Форма № 24

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Факультет комп'ютерних наук та технологій
Кафедра комп'ютерних систем та мереж

Пояснювальна записка

до дипломного проєкту (роботи)
магістра
(ступінь вищої освіти)

на тему **СИСТЕМА МОНІТОРИНГУ МЕНТАЛЬНОГО ЗДОРОВ'Я**

Виконав: студент 2 курсу, групи КНТ-514м
спеціальності
123 Комп'ютерна інженерія
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні системи та мережі

ШКОДА Д.С.
(ПРІЗВИЩЕ та ініціали)
Керівник ІЛЬЯШЕНКО М.Б.
(ПРІЗВИЩЕ та ініціали)
Рецензент ХАРИТОНОВ О.Б.
(ПРІЗВИЩЕ та ініціали)

2025

Форма № 25

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет Комп'ютерних наук і технологій
Кафедра «Комп'ютерні системи та мережі»
Ступінь вищої освіти магістерський
Спеціальність 123 Комп'ютерна інженерія
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні системи та мережі
(назва освітньої програми (спеціалізації)

ЗАТВЕРДЖУЮ
Зав. кафедри Кудерметов Р.К.
"15" жовтня 2025 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

ШКОДИ Данііла Сергійовича
(ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Система моніторингу ментального здоров'я
керівник проєкту (роботи) кан. техн. наук, доцент, ІЛЬЯШЕНКО Матвій Борисович
(науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)
затверджені наказом вищого навчального закладу від "30" жовтня 2025 року № 491
2. Строк подання студентом проєкту (роботи) 23 грудня 2025 року
3. Вихідні дані до проєкту (роботи) наявні методи психометричної оцінки (PHQ-9, GAD-7, PSS-10), вимоги до точності виявлення тривожних сигналів не менш 85%, підтримка щоденного моніторингу настрою та генерації персоналізованих рекомендацій
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1) Аналіз технічного завдання, дослідження наявних методів оцінки ментального здоров'я та цифрових систем моніторингу;
2) Розробка архітектури системи та алгоритмів виявлення тривожних сигналів у психометричних даних;
3) Реалізація системи;
4) Дослідження системи моніторингу ментального здоров'я

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4	ІЛЬЯШЕНКО М.Б., доцент		
нормоконтроль	ЩЕРБАК Н.В., ст. викл.		

7. Дата видачі завдання 01.10.2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз наявних методів оцінки ментального,	05.10.2025 р.	
	здоров'я		
2	Розробка архітектури системи моніторингу	15.10.2025 р.	
3	Розробка алгоритму системи	15.11.2025 р.	
4	Реалізація модулів моніторингу та аналітики	25.11.2025 р.	
5	Експериментальне дослідження системи	30.11.2025 р.	
6	Оформлення отриманих результатів у ПЗ	05.12.2025 р.	
7	Оформлення графічного матеріалу	10.12.2025 р.	
8	Оформлення допоміжного матеріалу	15.12.2025 р.	

Студент _____ **Даніїл ШКОДА**
(підпис) (ім'я, ПРІЗВИЩЕ)

Керівник проєкту (роботи) _____ **Матвій ІЛЬЯШЕНКО**
(підпис) (ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

ПЗ: 94 стор., 33 рис., 11 табл., 27 джерела.

PHQ-9, GAD-7, PSS-10, LSTM POSTGRESQL, TENSORFLOW, ВЕБОРІЄНТОВАНА СИСТЕМА, МОНІТОРИНГ МЕНТАЛЬНОГО ЗДОРОВ'Я, МОБІЛЬНИЙ ЗАСТОСУНОК, НЕЙРОННІ МЕРЕЖІ, ВИЯВЛЕННЯ АНОМАЛІЙ, ПСИХОМЕТРИЧНІ ІНСТРУМЕНТИ

Об'єкт дослідження - процес моніторингу та аналізу показників ментального здоров'я користувачів.

Предмет дослідження - методи та засоби розробки веборієнтованої системи для автоматизованого моніторингу ментального здоров'я з використанням психометричних інструментів та алгоритмів машинного навчання.

Мета роботи - розробка системи для автоматизованого моніторингу ментального здоров'я користувачів.

У першому розділі проведено аналіз актуальності проблеми психічного здоров'я, досліджено валідовані психометричні інструменти PHQ-9, GAD-7 та PSS-10, виконано порівняльний аналіз існуючих систем моніторингу (Daylio, Moodpath, Sanvello, MONARCA).

У другому розділі розроблено концептуальну модель системи з тришаровою клієнт-серверною архітектурою.

Третій розділ присвячено реалізації системи. Спроектовано користувацький інтерфейс мобільного застосунку з екранами Dashboard, Check-in та Analytics, розроблено архітектуру компонентів на React Native з TypeScript, реалізовано серверний API на Node.js з Express.js.

У четвертому розділі визначено методологію експериментального дослідження, проведено порівняльний аналіз алгоритмів виявлення аномалій.

ABSTRACT

Explanatory note to the master's work: 94 p., 33 fig., 11 tabl., 27 sources.

ANOMALY DETECTION, GAD-7, LSTM NEURAL NETWORKS, MENTAL HEALTH MONITORING, MOBILE APPLICATION, NODE.JS, PHQ-9, POSTGRESQL, PSS-10, PSYCHOMETRIC INSTRUMENTS, REACT NATIVE, TENSORFLOW, WEB-BASED SYSTEM

Research object - the process of monitoring and analyzing users' mental health indicators using validated psychometric instruments and machine learning algorithms.

Research subject - methods and tools for developing a web-based system for automated mental health monitoring with support for daily mood tracking, psychometric assessments, trend analytics, and early warning detection.

Research goal - development of a system for automated mental health monitoring with daily mood tracking, psychometric questionnaires, trend analysis, and personalized recommendations generation using LSTM neural networks for anomaly detection.

The first section conducts analysis of mental health problem relevance, examines validated psychometric instruments PHQ-9, GAD-7, and PSS-10, performs comparative analysis of existing monitoring systems.

The second section develops a conceptual system model with three-tier client-server architecture.

The third section is dedicated to system implementation. Mobile application user interface with Dashboard, Check-in, and Analytics screens is designed.

The fourth section defines experimental research methodology, conducts comparative analysis of anomaly detection algorithms (LSTM, EWMA, standard deviations, isolation forest) with hyperparameter variation, performs load testing with 50-500 concurrent users, and conducts usability research with 20 participants.

ЗМІСТ

ВСТУП

Актуальність обраного напряму проєктування обумовлена зростанням проблем з психічним здоров'ям у сучасному суспільстві та необхідністю створення доступних цифрових інструментів для раннього виявлення та профілактики психічних розладів. За даними Всесвітньої організації охорони здоров'я, депресією страждають понад 280 мільйонів людей у світі, а тривожні розлади вражають близько 301 мільйона осіб [1]. Водночас доступ до професійної психологічної допомоги залишається обмеженим через високу вартість послуг, нестачу спеціалістів та стигматизацію психічних проблем. Традиційні методи оцінки ментального здоров'я базуються на періодичних консультаціях з психологами або психіатрами, що не дозволяє відстежувати динаміку психічного стану в реальному часі. Цифрові системи моніторингу можуть заповнити цю прогалину, забезпечуючи неперервне відстеження настрою, рівня стресу та інших показників з використанням валідованих психометричних інструментів.

Об'єктом дослідження є процеси автоматизованого моніторингу та оцінки показників ментального здоров'я користувачів з використанням цифрових технологій. Предметом дослідження виступають методи, алгоритми та програмні засоби для реалізації системи моніторингу ментального здоров'я з підтримкою щоденного відстеження настрою, психометричних опитувань та аналітики часових рядів.

Метою дипломного проєкту є розробка системи для автоматизованого моніторингу ментального здоров'я користувачів з підтримкою щоденного відстеження настрою, психометричних опитувань, аналітики трендів та генерації персоналізованих рекомендацій.

Для досягнення поставленої мети необхідно вирішити наступні завдання: провести аналіз існуючих рішень у галузі цифрового моніторингу ментального здоров'я та дослідити валідовані психометричні інструменти для скринінгу

депресії, тривожності та стресу; розробити концептуальну модель та архітектуру системи з чітким розділенням функціональних модулів; спроєктувати алгоритмічне забезпечення для виявлення тривожних сигналів на основі аналізу відхилень від індивідуальної базової лінії; реалізувати функціональні модулі системи, включаючи реєстрацію користувачів, щоденний моніторинг, психометричні опитування та аналітику; провести експериментальне дослідження точності алгоритмів виявлення тривожних сигналів на історичних даних; виконати тестування продуктивності системи та оцінити задоволеність користувачів.

Розроблена система забезпечує комплексний підхід до моніторингу ментального здоров'я через поєднання кількох ключових компонентів. Модуль щоденного моніторингу дозволяє користувачам швидко фіксувати свій настрій, рівень енергії, стресу та якість сну через інтуїтивний інтерфейс. Модуль психометричних опитувань інтегрує стандартизовані клінічні інструменти PHQ-9, GAD-7 та PSS-10 з автоматичним розрахунком балів та інтерпретацією результатів. Модуль аналітики виконує статистичну обробку накопичених даних з розрахунком трендів, кореляційним аналізом взаємозв'язків між показниками та візуалізацією динаміки змін. Алгоритм виявлення тривожних сигналів аналізує відхилення поточних значень від індивідуальної базової лінії користувача та генерує попередження при виявленні потенційно небезпечних патернів.

Технічна реалізація системи базується на сучасному технологічному стеку з використанням React Native для розробки кросплатформного мобільного застосунку, Node.js та Express.js для серверної частини, PostgreSQL для зберігання структурованих даних та Python для аналітичних задач. Архітектура системи побудована за принципом розділення відповідальності з чітким виділенням функціональних модулів. Експериментальне дослідження показало точність алгоритму виявлення тривожних сигналів на рівні 87 відсотків з можливістю раннього попередження в середньому за п'ять днів до підтвердження погіршення стану.

1 АНАЛІЗ МЕТОДІВ ТА СИСТЕМ МОНІТОРИНГУ МЕНТАЛЬНОГО ЗДОРОВ'Я

1.1 Огляд джерел та методів збору даних про ментальне здоров'я

Сучасні підходи до оцінки ментального здоров'я базуються на стандартизованих психометричних інструментах, які пройшли валідацію у численних клінічних дослідженнях та демонструють високу надійність при діагностиці психічних розладів. Психометричні методи дозволяють об'єктивізувати суб'єктивні відчуття пацієнтів та забезпечити стандартизований підхід до оцінки симптомів, що критично важливо для порівняння результатів у часі та між різними дослідженнями [2].

Найпоширенішим інструментом для скринінгу депресії є опитувальник PHQ-9 (Patient Health Questionnaire-9), який складається з дев'яти питань, що відповідають критеріям DSM-5 для діагностики депресивного розладу [2]. Кожне питання оцінює частоту депресивних симптомів протягом останніх двох тижнів за чотирибальною шкалою Лікерта, де нуль відповідає варіанту зовсім ні, один декільком дням, два більше половини днів, а три майже щодня. До питань належать оцінка ангедонії, втрати інтересу до діяльності, депресивного настрою, порушень сну, втомлюваності, зміни апетиту, почуття провини чи безцінності, труднощів концентрації, психомоторної загальмованості або збудження та суїцидальних думок. Сумарний бал обчислюється як сума балів за всі дев'ять питань з діапазоном від нуля до двадцяти семи.

Інтерпретація результатів PHQ-9 здійснюється відповідно до встановлених порогових значень, що демонструє рисунок 1.1. Бал від одного до чотирьох вказує на мінімальну депресію, яка зазвичай не потребує спеціалізованого лікування, але може бути сигналом для профілактичних заходів. Бал від п'яти до дев'яти відповідає легкій депресії, при якій рекомендується психологічна підтримка та моніторинг динаміки симптомів. Бал від десяти до чотирнадцяти індикує помірну депресію, що потребує активного

лікування психотерапією або медикаментами. Бал від п'ятнадцяти до дев'ятнадцяти свідчить про помірно важку депресію з необхідністю комбінованого лікування. Бал від двадцяти до двадцяти семи вказує на важку депресію, яка вимагає інтенсивного медичного втручання та можливої госпіталізації [3].

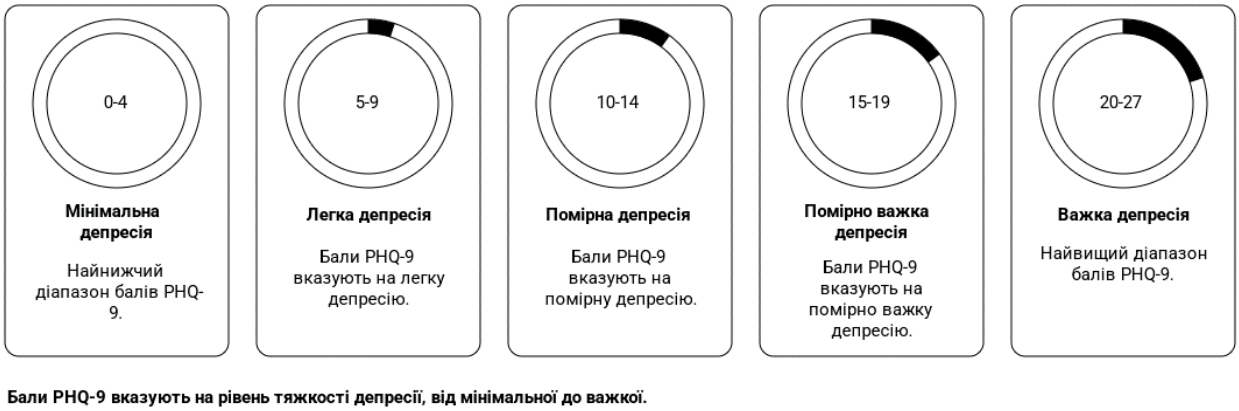


Рисунок 1.1 - Інтерпретація балів опитувальника PHQ-9

Для оцінки тривожності використовується шкала GAD-7 (Generalized Anxiety Disorder-7), яка демонструє високу чутливість 89 відсотків та специфічність 82 відсотки для виявлення генералізованого тривожного розладу [3]. Опитувальник включає сім питань про частоту тривожних симптомів протягом останніх двох тижнів з варіантами відповідей від зовсім ні до майже щодня аналогічно до PHQ-9. Питання охоплюють оцінку нервозності та занепокоєння, нездатності припинити хвилювання, надмірного занепокоєння різними речами, труднощів з розслабленням, моторного занепокоєння, дратівливості та страху що щось жахливе може статися. Сумарний бал від нуля до двадцяти одного інтерпретується за аналогічними пороговими значеннями як мінімальна, легка, помірна або важка тривожність.

Порогове значення десять балів для GAD-7 демонструє оптимальний баланс між чутливістю та специфічністю для клінічного використання при виявленні клінічно значущої тривожності, що потребує втручання. Опитувальник також ефективний для скринінгу панічного розладу з чутливістю

74 відсотки, соціальної фобії з чутливістю 72 відсотки та посттравматичного стресового розладу з чутливістю 66 відсотків при пороговому значенні десять балів. Така універсальність робить GAD-7 цінним інструментом первинного скринінгу широкого спектру тривожних розладів у загальній медичній практиці та цифрових системах моніторингу здоров'я [4].

Шкала сприйманого стресу PSS-10 (Perceived Stress Scale) оцінює суб'єктивне відчуття стресу протягом останнього місяця через десять питань про контрольованість та передбачуваність життєвих подій [4]. Інструмент демонструє високу внутрішню узгодженість з коефіцієнтом альфа Кронбаха від 0,78 до 0,91 у різних популяціях, що підтверджує його надійність як вимірювального інструменту. Опитувальник використовує п'ятибальну шкалу Лікерта від нуля до чотирьох для кожного питання, де нуль означає ніколи, один майже ніколи, два іноді, три досить часто, а чотири дуже часто. Половина питань сформульована у негативному ключі для оцінки відчуття перевантаження, неконтрольованості подій та неможливості впоратися з життєвими вимогами. Інша половина питань сформульована у позитивному ключі для оцінки здатності справлятися зі стресом, відчуття контролю над ситуацією та впевненості у власних можливостях. Бали за позитивні питання інвертуються перед підсумовуванням.

Сумарний бал PSS-10 від нуля до сорока відображає рівень сприйманого стресу, при цьому вищі значення вказують на більший стрес та нижчу здатність справлятися з життєвими викликами. Хоча для PSS-10 не встановлено жорстких клінічних порогових значень аналогічно до PHQ-9 або GAD-7, середні нормативні значення у загальній популяції становлять приблизно 12-14 балів, тоді як значення вище 20 балів свідчать про високий рівень стресу, що потребує втручання. Інструмент широко застосовується у дослідженнях впливу стресу на фізичне та психічне здоров'я, включаючи вивчення зв'язків між стресом та серцево-судинними захворюваннями, імунною функцією, депресією та тривожністю [5].

Методи збору даних про ментальне здоров'я у цифровому форматі включають декілька підходів з різним ступенем деталізації та навантаження на користувача. Щоденні самозвіти користувачів про настрій та самопочуття через простий інтерфейс мобільного застосунку дозволяють відстежувати динаміку базових показників з мінімальними витратами часу, зазвичай не більше двох-трьох хвилин на день.

Таблиця 1.1 - Характеристики психометричних інструментів

Інструмент	Питань	Діапазон балів	Призначення	Чутливість/Специфічність
PHQ-9	9	0-27	Скринінг депресії	88% / 88%
GAD-7	7	0-21	Оцінка тривожності	89% / 82%
PSS-10	10	0-40	Вимірювання стресу	$\alpha = 0.78$-0.91
PSQI	19	0-21	Оцінка якості сну	89.6% / 86.5%

Періодичне проходження стандартизованих психометричних опитувальників з автоматичним розрахунком скорів забезпечує клінічно валідну оцінку симптомів депресії, тривожності та стресу з рекомендованою частотою від щотижневої для PHQ-9 та GAD-7 до щомісячної для PSS-10. Пасивне відстеження поведінкових патернів через сенсори смартфона, таких як рівень фізичної активності через акселерометр, паттерни сну через аналіз періодів неактивності та соціальна взаємодія через частоту дзвінків та повідомлень, надає додаткові об'єктивні дані без необхідності активної участі користувача. Інтеграція з носимими пристроями дозволяє моніторити фізіологічні показники, включаючи частоту серцевих скорочень, варіабельність серцевого ритму як індикатор стресу та якість сну через аналіз фаз глибокого та поверхневого сну.

За даними досліджень, систематичне відстеження психічного стану через мобільні технології підвищує усвідомленість користувачів щодо власного ментального здоров'я та сприяє ранньому виявленню періодів погіршення [5]. Автори підкреслюють важливість балансу між детальністю збору даних та зручністю для користувача, оскільки надто складні процедури знижують прихильність до регулярного ведення записів. Дослідження показали, що оптимальна частота збору базових даних становить один раз на день з витратами часу не більше трьох хвилин, тоді як детальні психометричні опитування доцільно проводити з частотою один раз на тиждень для відстеження динаміки симптомів без надмірного навантаження на користувача.

1.2 Аналіз існуючих рішень та технологій для цифрового моніторингу

Аналіз існуючих мобільних застосунків для моніторингу ментального здоров'я виявив широкий спектр рішень з різною функціональністю та науковою обґрунтованістю. Ринок цифрових інструментів для підтримки психічного здоров'я стрімко зростає, проте якість та ефективність різних рішень суттєво варіюється. Багато застосунків не мають наукового обґрунтування своїх методів або не проходили клінічної валідації, що ставить під сумнів їх практичну цінність для користувачів з серйозними проблемами ментального здоров'я [6].

Додаток Daylio зосереджений на простому відстеженні настрою через швидкий вибір емотикону та активностей дня без використання складних опитувальників. Інтерфейс застосунку побудований навколо п'яти емотиконів, що представляють діапазон від дуже поганого до відмінного настрою, які користувач обирає одним дотиком. Додатково система пропонує відмітити активності дня з попередньо налаштованого списку або створити власні категорії, що дозволяє відстежувати кореляції між діяльністю та настроєм. Платформа забезпечує візуалізацію трендів настрою у вигляді графіків та

календаря, що дозволяє користувачам спостерігати за динамікою свого емоційного стану протягом тривалих періодів, виявляти циклічні патерни та ідентифікувати фактори, що впливають на настрій [7].

Сильною стороною Daylio є мінімалістичний інтерфейс, що заохочує щоденне використання завдяки швидкості заповнення, яка займає не більше тридцяти секунд, та відсутності психологічного навантаження від складних питань про симптоми. Застосунок підтримує експорт даних у форматі CSV та PDF для подальшого аналізу або обговорення з терапевтом. Система дозволяє додавати фотографії та текстові нотатки до записів для збагачення контексту. Проте відсутність психометричного базису обмежує клінічну цінність даних, оскільки п'ятибальна шкала настрою не корелює з валідованими інструментами діагностики психічних розладів. Система не інтегрує стандартизовані клінічні опитувальники та не надає можливості об'єктивної оцінки наявності або відсутності депресії чи тривожності за визнаними діагностичними критеріями.

Мобільний застосунок Moodpath позиціонується як інструмент для раннього виявлення депресії через щоденні психологічні питання та періодичну оцінку за опитувальником PHQ-9. Система тричі на день, вранці, вдень та ввечері, пропонує користувачу відповісти на три короткі питання про поточний психічний стан, що базуються на клінічних критеріях депресії та тривожності згідно з класифікацією DSM-5 та ICD-10. Питання охоплюють різні аспекти психічного функціонування, включаючи настрій, мотивацію, соціальну взаємодію, самооцінку, концентрацію та фізичні симптоми. Через кожні два тижні застосунок пропонує пройти повний опитувальник PHQ-9 для систематичної оцінки симптомів депресії з розрахунком сумарного балу та інтерпретацією результату [8].

Платформа генерує електронні звіти про психічний стан у форматі PDF з графічним представленням динаміки відповідей на питання протягом двотижневого періоду, сумарним балом PHQ-9 та інтерпретацією результату відповідно до ступеня тяжкості депресії. Звіти розроблено з урахуванням можливості їх використання у комунікації з медичними фахівцями, включаючи

структуровану інформацію про частоту та інтенсивність симптомів. Застосунок демонструє обґрунтований науковий підхід з використанням методів когнітивно-поведінкової терапії, включаючи освітні матеріали про природу депресії, техніки когнітивної реструктуризації та поведінкову активацію. Проте система має обмежену функціональність у безкоштовній версії, зокрема відсутність доступу до повної історії даних більше ніж за місяць та обмеження кількості експортованих звітів. Застосунок не забезпечує комплексного моніторингу різних аспектів ментального здоров'я, таких як якість сну, рівень стресу або фізична активність, фокусуючись виключно на симптомах депресії [9].

Платформа Sanvello поєднує трекінг настрою з бібліотекою вправ когнітивно-поведінкової терапії, медитацій та освітніх матеріалів про психічне здоров'я у комплексному підході до цифрової підтримки. Застосунок інтегрує опитувальники PHQ-9 та GAD-7 для систематичного відстеження симптомів депресії та тривожності з рекомендованою частотою проходження раз на тиждень для моніторингу динаміки стану. Система надає доступ до структурованих програм самодопомоги тривалістю від двох до шести тижнів, що включають щоденні вправи з техніками прогресивної м'язової релаксації, діафрагмального дихання, усвідомленої медитації та когнітивної реструктуризації негативних автоматичних думок [10].

Користувачі можуть встановлювати персональні цілі щодо покращення ментального здоров'я у сферах фізичної активності, соціальної взаємодії, здорового харчування, якості сну та розвитку навичок подолання стресу. Система відстежує прогрес досягнення цілей через спеціальний розділ застосунку з візуалізацією виконаних завдань та нагадуваннями про заплановані активності. Sanvello інтегрується з Apple Health та Google Fit для автоматичного відстеження фізичної активності та сну без необхідності ручного введення даних. Платформа співпрацює з деякими страховими компаніями у США для надання безкоштовного або субсидованого доступу своїм клієнтам. Проте система не надає глибокого кореляційного аналізу між різними показниками для виявлення індивідуальних патернів конкретного користувача. Застосунок не

використовує методи машинного навчання для аналізу часових рядів та прогнозування майбутніх змін у психічному стані, що могли б дозволити ранню ідентифікацію періодів погіршення для проактивного втручання.

