Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

Допущено до захисту
Завідувач кафедри
комп'ютерних наук,кандидат
технічних наук, доцент

_____Ірина МАШКІНА
(підпис)
«____» _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня «бакалавр»

Спеціальність 122 Комп'ютерні науки

Освітня програма 122.00.01 Інформатика

Тема: РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ДЛЯ ПІДТРИМКИ ПСИХОЕМОЦІЙНОГО СТАНУ

Виконав
студентка групи ІНб-1-21-4.0д
(шифр академічної групи)
Бондаренко Антоніна Олександрівна
(прізвище, ім'я, по батькові)

_____
(підпис)
Науковий керівник
к.п.н., доцент кафедри комп'ютерних наук
(науковий ступінь, наукове звання)
Глушак О. М.
(прізвище, ініціали)

_____
(підпис)

Київ – 2025

Київський столичний університет імені Бориса Грінченка
Факультет інформаційних технологій та математики
Кафедра комп'ютерних наук

«Затверджую»

Завідувач кафедри комп'ютерних наук, кандидат технічних наук, доцент
_Ірина МАШКІНА
(підпис)
«_____» _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

«Розробка мобільного додатку для підтримки психоемоційного стану»

Виконавець – студентка групи ІНб-1-21-4.0д,
спеціальності 122 Комп'ютерні наук,
освітньої програми 122.00.01 Інформатика
Бондаренко Антоніна Олександрівна

1. Вихідні дані: *Матеріали з UX/UI-дизайну та емоційного дизайну (Дон Норман, Emotional Design), офіційна документація React Native та Expo, приклади реалізації мобільних застосунків у сфері ментального здоров'я, дослідження ВООЗ, статистика психоемоційного стану молоді, методичні рекомендації щодо створення цифрових інтерфейсів для самопідтримки, аналітика ринку додатків Calm, Headspace, VOS тощо.*
2. Основні завдання: *Проаналізувати існуючі цифрові рішення у сфері емоційної підтримки. Сформувати концепцію взаємодії, що орієнтована на емоційний стан. Створити UX/UI-дизайн з урахуванням персоналізації під вибрану емоцію. Реалізувати прототип на базі React Native. Адаптувати інтерфейс додатку до емоційного стану користувача.*
3. Пояснювальна записка: *Обсяг – до 45 сторінок формату А4 комп'ютерного набору, оформлення згідно з державним стандартом і методичними рекомендаціями кафедри.*
4. Графічні матеріали: *Не передбачено окремо, але макети інтерфейсу включені як фрагменти реалізації.*
5. Додатки: *Лістинг основних конфігураційних файлів, фрагменти коду модулів додатку, зразки користувацьких сценаріїв (user flow).*
6. Строк подання роботи на кафедру: *«_____» _____2025 р.*

Науковий керівник
к.п.н., доцент
Глушак О.М.
_____ дата

Виконавець:
Бондаренко А. О.
_____ дата

Календарний план

№	Етапи виконання роботи	Термін виконання	Примітка
1	Формування теми кваліфікаційної роботи бакалавра та її затвердження	07.10.24 – 14.10.24	виконано
2	Аналіз проблеми, вивчення аналогів, формування цілей і завдань	25.11.24 – 08.12.24	виконано
3	Аналіз сучасних підходів до UX/UI, емоційного дизайну та вибір інструментів (React Native, Expo)	09.12.24 – 15.12.24	виконано
4	Розробка концепції додатку: сценарії взаємодії, структура інтерфейсу, персоналізація під емоції	15.12.24 – 05.03.25	виконано
5	Реалізація базових екранів і модулів (вибір емоції, головний екран, рефлексія, щоденник, список приємностей)	06.03.25 – 20.03.25	виконано
6	Впровадження локального збереження даних (AsyncStorage), адаптація інтерфейсу під емоційні сценарії	21.03.25 – 27.03.25	виконано
7	UX-тестування, усунення помилок, оптимізація логіки	28.03.25 – 35.04.25	виконано
8	Доробка інтерфейсів, впровадження статистики, тестування стабільності	25.04.25 – 15.05.25	виконано
9	Підготовка пояснювальної записки, графічних матеріалів, додатків	15.05.25 – 31.05.25	виконано
11	Захист кваліфікаційної роботи	16.06.2025	виконано

Здобувач

студентка ІНБ 1-21-4.0д

Бондаренко Антоніна Олександрівна

Керівник роботи

к.п.н., доцент кафедри комп'ютерних наук

Глушак О.М

РЕФЕРАТ бакалаврської роботи

Кваліфікаційна робота: 31 сторінка, 25 рисунків, 2 таблиці, 18 посилань.

Актуальність: Психоемоційна нестабільність сьогодні є ключовим викликом для молоді. Сучасні цифрові рішення переважно орієнтовані на формальний підхід, який не забезпечує достатнього рівня емпатії та адаптивності до індивідуальних потреб користувача.

Об'єкт дослідження: Цифрові мобільні інструменти емоційної самопідтримки.

Предмет дослідження: Структура та логіка адаптивного мобільного інтерфейсу, який активно взаємодіє з емоційним станом користувача через персоналізацію.

Мета роботи: Розробити мобільний додаток, який ефективно підтримує психоемоційний стан завдяки адаптивному інтерфейсу та емоційному дизайну.

Завдання:

1. Проаналізувати існуючі цифрові рішення у сфері емоційної підтримки.
2. Сформувати концепцію взаємодії, що орієнтована на емоційний стан.
3. Створити UX/UI-дизайн з урахуванням персоналізації під вибрану емоцію.
4. Реалізувати прототип на базі React Native.
5. Адаптувати інтерфейс додатку до емоційного стану користувача.

