а також властива їм структура. На відміну від баз даних SQL, де перед вставленням даних потрібно визначити та запустити схему таблиці, колекції MongoDB за замовчуванням не вимагають, щоб їх моделі мали однакову схему. Це означає, що:

- Моделі в одній колекції не повинні мати однаковий набір полів, і тип даних для поля може відрізнитися між моделями в колекції;
- Щоб змінити структуру моделі у колекції, наприклад, додати нові поля, видалити наявні поля або змінити значення полів на новий тип, потрібно оновити моделі до нової структури;

Ця гнучкість полегшує відображення моделей на сутності або об'єкті. Кожна модель може відповідати полям даних представленої сутності, навіть якщо модель має суттєві відмінності від інших моделей у колекції. Ключове рішення при розробці моделей даних для додатків MongoDB обертається навколо структури документів і того, як програма представляє відносини між даними. MongoDB дозволяє вбудовувати пов'язані дані в один документ. При проектуванні бази даних було

визначено такі моделі: Users, Group, Chat, Message. Характеристики кожної моделі наведені у таблиці 2.6

Таблиця 2.6

## Структура моделей даних

Модель	Структура	
	Поле	Тип
Users	id	string
	image	file
	fullname	string
	chatsId	string[]
	login	string
	password	string
	groupId	string
	role	string
Group	id	string
	name	string
	participants	string[]
Chat	id	string
	groupId,	string
	teacherId	string
	chatName	string
Message	id	String
	chatId	String
	authorId	String
	contentType	String
	content	File
	unread	boolead

Вбудовані документи фіксують зв'язки між даними, зберігаючи пов'язані дані в єдиній структурі документа. Документи MongoDB дають можливість вбудовувати структури документів у поле чи масив у документі. Ці денормалізовані моделі даних дозволяють програмам отримувати та обробляти пов'язані дані в одній операції з базою даних.

### **2.4.3. Вибір технологій для розробки серверної підсистеми**

Серверна частина була побудована за допомогою фреймворку Express. Express - це мінімалістичний і гнучкий веб-фреймворк для додатків Node.js, що надає великий набір функцій для мобільних і веб-додатків. Express використовується для розробки додатків досить давно і завдяки своїй стабільності міцно займає позицію одного з найпопулярніших фреймворків Node.js. Для цього фреймворка існує велика кількість докладних інструкцій та описів, які складені розробниками, які перевірили його ефективність на практиці. Тому саме з Express рекомендується починати роботу, якщо ви маєте намір навчитися створювати додатки на платформі Node.js.

Основна особливість цього фреймворка полягає в тому, що для Express характерний невеликий обсяг базового функціоналу. Всі інші необхідні вам функції потрібно буде добирати за рахунок зовнішніх модулів. По суті, Express в чистому вигляді - це сервер і у нього може не бути жодного модуля. Завдяки такому мінімалізму, розробник спочатку отримує в своє розпорядження легкий і швидкий інструмент, який він може розширювати і розвивати.

Для того, щоб користувачі швидко отримували свої повідомлення, без затримок та очікувань, у програмній системі використовується технологія WebSockets. WebSockets дозволяють розробникам вбудовувати функціональні можливості в реальному часі у свої програми, дозволяючи надсилати невеликі шматки даних через одне

постійне з'єднання в обох напрямках. WebSockets - це протокол, що забезпечує повний дуплексний зв'язок, що означає, що клієнт і сервер залишаються на зв'язку через одне з'єднання TCP. На відміну від зв'язку запит-відповідь, з'єднання тут не закривається. Оскільки це дуплекс, сервер також може надсилати дані клієнту без запиту. Це допомагає заощадити пропускну здатність та час відгуку.

В якості бази даних для проекту було обрано документо-орієнтовану базу даних MongoDB. MongoDB реалізує новий підхід до побудови баз даних, де немає таблиць, схем, запитів SQL, зовнішніх ключів і багатьох інших речей, які притаманні об'єктно-реляційним базам даних. MongoDB призначена для використання в якості бази даних загального призначення, тому крім створення, читання, оновлення та видалення даних вона надає постійно зростаючий список унікальних функцій (Меджуї М., Уайлд Е., Мітра Р., Амундсен М.) [22]:

- Індексування. MongoDB підтримує загальні вторинні індекси, дозволяючи виконувати різні швидкі запити, а також надає унікальні, складові, геопросторові і повнотекстові можливості індексації;
- Агрегування. MongoDB підтримує «конвеєр агрегації», який дозволяє створювати складні агрегації з простих частин і дозволяє базі даних оптимізувати їх;
- Спеціальні типи колекцій. MongoDB підтримує колекції часу для даних, термін дії яких повинен закінчитися в певний час, таких як сесии. Він також підтримує колекції фіксованого розміру, які корисні для зберігання останніх даних, таких як журнали;
- Файлове сховище. MongoDB підтримує простий у використанні протокол для зберігання великих файлів і метаданих файлів.

Система управління базою даних (СУБД) MongoDB набуває широкої популярності та застосовується у різних сферах діяльності. Вона використовується для зберігання подій у системі (логировання),

запису інформації з датчиків моніторингу підприємств, а також у сфері електронної комерції та у мобільних додатків. Часто MongoDB застосовується як сховище даних у сфері машинного навчання та штучного інтелекту. MongoDB відноситься до класу NoSQL СУБД і працює з документами, а не з записами.