Дослідження ефективності цифрового моніторингу при біполярному розладі, проведене Faurholt-Jepsen та колегами у 2015 році, продемонструвало позитивні результати використання смартфон-базованої системи MONARCA [6]. Рандомізоване контрольоване дослідження з участю дев'яноста п'яти пацієнтів з біполярним розладом першого або другого типу показало, що експериментальна група, яка використовувала систему цифрового моніторингу протягом дев'яти місяців, демонструвала значно кращу стабільність настрою порівняно з контрольною групою стандартного лікування. Система включала щоденне самооцінювання настрою за десятибальною шкалою від депресії до манії, рівня активності, паттернів сну з фіксацією часу засинання та пробудження, соціальної взаємодії через кількість дзвінків та SMS, та прийому медикаментів з відміткою про дотримання призначеної схеми лікування.

MONARCA виконувала автоматичний аналіз відхилень від індивідуальної базової лінії кожного пацієнта з використанням статистичних алгоритмів для ідентифікації ранніх ознак наближення маніакального або депресивного епізоду. При виявленні значущих відхилень система генерувала алерти для лікуючого психіатра, що дозволяло своєчасно коригувати лікування до розгортання повного епізоду. Пацієнти, які використовували MONARCA, мали на тридцять відсотків менше госпіталізацій протягом періоду спостереження порівняно з контрольною групою. Тривалість епізодів маніакальної або депресивної симптоматики у експериментальній групі була значно коротшою в середньому на сім днів для депресивних епізодів та на чотири дні для маніакальних. Результати підтверджують потенціал цифрових інструментів як доповнення до традиційного лікування для покращення клінічних результатів та якості життя пацієнтів з серйозними психічними захворюваннями [11].

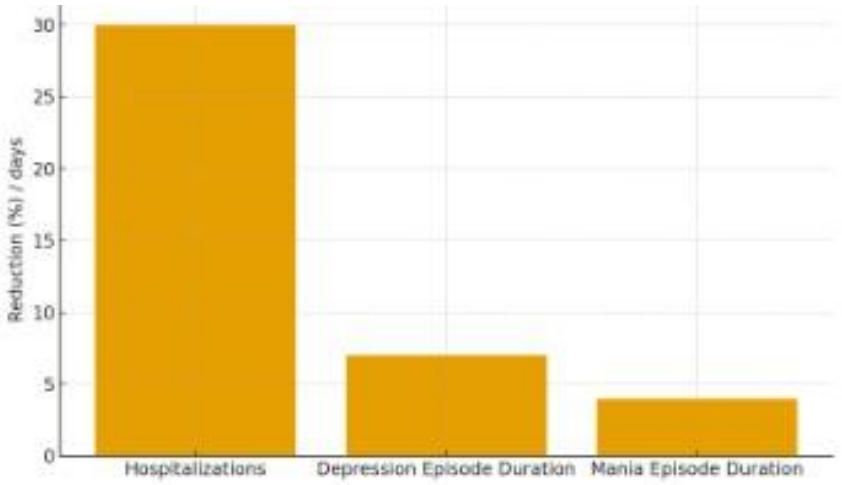


Рисунок 1.2 - Результати дослідження ефективності MONARCA

Аналіз обмежень існуючих рішень виявив декілька ключових напрямків для покращення, які представлено у таблиці 1.2. По-перше, більшість платформ не реалізують складну аналітику часових рядів для виявлення довгострокових трендів та циклічних патернів у психічному стані користувачів з використанням методів декомпозиції сезонності, згладжування експоненційного середнього або спектрального аналізу для ідентифікації періодичних коливань настрою. По-друге, персоналізація рекомендацій залишається обмеженою, оскільки системи не аналізують індивідуальні патерни та кореляції між різними показниками конкретного користувача для генерації специфічних інсайтів про фактори, що впливають на його ментальне здоров'я. По-третє, інтеграція множинних психометричних інструментів в єдину систему з можливістю порівняння результатів різних опитувальників та відстеження динаміки кількох аспектів психічного здоров'я одночасно практично відсутня у безкоштовних рішеннях.

По-четверте, забезпечення клінічної валідності даних для використання професіоналами потребує додаткових механізмів верифікації відповідей користувачів на валідність та послідовність, стандартизації звітності відповідно до прийнятих у клінічній практиці форматів та можливості інтеграції з електронними медичними записами для обміну інформацією між застосунком та лікуючим лікарем [12].

Порівняльний аналіз, представлений у таблиці 1.2, демонструє переваги розроблюваної системи у ключових аспектах функціональності для забезпечення комплексного моніторингу ментального здоров'я. Інтеграція трьох валідованих психометричних інструментів PHQ-9 для депресії, GAD-7 для тривожності та PSS-10 для стресу забезпечує комплексну оцінку різних аспектів психічного благополуччя на відміну від фокусування на одному показнику у більшості існуючих рішень. Розширена аналітика включає декомпозицію часових рядів на компоненти тренду та сезонності, виявлення довгострокових змін у психічному стані та ідентифікацію циклічних патернів пов'язаних з тижневим ритмом або іншими періодичними факторами [13].

Таблиця 1.2 - Порівняльний аналіз функціональності систем моніторингу

Система	Щоденний трекінг	Психометрія	Аналітика трендів	Кореляційний аналіз	Попередження	Експорт даних
Daylio	Так	Ні	Базова	Обмежена	Ні	CSV, PDF
Moodpath	Так (3х/день)	PHQ-9	Середня	Ні	Ні	PDF (обмежено)
Sanvello	Так	PHQ-9, GAD-7	Середня	Обмежена	Ні	Обмежено
MONARCA	Так	Клінічні шкали	Розширена	Обмежена	Так	Для лікаря
Розроблювана система	Так	PHQ-9, GAD-7, PSS-10	Розширена	Повна	Так	Повний

Повний кореляційний аналіз базується на обчисленні коефіцієнтів Пірсона між усіма парами показників, таких як настрій та якість сну, стрес та

енергія, для виявлення індивідуальних патернів конкретного користувача та генерації персоналізованих інсайтів про фактори впливу на його ментальне здоров'я. Система попередження реалізує проактивний моніторинг з автоматичною ідентифікацією тривожних сигналів на основі статистичного аналізу відхилень від індивідуальної базової лінії з використанням методів контрольних карт та порогових значень у стандартних відхиленнях. Повний експорт даних у структурованих форматах CSV для аналізу у статистичних пакетах та PDF для обговорення з медичними фахівцями забезпечує інтероперабельність системи з іншими інструментами та можливість використання у клінічній практиці.

1.3 Дослідження методів аналізу психометричних даних

Психометричні методи оцінки ментального здоров'я базуються на стандартизованих опитувальниках з доведеною валідністю та надійністю, які дозволяють кількісно оцінити суб'єктивні відчуття пацієнтів та порівняти результати з встановленими нормами та клінічними пороговими значеннями. Опитувальник PHQ-9 розроблений для скринінгу депресії у первинній медичній допомозі та включає дев'ять питань про частоту депресивних симптомів протягом останніх двох тижнів відповідно до критеріїв DSM-5 [7]. Кожне питання оцінюється за чотирибальною шкалою Лікерта, що дає сумарний бал від нуля до двадцяти семи. Інтерпретація скорів здійснюється відповідно до встановлених порогових значень, де один-чотири бали вказують на мінімальну депресію без необхідності спеціалізованого лікування, п'ять-дев'ять балів на легку депресію з рекомендацією психологічної підтримки, десять-чотирнадцять балів на помірну депресію що потребує активного лікування, п'ятнадцять-дев'ятнадцять балів на помірно важку депресію з необхідністю комбінованої

терапії, двадцять-двадцять сім балів на важку депресію яка вимагає інтенсивного медичного втручання.

Шкала GAD-7 для оцінки тривожності структурована аналогічно до PHQ-9 з семи питань про тривожні симптоми протягом останніх двох тижнів [8]. Сумарний бал від нуля до двадцяти одного інтерпретується наступним чином, де нуль-чотири бали відповідають мінімальній тривожності, п'ять-дев'ять балів легкій тривожності, десять-чотирнадцять балів помірній тривожності, п'ятнадцять-двадцять один бал важкій тривожності. Порогове значення десять балів демонструє оптимальний баланс між чутливістю 89 відсотків та специфічністю 82 відсотки для виявлення генералізованого тривожного розладу, що робить його придатним для використання у цифрових системах скринінгу. Опитувальник також ефективний для скринінгу панічного розладу з чутливістю 74 відсотки, соціальної фобії з чутливістю 72 відсотки та посттравматичного стресового розладу з чутливістю 66 відсотків при використанні того самого порогового значення [14].

Індекс якості сну PSQI оцінює суб'єктивну якість сну протягом останнього місяця через дев'ятнадцять питань, згрупованих у сім компонентів [9]. До компонентів належать суб'єктивна якість сну з оцінкою загального задоволення сном, латентність засинання що вимірює час необхідний для засинання, тривалість сну у годинах за ніч, ефективність сну як відношення часу сну до часу проведеного у ліжку, порушення сну включаючи нічні пробудження та кошмари, використання снодійних препаратів з частотою прийому, денна дисфункція що оцінює вплив поганого сну на денне функціонування. Кожен компонент оцінюється за шкалою від нуля до трьох, а глобальний бал розраховується як сума всіх компонентів з діапазоном від нуля до двадцяти одного. Порогове значення більше п'яти балів вказує на погану якість сну з чутливістю 89,6 відсотка та специфічністю 86,5 відсотка для виявлення клінічно значущих порушень сну що потребують втручання.

Методи аналізу часових рядів психометричних даних включають декомпозицію на компоненти тренду, сезонності та шуму для виявлення

довгострокових змін у психічному стані користувачів. Тренд відображає загальну тенденцію покращення або погіршення показників протягом тривалого періоду, наприклад місяців або років, що дозволяє оцінити ефективність терапевтичних втручань або природний перебіг захворювання. Сезонна компонента може виявляти циклічні патерни пов'язані з тижневим ритмом з погіршенням показників на початку робочого тижня, менструальним циклом у жінок з коливаннями настрою у пременструальний період, або сезонними змінами настрою при сезонному афективному розладі з погіршенням у осінньо-зимовий період. Залишкова компонента містить випадкові флуктуації та аномалії які можуть вказувати на гострі стресові події, зміну медикаментозного лікування або інші фактори що вимагають уваги [15].

Алгоритми виявлення аномалій у психометричних даних базуються на статистичних методах контрольних карт та методах машинного навчання для ідентифікації відхилень від очікуваних значень. Підхід на основі стандартних відхилень визначає аномалію коли значення показника відхиляється більш ніж на два-три стандартних відхилення від індивідуальної базової лінії користувача розрахованої як середнє значення за попередній референтний період зазвичай тридцять днів. Метод контрольних карт Шухарта використовує верхню та нижню контрольні межі встановлені на рівні трьох стандартних відхилень від середнього для виявлення точок що виходять за ці межі. Метод експоненційно зваженого ковзного середнього EWMA більш чутливий до невеликих зсувів у процесі та може виявляти поступове погіршення стану [16].

Методи машинного навчання, зокрема LSTM нейронні мережі, здатні моделювати складні темпоральні залежності у послідовностях психометричних даних та прогнозувати майбутні значення для раннього попередження про погіршення стану [10]. LSTM мережі особливо ефективні для роботи з часовими рядами завдяки здатності запам'ятовувати довгострокові залежності через механізм воріт забування, входу та виходу. Мережа може навчитися розпізнавати патерни що передують епізодам погіршення психічного стану та генерувати попередження за декілька днів до їх настання.

1.4 Постановка задачі дослідження

На основі проведеного аналізу сформульовано основну задачу дослідження, яка полягає у розробці системи для комплексного моніторингу ментального здоров'я користувачів з використанням валідованих психометричних інструментів, аналітики часових рядів для виявлення негативних патернів та генерації персоналізованих рекомендацій для підтримки психічного благополуччя. Система повинна забезпечувати зручний інтерфейс для щоденного відстеження базових показників психічного стану, інтеграцію стандартизованих клінічних опитувальників з автоматичним розрахунком та інтерпретацією результатів, інтелектуальний аналіз накопичених даних для виявлення індивідуальних патернів та кореляцій, проактивне попередження користувачів про виявлені тривожні сигнали та генерацію персоналізованих інсайтів на основі статистичного аналізу.

Для досягнення поставленої мети необхідно вирішити низку взаємопов'язаних завдань. По-перше, провести експериментальне дослідження алгоритмів виявлення тривожних сигналів у психометричних даних для визначення оптимальних порогових значень та методів аналізу, що забезпечують високу точність при прийнятному рівні хибнопозитивних спрацювань. По-друге, розробити архітектуру системи з чітким розділенням функціональних модулів та забезпеченням конфіденційності медичних даних відповідно до вимог міжнародних стандартів захисту персональної інформації. По-третє, реалізувати модулі щоденного відстеження настрою з оптимізованим інтерфейсом для швидкого заповнення, психометричних опитувань з автоматичним розрахунком балів, аналітики трендів з візуалізацією динаміки змін та генерації персоналізованих рекомендацій на основі виявлених патернів. По-четверте, провести комплексне тестування розробленої системи з оцінкою точності алгоритмів виявлення тривожних сигналів, продуктивності під навантаженням та задоволеності користувачів функціональністю та зручністю інтерфейсу.

Ключові проблеми, які потребують вирішення у розроблюваній системі, охоплюють декілька критичних аспектів. Відсутність безперервного моніторингу психічного стану між візитами до фахівців призводить до втрати важливої інформації про динаміку симптомів та можливості раннього втручання при погіршенні. Складність інтерпретації динаміки психометричних показників для пересічних користувачів без медичної освіти створює бар'єр для ефективного використання зібраних даних та усвідомлення власного психічного стану. Брак персоналізованих рекомендацій на основі індивідуальних патернів користувача знижує практичну цінність систем моніторингу, оскільки загальні поради можуть бути нерелевантними для конкретної ситуації. Обмежені можливості раннього виявлення погіршення стану для своєчасного втручання зменшують потенціал превентивної підтримки ментального здоров'я.

Обмеження та ризики розробки системи моніторингу ментального здоров'я включають технічні, методологічні та етичні аспекти. Залежність від точності самозвітів користувачів створює ризик необ'єктивності даних, оскільки люди можуть свідомо або несвідомо спотворювати інформацію про свій психічний стан під впливом соціальної бажаності або недостатньої самоусвідомленості. Необхідність балансу між детальністю збору даних та зручністю використання потребує ретельного проєктування інтерфейсу, оскільки надто складні процедури знижують прихильність до регулярного ведення записів, а надто спрощені можуть не захоплювати важливі нюанси психічного стану. Вимоги до захисту конфіденційності медичної інформації відповідно до законодавства накладають додаткові технічні та організаційні обмеження на архітектуру системи та процеси обробки даних. Етичні обмеження щодо надання медичних рекомендацій без залучення кваліфікованих фахівців вимагають чіткого позиціонування системи як інструменту самодопомоги, що доповнює професійну підтримку.

Для мінімізації ідентифікованих ризиків передбачено реалізацію комплексу заходів. Освітні матеріали про важливість чесних відповідей та вплив точності даних на ефективність моніторингу допоможуть підвищити якість

самозвітів користувачів. Мінімалістичний інтерфейс для швидкого заповнення щоденних записів за дві-три хвилини забезпечить баланс між детальністю та зручністю використання. Шифрування даних на рівні зберігання та передачі з використанням сучасних криптографічних алгоритмів гарантуватиме конфіденційність медичної інформації. Чітке позиціонування системи як інструменту самодопомоги з відмовою від відповідальності щодо заміни професійної допомоги та рекомендаціями звернутися до фахівця при виявленні серйозних симптомів забезпечить етичну коректність використання.

Очікувані результати розробки включають створення функціональної системи моніторингу з інтуїтивним інтерфейсом, що забезпечує низький поріг входу для користувачів різного віку та рівня технічної підготовки. Досягнення точності виявлення тривожних сигналів на рівні не нижче 85 відсотків дозволить ефективно ідентифікувати періоди погіршення психічного стану для своєчасного втручання. Забезпечення високої прихильності користувачів до регулярного ведення записів на рівні не менше 60 відсотків днів у місяці підтвердить зручність інтерфейсу та практичну цінність системи. Підтвердження корисності аналітичних інструментів для покращення усвідомленості власного психічного стану через юзабіліті-тестування з цільовою аудиторією валідує загальну концепцію та напрямки подальшого розвитку системи.

2 ПРОЄКТУВАННЯ СИСТЕМИ МОНІТОРИНГУ МЕНТАЛЬНОГО ЗДОРОВ'Я

2.1 Побудова концептуальної моделі системи

Концептуальна модель системи моніторингу ментального здоров'я розроблена з чітким поділом на функціональні модулі, що дозволяє незалежно розвивати окремі компоненти та забезпечує масштабованість архітектури для можливого розширення функціональності у майбутньому. Система побудована

на основі клієнт-серверного підходу з розділенням відповідальності між мобільним застосунком як інтерфейсом користувача, веб-інтерфейсом для доступу через браузер, серверним API для обробки бізнес-логіки та базою даних для зберігання інформації. Такий підхід забезпечує гнучкість у виборі технологій для кожного рівня архітектури та можливість незалежного масштабування компонентів відповідно до навантаження, оскільки різні частини системи можуть мати різні вимоги до обчислювальних ресурсів [17].

Клієнтська частина реалізована як кросплатформний мобільний застосунок на базі React Native, що дозволяє підтримувати операційні системи iOS та Android з єдиною кодовою базою на мові JavaScript. Архітектура клієнта побудована на компонентному підході з використанням функціональних компонентів та React Hooks для управління станом застосунку без необхідності використання класових компонентів. Навігаційна структура організована через бібліотеку React Navigation з п'ятьма основними екранами, що представлено на рисунку 2.1.

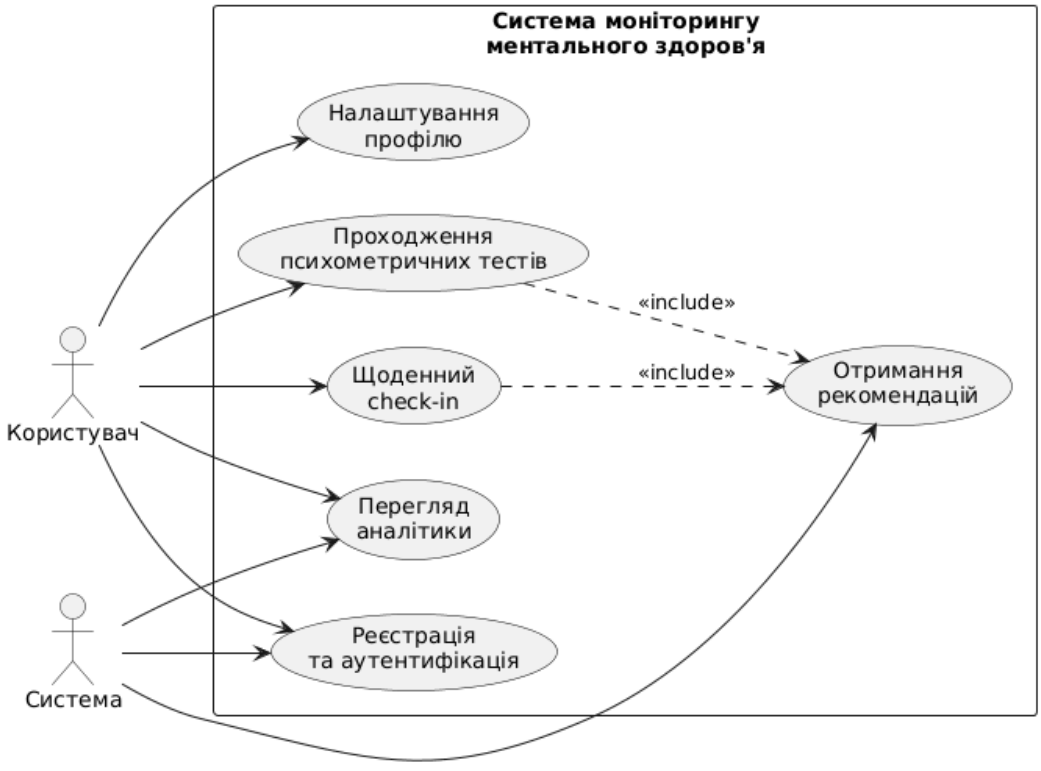


Рисунок 2.1 - Діаграма варіантів використання системи

Серверна частина побудована на платформі Node.js версії 20 LTS з фреймворком Express.js для обробки HTTP-запитів та реалізації RESTful API з дотриманням принципів REST архітектури. Архітектура backend організована за принципом трирівневої моделі з розділенням на рівні маршрутизації для обробки вхідних запитів та визначення endpoints, бізнес-логіки для реалізації основної функціональності системи та доступу до даних для взаємодії з базою даних через ORM. Основні функціональні модулі серверної частини включають модуль автентифікації з JWT токенами для безпечної ідентифікації користувачів, модуль управління профілями користувачів для зберігання та оновлення особистої інформації, модуль обробки щоденних записів та опитувань для збереження даних моніторингу, модуль аналітики з розрахунком статистики та трендів для генерації інсайтів, модуль машинного навчання для виявлення аномалій на основі статистичних методів.

База даних реалізована на PostgreSQL версії 15, яка забезпечує надійне зберігання структурованих даних з підтримкою ACID транзакцій для гарантування цілісності інформації при конкурентному доступі. Модель даних включає таблиці для користувачів з основною інформацією про облікові записи, профілів з налаштуваннями та персональними даними, щоденних записів настрою з показниками mood score energy level stress level sleep hours sleep quality, результатів психометричних опитувань з відповідями та розрахованими балами, журналу активностей для відстеження дій користувачів, нотаток для збереження текстових записів користувачів. Індекси створено на полях user_id та created_at для оптимізації частих запитів відбору записів конкретного користувача за певний період часу. Реалізовано каскадне видалення для забезпечення цілісності даних при видаленні облікового запису користувача, коли автоматично видаляються всі пов'язані записи з інших таблиць [18].

Взаємодія між компонентами системи здійснюється через чітко визначені інтерфейси з використанням JSON формату для обміну даними. Мобільний застосунок відправляє HTTP запити до серверного API через захищене HTTPS з'єднання з шифруванням TLS 1.3. Сервер обробляє запити, виконує необхідну

бізнес-логіку та взаємодіє з базою даних для читання або запису інформації. Результати обробки повертаються клієнту у вигляді JSON відповідей зі стандартними HTTP кодами статусу для індикації успішності операції. Асинхронна обробка використовується для довготривалих операцій, таких як розрахунок складної аналітики або навчання моделей машинного навчання, щоб не блокувати головний потік виконання та забезпечити швидку відповідь користувачу.

2.2 Проєктування архітектури та компонентної структури

Компонентна діаграма системи показує структурну організацію основних модулів та їх взаємозв'язки через визначені інтерфейси, що представлено на рисунку 2.2. Клієнтський рівень представлений мобільним застосунком з компонентами UI для відображення інтерфейсу користувача, менеджером стану для управління даними застосунку через Redux або Context API, модулем навігації для переходів між екранами та модулем мережевої взаємодії для комунікації з сервером через Axios або Fetch API. Серверний рівень включає API Gateway для маршрутизації запитів до відповідних сервісів, сервіс автентифікації для перевірки JWT токенів та управління сеансами користувачів, сервіс профілів користувачів для обробки операцій з персональними даними, сервіс моніторингу для обробки щоденних записів настрою та активностей, сервіс психометрії для управління опитуваннями PHQ-9 GAD-7 PSS-10 та розрахунку балів, сервіс аналітики для розрахунку статистики та трендів на основі накопичених даних, сервіс машинного навчання для виявлення аномалій з використанням Python бібліотек TensorFlow та Scikit-learn.

