

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка web-застосунку для аналізу та підтримки
емоційного стану»

на здобуття освітнього ступеня бакалавра
зі спеціальності 121 Інженерія програмного забезпечення
освітньо-професійної програми «Інженерія програмного забезпечення»

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Артем БРАЖНИК
(підпис)

Виконав: здобувач вищої освіти групи ПД-42

_____ Артем БРАЖНИК

Керівник: _____ Ігор ГАМАНЮК
ст. викладач

Рецензент: _____

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Бакалавр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« ____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Бражнику Артему Михайловичу

1. Тема кваліфікаційної роботи: «Розробка web-застосунку для аналізу та підтримки емоційного стану»

керівник кваліфікаційної роботи старший викладач кафедри Ігор ГАМАНЮК,
затверджені наказом Державного університету інформаційно-комунікаційних
технологій від «24» лютого 2025 р. № 56.

2. Строк подання кваліфікаційної роботи «02» червня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: теоретичні відомості про цифрову підтримку ментального здоров'я, приклади сучасних застосунків з інтеграцією ІІІ, технічна документація для API ChatGPT, специфікації сучасних web-технологій (Spring Boot, MySQL, JavaScript, Web Speech API), матеріали з UI/UX-дизайну.

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

1. Аналіз сучасних аналогів застосунків для підтримки емоційного стану користувачів.
2. Формування вимог до функціоналу системи.
3. Проектування архітектури web-застосунку з інтеграцією API ChatGPT.

4. Програмна реалізація основних модулів: чат із ШІ, щоденник настрою, поради.
5. Тестування, аналіз роботи системи, оцінка відповідності заданим вимогам.

5. Перелік графічного матеріалу: *презентація*

1. Аналіз аналогів.
2. Вимоги до програмного забезпечення.
3. Програмні засоби реалізації.
4. Діаграма варіантів використання.
5. Архітектура системи
6. Діаграма класів.
7. Екранні форми (чат, щоденник, поради)
8. Результати тестування
9. Апробація результатів

6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Підбір та аналіз науково-технічної літератури	25.02-04.03.2025	
2	Аналіз та дослідження існуючих аналогів	05.03-13.03.2025	
3	Формування вимог до функціоналу web-застосунку	14.03-20.03.2025	
4	Проектування архітектури та інтерфейсу застосунку	21.03-28.03.2025	
5	Програмна реалізація модулів: чат із ШІ, щоденник, аналіз даних	29.03-18.04.2025	
6	Тестування реалізованої системи та її компонентів	19.04-28.04.2025	
7	Оформлення роботи: вступ, висновки, реферат	29.04-11.05.2025	
8	Розробка демонстраційних матеріалів	12.05-18.05.2025	
9	Попередній захист роботи	19.05-30.05.2025	

Здобувач вищої освіти

_____ (підпис)

Артем БРАЖНИК

Керівник
кваліфікаційної роботи

_____ (підпис)

Ігор ГАМАНЮК

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 66 стор., 1 табл., 33 рис., 25 джерел.

Мета роботи – підвищення рівня емоційного благополуччя користувачів за рахунок функціоналу web-застосунку для аналізу та підтримки емоційного стану.

Об'єкт дослідження – процес взаємодії користувача з цифровими інструментами підтримки емоційного стану.

Предмет дослідження – web-застосунок для аналізу емоційного стану з інтеграцією штучного інтелекту.

Короткий зміст роботи: В роботі проаналізовано сучасні технології підтримки ментального здоров'я, а також функціональні можливості аналогічних систем (Woebot, Youper, Wysa, Mindspa). Визначено ключові вимоги до розробки власного застосунку. Розроблено архітектуру системи та інтерфейс користувача, програмно реалізовані основні модулі: чат із ШІ-асистентом (інтеграція з API ChatGPT), щоденник настрою з побудовою графіків на основі Chart.js, система порад та ресурсів, робота з текстовими файлами та голосовий ввід (Web Speech API). Особливу увагу приділено персоналізації взаємодії та аналітиці емоційного стану користувача. Проведено модульне та функціональне тестування застосунку. У роботі використано: Spring Boot, MySQL, JavaScript, HTML/CSS, Web Speech API, Chart.js, OpenAI API.

Сферою використання застосунку є самостійне користування людьми, які потребують м'якої емоційної підтримки, а також у роботі психологів як інструмент первинної аналітики стану клієнтів.

КЛЮЧОВІ СЛОВА: ЕМОЦІЙНЕ ЗДОРОВ'Я, ШТУЧНИЙ ІНТЕЛЕКТ, ЩОДЕННИК НАСТРОЮ, ВІЗУАЛІЗАЦІЯ ДАНИХ, ЕМОЦІЙНА ПІДТРИМКА.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	9
ВСТУП.....	10
1 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ ТА ВИМОГ ДО WEB-СИСТЕМ ДЛЯ ПІДТРИМКИ ЕМОЦІЙНОГО СТАНУ.....	10
1.1 Сучасні web-системи для підтримки емоційного стану.....	10
1.2 Цільова аудиторія застосунків для емоційної підтримки.....	12
1.2.1 Основні категорії цільової аудиторії.....	12
1.2.2 Спільні характеристики та потреби.....	15
1.3 Аналіз аналогічних рішень.....	15
1.4 Порівняльний аналіз функціональності.....	16
1.4.1 Woebot.....	17
1.4.2 Youper.....	19
1.4.3 Wysa.....	20
1.4.4 Mindspace.....	22
1.4.5 Порівняльна таблиця.....	24
1.5 Формування вимог до функціоналу власного застосунку.....	24
1.5.1 Функціональні вимоги.....	25
1.5.2 Діаграма діяльності голосового вводу.....	27
1.5.3 Деталізація функціональних модулів системи.....	29
1.5.4 Діаграма предметної галузі.....	31
1.5.5 Нефункціональні вимоги.....	33
1.5.6 Очікувані переваги реалізованої системи.....	35
2 ЗАСОБИ ТА ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ WEB-ЗАСТОСУНКУ.....	36
2.1 Стек технологій.....	36
2.2 Використання API ChatGPT.....	37
2.3 Інструменти для обробки файлів.....	39
2.4 Побудова графіків за допомогою Chart.js.....	39
2.5 Використання Spring Security для авторизації.....	40
3 ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ.....	42
3.1 Архітектура системи.....	42
3.2 Структура класів системи.....	43
3.3 Компоненти застосунку.....	45
3.4 Моделювання діяльності.....	47
3.5 Моделювання функціональності.....	49
3.6 Логічна архітектура.....	50
3.7 Макети інтерфейсу.....	52

3.8 Розгортання застосунку.....	52
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ.....	54
4.1 Опис реалізованих функціональних можливостей.....	54
4.1.1 Взаємодія з ШІ-асистентом.....	55
4.1.2 Послідовність взаємодії в чат-системі.....	57
4.1.3 Щоденник настрою.....	58
4.1.4 Поради та ресурси.....	60
4.1.5 Персоналізація.....	61
4.1.6 Робота з файлами.....	62
4.2 Голосовий ввід, збереження історії, аналіз файлів.....	63
4.3 Щоденник настрою та його статистика.....	65
4.4 Поради, інтеграція для спеціалістів.....	66
4.5 Тестування.....	66
4.5.1 Unit-тестування.....	67
4.5.2 Інтеграційне тестування.....	69
4.5.3 Функціональне тестування.....	70
ВИСНОВКИ.....	72
ПЕРЕЛІК ПОСИЛАНЬ.....	76
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ.....	78
ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ СЕРВЕРНОЇ ЧАСТИНИ.....	88

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ПЗ - Програмне забезпечення

IDE - Integrated Development Environment – інтегроване середовище розробки

API - Application Programming Interface – інтерфейс прикладного програмування

SPA - Single Page Application – односторінковий застосунок

HTML - HyperText Markup Language – мова гіпертекстової розмітки

CSS - Cascading Style Sheets – каскадні таблиці стилів

JS - JavaScript – мова програмування для клієнтської частини

SQL - Structured Query Language – мова структурованих запитів

ORM - Object-Relational Mapping – об'єктно-реляційне відображення

DI - Dependency Injection – ін'єкція залежностей

JSON - JavaScript Object Notation – формат обміну даними

AI - Artificial Intelligence – штучний інтелект

UI - User Interface – інтерфейс користувача

UX - User Experience – досвід користувача

JWT - JSON Web Token – токен для аутентифікації

REST - Representational State Transfer – архітектурний стиль взаємодії компонентів

ВСТУП

Актуальність: підтримка ментального здоров'я є однією з найважливіших потреб сучасної людини. Стрес, інформаційне перевантаження, соціальна ізоляція та нестача якісної психологічної підтримки призводять до погіршення емоційного стану. Водночас, технології відкривають нові можливості для самодопомоги у сфері емоційного благополуччя. Використання web-застосунків та інтелектуальних асистентів стає ефективним інструментом для щоденної емоційної підтримки.

Одним із найзручніших способів забезпечення такої підтримки є спеціалізований web-застосунок. Він дозволяє користувачам аналізувати свій настрій, взаємодіяти з ШІ, отримувати поради, рекомендації та інші форми емоційної допомоги у зручному для себе форматі. Завдяки використанню сучасних веб технологій, чат-бота, щоденника настрою та адаптивного інтерфейсу, такий застосунок може стати доступним, ефективним і безпечним інструментом для покращення якості життя.

Об'єктом дослідження є процес взаємодії користувача з цифровими інструментами підтримки емоційного стану.

Предметом дослідження є web-застосунок для аналізу емоційного стану з інтеграцією штучного інтелекту.

Метою роботи є підвищення рівня емоційного благополуччя користувачів шляхом створення web-застосунку, що забезпечує аналіз настрою, інтелектуальну підтримку, персоналізовані рекомендації та доступ до корисних ресурсів.

Методи дослідження: на першому етапі було проаналізовано існуючі рішення в галузі цифрової психологічної підтримки та сформульовано функціональні й нефункціональні вимоги до системи. Далі було обрано відповідний стек технологій: Java Spring Boot для бекенд-частини, MySQL для

роботи з базою даних, JavaScript / HTML / CSS для клієнтської частини, API ChatGPT — для реалізації функціональності інтелектуального асистента. Проектування інтерфейсу виконано у Figma. Розробка, тестування й перевірка працездатності здійснювались протягом реалізації проєкту.

Наукова новизна роботи полягає у поєднанні функцій аналізу настрою, збереження емоційних записів, голосової взаємодії, завантаження файлів та персоналізованої роботи ІІІ-асистента у єдиній системі. Web-застосунок є універсальним інструментом, який охоплює одразу кілька ключових аспектів підтримки ментального здоров'я.

Практична значущість результатів полягає у можливості використання застосунку будь-якою категорією користувачів — як для самодопомоги, так і для аналізу динаміки емоційного стану за участі психологів. Рішення може бути адаптоване для використання як окремими особами, так і в медичних чи освітніх установах.

1 АНАЛІЗ ІСНУЮЧИХ АНАЛОГІВ ТА ВИМОГ ДО WEB-СИСТЕМ ДЛЯ ПІДТРИМКИ ЕМОЦІЙНОГО СТАНУ

1.1 Сучасні web-системи для підтримки емоційного стану

Web-системи для підтримки емоційного стану — це спеціалізовані цифрові продукти, що поєднують в собі інструменти моніторингу, аналізу та покращення ментального благополуччя. Такі застосунки допомагають користувачам зменшити рівень стресу, тривоги, подолати емоційне вигорання та інші психологічні труднощі за допомогою інтерактивної взаємодії зі штучним інтелектом або заздалегідь підготовленими вправами та порадами.

Головною метою таких систем є забезпечення безпечного, зручного та доступного способу отримання емоційної підтримки без необхідності прямого контакту зі спеціалістом. Це особливо актуально для користувачів, які через особисті, географічні або фінансові обставини не можуть регулярно звертатися до психолога.

Основні функції подібних web-застосунків:

- Чат-бот на базі ШІ: веде діалог із користувачем, виявляє емоційний стан, дає рекомендації, пропонує вправи або ресурси.
- Щоденник настрою: дозволяє фіксувати емоційні стани протягом дня, формувати емоційний профіль та аналізувати динаміку.
- Бібліотека матеріалів: статті, відео, поради з покращення самопочуття, дихальні вправи, методики релаксації тощо.
- Персоналізація: можливість адаптації відповідей та рекомендацій відповідно до індивідуальних потреб користувача.
- Мобільна доступність: використання з будь-якого пристрою — смартфона, планшета або комп'ютера.

Приклади сучасних систем:

Woebot — один із перших застосунків, що використовує принципи когнітивно-поведінкової терапії (CBT) у чат-форматі. Станом на 2025 рік, додаток має понад 500 тис. завантажень і використовується у понад 135 країнах [1].

Youper — застосунок із ШІ-асистентом, який вивчає емоційний стан користувача на основі діалогів та пропонує адаптивну підтримку [2].

Wysa — поєднує ШІ-помічника з можливістю підключення до живого психолога. Має високі оцінки користувачів за інтуїтивність і надійність [3].

Mindspa — орієнтований на щоденну практику ментального благополуччя. Включає курси, аудіо інструкції та щоденник настрою [4].

Переваги використання web-систем:

- Доступність: користувачі можуть отримати психологічну підтримку у будь-який час, без черг і очікування.
- Анонімність: особливо важливо для людей, які не готові звертатися до психолога особисто.
- Швидка реакція: миттєва взаємодія з ботом або системою без попереднього запису.
- Аналітика: користувач може бачити зміну власного стану у вигляді графіків і звітів.
- Мобільність: застосунки доступні з будь-якого пристрою, що дозволяє підтримку "на ходу".

Недоліки та обмеження:

- Обмеженість допомоги: системи не є заміною кваліфікованої психотерапії у складних випадках.
- Необхідність самодисципліни: ефективне використання залежить від активності самого користувача.
- Якість відповіді ШІ: чат-боти можуть давати узагальнені або шаблонні відповіді, особливо без глибокої контекстуалізації.

- Проблеми персоналізації: не всі застосунки здатні адекватно реагувати на індивідуальні особливості користувача.
- Мовні обмеження: більшість таких систем орієнтовані на англomовну аудиторію, що обмежує доступність для українських користувачів.

1.2 Цільова аудиторія застосунків для емоційної підтримки

Цільова аудиторія веб-застосунків для аналізу та підтримки емоційного стану є різноманітною, охоплюючи користувачів із різними соціальними, професійними та психологічними потребами. Такі системи спрямовані на забезпечення доступної, конфіденційної та персоналізованої підтримки для осіб, які прагнуть покращити свій психологічний добробут, впоратися зі стресом або відстежувати динаміку емоційного стану. Визначення цільової аудиторії є ключовим етапом розробки, оскільки воно дозволяє адаптувати функціональність, інтерфейс та контент застосунку до конкретних потреб користувачів. Нижче наведено детальний аналіз основних груп цільової аудиторії, їхніх характеристик, потреб і мотивацій.

1.2.1 Основні категорії цільової аудиторії

Студенти та молодь: студенти вищих навчальних закладів, коледжів та старшокласники є однією з найбільш вразливих груп до психологічних труднощів. Високе навчальне навантаження, необхідність соціальної адаптації, тиск щодо кар'єрного вибору та фінансові обмеження часто призводять до підвищеного рівня стресу, тривожності та депресивних станів. Згідно з дослідженням, проведеним у 2023 році, 64,17% студентів мали тривожні стани, а 52,94% — депресивні прояви. [5].

Ця група активно використовує цифрові технології, зокрема мобільні додатки та веб-платформи, що робить їх основними користувачами застосунків для емоційної підтримки. Вони потребують швидкого доступу до інструментів для управління стресом, таких як чат-боти з мотиваційними порадами, щоденники настрою для саморефлексії та короткі рекомендації щодо покращення ментального здоров'я. Важливим є інтуїтивний інтерфейс, адаптивний дизайн для мобільних пристроїв і можливість швидкої взаємодії, оскільки студенти часто мають обмежений час.

Працюючі люди: люди віком від 25 до 45 років становлять значну частину цільової аудиторії. Ця група стикається з емоційним вигоранням, багатозадачністю, тиском дедлайнів і необхідністю балансувати між роботою та особистим життям. За даними Всесвітньої організації охорони здоров'я (2024), емоційне вигорання офіційно визнано професійним феноменом, який впливає на продуктивність і психологічний стан працівників [6].

Працюючі люди цінують інструменти, які допомагають швидко знизити рівень стресу, наприклад, через короткі мотиваційні повідомлення, голосовий ввід для зручного спілкування з чат-ботом або доступ до бібліотеки ресурсів із порадами щодо управління часом і емоціями. Вони також потребують конфіденційності, оскільки можуть уникати звернення до традиційних психотерапевтів через стигматизацію або брак часу.

Особи з обмеженим доступом до психологічної допомоги: мешканці віддалених регіонів, сільських місцевостей або країн із недостатньо розвиненою інфраструктурою психологічної підтримки є важливою частиною цільової аудиторії. Крім того, до цієї групи належать користувачі з обмеженими фінансовими можливостями, для яких традиційна психотерапія є недоступною. Веб-застосунки, такі як розроблений у цій роботі, надають альтернативу, пропонуючи безкоштовний або недорогий доступ до інструментів самодопомоги.

Ця категорія користувачів потребує простого інтерфейсу, підтримки кількох мов (зокрема української) та можливості працювати в автономному

режимі або з низькою швидкістю інтернету. Функції, такі як збереження історії чатів, аналіз файлів або голосовий ввід, є особливо цінними, оскільки дозволяють адаптувати взаємодію до технічних обмежень.

Особи з підвищеним емоційним навантаженням: до цієї групи належать військові, волонтери, медичні працівники, вчителі та інші професіонали, які регулярно стикаються з високим рівнем стресу та травматичними ситуаціями. Наприклад, в Україні, враховуючи поточний контекст війни, ця категорія є особливо актуальною. Дослідження 2023 року показали, що 60% медичних працівників в Україні відчувають симптоми посттравматичного стресового розладу [7].

Для таких користувачів важливими є функції, які дозволяють швидко отримати підтримку, наприклад, через швидкі промпти для мотивації чи розради, а також інструменти для відстеження емоційного стану, такі як щоденник настрою з аналітикою. Конфіденційність і можливість налаштування чат-бота (наприклад, зміна імені помічника) сприяють створенню довірливого середовища.

Психологи, консультанти та коучі: професіонали у сфері ментального здоров'я можуть використовувати веб-застосунок як допоміжний інструмент для роботи з клієнтами. Наприклад, вони можуть рекомендувати клієнтам вести щоденник настрою або спілкуватися з чат-ботом між сесіями, а потім аналізувати звіти, згенеровані системою. Функція експорту історії чатів і статистики емоційного стану дозволяє фахівцям оцінювати прогрес клієнтів і адаптувати терапевтичні підходи.

Ця група потребує розширених функціональних можливостей, таких як інтеграція з професійними інструментами, підтримка API для імпорту даних у сторонні системи та високий рівень безпеки для захисту конфіденційних даних клієнтів.

1.2.2 Спільні характеристики та потреби

Незважаючи на різноманітність груп, цільова аудиторія має низку спільних рис, які визначають вимоги до веб-застосунку:

Потреба в конфіденційності: усі категорії користувачів цінують захист особистих даних, особливо в контексті ментального здоров'я, де стигматизація залишається проблемою. Застосунок повинен гарантувати безпечне зберігання даних, шифрування комунікацій і можливість анонімного використання.

Персоналізація: користувачі очікують, що система адаптуватиметься до їхніх потреб, наприклад, через налаштування чат-бота, вибір швидких промптів або рекомендації на основі аналізу настрою.

Простота використання: інтуїтивний інтерфейс, підтримка голосового вводу та швидкий доступ до основних функцій є критичними, особливо для користувачів із низьким рівнем технічної підготовки.

Доступність: застосунок має бути доступним на різних платформах (веб, мобільні пристрої) і підтримувати роботу в умовах обмеженого інтернет-з'єднання.

Емоційна підтримка: користувачі шукають інструменти, які надають мотивацію, розраду або практичні поради в моменти емоційної вразливості.

1.3 Аналіз аналогічних рішень

Аналіз конкурентних рішень дозволяє краще зрозуміти існуючий стан ринку цифрових інструментів для підтримки ментального здоров'я. Нижче розглянуто найбільш популярні застосунки, що мають схожу функціональність або орієнтацію на емоційну підтримку.

Woebot — цифровий психотерапевт на основі когнітивно-поведінкової терапії. Його мета — допомогти користувачам опанувати стрес та тривогу через інтерактивні діалоги. Інтерфейс максимально простий: користувач щодня відповідає на запитання, отримує поради та вправи.

Youper — застосунок із ШІ-асистентом, який веде діалоги з користувачем, відстежує його емоції та пропонує адаптивні поради. Крім того, має модуль фіксації настрою, дихальні практики, медитації, а також базову статистику.

Wysa поєднує чат-бота та можливість підключення професійного психолога. Користувач має доступ до окремих модулів з опрацювання тривоги, низької самооцінки, прокрастинації тощо. Також система пропонує вправи, медитації, мотиваційні фрази.

Mindspa — застосунок для щоденної ментальної гігієни, орієнтований на емоційний журнал, аналітику та освітні матеріали. Головна мета — навчити користувача підтримувати власний емоційний стан через регулярну практику.

1.4 Порівняльний аналіз функціональності

Для оцінки конкурентних рішень у сфері веб-застосунків для підтримки емоційного благополуччя було проведено порівняльний аналіз чотирьох популярних платформ: Woebot, Youper, Wysa та Mindspa. Ці застосунки були обрані через їхню популярність, використання ШІ-технологій та орієнтацію на підтримку ментального здоров'я. Аналіз базується на семи критеріях, які відображають ключові аспекти функціональності таких систем:

- Наявність ШІ-асистента: чи включає застосунок автоматизований чат-бот, який використовує штучний інтелект для взаємодії з користувачем.
- Щоденник настрою: можливість фіксувати та відстежувати емоційні стани користувача.
- Персоналізація: адаптація функціоналу до індивідуальних потреб користувача на основі їхніх даних або вподобань.
- Доступ до ресурсів: наявність бібліотеки зі статтями, вправами, медитаціями чи іншими матеріалами для самодопомоги.
- Аналітика: візуалізація даних про емоційний стан у вигляді графіків, діаграм чи звітів.

- Зв'язок із фахівцем: можливість консультацій із психологами, коучами чи іншими спеціалістами.
- Україномовність/локалізація: підтримка української мови в інтерфейсі та контенті.

Нижче наведено детальний аналіз кожного застосунку, їхні переваги, недоліки та порівняння з розробленим веб-застосунком, який інтегрує ІІІ-асистента на базі ChatGPT API, щоденник настрою, голосовий ввід, аналітику, ресурси, зв'язок із фахівцями та україномовний інтерфейс.

1.4.1 Woebot

Опис функціоналу: Woebot – це ІІІ-чат-бот, розроблений у 2017 році, який використовує принципи когнітивно-поведінкової терапії (CBT) для підтримки користувачів із тривожністю, депресією та стресом. Застосунок пропонує щоденні текстові взаємодії, під час яких бот запитує про настрій і пропонує вправи для рефреймінгу негативних думок. Woebot є безкоштовним і доступним 24/7, що робить його популярним серед користувачів, які шукають швидку підтримку.

Переваги:

- ІІІ-асистент: Woebot використовує семікеровану модель діалогу, дозволяючи користувачам обирати готові відповіді або вводити текст, що забезпечує гнучкість у взаємодії.
- Доступ до ресурсів: застосунок містить бібліотеку з СВТ-вправами, відео та квізами, які допомагають користувачам навчатися технікам самопомогі.
- Дружній інтерфейс: інтерактивний і простий дизайн із кумедним аватаром робота сприяє залученості користувачів.

Недоліки:

- Відсутність щоденника настрою: Woebot не пропонує функцію систематичного відстеження емоційних станів, що обмежує можливості моніторингу прогресу.

- Відсутність аналітики: застосунок не надає візуалізації даних про емоційний стан, що знижує його цінність для довгострокового використання.
- Відсутність зв'язку з фахівцями: Woebot не інтегрує консультації з реальними психологами, що може бути обмеженням для користувачів із серйозними психологічними проблемами.
- Відсутність локалізації: застосунок доступний лише англійською мовою, що ускладнює використання для україномовних користувачів.