Це кроссплатформенний продукт, який легко входить у будь-якому операційну систему. Ряд унікальних особливостей дозволяє використовувати СУБД під визначені завдання, в яких вона забезпечить максимальну продуктивність та надійність.

## **2.5. Висновок до розділу 2**

У розділі 2 «Проектування сервісу обміну повідомленнями «GroupMessenger». Визначення основних підходів до розробки» було виконано такі поставлені завдання:

1. Спроекувати архітектуру програмної системи, а саме було:
  - Спроековано клієнтську підсистему програмної системи обміну повідомленнями «GroupMessenger», яка представляє собою мобільний додаток;
  - Спроековано серверну підсистему програмної системи обміну повідомленнями «GroupMessenger», яка представляє собою REST API сервер;
  - Спроековано базу даних, яка представляє собою моделі та колекції даних;
2. Визначити набір програмних інструментів за допомогою яких, буде вестися розробка. Внаслідок цього, було вирішено розробляти клієнтську підсистему, за допомогою сучасної та продуктивної бібліотеки розробки інтерфейсів під мобільні додатки React Native, разом з фреймворком Expo. Серверна підсистема буде розроблюватися за допомогою бібліотеки

створення REST API серверів Express. Вибір бази даних зупинився на документо-орієнтованій базі даних MongoDB.

Всі завдання, які були поставлені у цьому розділі, виконані успішно та у визначені терміни.

## РОЗДІЛ 3

### Розробка програмного продукту та опис його функціонування

#### 3.1. Розробка інтерфейсу користувача

##### 3.1.1. Екран «Реєстрація»

Перше що побачить користувач при запуску мобільного додатку на своєму пристрої це екран реєстрації у системі (рис. 3.1).

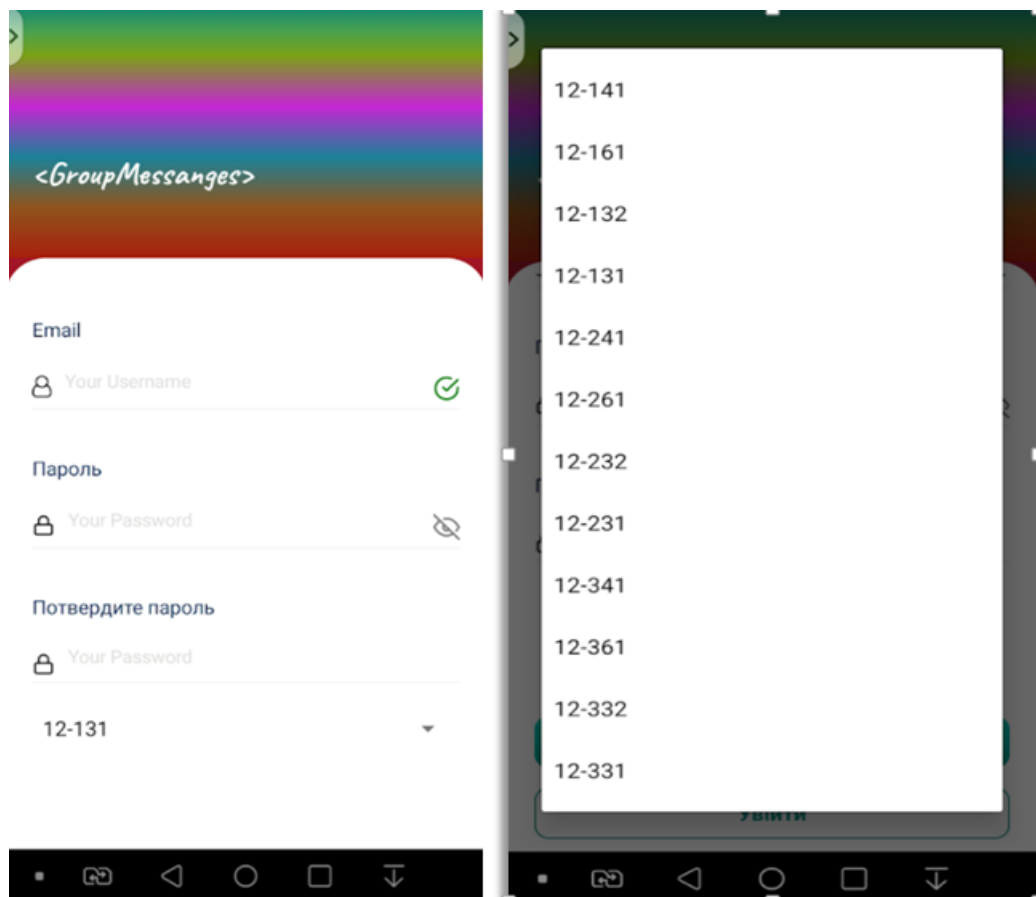


Рисунок 3.1 – «Екран реєстрації»



Щоб створити аккаунт у системі користувачу потрібно буде ввести свої дані, а саме повністю ім'я та прізвище, пароль, електронну пошту, вибрати статус викладача або студента, і у випадку, якщо це студент, вибрати групу навчання та підтвердити пароль. Якщо користувач буде вводити некоректно інформацію форма буде показувати помилки для того щоб користувач виправив їх (рис. 3.2).