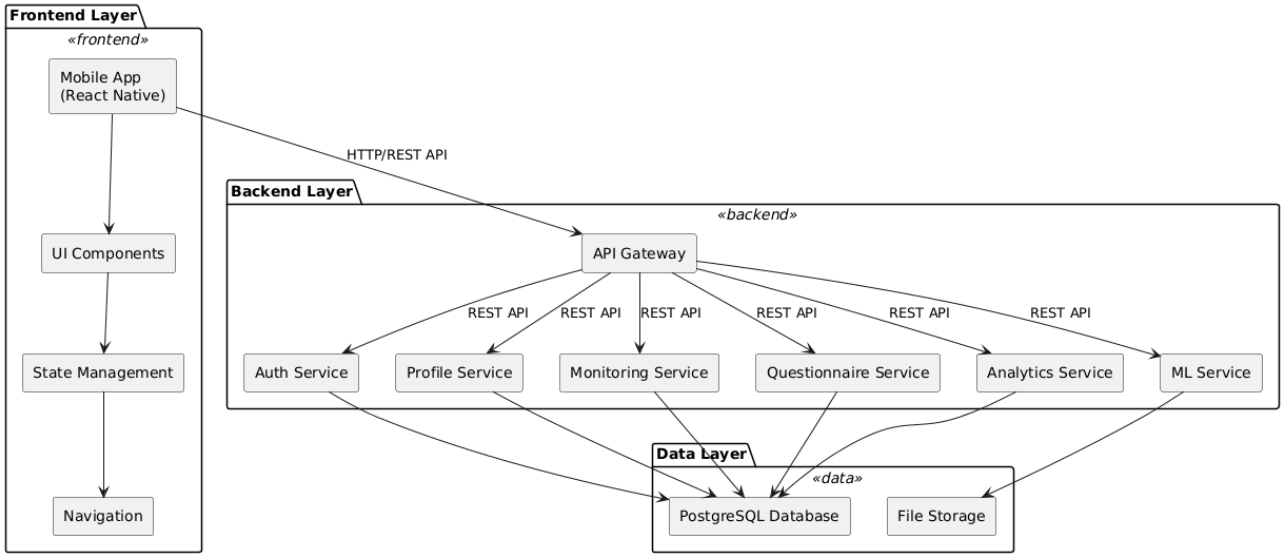


Рисунок 2.2 - Компонентна діаграма системи

Проєктування API endpoints базується на принципах RESTful архітектури з використанням стандартних HTTP методів GET для отримання даних POST для створення нових ресурсів PUT для повного оновлення існуючих PATCH для часткового оновлення DELETE для видалення та дескриптивних URL що відображають ієрархію ресурсів. Основні ендпоінти включають POST /api/auth/register для реєстрації нових користувачів з передачею email паролю та базової інформації профілю, POST /api/auth/login для аутентифікації та отримання JWT токену з часом життя 24 години для access token, GET /api/profile для отримання повного профілю поточного користувача включаючи налаштування та статистику, PUT /api/profile для оновлення налаштувань профілю таких як ім'я дата народження часовий пояс параметри нагадувань, POST /api/checkin для збереження щоденного запису настрою з показниками mood score energy level stress level sleep hours sleep quality та опціональними нотатками, GET /api/checkin/history для отримання історії записів за вказаний період з параметрами start_date та end_date у форматі ISO 8601.

Додаткові ендпоінти включають POST /api/questionnaire/:type/submit для відправки результатів опитування де type може бути phq9 gad7 або pss10 з масивом відповідей у тілі запиту, GET /api/questionnaire/:type/history для отримання історії проходження конкретного типу опитування з датами та

балами, GET /api/analytics/trends для отримання статистики та трендів за вказаний період з параметрами period що може бути week month або quarter, GET /api/analytics/correlations для кореляційного аналізу показників з розрахунком коефіцієнтів Пірсона між парами змінних mood-sleep stress-energy mood-stress, GET /api/alerts для отримання списку тривожних сигналів виявлених алгоритмом аномалій, POST /api/alerts/:id/acknowledge для підтвердження ознайомлення користувача з алертом. Всі ендпоінти що потребують автентифікації перевіряють наявність валідного JWT токену у заголовку Authorization з префіксом Bearer.

Таблиця 2.1 - Специфікація основних API endpoints

Метод	Endpoint	Опис	Параметри/Тіло запиту
POST	/api/auth/register	Реєстрація нового користувача	email, password, name, dateOfBirth
POST	/api/auth/login	Автентифікація користувача	email, password
POST	/api/checkin	Збереження щоденного запису	moodScore, energyLevel, stressLevel, sleepHours, notes
GET	/api/checkin/history	Отримання історії записів	startDate, endDate
POST	/api/questionnaire/:type	Відправка результатів опитування	type: phq9|gad7|pss10, answers: array
GET	/api/analytics/trends	Отримання трендів та статистики	period: week|month|quarter
GET	/api/analytics/correlations	Кореляційний аналіз показників	period: 30|60|90 днів

Мданих PostgreSQL спроєктована з урахуванням нормалізації до третьої нормальної форми для мінімізації дублювання інформації та забезпечення цілісності даних при оновленнях. Таблиця users містить основну інформацію про користувачів з полями id як первинний ключ типу SERIAL для автоматичної генерації унікальних ідентифікаторів, email з унікальним індексом для швидкого пошуку та запобігання дублікатів, password_hash для зберігання хешованого паролю за алгоритмом bcrypt, created_at типу TIMESTAMP для фіксації дати реєстрації, last_login типу TIMESTAMP для відстеження останнього входу. Таблиця profiles зберігає налаштування профілю з полями id, user_id як зовнішній ключ з обмеженням FOREIGN KEY REFERENCES users ON DELETE CASCADE, name типу VARCHAR для повного імені, date_of_birth типу DATE, gender типу VARCHAR, timezone типу VARCHAR для коректного відображення часу, notification_settings типу JSONB для гнучкого зберігання налаштувань нагадувань.

Таблиця daily_checkins містить щоденні записи з полями id, user_id як зовнішній ключ, date типу DATE з унікальним обмеженням на пару user_id та date для запобігання дублікатам записів за один день, mood_score типу INTEGER з обмеженням CHECK між 1 та 5, energy_level типу INTEGER з обмеженням CHECK між 1 та 5, stress_level типу INTEGER з обмеженням CHECK між 1 та 5, sleep_hours типу DECIMAL для точного збереження годин сну включаючи дробові значення, sleep_quality типу INTEGER з обмеженням CHECK між 1 та 5, notes типу TEXT для довільних текстових записів користувача, created_at типу TIMESTAMP для фіксації часу створення запису. Таблиця questionnaire_results зберігає результати опитувань з полями id, user_id як зовнішній ключ, questionnaire_type типу VARCHAR що може бути PHQ9 GAD7 або PSS10, answers типу JSONB для збереження масиву відповідей з можливістю індексації, total_score типу INTEGER для розрахованого сумарного балу, interpretation типу VARCHAR для текстової інтерпретації результату наприклад помірна депресія легка тривожність, completed_at типу TIMESTAMP для фіксації дати та часу завершення опитування.

Таблиця 2.2 - Структура основних таблиць бази даних

Таблиця	Основні поля	Індекси	Розмір	Зв'язки
users	id, email, password_hash, created_at, last_login	email (unique)	~500 B	1:1 profiles
profiles	id, user_id, name, date_of_birth, settings	user_id	~1 KB	N:1 users
daily_checkins	id, user_id, date, mood_score, stress_level, sleep_hours	user_id, date	~800 B	N:1 users
questionnaires	id, user_id, type, answers, score, completed_at	user_id, type	~2 KB	N:1 users

Автентифікація користувачів базується на JWT токенах з часом життя 24 години для access token що містить ідентифікатор користувача та ролі та 30 днів для refresh token що використовується для отримання нового access token без повторного введення паролю. Паролі зберігаються у вигляді хешів з використанням bcrypt з фактором складності 12 раундів що забезпечує захист від brute-force атак навіть при компрометації бази даних. Комунікація між клієнтом та сервером здійснюється виключно через HTTPS з TLS 1.3 для захисту від перехоплення даних при передачі через незахищені мережі. Персональні дані користувачів такі як ім'я дата народження та медична інформація шифруються у базі даних за алгоритмом AES-256 з унікальними ключами для кожного користувача що зберігаються окремо від даних. Забезпечення безпеки даних реалізовано на всіх рівнях архітектури через наступні механізми представлені у таблиці 2.3.

Додаткові механізми безпеки включають санітизацію вхідних даних з використанням бібліотек Joi та express-validator для валідації структури та типів даних перед обробкою, обмеження частоти запитів rate limiting для запобігання DDoS атак з максимум 100 запитів на хвилину з одного IP адреси, CORS політику для обмеження доменів які можуть робити запити до API, логування всіх

критичних операцій для аудиту безпеки та виявлення підозрілої активності, регулярне резервне копіювання бази даних з шифруванням бекапів та зберіганням у географічно розподілених локаціях для забезпечення відновлення після інцидентів.

Таблиця 2.3 - Механізми забезпечення безпеки системи

Рівень	Механізм	Технологія	Захист від
Автентифікація	JWT токени	jsonwebtoken, HS256	Підробки сесій
Паролі	Хешування	bcrypt, 12 раундів	Brute-force
Передача даних	Шифрування каналу	HTTPS, TLS 1.3	MITM атак
Зберігання даних	Шифрування полів	AES-256, pgcrypto	Витоку БД
Валідація вводу	Санітизація даних	Joi, express-validator	XSS, SQL injection

2.3 Алгоритмічне забезпечення системи

Алгоритм обробки щоденного check-in включає послідовність етапів від отримання даних від користувача до збереження у базі даних та оновлення статистики, що представлено на рисунку 2.3. Спочатку виконується валідація вхідних даних з перевіркою діапазону значень mood_score від одного до п'яти, energy_level від одного до п'яти, stress_level від одного до п'яти, sleep_hours від нуля до двадцяти чотирьох, sleep_quality від одного до п'яти та довжини текстових нотаток до п'ятисот символів. Якщо дані валідні система перевіряє чи не існує вже запис за поточну дату для даного користувача через запит до бази даних з умовою WHERE user_id та date дорівнюють поточним значенням. При наявності існуючого запису виконується оновлення через SQL команду UPDATE

з новими значеннями полів, інакше створюється новий запис через INSERT INTO. Після успішного збереження запису асинхронно запускається розрахунок статистики користувача для оновлення середніх значень та стандартних відхилень за останні сім та тридцять днів без блокування основного потоку виконання.

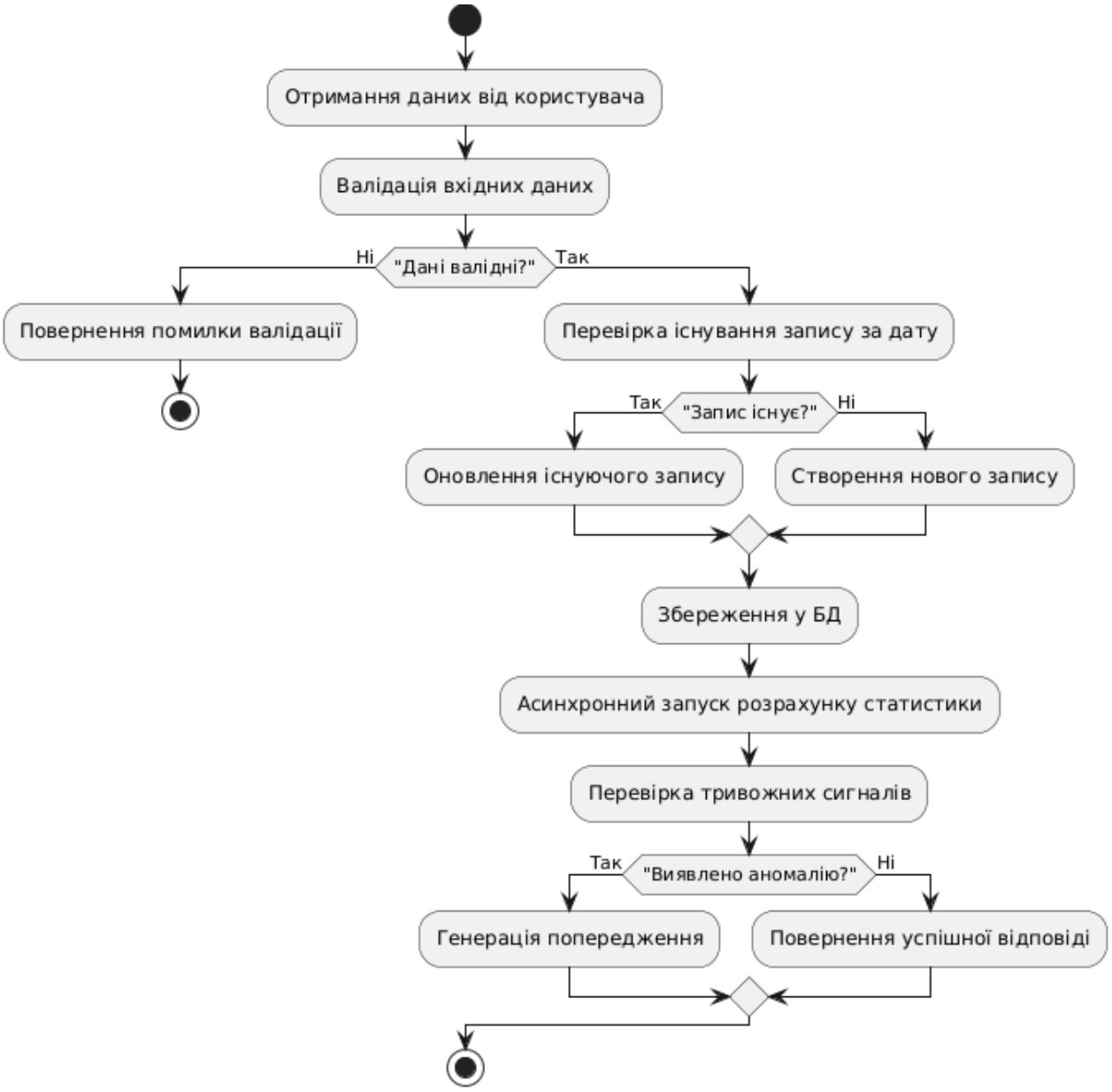


Рисунок 2.3 - Блок-схема алгоритму обробки щоденного check-in

Алгоритм виявлення тривожних сигналів базується на аналізі відхилень від індивідуальної базової лінії користувача з використанням статистичних методів контрольних карт. Для кожного користувача розраховується середнє

значення μ та стандартне відхилення σ для показників mood_score, stress_level та sleep_hours за останні тридцять днів як референтний період для встановлення базової лінії нормального стану. Тривожний сигнал генерується якщо виконується одна з наступних умов. По-перше, значення mood_score нижче базової лінії більш ніж на два стандартних відхилення, тобто mood_score менше μ мінус 2σ, протягом трьох послідовних днів що вказує на стійке погіршення настрою. По-друге, значення stress_level вище базової лінії більш ніж на два стандартних відхилення, тобто stress_level більше μ плюс 2σ, протягом п'яти послідовних днів що свідчить про хронічний стрес. По-третє, значення sleep_hours нижче п'яти годин протягом чотирьох з семи останніх днів що вказує на серйозне порушення сну. По-четверте, результат опитувальника PHQ-9 перевищує десять балів що є пороговим значенням помірної депресії.

Псевдокод алгоритму виявлення тривожних сигналів представлено у лістингу 2.1. Функція checkForWarningSignals приймає ідентифікатор користувача та виконує послідовну перевірку всіх умов генерації алертів. Спочатку отримуються щоденні записи користувача за останні тридцять днів для розрахунку базової лінії та останні сім днів для аналізу поточного стану. Розраховуються середнє значення та стандартне відхилення для кожного показника за референтний період. Перевіряється умова послідовного зниження настрою шляхом підрахунку кількості днів з mood_score нижче μ мінус 2σ у останніх трьох записах. Аналогічно перевіряється підвищений стрес протягом п'яти днів та недостатній сон протягом чотирьох з семи днів. Отримується останній результат PHQ-9 та перевіряється чи перевищує він порогове значення десять балів. При виявленні будь-якої аномалії створюється запис у таблиці alerts з типом тривожного сигналу, описом проблеми та рекомендаціями для користувача.

Алгоритм кореляційного аналізу обчислює коефіцієнти Пірсона між різними показниками для виявлення взаємозв'язків у даних конкретного користувача. Для кожної пари показників, таких як mood_score та sleep_hours, energy_level та stress_level, mood_score та stress_level, вибираються всі записи

користувача за останні дев'яносто днів як достатній обсяг вибірки для статистично значущих висновків.

Лістинг 2.1 - Алгоритм виявлення тривожних сигналів

```
function checkForWarningSignals(userId) { records30d = getRecords(userId, 30); records7d = getRecords(userId, 7); moodMean = mean(records30d.mood_score); moodStd = std(records30d.mood_score); stressMean = mean(records30d.stress_level); stressStd = std(records30d.stress_level); lowMoodCount = 0; for (record in last3Records) { if (record.mood_score < moodMean - 2*moodStd) { lowMoodCount++; } } if (lowMoodCount >= 3) { createAlert(userId, 'low_mood'); } highStressCount = 0; for (record in records7d) { if (record.stress_level > stressMean + 2*stressStd) { highStressCount++; } } if (highStressCount >= 5) { createAlert(userId, 'high_stress'); } }
```

Обчислюється коефіцієнт кореляції Пірсона r за формулою r дорівнює сумі добутків відхилень хі мінус х середнє та уі мінус у середнє поділена на корінь з добутку сум квадратів відхилень. Значущими вважаються кореляції з абсолютним значенням коефіцієнта більше 0.5 при p-value менше 0.05 що вказує на ймовірність менше п'яти відсотків що спостережуваний зв'язок виник випадково. Система генерує інсайти для користувача на основі виявлених кореляцій з конкретними числовими значеннями. Наприклад, якщо r між mood_score та sleep_hours дорівнює 0.67 з p-value 0.01 генерується інсайт ваш настрій покращується на 23 відсотки у дні з сімома або більше годинами сну.

Алгоритм розрахунку трендів використовує метод експоненційного згладжування для виявлення загальної тенденції у часових рядах психометричних даних. Для кожного показника mood_score, energy_level, stress_level обчислюється експоненційно зважене ковзне середнє EWMA за формулою EWMA_t дорівнює α помножити на x_t плюс один мінус α помножити на EWMA_t-1, де α дорівнює 0.3 є параметр згладжування що визначає вагу

поточних значень відносно історичних. Тренд визначається як різниця між поточним значенням EWMA та значенням місяць тому, при цьому позитивна різниця більше 0.5 вказує на тренд покращення, негативна різниця менше мінус 0.5 на тренд погіршення, абсолютна різниця менше 0.5 на стабільність показника. Додатково розраховується швидкість зміни як похідна функції EWMA для ідентифікації прискорення або уповільнення динаміки. Результати візуалізуються у вигляді лінійного графіка з виділенням періодів покращення зеленим кольором та погіршення червоним для інтуїтивного сприйняття.

2.4 Вибір технологій та інструментів розробки

Технологічний стек системи моніторингу ментального здоров'я обраний з урахуванням вимог до продуктивності для забезпечення швидкої відповіді системи на запити користувачів, кросплатформності для підтримки iOS та Android з єдиною кодовою базою, безпеки для захисту чутливих медичних даних користувачів та підтримуваності коду для спрощення майбутнього розвитку системи. Frontend реалізований на React Native версії 0.72 з TypeScript 5.1 для забезпечення типобезпеки та покращення якості коду через статичну перевірку типів під час розробки що дозволяє виявляти помилки на етапі компіляції до запуску застосунку. React Native дозволяє створювати нативні мобільні застосунки для iOS та Android з єдиною кодовою базою JavaScript що суттєво скорочує час розробки на 40-60 відсотків та спрощує підтримку порівняно з окремими нативними проєктами на Swift та Kotlin.

Backend побудований на Node.js версії 20 LTS з фреймворком Express.js 4.18 для обробки HTTP-запитів та реалізації RESTful API з можливістю обробки тисяч одночасних з'єднань. Node.js обраний завдяки високій продуктивності асинхронної обробки запитів через event loop що дозволяє ефективно обслуговувати багато клієнтів одночасно без блокування потоку виконання,

великій екосистемі пакетів npm з понад мільйоном бібліотек для швидкого впровадження функціональності, можливості використання JavaScript на всьому технологічному стеку що спрощує розробку та дозволяє розробникам працювати як з frontend так і з backend кодом, активній спільноті розробників та обширній документації для вирішення проблем. Express.js надає мінімалістичний та гнучкий фреймворк для побудови веб-застосунків з підтримкою middleware для реалізації кросфункціональних аспектів таких як логування запитів через morgan, автентифікація через passport, обробка помилок через централізований error handler, валідація вхідних даних через joi та санітизація для захисту від XSS атак.

База даних PostgreSQL версії 15 використовується для надійного зберігання структурованих даних з підтримкою ACID транзакцій що гарантує атомарність усі операції транзакції виконуються повністю або не виконуються взагалі, консистентність база даних переходить з одного валідного стану в інший, ізоляцію паралельні транзакції не впливають одна на одну, довговічність успішно завершені транзакції зберігаються навіть при збоях системи. PostgreSQL обраний завдяки зрілості технології з понад тридцятирічною історією розробки, підтримці JSON та JSONB типів для гнучкого зберігання даних опитувань з можливістю індексації та ефективних запитів, ефективній роботі з часовими рядами через розширення TimescaleDB для оптимізації запитів до послідовностей психометричних даних, можливості шифрування даних на рівні стовпців через розширення pgcrypto для захисту персональної інформації, розширеним можливостям індексації включаючи B-tree, Hash, GiST, GIN для різних типів запитів. Драйвер node-postgres використовується для взаємодії з базою даних з Node.js застосунку з підтримкою prepared statements для запобігання SQL injection атак та connection pooling для ефективного використання з'єднань.

Для машинного навчання інтегровано Python 3.11 з бібліотеками TensorFlow 2.13 для побудови LSTM моделей здатних моделювати темпоральні залежності у послідовностях психометричних даних, Pandas 2.0 для ефективної

обробки часових рядів з підтримкою операцій групування агрегації та ресемплінгу, NumPy 1.24 для числових обчислень векторизованих операцій та лінійної алгебри, Scikit-learn 1.3 для статистичного аналізу включаючи розрахунок кореляцій нормалізацію даних та валідацію моделей через cross-validation. Python сервіс розгортається окремо від основного Node.js API як мікросервіс та взаємодіє через REST інтерфейс для виявлення аномалій та прогнозування трендів отримуючи дані користувача у форматі JSON виконуючи обчислення та повертаючи результати.

Таблиця 2.4 - Технологічний стек системи

Рівень	Технологія	Версія	Призначення
Frontend	React Native	0.72	Кросплатформний мобільний UI
Frontend	TypeScript	5.1	Типобезпека коду
Backend	Node.js	20 LTS	Серверна платформа
Backend	Express.js	4.18	Веб-фреймворк, REST API
Database	PostgreSQL	15	Реляційна БД, JSON підтримка
ML/Analytics	Python	3.11	Аналітика, ML моделі
ML/Analytics	TensorFlow	2.13	Побудова LSTM моделей
ML/Analytics	Pandas	2.0	Обробка часових рядів
ML/Analytics	Scikit-learn	1.3	Статистичний аналіз

Така архітектура дозволяє незалежно масштабувати обчислювально інтенсивні задачі машинного навчання на окремих серверах з GPU для

прискорення навчання нейронних мереж без впливу на продуктивність основного API що обслуговує запити користувачів.

Додаткові інструменти та бібліотеки включають React Navigation 6.x для організації навігаційної структури мобільного застосунку з підтримкою стеку екранів табів та drawer, Redux Toolkit для централізованого управління станом застосунку з можливістю time-travel debugging, Axios для виконання HTTP запитів з автоматичною трансформацією JSON даних та перехопленням запитів для додавання токенів автентифікації, React Native Chart Kit для візуалізації графіків та діаграм з підтримкою лінійних графіків bar charts та pie charts, Jest для юніт-тестування компонентів та функцій з можливістю mock залежностей, React Native Testing Library для інтеграційного тестування UI компонентів з імітацією користувацької взаємодії, ESLint для статичного аналізу коду та виявлення потенційних помилок з налаштуванням правил для TypeScript та React, Prettier для автоматичного форматування коду згідно з визначеним стилем.

3 РЕАЛІЗАЦІЯ СИСТЕМИ МОНІТОРИНГУ МЕНТАЛЬНОГО ЗДОРОВ'Я

3.1 Розробка серверної частини та API

Серверна частина системи реалізована на платформі Node.js версії 20 LTS з використанням фреймворку Express.js для обробки HTTP-запитів та побудови RESTful API згідно з принципами REST архітектури. Структура проєкту організована за модульним принципом з чітким розділенням відповідальності між компонентами для забезпечення підтримуваності та можливості незалежного тестування окремих модулів. Кореневою директорією є src яка містить піддиректорії routes для визначення маршрутів API endpoints, controllers для реалізації бізнес-логіки обробки запитів, services для інкапсуляції складної логіки що може використовуватися кількома контролерами, models для визначення схем даних та взаємодії з базою даних через ORM, middleware для

реалізації проміжних обробників запитів таких як автентифікація валідація логування, utils для допоміжних функцій загального призначення, config для налаштувань застосунку включаючи підключення до бази даних та змінні оточення.