Порівняння з розробленим застосунком: На відміну від Woebot, наш веб-застосунок включає щоденник настрою з аналітикою у вигляді графіків, інтеграцію з фахівцями через текстові консультації, україномовний інтерфейс і голосовий ввід для зручності. Завдяки ChatGPT API наш ШІ-асистент пропонує більш природні та контекстно-залежні діалоги порівняно з семікерованим підходом Woebot.

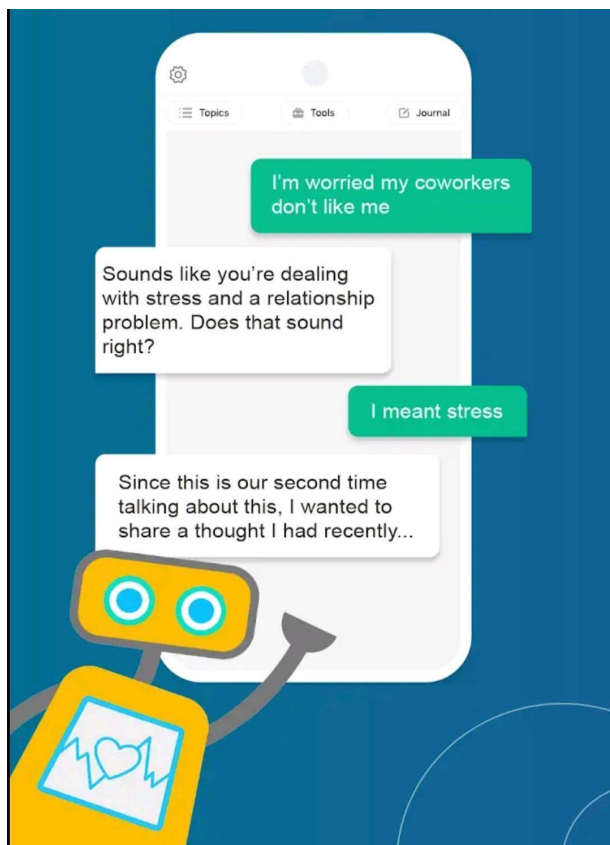


Рис. 1.1 Діалог із ШІ-асистентом у застосунку Woebot

1.4.2 Youper

Опис функціоналу: Youper – це ШІ-платформа, заснована у 2016 році, яка комбінує СВТ, прийняття та терапію зобов'язань (АСТ) і медитацію для підтримки ментального здоров'я. Застосунок пропонує щоденні check-ins, під час яких користувачі фіксують настрій і отримують персоналізовані рекомендації. Youper має 2 мільйони користувачів і визнаний за ефективність у зменшенні симптомів тривожності та депресії.

Переваги:

- ШІ-асистент: чат-бот Youper використовує обробку природної мови (NLP) для створення природних діалогів, що сприяє емоційній відкритості.
- Щоденник настрою: користувачі можуть фіксувати емоційні стани щодня, що допомагає відстежувати прогрес.
- Персоналізація: застосунок адаптує вправи та рекомендації на основі даних із check-ins, що підвищує релевантність контенту.
- Доступ до ресурсів: Youper містить бібліотеку з медитаціями, СВТ-вправами та статтями.

Недоліки:

- Відсутність аналітики: хоча Youper відстежує настрій, він не пропонує детальних графіків чи звітів для аналізу даних.
- Відсутність зв'язку з фахівцями: застосунок не інтегрує консультації з психологами, що обмежує його для користувачів із серйозними потребами.
- Відсутність локалізації: Youper доступний лише англійською, що створює бар'єр для україномовних користувачів.
- Обмежений безкоштовний доступ: більшість функцій доступні лише в преміум-версії (7-денний пробний період, потрібна кредитна картка).

Порівняння з розробленим застосунком: наш застосунок перевершує Youper завдяки наявності аналітики (графіки настрою), інтеграції з фахівцями, україномовного інтерфейсу та голосового вводу. ChatGPT API забезпечує більш розширені діалоги порівняно з NLP Youper, а безкоштовний доступ до всіх функцій робить наш застосунок більш інклюзивним.

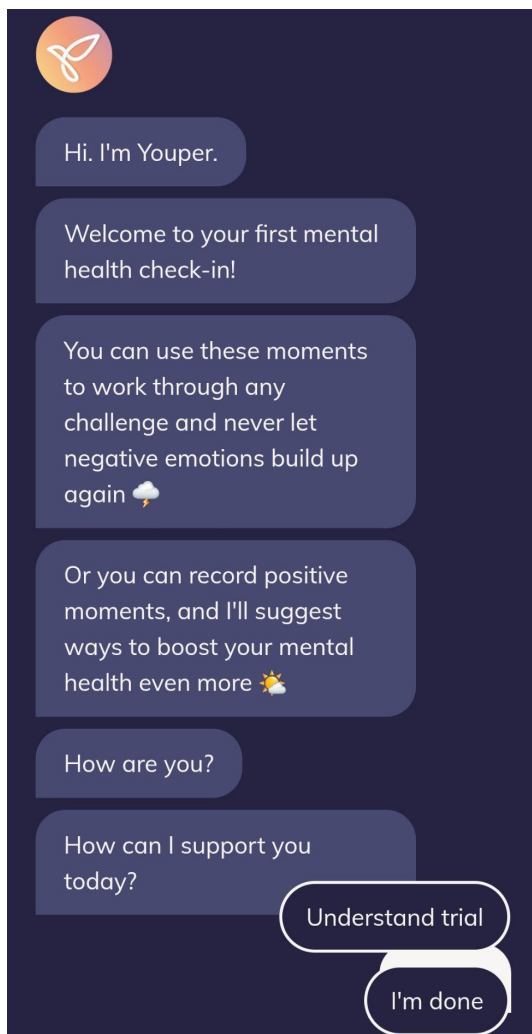


Рис. 1.2 Щоденний check-in у застосунку Youper

1.4.3 Wysa

Опис функціоналу: Wysa – це ШІ-чат-бот, запущений у 2016 році, який використовує СBT, діалектичну поведінкову терапію (DBT) і медитацію для управління тривожністю, депресією та стресом. Застосунок має 3 мільйони користувачів і підтримує анонімність, що сприяє довірі. Wysa пропонує як

безкоштовну, так і преміум-версію (\$74.99/рік), а також текстові консультації з коучами.

Переваги:

- ІІІ-асистент: Wysa використовує відкритий стиль діалогу, що дозволяє більш вільну взаємодію порівняно з Woebot.
- Щоденник настрою: користувачі можуть оцінювати настрій за допомогою слайдера з емоді, що є інтуїтивним.
- Персоналізація: застосунок адаптує вправи та рекомендації на основі історії взаємодій.
- Доступ до ресурсів: Wysa пропонує бібліотеку з 150+ вправ, медитацій і технік дихання.
- Зв'язок із фахівцем: преміум-версія включає текстові сесії з коучами (\$19.99/сесія).

Недоліки:

- Відсутність аналітики: Wysa не надає графіків чи звітів для аналізу емоційного стану, що обмежує довгостроковий моніторинг.
- Відсутність локалізації: застосунок не підтримує українську мову, що знижує його доступність для українських користувачів.
- Обмеження ІІІ: чат-бот іноді неправильно інтерпретує відповіді, що може викликати роздратування.
- Платний доступ до ключових функцій: багато вправ і консультацій доступні лише в преміум-версії.

Порівняння з розробленим застосунком: Наш застосунок перевершує Wysa завдяки наявності аналітики, україномовного інтерфейсу та безкоштовного доступу до всіх функцій. Голосовий ввід і інтеграція з ChatGPT API забезпечують більш зручну та природну взаємодію.



Рис. 1.3 Ресурси для самодопомоги у застосунку Wysa

1.4.4 Mindspa

Опис функціоналу: Mindspa – це застосунок для самостійної терапії, який фокусується на курсах із CBT і гештальт-терапії. Він включає щоденник настрою, статті, медитації та аудіотерапію, але не має ШІ-чат-бота. Курси коштують від \$4.99 до \$99.99, а безкоштовна версія обмежена.

Переваги:

- Щоденник настрою: Mindspa дозволяє фіксувати емоційні стани та вести щоденник.
- Доступ до ресурсів: застосунок пропонує курси, статті, аудіо- та відеоконтент, розроблені психологами.

- Аналітика: Mindspace надає базові звіти про прогрес у курсах, що є перевагою порівняно з іншими аналогами.

Недоліки:

- Відсутність ШІ-асистента: відсутність чат-бота обмежує інтерактивність і швидку підтримку.
- Відсутність персоналізації: курси є стандартизованими, без адаптації до індивідуальних потреб.
- Відсутність зв'язку з фахівцями: Mindspace не пропонує консультацій із психологами.
- Відсутність локалізації: застосунок не підтримує українську мову.
- Висока вартість: більшість контенту доступна лише за плату, що знижує доступність.

Порівняння з розробленим застосунком: Наш застосунок перевершує Mindspace завдяки наявності ШІ-асистента, персоналізації, інтеграції з фахівцями, україномовного інтерфейсу та безкоштовного доступу. Голосовий ввід і розширена аналітика також роблять наш продукт більш універсальним.



Рис. 1.4 Щоденник настрою у застосунку Mindspace

1.4.5 Порівняльна таблиця

Таблиця 1.1

Порівняльна характеристика аналогічних застосунків для підтримки емоційного благополуччя

Застосунок	ШІ-асистент	Щоденник настрою	Персоналізація	Інформаційні матеріали	Аналіз даних	Фахівці	Локалізація
Woebot	+	–	–	+	–	–	–
Youper	+	–	+	+	–	–	–
Wysa	+	+	–	–	–	+	–
Mindspa	–	+	–	+	–	–	–
Emotionalee	+	+	+	+	+	+	+

1.5 Формування вимог до функціоналу власного застосунку

На основі проведеного аналізу аналогічних рішень (Woebot, Youper, Wysa, Mindspa), визначення цільової аудиторії та врахування сучасних викликів у сфері ментального здоров'я сформовано вимоги до функціоналу веб-застосунку. Розроблена система покликана не лише надавати емоційну підтримку, але й забезпечувати комплексну взаємодію користувача з інструментами для аналізу та покращення психологічного стану. Вимоги враховують сильні сторони конкурентів, компенсують їхні недоліки та відповідають потребам

україномовної аудиторії, зокрема в контексті соціально-психологічної ситуації в Україні.

Функціональні вимоги відображають необхідність створення універсального рішення, яке поєднує ШІ-технології, аналітику, ресурси для самодопомоги та інтеграцію з фахівцями.

Нефункціональні вимоги гарантують надійність, доступність і зручність системи. Усі вимоги сформульовано з урахуванням технологічного стеку (Java Spring Boot для серверної частини, MySQL для зберігання даних, JavaScript і HTML/CSS для клієнтської частини, ChatGPT API для ШІ-асистента), що забезпечує ефективну реалізацію.

1.5.1 Функціональні вимоги

Функціональні вимоги визначають основні можливості системи, які забезпечують її цінність для користувачів. Вони розроблені з урахуванням потреб цільової аудиторії (студенти, професіонали, особи з обмеженим доступом до психологічної допомоги, фахівці) та недоліків конкурентів, таких як відсутність україномовної локалізації чи аналітики.

ШІ-асистент є центральним компонентом системи, який використовує ChatGPT API для створення інтерактивного чат-бота. Асистент надає:

- Емоційну підтримку: мотиваційні повідомлення, поради щодо управління стресом і розраду в складних ситуаціях (наприклад, відповіді на запити типу “мені сумно”).
- Психоедукацію: короткі пояснення психологічних концепцій (наприклад, як працює тривожність) для підвищення обізнаності користувачів.
- Контекстну взаємодію: збереження історії діалогів для забезпечення послідовності в розмовах, що особливо важливо для довгострокового використання.

На відміну від Woebot, який використовує семікеровані діалоги, або Wysa, де ШІ іноді неправильно інтерпретує запити, ChatGPT API забезпечує більш природні та точні відповіді. Асистент підтримує українську мову, що усуває

бар'єри для україномовних користувачів, на відміну від усіх аналізованих аналогів.

Швидкі промпти – це готові опції для ініціації діалогу з ШІ-асистентом, які спрощують взаємодію для користувачів із обмеженим часом або емоційною енергією. Приклади промптів:

- “розвесели мене” – бот пропонує жарт, позитивну цитату або коротку вправу.
- “дай пораду” – рекомендація щодо управління стресом чи вирішення конфлікту.
- “мотивація на день” – натхненне повідомлення для підвищення продуктивності.

Ця функція враховує потреби працюючих професіоналів і студентів, які потребують швидкої підтримки. На відміну від Youper, де рекомендації прив'язані до check-ins, швидкі промпти доступні в будь-який момент і не вимагають попереднього введення даних.

Індивідуалізація відповіді: користувачі можуть *персоналізувати* взаємодію з ШІ-асистентом, що підвищує довіру та комфорт. Функція включає:

- Налаштування імені: можливість обрати ім'я для ШІ-помічника (наприклад, “Олена” чи “Максим”).
- Адаптація стилю: вибір тону спілкування (формальний, дружній, мотиваційний).
- Збереження вподобань: система запам'ятовує налаштування для кожної сесії.

Ця функція перевершує обмежену персоналізацію Wysa чи Youper, де адаптація базується лише на історії настрою, і відсутня в Woebot та Mindspace. Індивідуалізація особливо важлива для осіб із підвищеним емоційним навантаженням (військові, медики), які потребують довірливого середовища.

Голосовий ввід дозволяє користувачам взаємодіяти з ШІ-асистентом через розпізнавання мови, що конвертується в текст і передається на обробку ChatGPT API. Ця функція:

- Підвищує доступність для користувачів із фізичними обмеженнями чи тих, хто уникає текстового вводу.
- Спрощує взаємодію для осіб із низьким рівнем технічної підготовки (наприклад, мешканці віддалених регіонів).
- Забезпечує зручність у ситуаціях, коли друкування неможливе (наприклад, під час прогулянки).
- Голосовий ввід є унікальною функцією, відсутньою в Woebot, Youper, Wysa та Mindspace, що робить наш застосунок більш інклюзивним.

1.5.2 Діаграма діяльності голосового вводу

На рисунку 1.5 зображено блок-схему, яка відображає логіку роботи модуля голосового введення, реалізованого у web-застосунку для підтримки емоційного стану користувачів. Основна мета цього модуля — забезпечити альтернативний спосіб взаємодії з ШІ-асистентом через мову, що особливо актуально в ситуаціях підвищеної емоційної напруги або коли користувачеві важко друкувати текст вручну.

Алгоритм починається з активації мікрофона. Після натискання користувачем кнопки активації здійснюється запис голосового повідомлення. Після завершення запису система перевіряє якість отриманого аудіо. Це необхідно, оскільки фоновий шум або нечітка вимова можуть ускладнити подальше розпізнавання тексту. Якщо запис виявляється недостатньо якісним, користувач отримує повідомлення про помилку та можливість повторити спробу. У разі згоди процес повертається до етапу запису.

Якщо запис якісний, система переходить до наступного кроку — перетворення аудіо в текст. Цей процес реалізовано за допомогою Web Speech API, вбудованого у браузерну платформу. Важливо, що дані обробляються локально або через захищений канал, що відповідає вимогам до конфіденційності.

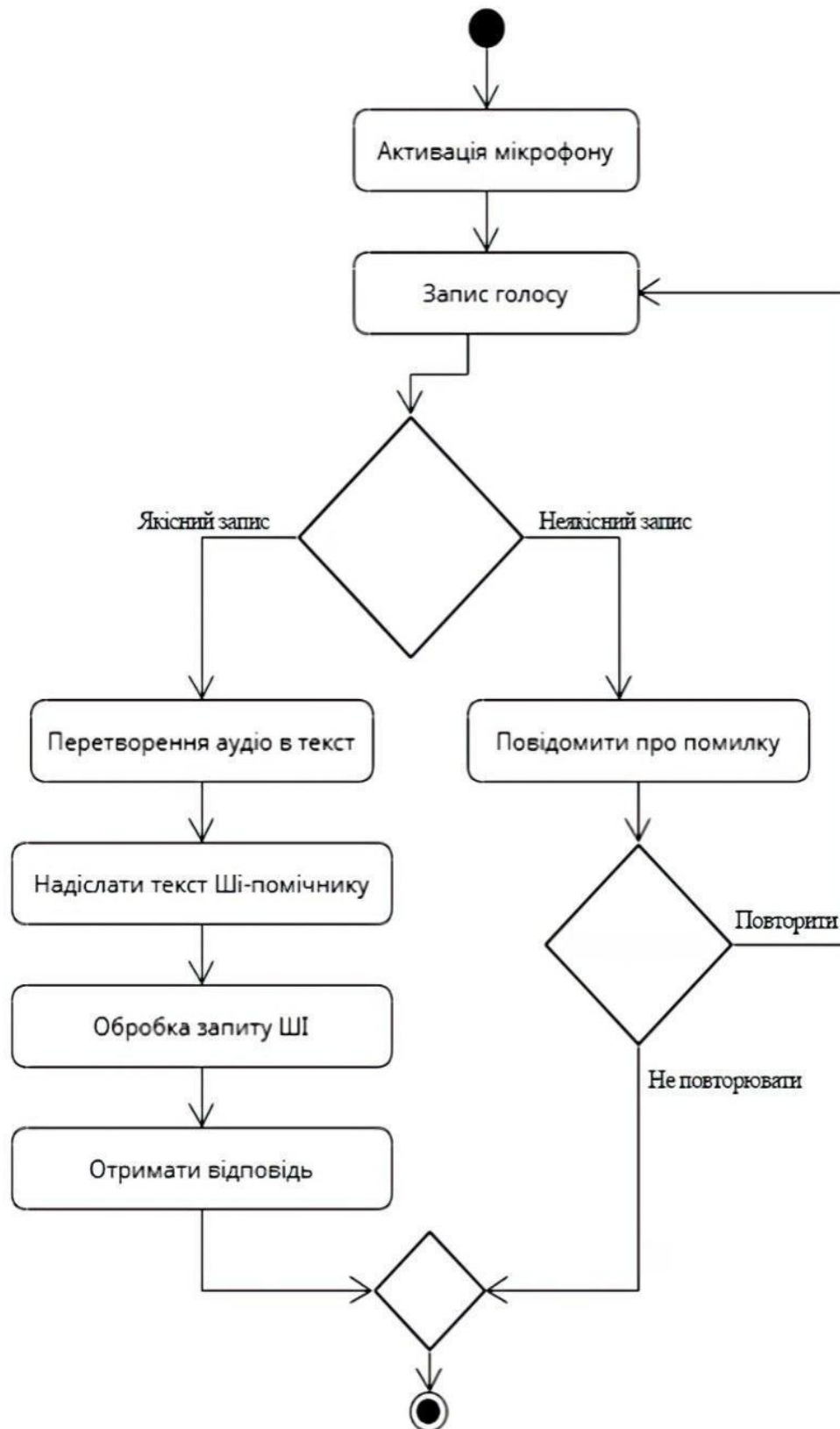


Рис. 1.5 Діаграма діяльності роботи голосового вводу: від запису аудіо до обробки тексту ChatGPT API та отримання відповіді

Після успішного перетворення мова стає звичайним текстом, який передається до ШІ-асистента. Далі відбувається стандартна обробка запиту за допомогою API ChatGPT: текст аналізується, інтерпретується і формується відповідь. Коли ШІ завершить обробку, згенерована відповідь повертається користувачу через інтерфейс чату.

Таким чином, реалізований алгоритм забезпечує повноцінний цикл голосової взаємодії — від аудіозапису до отримання осмисленої текстової відповіді. Це підвищує доступність системи, дозволяє користувачам виражати свої думки природніше та робить застосунок більш інклюзивним.

1.5.3 Деталізація функціональних модулів системи

Щоденник настрою дозволяє користувачам фіксувати емоційні стани, аналізувати їх і отримувати візуалізацію прогресу. Функція включає:

- Введення записів: вибір емоцій (наприклад, радість, тривога, сум) із коментарем.
- Історія: перегляд усіх записів із фільтрацією за датою.
- Аналітика: побудова графіка (лінійного) для відображення динаміки настрою за тиждень, місяць чи рік.
- Рекомендації: ШІ-асистент пропонує вправи чи ресурси на основі аналізу (наприклад, медитація при тривалій тривожності).

Ця функція перевершує Wysa та Youper, які не надають графіків, і Mindspa, де аналітика обмежена звітами про курси. Аналітика реалізована за допомогою JavaScript-бібліотек (Chart.js) і даних із MySQL.

Ресурси та поради: система містить бібліотеку матеріалів для самодопомоги, що включає:

- Статті: категоризовані тексти про управління стресом, емоційну гігієну, СБТ-техніки.

- Вправи: інтерактивні завдання (наприклад, техніка “5-4-3-2-1” для заспокоєння).

- Поради: щоденні рекомендації від ШІ на основі профілю користувача.

На відміну від Woebot чи Wysa, де ресурси частково платні, наша бібліотека безкоштовна і локалізована українською мовою. Контент зберігається в MySQL і динамічно відображається через Spring Boot.

Робота з файлами: користувачі можуть завантажувати текстові документи (наприклад, есе, нотатки, листи) для аналізу ШІ-асистентом. Функція дозволяє:

- Аналіз змісту: ШІ оцінює емоційний тон документа і пропонує рекомендації (наприклад, “текст містить тривожні нотки, спробуйте вправу на релаксацію”).
- Зворотний зв'язок: коментарі щодо структури чи змісту (корисно для студентів).
- Конфіденційність: файли обробляються локально і не зберігаються після сесії.

Ця функція є унікальною, відсутньою в аналізованих аналогах, і реалізована через Spring Boot та JavaScript для обробки файлів і ChatGPT API для аналізу.

Особистий кабінет користувача: особистий кабінет забезпечує управління профілем і даними користувача, включаючи:

- Налаштування: зміна імені ШІ, стилю взаємодії.
- Історія чатів: перегляд і експорт діалогів із ШІ.
- Завантаження даних: управління файлами та записами настрою.
- Вихід: безпечне завершення сесії з очищенням кешу.

Кабінет розроблено з використанням HTML/CSS і JavaScript для адаптивного дизайну, а серверна логіка реалізована через Spring Boot із MySQL

для зберігання даних. Ця функція перевершує аналоги, де управління даними обмежене (наприклад, Woebot не пропонує експорту).

Інтеграція з фахівцями: система підтримує взаємодію з психологами чи коучами, що включає:

- Експорт даних: користувач може завантажити звіти про динаміку свого настрою та результати взаємодії з системою у зручному форматі, а потім поділитися цими матеріалами зі спеціалістом через електронну пошту або під час очної зустрічі.
- Спільний аналіз: система створює зручні аналітичні звіти, які користувач і спеціаліст можуть разом аналізувати для відстеження прогресу та корегування підходів.
- Персоналізовані рекомендації: психологи та коучі можуть створювати індивідуальні плани вправ чи рекомендації на основі отриманих від користувача даних, які потім користувач може самостійно завантажити вказівки в систему для щоденного використання.

Ця функція перевершує Wysa, де консультації платні, і відсутня в Woebot, Youper та Mindspa. Реалізація базується на Spring Boot для управління користувачами та MySQL для зберігання звітів.

1.5.4 Діаграма предметної галузі

На рисунку 1.6 зображено діаграму предметної галузі, яка відображає основні об'єкти системи та зв'язки між ними у контексті підтримки емоційного стану користувачів. Діаграма дозволяє візуально уявити логіку роботи застосунку на концептуальному рівні, до початку розробки конкретної реалізації.