Основні результати: Розроблено мобільний додаток для психоемоційної підтримки молоді з персоналізованим UX/UI на базі React Native. Інтерфейс адаптується до вибраної емоції користувача, використовуючи візуальні та текстові сценарії, натхненні мультфільмом «Головоломка». Реалізовано модулі щоденника, рефлексій, тестів та вправ. Забезпечено просту навігацію, збереження даних через AsyncStorage. Протестовано функціональність і стабільність додатку на різних пристроях.

Ключові слова: психоемоційна підтримка, мобільний додаток, React Native, UX/UI, персоналізація, емоційний дизайн, «Головоломка», емоції, щоденник, рефлексія, тестування, вправи для емоцій, адаптивний інтерфейс, AsyncStorage, збереження даних, молодь, цифрове здоров’я, user experience, психологічна підтримка.

ЗМІСТ

ВСТУП

Актуальність теми: Психоемоційна нестабільність сьогодні є ключовим викликом для молоді. Сучасні цифрові рішення переважно орієнтовані на формальний підхід, який не забезпечує достатнього рівня емпатії та адаптивності до індивідуальних потреб користувача.

На ринку мобільних застосунків уже присутні десятки додатків, що позиціонують себе як “емоційна підтримка” - Calm, Headspace, VOS, UpLife тощо. Проте більшість із них використовують універсальні протоколи взаємодії (медитації, дихальні вправи), які не враховують особистісний контекст, моментальний стан чи індивідуальну чутливість користувача. Їхні інтерфейси формалізовані, статичні, часто сприймаються як «платформа з порадами», а не як суб’єктна взаємодія. Вони не реагують на користувача, а просто подають контент.

Водночас розвиток емоційного дизайну в UX-сфері відкриває нові можливості для створення інтерфейсів, які не лише передають функцію, але й викликають відчуття підтримки, безпеки, участі. Адаптація інтерфейсу, текстів і логіки додатку відповідно до емоційного стану користувача є сучасним підходом у розробці цифрових сервісів підтримки, що дозволяє створювати більш персоналізовані та ефективні рішення.

Таким чином, побудова інтерфейсу на основі поточного емоційного стану користувача підвищує ефективність цифрової самодопомоги та забезпечує персоналізовану взаємодію. Такий підхід дозволяє краще враховувати індивідуальні потреби користувача й адаптувати підтримку до його емоційного стану, що особливо важливо для людей у стані психоемоційної нестабільності.

Крім того, використання сучасних технологій, зокрема React Native, локального зберігання даних із застосуванням контекстів та конфігурацій, а також мобільної адаптивності, повністю відповідає актуальним вимогам IT-індустрії та стандартам підготовки бакалаврів у галузі комп’ютерних наук.

Мета роботи: розробити мобільний додаток, який ефективно підтримує психоемоційний стан завдяки адаптивному інтерфейсу та емоційному дизайну.

Завдання роботи:

1. Проаналізувати існуючі цифрові рішення у сфері емоційної підтримки.
2. Сформувати концепцію взаємодії, що орієнтована на емоційний стан.
3. Створити UX/UI-дизайн з урахуванням персоналізації під вибрану емоцію.
4. Реалізувати прототип на базі React Native.
5. Адаптувати інтерфейс додатку до емоційного стану користувача.

Об'єктом дослідження є цифрові мобільні інструменти емоційної самопідтримки, тобто застосунки, які призначені для щоденного використання з метою самостійного відслідковування, аналізу та покращення психоемоційного стану користувача. Це продукти, що поєднують інтерфейсні рішення, психологічні методики та цифрову взаємодію для створення відчуття підтримки без участі фахівця.

Методи дослідження: аналіз цифрових рішень у сфері емоційної підтримки, UX-аналіз інтерфейсів конкурентів, моделювання структури взаємодії, прототипування в середовищі React Native, а також UX-тестування з залученням користувачів для оцінки ефективності адаптивного інтерфейсу.

Практична значущість роботи полягає у створенні прототипу мобільного застосунку, який демонструє принципи емоційно-адаптивного UX. Розроблене рішення може бути використане як основа для повноцінного цифрового продукту в сфері ментального здоров’я, адаптованого до потреб молоді. Запропонований підхід до персоналізації інтерфейсу за емоційним станом користувача може бути інтегрований у психологічні сервіси, освітні платформи або інші застосунки, де важливе емпатійне середовище.

РОЗДІЛ 1: АНАЛІЗ ПРОБЛЕМИ ТА ДОСЛІДЖЕННЯ ІСНУЮЧИХ РІШЕНЬ

1.1. Проблематика психоемоційної нестабільності української молоді

Після 2020 року психоемоційний стан української молоді суттєво погіршився через пандемію COVID-19, початок повномасштабної війни, нестабільність в освіті й економіці.

Дослідження HBSC (2021/2022 навчальний рік) показало, що лише 36% українських підлітків регулярно прокидалися відпочилими, що на 8% менше порівняно з 44% у 2018 році. Це свідчить про погіршення якості сну та загального психоемоційного стану молоді за останні роки. Опитування, проведене VoxUkraine у листопаді 2023 року серед студентів київських університетів, виявило середній рівень тривожності на рівні 5,4 за 10-бальною шкалою, а відчуття безпеки — 6,4. Ці дані вказують на хронічне емоційне напруження, яке стало повсякденною реальністю для багатьох молодих людей [1].

Особливо вразливими є студенти з регіонів, постраждалих від бойових дій, та вимушені переселенці. За даними кафедри психіатрії НМУ імені О.О. Богомольця, у таких груп спостерігаються симптоми депресії, тривожних розладів, посттравматичного стресового розладу (ПТСР), а також порушення сну, зниження концентрації та соціальна ізоляція. Ці стани негативно позначаються на академічній успішності та соціальній інтеграції [2].