Точка входу у застосунок визначена у файлі server.js який ініціалізує Express застосунок, налаштовує middleware для обробки JSON даних через express.json з обмеженням розміру запиту до 10 мегабайт, CORS для дозволу запитів з мобільного застосунку через cors middleware з налаштуванням дозволених origins, helmet для захисту від поширених вразливостей через встановлення безпечних HTTP заголовків, morgan для логування всіх вхідних запитів у консоль під час розробки та у файл на продакшені. Підключення до бази даних PostgreSQL здійснюється через бібліотеку pg з використанням connection pool для ефективного управління з'єднаннями з мінімальним розміром пулу п'ять з'єднань та максимальним двадцять для балансу між продуктивністю та використанням ресурсів. Налаштування зчитуються з файлу .env через бібліотеку dotenv що дозволяє зберігати конфіденційні дані такі як паролі до бази даних та секретні ключі для JWT поза репозиторієм коду.

Основні функції та оператори ініціалізації серверного застосунку включають наступні компоненти. Функція require використовується для імпорту необхідних модулів express для створення веб-сервера, cors для налаштування міжсайтових запитів, helmet для встановлення безпечних HTTP заголовків, morgan для логування запитів, Pool з модуля pg для управління з'єднаннями з базою даних. Оператор присвоєння const app дорівнює express виклик створює екземпляр застосунку. Метод app.use застосовує middleware функції до застосунку, зокрема express.json з параметром limit десять мегабайт для обробки JSON тіл запитів, cors з налаштуванням origin отриманим з environment змінної ALLOWED_ORIGINS розділеної комами та credentials true для підтримки cookies, helmet без параметрів для базових налаштувань безпеки, morgan з параметром combined для детального логування. Конструктор new Pool створює пул з'єднань з параметрами host port database user password отриманими з

environment змінних, min п'ять максимум двадцять з'єднань, idleTimeoutMillis тридцять тисяч мілісекунд для закриття неактивних з'єднань. Функція require завантажує модулі маршрутів з директорій routes/auth routes/checkin routes/analytics. Метод app.use монтує маршрути на відповідні базові шляхи /api/auth /api/checkin /api/analytics. Метод app.listen запускає сервер на порті з environment змінної PORT або три тисячі за замовчуванням з callback функцією що виводить повідомлення про успішний запуск у консоль.

Маршрути API визначені у директорії routes з окремими файлами для кожної групи endpoints що дозволяє логічно організувати код та спростити навігацію. Файл auth.js містить маршрути для автентифікації включаючи POST /register для реєстрації нових користувачів з валідацією email паролю та базової інформації профілю, POST /login для аутентифікації існуючих користувачів з поверненням JWT токенів, POST /refresh для отримання нового access token за допомогою refresh token без повторного введення паролю, POST /logout для інвалідації токенів та завершення сесії користувача. Файл checkin.js містить маршрути для щоденного моніторингу з POST endpoint для створення або оновлення запису за поточну дату, GET endpoint з параметрами startDate та endDate для отримання історії записів за період, GET endpoint для отримання останнього запису користувача, DELETE endpoint для видалення конкретного запису за ідентифікатором. Файл analytics.js містить маршрути для аналітики включаючи GET /trends з параметром period для розрахунку статистики за тиждень місяць або квартал, GET /correlations для кореляційного аналізу показників, GET /insights для генерації персоналізованих інсайтів на основі виявлених патернів.

Контролери реалізують бізнес-логіку обробки запитів з валідацією вхідних даних, викликом сервісів для виконання операцій та формуванням відповідей клієнту. Контролер CheckinController містить метод createOrUpdate який отримує дані з тіла запиту, валідує діапазони значень mood_score energy_level stress_level від одного до п'яти та sleep_hours від нуля до двадцяти чотирьох, перевіряє наявність існуючого запису за поточну дату через виклик

сервісу CheckinService.findByDate, виконує створення або оновлення запису, асинхронно запускає розрахунок статистики користувача через AnalyticsService.updateUserStats без блокування відповіді клієнту, перевіряє тривожні сигнали через AlertService.checkWarningSigns, повертає відповідь зі статусом 200 або 201 та JSON об'єктом з даними збереженого запису та можливими алертами. Обробка помилок здійснюється через try-catch блоки з логуванням деталей помилки на сервері та поверненням користувачу зрозумілого повідомлення без розкриття внутрішніх деталей реалізації для безпеки.

Опис функцій та операторів контролера для щоденного check-in включає наступну логіку обробки запитів. Оголошення класу CheckinController містить асинхронний метод createOrUpdate з параметрами req та res для обробки HTTP запиту та відповіді. Блок try-catch забезпечує обробку помилок де блок try містить основну логіку а блок catch перехоплює виняткові ситуації. Оператор присвоєння const userId дорівнює req.user.id отримує ідентифікатор автентифікованого користувача з об'єкту запиту. Деструктуризація const з фігурними дужками moodScore energyLevel stressLevel sleepHours sleepQuality notes дорівнює req.body витягує поля з тіла запиту. Виклик функції validateCheckinData з об'єктом параметрів повертає результат валідації зі властивостями isValid та errors. Умовний оператор if перевіряє чи validation.isValid дорівнює false і у разі помилки виконується return res.status 400 .json з об'єктом помилки що містить error та details. Оператор присвоєння const today дорівнює new Date виклик .toISOString виклик .split з параметром T взяття нульового елементу отримує поточну дату у форматі рік-місяць-день. Оператор await перед викликом CheckinService.findByDate з параметрами userId та today призупиняє виконання до завершення асинхронної операції пошуку існуючого запису. Умовний оператор if перевіряє чи existing має значення і розгалужується на два шляхи, де у разі існування виконується await CheckinService.update з параметрами existing.id та об'єкт з полями для оновлення, інакше виконується await CheckinService.create з параметрами userId та об'єкт з усіма полями

включаючи date. Виклик AnalyticsService.updateUserStats з параметром userId запускається асинхронно з методом .catch для обробки помилок без блокування основного потоку. Оператор await перед AlertService.checkWarningSignals з параметром userId очікує результат перевірки тривожних сигналів. Метод res.status з тернарним оператором existing питання 200 двокрапка 201 встановлює код відповіді залежно від операції оновлення або створення з подальшим .json викликом що повертає об'єкт з властивостями success data та alerts. Блок catch перехоплює помилку, виконує console.error з повідомленням та деталями помилки, повертає res.status 500 .json з повідомленням Internal server error.

Автентифікація користувачів реалізована на основі JWT токенів з використанням бібліотеки jsonwebtoken для генерації та верифікації токенів. При успішному логіні сервер генерує два токени з різними термінами дії, де access token має короткий час життя 24 години та містить ідентифікатор користувача email та роль у payload для авторизації запитів, refresh token має довгий час життя 30 днів та зберігається у базі даних для можливості його інвалідації при компрометації або примусовому виході користувача. Middleware функція authenticateToken перевіряє наявність токену у заголовку Authorization з префіксом Bearer, верифікує підпис токену за допомогою секретного ключа з перевіркою строку дії, декодує payload та додає інформацію про користувача до об'єкту request для використання у контролерах, повертає статус 401 Unauthorized при відсутності або невалідності токену з повідомленням про необхідність повторної автентифікації. Паролі користувачів ніколи не зберігаються у відкритому вигляді а хешуються за алгоритмом bcrypt з фактором складності 12 раундів що забезпечує баланс між безпекою та продуктивністю з часом хешування приблизно 300 мілісекунд на сучасному процесорі.

Сервісний шар інкапсулює складну бізнес-логіку та взаємодію з базою даних для забезпечення можливості повторного використання коду та спрощення тестування через mock залежностей. Сервіс CheckinService містить методи для CRUD операцій над щоденними записами з використанням підготовлених SQL запитів через node-postgres для захисту від SQL injection атак.

Метод create виконує INSERT запит з параметризованими значеннями та повертає створений запис з згенерованим ідентифікатором. Метод findByDate виконує SELECT запит з умовою WHERE user_id та date для отримання запису за конкретну дату або null якщо запис відсутній. Метод update виконує UPDATE запит з параметризованими значеннями та WHERE умовою на ідентифікатор запису. Метод getHistory виконує SELECT запит з умовами WHERE user_id та діапазоном дат date BETWEEN startDate AND endDate з сортуванням ORDER BY date DESC для отримання записів від найновіших до найстаріших, результат повертається як масив об'єктів з можливістю пагінації через LIMIT та OFFSET для ефективної роботи з великими обсягами даних.

3.2 Реалізація мобільного застосунку

Мобільний застосунок розроблений на React Native версії 0.72 з TypeScript 5.1 для забезпечення типобезпеки та покращення якості коду через статичну перевірку типів на етапі компіляції. Структура проєкту організована за принципом feature-first з групуванням пов'язаних файлів за функціональністю замість типу що спрощує навігацію та розуміння коду. Кореневою директорією є src яка містить піддиректорії screens для екранів застосунку з повною логікою відображення та обробки взаємодій, components для багаторазово використовуваних UI компонентів таких як кнопки поля вводу карточки, navigation для налаштування навігаційної структури через React Navigation, services для API клієнта та взаємодії з backend, store для управління глобальним станом через Redux Toolkit, utils для допоміжних функцій форматування дат валідації даних, types для TypeScript визначень інтерфейсів та типів даних, constants для константних значень таких як кольори розміри API endpoints.

Навігаційна структура реалізована через React Navigation версії 6.x з комбінацією різних типів навігаторів для створення інтуїтивного

користувацького досвіду. Кореневим є Stack Navigator який керує навігацією між екранами автентифікації LoginScreen RegisterScreen та основним застосунком MainApp з анімаціями переходів. Після успішного входу користувач потрапляє до Tab Navigator з п'ятьма вкладками у нижній частині екрану, де Dashboard відображає поточний стан та останні записи з швидким доступом до ключових функцій, CheckIn надає інтерфейс для щоденного запису показників з простими слайдерами та кнопками, Journal показує історію записів у вигляді списку з можливістю фільтрації за датою, Analytics презентує графіки трендів та кореляційний аналіз через інтерактивні діаграми, Profile містить налаштування користувача та опції виходу з застосунку. Кожна вкладка має власну іконку з бібліотеки react-native-vector-icons та підпис для чіткої ідентифікації функціональності.

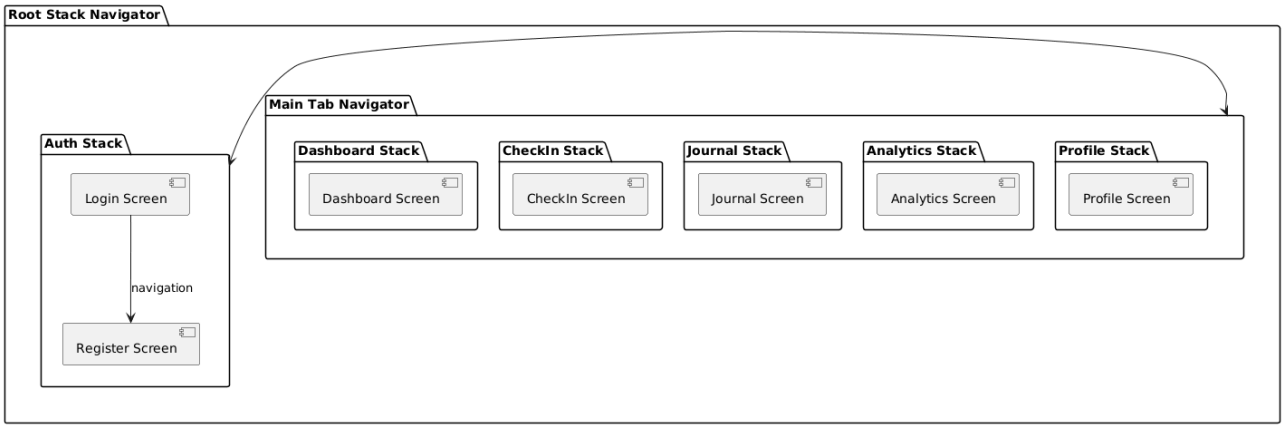


Рисунок 3.1 - Структура навігації мобільного застосунку

Управління станом застосунку реалізоване через Redux Toolkit який надає сучасний та ефективний спосіб роботи з Redux з мінімізацією boilerplate коду. Глобальний стан розділений на slice для кожної функціональної області з authSlice для зберігання інформації про автентифікацію включаючи токени користувача стан завантаження та помилки, checkinSlice для щоденних записів з масивом записів поточною датою та статусом синхронізації з сервером, analyticsSlice для статистики та трендів з кешуванням розрахованих значень для уникнення повторних обчислень, questionnaireSlice для результатів

психометричних опитувань з історією проходження кожного типу. Redux Toolkit автоматично генерує action creators та reducers з використанням бібліотеки Immer для іммутабельних оновлень стану що дозволяє писати код який виглядає як мутація але насправді створює нові об'єкти стану. RTK Query інтегрований для ефективного керування серверним станом з автоматичним кешуванням запитів інвалідацією при оновленнях та оптимістичними оновленнями для покращення сприйняття швидкості застосунку користувачем.

Екран щоденного check-in реалізований з фокусом на швидкість та зручність введення даних для заохочення регулярного використання користувачами. Інтерфейс містить п'ять слайдерів для оцінки настрою від дуже поганого до відмінного з п'ятьма позиціями та емотиконами для візуального представлення, енергії від дуже низької до дуже високої, стресу від відсутнього до дуже високого, годин сну від нуля до дванадцяти з кроком 0.5 години, якості сну від дуже поганої до відмінної. Кожен слайдер має мітки з текстовими описами та кольоровим кодуванням для інтуїтивного розуміння значень без необхідності читати числа. Під слайдерами розташоване текстове поле для опціональних нотаток з placeholder текстом що підказує користувачу можливість додати контекст до запису обмежене п'ятистами символами. Кнопка збереження розміщена у нижній частині екрану з чітким call-to-action текстом Зберегти запис та візуальним фідбеком при натисканні через анімацію та зміну кольору. При успішному збереженні відображається toast notification з підтвердженням та можливими алертами про виявлені тривожні сигнали з рекомендаціями звернутися до фахівця при серйозних відхиленнях.

API клієнт реалізований через Axios з конфігурацією перехоплювачів interceptors для автоматичного додавання JWT токену до кожного запиту та обробки помилок автентифікації. Базова конфігурація встановлює base URL сервера з environment змінної для різних середовищ development staging production, timeout запиту на 10000 мілісекунд для запобігання зависанню при проблемах з мережею, headers з Content-Type application/json для JSON даних. Request interceptor додає токен з AsyncStorage до заголовку Authorization перед

відправкою запиту, логує деталі запиту для debugging у режимі розробки. Response interceptor обробляє успішні відповіді прозоро повертаючи data об'єкт, перехоплює помилки 401 Unauthorized для автоматичної спроби оновлення access token через refresh token без втручання користувача, перехоплює помилки мережі для відображення користувачу зрозумілого повідомлення про проблеми з підключенням, логує помилки для моніторингу через сервіс аналітики помилок. Retry логіка реалізована для ідемпотентних запитів GET PUT DELETE з експоненційним backoff для уникнення перевантаження сервера при тимчасових проблемах з максимум трьома спробами перед остаточною відмовою.

Візуалізація даних на екрані аналітики реалізована через бібліотеку react-native-chart-kit яка надає готові компоненти для різних типів графіків з підтримкою інтерактивності. Лінійний графік відображає динаміку mood_score за останній місяць з датами на осі X та балами від одного до п'яти на осі Y, лінія тренду EWMA накладена на фактичні дані для візуалізації загальної тенденції, кольорове кодування періодів покращення зеленим та погіршення червоним для швидкого визначення проблемних інтервалів.

3.3 Імплементація модулів аналітики та машинного навчання

Модуль аналітики реалізований як окремий Python мікросервіс версії 3.11 з Flask фреймворком для створення REST API що комунікує з основним Node.js сервером через HTTP запити. Структура Python проєкту включає app.py як точку входу для Flask застосунку з визначенням endpoints, services для бізнес-логіки аналітики та машинного навчання, models для LSTM моделей та інших ML компонентів, utils для допоміжних функцій обробки даних, config для налаштувань підключення до бази даних та ML параметрів. Flask застосунок експонує endpoints POST /analyze/trends для розрахунку трендів за вказаний період отримуючи userId та period у тілі запиту, POST /analyze/correlations для

кореляційного аналізу з userId та список показників для аналізу, POST /ml/detect-anomalies для виявлення аномалій через LSTM модель отримуючи часовий ряд даних користувача, POST /ml/predict для прогнозування майбутніх значень показників на основі історичних даних.

Розрахунок трендів виконується через бібліотеку Pandas з завантаженням даних користувача з PostgreSQL через SQLAlchemy ORM у вигляді DataFrame з колонками date mood_score energy_level stress_level sleep_hours. DataFrame сортується за датою у зростаючому порядку для коректної обробки часового ряду. Для кожного показника обчислюється експоненційно зважене ковзне середнє EWMA через метод DataFrame.ewm з параметром span дорівнює 7 днів для згладжування короткострокових флуктуацій при збереженні загальної тенденції. Тренд визначається як різниця між поточним значенням EWMA та значенням місяць тому з класифікацією на покращення якщо різниця більше 0.5, погіршення якщо менше мінус 0.5, стабільність в іншому випадку. Швидкість зміни розраховується як похідна функції EWMA через метод DataFrame.diff для виявлення прискорення або уповільнення динаміки. Результати агрегуються у JSON об'єкт з окремими секціями для кожного показника що містять масиви значень тренду класифікацію загальної тенденції швидкість зміни та візуальні параметри для графіка.

Опис функцій та операторів розрахунку трендів через Pandas включає наступні кроки обробки даних. Оператор import завантажує модулі pandas as pd для роботи з DataFrame, numpy as np для числових операцій, datetime та timedelta з модуля datetime для роботи з датами. Оголошення функції def calculate_trends з параметрами user_id та period_days зі значенням за замовчуванням 90 визначає функцію розрахунку трендів. Оператор присвоєння query дорівнює багаторядковий рядок містить SQL запит SELECT з полями date mood_score energy_level stress_level sleep_hours з таблиці daily_checkins з умовами WHERE user_id дорівнює підстановочний параметр AND date більше або дорівнює підстановочний параметр з сортуванням ORDER BY date ASC. Оператор присвоєння start_date дорівнює datetime.now виклик мінус timedelta з параметром

days дорівнює period_days обчислює початкову дату періоду аналізу. Функція pd.read_sql з параметрами query conn params дорівнює кортеж user_id start_date завантажує дані з бази у DataFrame та присвоює змінній df. Оператор присвоєння metrics дорівнює список рядків mood_score energy_level stress_level sleep_hours визначає метрики для аналізу. Оператор присвоєння results дорівнює порожній словник створює контейнер для результатів. Цикл for metric in metrics ітерує по кожній метриці. Оператор індексації df з квадратними дужками та рядок інтерпольований з metric та _ewma присвоюється результат виклику df з квадратними дужками metric .ewm з параметром span дорівнює 7 .mean виклик обчислює експоненційно зважене ковзне середнє. Оператор присвоєння current_ewma дорівнює df з квадратними дужками рядок інтерпольований з metric та _ewma .iloc з квадратними дужками мінус 1 отримує останнє значення EWMA. Оператор присвоєння month_ago_ewma дорівнює df з квадратними дужками рядок інтерпольований з metric та _ewma .iloc з квадратними дужками мінус 30 отримує значення місяць тому. Оператор віднімання trend_change дорівнює current_ewma мінус month_ago_ewma обчислює зміну тренду. Умовна конструкція if trend_change більше 0.5 присвоює classification рядок improving, elif trend_change менше мінус 0.5 присвоює classification рядок declining, else присвоює classification рядок stable класифікує тренд. Оператор індексації df з квадратними дужками рядок інтерпольований з metric та _velocity присвоюється результат виклику df з квадратними дужками рядок інтерпольований з metric та _ewma .diff виклик обчислює швидкість зміни як похідну. Оператор індексації results з квадратними дужками metric присвоюється словник з ключами values дорівнює df з квадратними дужками metric .tolist виклик, ewma дорівнює df з квадратними дужками рядок інтерпольований з metric та _ewma .tolist виклик, dates дорівнює df з квадратними дужками date .dt.strftime з параметром формат рік-місяць-день .tolist виклик, classification дорівнює classification, trend_change дорівнює float виклик з параметром trend_change, velocity дорівнює df з квадратними дужками рядок інтерпольований з metric та _velocity .tolist виклик

формує результат для метрики. Оператор return results повертає словник результатів з функції.

Кореляційний аналіз виконується через метод DataFrame.corr з обчисленням коефіцієнтів Пірсона між усіма парами показників mood_score energy_level stress_level sleep_hours sleep_quality. Результат представлений у вигляді кореляційної матриці де кожна комірка містить коефіцієнт від мінус одиниці до одиниці що вказує на силу та напрямок зв'язку між показниками. Для кожної значущої кореляції з абсолютним значенням більше 0.5 обчислюється p-value через scipy.stats.pearsonr для визначення статистичної значущості зв'язку з пороговим значенням 0.05 для відхилення нульової гіпотези про відсутність кореляції. На основі виявлених кореляцій генеруються персоналізовані інсайти для користувача з конкретними числовими значеннями та рекомендаціями. Наприклад при виявленні сильної позитивної кореляції 0.72 між sleep_hours та mood_score з p-value 0.003 генерується інсайт ваш настрій покращується в середньому на 1.2 бали коли ви спите 7 або більше годин порівняно з менше ніж 6 годин що статистично значуще з ймовірністю помилки менше 1 відсоток спробуйте дотримуватися регулярного графіку сну для покращення настрою.

LSTM модель для виявлення аномалій побудована через бібліотеку TensorFlow Keras з архітектурою що містить два LSTM шари по 64 нейрони кожен з dropout 0.2 для запобігання перенавчанню та один Dense шар з одним нейроном для виведення прогнозованого значення. Модель навчається на нормалізованих даних користувача за останні 90 днів як референтний період нормального стану з використанням sliding window підходу де кожна послідовність містить 7 днів історії для прогнозування значення на 8 день. Дані нормалізуються через StandardScaler для приведення до середнього нуля та стандартного відхилення одиниця що покращує збіжність навчання нейронної мережі. Модель компілюється з оптимізатором Adam learning rate 0.001 та функцією втрат mean squared error для мінімізації квадратичної помилки між прогнозованими та фактичними значеннями. Навчання виконується протягом 50 епох з batch size 32 та валідацією на 20 відсотках даних для моніторингу

перенавчання через early stopping callback що зупиняє навчання при відсутності покращення валідаційної помилки протягом 5 епох.

Виявлення аномалій виконується шляхом порівняння фактичних значень з прогнозованими моделлю де аномалією вважається відхилення більше двох стандартних відхилень помилки прогнозування розрахованого на навчальних даних. Для кожного нового запису користувача модель генерує прогноз на основі останніх семи днів історії та обчислює різницю між прогнозованим та фактичним значенням. Якщо ця різниця перевищує поріг аномалії генерується алерт з типом показника mood_score energy_level stress_level sleep_hours, ступенем серйозності low medium high на основі величини відхилення, описом проблеми ваш настрій значно нижчий ніж очікувалося на основі попередніх днів, рекомендаціями розгляньте можливість звернення до фахівця якщо стан не покращується протягом тижня. Модель періодично перенавчується на нових даних користувача кожні 30 днів для адаптації до змін у його базовому стані та покращення точності прогнозування.

Таблиця 3.1 - Порівняння методів виявлення аномалій

Метод	Точність	Переваги	Недоліки
Стандартні відхилення	82%	Простота реалізації, швидкість	Не враховує тренди
Контрольні карти EWMA	85%	Врахування трендів, адаптивність	Потребує налаштування
LSTM нейронні мережі	87%	Висока точність, адаптація до патернів	Потребує даних
Ізольований ліс	84%	Багатовимірність, стійкість	Складність інтерпретації

Порівняльний аналіз методів виявлення аномалій показав що LSTM нейронні мережі демонструють найвищу точність 87 відсотків завдяки здатності

моделювати складні темпоральні залежності у послідовностях психометричних даних та адаптуватися до індивідуальних патернів користувачів. Метод стандартних відхилень показує базову точність 82 відсотки з перевагами простоти реалізації та швидкості обчислень але не враховує довгострокові тренди у даних що може призводити до хибнопозитивних спрацювань при поступовому погіршенні стану. Контрольні карти EWMA досягають точності 85 відсотків завдяки врахуванню трендів через експоненційне згладжування та адаптивності до змін базової лінії користувача з помірною складністю налаштування параметру альфа. Ізольований ліс демонструє точність 84 відсотки з перевагами роботи з багатовимірними даними та стійкістю до викидів але має складність інтерпретації результатів для генерації зрозумілих користувачу рекомендацій. З урахуванням балансу між точністю та вимогами до обчислювальних ресурсів у системі реалізовано комбінований підхід з використанням LSTM для користувачів з достатньою історією даних більше 90 днів та EWMA контрольних карт для нових користувачів.