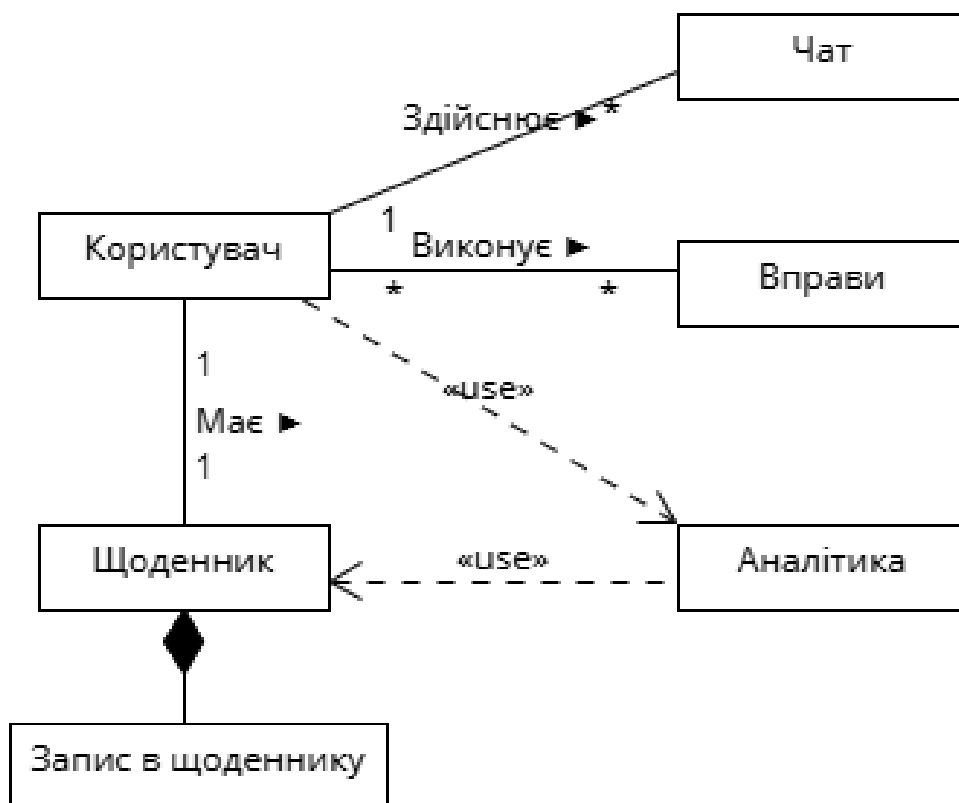


Рис. 1.6 Діаграма предметної галузі

Ключовим об'єктом є Користувач, який взаємодіє з усіма основними підсистемами. Зокрема, він:

- здійснює взаємодію з чатом, де отримує поради та підтримку від ІІІ-асистента;
- виконує вправи, доступні у розділі ресурсів;
- має щоденник, у якому фіксує записи про власний емоційний стан;
- використовує аналітику, що базується на внесених даних.

Щоденник реалізовано як окремий об'єкт, що має один або кілька записів. У свою чергу, Запис в щоденнику зв'язаний з Щоденником через композицію, що вказує на залежність життєвого циклу: записи не існують самостійно, а лише як частина конкретного щоденника.

Додатково, за допомогою асоціацій «use», позначено, що Щоденник користується аналітикою, а сам Користувач також прямо або опосередковано взаємодіє з модулями статистики та рекомендацій.

Такий підхід дозволяє забезпечити структуровану та розширювану модель системи, де кожен компонент виконує окрему функцію в межах загального сценарію роботи користувача.

1.5.5 Нефункціональні вимоги

Нефункціональні вимоги забезпечують якість, надійність і зручність системи, враховуючи потреби цільової аудиторії та технічні обмеження.

Безпека та конфіденційність

- Шифрування: усі дані (чати, файли, записи настрою) передаються через HTTPS і шифруються за допомогою AES-256.
- Авторизація: доступ лише для зареєстрованих користувачів через JWT-токени, реалізовані в Spring Security.
- Анонімність: можливість використання без надання особистих даних (лише email).
- GDPR-сумісність: видалення даних за запитом користувача.

Ці заходи відповідають потребам усіх груп аудиторії, особливо осіб із підвищеним емоційним навантаженням, які цінують конфіденційність.

Доступність

- Кросплатформність: підтримка десктопів, планшетів і смартфонів через адаптивну верстку (HTML/CSS із Bootstrap).
- Офлайн-режим: кешування ресурсів і записів настрою для роботи при слабкому інтернеті.

Це забезпечує інклюзивність для користувачів із віддалених регіонів і тих, хто має технічні обмеження.

Мовна локалізація

- Українська мова: повна локалізація інтерфейсу, чат-бота, ресурсів і звітів.
- Мультимовність: можливість додавання інших мов (англійської, польської) у майбутньому.
- Ця вимога усуває недолік конкурентів, які не підтримують українську мову, і відповідає потребам україномовної аудиторії.

- *Простота інтерфейсу*
- Мінімалізм: чітка структура з інтуїтивною навігацією (меню, кнопки, підказки).
- Консистентність: єдиний стиль елементів (кольори, шрифти) для всіх сторінок.
- Навчання: вбудовані підказки для нових користувачів (наприклад, як почати чат).

Простота враховує потреби користувачів із низьким рівнем технічної підготовки, таких як мешканці сільських регіонів.

Швидкодія

- Час відповіді: серверна обробка запитів до 1 секунди (Spring Boot із оптимізованими контролерами).
- Завантаження сторінок: до 2 секунд на клієнтській частині (асинхронний JavaScript).
- Обробка даних: швидкий аналіз файлів і побудова графіків завдяки MySQL і Chart.js.

Це забезпечує комфорт для працюючих професіоналів і студентів, які цінують ефективність.

Можливість масштабування

- Модульна архітектура: Spring Boot із мікросервісним підходом для легкого додавання функцій.
- База даних: MySQL із підтримкою індексації для обробки зростаючої кількості користувачів.
- API: гнучка інтеграція з ChatGPT API та іншими сервісами в майбутньому.

Ця вимога гарантує довгострокову актуальність системи, враховуючи потенційне розширення аудиторії.

1.5.6 Очікувані переваги реалізованої системи

Розроблений веб-застосунок пропонує низку унікальних переваг порівняно з аналогами:

- Комплексність: поєднання ШІ-чату, щоденника настрою, аналітики, ресурсів, голосового вводу та інтеграції з фахівцями в одному рішенні, на відміну від фрагментарних функцій Woebot, Youper, Wysa чи Mindspa.
- Персоналізація: гнучке налаштування ШІ, промптів і стилю взаємодії, що перевершує обмежену адаптацію конкурентів.
- Аналітика: розширена візуалізація даних (графіки, звіти), відсутня в більшості аналогів.
- Локалізація: повна підтримка української мови, що робить застосунок доступним для 70% українців, які потребують психологічної підтримки [8].
- Універсальність: орієнтація на звичайних користувачів (студентів, професіоналів, осіб із обмеженим доступом) і фахівців (психологів, коучів).
- Інклюзивність: голосовий ввід і адаптивний дизайн забезпечують доступність для всіх груп аудиторії.
- Безкоштовність: усі функції доступні без додаткових витрат, на відміну від платних моделей Wysa чи Mindspa.

2 ЗАСОБИ ТА ТЕХНОЛОГІЇ РЕАЛІЗАЦІЇ WEB-ЗАСТОСУНКУ

Ефективна реалізація web-застосунку для підтримки емоційного стану потребує вибору відповідного технологічного стеку, що забезпечить надійність, масштабованість, простоту використання і швидкодію системи. У цьому розділі розглянуто інструменти та програмні засоби, які були використані при створенні системи.

2.1 Стек технологій

Розроблений web-застосунок побудований за принципом клієнт-серверної архітектури. Для реалізації було обрано сучасний стек технологій, який забезпечує гнучкість, масштабованість і сумісність між усіма компонентами.

Основні складові стеку:

- Back-end: Java 23, Spring Boot 3

Spring Boot — це популярний Java-фреймворк, який дозволяє швидко створювати готові до використання серверні застосунки з підтримкою REST API, інтеграцією баз даних і систем аутентифікації [9].

- База даних: MySQL 8

MySQL — одна з найпоширеніших систем управління реляційними базами даних (RDBMS). Вона забезпечує збереження даних користувачів, історії чатів, записів щоденника та іншої інформації.

- Front-end: HTML5, CSS3, JavaScript ES6

Ці технології забезпечують побудову інтерактивного та адаптивного користувацького інтерфейсу.

Інтеграція III: API ChatGPT (OpenAI)

- API ChatGPT використовується для реалізації функціональності інтелектуального асистента — генерації відповідей, аналізу введених даних користувача, формування рекомендацій [10].

- Голосовий ввід: Web Speech API

Ця технологія дозволяє реалізувати голосовий ввід на стороні клієнта, розпізнаючи мову і перетворюючи її на текст для подальшої обробки.

Інструменти розробки:

- IDE: IntelliJ IDEA (для Java)
- Figma (для прототипування інтерфейсу)
- Postman (для тестування API)
- GitHub (для контролю версій)

Додаткові бібліотеки:

- Lombok — для спрощення написання коду в Java (геттери, сеттери, конструктори);
- Spring Security — для аутентифікації та авторизації;
- Chart.js — для побудови графіків зміни настрою на клієнті.

Таким чином, обраний стек дозволяє забезпечити надійне функціонування системи як на серверній, так і на клієнтській стороні.

2.2 Використання API ChatGPT

Ключовою особливістю розробленого застосунку є інтеграція із сервісом OpenAI ChatGPT. Завдяки цьому користувач отримує доступ до сучасного штучного інтелекту, який здатен вести природну діалогову взаємодію.

Основні функції, реалізовані через API ChatGPT:

- генерація відповідей на введені текстові або голосові запити;
- формування рекомендацій (поради, мотивація, підтримка);
- аналіз емоційного тону введених повідомлень (на основі тексту);
- збереження контексту бесіди для підтримки зв'язності діалогу.

Крім базових можливостей, у системі реалізовано низку додаткових функцій, які розширюють взаємодію користувача з асистентом:

- Голосовий ввід (Web Speech API): користувач може диктувати запити голосом, які автоматично конвертуються у текст та надсилаються на обробку ChatGPT;
- Початок нового чату: передбачено функцію очищення історії діалогу та створення нового контексту взаємодії;
- Збереження історії чатів: усі сесії зберігаються у базі даних для подальшого перегляду, аналізу чи експорту;
- Завантаження файлів: користувач може надсилати текстові файли (формати .txt, .md, .json), вміст яких опрацьовується ШІ, що дозволяє аналізувати великі обсяги тексту або отримувати рекомендації на основі власних документів;
- Експорт історії: користувач може завантажити історію чатів у зручному форматі для подальшого використання.

Механізм інтеграції:

- підключення API-ключа OpenAI на стороні бекенду;
- використання REST-запитів (POST) до API ChatGPT;
- передача параметрів: історія чату, поточне повідомлення, інструкції;
- обробка відповіді та її відправка користувачеві.

Використання ChatGPT дозволяє значно підвищити якість взаємодії з користувачем, зробити асистента більш гнучким і природним, а також забезпечити широкий спектр можливостей для персоналізації. На наступній діаграмі зображена логіка на кожному рівні системи, включно з роботою ChatGPT API.

2.3 Інструменти для обробки файлів

У системі реалізовано функцію завантаження користувачем текстових файлів для аналізу їх змісту ШІ-асистентом.

Основні технічні рішення:

- JavaScript FileReader API — дозволяє на стороні браузера зчитувати вміст завантаженого текстового файлу без надсилання його на сервер;
- Обмеження: підтримуються файли форматів .txt, .md, .json;
- Механізм: після зчитування вміст надсилається на бекенд разом із запитом до API ChatGPT.

Це розширює можливості системи та дозволяє користувачам отримувати гнучкі відповіді на основі власних текстів (есе, нотатки тощо).

2.4 Побудова графіків за допомогою Chart.js

Щоб надати користувачам зручний інструмент для аналізу змін емоційного стану, у системі реалізовано модуль побудови графіків на основі даних щоденника настрою. Для цього використано бібліотеку Chart.js — популярне JavaScript-рішення для інтерактивної візуалізації даних [11].

Основні можливості реалізованого модуля:

- відображення динаміки настрою користувача у вигляді лінійного графіка;
- можливість обирати період аналізу (тиждень, місяць, рік);
- відображення середнього настрою за вибраний період;
- інтерактивність: підказки при наведенні на точку графіка.

Технічна реалізація:

- дані про настрої отримуються з бази даних через REST API;
- на стороні клієнта дані обробляються JavaScript-методами;
- на основі оброблених даних будується графік у canvas-елементі.

Використання Chart.js дозволяє візуалізувати інформацію, що полегшує користувачеві розуміння змін у власному емоційному стані та мотивує до регулярного введення даних.

```
// Створюємо графік
const moodChart = new Chart(ctx, {
  type: 'line',
  data: {
    labels: moodData.map(entry => entry.formattedDate),
    datasets: [{
      label: 'Настрої за вибраний період',
      data: moodData.map(entry => entry.mood),
      borderColor: '#49c2bb',
      backgroundColor: 'rgba(135, 206, 250, 0.5)',
      fill: true,
      tension: 0.4,
    }]
  },
  options: {
    scales: {
      y: {
        ticks: {
          callback: function(value) {
            if (value === 3) return '😊 Радісний';
            if (value === 2) return '😐 Нейтральний';
            if (value === 1) return '😞 Сумний';
          }
        },
        beginAtZero: true,
        suggestedMax: 3
      }
    }
  }
});
```

Рис. 2.1 Реалізація графіка через Chart.js

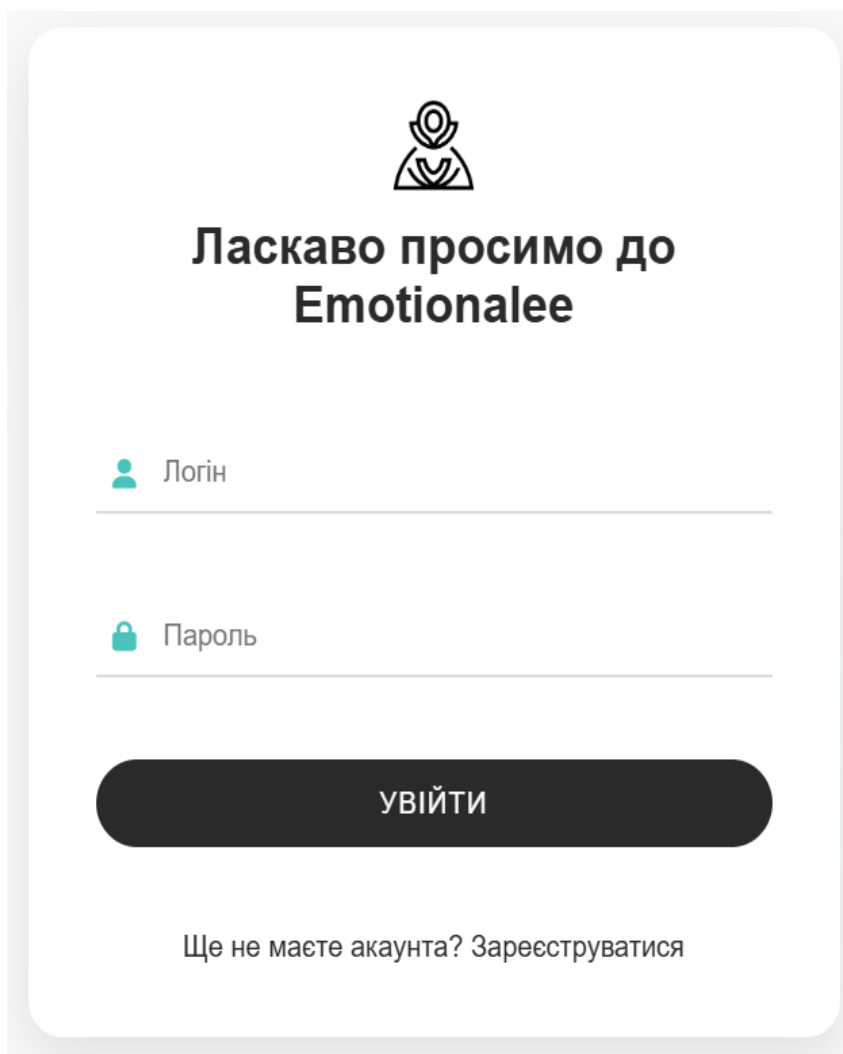
2.5 Використання Spring Security для авторизації

Для забезпечення захисту персональних даних та управління доступом до функціоналу системи реалізовано механізм аутентифікації та авторизації на основі фреймворку Spring Security [12].

Основні можливості:

- реєстрація нових користувачів;
- аутентифікація за допомогою логіна і пароля;
- захист REST API-ендпойнтів (доступ тільки авторизованим користувачам);
- управління сесіями користувачів;
- шифрування паролів (алгоритм BCrypt).

Використання Spring Security забезпечує високий рівень безпеки взаємодії та дозволяє гнучко керувати правами доступу до ресурсів системи.



The image shows a login form for a system named "Emotionalee". At the top, there is a logo of a person with a gear-like head. Below the logo, the text "Ласкаво просимо до Emotionalee" is displayed. The form contains two input fields: "Логін" (Login) and "Пароль" (Password), each with a corresponding icon (a person and a lock respectively). Below these fields is a large, dark button labeled "УВІЙТИ" (Login). At the bottom, there is a link that says "Ще не маєте акаунта? Зареєструватися" (Don't have an account? Register).

Рис. 2.2 Форма авторизації

3 ПРОЄКТУВАННЯ WEB-ЗАСТОСУНКУ

Ефективна робота системи для підтримки емоційного стану користувачів значною мірою залежить від правильної архітектури, що забезпечує надійну, гнучку і масштабовану взаємодію між компонентами. У цьому розділі розглянуто архітектурні рішення, діаграми варіантів використання, діаграми класів, модель бази даних та макети інтерфейсу системи.

3.1 Архітектура системи

Розроблений web-застосунок побудовано за клієнт-серверною архітектурою з використанням архітектурного стилю REST (Representational State Transfer). Це дозволяє чітко розмежувати клієнтську та серверну частини, забезпечити масштабованість і спрощення інтеграції зовнішніх сервісів [13].

Основні компоненти системи:

- Клієнтська частина (Front-end)

Розроблена на основі HTML, CSS, JavaScript. Забезпечує користувацький інтерфейс, збір і візуалізацію даних, а також відправку запитів до серверу [21].

- Серверна частина (Back-end)

Реалізована на базі Java Spring Boot. Відповідає за бізнес-логіку застосунку, взаємодію з базою даних, обробку запитів клієнта, інтеграцію з API ChatGPT та забезпечення аутентифікації й авторизації [22].

- База даних

Використовується MySQL для збереження інформації про користувачів, історію чатів, записи настрою, налаштування системи [23].

- Зовнішній сервіс — API ChatGPT (OpenAI)

Інтеграція із зовнішнім API дозволяє реалізувати інтелектуального асистента на основі ШІ [25].

- Web Speech API

Використовується на стороні клієнта для голосового введення тексту.

- Сховище файлів (локально/на сервері)

Призначене для тимчасового зберігання завантажених користувачем текстових файлів.

Взаємодія між компонентами здійснюється за допомогою REST API через HTTP-запити. Для захисту даних і контролю доступу використовується Spring Security.

3.2 Структура класів системи

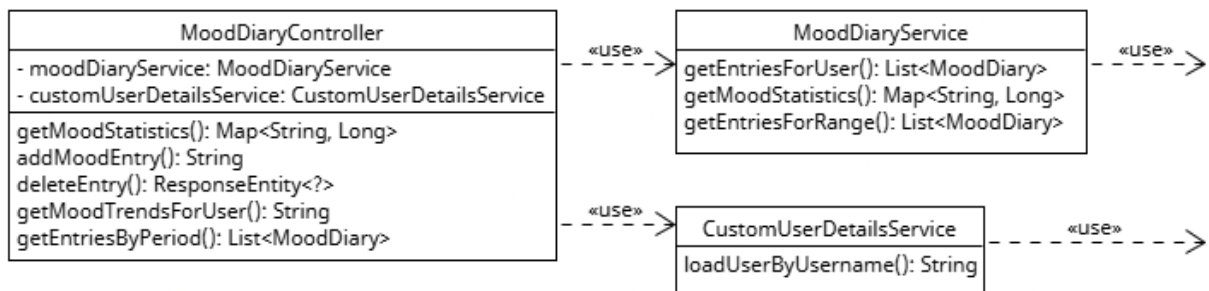


Рис. 3.1 Діаграма класів. Частина 1: контроллер і сервіси

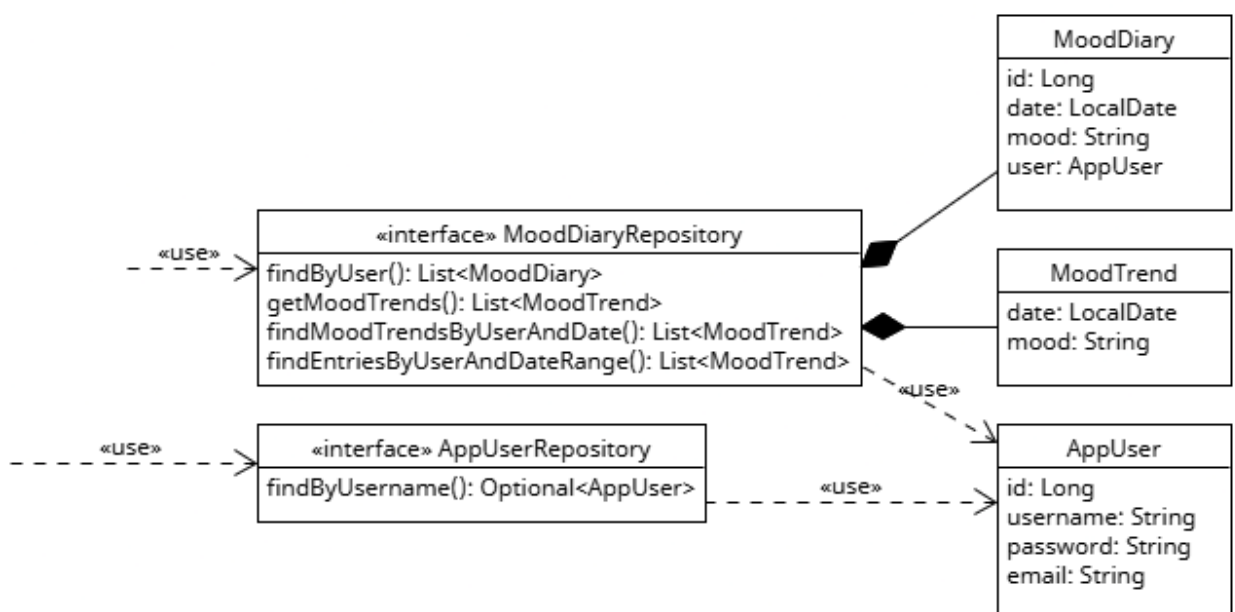


Рис. 3.2 Діаграма класів. Частина 2: репозиторії та моделі

Діаграма класів відображає архітектуру модуля управління щоденником настрою в додатку, побудовану за принципом багатошарової моделі з чітким розподілом обов'язків між компонентами.

На рівні контролерів розташований клас `MoodDiaryController`, який взаємодіє з сервісами `MoodDiaryService` і `CustomUserDetailsService`. Контролер обробляє запити користувача, викликаючи методи сервісів, такі як `getMoodStatistics()` для отримання статистики настрою у вигляді мапи (`Map<String, Long>`), `addMoodEntry()` для додавання нового запису, `deleteEntry()` для видалення, `getMoodTrendsForUser()` для аналізу трендів настрою та `getEntriesByPeriod()` для вибірки записів за періодом.

`MoodDiaryService` виступає центральним елементом бізнес-логіки, координуючи взаємодію з репозиторіями та обробляючи запити контролера. Він використовує методи `getEntriesForUser()`, `getMoodStatistics()` і `getEntriesForRange()` для роботи з даними настрою, а також делегує виклик `loadUserByUsername()` до `CustomUserDetailsService` для отримання даних користувача за його ім'ям.