Значним викликом є також інтеграція молодих українців, які виїхали за кордон через війну. Згідно з опитуванням U-Report Europe (лютий 2024, 616 респондентів), лише 19% молодих українців за кордоном відчувають себе повністю інтегрованими в нові спільноти. Основними перешкодами є мовний бар’єр, культурні відмінності та обмежений доступ до місцевих ресурсів, таких як психологічна підтримка чи освітні програми. Ці фактори посилюють відчуття ізоляції, сприяють розвитку тривожності та ускладнюють

психоемоційну адаптацію, що підкреслює потребу в доступних цифрових інструментах для підтримки [3].

Враховуючи зазначені проблеми, розробка мобільного додатку з емоційною підтримкою і персоналізованим зворотним зв'язком є актуальною. Такий інструмент здатний зменшити відчуття самотності, сприяти адаптації та покращити загальний психоемоційний стан молоді.

1.2 Аналіз існуючих цифрових рішень у сфері психоемоційної підтримки

Для оцінки сучасних підходів до психоемоційної підтримки молоді було проаналізовано три популярні в Україні мобільні застосунки: BetterMe: Mental Health, VOS і Melty. Вибір цих платформ зумовлений їхньою популярністю, доступністю та наявністю функцій персоналізації. Метою аналізу було визначення рівня адаптації цих застосунків до емоційного стану користувачів, а також виявлення їхніх недоліків для подальшого врахування при розробці нового рішення.

Таблиця 1

Порівняльна характеристика мобільних застосунків для психоемоційної підтримки

Критерій	BetterMe: Mental Health	VOS	Melty
Тип платформи	мобільний застосунок	мобільний застосунок	мобільний застосунок
Основна ціль	зниження стресу, сон, медитації	моніторинг настрою, афірмації	самодопомога, поради
Емоція як точка входу	ні	ні	ні
Трекінг емоцій у реальному часі	(фіксована програма)	(щоденник настрою)	(настрій після тесту)

Персоналізація контенту	за цілями і часом доби	за настроєм	за результатами тестів
UX-адаптація під емоційний стан	ні	змінюється контент, але не інтерфейс	ні
Зворотний емоційний фідбек	немає	немає	немає
Гнучкість сценарію взаємодії	фіксований	частково	частково
Мова інтерфейсу	українська	англійська	українська
Наявність консультацій	немає	є доступ до фахівця	можливість запису

Аналіз показав, що усі три застосунки використовують реактивний підхід до персоналізації, тобто адаптують контент на основі попередньо введених даних про настрій або цілі користувача [4]. Наприклад, VOS пропонує зміну контенту залежно від емоційного стану, але не адаптує логіку інтерфейсу чи навігацію. BetterMe фокусується на цілях і часі доби, ігноруючи поточний емоційний контекст. Melty використовує тести для визначення настрою, але не забезпечує зворотного емоційного зв’язку чи гнучкості взаємодії.

Критичним недоліком усіх застосунків є відсутність емоції як основного тригера для взаємодії. Жоден із них не адаптує інтерфейс або логіку навігації в реальному часі залежно від емоційного стану користувача. Це обмежує їхню ефективність у контексті швидкої адаптації до потреб української молоді, яка часто перебуває в стані стресу чи тривоги.

1.3. Актуальність емоційного дизайну як нового UX-підходу

Сучасні мобільні застосунки для ментального здоров’я зазвичай орієнтуються на вибір функцій користувачем, а не на його поточний емоційний стан. Однак для української молоді, яка зазнає хронічного стресу,

тривожності чи труднощів адаптації, саме емоційний стан є ключовим фактором, що визначає ефективність взаємодії з цифровими продуктами.

Емоційно-чутливий UX-підхід передбачає початок взаємодії з оцінки емоційного стану користувача (наприклад, питання «Як ти себе почуваєш?»). Це дозволяє адаптувати не лише контент, а й візуальні елементи, тональність текстів, типи вправ і логіку навігації. Такий підхід ґрунтується на теорії емоційного дизайну Дональда Нормана, який наголошує, що емоційний зв’язок між користувачем і продуктом підвищує довіру, залученість і загальну ефективність взаємодії [5].

Для підкреслення унікальності емоційного дизайну доцільно порівняти його з іншими теоріями UX-дизайну. Наприклад, модель «Flow» Мігая Чиксентміхайї акцентує на створенні стану повного занурення користувача в діяльність через баланс між складністю завдання та навичками користувача [6]. Ця модель ефективна для ігор чи освітніх платформ, але менш застосовна до психоемоційної підтримки, де пріоритетом є емпатична взаємодія, а не досягнення стану потоку. Емоційно-чутливий дизайн, на відміну від «Flow», фокусується на адаптації до миттєвих емоційних потреб користувача, що робить його більш релевантним для вирішення проблем тривожності та стресу української молоді.

Емоційно-чутливий дизайн пропонує значні переваги для створення ефективного користувацького досвіду в мобільних додатках. Він забезпечує персоналізацію в реальному часі, адаптуючи інтерфейс і контент до поточного емоційного стану користувача, що створює індивідуальний і комфортний досвід. Завдяки використанню тональності, текстів і візуальних елементів, які відповідають емоційним потребам, дизайн сприяє емпатичній взаємодії, допомагаючи користувачам відчувати розуміння та підтримку. Такий підхід значно підвищує залученість, оскільки створює відчуття турботи, що є особливо важливим для української молоді в кризових умовах, де емоційна підтримка відіграє ключову роль. Впровадження емоційно-чутливого дизайну в додаток для психоемоційної підтримки

дозволяє розробити емпатичний і ефективний користувацький досвід, який враховує унікальні потреби цільової аудиторії та сприяє їхньому психологічному комфорту.