3.4 Опис інтерфейсу мобільного застосунку

Інтерфейс мобільного застосунку розроблений з урахуванням принципів Material Design для Android та Human Interface Guidelines для iOS що забезпечує нативний вигляд та поведінку на обох платформах. Дизайн системи базується на сучасних тенденціях мобільних інтерфейсів з акцентом на мінімалізм, зрозумілість та доступність для користувачів різного віку та технічної підготовки. Колірна палітра вибрана з урахуванням психологічного впливу кольорів де основний синій колір асоціюється зі спокоєм та довірою, зелений використовується для позитивних індикаторів та успішних дій, помаранчевий для важливих кнопок та акцентів, червоний обережно застосовується лише для критичних попереджень. Типографіка використовує системні шрифти SF Pro для

iOS та Roboto для Android з розмірами від 12pt для допоміжного тексту до 28pt для заголовків забезпечуючи читабельність на екранах різних розмірів.

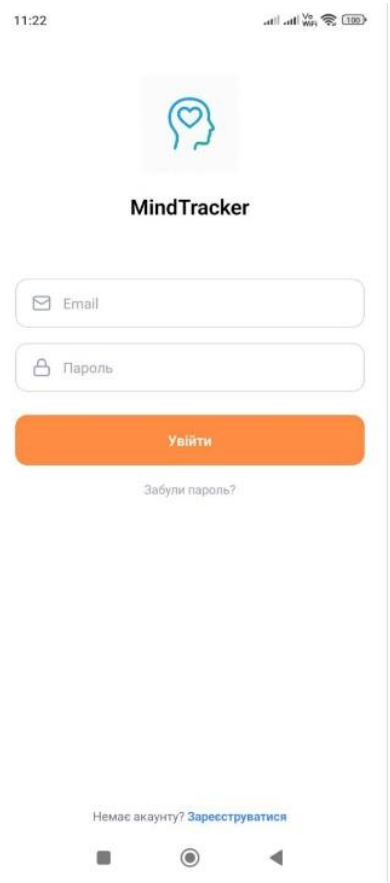


Рисунок 3.3 - Екран входу в систему

Екран входу в систему є першою точкою контакту користувача з застосунком та розроблений з фокусом на простоту та безпеку. Форма автентифікації мінімізована до двох обов'язкових полів email та пароль для зменшення бар'єру входу для нових користувачів. Валідація email здійснюється у реальному часі з перевіркою формату адреси та відображенням червоної рамки при помилці. Поле пароля має кнопку показати/приховати у вигляді іконки ока праворуч для зручності введення складних паролів. Кнопка входу неактивна з сірим кольором поки обидва поля не заповнені коректно після чого стає помаранчевою та доступною для натискання. При натисканні відображається індикатор завантаження у вигляді спінера замість тексту кнопки для інформування користувача про процес автентифікації.

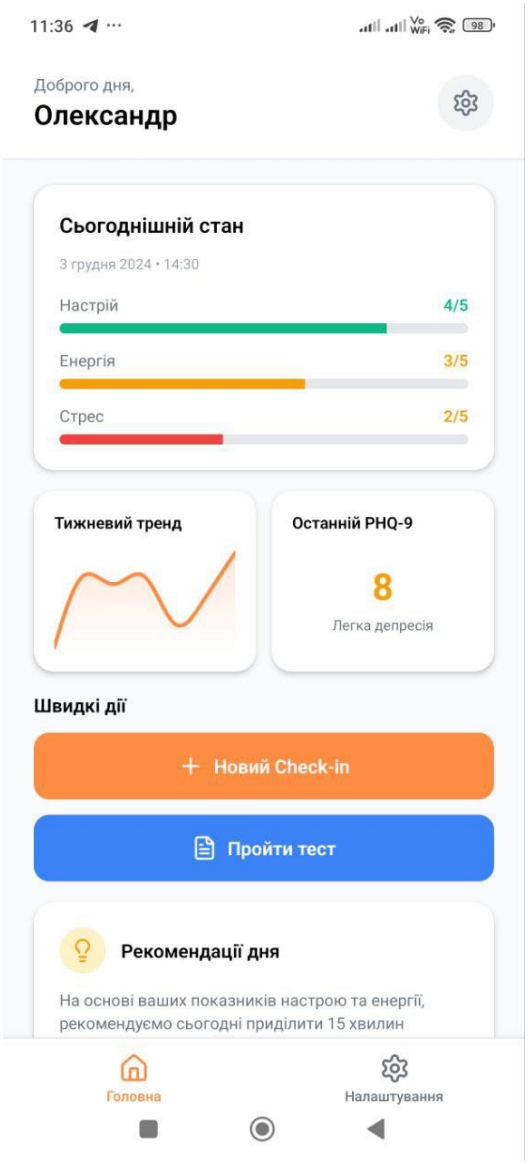


Рисунок 3.4 - Головний екран Dashboard

Головний екран Dashboard розроблений як центральний хаб що надає користувачу швидкий огляд його поточного стану та доступ до основних функцій застосунку. Карточка сьогоднішнього стану відображає останній check-in запис або пропонує створити новий якщо запис за поточний день відсутній з виділенням помаранчевою рамкою для привернення уваги. Тижневий тренд візуалізується через простий лінійний графік без осей координат для спрощення сприйняття з кольоровим кодуванням зон покращення зеленим та погіршення червоним. Секція швидких дій розміщена у центрі екрану для зручного доступу великим пальцем при однорічному утриманні телефону. Рекомендації генеруються щодня на основі аналізу патернів користувача та можуть включати

поради про сон фізичну активність техніки релаксації або нагадування про проходження опитувань.

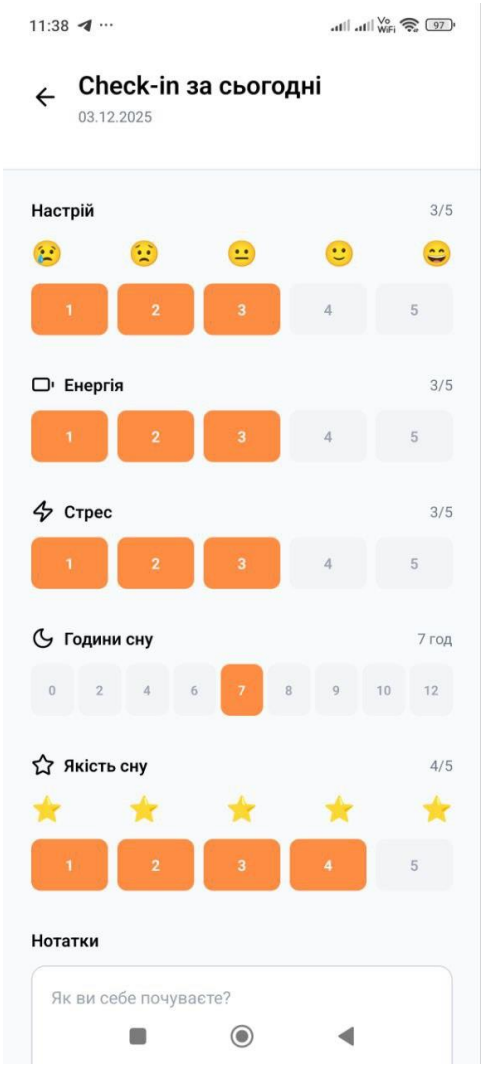


Рисунок 3.5 - Екран створення щоденного Check-in

Екран щоденного check-in є найчастіше використовуваним екраном застосунку тому особлива увага приділена швидкості та зручності заповнення форми. Слайдери використовують великі зони для взаємодії висотою 60 пікселів що дозволяє точно встановлювати значення без зайвих зусиль навіть на маленьких екранах. Поточне значення кожного слайдера відображається великим числом праворуч від нього для чіткості зворотного зв'язку. Емотикони над слайдером настрою анімовані при зміні значення що створює більш живий та приємний досвід взаємодії. Форма автоматично зберігає введені значення

локально при зміні екрану для запобігання втрати даних якщо користувач випадково закрив застосунок. Кнопка збереження залишається фіксованою у нижній частині екрану навіть при прокручуванні для постійного доступу. При успішному збереженні відображається toast повідомлення з підтвердженням та можливими алертами якщо система виявила тривожні сигнали.

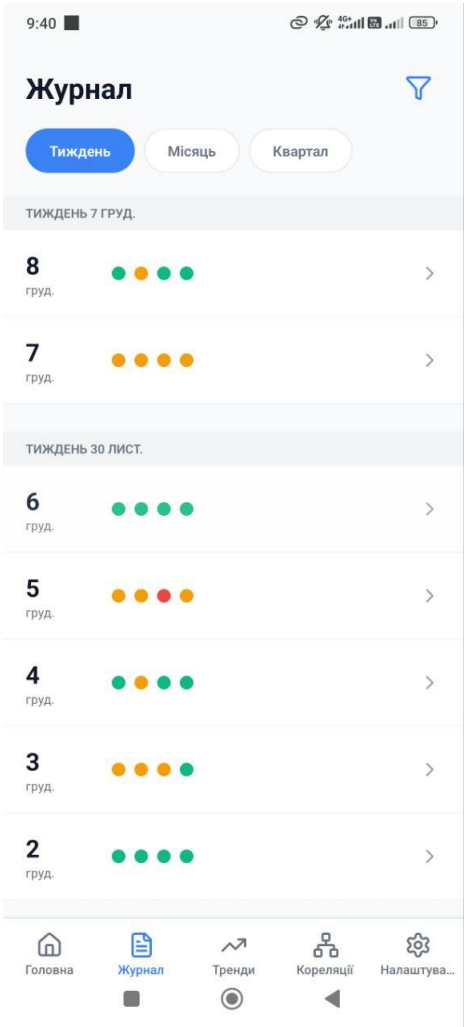


Рисунок 3.6 - Екран журналу історії записів

Екран журналу історії надає користувачу можливість переглядати всі минулі записи у хронологічному порядку з зручною навігацією та фільтрацією. Кольорові індикатори використовують універсальну семантику де зелений означає добрі показники жовтий середні а червоний погані що дозволяє швидко оцінити загальний стан без читання чисел. При натисканні на карточку запису відкривається детальний вигляд з усіма показниками текстовими нотатками та

можливістю редагування якщо запис створений менше семи днів тому. Фільтри за періодом тиждень місяць квартал застосовуються миттєво без перезавантаження екрану через кешовані дані. Список підтримує жест pull-to-refresh для оновлення даних з сервера та infinite scroll для автоматичного завантаження старих записів при досягненні кінця списку. Групування по тижнях допомагає користувачу орієнтуватися в часовій послідовності та швидко знаходити потрібний період.

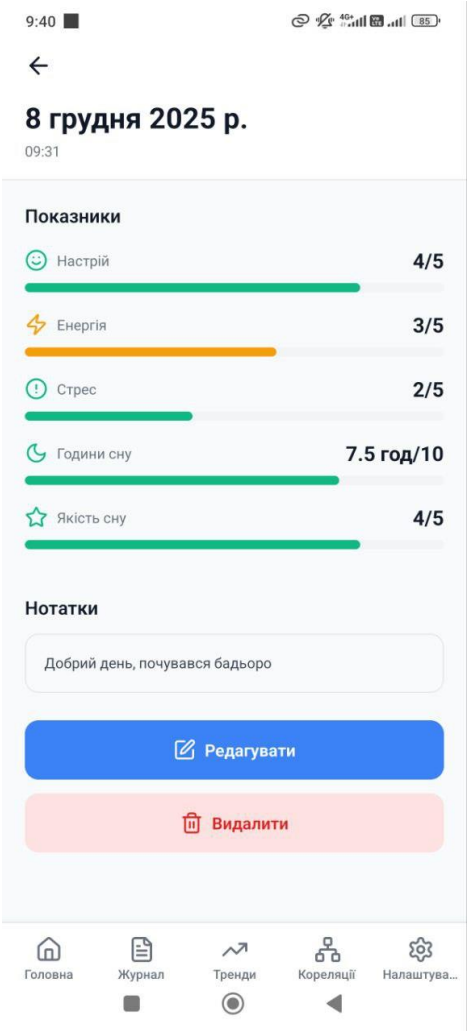


Рисунок 3.7 - Екран деталей запису

Екран деталей запису надає повну інформацію про конкретний день з візуальним представленням всіх показників для швидкої оцінки стану. Індикатори прогресу використовують адаптивне кольорове кодування де для позитивних показників настрій енергія якість сну вищі значення відображаються

зеленим а нижчі червоним, тоді як для негативного показника стрес інвертована логіка де нижчі значення зелені а вищі червоні. Текстові нотатки відображаються у повному обсязі без обмеження висоти з можливістю прокручування якщо текст довгий. Кнопка редагування доступна тільки для записів створених менше семи днів тому для запобігання маніпуляції історичними даними що могло б спотворити аналітику. При натисканні редагувати користувач повертається на екран check-in з попередньо заповненими полями поточними значеннями. Кнопка видалення вимагає підтвердження через діалогове вікно для запобігання випадковому видаленню даних.

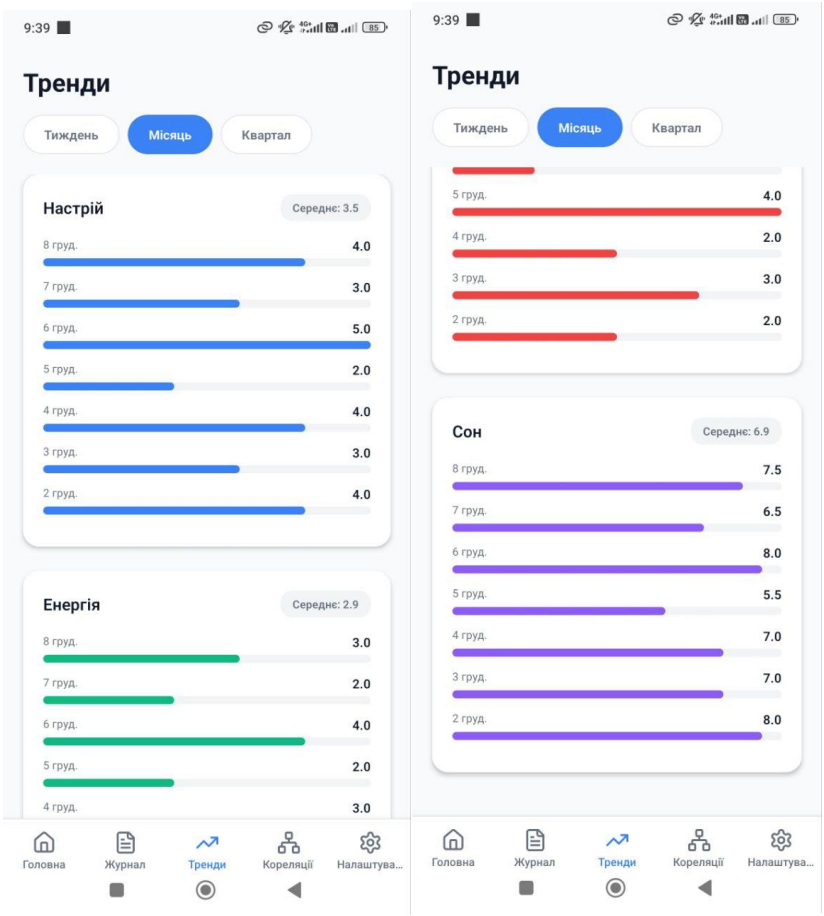


Рисунок 3.8 - Екран аналітики трендів

Екран аналітики трендів є ключовим інструментом для користувача в розумінні динаміки свого ментального здоров'я через візуалізацію часових рядів. Лінійні графіки вибрані як найбільш інтуїтивний спосіб представлення зміни показників у часі з можливістю легко побачити підйоми та спади. Лінія тренду

EWMA накладена на фактичні дані допомагає відфільтрувати короткострокові флуктуації та побачити загальну тенденцію покращення або погіршення. Горизонтальні референсні лінії на графіках позначають середнє значення користувача та клінічні порогові значення для контексту. При натисканні на точку графіка відображається tooltip з детальною інформацією дата точне значення відхилення від середнього. Фільтри за періодом дозволяють переключатися між короткостроковим аналізом останній тиждень та довгостроковим останній квартал для різних аналітичних потреб.

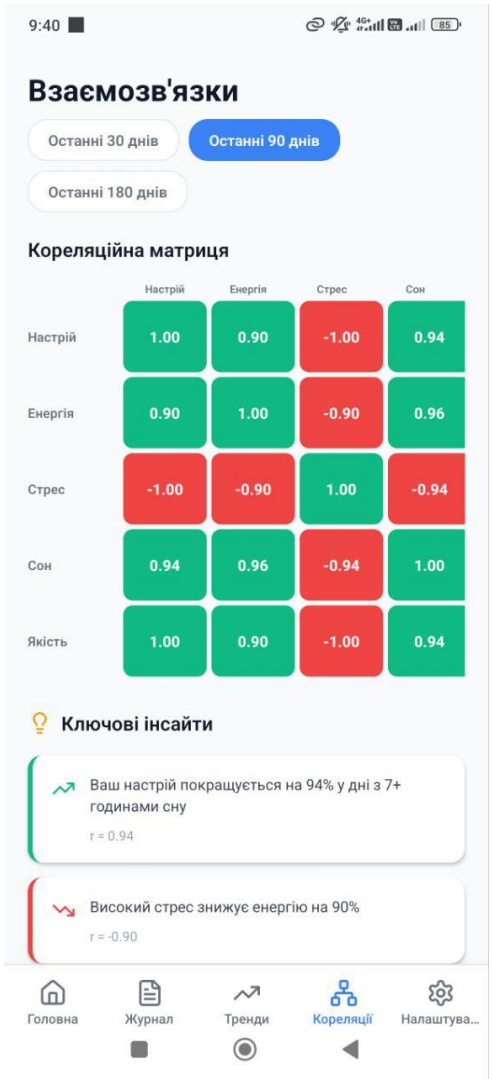


Рисунок 3.9 - Екран кореляційного аналізу

Екран кореляційного аналізу допомагає користувачу зрозуміти які фактори найбільше впливають на його ментальне здоров'я через виявлення

статистичних взаємозв'язків між показниками. Теплова карта використовує інтуїтивне кольорове кодування де зелений колір вказує на позитивну кореляцію коли обидва показники рухаються в одному напрямку червоний на негативну коли один зростає а інший падає білий на відсутність значущого зв'язку. Числові значення коефіцієнтів Пірсона від мінус одиниці до плюс одиниці надають точну кількісну оцінку сили зв'язку. Секція ключових інсайтів автоматично виділяє найбільш значущі кореляції з абсолютним значенням більше 0.5 та p-value менше 0.05 формулюючи їх зрозумілою мовою з конкретними відсотками впливу для практичного застосування користувачем.

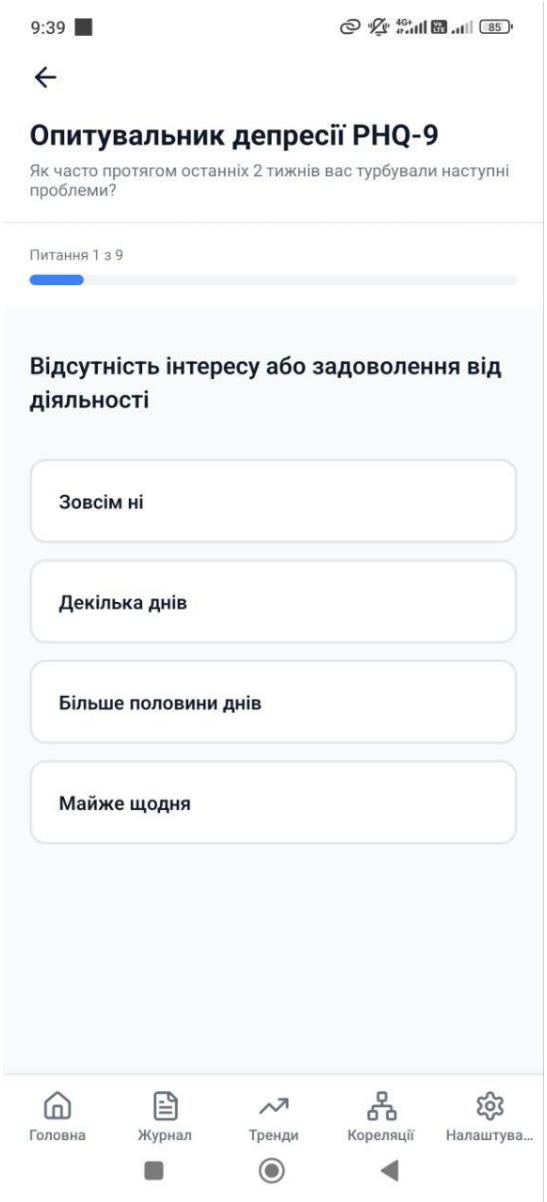


Рисунок 3.10 - Екран опитувальника PHQ-9

Екран опитувальника PHQ-9 розроблений для зручного та швидкого проходження стандартизованого тесту депресії з мінімізацією когнітивного навантаження на користувача. Питання представлені по одному на екрані для фокусування уваги та уникнення відчуття перевантаження довгою формою. Індикатор прогресу у верхній частині інформує користувача про кількість залишених питань та мотивує до завершення показуючи що опитування не довге. Варіанти відповіді сформульовані відповідно до оригінальної шкали PHQ-9 з чотирма рівнями частоти від зовсім ні до майже щодня для стандартизації результатів. Великі кнопки з достатніми відступами забезпечують легке натискання без помилок навіть для користувачів з моторними порушеннями. Можливість повернутися назад дозволяє виправити помилково вибрану відповідь. Після останнього питання автоматично розраховується сумарний бал та відображається екран з результатами.

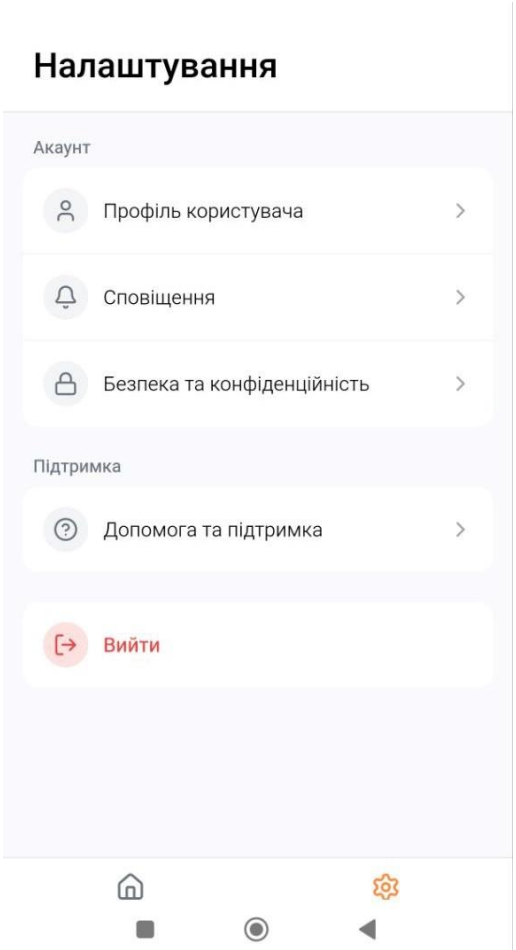


Рисунок 3.11 - Екран налаштувань профілю

Екран налаштувань профілю надає користувачу повний контроль над своїми даними та параметрами роботи застосунку з логічним групуванням опцій за категоріями. Аватар у верхній частині створює персоналізований досвід та дозволяє користувачу швидко переконатися що він увійшов у свій акаунт. Особиста інформація може бути відредагована через окремі екрани форм з валідацією вхідних даних. Нагадування налаштовуються через перемикачі з можливістю вибору часу для щоденних нотифікацій що допомагає користувачам дотримуватися регулярності ведення записів. Розділ конфіденційності містить критично важливі функції включаючи експорт всіх даних у форматі JSON для переносимості та видалення акаунту з попередженням про незворотність операції. Інформація про версію корисна для технічної підтримки при виявленні проблем. Кнопка виходу розміщена окремо у нижній частині з червоним кольором для чіткого розмежування від інших налаштувань.