На рівні даних `MoodDiaryRepository` забезпечує доступ до об'єктів `MoodDiary` через методи `findByUser()`, `getMoodTrends()`, `findMoodTrendsByUserAndDate()` і `findEntriesByUserAndDateRange()`. Репозиторій працює з класом `MoodDiary`, який містить поля `id` (`Long`), `date` (`LocalDate`), `mood` (`String`) і посилання на користувача `user` (`AppUser`). Дані користувача зберігаються в `AppUserRepository`, звідки через метод `findByUsername()` отримуються об'єкти `AppUser` з полями `id` (`Long`), `username` (`String`), `password` (`String`) і `email` (`String`). Для аналізу трендів використовується клас `MoodTrend` з полями `date` (`LocalDate`) і `mood` (`String`).

Така архітектура забезпечує модульність і гнучкість: контролер ізолює логіку представлення, сервіси обробляють бізнес-правила, а репозиторії відповідають за доступ до даних. Взаємодія між шарами чітко структурована через інтерфейси, що полегшує масштабування та підтримку системи.

3.3 Компоненти застосунку

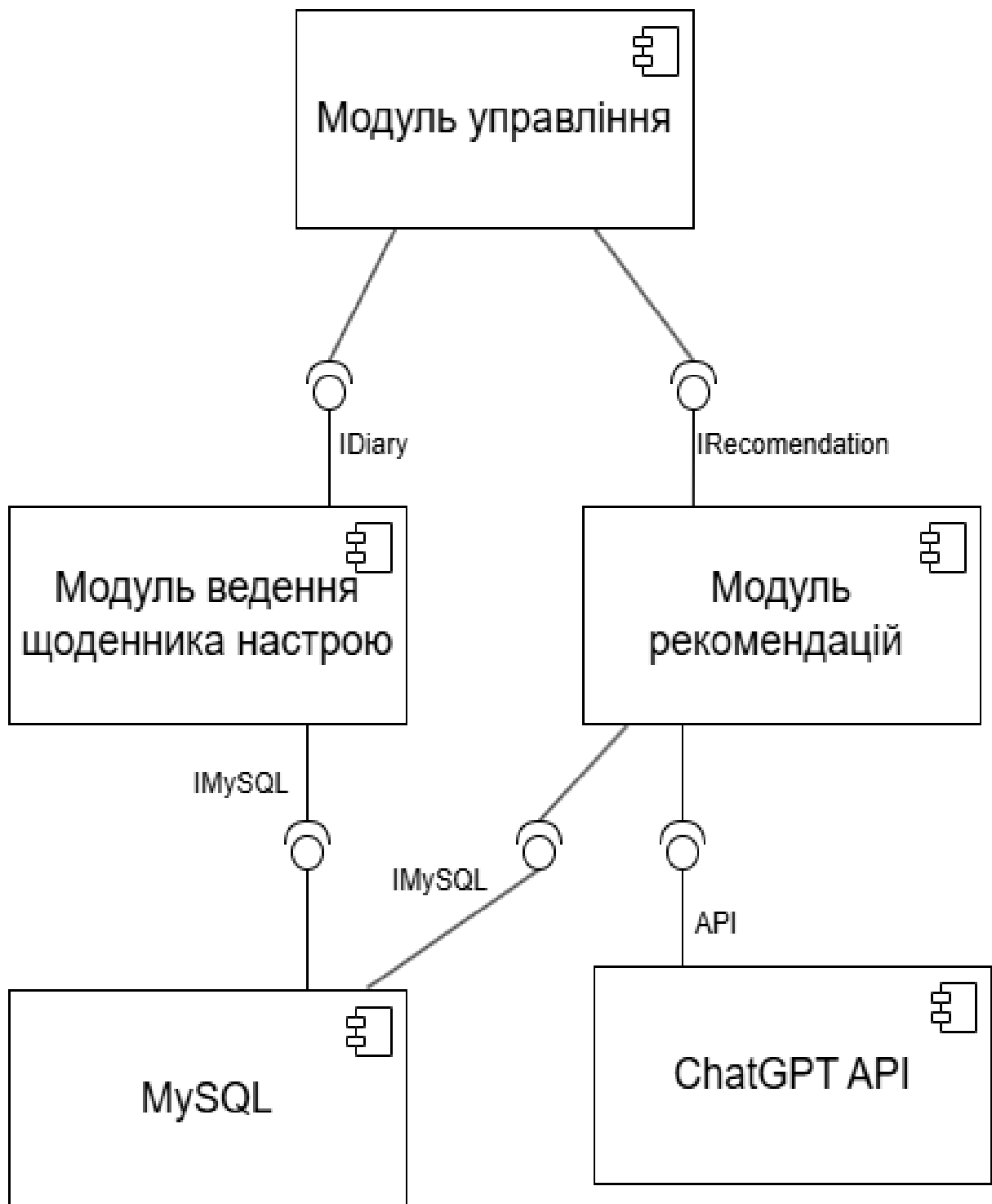


Рис. 3.2 Діаграма компонентів

Діаграма показує, як різні частини системи працюють разом.

Ось як це виглядає простіше:

1. Модуль управління – це ніби серце системи, яке з'єднує всі інші частини і керує їхньою роботою.
2. Модуль щоденника настрою – відповідає за те, щоб записувати і аналізувати, як почувається користувач. Він спілкується з модулем управління через інтерфейс IDiary і відправляє дані в базу MySQL через IMySQL.
3. Модуль рекомендацій – створює персональні поради для користувача. Він бере дані від модуля управління (IRecomendation) і бази MySQL (IMySQL), а ще підключається до ChatGPT API через спеціальний інтерфейс, щоб робити поради ще кращими.
4. MySQL – це база даних, де зберігається вся інформація від щоденника настрою і модуля рекомендацій.
5. ChatGPT API – зовнішній сервіс, який допомагає модулю рекомендацій давати точніші й корисніші поради.

Все працює злагоджено, щоб користувачу було зручно!

Основні переваги обраної архітектури:

1. Масштабованість: клієнт і сервер можуть розвиватись незалежно.
2. Гнучкість: легко інтегрувати нові сервіси або змінювати існуючі.
3. Безпека: чітке розділення відповідальності дозволяє краще контролювати доступ до даних.
4. Продуктивність: мінімізація навантаження на сервер за рахунок обробки частини логіки на клієнті.

3.4 Моделювання діяльності

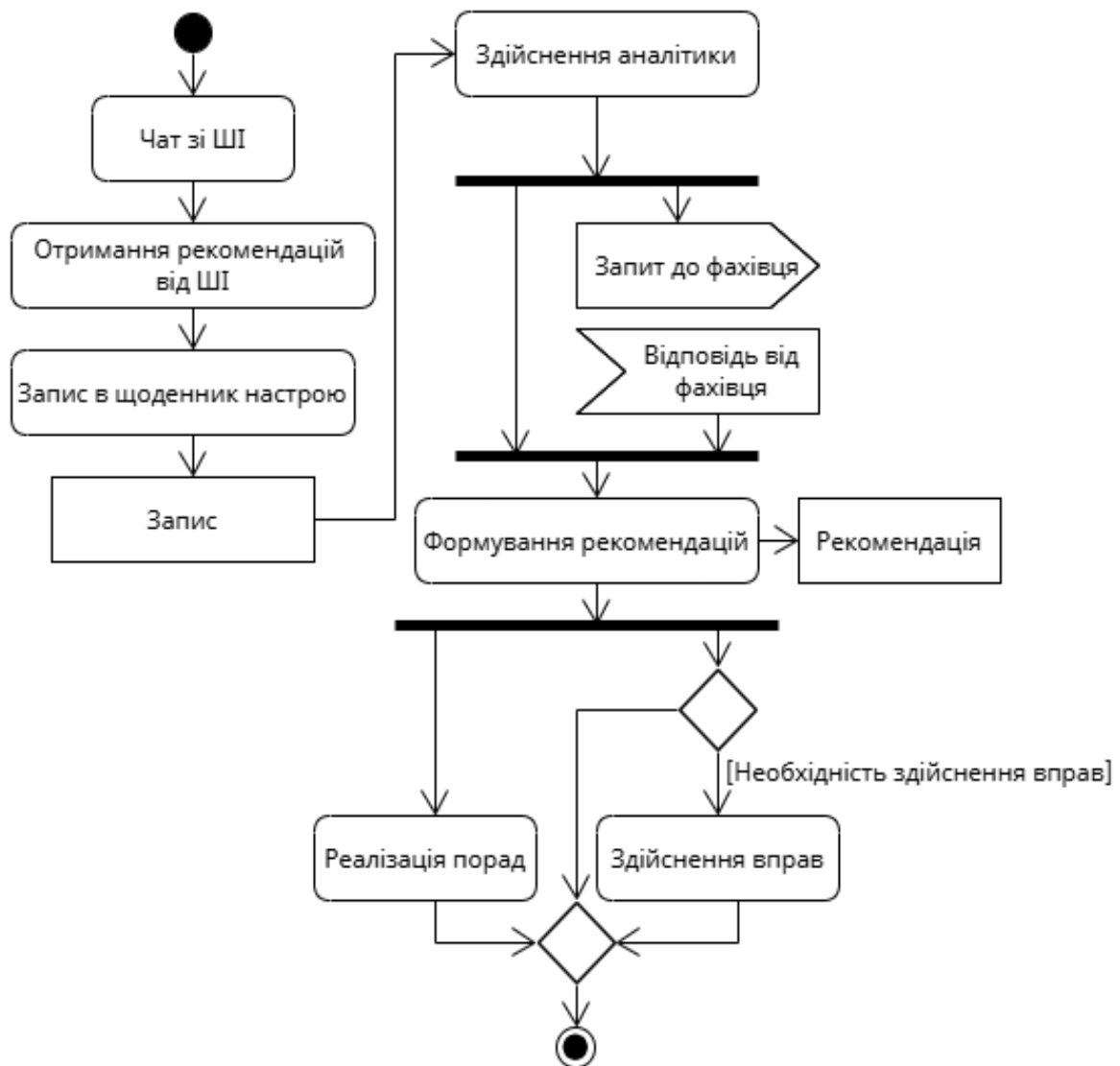


Рис. 3.3 Діаграма діяльності

Ця схема пояснює, як людина може отримувати поради від розумної системи та використовувати їх для покращення свого стану. Ось як це працює, крок за кроком:

- Чат із системою: людина звертається до розумної програми, щоб отримати пораду.
- Отримання порад: система уважно аналізує запит і пропонує ідеї, які можуть допомогти.

- Запис у щоденник настрою: людина фіксує, як почувається після того, як отримала пораду.
- Збереження даних: усі записи зберігаються, щоб потім їх можна було проаналізувати.
- Аналіз інформації: система вивчає зібрані дані, щоб зрозуміти, наскільки поради виявилися корисними.
- Звернення до фахівця: якщо потрібна додаткова допомога, людина може проконсультуватися з експертом.
- Відповідь фахівця: спеціаліст ділиться професійною думкою і дає свої рекомендації.
- Оновлення порад: на основі аналізу і думки фахівця створюються нові, ще кращі поради.
- Нова рекомендація: людина отримує оновлену пораду, яка допомагає покращити самопочуття.
- Оцінка вправ: система визначає, чи потрібно виконувати якісь конкретні дії, наприклад, вправи.
- Застосування порад: людина починає використовувати отримані ідеї на практиці.
- Виконання вправ: практичні дії, які допомагають відчувати себе краще.

Ця схема показує, як розумна система, фахівці та власні дії людини працюють разом, щоб досягти найкращого результату!

3.5 Моделювання функціональності

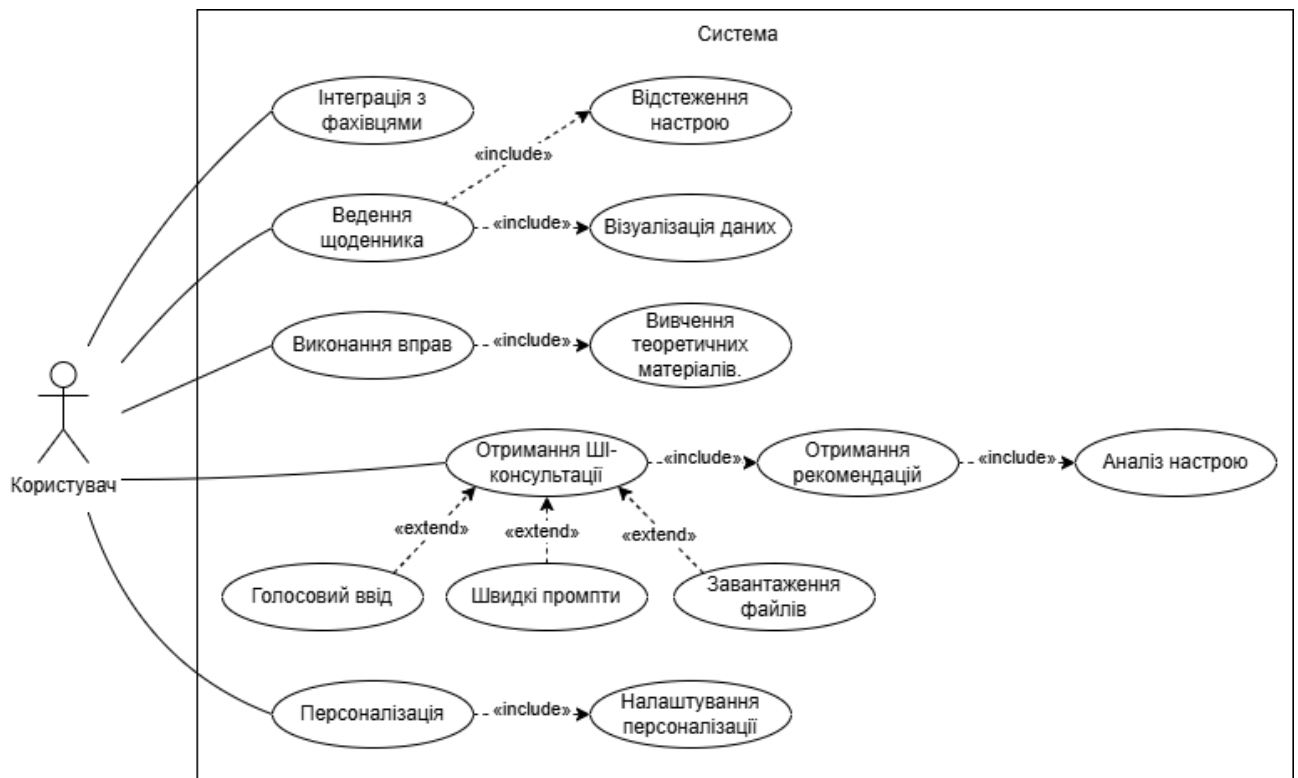


Рис. 3.4 Діаграма прецедентів

Ця схема показує, як людина може використовувати систему і що вона може з нею робити. Ось як це виглядає простою мовою:

1. Користувач – головна людина, яка працює з системою.
2. Запит порад – людина просить систему або спеціаліста підказати, що робити.
3. Отримання порад – система пропонує ідеї, які підходять саме цій людині, спираючись на її дані.
4. Запис настрою – людина записує, як почувається, щоб система краще розуміла її стан.
5. Аналіз записів – система переглядає ці записи, щоб зробити поради ще кориснішими.
6. Звернення до спеціаліста – якщо потрібна допомога, людина може поспілкуватися з експертом.

7. Відповідь спеціаліста – експерт уважно вивчає ситуацію і дає свої рекомендації.
8. Оновлення порад – система поєднує аналіз записів і думки спеціаліста, щоб створити нові, кращі поради.
9. Рішення про дії – людина вирішує, чи варто виконувати запропоновані дії, наприклад, якісь вправи.
10. Виконання дій – якщо поради включають практичні кроки, людина їх робить, щоб почуватися краще.

Ця схема допомагає зрозуміти, як система підтримує людину, щоб їй було зручно і корисно!

3.6 Логічна архітектура

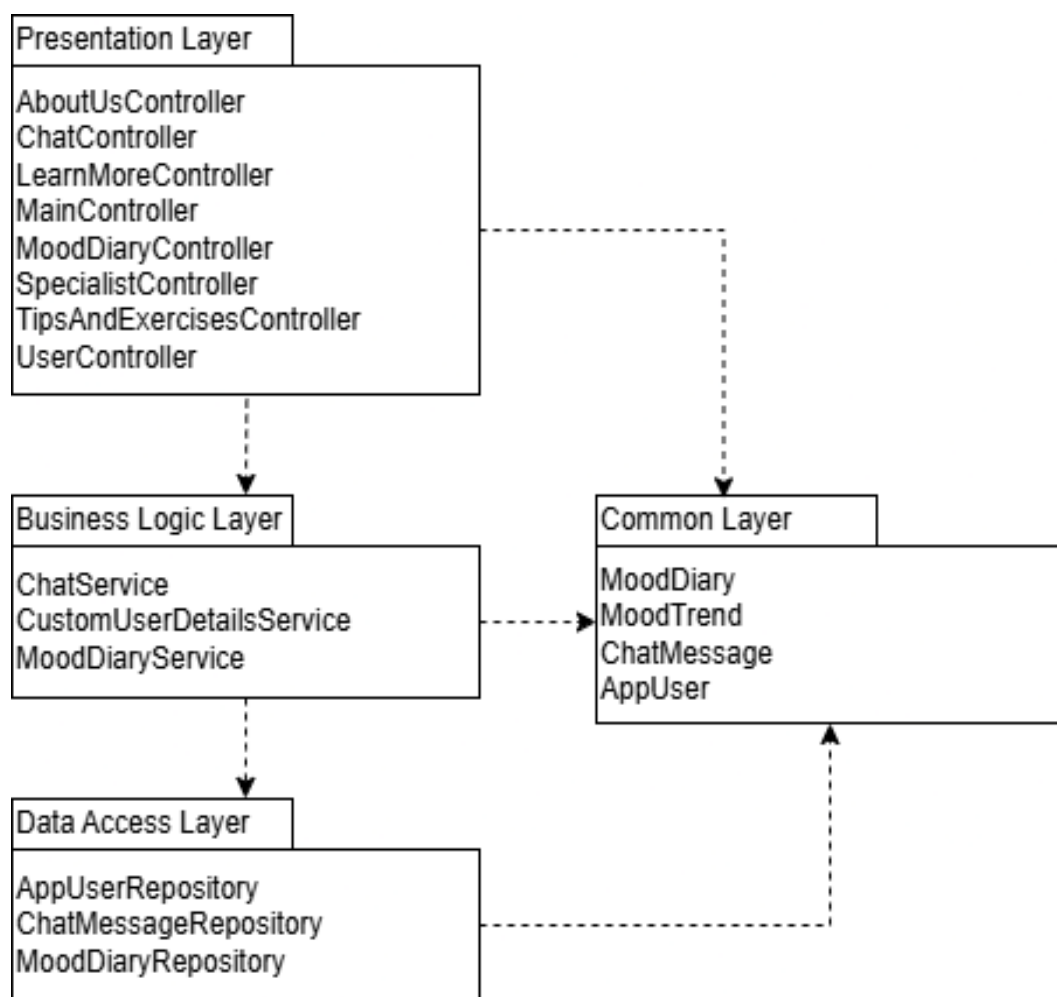


Рис. 3.5 Діаграма пакетів

Діаграма, представлена на рисунку 3.7, ілюструє архітектуру пакетів програмного забезпечення, побудовану за принципом багатoshарової моделі.

Вона складається з чотирьох основних рівнів, які взаємодіють між собою для забезпечення функціональності системи.

На верхньому рівні розташований Presentation Layer (Рівень представлення), що включає контролери, такі як AboutUsController, ChatController, LearnMoreController, MainController, MoodDiaryController, SpecialistController, TipsAndExercisesController та UserController. Цей рівень відповідає за взаємодію з користувачем, обробку запитів та відображення даних.

Нижче розміщений Business Logic Layer (Рівень бізнес-логіки), який містить сервіси ChatService, CustomUserDetailsService та MoodDiaryService. Цей рівень обробляє бізнес-логіку додатку, координуючи взаємодію між рівнем представлення та нижніми рівнями.

Data Access Layer (Рівень доступу до даних) включає репозиторії AppUserRepository, ChatMessageRepository та MoodDiaryRepository, що забезпечують доступ до даних, їх збереження та отримання з бази даних.

Окремо виділено Common Layer (Загальний рівень), який містить класи MoodDiary, MoodTrend, ChatMessage та AppUser. Цей рівень забезпечує загальні моделі даних, які використовуються іншими рівнями для уніфікації обробки інформації.

Взаємодія між рівнями відбувається через чітко визначені зв'язки: Presentation Layer звертається до Business Logic Layer, який, у свою чергу, взаємодіє з Data Access Layer. Common Layer забезпечує обмін даними між усіма рівнями, створюючи основу для їхньої інтеграції.

Така структура сприяє модульності, спрощує підтримку та масштабування системи.

3.7 Макети інтерфейсу

На етапі проєктування було створено прототипи основних екранів системи для забезпечення зручності користувача та логічної структури інтерфейсу. Прототипування виконувалося у Figma.

Основні сторінки:

1. Головна сторінка — короткий опис системи, кнопки “Дізнатися більше” і “Розпочати чат”
2. Чат із ШІ-асистентом — діалогове вікно, швидкі промпти, кнопки завантаження файлів і голосового введення
3. Щоденник настрою — форма введення емоцій, перегляд історії та графіків
4. Поради та ресурси — добірки статей, практик тощо.
5. Особистий кабінет — налаштування акаунту, перегляд історії чатів, завантаження звітів

3.8 Розгортання застосунку

Схема, зображена на рисунку 3.6, пояснює як влаштована робота веб-застосунку і як його частини взаємодіють між собою.

Ось просте пояснення:

- Комп’ютер — це пристрій, з якого людина заходить у додаток. Вона відкриває браузер і через нього спілкується з сервером, використовуючи безпечне з’єднання (HTTP/HTTPS).
- Сервер — головний центр, який приймає запити від людини, обробляє їх, робить потрібні обчислення і працює з базою даних, щоб зберігати або отримувати інформацію.
- Хмарний сервер — додатковий помічник, який бере на себе окремі завдання, наприклад, зв’язок із зовнішніми сервісами через API або обробку складних запитів.

Як усе працює: комп'ютер надсилає запити до сервера через інтернет, а сервер, якщо потрібно, звертається до хмарного сервера, щоб отримати додаткові дані чи функції.