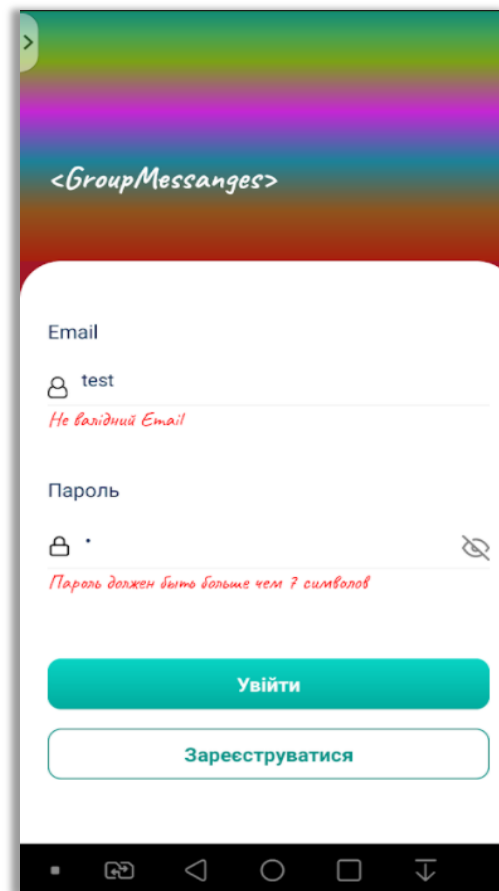


Рисунок 3.2 – «Валідація форми»

### 3.1.2. Екран «Входу»

Після того як користувач вийде із системи (або закінчиться термін дії refresh-токена), а потім захоче повторно увійти в систему по своїй обліковій записі, яку було створено під час реєстрації, йому потрібно буде ввести таку інформацію, як електронна пошта та пароль на екрані

входу Після успішної авторизації, користувача буде перекинуто на домашню сторінку (рис. 3.3).

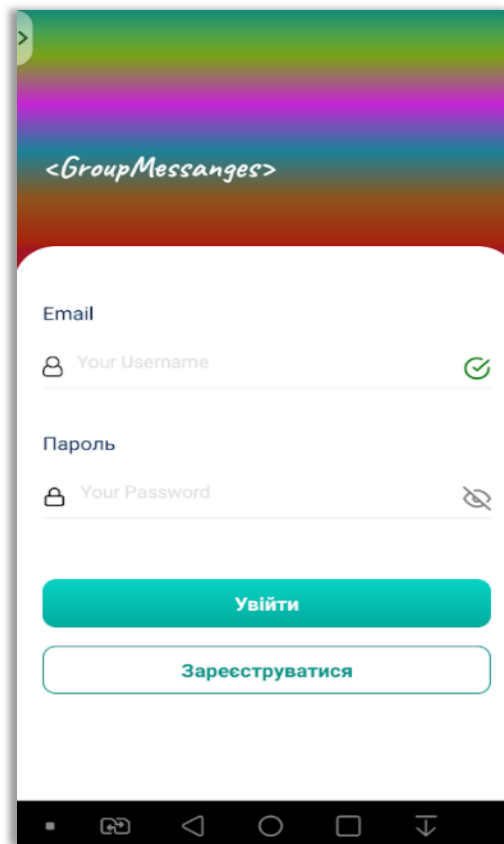


Рисунок 3.3 – «Екран входу в систему»

### 3.1.3. Екран «Чати»

Після успішного проходження авторизації у системі користувач отримає змогу використовувати усі можливості програмної системи, які були надані для його ролі. Одна із цих можливостей це перегляд усіх чатів у яких перебуває поточний користувач. На головній сторінці у вкладці «Chats» буде виводитись список усіх чат кімнат (рис. 3.4).

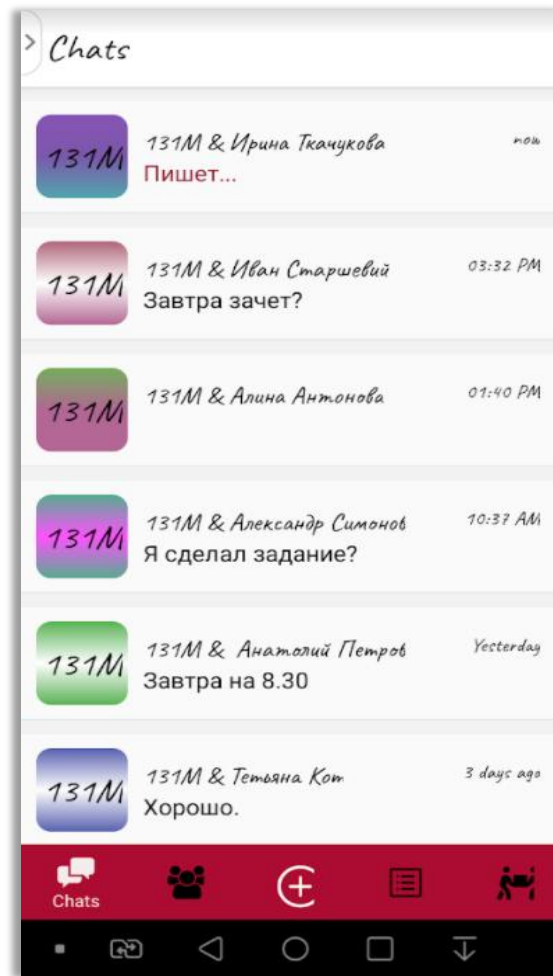


Рисунок 3.4 – Экран «Чати»

### 3.1.4. Экран «Чат»

При виборі однієї з чат-кімнаті користувач переходить на екран чату, де йому будуть відображатися усі повідомлення, які були надіслані учасниками цієї групи навчання разом з повідомленнями викладача (рис. 3.5). Якщо користувач натисне кнопку «Скріпка», то йому відкриється вікно вибору медіа-файлів, де він має можливість обрати формат повідомлення (рис. 3.6).