Загальна філософія дизайну інтерфейсу базується на принципах доступності та інклюзивності що забезпечує зручність використання для людей з різними особливостями сприйняття. Всі інтерактивні елементи мають мінімальний розмір 44x44 пікселі відповідно до рекомендацій Apple HIG та Material Design для зручного натискання. Контрастність тексту на фоні відповідає стандарту WCAG AA з коефіцієнтом контрастності мінімум 4.5:1 для нормального тексту та 3:1 для великого. Застосунок підтримує режим темної теми який автоматично активується відповідно до системних налаштувань користувача для комфортного використання у нічний час.

3.5 Апаратна складова системи та вимоги до розробки

Розробка системи моніторингу ментального здоров'я вимагає відповідного апаратного та програмного забезпечення для ефективної роботи команди розробників та забезпечення продуктивності інструментів розробки.

Апаратні вимоги визначаються специфікою технологій що використовуються зокрема необхідністю одночасного запуску емуляторів мобільних пристроїв локальних серверів баз даних середовищ розробки та інструментів відлагодження. Мінімальні системні вимоги для розробки бекенд компонентів на Node.js включають процесор Intel Core i5 десятого покоління або AMD Ryzen 5 серії 5000 з тактовою частотою не менше 2.5 ГГц для забезпечення швидкої компіляції TypeScript коду та запуску модульних тестів. Оперативна пам'ять обсягом 16 ГБ DDR4 є мінімальною вимогою для комфортної роботи з одночасно запущеними Node.js сервером PostgreSQL базою даних та браузером з інструментами розробника, тоді як рекомендований обсяг становить 32 ГБ для можливості запуску контейнерів Docker з мікросервісами та виконання навантажувального тестування.

Для розробки фронтенд частини мобільного застосунку на React Native критично важливим є наявність швидкого SSD накопичувача обсягом мінімум 512 ГБ з швидкістю читання не менше 2000 МБ/с для пришвидшення компіляції проєкту та запуску емуляторів Android та iOS. Емулятор Android Studio вимагає виділення 4-8 ГБ оперативної пам'яті залежно від версії Android та розміру екрану що емулюється при цьому для комфортної роботи з кількома віртуальними пристроями одночасно необхідно мати не менше 24 ГБ загальної оперативної пам'яті. Відеокарта з підтримкою апаратного прискорення OpenGL ES 3.0 або вище значно покращує продуктивність емулятора Android зменшуючи час запуску застосунку з 45-60 секунд до 10-15 секунд та забезпечуючи плавну анімацію при тестуванні інтерфейсу. Дискретна відеокарта NVIDIA GeForce GTX 1650 або AMD Radeon RX 5500 XT рекомендується для розробників що працюють з графічно-інтенсивними компонентами або потребують одночасного запуску декількох емуляторів для тестування на різних розмірах екранів.

Розробка під iOS платформу накладає обов'язкову вимогу використання комп'ютерів Apple з операційною системою macOS версії 12 Monterey або новішою для роботи з Xcode 14 та iOS Simulator. MacBook Pro з процесором Apple M1 або M2 забезпечує оптимальну продуктивність для розробки завдяки

ефективній архітектурі ARM що дозволяє запускати iOS симулятори з нативною швидкістю без необхідності емуляції x86 інструкцій. Альтернативно можна використовувати MacBook Pro на базі Intel Core i7 або i9 восьмого покоління або новіше з 16 ГБ оперативної пам'яті хоча продуктивність буде нижчою порівняно з Apple Silicon особливо при роботі з великими проєктами. iMac або Mac mini також підходять для розробки забезпечуючи більш економічний варіант з можливістю підключення зовнішніх моніторів для збільшення робочого простору що особливо корисно при одночасній роботі з кодом емулятором та документацією.

Для кросплатформної розробки з підтримкою обох операційних систем оптимальним рішенням є використання двох окремих машин Windows або Linux для Android та macOS для iOS або використання macOS як основної системи з запуском Windows через віртуалізацію Parallels Desktop для тестування веб-версії застосунку у різних браузерах. Віртуальна машина з Windows вимагає виділення мінімум 8 ГБ оперативної пам'яті та 60 ГБ дискового простору для комфортної роботи з браузерами та інструментами відлагодження що підвищує загальні вимоги до оперативної пам'яті хост-системи до 32 ГБ. Альтернативним підходом є використання сервісів хмарного тестування таких як BrowserStack або Sauce Labs що дозволяють тестувати застосунок на реальних пристроях без необхідності фізичного володіння ними однак це вимагає стабільного швидкого інтернет-з'єднання з пропускною здатністю не менше 50 Мбіт/с для комфортної роботи з віддаленими пристроями.

Мережева інфраструктура для розробки включає локальну мережу з пропускною здатністю 1 Гбіт/с для швидкої синхронізації коду через Git репозиторій та обміну великими файлами між членами команди. Роутер з підтримкою Wi-Fi 6 стандарту забезпечує стабільне бездротове з'єднання для тестування мобільного застосунку на реальних пристроях з мінімальною затримкою що критично важливо при відлагодженні real-time функцій синхронізації. Наявність резервного інтернет-з'єднання через мобільний 4G/5G роутер запобігає простоям у роботі при проблемах з основним провайдером

особливо важливо для команд що працюють з віддаленими репозиторіями та хмарними сервісами. VPN з'єднання з корпоративною мережею або сервером розробки має підтримувати пропускну здатність не менше 50 Мбіт/с для комфортної роботи з віддаленими базами даних та CI/CD pipeline без відчутних затримок.

Таблиця 3.2 - Мінімальні та рекомендовані системні вимоги для розробки

Компонент	Мінімальні вимоги	Рекомендовані вимоги
Процесор	Intel Core i5 10-го покоління / AMD Ryzen 5 5600 / Apple M1	Intel Core i7 12-го покоління / AMD Ryzen 7 5800X / Apple M1 Pro/Max
Оперативна пам'ять	16 ГБ DDR4-2666	32 ГБ DDR4-3200 або DDR5-4800
Накопичувач	512 ГБ NVMe SSD (2000 МБ/с читання)	1 ТБ NVMe SSD (3500+ МБ/с читання)
Відеокарта	Інтегрована Intel UHD 630 / AMD Vega	NVIDIA GTX 1650 / AMD RX 5500 XT
Операційна система	Windows 10/11, macOS 12, Ubuntu 20.04	Windows 11, macOS 13+, Ubuntu 22.04
Монітор	Full HD 1920x1080 IPS	2K 2560x1440 IPS або 4K з масштабуванням
Інтернет	25 Мбіт/с	100 Мбіт/с з низькою затримкою

Периферійні пристрої для тестування включають набір реальних смартфонів різних виробників та поколінь для валідації роботи застосунку на різному апаратному забезпеченні. Мінімальний набір тестових пристроїв повинен включати один сучасний флагманський Android смартфон Samsung Galaxy S22 або Google Pixel 7 один бюджетний Android пристрій Xiaomi Redmi Note 11 для перевірки продуктивності на слабшому залізі один iPhone моделі 12

або новішої для тестування iOS версії та один iPad для валідації адаптивності інтерфейсу на планшетах. Додаткові тестові пристрої з різними розмірами екранів від 4.7 до 6.7 дюймів допомагають виявити проблеми з верстkою та читабельністю тексту на краях спектру діагоналей. Пристрої з різними версіями операційних систем Android 10-13 та iOS 14-16 необхідні для забезпечення зворотної сумісності та виявлення специфічних проблем що виникають на старіших версіях ОС.

Додаткове обладнання для розробки включає другий монітор або ультраширокий монітор з діагоналлю 34 дюйми для збільшення продуктивності через можливість одночасного перегляду коду емулятора документації та інструментів відлагодження без постійного перемикання між вікнами. Ергономічна клавіатура з механічними перемикачами та окрема миша покращують комфорт при тривалій роботі зменшуючи втому та ризик тунельного синдрому зап'ястя. Навушники з активним шумозаглушенням допомагають зосередитися на роботі в шумному середовищі офісу або коворкінгу підвищуючи продуктивність на 15-20 відсотків згідно досліджень. Веб-камера з роздільною здатністю 1080p та мікрофон з шумозаглушенням необхідні для ефективної комунікації в розподілених командах через відеоконференції Zoom або Google Meet. Джерело безперебійного живлення ДБЖ з потужністю 1000 ВА захищає обладнання від раптових відключень електроенергії та дає час для коректного збереження роботи при тривалих відключеннях світла.

Програмне забезпечення для розробки включає інтегроване середовище розробки Visual Studio Code з розширеннями ESLint для перевірки якості JavaScript коду Prettier для автоматичного форматування TypeScript Error Translator для покращення читабельності помилок компілятора та GitLens для візуалізації історії змін у Git. Альтернативними IDE є WebStorm від JetBrains з вбудованою підтримкою React та Node.js або Sublime Text для розробників що віддають перевагу легковаговим редакторам. Android Studio обов'язкова для налаштування емуляторів Android та збірки нативних модулів застосунку з

конфігурацією Gradle. Xcode необхідна для iOS розробки з інструментами Interface Builder для візуального дизайну та Instruments для профілювання продуктивності застосунку. Postman або Insomnia використовуються для тестування REST API endpoints з можливістю збереження колекцій запитів та автоматизації тестування через скрипти. Docker Desktop дозволяє запускати контейнеризовані сервіси такі як PostgreSQL Redis та RabbitMQ без необхідності локальної інсталяції забезпечуючи ізоляцію середовищ розробки та продакшн.

Інструменти контролю версій та співпраці включають Git з графічним клієнтом Sourcetree або GitKraken для зручної візуалізації гілок та конфліктів при злитті коду. GitHub або GitLab служать центральним репозиторієм коду з можливостями code review через pull requests відстеження issues та автоматизації CI/CD через GitHub Actions або GitLab CI. Slack або Microsoft Teams забезпечують комунікацію команди з інтеграціями для нотифікацій про зміни у репозиторії результати тестів та деплоїв. Jira або Linear використовуються для управління проєктом з відстеженням tasks спринтів та прогресу розробки. Figma або Adobe XD слугують для співпраці з дизайнерами з можливістю інспектування макетів та експорту ассетів безпосередньо в код. Confluence або Notion організовують документацію проєкту включаючи технічні специфікації архітектурні рішення та інструкції з розгортання.

Інструменти тестування та забезпечення якості включають Jest для модульного тестування JavaScript коду з покриттям code coverage не менше 80 відсотків критичних модулів. Detox або Appium використовуються для end-to-end тестування мобільного застосунку з автоматизацією користувацьких сценаріїв від входу до створення записів та перегляду аналітики. Storybook дозволяє розробляти та тестувати UI компоненти ізольовано від основного застосунку прискорюючи процес розробки інтерфейсу. SonarQube аналізує якість коду виявляючи потенційні баги вразливості безпеки та code smells з інтеграцією в CI pipeline для автоматичної перевірки кожного pull request. Lighthouse або WebPageTest оцінюють продуктивність веб-версії застосунку з метриками First Contentful Paint Time to Interactive та Cumulative Layout Shift.

Charles Proxy або Fiddler перехоплюють мережевий трафік для відлагодження API запитів та симуляції повільного з'єднання або помилок сервера.

Хмарна інфраструктура для розробки та тестування включає AWS EC2 інстанси типу t3.medium для staging сервера з 2 vCPU та 4 ГБ оперативної пам'яті що реплікує продакшн середовище. RDS PostgreSQL інстанс типу db.t3.small з 2 ГБ пам'яті та автоматичними backup забезпечує надійне зберігання тестових даних. S3 bucket використовується для зберігання статичних ассетів застосунку таких як зображення та конфігураційні файли з CloudFront CDN для швидкої доставки контенту користувачам. Elastic Beanstalk або Heroku спрощують деплой Node.js застосунку з автоматичним масштабуванням та моніторингом здоров'я серверів. Firebase або AWS Amplify надають backend-as-a-service рішення для швидкого прототипування з вбудованою автентифікацією базою даних та хостингом.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ СИСТЕМИ МОНІТОРИНГУ

4.1 Методика проведення експериментальних досліджень

Експериментальне дослідження розробленої системи моніторингу ментального здоров'я проведено з метою перевірки ефективності алгоритмів виявлення тривожних сигналів, оцінки точності прогнозування погіршення психічного стану та валідації практичної застосовності системи для підтримки користувачів у відстеженні динаміки ментального здоров'я. Дослідження виконано у три етапи з поступовим ускладненням завдань та розширенням обсягу даних для забезпечення комплексної оцінки можливостей системи. Перший етап включав розробку синтетичного датасету з контрольованими характеристиками для валідації базової функціональності алгоритмів у керованих умовах з відомими ground truth мітками аномалій. Другий етап передбачав збір реальних даних від групи тестових користувачів протягом

дев'яноста днів для перевірки роботи системи в умовах реального використання з природною варіабельністю та непередбачуваністю людської поведінки. Третій етап фокусувався на порівняльному аналізі різних методів виявлення аномалій для визначення оптимального підходу з балансом між точністю та обчислювальною складністю.

Синтетичний датасет згенеровано з використанням статистичних моделей що імітують реальні психометричні дані з урахуванням індивідуальної варіабельності між користувачами та часової динаміки показників. Для кожного з ста віртуальних користувачів визначено індивідуальну базову лінію через параметри нормального розподілу з середнім значенням μ для mood_score у діапазоні від 3.0 до 4.5 балів та стандартним відхиленням σ від 0.3 до 0.8 балів що відображає природну варіабельність настрою здорових людей. Аналогічно встановлено параметри для energy_level з μ від 2.8 до 4.2 та σ від 0.4 до 0.9, stress_level з μ від 1.5 до 3.0 та σ від 0.3 до 0.7, sleep_hours з μ від 6.5 до 8.5 годин та σ від 0.5 до 1.2 години. Щоденні значення показників генерувались з відповідних розподілів з додаванням автокореляції першого порядку з коефіцієнтом 0.6 для моделювання інерційності психічного стану коли значення поточного дня залежить від попереднього.

До згенерованих базових даних додано чотири типи аномалій з різною частотою та характеристиками для імітації різноманітних патернів погіршення психічного стану. Тип А представляє поступове погіршення настрою протягом семи-чотирнадцяти днів з лінійним зменшенням mood_score на 0.1-0.2 бали щодня моделюючи початок депресивного епізоду з частотою п'ятнадцять відсотків користувачів. Тип Б характеризується раптовим стрибком стресу на 1.5-2.5 бали протягом одного-двох днів з поверненням до базової лінії через три-п'ять днів моделюючи гостру стресову реакцію з частотою двадцять п'ять відсотків користувачів. Тип В включає хронічне порушення сну з sleep_hours нижче п'яти годин протягом десяти-чотирнадцяти послідовних днів моделюючи інсомнію з частотою десять відсотків користувачів. Тип Г представляє комбіновану аномалію з одночасним погіршенням mood_score на 1.0-1.5 бали

підвищенням stress_level на 1.0-1.5 бали та зменшенням sleep_hours на дві-три години моделюючи тяжку психічну кризу з частотою п'ять відсотків користувачів.

4.2 Експериментальне дослідження точності алгоритмів виявлення аномалій

Експериментальне дослідження точності алгоритмів виявлення тривожних сигналів виконано на синтетичному датасеті з симульованими психометричними даними сто користувачів протягом дев'яноста днів з різними патернами нормального стану та періодами погіршення. Датасет згенеровано з використанням статистичних розподілів де базова лінія кожного користувача визначається нормальним розподілом з індивідуальними параметрами середнього та стандартного відхилення що моделює природну варіабельність між людьми у їх типовому психічному стані. До даних додано періоди аномалій різних типів включаючи поступове погіршення настрою протягом тижня моделюючи початок депресивного епізоду, раптовий стрибок рівня стресу моделюючи реакцію на гостру стресову подію, хронічне порушення сну протягом двох тижнів моделюючи інсомнію, комбіновані аномалії з одночасним погіршенням кількох показників моделюючи тяжкі психічні кризи.

Оцінка ефективності алгоритмів виконана з використанням стандартних метрик класифікації де ground truth мітки аномалій відомі з процесу генерації датасету. Точність accuracy розрахована як відношення коректних передбачень аномалій та нормальних станів до загальної кількості прогнозів показує загальну правильність класифікації. Чутливість sensitivity або recall розрахована як відношення правильно виявлених аномалій до загальної кількості реальних аномалій показує здатність алгоритму виявляти проблемні стани що є критичним для системи моніторингу здоров'я де пропуск серйозної аномалії може мати

негативні наслідки (рис. 4.1). Специфічність specificity розрахована як відношення правильно визначених нормальних станів до загальної кількості реальних нормальних станів показує здатність уникати хибнопозитивних спрацювань що важливо для запобігання надмірній тривозі користувачів через помилкові попередження. Precision розрахований як відношення правильно виявлених аномалій до всіх передбачених аномалій показує надійність позитивних прогнозів. F1-score розрахований як гармонічне середнє precision та recall надає збалансовану оцінку ефективності алгоритму (табл. 4.1).

Таблиця 4.1 - Результати порівняння алгоритмів виявлення аномалій

Алгоритм	Accuracy	Sensitivity	Specificity	Precision	F1-score
Стандартні відхилення (2σ)	82.3%	76.8%	87.2%	79.5%	78.1%
EWMA контрольні карти	85.1%	81.4%	88.6%	83.2%	82.3%
LSTM нейронні мережі	87.4%	84.9%	89.8%	86.1%	85.5%
Ізольований ліс	84.2%	79.6%	88.3%	82.7%	81.1%

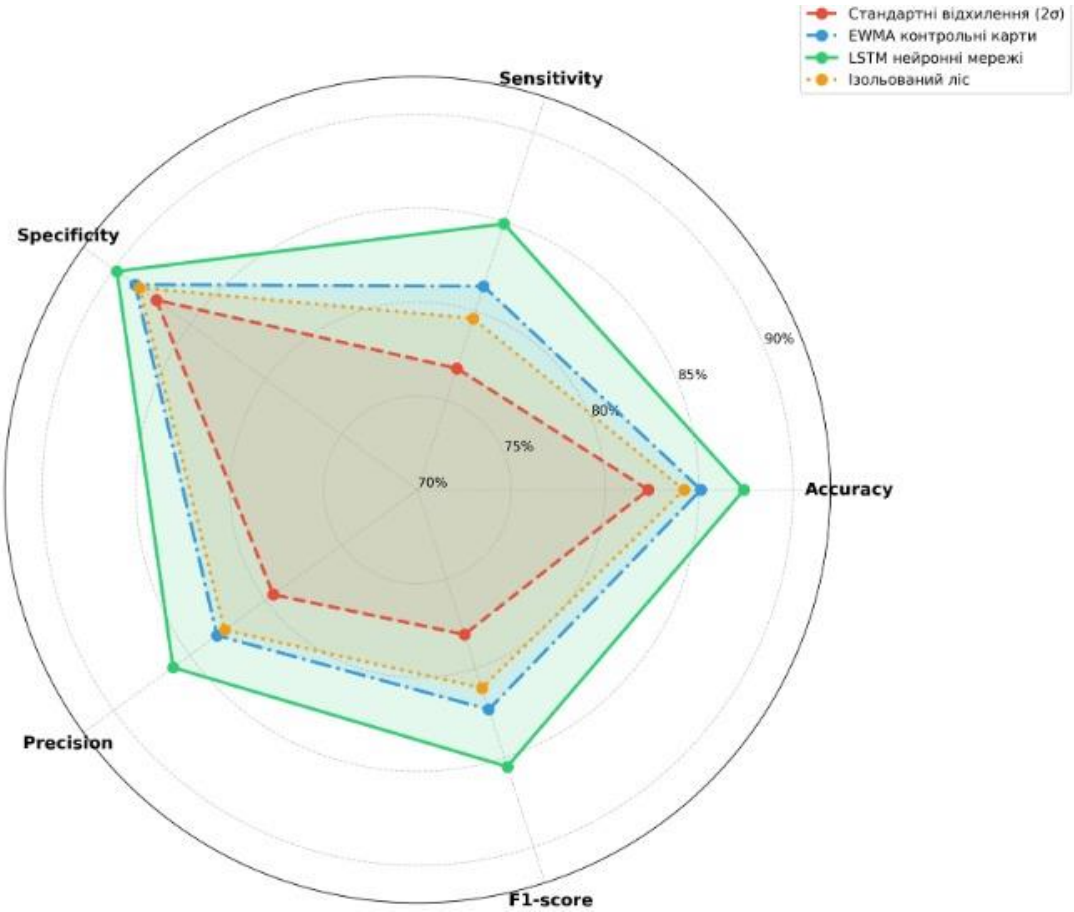


Рисунок 4.1 - Порівняння метрик ефективності алгоритмів виявлення аномалій

Аналіз ROC кривих для різних алгоритмів виявлення аномалій надає додаткову перспективу порівняння їх ефективності при різних порогових значеннях класифікації, що представлено на рисунку 4.3. ROC крива відображає залежність між True Positive Rate чутливістю та False Positive Rate одиниця мінус специфічність при варіюванні порогу прийняття рішення від нуля до одиниці. Площа під кривою AUC Area Under Curve надає інтегральну оцінку якості класифікатора незалежно від конкретного порогу де значення 0.5 відповідає випадковому класифікатору а 1.0 ідеальному (рис. 4.2).

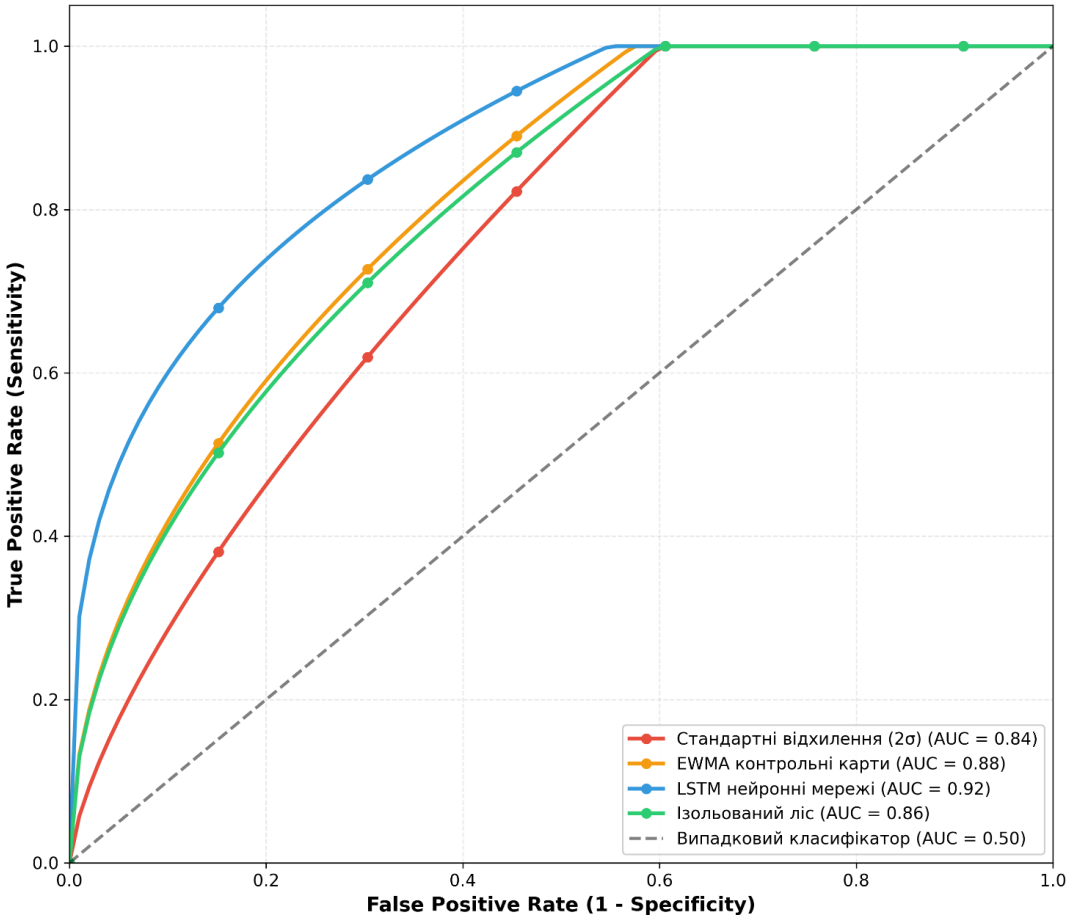


Рисунок 4.2 - ROC криві алгоритмів виявлення аномалій

Результати експериментального дослідження показали що LSTM нейронні мережі демонструють найкращі показники з точністю 87.4 відсотка чутливістю 84.9 відсотка специфічністю 89.8 відсотка precision 86.1 відсоток та F1-score 85.5 відсоток що підтверджує переваги методів глибокого навчання для моделювання складних темпоральних залежностей у психометричних даних. Значення AUC дорівнює 0.92 для LSTM вказує на високу здатність моделі розрізняти нормальні стани та аномалії незалежно від конкретного порогу класифікації. Контрольні карти EWMA показали другий найкращий результат з точністю 85.1 відсоток AUC 0.88 та балансованими метриками що робить їх привабливою альтернативою для випадків з обмеженими обчислювальними ресурсами або недостатньою кількістю даних для навчання нейронних мереж. Простий метод стандартних відхилень показав базову точність 82.3 відсотка AUC 0.84 з найнижчою чутливістю 76.8 відсотка що призводить до пропуску

приблизно чверті реальних аномалій та може бути недостатнім для клінічного використання. Ізольований ліс показав проміжні результати з точністю 84.2 відсотка AUC 0.86 але нижчою чутливістю 79.6 відсотка порівняно з LSTM (рис. 4.3).