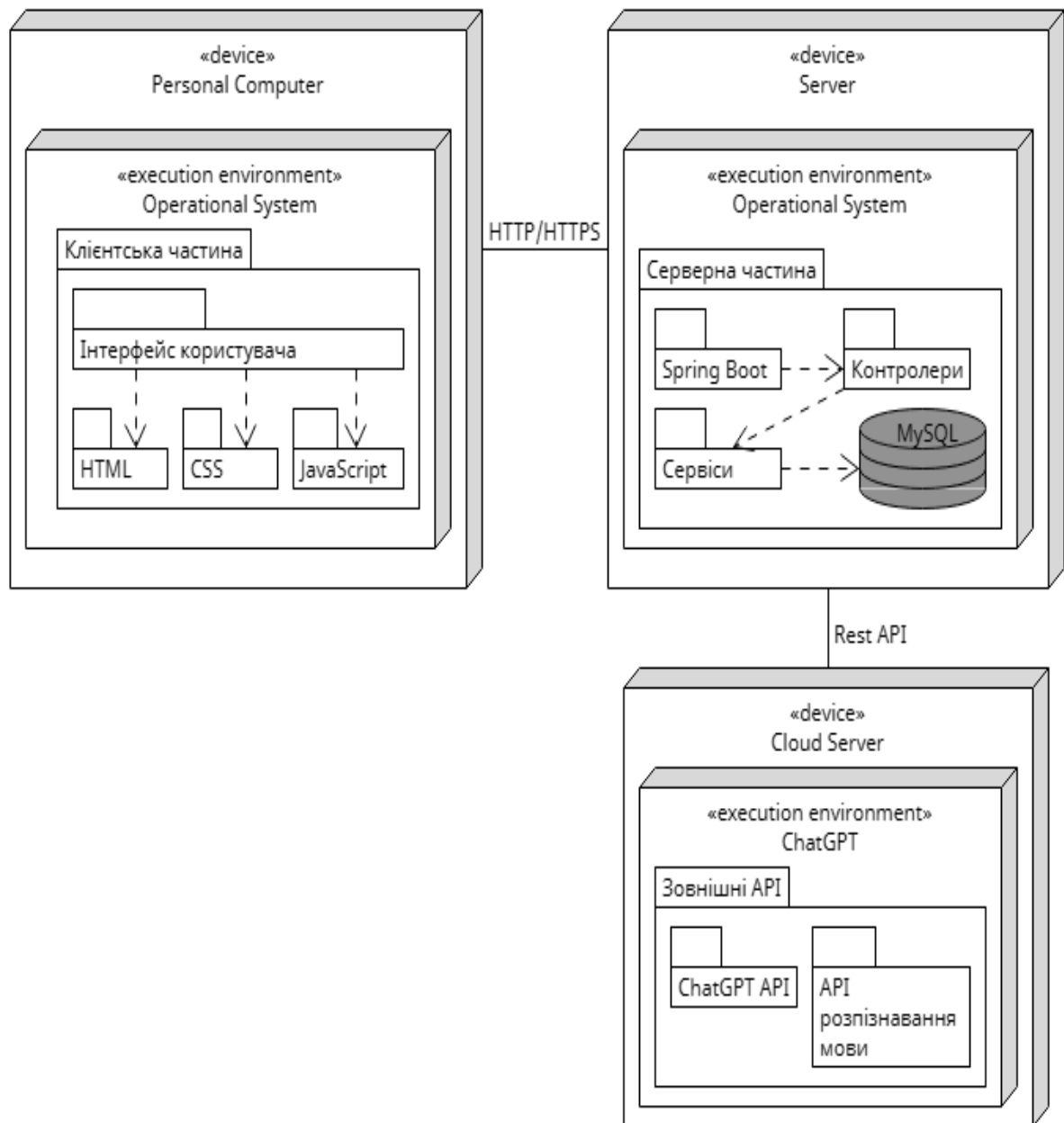


Рис. 3.6 Діаграма розгортання

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

У цьому розділі детально описано реалізовані функціональні можливості web-застосунку, технічні аспекти їхньої роботи та результати тестування системи.

4.1 Опис реалізованих функціональних можливостей

Розроблений застосунок надає користувачам широкий спектр функцій для підтримки емоційного стану.

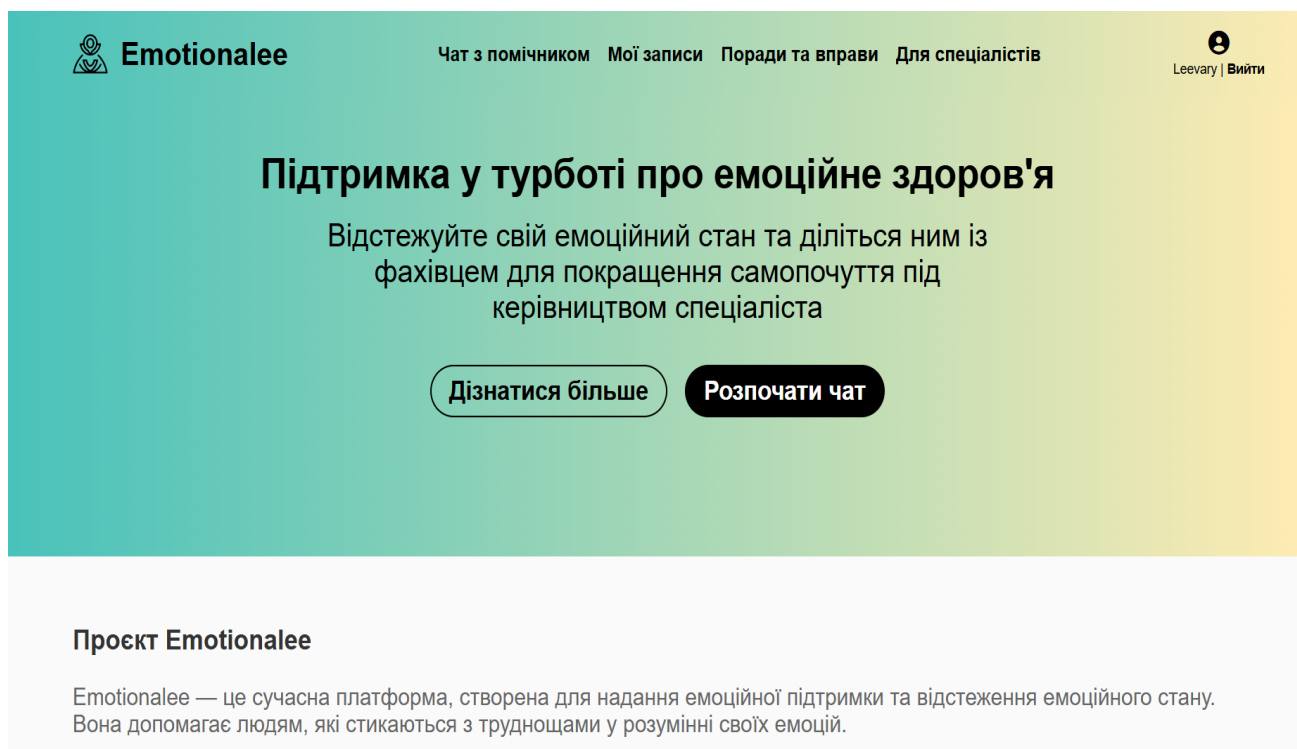


Рис. 4.1 Головна сторінка

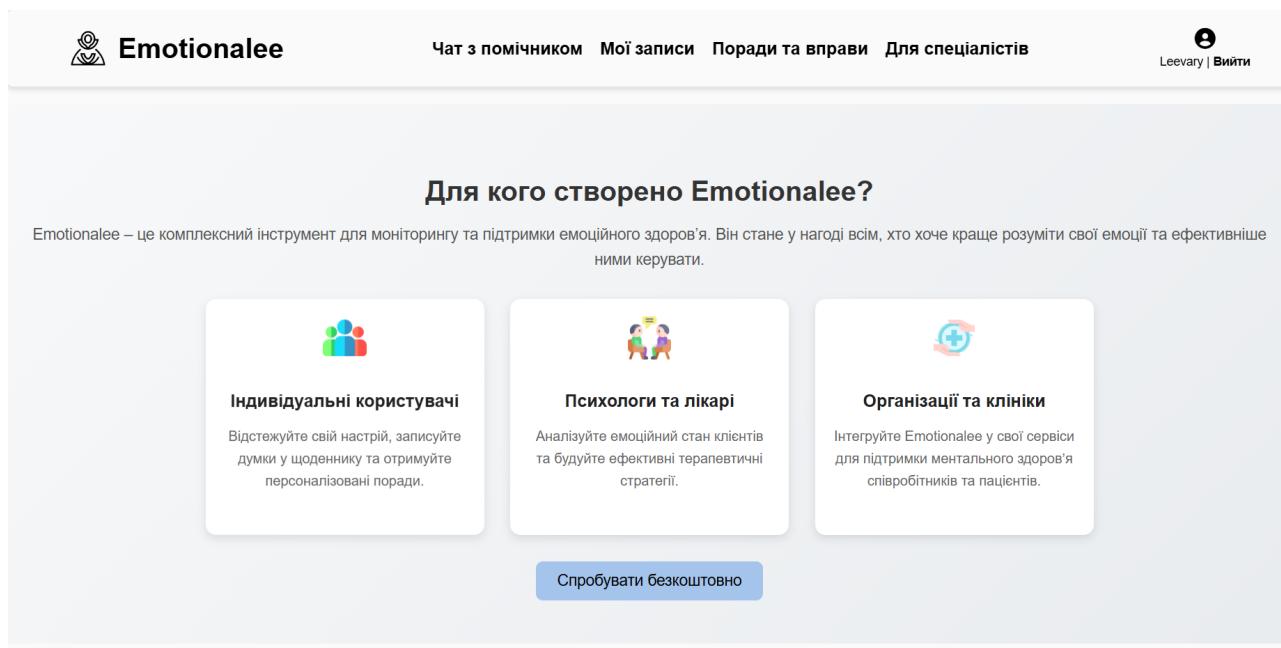


Рис. 4.2 Інформаційна секція пояснює, для кого створена наша платформа

4.1.1 Взаємодія з ШІ-асистентом

Одна з основних функціональних можливостей системи — інтерактивний чат із інтелектуальним асистентом на основі API ChatGPT. Цей модуль дозволяє користувачам отримувати емоційну підтримку, поради та рекомендації у зручному чат-форматі.

Функціонал реалізовано таким чином:

- Чат із ШІ-асистентом на основі ChatGPT

Користувач може спілкуватися із системою шляхом введення текстових повідомлень. Асистент надає відповіді у форматі дружнього діалогу, підтримує бесіду, ставить уточнюючі запитання, пропонує практичні поради для покращення емоційного стану. Відповіді формуються на основі аналізу введеного тексту користувача і контексту попередньої розмови.

- Підтримка швидких промтів (порада, мотивація, запит на кшталт: “розвесели мене”)

Для зручності користувача реалізовано набір готових запитів, які можна швидко обрати натисканням однієї кнопки. Наприклад: “Дай

пораду на день”, “Розвесели мене”, “Надихни мене”, “Що робити, якщо мені сумно?”. Це дозволяє отримати корисну інформацію або підтримку навіть без потреби формулювати запит самостійно.

- Голосовий ввід (Web Speech API)

Функція голосового введення дає змогу користувачу надиктувати повідомлення замість набору тексту на клавіатурі. Це особливо корисно у ситуаціях, коли користувач має емоційне напруження або бажає швидко поділитися думками. Після натискання кнопки мікрофона система починає розпізнавати мову і автоматично перетворює голос на текст для подальшої обробки.

- Можливість почати новий чат

Для збереження конфіденційності або зміни тематики розмови реалізовано функцію створення нового чату. При натисканні кнопки “Новий чат” очищується історія поточної сесії, а ШІ-асистент починає спілкування “з чистого листа”. Це забезпечує чітке розмежування різних діалогів і спрощує управління контекстом бесіди.

Взаємодія з асистентом відбувається у зрозумілому, інтуїтивному інтерфейсі. Завдяки інтеграції API ChatGPT діалоги є природними, контекстно релевантними та дружніми. Це сприяє створенню позитивного користувацького досвіду та ефективній емоційній підтримці [15].

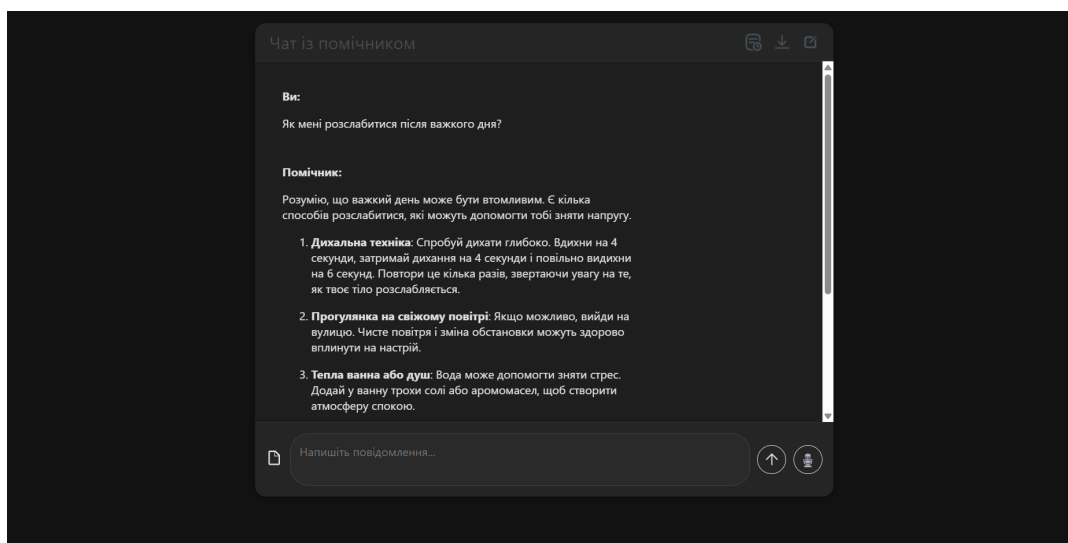


Рис. 4.3 Спілкування з ШІ-помічником

4.1.2 Послідовність взаємодії в чат-системі

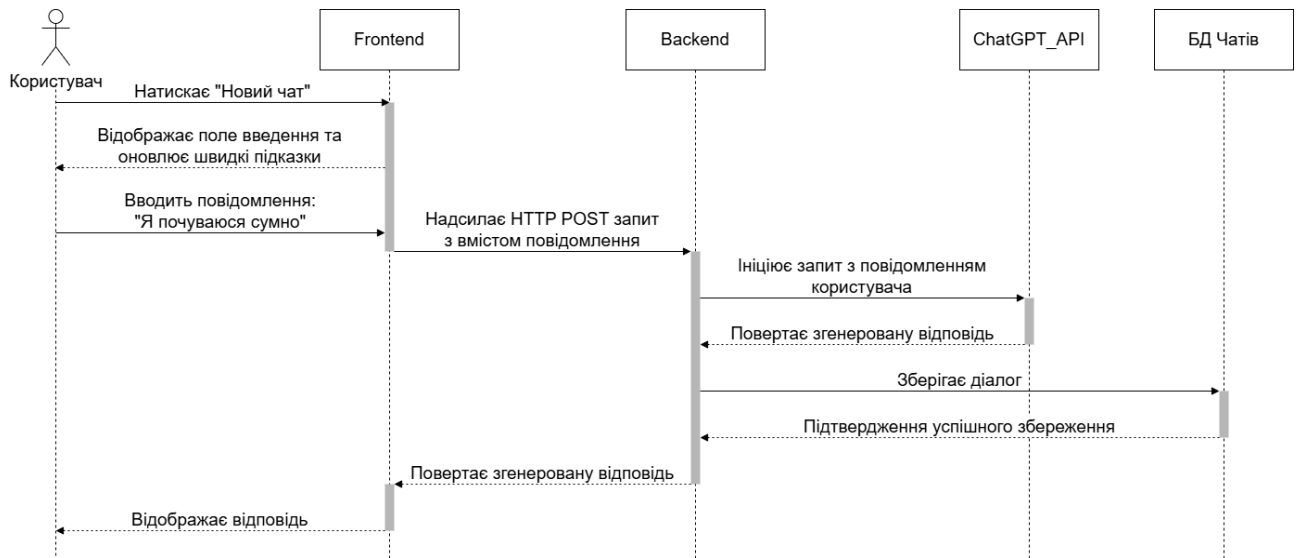


Рис. 4.4 Діаграма послідовності

Діаграма послідовності ілюструє взаємодію користувача з системою чату, побудовану на основі клієнт-серверної архітектури з інтеграцією зовнішнього API ChatGPT. Процес починається з користувача, який через фронтенд-інтерфейс надсилає запит "Новий чат" або вводить повідомлення та обирає опцію "Я почуваюся сумно". Цей запит передається на бекенд через HTTP POST-запит із вбудованим повідомленням. Бекенд обробляє запит, передаючи його до ChatGPT API, яке генерує відповідь на основі введених даних. Отримана відповідь повертається до бекенду, де вона адаптується для подальшої обробки, включаючи аналіз емоційного стану (наприклад, "Повертає згенеровану відповідь"). Потім бекенд зберігає отримані дані в базі даних (БД Чатів) та повертає згенеровану відповідь користувачу через фронтенд. Цикл завершується, коли користувач отримує відповідь і має можливість продовжити діалог або переглянути історію.

Така послідовність підкреслює чіткий розподіл ролей між фронтендом, бекендом і зовнішнім API, забезпечуючи ефективну обробку та збереження даних чату.

4.1.3 Щоденник настрою

Щоденник настрою — один із ключових модулів застосунку, який дозволяє користувачеві фіксувати, аналізувати та відстежувати зміни власного емоційного стану протягом часу. Цей інструмент сприяє підвищенню емоційної грамотності, саморефлексії та своєчасному виявленню змін у психоемоційному стані.

Основні функції модуля:

- Введення записів настрою (вибір емоції, нотатки)

Користувач має можливість щоденно зазначати свій емоційний стан, обираючи одну з доступних категорій (наприклад: радість, сум або спокій). Для більш детальної фіксації можна додати текстову нотатку — опис подій чи думок, що вплинули на настрій. Це дозволяє створити повноцінну хроніку емоційних змін.

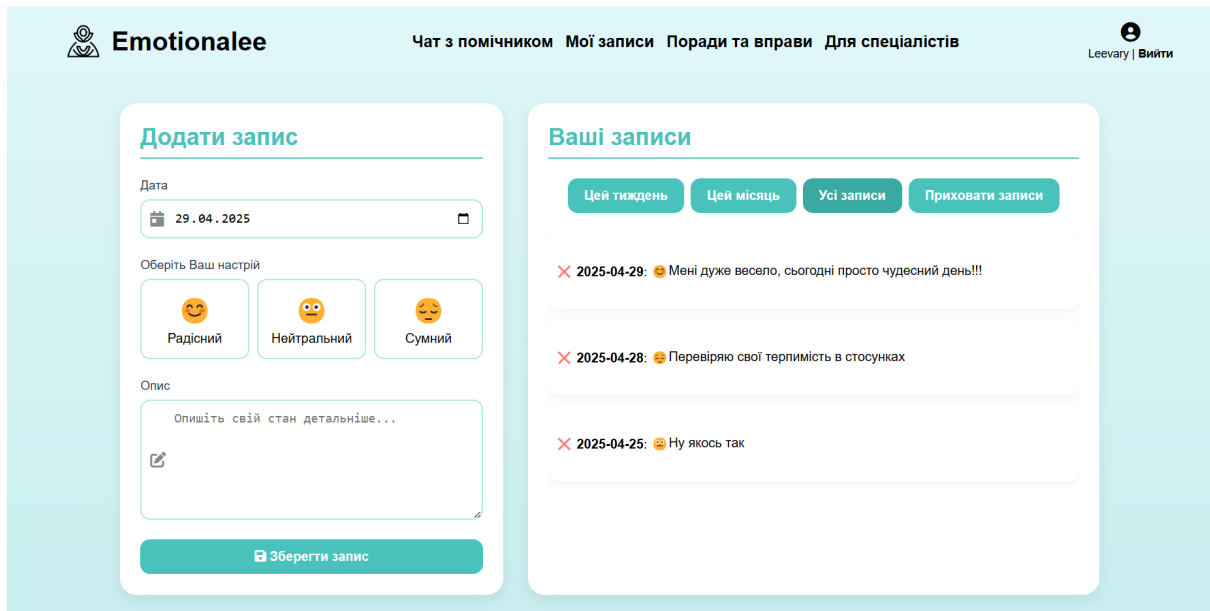
- Перегляд історії записів

Усі внесені записи автоматично зберігаються у базі даних і доступні для перегляду в особистому кабінеті користувача. Інтерфейс відображає список попередніх записів із зазначенням дати, емоції та нотаток. Це дає змогу користувачеві відстежувати зміни настрою, аналізувати повторювані патерни та робити висновки про фактори, що впливають на емоційний стан.

- Побудова графіків динаміки настрою

Для візуалізації емоційних змін реалізовано інтерактивні графіки на основі бібліотеки Chart.js. Користувач може обрати період (тиждень, місяць, рік) і переглянути динаміку настрою у вигляді лінійного графіка або гістограми. При наведенні на точки графіка виводиться детальна інформація про конкретний запис. Така візуалізація полегшує аналіз емоційних тенденцій та сприяє підвищенню усвідомленості користувача щодо власного стану.

Щоденник настрою є важливим інструментом самоспостереження, що дозволяє своєчасно виявляти негативні тенденції, а також слугує джерелом даних для аналізу психологами (у разі використання застосунку у терапевтичних цілях) [16].



The screenshot displays the Emotionalee app interface. The top navigation bar includes the app logo, a chat icon, and links for 'Чат з помічником', 'Мої записи', 'Поради та вправи', and 'Для спеціалістів'. The user is logged in as 'Leevary | Вийти'.

The main content area is divided into two panels:

- Додати запис (Add entry):** This panel contains a date selector set to '29.04.2025'. Below it, users can select their mood from three options: 'Радісний' (Happy), 'Нейтральний' (Neutral), and 'Сумний' (Sad). A text area labeled 'Опис' (Description) allows users to 'Опишіть свій стан детальніше...' (Describe your state in more detail...). A green button at the bottom says 'Зберегти запис' (Save entry).
- Ваші записи (Your entries):** This panel shows a list of recent entries. It includes filters for 'Цей тиждень' (This week), 'Цей місяць' (This month), 'Усі записи' (All entries), and 'Приховати записи' (Hide entries). The entries are:
 - 2025-04-29: 😊 Мені дуже весело, сьогодні просто чудесний день!!!
 - 2025-04-28: 😊 Перевіряю свої терпимість в стосунках
 - 2025-04-25: 😊 Ну якось так

Рис. 4.5 Додавання нотаток у щоденник настрою

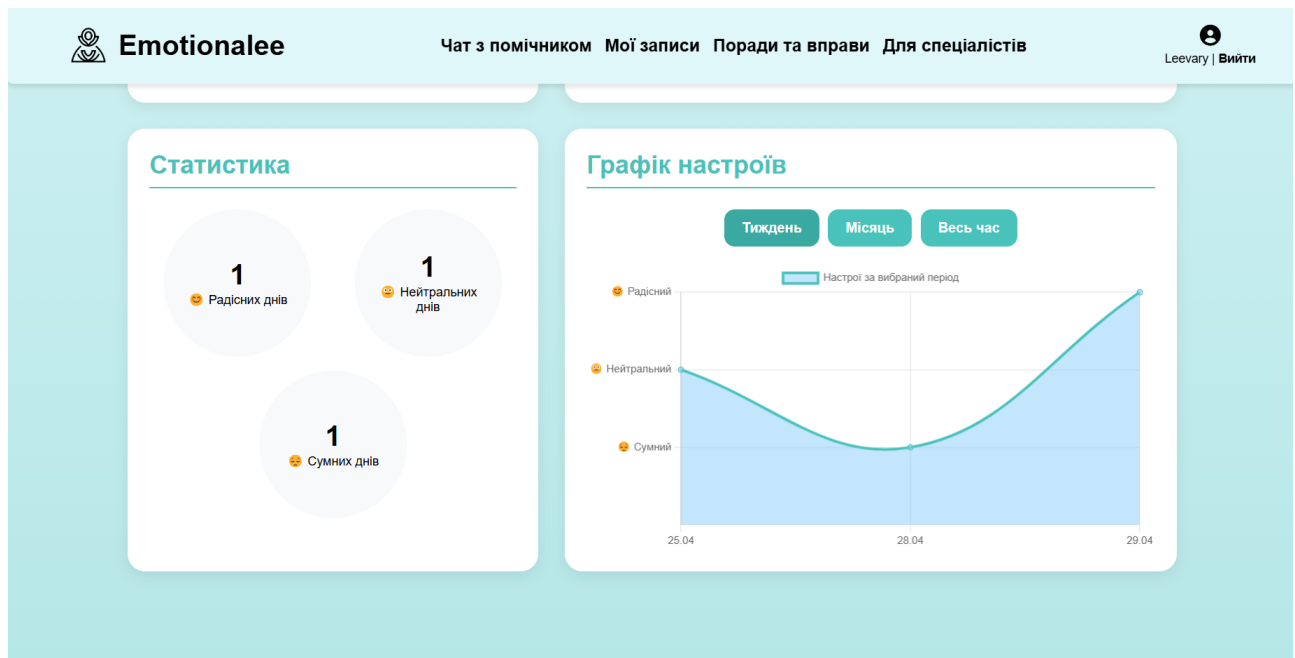


Рис. 4.6 Реалізований графік та статистика настроїв

4.1.4 Поради та ресурси

Модуль “Поради та ресурси” є важливою складовою системи, оскільки надає користувачам перевірену інформацію та практичні матеріали для підтримки психоемоційного стану.

Основні можливості:

Добірка статей, відео, вправ.

У системі створено бібліотеку тематичних матеріалів, які містять статті про ментальне здоров'я, інструкції з технік релаксації, медитації, дихальні вправи, рекомендації для боротьби зі стресом і тривожністю.