РОЗДІЛ 2: ВИМОГИ ДО ПРОЄКТУВАННЯ ТА ДИЗАЙН-СТРАТЕГІЇ МОБІЛЬНОГО ДОДАТКУ

2.1. Емоційний дизайн як основа UX

В основі архітектури додатку закладено принципи емоційного дизайну. Ці принципи забезпечують користувачу відчуття підтримки з першої секунди взаємодії з додатком. Головним інструментом цього підходу є вибір емоційного стану як стартової точки взаємодії, що дозволяє одразу персоналізувати інтерфейс, змінюючи кольорову гаму, тексти та функціональні сценарії відповідно до поточного стану користувача.

Перевага цього підходу полягає у зменшенні когнітивного навантаження та підвищенні залученості, що підтверджується сучасними дослідженнями у сфері UX для ментального здоров'я. Складні анкети, багатоступеневі меню та вимоги до введення тексту були повністю виключені - вибір емоції фіксується автоматично, а наступний екран миттєво адаптується до обраного стану [7].

2.2. Візуальна ідентичність на основі мультфільму «Головоломка» (Inside Out)

Для створення емоційно-зрозумілого інтерфейсу було обрано візуальну систему, засновану на персонажах мультфільму «Головоломка» (Inside Out) - Радість, Сум, Злість, Страх, Відраза та інші [8].

З технічної сторони це рішення забезпечує швидку та інтуїтивну навігацію навіть для користувачів у стресовому стані. Образи персонажів виконують не лише декоративну функцію, а й виступають посередниками між внутрішнім емоційним досвідом користувача та логікою програми адже кожен персонаж звертається до користувача індивідуально підібраними повідомленнями. Наприклад, для емоції Страх розроблено спеціальні тексти-звернення: «Я не ворог. Я — сигнал. Я приходжу не для того, щоб паралізувати, а щоб зупинити тебе на секунду і дати подумати: “А як зробити

безпечніше?”. Інколи я з’являюсь навіть тоді, коли загрози нема. Але це тому, що я стараюсь. Бо мені не байдуже. Бо мені важливо, щоб ти був цілим.».

Весь інтерфейс та вербальна частина були спроєктовані таким чином, щоб активувати вісцеральний рівень сприйняття за Доном Норманом [5]. Замість нейтральних формулювань або сухих діагнозів, користувач взаємодіє з емпатичним візуальним і текстовим контентом.

Актуальність такого підходу посилюється ностальгійною прив’язкою до франшизи - вихід Inside Out 2 у 2024 році викликав нову хвилю популярності, також додатково частка аудиторії у віці 25–44 роки, склала близько чверті всіх глядачів. За перший вікенд фільм зібрав 154 млн доларів, що свідчить про масову впізнаваність і емоційну залученість цієї вікової групи [9]. Таким чином, використання візуальних кодів Inside Out у додатку формує довіру через апеляцію до емоційної пам’яті дорослішої аудиторії, що особливо важливо для створення дружнього інтерфейсу.

Крім того, проведений аналіз ринку показав відсутність прямих конкурентів у сегменті wellness-додатків, які б використовували подібну емоційно-персоналізовану стратегію. Більшість аналогічних продуктів пропонують клінічно-нейтральний дизайн, тоді як запропонований підхід ґрунтується на унікальній візуальній і вербальній комунікації від імені емоцій, що виділяє продукт на ринку та підвищує залученість користувачів [10].

Ще однією перевагою такої візуальної ідентичності є її кросвіковість: персонажі Inside Out відомі як дітям і підліткам, так і дорослим користувачам. Завдяки цьому інтерфейс додатку можна легко адаптувати до різних вікових груп: для молодших реалізована гейміфікована форма взаємодії, для старших - простір для рефлексії та самоусвідомлення. Це відповідає сучасному тренду персоналізації UX-дизайну та інклюзивності цифрових рішень [10].

Психоемоційна релевантність вибраної стратегії також підтверджується зростаючим запитом на м’які, емпатичні інструменти самодопомоги серед молоді, особливо на фоні підвищення рівня тривожності та нестабільності

емоційного стану [11]. Замість боротьби з емоціями додаток вибудовує діалог, допомагаючи користувачам приймати й осмислювати власний стан. Таким чином, вибір Inside Out як концептуальної бази для візуальної ідентичності не лише спрощує користування додатком, а й сприяє формуванню довіри, емпатії та глибшого розуміння себе — що й становить головну мету продукту.

2.3. Кольорові та вербальні стратегії

У додатку для кожної емоції використовується окремий колір, що задає основну стилістику інтерфейсу: жовтий для радості, синій для суму, червоний для злості, фіолетовий для страху, зелений для відрази. Кольорова палітра реалізується централізовано й підтягується для фону, кнопок і значущих елементів інтерфейсу після вибору емоції. Це забезпечує узгодженість візуальної системи, прискорює навігацію та мінімізує помилки при взаємодії [5]. Дослідження показують, що використання кольорів як інструменту кодування інформації підсилює швидкість сприйняття, сприяє емоційній ідентифікації й знижує когнітивне навантаження на користувача [12]. Такий підхід також спрощує розширення додатку та його подальшу підтримку завдяки єдиній структурі стилів. Кольорове диференціювання емоцій підсилює вісцеральний рівень взаємодії та позитивно впливає на суб'єктивне відчуття контролю і безпеки в інтерфейсі [5].

2.4. Сценарії користувача та архітектура функціоналу

Основні сценарії взаємодії користувача з додатком передбачають послідовність дій від запуску програми та вибору емоційного стану до виконання ключових функцій, таких як запис рефлексії, ведення щоденника, проходження психологічного тесту або взаємодія зі списком приємностей. Після виконання будь-якої дії користувач повертається на персоналізований головний екран, що дозволяє оперативно реагувати на зміну емоційного стану або виконати іншу дію. Навігація організована так, щоб мінімізувати

кількість кроків і забезпечити інтуїтивність використання навіть для користувачів без досвіду роботи з подібними цифровими продуктами.