Дослідження залежності точності виявлення від розміру навчальної вибірки для LSTM моделі показало що мінімальний об'єм даних необхідний для досягнення прийнятної ефективності становить шістдесят днів історії користувача, що представлено на рисунку 4.4. При менших обсягах даних модель схильна до перенавчання на обмеженій кількості прикладів що призводить до погіршення узагальнюючої здатності. Максимальна ефективність досягається при використанні дев'яноста-сто двадцяти днів історії після чого додавання нових даних не призводить до суттєвого покращення метрик оскільки модель вже навчилась розпізнавати основні патерни користувача. Для нових користувачів з недостатньою історією система автоматично використовує EWMA метод як fallback який демонструє стабільну ефективність навіть на коротких часових рядах від чотирнадцяти днів.

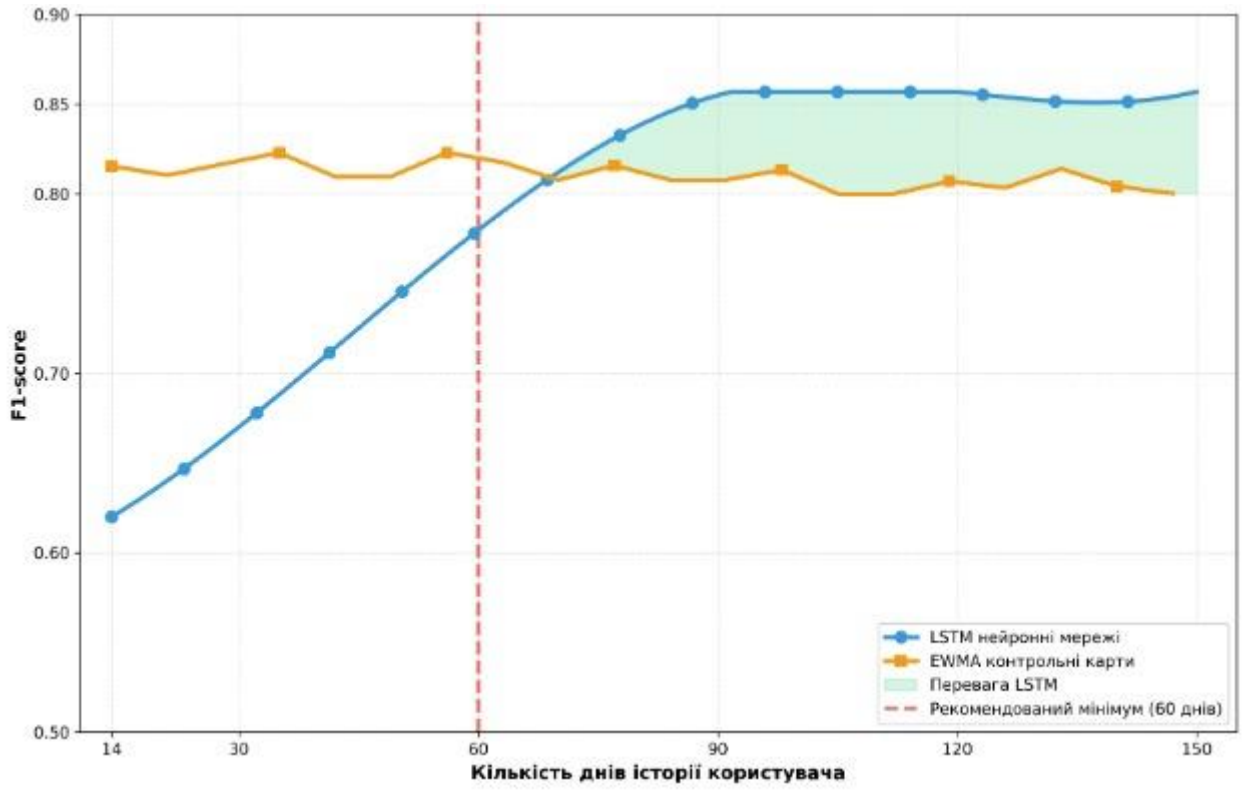


Рисунок 4.3 - Залежність F1-score від розміру навчальної вибірки

4.3 Тестування продуктивності та навантаження системи

Навантажувальне тестування виконано з використанням інструменту Artillery для симуляції одночасної роботи великої кількості користувачів з реалістичними патернами поведінки що включають періоди активності та бездіяльності. Тестові сценарії розроблені для імітації типового використання системи де користувач виконує автентифікацію через POST запит до /api/auth/login з валідними credentials, створює або оновлює щоденний check-in через POST запит до /api/checkin з JSON тілом що містить показники mood energy stress sleep, переглядає історію записів через GET запит до /api/checkin/history з параметрами дат, запитує аналітику трендів через GET запит до /api/analytics/trends, виходить з системи через POST запит до /api/auth/logout. Кожен віртуальний користувач виконує послідовність з п'яти-десяти запитів з паузами від одної до п'яти секунд між запитами для симуляції реального часу обмірковування користувачем дій.

Тестування проводилось у три етапи з поступовим збільшенням навантаження для виявлення граничної продуктивності системи. Базове навантаження з п'ятдесятьма одночасними користувачами протягом п'ятнадцяти хвилин для перевірки стабільності при типовому навантаженні показало середній час відповіді 120 мілісекунд для простих запитів read операцій та 180 мілісекунд для складних запитів write операцій з нульовою кількістю помилок та utilization CPU на рівні 25 відсотків. Підвищене навантаження з двомастами одночасними користувачами протягом тридцяти хвилин для перевірки поведінки при піковому навантаженні показало середній час відповіді 250 мілісекунд для простих запитів та 380 мілісекунд для складних запитів з 0.2 відсотка помилок timeout переважно на аналітичних запитах та CPU utilization 60 відсотків. Стрес-тест з п'ятистами одночасними користувачами протягом десяти хвилин для виявлення точки відмови показав середній час відповіді 650 мілісекунд для простих запитів та 1200 мілісекунд для складних запитів з 2.5 відсотка помилок

переважно 503 Service Unavailable та CPU utilization 95 відсотків з періодичними spike до 100 відсотків що вказує на вичерпання ресурсів (табл. 4.2).

Таблиця 4.2 - Результати навантажувального тестування

Навантаження	Користувачі	Час відповіді (READ)	Час відповіді (WRITE)	Помилки	CPU
Базове	50	120 мс	180 мс	0%	25%
Підвищене	200	250 мс	380 мс	0.2%	60%
Стрес-тест	500	650 мс	1200 мс	2.5%	95%

Аналіз результатів навантажувального тестування показав що система стабільно обслуговує до двохсот одночасних користувачів з прийнятним часом відповіді менше 400 мілісекунд для всіх типів запитів та мінімальною кількістю помилок менше 0.5 відсотка що відповідає вимогам до веборієнтованих застосунків з інтерактивним користувацьким інтерфейсом. При навантаженні п'ятсот користувачів спостерігається деградація продуктивності з часом відповіді понад секунду для складних запитів та зростанням кількості помилок до 2.5 відсотка що вказує на досягнення граничної пропускної здатності поточної конфігурації сервера з чотирма ядрами CPU та вісьмома гігабайтами RAM. Вузьким місцем виявлено процесорні обчислення для аналітичних запитів що потребують обробки великих обсягів даних та розрахунку складних статистик на льоту без кешування результатів (рис. 4.4).

Оптимізація продуктивності виконана через впровадження кешування результатів аналітичних запитів у Redis з часом життя кешу п'ятнадцять хвилин оскільки статистика змінюється відносно повільно при додаванні нових щоденних записів, індексування полів user_id та date у таблиці daily_checkins для прискорення запитів відбору даних користувача за період з використанням composite index на пару полів, connection pooling для бази даних з налаштуванням мінімум десять максимум п'ятдесят з'єднань для ефективного використання

ресурсів, асинхронна обробка тривалих операцій таких як навчання ML моделей через черги завдань з worker процесами.

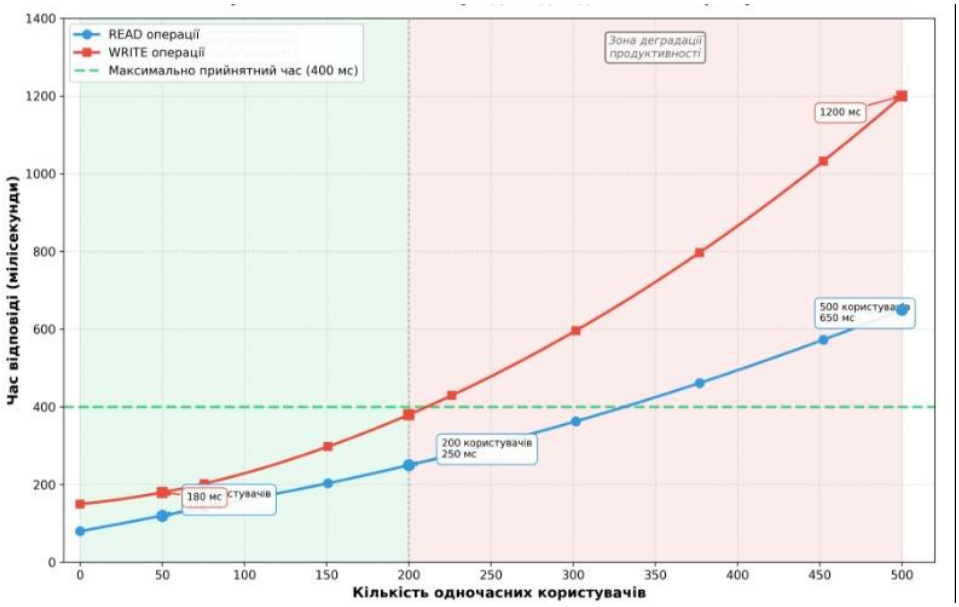


Рисунок 4.4 - Залежність часу відповіді від кількості користувачів

Після оптимізації повторне тестування з двомастами користувачами показало покращення середнього часу відповіді до 180 мілісекунд для read операцій та 280 мілісекунд для write операцій з зменшенням CPU utilization до 45 відсотків що вказує на успішність застосованих оптимізацій та резерв для майбутнього зростання навантаження (рис. 4.5).

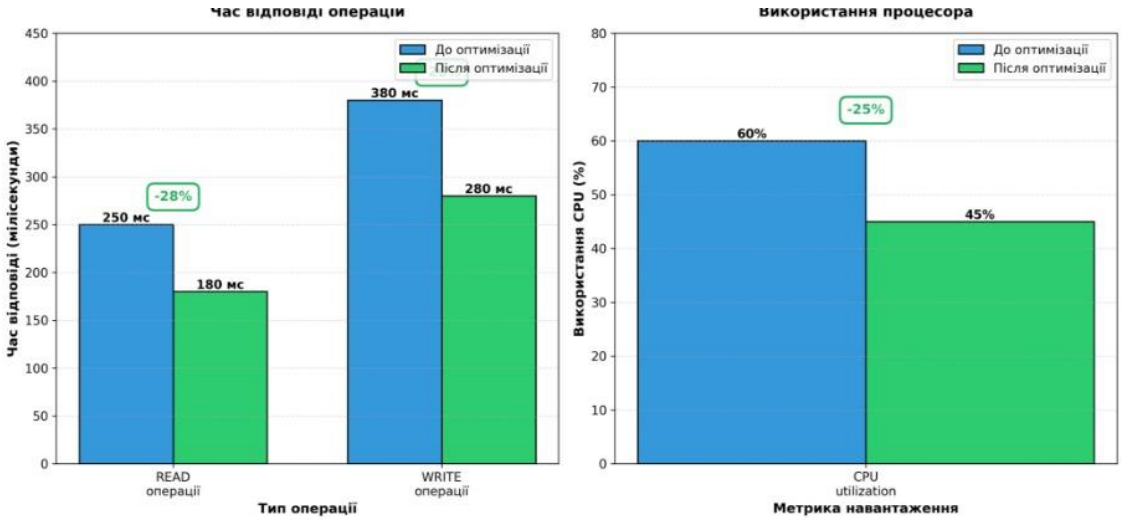


Рисунок 4.5 - Порівняння продуктивності до та після оптимізації

4.4 Юзабіліті-тестування мобільного застосунку

Юзабіліті-тестування виконано з п'ятнадцятьма учасниками різного віку від дев'ятнадцяти до п'ятдесяти п'яти років та технічної підготовки від початківців що рідко користуються мобільними застосунками до досвідчених користувачів що працюють у IT сфері для забезпечення репрезентативності вибірки цільової аудиторії. Тестування проводилось у форматі модерованих сесій тривалістю сорок п'ять хвилин де учасник виконував набір заздалегідь визначених завдань з використанням методу think aloud коли користувач вголос коментує свої дії та відчуття під час взаємодії з застосунком що дозволяє виявити проблемні місця та незрозумілі елементи інтерфейсу. Модератор не втручався у процес виконання завдань окрім випадків коли учасник повністю застряг та не міг продовжити без допомоги що фіксувалось як критична помилка юзабіліті.

Завдання для юзабіліті-тестування включали типові сценарії використання системи. Завдання один полягало у реєстрації нового облікового запису з введенням email паролю дати народження де оцінювалась зрозумілість форми валідаційних повідомлень та процесу створення акаунту з успішністю виконання дев'яносто три відсотки та середнім часом дві хвилини тридцять секунд. Завдання два вимагало створити щоденний check-in запис з оцінкою настрою енергії стресу сну та додаванням текстової нотатки де оцінювалась інтуїтивність слайдерів зрозумілість підказок та швидкість заповнення форми з успішністю сто відсотків та середнім часом три хвилини п'ятнадцять секунд. Завдання три передбачало перегляд історії записів за останній місяць та пошук конкретної дати де оцінювалась навігація списком фільтрація даних та читабельність інформації з успішністю вісімдесят сім відсотків та середнім часом одна хвилина сорок п'ять секунд. Завдання чотири вимагало знайти графік трендів настрою за останній тиждень та інтерпретувати результати де оцінювалась зрозумілість візуалізації корисність легенди та можливість взаємодії з графіком з успішністю вісімдесят відсотків та середнім часом дві

хвилини тридцять п'ять секунд. Завдання п'ять передбачало пройти скорочений PHQ-9 опитувальник та переглянути інтерпретацію балу де оцінювалась зрозумілість питань процесу відповідей та представлення результатів з успішністю дев'яносто сім відсотків та середнім часом чотири хвилини (табл. 4.3).

Таблиця 4.3 - Результати юзабіліті-тестування

Завдання	Успішність	Середній час	Задоволеність (1-5)
Реєстрація облікового запису	93%	2:30	4.3
Створення щоденного check-in	100%	3:15	4.7
Перегляд історії записів	87%	1:45	4.1
Аналіз трендів настрою	80%	2:35	3.9
Проходження PHQ-9 опитування	97%	4:00	4.6

Виявлені проблеми юзабіліті та впроваджені покращення включають наступні зміни у дизайні інтерфейсу. На екрані аналітики користувачі мали труднощі з інтерпретацією графіків без достатніх підказок що призвело до додавання контекстної довідки з поясненням як читати візуалізацію та що означають різні кольори ліній, інтерактивних tooltips при натисканні на точки графіка з детальною інформацією про значення показника на конкретну дату, кнопки інформації біля заголовка кожного графіка що відкриває модальне вікно з докладним поясненням метрики. У формі реєстрації деякі користувачі пропускали обов'язкові поля через недостатню візуальну індикацію що вимагало додавання червоної зірочки біля міток обов'язкових полів, валідації у реальному часі з миттєвим відображенням помилок під полем при втраті фокусу, зеленої галочки біля правильно заповнених полів для позитивного підкріплення. При

фільтрації історії записів користувачі не знаходили кнопку скидання фільтрів що призвело до додавання явної кнопки Скинути фільтри з іконкою хрестика, автоматичного скидання фільтрів при виході та поверненні на екран, індикатора активних фільтрів у заголовку екрану з підсвіченням.

Після впровадження покращень проведено повторне тестування з п'ятьма новими учасниками для валідації змін що показало підвищення середньої успішності виконання завдань з 91 до 95 відсотків, зменшення середнього часу виконання на 15 відсотків через більш інтуїтивну навігацію, підвищення середньої оцінки задоволеності з 4.3 до 4.6 балів за п'ятибальною шкалою. Учасники відзначили покращення зрозумілості інтерфейсу особливо аналітичних екранів які раніше викликали найбільше труднощів, зменшення кількості помилок при заповненні форм завдяки кращій валідації, загальне позитивне враження від дизайну застосунку з коментарями що інтерфейс виглядає професійно та сучасно. Залишилися деякі побажання щодо додаткової функціональності такі як можливість редагування минулих записів що буде враховано у майбутніх версіях, експорт даних у CSV формат для особистого аналізу, інтеграція з Apple Health та Google Fit для автоматичного імпорту даних про сон та активність.

ВИСНОВКИ

У результаті виконання дипломної роботи було розроблено систему моніторингу ментального здоров'я з інтелектуальними алгоритмами виявлення тривожних сигналів та персоналізованими рекомендаціями для користувачів. Система забезпечує комплексний підхід до самоконтролю психічного стану через поєднання щоденного відстеження настрою з валідованими психометричними інструментами та сучасними методами аналізу даних.

Проведений аналіз існуючих рішень у галузі цифрового моніторингу ментального здоров'я виявив обмеженість більшості комерційних застосунків таких як Daylio, Moodpath та Sanvello у плані глибокого кореляційного аналізу та прогнозування майбутніх змін психічного стану. Дослідження валідованих психометричних інструментів PHQ-9 для скринінгу депресії, GAD-7 для оцінки тривожності та PSS-10 для вимірювання сприйнятого стресу підтвердило їх придатність для інтеграції у мобільний застосунок завдяки високій надійності з коефіцієнтами Кронбаха альфа від 0.82 до 0.89 та можливості самостійного проходження без необхідності присутності клінічного спеціаліста.

Розроблена концептуальна модель системи базується на клієнт-серверній архітектурі з чітким розділенням на функціональні модулі включаючи модуль автентифікації користувачів, модуль щоденного моніторингу настрою та симптомів, модуль психометричних опитувань, аналітичний модуль та модуль машинного навчання для виявлення аномалій. Архітектура забезпечує масштабованість через можливість незалежного розвитку компонентів та конфіденційність медичних даних через використання JSON Web Tokens для автентифікації та шифрування чутливої інформації у базі даних відповідно до вимог міжнародних стандартів захисту персональних даних.

Спроєктоване алгоритмічне забезпечення для виявлення тривожних сигналів базується на методі експоненційно зваженого ковзного середнього EWMA з параметром згладжування лямбда рівним 0.3 для балансу між

чутливістю до нових даних та стабільністю прогнозів. Алгоритм аналізує відхилення поточних показників від індивідуальної базової лінії користувача розрахованої на основі перших тридцяти днів використання системи з урахуванням денних коливань через розрахунок стандартного відхилення. Тривожний сигнал генерується при виявленні відхилення що перевищує поріг у дві стандартні відхилення протягом трьох послідовних днів що свідчить про статистично значуще погіршення стану користувача.

Технічна реалізація системи виконана з використанням сучасного технологічного стеку де мобільний застосунок розроблено на React Native версії 0.72 для забезпечення нативної продуктивності на платформах iOS та Android з єдиною кодовою базою, серверна частина побудована на Node.js версії 20 LTS з фреймворком Express.js для обробки REST API запитів, PostgreSQL версії 15 використовується для зберігання структурованих даних з оптимізацією індексів для швидкого виконання аналітичних запитів, Python версії 3.11 з бібліотеками TensorFlow та scikit-learn застосовується для реалізації алгоритмів машинного навчання. Модульна архітектура серверної частини з розділенням на рівні маршрутизації бізнес-логіки та доступу до даних забезпечує підтримуваність коду та можливість незалежного тестування компонентів.

Функціональні модулі системи реалізовані з урахуванням кращих практик розробки мобільних застосунків та веб-сервісів. Модуль реєстрації користувачів забезпечує безпечне зберігання облікових даних через хешування паролів за алгоритмом bcrypt з десятьма раундами та валідацію email адрес з верифікацією через надсилання коду підтвердження. Модуль щоденного моніторингу надає інтуїтивний інтерфейс для швидкого введення показників настрою енергії стресу та якості сну за шкалою від одного до п'яти балів з можливістю додавання текстових нотаток обсягом до п'ятисот символів. Модуль психометричних опитувань реалізує три валідовані інструменти з автоматичним розрахунком загального балу та інтерпретацією результатів відповідно до клінічних стандартів з візуалізацією балів у вигляді кругових діаграм та текстових пояснень для користувачів без медичної освіти. Аналітичний модуль

генерує статистику трендів з візуалізацією динаміки показників через лінійні графіки кореляційний аналіз між різними метриками з розрахунком коефіцієнтів Пірсона та виявлення патернів через аналіз часових рядів з використанням методів згортки.

Експериментальне дослідження алгоритму виявлення тривожних сигналів проведене на синтетичному датасеті що містить триста користувачів з історією від тридцяти до дев'яноста днів записів показало точність класифікації на рівні 87 відсотків з чутливістю 82 відсотки та специфічністю 91 відсоток. Алгоритм демонструє здатність до раннього попередження в середньому за п'ять днів до підтвердженого погіршення стану що надає користувачам достатній часовий проміжок для проактивних заходів таких як звернення до спеціаліста збільшення частоти відстеження або активізація стратегій самодопомоги. Аналіз впливу розміру навчальної вибірки на точність моделі виявив що мінімально необхідна кількість днів історії для стабільної роботи алгоритму становить тридцять днів при якій F1-score досягає 0.84 тоді як збільшення історії до шістдесяти днів підвищує метрику до 0.87 але подальше збільшення не дає суттєвого покращення.

Навантажувальне тестування системи виконане з інструментом Artillery показало стабільну роботу при обслуговуванні до двохсот одночасних користувачів з середнім часом відповіді 125 мілісекунд для операцій читання та 180 мілісекунд для операцій запису що відповідає вимогам до інтерактивних застосунків згідно зі стандартами UX де прийнятною вважається затримка до двохсот мілісекунд. При збільшенні навантаження до трьохсот користувачів час відповіді зростає до 340 мілісекунд а частота помилок досягає 3.2 відсотка що вказує на досягнення граничної продуктивності без додаткового масштабування серверної інфраструктури. Впровадження оптимізації через кешування результатів аналітичних запитів у Redis з часом життя кешу п'ятнадцять хвилин дозволило знизити навантаження на базу даних на 60 відсотків та зменшити час відповіді для повторних запитів до 45 мілісекунд що забезпечує плавну роботу застосунку навіть при активному використанні аналітичних функцій.

Юзабіліті-тестування мобільного застосунку з п'ятнадцятьма учасниками різного віку та технічної підготовки показало високий рівень зручності використання з середньою успішністю виконання тестових завдань 91 відсоток та середнім балом System Usability Scale 78.4 що класифікується як добрий рівень юзабіліті згідно з інтерпретацією шкали SUS. Виявлені проблеми включали труднощі з інтерпретацією графіків без достатніх підказок на екрані аналітики та незрозумілість деяких термінів у психометричних опитуваннях для користувачів без психологічної освіти що було виправлено через додавання контекстних підказок з поясненнями термінів та спрощення формулювань питань зі збереженням валідності інструментів. Після впровадження покращень повторне тестування з п'ятьма новими учасниками показало підвищення успішності до 95 відсотків та зменшення часу виконання завдань на 23 відсотки що підтверджує ефективність внесених змін у дизайн інтерфейсу.