Матеріали згруповані за тематикою (профілактика стресу, подолання тривожності, мотивація, психоемоційна самодопомога тощо) [17]. Це дозволяє користувачам розширювати свої знання та застосовувати практичні методики у повсякденному житті.

Сортування та пошук матеріалів за категоріями

Реалізовано можливість вибору матеріалів за типом та тематикою.

Завдяки цьому модулю користувачі отримують доступ до надійних джерел знань і можуть самостійно підвищувати рівень емоційної компетентності [18].

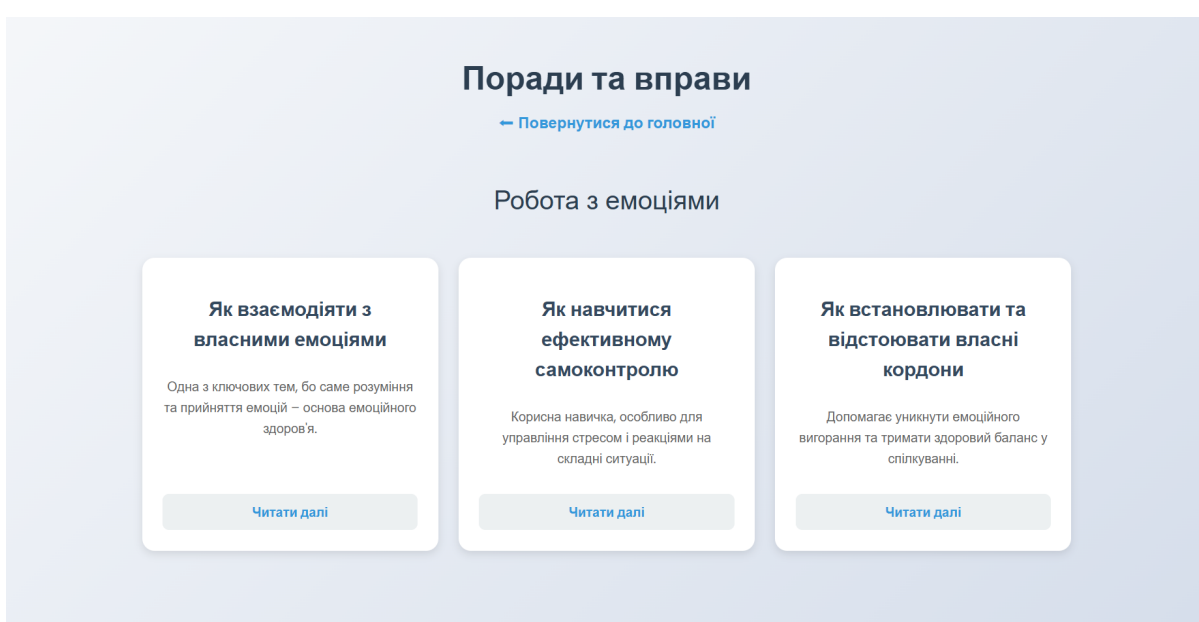


Рис. 4.7 Сторінка з порадами та вправами

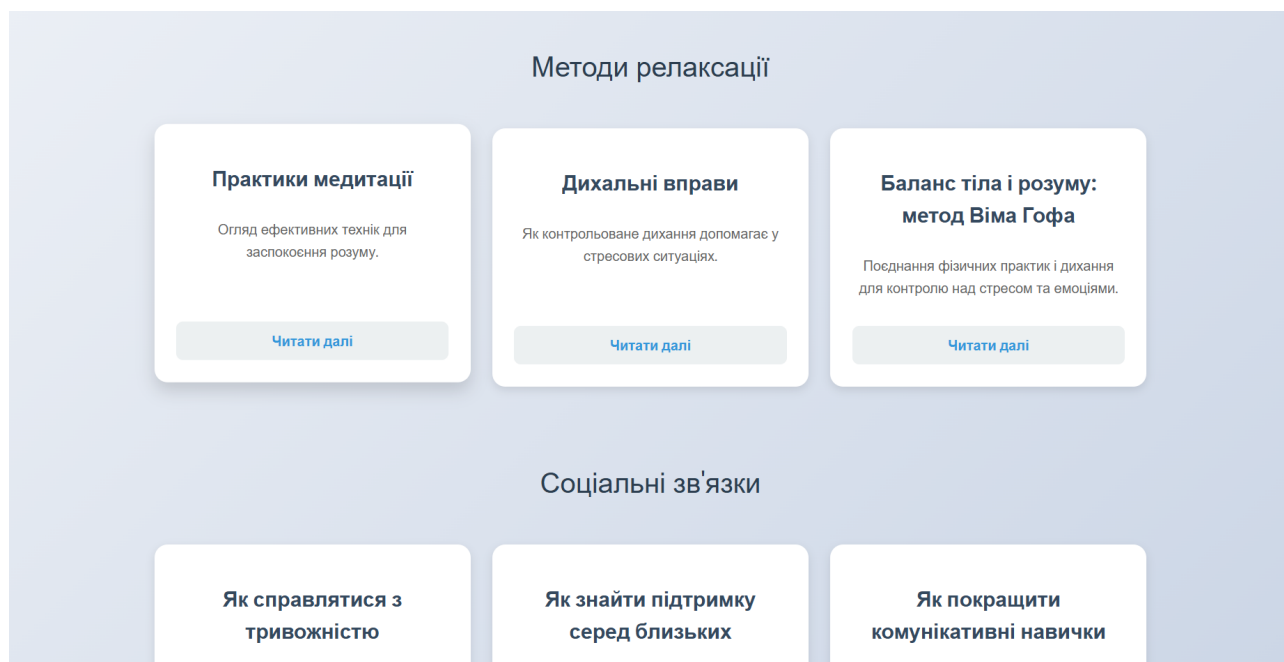


Рис. 4.8 Продовження Порад та вправ по різних розділах

4.1.5 Персоналізація

Персоналізація— це розділ системи, який надає користувачеві можливість керувати своїм профілем, налаштуваннями і даними.

Основні функції:

1. Перегляд історії чатів

Усі сесії спілкування з ШІ-асистентом автоматично зберігаються. У особистому кабінеті доступний список чатів із зазначенням дати, тематики та короткого опису. Це дозволяє користувачеві швидко знайти потрібну інформацію та відновити зміст попередніх діалогів [19].

2. Завантаження історії чатів (експорт)

Передбачено можливість експортувати історію чатів у текстовому форматі (.txt), що дозволяє зберігати важливі розмови локально або ділитися ними зі спеціалістом (психологом).

3. Зміна імені ШІ-асистента

Для підвищення персоналізації передбачено функцію зміни імені чат-бота. Користувач може обрати будь-яке ім'я для асистента — у подальшому система буде звертатися саме до цього імені.

Такий функціонал дозволяє користувачу максимально адаптувати систему під власні потреби та забезпечує зручність взаємодії [20].

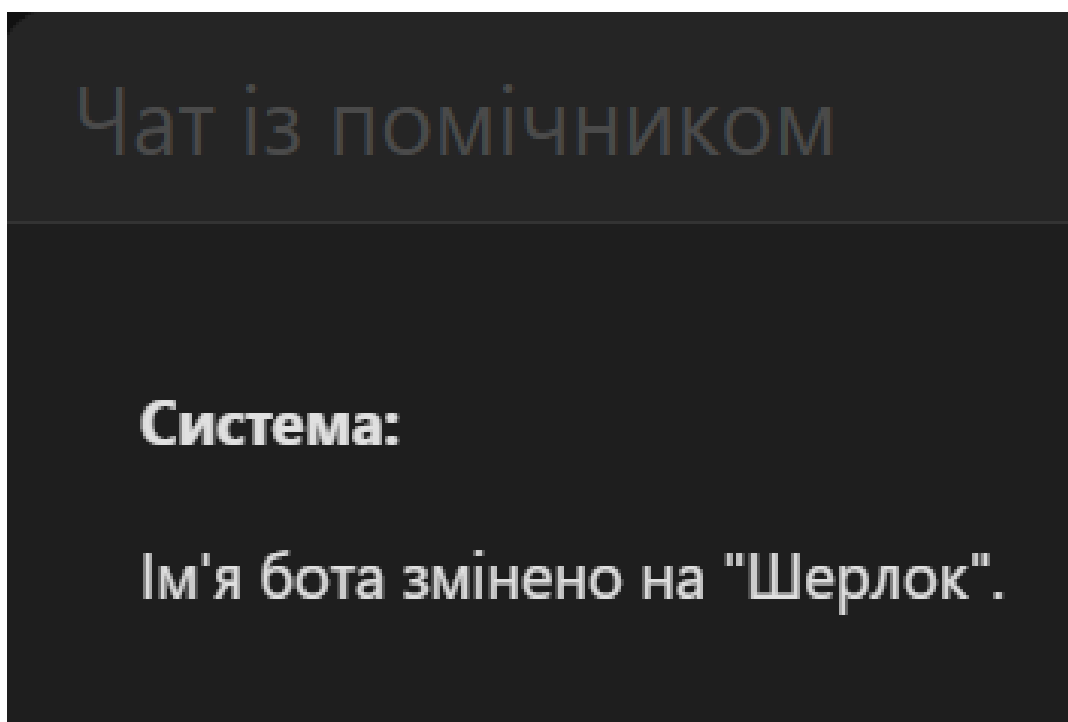


Рис. 4.9 Зміна імені ШІ-помічника

4.1.6 Робота з файлами

Для розширення можливостей взаємодії з ШІ-асистентом у системі реалізовано функцію завантаження текстових файлів.

Основні можливості:

1. Завантаження текстових файлів (.txt, .md, .json)

Користувач може завантажити файл зі своїми нотатками, есе, списками завдань чи іншими текстовими матеріалами. Підтримуються популярні формати для зручності.

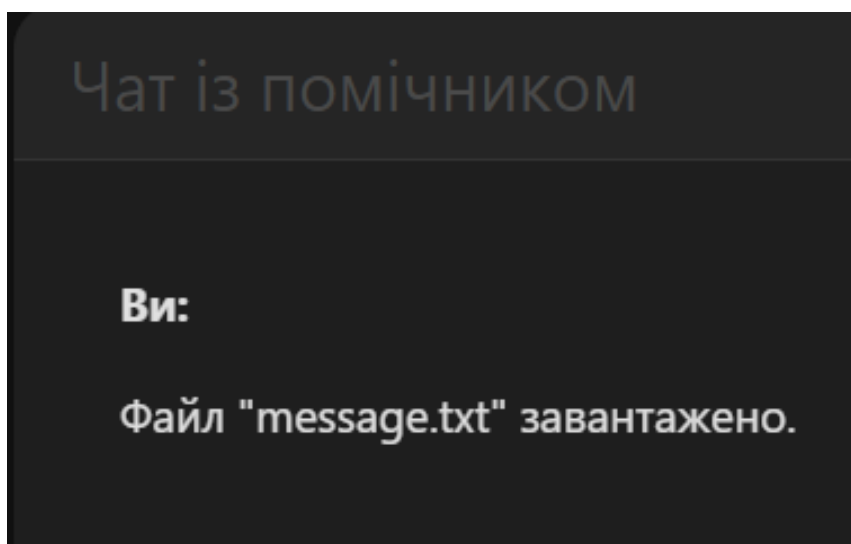


Рис. 4.10 Завантаження файла до системи

2. Відправлення вмісту на обробку ШІ

Після завантаження файл автоматично зчитується за допомогою JavaScript FileReader API, а його вміст передається ШІ-асистенту. ШІ може проаналізувати текст, надати рекомендації, резюмувати інформацію або виділити ключові моменти [21].

3. Збереження завантажених файлів у профілі користувача

Всі завантажені документи зберігаються у профілі користувача. Це дозволяє переглядати історію файлів, повторно надсилати їх на обробку або видаляти непотрібні.

Цей модуль суттєво розширює функціональні можливості застосунку та дозволяє гнучко працювати з власними текстовими матеріалами.

4.2 Голосовий ввід, збереження історії, аналіз файлів

Цей розділ об'єднує додаткові функціональні можливості, що підвищують зручність і ефективність взаємодії користувача із системою.

1. Голосовий ввід

Використовуючи Web Speech API, користувач може надиктовувати повідомлення замість набору тексту на клавіатурі. Це особливо корисно у стані

емоційного напруження або при бажанні швидко висловити думки. Система автоматично перетворює голос на текст, що спрощує комунікацію [22].

```
// Голосовий ввід
voiceInputBtn.addEventListener("click", () => {
  const SpeechRecognition = window.SpeechRecognition || window.webkitSpeechRecognition;
  if (!SpeechRecognition) {
    addMessage(botName, "⚠️ Ваш браузер не підтримує голосовий ввід");
    return;
  }

  const recognition = new SpeechRecognition();
  recognition.lang = "uk-UA"; // Фіксована українська мова
  recognition.interimResults = false;
  recognition.maxAlternatives = 1;

  recognition.onstart = () => {
    voiceInputBtn.classList.add("recording");
    addMessage("Система", "Слухаю...");
    messageInput.value = ""; // Очищаємо поле перед новим записом
  };

  recognition.onresult = (event) => {
    const transcript = event.results[0][0].transcript;
    messageInput.value = transcript; // Тільки заповнюємо поле вводу
    voiceInputBtn.classList.remove("recording");
    adjustTextareaHeight(); // Автоматично регулюємо висоту поля
  };
});
```

Рис. 4.11 Частина реалізації голосового вводу

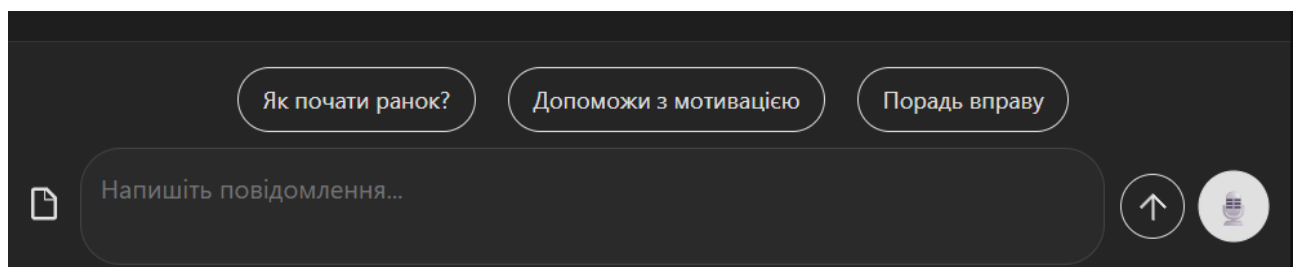


Рис. 4.12 Зображено голосовий ввід, завантаження файлів та швидкі промпти

2. Збереження історії чатів

Всі діалоги користувача із ШІ-асистентом зберігаються у базі даних разом із датами та контекстом розмов. Історія доступна для перегляду в особистому кабінеті. Це дозволяє відстежувати динаміку звернень,

аналізувати емоційний фон і повертатися до важливих бесід у разі потреби [19].

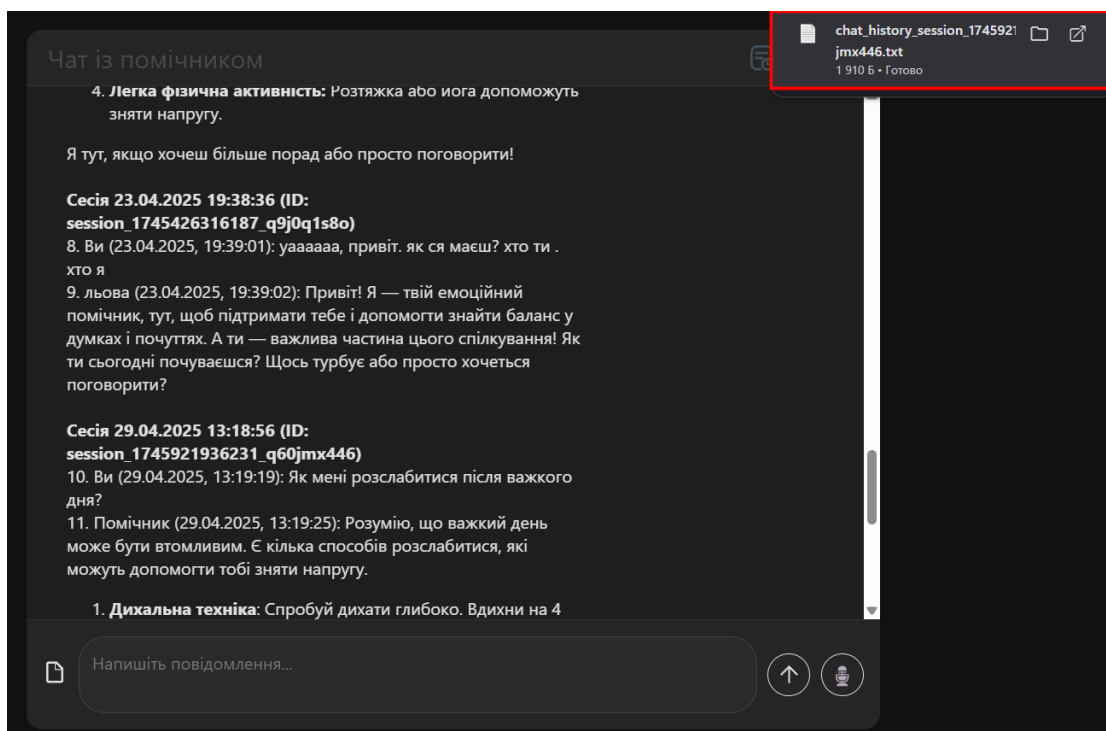


Рис. 4.13 Збереження історії чату

3. Аналіз файлів

Завдяки можливості завантаження текстових файлів користувач може надавати на обробку ШІ великі обсяги тексту, наприклад, особисті щоденники або документи. Це дозволяє отримувати детальний аналіз, резюме або рекомендації на основі наданої інформації [21].

Ці функції сприяють підвищенню інтерактивності системи, забезпечують гнучкість використання і відповідають сучасним вимогам до зручності користувацьких web-застосунків [17].

4.3 Щоденник настрою та його статистика

У розділі щоденника настрою реалізовано можливість:

- вибору емоційного стану із запропонованих варіантів (радість, сум, тривога тощо);
- додавання нотатки до кожного запису;
- перегляду історії записів у вигляді списку;
- побудови графіка зміни настрою (Chart.js), що дозволяє візуально оцінити динаміку емоцій.

4.4 Поради, інтеграція для спеціалістів

У розділі “Поради та ресурси” реалізовано:

- категоризовану підбірку матеріалів (статті, вправи, відео);
- можливість завантаження ресурсів.
- Для фахівців (психологів):
- доступ до перегляду історії взаємодії клієнтів із системою;
- завантаження звітів про динаміку настрою користувача.

4.5 Тестування

Для перевірки працездатності застосунку було проведено кілька типів тестування.

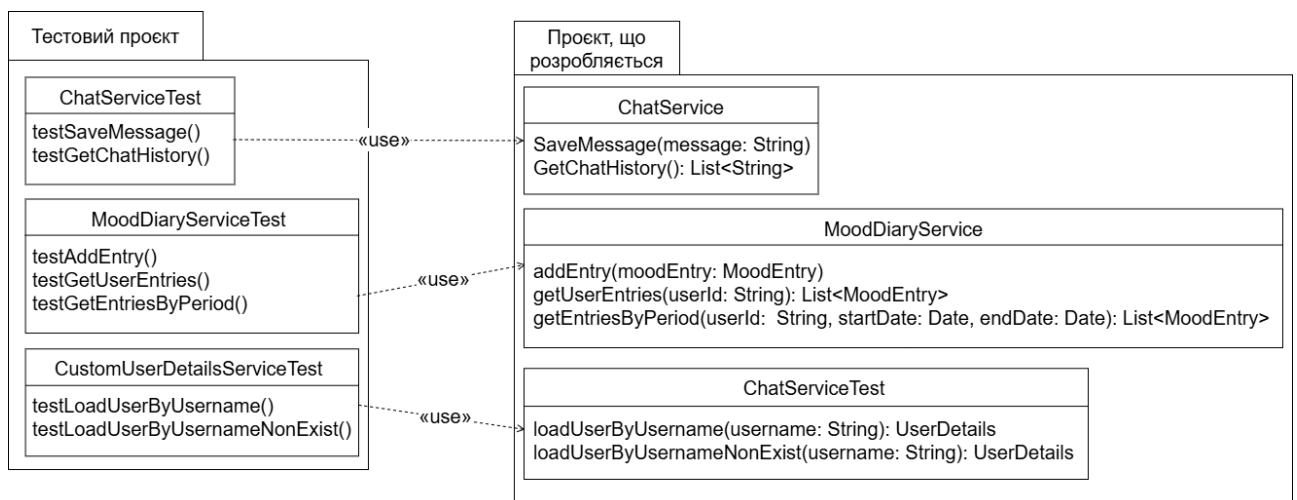


Рис. 4.14 Діаграма тестування класів

Діаграма тестування класів відображає організацію тестування системи, поділену на два основні пакети: «Тестовий проект» і «Проект, що розробляється». У пакеті «Тестовий проект» зосереджені тестові класи, які перевіряють функціонал основних компонентів. Клас `ChatServiceTest` містить методи `testSaveMessage` і `testGetChatHistory`, що тестують збереження повідомлень і отримання історії чату. `MoodDiaryServiceTest` включає методи `testAddEntry`, `testGetUserEntries` і `testGetEntriesByPeriod`, які перевіряють додавання записів у щоденник настрою, отримання записів користувача та вибірку за періодом. `CustomUserDetailsServiceTest` із методами `testLoadUserByUsername` і `testLoadUserByUsernameNonExist` тестує автентифікацію користувачів у двох сценаріях: коли користувач існує і коли його немає.

У пакеті «Проект, що розробляється» розташовані основні класи. `ChatService` має методи `saveMessage` для збереження повідомлень і `getChatHistory` для отримання історії у вигляді списку рядків. `MoodDiaryService` включає `addEntry` для додавання записів, `getUserEntries` для отримання списку записів користувача та `getEntriesByPeriod` для вибірки записів за періодом.

`CustomUserDetailsService` містить методи `loadUserByUsername` і `loadUserByUsernameNonExist`, які повертають деталі користувача за його іменем.

Залежності між тестовими і основними класами позначені стрілками «use», що показують, які методи тестуються, забезпечуючи чітке покриття функціоналу системи.

4.5.1 Unit-тестування

Проводилося за допомогою JUnit для окремих компонентів бекенду (сервіси, репозиторії) [20].

Усі модулі пройшли тестування успішно.

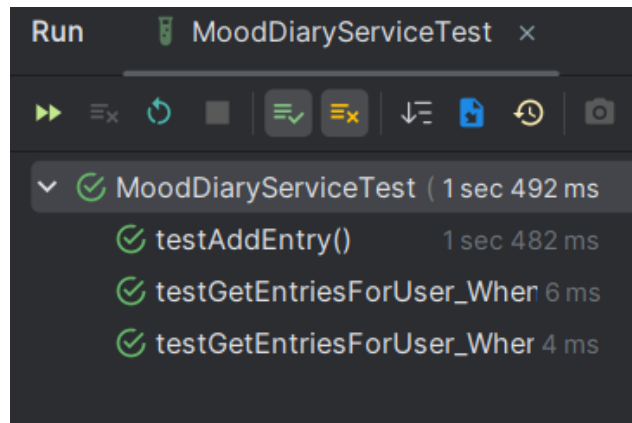


Рис. 4.15 Тестування методом JUnit

Проведення тестування на рівні окремих одиниць (unit-тестів) відіграє ключову роль у забезпеченні надійності програмного забезпечення (сервіси, репозиторії). У цьому розділі розглядається підхід до реалізації unit-тестів, спрямованих на перевірку функціональності ключових компонентів системи. Основна увага приділяється класу `MoodDiaryServiceTest`, який демонструє ефективність модульного підходу до перевірки логіки.