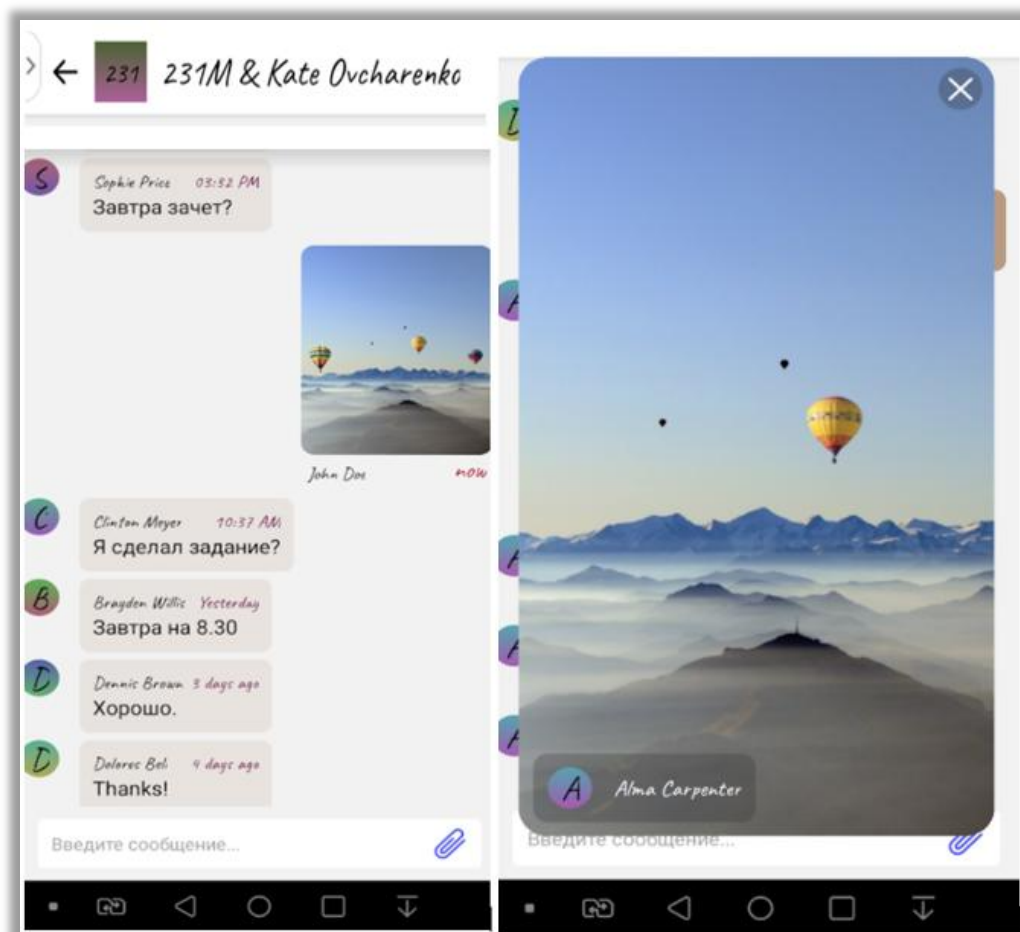


Рисунок 3.5 – Экран «Чат кімната»

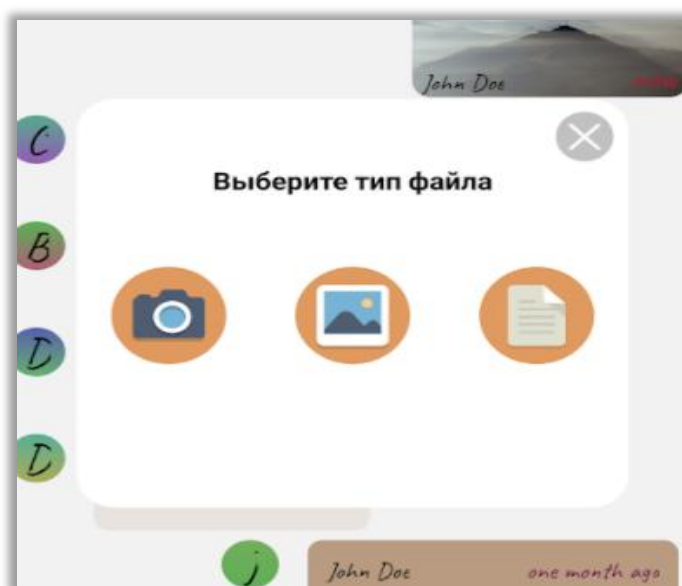


Рисунок 3.6 – «Вікно вибору медіа-файлів»

### 3.1.5. Екран «Мої викладачі»

На головному екрані при активуванні вкладці «Teachers», користувач, у якого роль в системі була визначена як студент, перейде на екран вибору викладача з яким може створити колективний чат (рис. 3.7).