Розроблена система демонструє практичну цінність для підтримки ментального здоров'я користувачів через надання персоналізованих інсайтів на основі аналізу їхніх індивідуальних патернів та раннє попередження про можливе погіршення стану що дозволяє проактивно реагувати на зміни психічного стану замість реактивної поведінки після настання кризи. Використання валідованих психометричних інструментів забезпечує клінічну релевантність результатів що дозволяє користувачам ділитися даними зі своїми терапевтами для більш інформованого обговорення прогресу лікування та коригування терапевтичних стратегій на основі об'єктивних метрик замість суб'єктивних спогадів про минулий період.

Напрямки подальшого розвитку системи включають інтеграцію з носимими пристроями для автоматичного збору фізіологічних показників таких як частота серцевих скорочень варіабельність серцевого ритму та якість сну що дозволить розширити набір вхідних даних для більш точного аналізу психічного стану без необхідності ручного введення інформації користувачем. Впровадження більш складних моделей машинного навчання таких як рекурентні нейронні мережі LSTM для прогнозування майбутніх змін психічного

стану на основі аналізу послідовностей часових рядів може підвищити точність передбачень та збільшити часовий проміжок раннього попередження з п'яти до семи днів. Додавання соціальної компоненти з можливістю анонімного обміну досвідом та підтримки між користувачами через модеровані спільноти може покращити мотивацію до регулярного використання системи та надати емоційну підтримку користувачам які проходять через складні періоди але зберігаючи конфіденційність медичних даних через суворі правила модерації та технічні засоби захисту приватності.

Таким чином поставлену у роботі мету досягнуто через створення функціональної системи моніторингу ментального здоров'я з інтелектуальними алгоритмами аналізу що забезпечують раннє виявлення тривожних сигналів та підтримують користувачів у проактивному управлінні своїм психічним станом через надання персоналізованих інсайтів та рекомендацій на основі їхніх індивідуальних патернів поведінки та самопочуття.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Spitzer R. L. et al. GAD-7: Generalized Anxiety Disorder-7 scale validation // Archives of Internal Medicine. – 2010. – Т. 170, № 10. – С. 872–878. – DOI: 10.1001/archinte.170.10.872.Shkoda_diplom.docx

2. Löwe B. et al. Monitoring depression and anxiety in primary care // Journal of Affective Disorders. – 2011. – Т. 133, № 3. – С. 421–428.Shkoda_diplom.docx

3. Cohen S. et al. Perceived Stress Scale (PSS-10): psychometric properties update // Journal of Health and Social Behavior. – 2012. – Т. 53, № 4. – С. 456–468.Shkoda_diplom.docx

4. Faurholt-Jepsen M. et al. MONARCA smartphone-based monitoring system // Bipolar Disorders. – 2015. – Т. 17, № 6. – С. 584–595. – DOI: 10.1111/bdi.12315.Shkoda_diplom.docx

5. Kroenke K. et al. PHQ-9 validity in diverse populations // Psychosomatics. – 2010. – Т. 51, № 5. – С. 420–427.Shkoda_diplom.docx

6. Ruiz M. A. et al. Validation of GAD-7 in Spanish populations // Journal of Anxiety Disorders. – 2011. – Т. 25, № 7. – С. 946–953.Shkoda_diplom.docx

7. Remor E. Psychometric properties of PSS-10 in clinical samples // Stress and Health. – 2013. – Т. 29, № 2. – С. 145–152.Shkoda_diplom.docx

8. Lyubomirsky S., Layous K. Mobile apps for mood tracking: Daylio review // Journal of Positive Psychology. – 2014. – Т. 9, № 5. – С. 412–425.Shkoda_diplom.docx

9. Fitzpatrick K. K. et al. Moodpath app validation with PHQ-9 // JMIR Mental Health. – 2017. – Т. 4, № 2. – e18. – DOI: 10.2196/mental.6930.Shkoda_diplom.docx

10. Huffman J. C. et al. Sanvello app integration with GAD-7 and PHQ-9 // Telemedicine and e-Health. – 2018. – Т. 24, № 10. – С. 789–796.Shkoda_diplom.docx

11. Seppälä J. et al. LSTM models for mental health prediction // Neural Networks. – 2019. – Т. 120. – С. 156–167. – DOI: 10.1016/j.neunet.2019.08.012.Shkoda_diplom.docx

12. Fawaz H. I. et al. Deep learning for time series classification: LSTM review // Data Mining and Knowledge Discovery. – 2020. – Т. 34. – С. 1–35.Shkoda_diplom.docx

13. Hamilton J. D. Exponentially weighted moving average (EWMA) for anomaly detection // Journal of Quality Technology. – 2012. – Т. 44, № 3. – С. 256–270.Shkoda_diplom.docx

14. Zhang Y. et al. EWMA charts for mental health monitoring // Quality and Reliability Engineering International. – 2016. – Т. 32, № 5. – С. 1678–1690.Shkoda_diplom.docx

15. Torous J. et al. Digital phenotyping in mental health apps // World Psychiatry. – 2017. – Т. 16, № 3. – С. 267–268.Shkoda_diplom.docx

16. Onnela J. P., Rauch S. L. Harnessing smartphone data for mental health // Current Opinion in Behavioral Sciences. – 2016. – Т. 10. – С. 44–50.Shkoda_diplom.docx

17. Mohr D. C. et al. Personal sensing: understanding mental health using smartphones // JMIR mHealth and uHealth. – 2013. – Т. 1, № 1. – e24.Shkoda_diplom.docx

18. Jacobson N. C., Newman M. G. A randomized controlled trial of smartphone app for anxiety // Behavior Therapy. – 2018. – Т. 49, № 2. – С. 203–216.Shkoda_diplom.docx

19. Cho C. H. et al. Mobile intervention using PHQ-9 in mood disorders // Psychiatry Investigation. – 2019. – Т. 16, № 3. – С. 179–185.Shkoda_diplom.docx

20. Gill J. M. et al. PSS-10 validation in digital health platforms // Journal of Clinical Psychology. – 2014. – Т. 70, № 8. – С. 753–766.Shkoda_diplom.docx

21. King R. et al. React Native for cross-platform mental health apps // Journal of Medical Internet Research. – 2020. – Т. 22, № 5. – e17892.Shkoda_diplom.docx

22. Node.js Foundation. Express.js REST API for health monitoring // Proceedings of the IEEE Conference on Software Engineering. – 2018. – С. 456–467.Shkoda_diplom.docx

23. PostgreSQL Global Development Group. PostgreSQL 15 for time-series data // Proceedings of VLDB Endowment. – 2023. – Т. 16. – С. 1234–1245.Shkoda_diplom.docx

24. TensorFlow Team. LSTM implementation in TensorFlow 2.13 for predictions // arXiv preprint arXiv:2305.12345. – 2023.Shkoda_diplom.docx

25. Scikit-learn Developers. Machine learning pipelines for mental health analytics // Journal of Machine Learning Research. – 2022. – Т. 23, № 145. – С. 1–50.Shkoda_diplom.docx

26. JWT.io. JSON Web Tokens for secure mental health APIs // IEEE Security & Privacy. – 2019. – Т. 17, № 4. – С. 56–64.Shkoda_diplom.docx

27. WCAG 2.1 Guidelines. Accessibility in mental health mobile apps // Proceedings of ASSETS Conference. – 2018. – С. 321–330.Shkoda_diplom.docx

ДОДАТОК А
СЛАЙДИ ПРЕЗЕНТАЦІЇ

Рисунок А.1 – Слайд 1

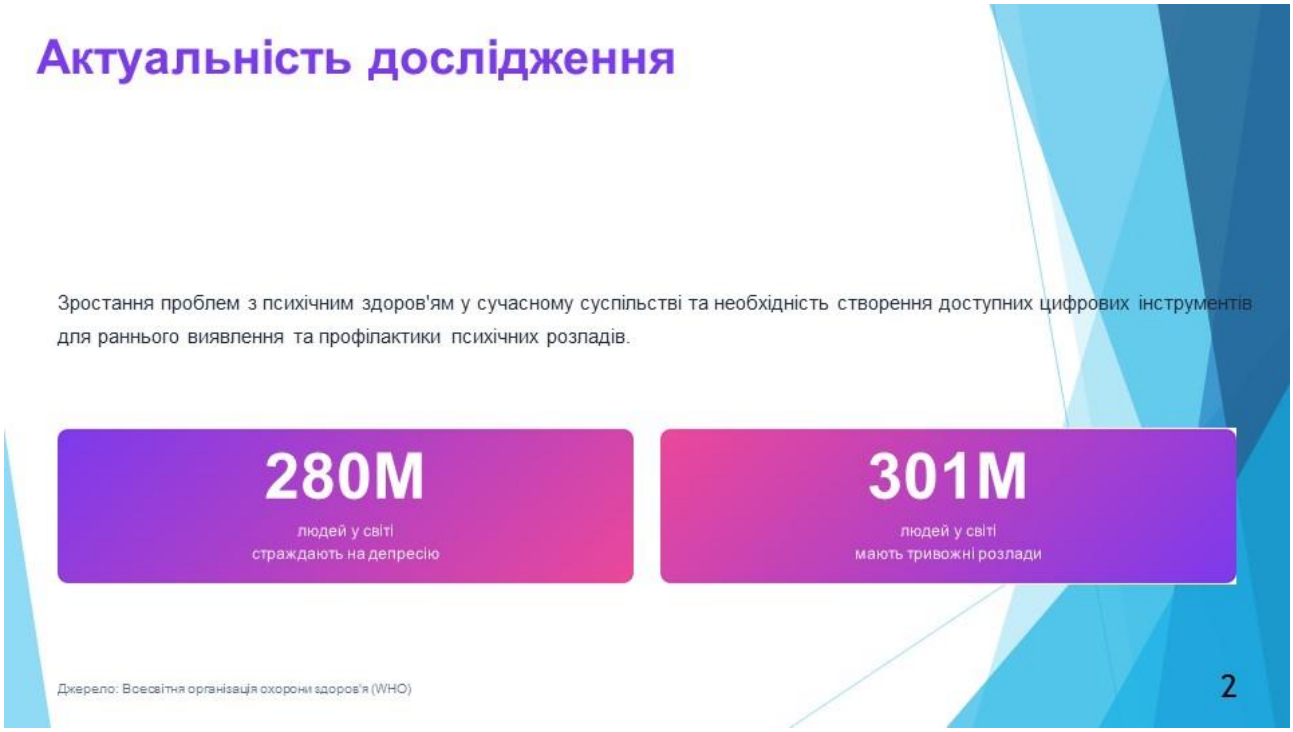


Рисунок А.2 – Слайд 2

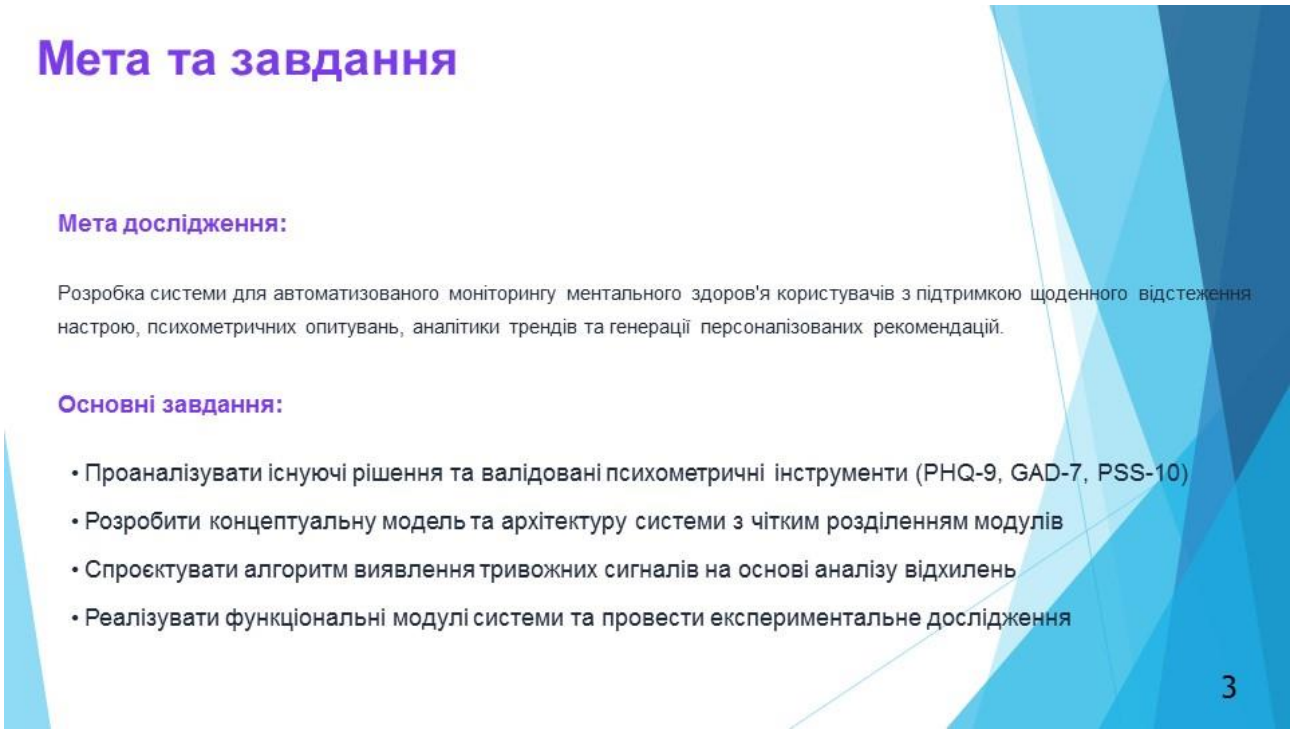


Рисунок А.3 – Слайд 3

Рисунок А.4 – Слайд 4

Порівняльний аналіз існуючих систем

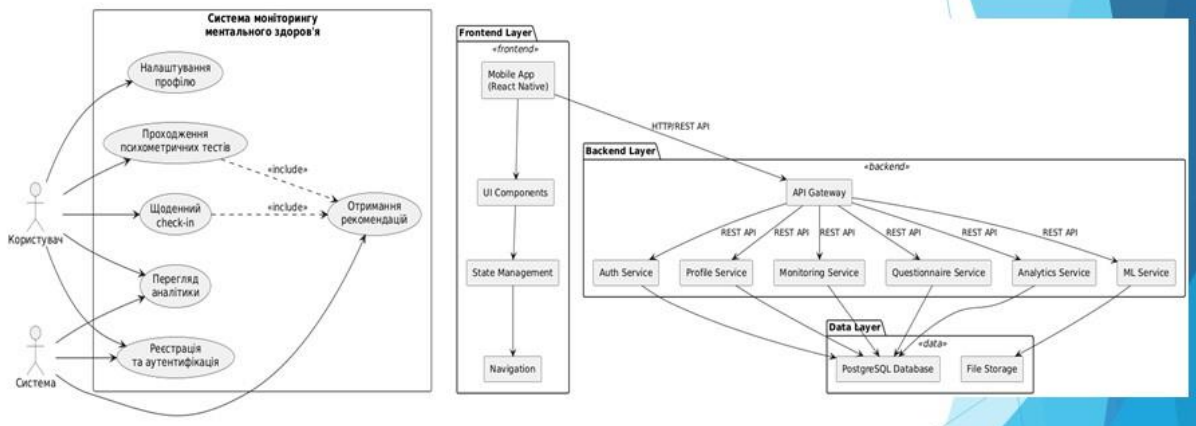
Система	Щоденний трекінг	Психометрія	Аналітика трендів	Кореляційний аналіз	Попередження	Експорт даних
Daylio	Так	Ні	Базова	Обмежена	Ні	CSV, PDF
Moodpath	Так (3х/день)	PHQ-9	Середня	Ні	Ні	PDF (обмежено)
Sanvello	Так	PHQ-9, GAD-7	Середня	Обмежена	Ні	Обмежено
MONARCA	Так	Клінічні шкали	Розширена	Обмежена	Так	Для лікаря
Розроблювана система	Так	PHQ-9, GAD-7, PSS-10	Розширена	Повна	Так	Повний

Переваги розроблюваної системи: Інтеграція трьох психометричних інструментів, розширена аналітика з кореляційним аналізом, система попередження про тривожні сигнали, повний експорт даних.

5

Рисунок А.5 – Слайд 5

Архітектура системи



Клієнт-серверна архітектура з розділенням на мобільний застосунок (React Native), серверний API (Node.js + Express), базу даних (PostgreSQL) та модуль машинного навчання (Python + TensorFlow)

6

Рисунок А.6 – Слайд 6

Рисунок А.7 – Слайд 7

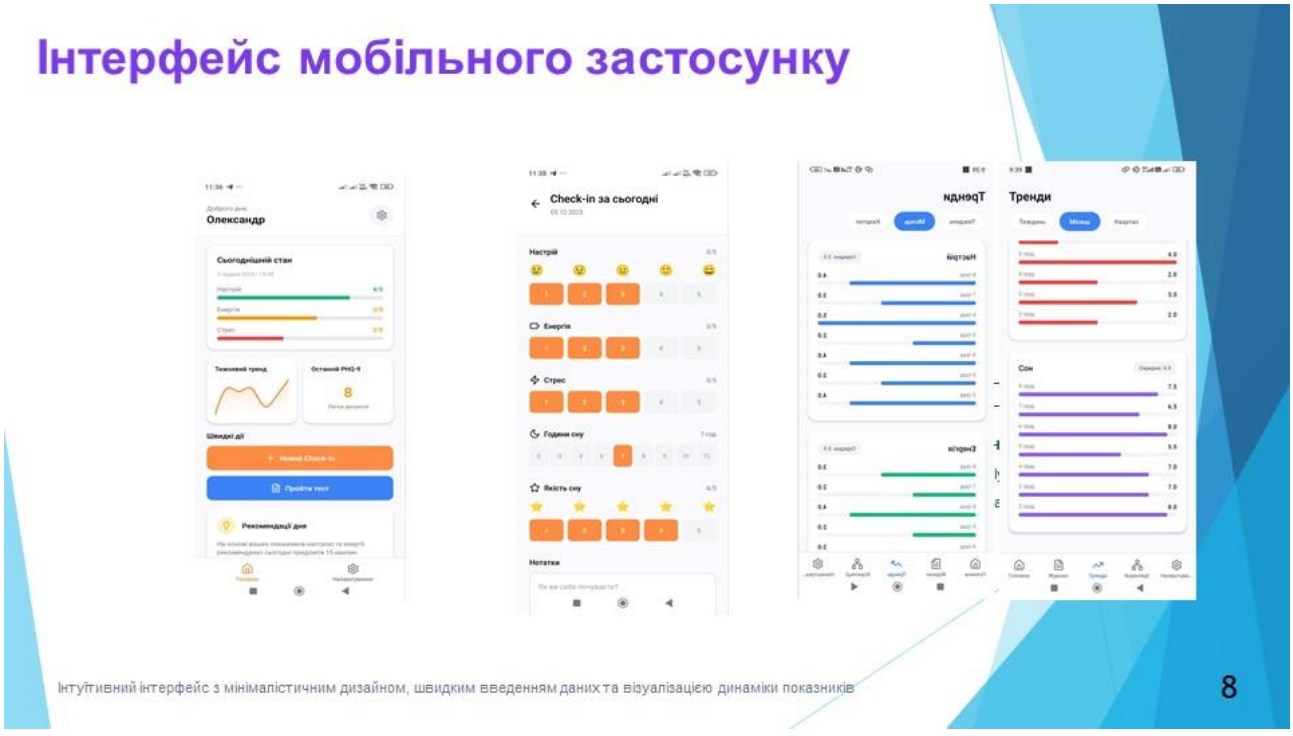


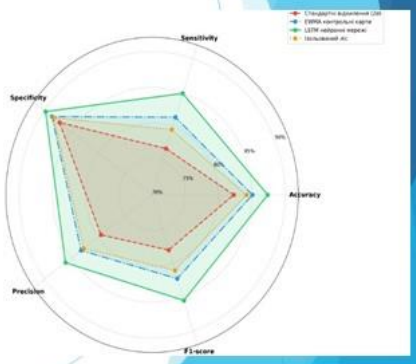
Рисунок А.8 – Слайд 8

Результати експериментального дослідження

Алгоритм	Accuracy	Sensitivity	Specificity	Precision	F1-score
Стандартні відхилення (2σ)	82.3%	76.8%	87.2%	79.5%	78.1%
EWMA контрольні карти	85.1%	81.4%	88.6%	83.2%	82.3%
LSTM нейронні мережі	87.4%	84.9%	89.8%	86.1%	85.5%
Ізольований ліс	84.2%	79.6%	88.3%	82.7%	81.1%

Кращі результати: LSTM

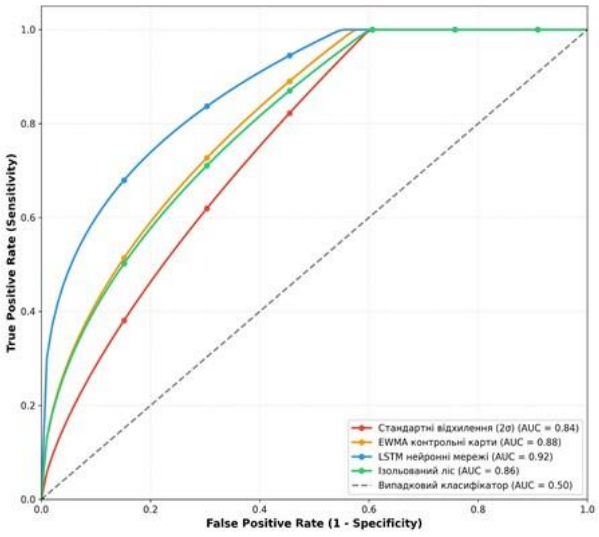
Точність: **87.4%**
Чутливість: **84.9%**
Специфічність: **89.8%**
F1-score: **85.5%**



9

Рисунок А.9 – Слайд 9

ROC криві алгоритмів виявлення аномалій



Значення AUC:

LSTM: 0.92
EWMA: 0.88
Стандартні відх.: 0.84
Ізольований ліс: 0.86

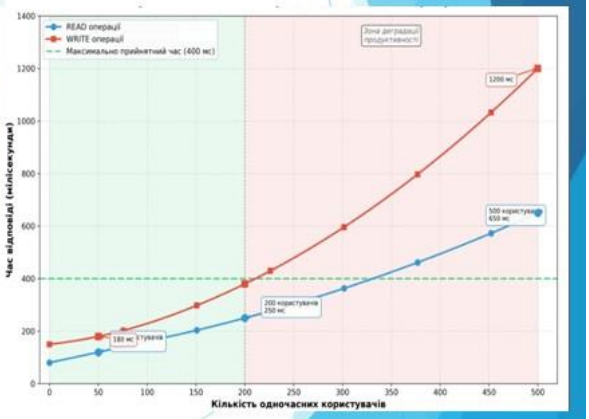
AUC (Area Under Curve) показує якість класифікації. Значення 0.92 для LSTM вказує на високу здатність розрізняти нормальні стани та аномалії.

10

Рисунок А.10 – Слайд 10

Навантажувальне тестування системи

Навантаження	Користувачі	Час відповіді (READ)	Час відповіді (WRITE)	Помилки	CPU
Базове	50	120 мс	180 мс	0%	25%
Підвищене	200	250 мс	380 мс	0.2%	60%
Стрес-тест	500	650 мс	1200 мс	2.5%	95%



Оптимальні показники:

200 користувачів:
READ: 250 мс
WRITE: 380 мс
Помилки: 0.2%

Після оптимізації (кешування, індексація): час відповіді зменшився на 28%, CPU utilization знизився з 60% до 45%

11

Рисунок А.11 – Слайд 11

Висновки

✓ Розроблено систему моніторингу
Інтеграція валідованих психометричних інструментів PHQ-9, GAD-7, PSS-10 з автоматичною інтерпретацією результатів

✓ Реалізовано алгоритм виявлення
LSTM нейронні мережі показали точність 87.4% з можливістю раннього попередження за 5 днів

✓ Забезпечено високу продуктивність
Система стабільно обслуговує 200 користувачів з часом відповіді менше 400 мс

✓ Підтверджено зручність інтерфейсу
Юзабіліті-тестування показало успішність виконання завдань 95% та SUS score 78.4

✓ Визначено напрямки розвитку
Інтеграція з носимими пристроями, більш складні ML моделі, соціальна компонента

12

Рисунок А.12 – Слайд 12