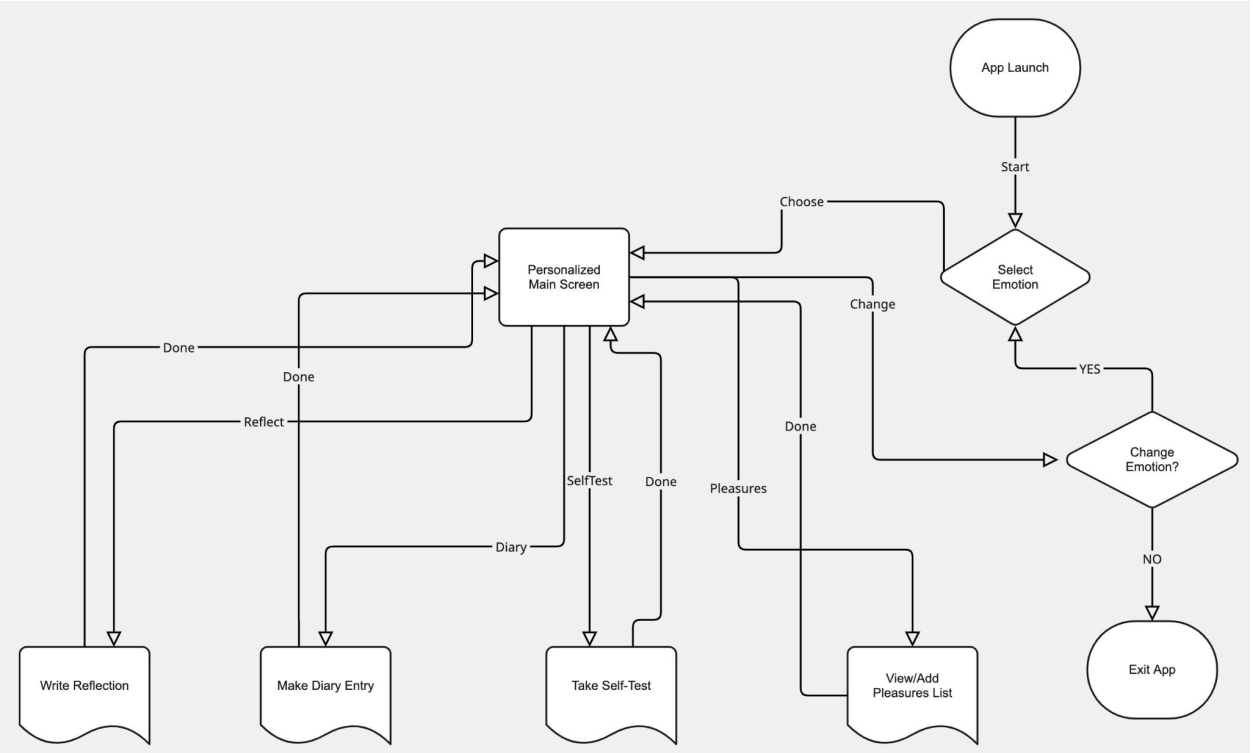

Рисунок 2.1. Сценарій користувача (user flow) мобільного додатку

На схемі (Рис. 2.1) представлено основні етапи взаємодії. Після запуску додатку користувач обирає емоцію, на основі якої формується персоналізований головний екран із динамічною адаптацією інтерфейсу. З цього екрану доступні чотири основні функції: запис рефлексії, ведення щоденника, проходження психологічного тесту та перегляд або доповнення списку приємностей. Після виконання будь-якої дії можливе повернення на головний екран, зміна емоції або вихід із додатку.

Функціонал додатку реалізовано за модульною архітектурою, де кожен основний компонент працює незалежно й взаємодіє з централізованим сховищем даних та конфігураційними файлами. Це спрощує підтримку, розширення й масштабування додатку, а також підвищує стабільність роботи системи. Конфігураційні файли містять налаштування для кожної емоції та тесту, що дозволяє оперативно змінювати параметри без втручання в

основний код. Всі дані зберігаються у локальному сховищі, забезпечуючи швидкий доступ і незалежність від зовнішніх серверів.