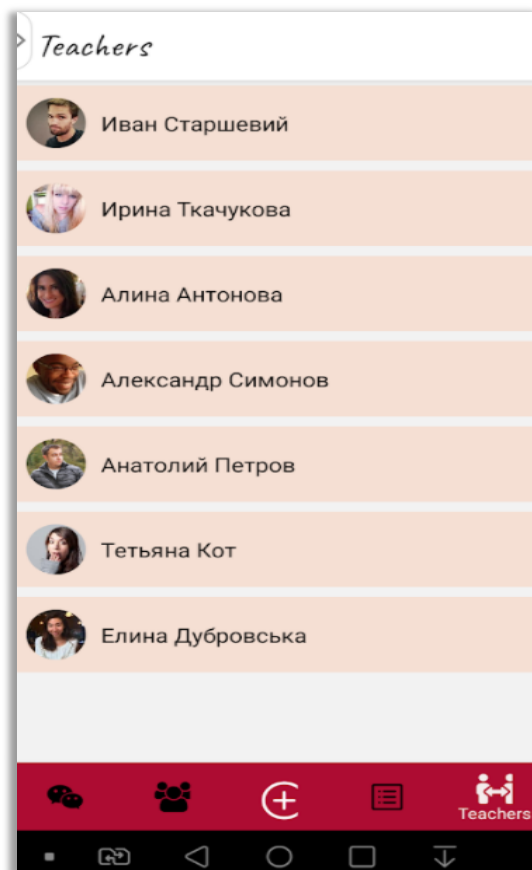


Рисунок 3.7 – Екран «Мої викладачі»

### 3.2. Захист інформації

Аутентифікація і авторизація - дві ключові функції сервісної інфраструктури для захисту конфіденційних даних і операцій для запобігання несанкціонованого доступу з боку злоумисників. Хоча ці два терміни використовуються в одному контексті, вони представляють собою принципово різні поняття, оскільки здійснюють захист взаємодоповнюючими способами. Аутентифікація використовується для підтвердження особи зареєстрованого користувача. Перевірка

справжності - це процес перевірки облікових даних: ідентифікатора користувача (імені, адреси електронної пошти, ролі у системі) і пароля. Дані, які відправляє користувач при вході в систему шифруються за допомогою пакету bcrypt. Bcrypt – адаптивний криптографічний пакет формування ключа, що використовується для захищеного зберігання паролів.

Для ефективного шифрування паролів користувачів слід використовувати комбінацію хешування і соління [23]. Хешування - це процес маскуванню паролю шляхом перетворення його в, здавалося б, випадковий рядок. Функції хешування є «односторонніми». Соління, в свою чергу, являє собою процес генерації довільного рядку з даних, який буде використовуватися в доповненні з паролем, який був закодованим функцією хешування. Це гарантує, що навіть якщо два пароля користувачів виявляться однаковими, ці версії будуть унікальні і не спричинять конфлікту у системі.

Якщо ідентифікатор і пароль співпадають із записами, що зберігаються в базі даних системи, генерується json web token (JWT) [24]. У разі неправильного введення даних програма викликає попередження безпеки і блокує вхід. У несеріалізованому вигляді JWT складається з заголовка і корисного навантаження, які є звичайними JSON-об'єктами.

Заголовок в основному використовується для опису криптографічних функцій, які застосовуються для підпису та / або шифрування токена. Тут також можна вказати додаткові властивості, наприклад, тип змісту, хоча це рідко потрібно. Заголовок описує криптографічні операції, які застосовуються до веб-токену. Але в деяких випадках підпис і шифрування можуть бути відсутні. Зазвичай це відбувається, коли JWT є частиною деякої вже зашифрованою структури даних. Корисні дані - частина токена, в якій розміщується вся необхідна інформація користувача. Як і заголовок, корисне навантаження являє

собою звичайний об'єкт JSON. Тут жодне поле не є обов'язковим. Процес серіалізації JWT складається з кодування заголовка, корисного навантаження та підпису, якщо він є, за допомогою алгоритму base64url. Це проста варіація base64. У коді це можна представити таким чином (рис. 3.10):

```
function encode(h, p) {  
  const header = base64UrlEncode(JSON.stringify(h));  
  const payload = base64UrlEncode(JSON.stringify(p));  
  return `${header}.${payload}`;  
}
```

Рисунок 3.10 – «Серіалізації JWT»

Щоб декодувати токен, досить просто розбити його по точках і конвертувати заголовок і корисне навантаження з коду base64url назад в рядок. Приклад коду, який це робить (рис. 3.11):

```
function decode(jwt) {  
  const [headerB64, payloadB64] = jwt.split('.');  
  const headerStr = base64UrlDecode(headerB64);  
  const payloadStr = base64UrlDecode(payloadB64);  
  return {  
    header: JSON.parse(headerStr),  
    payload: JSON.parse(payloadStr)  
  };  
}
```

Рисунок 3.11 – «Декодування JWT»

Зрозуміло, що JSON токени не кодують вручну. Існує безліч бібліотек, призначених для цього. Наприклад, `jsonwebtoken`. Приклад коду, який це робить (рис. 3.12):

```
const jwt = require('jsonwebtoken');
const secret = 'shhhhh';
// шифрування
const token = jwt.sign({ foo: 'bar' }, secret);
// перевірка та розшифровка
const decoded = jwt.verify(token, secret);
console.log(decoded.foo) // bar
```

Рисунок 3.12 – «Декодування JWT за допомогою `jsonwebtoken`»

Цей код створює підписаний JWT з використанням секретного слова. Потім він перевіряє справжність токена і декодує його, застосовуючи той же секрет.