Процес тестування охоплює кілька етапів. Метод `testAddEntry()` перевіряє коректність додавання нового запису до щоденника настрою, виконуючи його за 1 секунду 482 мілісекунди, що свідчить про високу швидкодію. Методи `testGetEntriesForUser_When` (з варіантами "Exists" та "Where") аналізують вибірку записів користувача в різних сценаріях: коли дані присутні (6 мілісекунд) і коли їх немає (4 мілісекунди), забезпечуючи повне покриття логіки. Загальний час виконання тесту `MoodDiaryServiceTest` становить 1 секунду 492 мілісекунди, що підкреслює оптимальність використаної реалізації.

Цей підхід дозволяє не лише виявити потенційні помилки на ранніх стадіях розробки, але й забезпечити стабільність системи при подальших оновленнях. Використання фреймворку JUnit у поєднанні з детальним аналізом результатів тестів сприяє підвищенню якості коду та полегшує підтримку проекту. Такий методологічний акцент на модульному тестуванні стає основою для створення гнучкої та надійної архітектури, адаптованої до потреб користувачів.

4.5.2 Інтеграційне тестування

Перевірка взаємодії між бекендом, базою даних і зовнішнім API ChatGPT. Використано Postman для тестування REST API [14].

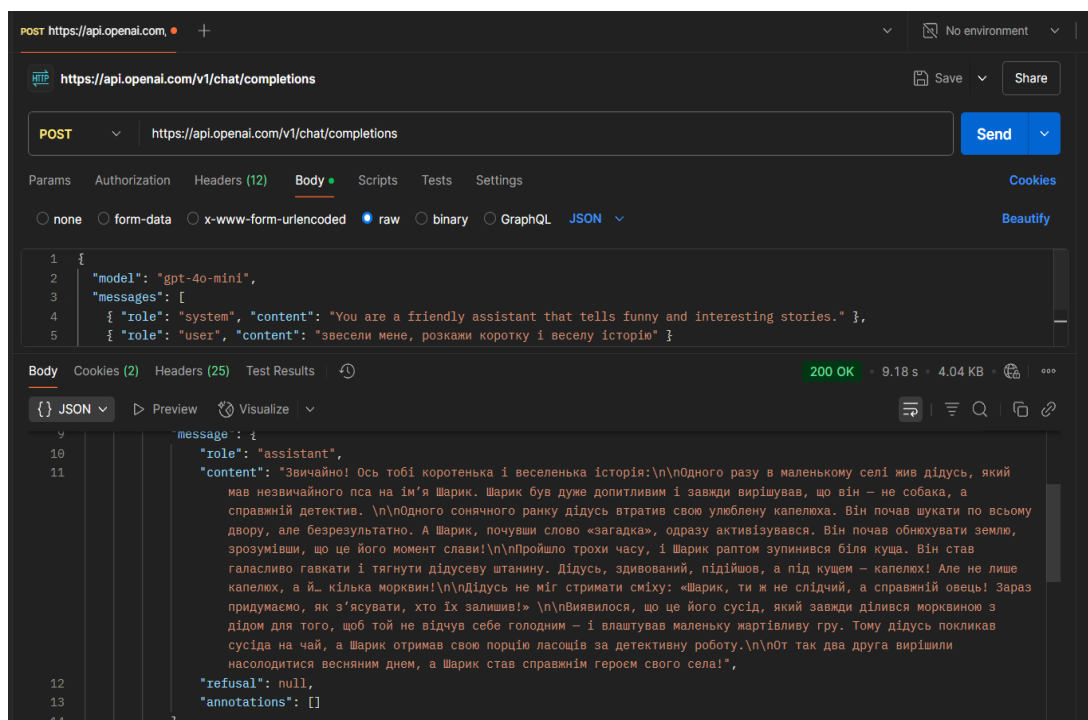


Рис. 4.16 Тестування API за допомогою Postman

Перевірка взаємодії між внутрішніми компонентами системи та зовнішнім API ChatGPT є ключовим етапом забезпечення стабільної роботи додатку. Дане тестування реалізовано за допомогою інструменту Postman, який дозволяє імітувати запити REST API [14]. Процес включає відправлення HTTP-запитів до ендпоінту `https://api.openai.com/v1/chat/completions`, де передаються параметри, такі як модель (`gpt-4o-mini`) та повідомлення з контекстом (наприклад, "You are a friendly assistant that tells funny and interesting stories" для системи та "розкажи коротку веселу історію" від користувача).

У відповідь API повертає деталізовану текстову відповідь, яку система обробляє та адаптує для подальшого використання. Тестування підтвердило успішне виконання запиту з кодом 200 OK за 9.18 секунд, що свідчить про коректну інтеграцію та прийнятну швидкість реагування. Цей підхід дозволяє

не лише перевірити з'єднання з зовнішнім сервісом, але й оцінити, як система справляється з обробкою отриманих даних. Використання Postman у цьому контексті забезпечує гнучкість для тестування різних сценаріїв і сприяє підвищенню надійності інтеграційних взаємодій.

4.5.3 Функціональне тестування

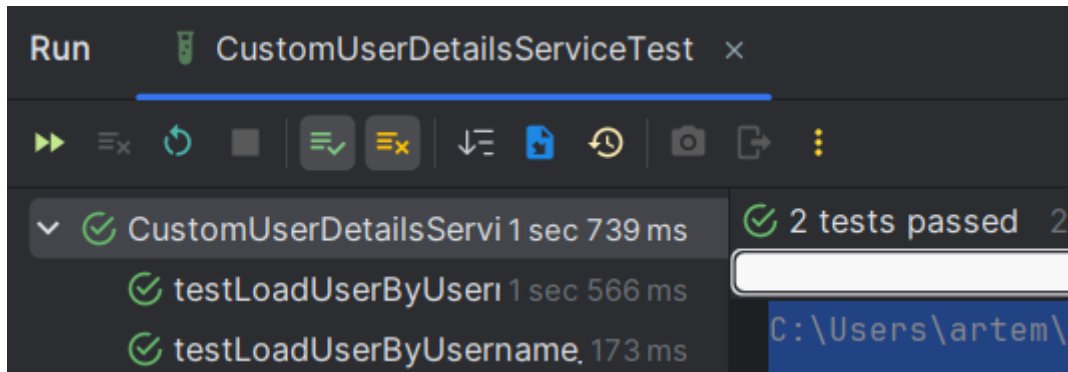


Рис. 4.17 Тестування додавання користувачів

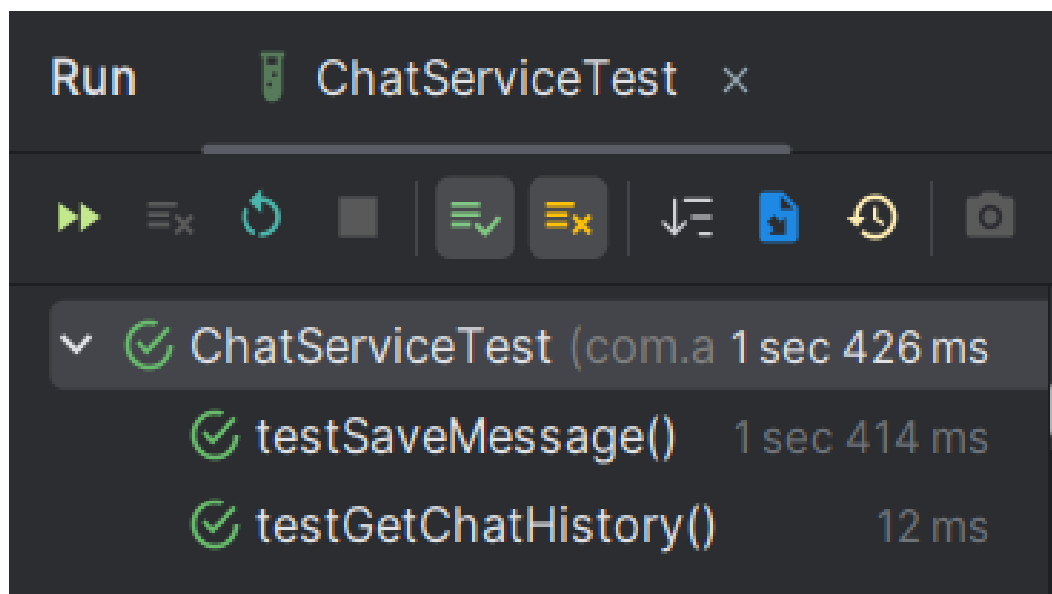


Рис. 4.18 Тестування отримання історії чату та зберігання повідомлень

Функціональне тестування проведено для забезпечення коректної роботи ключових сценаріїв взаємодії користувача з системою. Цей процес охоплює кілька основних аспектів, які відображають повноту функціоналу додатку:

1. Реєстрація та авторизація: Перевірено можливість створення нового облікового запису та успішного входу в систему, що є базовою умовою для доступу до інших функцій. Тестування автентифікації підтвердило стабільність цього процесу, включаючи обробку як існуючих, так і неіснуючих користувачів.
2. Взаємодія з ШІ-асистентом: Оцінено коректність роботи текстового чату з ШІ, зокрема збереження повідомлень та доступ до їхньої історії. Додатково розглянуто підготовку до інтеграції голосових команд, хоча ця функція ще потребує подальшого тестування.
3. Управління щоденником настрою: Перевірено додавання та перегляд записів про емоційний стан, що забезпечує точність аналізу настрою користувача. Цей етап включає тестування вибірки даних за періодом для зручного відстеження змін.
4. Аналіз даних: Оцінено коректність відображення статистики настрою та історії чату, що дозволяє користувачу отримувати огляд свого прогресу та приймати обґрунтовані рішення.
5. Візуалізація та статистика: Перевірено функціонал графічного відображення емоційних тенденцій, який надає користувачу інтуїтивно зрозумілу картину змін настрою.
6. Налаштування профілю: Тестовано можливість редагування особистих даних, зміни пароля та налаштування сповіщень, що забезпечує індивідуальний підхід до використання системи.

Цей підхід до функціонального тестування дозволив виявити потенційні слабкі місця, наприклад, потребу в додатковій перевірці голосових функцій, та підтвердити стабільність основних сценаріїв. Результатом стало підвищення довіри до системи та підготовка її до подальшого розширення, адаптованого до потреб користувачів.

ВИСНОВКИ

У межах кваліфікаційної роботи було розроблено web-застосунок для аналізу та підтримки емоційного стану користувачів. На основі вивчення аналогічних рішень, формування вимог і вибору оптимального технологічного стеку створено функціональну систему з інтелектуальним чат-асистентом, щоденником настрою та аналітикою емоцій. У процесі виконання поставлених завдань досягнуто наступних результатів:

1. Проаналізовано сучасні цифрові рішення у сфері емоційної підтримки, зокрема Woebot, Youper, Wysa та Mindspace, що дозволило виявити найбільш ефективні підходи та сформувати конкурентоспроможну ідею. Це дало змогу адаптувати найкращі практики і уникнути недоліків, виявлених у існуючих системах.
2. Визначено функціональні та нефункціональні вимоги до системи, які стали основою для архітектурного, інтерфейсного і технологічного проєктування. Завдяки цьому проєкт відповідає реальним потребам користувача й забезпечує зручність та ефективність взаємодії.
3. Обґрунтовано вибір технологій, серед яких Java Spring Boot, MySQL, HTML, CSS, JS, API ChatGPT та бібліотека Chart.js відповідно до вимог системи та очікувань цільової аудиторії. Такий підхід забезпечує швидке впровадження нових функціональних модулів і стабільну роботу під високим навантаженням.
4. Спроектовано архітектуру web-застосунку з інтеграцією API ChatGPT для забезпечення інтелектуальної взаємодії.

У результаті чат працює у реальному часі, підтримує контекст бесіди та підвищує залученість користувачів.

5. Реалізовано ключовий функціонал системи, зокрема: чат з ІІІ-асистентом, щоденник настрою, аналітика, персональний

кабінет, робота з файлами. Це дозволяє користувачеві відстежувати власні емоції, отримувати рекомендації та використовувати систему як інструмент самопідтримки.

6. Забезпечено персоналізацію і безпеку інтерфейсу за допомогою Spring Security та налаштувань профілю, що підвищує зручність і захищеність даних. Це сприяє формуванню довіри користувачів і гарантує конфіденційність особистої інформації.
7. Проведено комплексне тестування розробленої системи, включно з модульним, інтеграційним і функціональним рівнями перевірки, що підтвердило її відповідність функціональним та нефункціональним вимогам. Це доводить технічну готовність застосунку до впровадження і використання.

Обґрунтування доцільності впровадження

Розроблений застосунок є інноваційним рішенням у сфері цифрової підтримки емоційного здоров'я. Його унікальність полягає у поєднанні таких ключових можливостей:

- персоналізовану взаємодію з ШІ-асистентом через API ChatGPT;
- введення та аналіз емоційних станів через інтегрований щоденник настрою;
- візуалізацію динаміки настрою за допомогою графіків;
- доступ до матеріалів для самодопомоги: статей, вправ, порад;
- можливість використання системи фахівцями (психологами) для підтримки пацієнтів.

Сукупність цих можливостей робить застосунок гнучким інструментом, який сприяє покращенню ментального здоров'я, забезпечує зручну самопідтримку та підвищує ефективність психологічної допомоги. Це розширює доступ до допомоги та знижує бар'єри для її отримання, особливо серед молоді.

Подальші напрями розвитку

На основі аналізу потреб користувачів і технологічного потенціалу системи, сформовано стратегічні напрями її вдосконалення:

- розробка мобільного застосунку (Android/iOS) для підвищення доступності;
- розширення бібліотеки ресурсів, включно з інтерактивними вправами та відеоматеріалами;
- створення інтелектуального модуля рекомендацій, що буде аналізувати зміни в настрої та пропонувати індивідуальні поради;
- впровадження аналітичних звітів для фахівців із візуалізацією емоційної динаміки клієнтів.

Реалізація цих напрямів дозволить не лише підвищити якість взаємодії з користувачем, а й вивести застосунок на рівень професійного інструменту для психологів, освітян та HR-фахівців.

Робота пройшла апробацію. За її результатами було опубліковано наступні тези доповідей:

1. Бражник А.М., Гаманюк І. М. Технології захисту інформації у веб-системі моніторингу емоційного стану користувача: Матеріали XII Всеукраїнської науково-практичної конференції молодих учених «Інформаційні технології - 2025», 15.05.2025, м. Київ. Київський столичний університет імені Бориса Грінченка. Збірник тез. К.: КУБГ, 2025. С. 265-266.

2. Бражник А.М., Гаманюк І. М. Розробка архітектури ШІ-помічника на основі API ChatGPT у веб-застосунку для підтримки емоційного благополуччя користувачів: Матеріали VI Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT». 15.04.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (подано до друку).

3. Бражник А.М., Гаманюк І. М. Система прогнозування емоційного стану користувача на основі обробки текстових даних: Матеріали VI Всеукраїнська науково-технічна конференція «Застосування програмного

забезпечення в інформаційно-комунікаційних технологіях». 24.04.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (подано до друку).

4. Бражник А.М., Гаманюк І. М. Методи та засоби штучного інтелекту у web-застосунку для підтримки емоційного благополуччя користувачів: Матеріали V Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті». 15.05.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (подано до друку).

5. Бражник А.М., Гаманюк І. М. Використання діаграми компонентів для проектування архітектури застосунку з підтримки емоційного благополуччя користувачів: Матеріали Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». 24.04.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С. 30-33.

ПЕРЕЛІК ПОСИЛАНЬ

1. Woebot Health. Woebot Health. URL: <https://woebothealth.com/> (дата звернення: 17.05.2025).
2. Youper: Artificial Intelligence for Mental Health. Youper. URL: <https://www.youper.ai/> (дата звернення: 17.05.2025).
3. Wysa – Everyday Mental Health. Wysa. URL: <https://www.wysa.io/> (дата звернення: 17.05.2025).
4. Mindspa – #1 Self-Therapy App. Mindspa. URL: <https://mindspa.me/> (дата звернення: 17.05.2025).
5. Оцінка психоемоційного стану студентів медичного закладу в умовах військового конфлікту. Мірошніченко М. І., Андрієнко Ю. В. *Reports of Vinnytsia National Medical University*, 2023, № 27(3), с. 654–660. DOI: [https://doi.org/10.31393/reports-vnmedical-2023-27\(3\)-28](https://doi.org/10.31393/reports-vnmedical-2023-27(3)-28)
6. Occupational health. World Health Organization (WHO). URL: <https://www.who.int/health-topics/occupational-health> (дата звернення: 17.05.2025).
7. Psychological Impact of War on Healthcare Workers. Ukrainian Institute of Mental Health, 2023. Київ: UIMH Press.
8. Mental Health Support Needs in Ukraine. Ministry of Health of Ukraine. URL: <https://moz.gov.ua/mental-health> (дата звернення: 17.05.2025).
9. Spring Boot. Spring Boot. URL: <https://spring.io/projects/spring-boot> (дата звернення: 17.05.2025).
10. OpenAI API Documentation. OpenAI. URL: <https://platform.openai.com/docs/> (дата звернення: 17.05.2025).
11. Chart.js Documentation. Chart.js. URL: <https://www.chartjs.org/docs/latest/> (дата звернення: 17.05.2025).
12. Spring Security Documentation. Spring Security. URL: <https://spring.io/projects/spring-security> (дата звернення: 17.05.2025).
13. RESTful Web Services. Richardson L., Ruby S. *O'Reilly Media*, 2022.

14. Postman. Postman. URL: <https://www.postman.com/> (дата звернення: 17.05.2025).
15. Цифрові інструменти підтримки психічного здоров'я: можливості та обмеження. Вільчак О. Б. *Соціальна психологія*, 2021, № 48(2), с. 15–23.
16. Інформаційні технології в психологічному консультуванні. Кузьменко Н. І. *Наукові записки кафедри психології КНУ*, 2020, № 3(1), с. 43–49.
17. Використання мобільних додатків для підтримки емоційного стану студентів. Чекурова А. В. *Освіта і розвиток*, 2022, № 4(18), с. 79–85.
18. Персоналізація взаємодії користувачів у web-застосунках для підтримки ментального здоров'я. Савчук Н. І., Бондар Т. В. *Інформаційні технології і засоби навчання*, 2021, № 82(6), с. 65–74.
19. Обробка текстових документів в інформаційних системах психологічної підтримки. Дяченко А. О. *Інформаційні технології*, 2023, № 27(1), с. 33–39.
20. JUnit 5 Documentation. JUnit. URL: <https://junit.org/junit5/> (дата звернення: 17.05.2025).
21. Веб-розробка: технології клієнтської частини (HTML, CSS, JavaScript). Бойко С. В., Паламарчук М. О. Київ: КНЕУ, 2020.
22. Java Spring Boot: засоби розробки серверних веб-додатків. Ткаченко О. В. *Вісник КНУТД*, 2021, № 143(2), с. 89–95.
23. Get Started with the PageSpeed Insights API. Google for Developers. URL: <https://developers.google.com/speed/docs/insights/v5/get-started> (дата звернення: 20.05.2025).
24. Реляційні бази даних у сучасних інформаційних системах. Кушніренко О. В. *Вісник ХНУРЕ*, 2020, № 1(68), с. 41–47.
25. REST API як основа взаємодії програмних компонентів у веб-системах. Дьяков І. О. *Східноєвропейський журнал передових технологій*, 2022, № 2(1), с. 58–64.

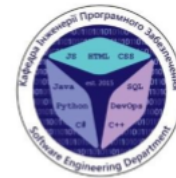
ДОДАТОК А. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ
ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



РОЗРОБКА WEB-ЗАСТОСУНКУ ДЛЯ АНАЛІЗУ ТА ПІДТРИМКИ ЕМОЦІЙНОГО СТАНУ

Виконав студент 4 курсу

групи ПД-42

Бражник Артем Михайлович

Керівник роботи

Старший викладач кафедри ІПЗ Гаманюк Ігор Михайлович

Київ – 2025

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** підвищення рівня емоційного благополуччя користувачів за рахунок функціоналу web-застосунку для аналізу та підтримки емоційного стану.
- **Об'єкт дослідження** процес взаємодії користувача з цифровими інструментами підтримки емоційного стану.
- **Предмет дослідження** web-застосунок для аналізу емоційного стану з інтеграцією штучного інтелекту.

ЗАДАЧІ ДИПЛОМНОЇ РОБОТИ

- 1. Проаналізувати сучасні рішення у сфері цифрової емоційної підтримки.
- 2. Сформувати функціональні та нефункціональні вимоги до системи.
- 3. Дослідити сучасні технології та обґрунтувати їх вибір відповідно до визначених вимог.
- 4. Спроектувати архітектуру веб-застосунку з інтеграцією API ChatGPT.
- 5. Реалізувати функціонал: щоденник настрою, ШІ-асистент, аналітика даних.
- 6. Забезпечити персоналізацію, безпеку та адаптивність інтерфейсу.
- 7. Перевірити реалізовану систему на відповідність визначеним функціональним та нефункціональним вимогам.

3

АНАЛІЗ АНАЛОГІВ

Застосунок	ШІ-асистент	Щоденник	Персон алізація	Інформаційні матеріали	Аналіз даних	Фахівці	Локалізація
Woebot	+	-	-	+	-	-	-
Youper	+	-	+	+	-	-	-
Wysa	+	+	-	-	-	+	-
Mindspa	-	+	-	+	-	-	-
Emotionalee	+	+	+	+	+	+	+

4

ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Функціональні вимоги:

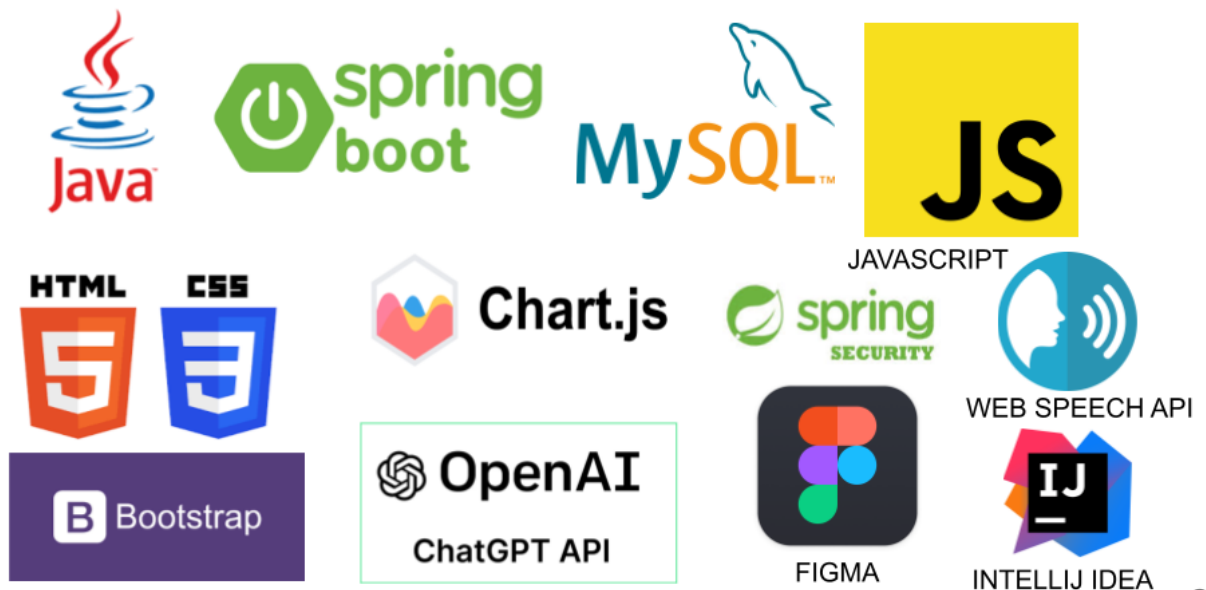
1. ШІ-асистент — діалогова взаємодія, отримання консультацій, рекомендацій та аналіз настрою.
2. Щоденник настрою — ведення записів, відстеження та візуалізація даних.
3. Швидкі промпти — миттєві запити типу "розвесели мене".
4. Виконання вправ — доступ до теоретичних матеріалів та практичних завдань.
5. Голосовий ввід — зручна взаємодія голосом при консультації.
6. Завантаження файлів — аналіз текстових документів системою.
7. Персоналізація — налаштування стилю відповіді та імені асистента.
8. Інтеграція з фахівцями — передача звітів психологу, спільний аналіз.