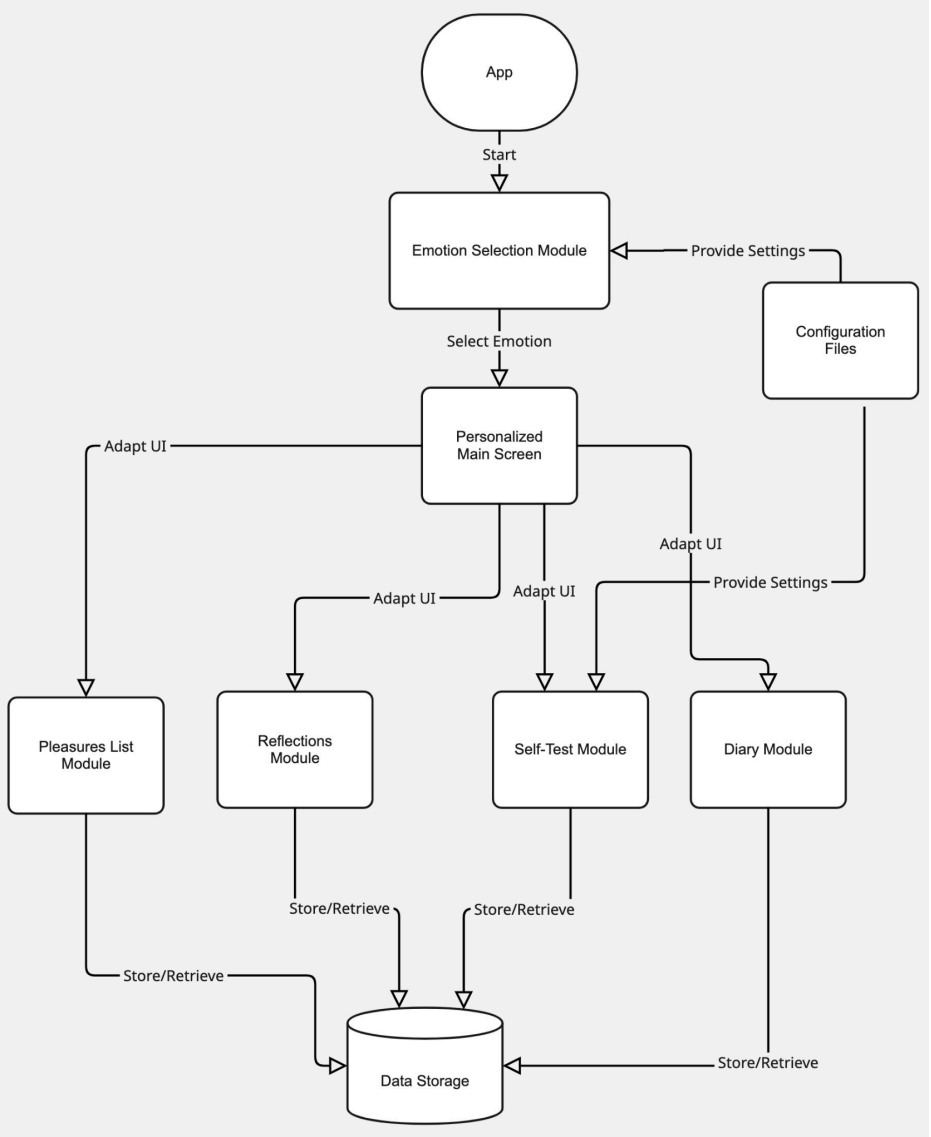

Рисунок 2.2. Модульна архітектура мобільного додатку

Схема (Рис. 2.2) ілюструє взаємозв’язки між основними модулями додатку. Після запуску додатку й вибору емоції налаштування інтерфейсу підтягуються з конфігураційних файлів, а головний екран адаптується відповідно до вибраного стану. Далі користувач може обирати будь-який із функціональних модулів – щоденник, рефлексії, психологічні тести або список приємностей. Всі модулі здійснюють збереження та отримання даних через централізований модуль зберігання (Data Storage), що забезпечує цілісність і синхронізацію інформації в додатку.

Такий підхід до побудови сценаріїв і архітектури функціоналу дозволяє забезпечити адаптивність інтерфейсу, простоту використання, можливість швидкого розширення функціоналу та підвищує загальну стійкість і якість цифрового продукту.

РОЗДІЛ 3. ТЕХНІЧНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ДОДАТКУ

3.1. Вибір технологічної платформи

Для розробки додатку було обрано React Native, прототипування здійснювалося в середовищі Snack від Expo.

React Native підтримує кросплатформну розробку і дозволяє створювати додатки для iOS і Android на єдиній кодовій базі, що значно скорочує час і ресурси. Його компонентний підхід дає змогу формувати інтерфейс із незалежних модульних елементів, таких як кнопки, текстові поля і анімації, які можна багаторазово використовувати. Ці елементи легко адаптувати для динамічних змін інтерфейсу, наприклад, для зміни кольорової палітри, зображень або тексту залежно від обраної користувачем емоції. Така гнучкість ідеально відповідає проєктам, орієнтованим на емоційний дизайн, де інтерфейс має бути інтуїтивно зрозумілим, адаптивним і залучати цільову аудиторію [13].

Використання Snack від платформи Expo значно спростило процес тестування. Середовище Snack дозволило переглядати зміни в реальному часі на реальних пристроях без необхідності налаштування складного локального середовища. Це прискорило ітерації дизайну, даючи змогу швидко тестувати різні варіанти інтерфейсу. Такий підхід виявився особливо цінним для UX-тестування, оскільки була можливість оперативно вдосконалювати кристувацький досвід.

Активна спільнота React Native і численні бібліотеки значно спростили розробку складних функцій. Завдяки цим інструментам вдалося оперативно реалізувати збереження налаштувань користувача та обробку зображень, що підвищило зручність і привабливість додатку. Крім того, компонентний підхід полегшив інтеграцію цих функцій, оскільки кожен елемент інтерфейсу розроблявся та тестувався окремо [14].

3.2. Архітектура збереження даних

Для збереження даних було обрано локальне сховище через AsyncStorage, що дозволило спростити архітектуру додатку, прискорити розробку та зосередитися на реалізації UX/UI-логіки. AsyncStorage зберігає даних користувачів безпосередньо на пристрої, що усуває потребу в серверній базі даних на данному етапі розробки і оптимізує продуктивність [15].

Для локальної ідентифікації користувачів під час створення нового профілю дані додаються до масиву профілів у локальному сховищі. Функція *registerUser* додає нового користувача до цього масиву та зберігає оновлений список у AsyncStorage під ключем 'USERS'. Цей підхід забезпечує швидкий доступ до даних і спрощує керування профілями (Рис. 3.1).

```
export async function registerUser(userObj) {
  const users = await getUsers();
  users.push(userObj);
  await AsyncStorage.setItem('USERS', JSON.stringify(users));
}
```

Рисунок 3.1. Фрагмент коду з модуля роботи з користувачами.