Таким чином, сервіс обміну повідомленнями «GroupMessenger» відрізняється від інших типів програм тим, що зберігає дані користувачів, а саме: контактні дані, повідомлення, файли. Ці дані потребують особливої уваги до захисту і забезпечення їх безпечного зберігання - одна з основних задач.

### 3.3. Об'єднання клієнтської та серверної підсистеми

Після розробки клієнтської та серверної підсистеми програмної системи, яка представляє собою сервіс обміну повідомленнями «GroupMessenger», постає завдання об'єднати їх в одну функціональну одиницю. Зв'язок між цими компонентами буде здійснюватися за допомогою HTTP-запитів на сервер.

HTTP визначає безліч методів запиту, які вказують, яку бажану дію потрібно виконати для даного ресурсу (Божинов В.) [26]. Кожен



реалізує свою семантику, але кожній групі команд притаманні загальні властивості:

- GET. Метод GET відправляє запит на отримання ресурсу. Запити з використанням цього методу можуть тільки отримувати дані. Використовується для запиту даних з певного ресурсу, на якому дані не змінюються, оскільки GET-запити не змінюють стан ресурсу;
- HEAD. Метод HEAD запитує ресурс так само, як і метод GET, але не передає у відповідь нічого. Метод HEAD використовують для перевірки існування ресурсу, перед виконанням методу GET;
- POST. Метод POST використовується для відправки даних до серверу на створення певного ресурсу. Часто викликає зміна стану або якісь побічні ефекти на сервері;
- PUT. Метод PUT оновить усі поточні уявлення ресурсу даними, які придуть у запиті;
- DELETE. Метод DELETE видаляє вказаний ресурс;
- OPTIONS. Метод OPTIONS використовується для опису параметрів з'єднання з ресурсом;
- PATCH. Метод PATCH використовується для часткової зміни ресурсу;

При безпосередній розробці сервісу обміну повідомленнями, було реалізовано такі методи запиту до серверу:

1. POST (<http://api/user/register>). Метод, який оброблює запит на реєстрацію користувача в системі;
2. POST (<http://api/user/login>). Метод, який оброблює запит на вхід користувача в систему;
3. GET (<http://api/user/verify>). Метод, який оброблює запит перевірки валідності токена користувача;
4. GET (<http://api/user/profile/:id>). Метод, який оброблює запит на отримання профілю користувача по його id;

5. POST (<http://api/chat/:id/message/create>). Метод, який оброблює запит на створення нового повідомлення;
6. DELETE (<http://api/chat/:id/message/delete/:id>). Метод, який оброблює запит на видалення повідомлення;
7. POST (<http://api/chat/create>). Метод, який оброблює запит на створення нової групи;
8. GET (<http://api/chat/all>). Метод, який оброблює запит на отримання усіх чатів;
9. POST (<http://api/user/profile/:id>). Метод, який оброблює запит на отримання профіля користувача;
10. POST (<http://api/user/profile/:id/update>). Метод, який оброблює запит на оновлення профіля користувача;

У додатку А представлено таблицю з загальною інформацією про основні методи, які, безпосередньо, були використані при розробки серверної частини програми.

### **3.4. Висновок до розділу 3**

У розділі 3 «Розробка програмного продукту та опис його функціонування» було виконано такі поставлені завдання:

1. Розробити на основі отриманих знань сервіс обміну повідомленнями «GroupMessenger» та спроектувати до нього базу даних, а саме було:
  - Розроблено клієнтську підсистему, яка представляє собою мобільний додаток;
  - Розроблено інтерфейс користувача у мобільному додатку;
  - Розроблено серверну підсистему, яка представляє собою REST API сервер;
  - Розроблено базу даних, яка представляє собою моделі та колекції даних;

- Описано усі НТТР запити, за допомогою яких здійснюється об'єднання двох підсистем в одну функціональну одиницю;

Усі завдання, які були поставлені в цьому розділі, виконані успішно та у визначені календарні терміни.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було розроблено сервіс обміну повідомленнями «GroupMessenger» факультету ФКНФМ ХДУ для мобільних пристроїв. Для досягнення поставленої в роботі мети, були вирішені наступні завдання:

- 1) Дослідити предметну область об'єкта розробки, а саме було:
  - досліджено вплив інформаційно-комунікативних технологій на освіту;
  - досліджено вплив використання сучасних месенджерів для покращення освітніх процесів;
- 2) Розглянути існуючі рішення об'єкта розробки, а саме такі програми, як:
  - Telegram;
  - Facebook;
  - Viber;
  - WhatApp;
- 3) Сформулювати постановку задачі дипломного проекту;
- 4) Спроектувати архітектуру програмної системи, а саме було:
  - Спроектовано клієнтську підсистему програмної системи обміну повідомленнями «GroupMessenger», яка представляє собою мобільний додаток;
  - Спроектовано серверну підсистему програмної системи обміну повідомленнями «GroupMessenger», яка представляє собою REST API сервер;
  - Спроектовано базу даних, яка представляє собою моделі та колекції даних;
- 5) Визначити набір програмних інструментів за допомогою яких, буде вестися розробка. Внаслідок цього, було вирішено розробляти клієнтську підсистему, за допомогою сучасної та

продуктивної бібліотеки розробки інтерфейсів під мобільні додатки React Native, разом з фреймворком Expo. Серверна підсистема буде розроблюватися за допомогою бібліотеки створення REST API серверів Express. Вибір бази даних зупинився на документо-орієнтованій базі даних MongoDB.