Нефункціональні вимоги:

1. Безпека — захист даних користувача.
2. Адаптивність — підтримка десктопів, планшетів, смартфонів.
3. Локалізація — повна підтримка української мови.
4. Основні функції доступні за два кліки.
5. Масштабованість системи.

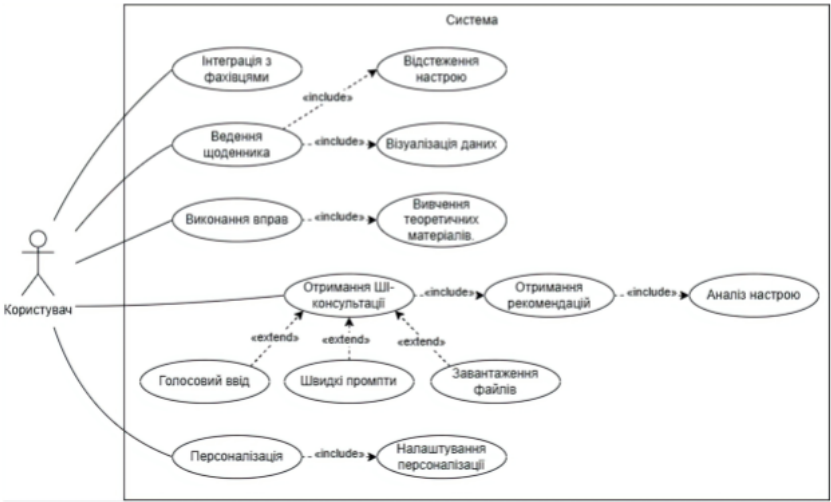
5

ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ



6

МОДЕЛЮВАННЯ ФУНКЦІОНУВАННЯ СИСТЕМИ



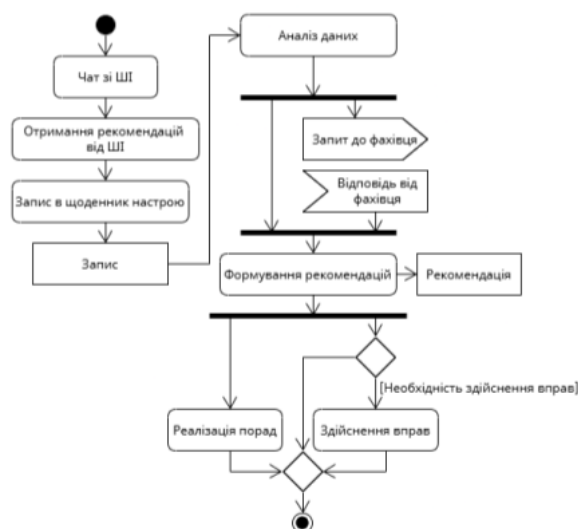
Діаграма прецедентів

БІЗНЕС МОДЕЛЬ



Діаграма предметної галузі

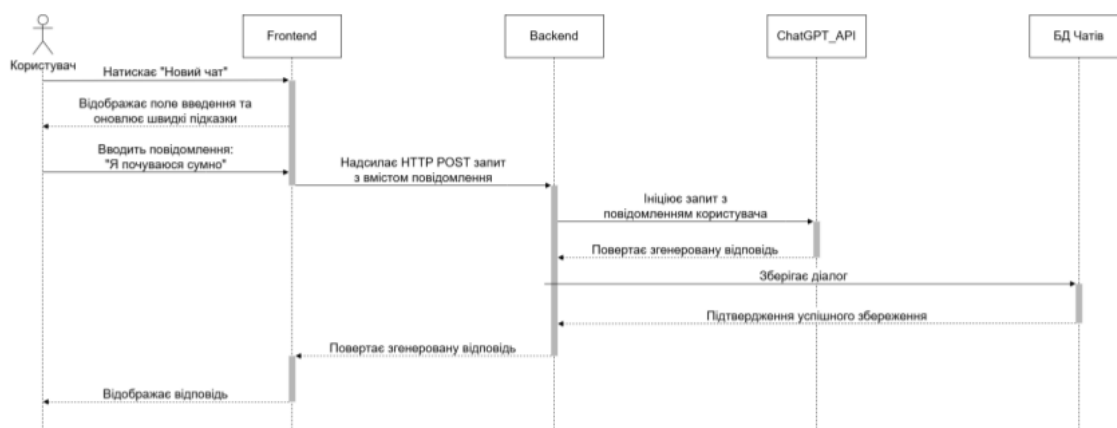
МОДЕЛЮВАННЯ ДІЯЛЬНОСТІ КОРИСТУВАЧА



Діаграма діяльності

9

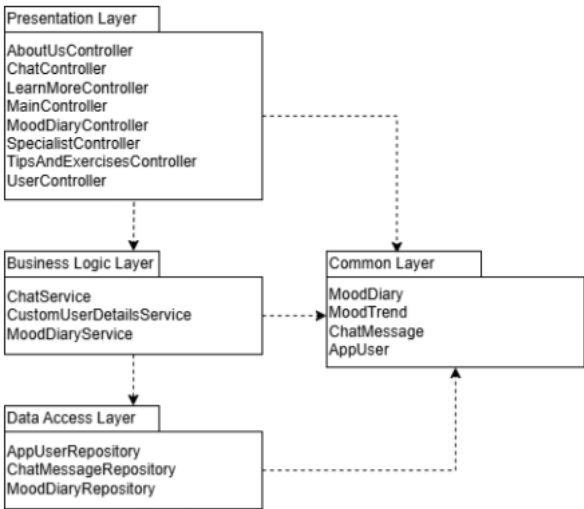
АЛГОРИТМ ДІАЛОГУ З ШІ-ПОМІЧНИКОМ



Діаграма послідовності

10

ЛОГІЧНА АРХІТЕКТУРА



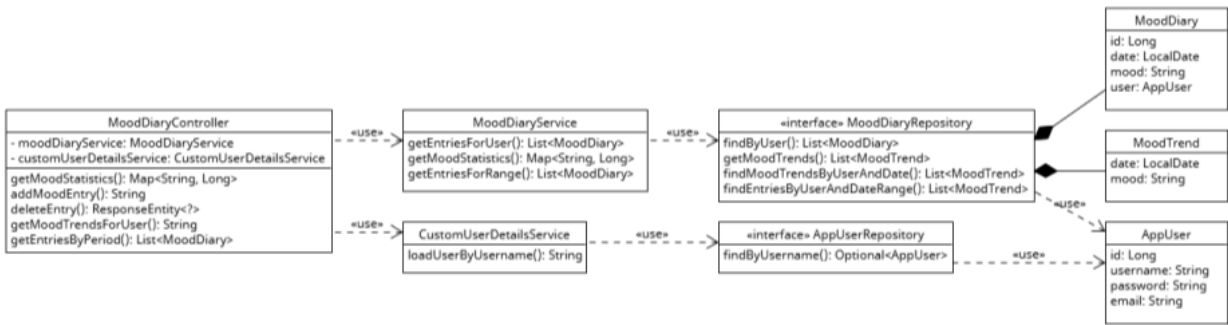
Діаграма пакетів

ПІДСИСТЕМИ ЗАСТОСУНКУ



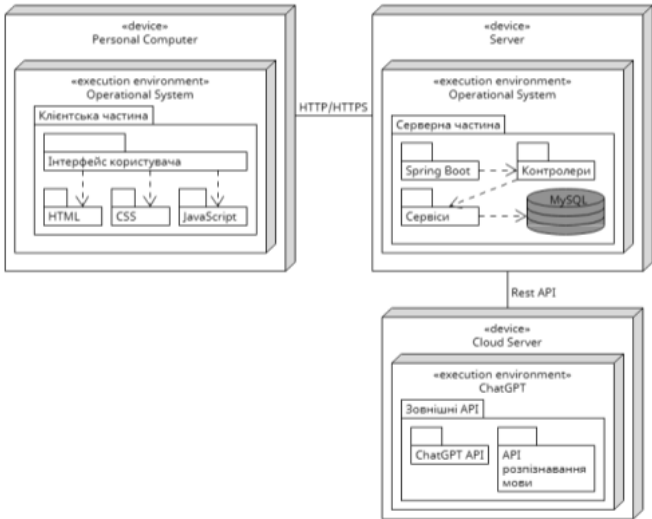
Діаграма компонентів

МОДЕЛЮВАННЯ СТРУКТУРИ СИСТЕМИ ТА ВІДНОСИН
МІЖ КЛАСАМИ



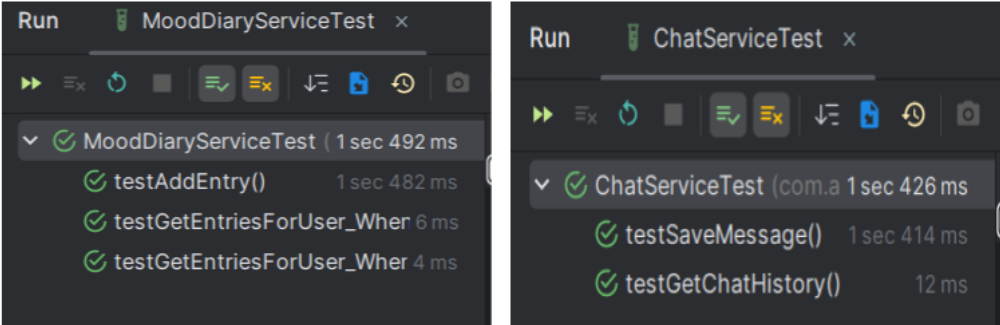
Діаграма класів

РОЗГОРТАННЯ ЗАСТОСУНКУ



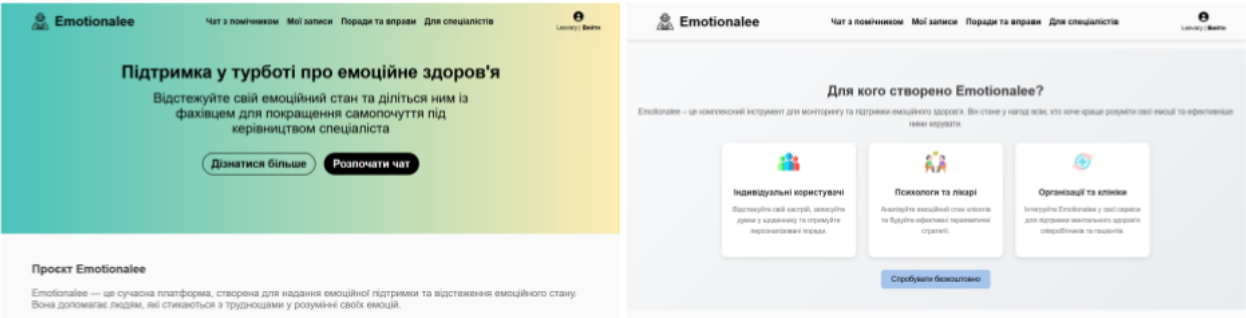
Діаграма розгортання

ТЕСТУВАННЯ ЗАСТОСУНКУ



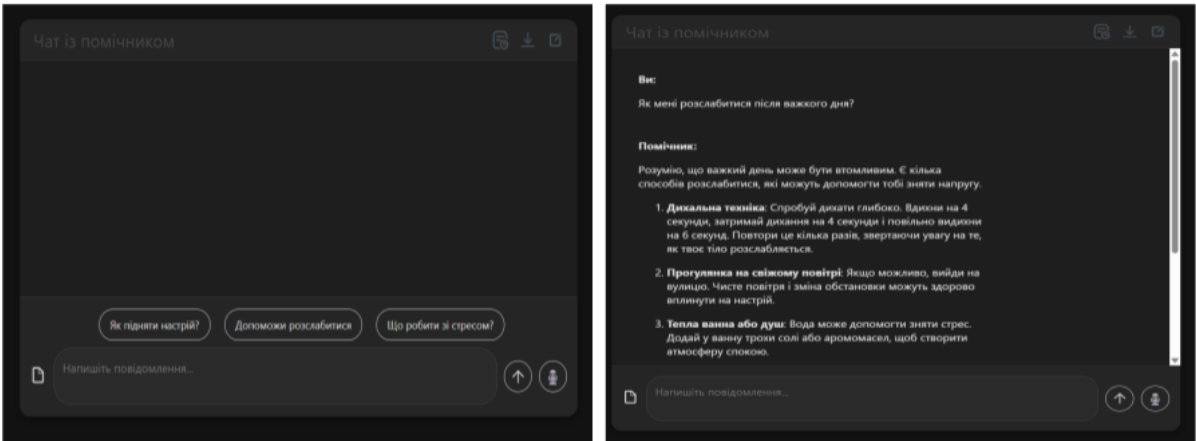
Результати модульного тестування застосунку через JUnit

ЕКРАННІ ФОРМИ



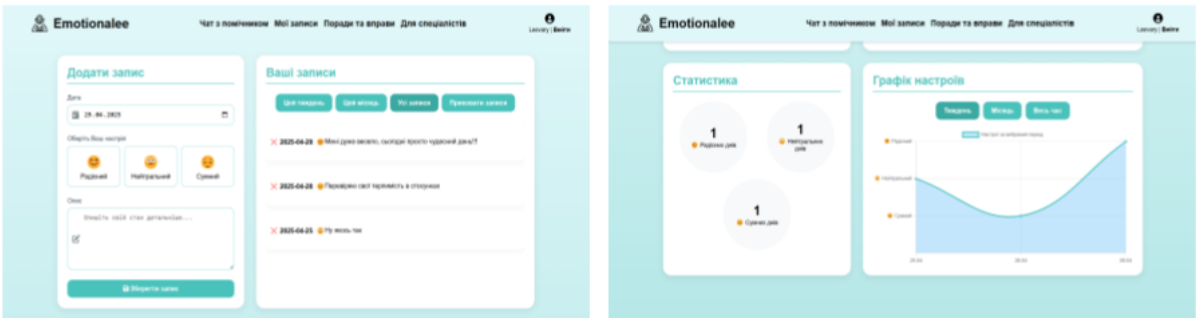
Головна сторінка системи

ЕКРАННІ ФОРМИ



Спілкування із ШІ-помічником

ЕКРАННІ ФОРМИ



Ведення щоденника настрою, перегляд аналітики

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Тези доповідей:

1. Бражник А.М., Гаманюк І. М. Технології захисту інформації у веб-системі моніторингу емоційного стану користувача: Матеріали XII Всеукраїнської науково-практичної конференції молодих учених «Інформаційні технології - 2025», 15.05.2025, м. Київ. Київський столичний університет імені Бориса Грінченка. Збірник тез. К.: КУБГ, 2025. С. 265-266.
2. Бражник А.М., Гаманюк І. М. Розробка архітектури ШІ-помічника на основі API ChatGPT у веб-застосунку для підтримки емоційного благополуччя користувачів: Матеріали VI Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT». 15.04.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (подано до друку).
3. Бражник А.М., Гаманюк І. М. Система прогнозування емоційного стану користувача на основі обробки текстових даних: Матеріали VI Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». 24.04.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (подано до друку).
4. Бражник А.М., Гаманюк І. М. Методи та засоби штучного інтелекту у веб-застосунку для підтримки емоційного благополуччя користувачів: Матеріали V Всеукраїнської науково-практичної конференції «Сучасні інтелектуальні інформаційні технології в науці та освіті». 15.05.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. (подано до друку).
5. Бражник А.М., Гаманюк І. М. Використання діаграми компонентів для проектування архітектури застосунку з підтримки емоційного благополуччя користувачів: Матеріали Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в ІКТ». 24.04.2025, м. Київ. Державний університет інформаційно-комунікаційних технологій. Збірник тез. К.: ДУІКТ, 2025. С. 30-33.

19

ВИСНОВКИ

1. Проаналізовано сучасні цифрові рішення у сфері емоційної підтримки, що дозволило визначити найбільш ефективні підходи для реалізації застосунку.
2. Сформовано функціональні та нефункціональні вимоги до системи, які стали основою для її архітектурного та технологічного проектування.
3. Досліджено сучасні технології, обґрунтовано їх вибір відповідно до вимог системи та очікувань користувачів.
4. Спроектовано архітектуру web-застосунку з інтеграцією API ChatGPT, що забезпечує інтелектуальну підтримку користувача.
5. Реалізовано основний функціонал системи: щоденник настрою, ШІ-асистент та аналітика даних.
6. Забезпечено персоналізацію, адаптивність інтерфейсу та базовий рівень безпеки відповідно до сформульованих вимог.
7. Проведено перевірку реалізованої системи, яка підтвердила її відповідність заданим функціональним і нефункціональним вимогам.

20

ДОДАТОК Б. ЛІСТИНГИ ПРОГРАМНИХ МОДУЛІВ СЕРВЕРНОЇ ЧАСТИНИ

```
// Генерація унікального sessionId
const generateSessionId = () => {
  return 'session_' + Date.now() + '_' + Math.random().toString(36).substr(2, 9);
};

// Генерація назви сесії
const generateSessionName = () => {
  const now = new Date();
  return `Сесія ${now.toLocaleDateString()} ${now.toLocaleTimeString()}`;
};

// Ініціалізація нової сесії
const initNewSession = () => {
  sessionId = generateSessionId();
  sessionName = generateSessionName();
  localStorage.setItem(`currentSessionId_${userId}`, sessionId);
  localStorage.setItem(`currentSessionName_${userId}`, sessionName);
  chatBox.innerHTML = "";
  chatBox.classList.remove("expanded");
  showQuickPrompts();
};

// Отримання ID користувача з бекенду
const getUserId = async () => {
  try {
    const response = await fetch("/api/user/id", {
      method: "GET",
      headers: {
        "Content-Type": "application/json",
      },
    });
    if (!response.ok) throw new Error("Не вдалося отримати ID користувача");
    const data = await response.json();
    userId = data.userId;
    if (!userId) throw new Error("ID користувача не визначено");
    console.log("Отримано userId:", userId);

    // Завантажуємо або ініціалізуємо сесію
    sessionId = localStorage.getItem(`currentSessionId_${userId}`);
    sessionName = localStorage.getItem(`currentSessionName_${userId}`);
    if (!sessionId || !sessionName) {
      initNewSession();
    }
  } catch (error) {
    console.error("Помилка отримання ID користувача:", error);
    userId = "anonymous";
    initNewSession();
  }
};

// Завантаження історії чату для поточної сесії
const loadChatHistory = async () => {
  try {
    const response = await fetch(`/chat/history?userId=${userId}&sessionId=${sessionId}`);
    if (!response.ok) throw new Error("Не вдалося завантажити історію чату");
    const history = await response.json();
    chatBox.innerHTML = "";
    if (history.length === 0) {
      chatTitle.classList.add("visible");
    } else {
      chatBox.classList.add("expanded");
      history.forEach(({ author, content }) => {
        const messageElement = document.createElement("div");

```

```

        messageElement.classList.add("message", author.toLowerCase());
        const formattedContent = marked.parse(content);
        messageElement.innerHTML = `<strong>${author}</strong> ${formattedContent}`;
        chatBox.appendChild(messageElement);
    });
}
chatBox.scrollTop = chatBox.scrollHeight;
} catch (error) {
    console.error("Помилка завантаження історії:", error);
    addMessage("Система", "Не вдалося завантажити історію чату.");
}
};

// Збереження повідомлення на бекенді
const saveMessage = async (author, content) => {
    try {
        const response = await fetch("/chat/save", {
            method: "POST",
            headers: {
                "Content-Type": "application/json",
            },
            body: JSON.stringify({
                userId: userId,
                sessionId: sessionId,
                sessionName: sessionName,
                author: author,
                content: content,
            }),
        });
    } catch (error) {
        console.error("Помилка збереження повідомлення:", error);
        addMessage("Система", "Не вдалося зберегти повідомлення: ${error.message}");
    }
};

// Читання текстового файлу на клієнті
const readFileContent = (file) => {
    return new Promise((resolve, reject) => {
        if (!file) {
            reject(new Error("Файл не вибрано"));
            return;
        }
        if (!file.type.includes("text")) {
            reject(new Error("Файл має бути текстовим (.txt)"));
            return;
        }
        const reader = new FileReader();
        reader.onload = (event) => resolve(event.target.result);
        reader.onerror = (error) => reject(error);
        reader.readAsText(file);
    });
};

// Відправка повідомлення
const sendMessage = async (userMessage) => {
    let finalMessage = userMessage;
    let fileName = selectedFile ? selectedFile.name : null;

    if (selectedFile) {
        try {
            const fileContent = await readFileContent(selectedFile);
            finalMessage = `${userMessage}\n\nВміст файлу "${fileName}":\n${fileContent}`;
        } catch (error) {
            addMessage("Система", "Помилка читання файлу: ${error.message}");
            return;
        }
    }
};

```

```

    }
  }

  addMessage("Ви", finalMessage);
  await saveMessage("Ви", finalMessage);
  quickPromptsContainer.style.display = "none";
  submitBtn.classList.add("loading");
  submitBtn.disabled = true;

  const typingIndicator = showTypingIndicator();

  try {
    const messages = [
      { role: "system", content: systemMessageContent },
      { role: "user", content: finalMessage },
    ];

    const response = await fetch(API_URL, {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        "Authorization": `Bearer ${API_KEY}`,
      },
      body: JSON.stringify({
        model: "gpt-4o-mini",
        messages: messages,
      }),
    });

    removeTypingIndicator();
    submitBtn.classList.remove("loading");
    submitBtn.disabled = false;

    if (response.ok) {
      const data = await response.json();
      const botMessage = data.choices[0].message.content;
      addMessage(botName, botMessage);
      await saveMessage(botName, botMessage);
    } else {
      addMessage(botName, "Помилка сервера.");
    }
  } catch (error) {
    removeTypingIndicator();
    submitBtn.classList.remove("loading");
    submitBtn.disabled = false;
    addMessage(botName, `Помилка з'єднання: ${error.message}`);
  } finally {
    selectedFile = null;
    fileInput.value = "";
  }
};

// Швидкі промпти
const promptSets = [
  [
    { text: "Як підняти настрій?", prompt: "Як мені швидко покращити настрій?" },
    { text: "Допоможи розслабитися", prompt: "Як мені розслабитися після важкого дня?" },
    { text: "Що робити зі стресом?", prompt: "Як впоратися зі стресом прямо зараз?" }
  ],
  [
    { text: "Дай пораду на день", prompt: "Яка одна порада зробить мій день кращим?" },
    { text: "Як бути продуктивним?", prompt: "Як мені залишатися продуктивним сьогодні?" },
    { text: "Заспокой мене", prompt: "Допоможи мені заспокоїтися, я відчуваю тривогу." }
  ],
  [
    { text: "Як почати ранок?", prompt: "Як мені позитивно почати свій ранок?" },
    { text: "Допоможи з мотивацією", prompt: "Як знайти мотивацію для роботи чи навчання?" },
    { text: "Порадь вправу", prompt: "Яка проста вправа допоможе мені зараз?" }
  ]
];

```

```

];

const getRandomPromptSet = () => {
  const randomIndex = Math.floor(Math.random() * promptSets.length);
  return promptSets[randomIndex];
};

const showQuickPrompts = async () => {
  try {
    const response = await fetch(`/chat/history?userId=${userId}&sessionId=${sessionId}`);
    if (!response.ok) throw new Error("Не вдалося перевірити історію чату");
    const history = await response.json();
    quickPromptsContainer.innerHTML = "";
    if (history.length === 0) {
      const selectedPrompts = getRandomPromptSet();
      selectedPrompts.forEach(({ text, prompt }) => {
        const btn = document.createElement("button");
        btn.classList.add("quick-prompt-btn");
        btn.textContent = text;
        btn.addEventListener("click", (e) => {
          e.preventDefault();
          sendMessage(prompt);
        });
        quickPromptsContainer.appendChild(btn);
      });
      quickPromptsContainer.style.display = "flex";
    } else {
      quickPromptsContainer.style.display = "none";
    }
  } catch (error) {
    console.error("Помилка відображення швидких промптів:", error);
    addMessage("Система", "Не вдалося завантажити швидкі промпти.");
  }
};

```