Після успішної реєстрації або входу логін активного користувача зберігається окремо в AsyncStorage під ключем CURRENT_LOGIN (Рис. 3.2).

```
await AsyncStorage.setItem('CURRENT_LOGIN', login);
```

Рисунок 3.2. Збереження логіну активного користувача у локальному сховищі

Далі всі персональні дані (з роділів щоденник, рефлексії, результати тестів, приємності) підтягуються та зберігаються саме для цього логіну. Для цього в кожному відповідному модулі під час завантаження даних спочатку дістається значення CURRENT_LOGIN, і вже потім дані фільтруються або записуються з прив’язкою до користувача (Рис. 3.3).

```
const login = await AsyncStorage.getItem('CURRENT_LOGIN');
```

Рисунок 3.3. Фрагмент із DiaryScreen.js. Отримання активного логіну

Під час завантаження записів з локального сховища відбираються лише ті, які належать активному користувачу, і ці дані записуються у локальний state для подальшого відображення на екрані (Рис. 3.4).

```
if (stored) {
        const allRecords = JSON.parse(stored);
        const filtered = {};
        Object.entries(allRecords).forEach(([date, entry]) => {
          if (entry.login === login) filtered[date] = entry;
        });
        setRecords(filtered);
      } else setRecords({});
```

Рисунок 3.4. Фрагмент із DiaryScreen.js.

При збереженні нового запису у щоденник до об’єкта entry додається логін активного користувача. Таким чином всі збережені дані однозначно належать конкретному профілю і далі при підвантаженні записи фільтруються саме по цьому логіну (Рис. 3.5).

```
const handleSave = async () => {
    if (!Object.values(answers).some(x => typeof x === 'string' ? x.trim() : x)) return;
    if (!currentLogin) return;
    const stored = await AsyncStorage.getItem(STORAGE_KEY);
    const allRecords = stored ? JSON.parse(stored) : {};
    const newEntry = { ...answers, date: selectedDate, login: currentLogin };
    allRecords[selectedDate] = newEntry;
    await AsyncStorage.setItem(STORAGE_KEY, JSON.stringify(allRecords));
    const filtered = {};
    Object.entries(allRecords).forEach(([date, entry]) => {
      if (entry.login === currentLogin) filtered[date] = entry;
    });
    setRecords(filtered);
    setShowForm(false);
    setShowEntry(true);
    setEditMode(false);
  };
```

Рисунок 3.5. Фрагмент із DiaryScreen.js.

Структура локального зберігання даних у додатку організована через унікальні ключі для кожного типу інформації (Таблиця 2).

Таблиця 2.

Основні ключі та їх призначення

Ключ	Призначення
'USERS'	Масив усіх зареєстрованих користувачів (логін, пароль, ім'я тощо)
'CURRENT_LOGIN'	Логін активного користувача
'DIARY_RECORDS'	Об’єкт усіх записів щоденника для всіх користувачів
'REFLECTIONS_RECORDS'	Масив усіх рефлексій для всіх користувачів
'SELFTEST_{login}_{testKey}'	Масив результатів проходження одного тесту для конкретного користувача
'PLEASURES_SETUP_{login}'	Об’єкт зі списком приємностей і статусів для конкретного користувача

3.3. Динамічна кастомізація інтерфейсу

Реалізований адаптивний інтерфейс змінюється залежно від обраної емоції та автоматично змінює кольорову схему, зображення емоцій та текстові повідомлення. Конфігурація емоцій винесена в окремий файл emotionConfigs.js, де для кожної емоції визначено ключові властивості: колір, градієнт, іконка персонажа та масив повідомлень (Рис. 3.6).

```
//КОНФУЗ
  'Конфуз': {
  themeColor: '#726A95', // темний лавандовий
    gradient: ['#726A95', '#927E9A'], // темно-ліловий до сіро-лілового
  characterImage: {
    uri: 'https://images-wixmp-ed30a86b8c4ca887773594c2.wixmp.com/f/846a9086
  },
  messages: [
    'Привіт, я — Конфуз. Здається, зараз ти трохи розгублений, щось пішло не
    'О, здається, ми знову в незручній ситуації. Ти вже знаєш, як це буває —
    'Я з'являюсь, коли щось іде не так, і тобі стає незручно. Іноді здається
  ],
},
```

Рисунок 3.6. Конфігурація емоцій у файлі emotionConfigs.js

При рендерингу головного екрану (HomeScreen.js) додаток визначає поточну активну емоцію та динамічно підтягує відповідні налаштування з файлу конфігурації, змінюючи фон, іконку персонажа та текст (Рис. 3.7).

```
const gradientColors = config.gradient || [config.themeColor, '#222'];

return (
  <LinearGradient colors={gradientColors} style={styles.container} start={{x: 0.2, y: 0.1}} end={{x: 1, y: 1}}>
```

Рисунок 3.7. Вибір кольорів градієнту для фону головного екрану відповідно до активної емоції та динамічне застосування їх у компоненті LinearGradient.

Після вибору емоції користувачем (EmotionSelectScreen.js), ця інформація зберігається функцією setActiveEmotion(), і інтерфейс адаптується під новий емоційний сценарій (Рис. 3.8).

```
const handleEmotionSelect = (emotionName) => {
    setActiveEmotion(emotionName);
    emotionVisits[emotionName] = (emotionVisits[emotionName] || 0) + 1;
    navigation.replace('Main');
  };
```

Рисунок 3.8. Збереження вибраної емоції та запуск адаптації інтерфейсу під новий емоційний сценарій після вибору користувачем.

Ця побудова за модулями дозволяє додавати нові емоційні стани або змінювати їхнє оформлення без зміни основної логіки додатку. Зміна всіх

стилів, іконок і повідомлень відбувається одразу після вибору емоції. Реалізація повністю відповідає принципам сучасного емоційного дизайну, а також підвищує масштабованість і гнучкість продукту.