б) Розробити на основі отриманих знань сервіс обміну повідомленнями «GroupMessenger» та спроектувати до нього базу даних, а саме було:

- Розроблено клієнтську підсистему, яка представляє собою мобільний додаток;
- Розроблено інтерфейс користувача у мобільному додатку;
- Розроблено серверну підсистему, яка представляє собою REST API сервер;
- Розроблено базу даних, яка представляє собою моделі та колекції даних;
- Описано усі HTTP запити, за допомогою яких здійснюється об'єднання двох підсистем в одну функціональну одиницю;

Програмна система моєї кваліфікаційної роботи може бути рекомендована до впровадження в інші вищі заклади освіти, зокрема, такі як університети, інститути, академії зі схожими процесами навчання, а також ознайомить з новими технологічними підходами розробки аналогічних проектів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кадемія М. Ю. Інформаційно-комунікаційні технології навчання: словник-госарій. Львів «СПОЛОМ». 2011. 327 с.
2. Информационные и коммуникационные технологии в образовании : монография / Под редакцией: Бадарча Дендева – М. : ИИТО ЮНЕСКО, 2013. – 320 стр.
3. Корольський В.В., Крамаренко Т.Г., Семеріков С.О., Шокалюк С.В. Інноваційні інформаційно-комунікаційні технології навчання: навч. посіб. /; науковий редактор академік АПН України, д. пед. н., проф. М. І. Жалдак. – Кривий Ріг: Книжкове видавництво Кирєєвського, 2009. – 265 с.
4. Долінер Л.І. Інформаційні та комунікаційні технології: проблеми та перспективи: навч. посіб. 2003. Вид. №6. 125-128 с.
5. Дадішов Р., Рахматиллаєва Д. Переваги та недоліки використання інформаційно-комунікаційних технологій: навч. посіб. 2013. 100 с.
6. Most popular global mobile messenger apps as of July 2021, based on number of monthly active users – [Електронний ресурс]. – Режим доступу: (<https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>).
7. Результаты опроса AIN.UA – [Електронний ресурс]. – Режим доступу: (<https://ain.ua/2021/01/26/dlya-506-telegram-osnovnoj-messendzher-rezultaty-oprosa-ain-ua/>).
8. Telegram: что это за программа UA – [Електронний ресурс]. – Режим доступу: (<https://semantica.in/blog/telegram-chto-eto-za-programma.html>).
9. Before It Conquered the World, Facebook Conquered Harvard - [Електронний ресурс]. – Режим доступу: (<https://www.theatlantic.com/technology/archive/2019/02/and-then-there-was-thefacebookcom/582004/>).

10. Advantages and disadvantages of facebook - [Електронний ресурс] – Режим доступу: <https://www.problogbooster.com/2011/09/advantages-disadvantages-of-facebook-drawbacks-benefits-of-social-media.html>.
11. Rakuten Viber Partners with Snap to Bring AR Lenses to its Messaging Platform – [Електронний ресурс]. – Режим доступу: (<https://www.forbusiness.viber.com/en/blog/post/rakuten-viber-snap-partnership/>).
12. The History Of WhatsApp – [Електронний ресурс]. – Режим доступу: (<https://www.feedough.com/history-of-whatsapp/?nowprocket=1>).
13. 12 of the best hidden WhatsApp features you need to know – [Електронний ресурс]. – Режим доступу: (<https://www.cnet.com/tech/services-and-software/12-of-the-best-hidden-whatsapp-features-you-need-to-know/>).
14. Advantages and Disadvantages of WhatsApp – [Електронний ресурс] – Режим доступу: (<https://thetechhacker.com/2019/03/18/advantages-and-disadvantages-of-whatsapp/>).
15. MVC Framework Introduction – [Електронний ресурс] – Режим доступу: ([https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm)).
16. 8 Main Benefits of Building an MVP – [Електронний ресурс]. – Режим доступу: (<https://incorainc.com/the-major-advantages-of-minimum-viable-product-mvp/>).
17. Banks A. Learning Flux: Functional Web Development with Flux and Redux: Підручник. 2017. 183 – 189 с.
18. Офіційна документація React.js [Електронний ресурс] – Режим доступу: (<https://facebook.github.io/react/docs/>).
19. Voduch A. Flux Architecture: Packt Publishing, 2016. 352 с.

20. React Native vs Flutter. Обзор архитектур и «под капотом» – [Електронний ресурс]. – Режим доступу: (<https://dou.ua/forums/topic/34042/>).
21. Masse M. REST API Design Rulebook: К. 2011. 57-60 с.
22. Меджуї М., Уайлд Е., Мітра Р., Амундсен М. Безперервний розвиток API. Правильні рішення в мінливому технологічному ландшафті: навч. посіб. Издательский дом «Питер». 2012. 202-206 с.
23. Разработка на JavaScript. Построение кроссплатформенных приложений с помощью GraphQL, React, React Native и Electron. — СПб.: Питер, 2021. — (Серия «Бестселлеры O'Reilly») – 77 стр.
24. Пять простых шагов для понимания JSON Web Tokens (JWT) [Електронний ресурс]. – Режим доступу: (<https://habr.com/ru/post/340146/>).
25. Использование HTTP методов для создания RESTful сервисов – [Електронний ресурс]. – Режим доступу: (<http://restapitutorial.ru/lessons/httpmethods.html/>).
26. Божинов В. RESTful Web API Design with Node.js: К. Софія. 2015. 56с.
27. Корольський В.В., Крамаренко Т.Г., Семеріков С.О., Шокалюк С.В. Інноваційні інформаційно-комунікаційні технології навчання: навч. посіб. /; науковий редактор академік АПН України, д. пед. н., проф. М. І. Жалдак. – Кривий Ріг: Книжкове видавництво Киреєвського, 2009. – 265 с.
28. Аніщенко О.В. Інформаційно-комунікаційні технології в сучасному світі: досвід, проблеми, перспективи: зб. матеріалів III міжнародної наук.-практ.конф. – К.; Львів: ЛДУ БЖД, 2012. – Вип. 3, ч. 1. – 100-103 с.
29. Lauret A. The Design of Web APIs: К. 2019. 134-138 с.
30. Работа с формами. Формы и их элементы – [Електронний ресурс] – Режим доступу: (<https://metanit.com/web/javascript/10.1.php>).



31. MySQL и MongoDB — когда и что лучше использовать – [Электронный ресурс]. – Режим доступа: (<https://habr.com/ru/post/322532/>).
32. Офіційна документація NodeJS [Електронний ресурс] – Режим доступу: (<https://nodejs.org/>).
33. Гибкая разработка веб-приложений / Сэм Руби., 2014. — 448 с. — (Для профессионалов).
34. Мартін Р. Чистий код: створення, аналіз, рефакторинг: К., 2019. 215-219 с.
35. Атре Ш. Структурний підхід до організації баз даних: навч. посіб. 2005. 320 с.
36. Фіртман М. jQuery Mobile: розробка мобільних інтерфейсів для смартфонів і планшетів СПб .: БХВ-Петербург. 2013. 256 с.
37. С. Хашимі, С. Коматінені, Д. Маклін. Розробка додатків для Android. - СПб .: Пітер. 2011.736 с.
38. Крєнке Д. Теорія і практика проектування баз даних: 8-е изд. - СПб .: Пітер, 2003. 800 с.
39. Миковски М., Разработка одностраничных веб-приложений // Миковски М. - ДМК Пресс, 2014 – с. 512.
40. Setting up a Node development environment – [Електронний ресурс]. – Режим доступу: ([https://developer.mozilla.org/ru/docs/Learn/Server-side/Express\\_Nodejs/development\\_environment](https://developer.mozilla.org/ru/docs/Learn/Server-side/Express_Nodejs/development_environment)).

## ДОДАТОК А

Таблиця А. 1

## HTTP-методи серверної підсистеми

Назва методу	Ресурс	Характеристика
POST	<code>http://api/user/register</code>	Метод, який оброблює запит на реєстрацію користувача в системі
POST	<code>http://api/user/login</code>	Метод, який оброблює запит на вхід користувача в систему
GET	<code>http://api/user/verify</code>	Метод, який оброблює запит перевірки валідності токена користувача
GET	<code>http://api/user/profile/:id</code>	Метод, який оброблює запит на отримання профілю користувача по його id
POST	<code>http://api/chat/:id/message/create</code>	Метод, який оброблює запит на створення нового повідомлення
DELETE	<code>http://api/chat/:id/message/delete/:id</code>	Метод, який оброблює запит на видалення повідомлення

## ДОДАТОК А

Продовження табл. А. 1

Назва методу	Ресурс	Характеристика
POST	<a href="http://api/chat/create">http: /api/chat/create</a>	Метод, який оброблює запит на створення нової групи
GET	<a href="http://api/chat/all">http: /api/chat/all</a>	Метод, який оброблює запит на отримання усіх чатів

## ДОДАТОК Б

Додаток І

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ  
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО  
ДЕРЖАВНОГО УНІВЕРСИТЕТУЯ, Кудін В. А.

учасник(ця) освітнього процесу Херсонського державного університету, УСВІДОМЛЮЮ, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

**ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності ЗОБОВ'ЯЗУЮСЯ:**

– дотримуватися:

- вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
- принципів та правил академічної доброчесності;
- нульової толерантності до академічного плагіату;
- моральних норм та правил етичної поведінки;
- толерантного ставлення до інших;
- дотримуватися високого рівня культури спілкування;

– надавати згоду на:

- безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
- оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
- використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;

– самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;

– надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;

– не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;

– своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;

– не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;

– підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;

– поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;

– не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;

– відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;

– запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;

– не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;

– не підроблювати документи;

– не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;

– не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;

– не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;

– не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;

– не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;

– не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягти власних корисних цілей;

– не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

18.10.2021  
(дата)  
(підпис)Кудін Віталій  
(ім'я, прізвище)