3.4. Реалізація підрахунку вибору емоції користувачем

Для персоналізації взаємодії також використовується простий об’єкт-лічильник emotionVisits, який зберігає кількість звернень до кожної емоції за час відкриття додатку (sessionMemory.js). Коли користувач обирає емоцію, значення лічильника відповідної емоції збільшується на одиницю.

```
const handleEmotionSelect = (emotionName) => {
  setActiveEmotion(emotionName);
  emotionVisits[emotionName] = (emotionVisits[emotionName] || 0) + 1;
  navigation.replace('Main');
};
```

Рисунок 3.9. Збільшення лічильника вибраної емоції у функції handleEmotionSelect (EmotionSelectScreen.js).

При формуванні повідомлення від персонажа на головному екрані додаток враховує, скільки разів користувач звертався до тієї чи іншої емоції.

```
const visitCount = emotionVisits[emotionName] || 1;
  let message = config.messages[0];
  if (visitCount === 2) message = config.messages[1] || config.messages[0];
  if (visitCount > 2) message = config.messages[2] || config.messages[0];
```

Рисунок 3.10. Вибір відповідного мотиваційного повідомлення залежно від кількості звернень до емоції (HomeScreen.js).

Залежно від лічильника відображаються різні тексти, якій залежить від поточного стану користувача та його попередньої взаємодії. Підрахунок вибору емоцій відбувається лише під час однієї сесії, після закриття додатку дані скидаються. Це просте рішення значно підвищує залученість і створює відчуття, що додаток “піклується” про користувача.

3.5. Реалізація екрану рефлексій

Модуль рефлексій створений, щоб користувачі могли швидко фіксувати свої емоційні стани та аналізувати їх надалі. Основна увага приділена простоті й персоналізації, щоб взаємодія була комфортною та відповідала принципам емоційного дизайну.

Збереження нотаток реалізовано через локальне сховище AsyncStorage. Кожна нотатка містить унікальний ідентифікатор, логін користувача, емоцію, заголовок, текст і дату. Емоція автоматично фіксується за активним станом користувача, що зменшує кількість дій для створення запису.

```
const saveRecord = async () => {
    if (!title.trim() || !currentLogin) return;
    try {
      const newRecord = {
        id: Date.now().toString(),
        login: currentLogin,
        emotion,
        title,
        text,
        date: new Date()
      };
      const stored = await AsyncStorage.getItem(STORAGE_KEY);
      const allRecords = stored ? JSON.parse(stored) : [];
      const updated = [...allRecords, newRecord];
      await AsyncStorage.setItem(STORAGE_KEY, JSON.stringify(updated));
      setRecords(updated.filter(r => r.login === currentLogin).sort((a, b) => new Date(b.date) - new Date(a.date)));
      setModalVisible(false);
      setTitle('');
      setText('');
    } catch (e) {}
  };
```

Рисунок 3.11. Формування й збереження нової нотатки користувача у локальному сховищі (ReflectionsScreen.js).

Кожен запис супроводжується іконкою та кольором відповідно до обраної емоції (з конфігурації emotionConfigs.js).

Окрім збереження нотаток, модуль рефлексій містить просту статистику. Додаток підраховує (Рис. 3.12), яка частка записів створена в кожному емоційному стані, і відображає ці дані у вигляді відсотків із відповідними іконками.

```
function getEmotionStats(list) {
  if (!list?.length) return {};
  const total = list.length;
  const stats = {};
  list.forEach(item => {
    stats[item.emotion] = (stats[item.emotion] || 0) + 1;
  });
  Object.keys(stats).forEach(key => {
    stats[key] = Math.round((stats[key] / total) * 100);
  });
  return stats;
}
```

Рисунок 3.12. Підрахунок частки нотаток для кожної емоції у функції getEmotionStats (ReflectionsScreen.js).

Модуль рефлексій створено так, щоб нотатки було легко додавати, редагувати чи видаляти через просте модальне вікно і переглядати у списку. Автоматична прив'язка до поточної емоції робить процес швидким і ненав'язливим.

3.6. Реалізація форми щоденника

Модуль щоденника розроблений так, щоб користувач міг швидко фіксувати свій емоційний стан і залишати короткі записи кожен день. Основний акцент зроблено на простоті й гнучкості тож користувач може заповнювати будь-які поля форми на власний розсуд, залишити лише фото або написати тільки підсумок дня і це не впливає на читабельність фінального тексту.

Після натискання на дату в календарі додаток перевіряє, чи запис на цю дату вже існує. Якщо запис знайдено, користувачу відображається готовий текстовий запис у вигляді "листа", який включає всі раніше збережені відповіді, вибрану емоцію та фото.

Якщо запису немає, відкривається форма для створення нового запису. Емоція автоматично підтягується як поточна активна, але користувач може змінити її на будь-яку іншу через скрол з іконками (Рис. 3.13). Колір фону і відтінок полів залежить від обраної емоції.

```
const handleDayPress = (day) => {
    setSelectedDate(day.dateString);
    if (records[day.dateString]) {
      setShowEntry(true);
      setEditMode(false);
      setAnswers({
        ...defaultAnswers,
        ...records[day.dateString],
        emotion: records[day.dateString].emotion && EMO_KEYS.includes(records[day.dateString].emotion)
          ? records[day.dateString].emotion : safeEmotion
      });
    } else {
      setShowForm(true);
      setEditMode(false);
      setShowEntry(false);
      setAnswers({ ...defaultAnswers, emotion: safeEmotion });
    }
  };
```

Рисунок 3.13. Логіка відкриття збереженого запису або створення нової форми після вибору дати (DiaryScreen.js).

Форма нового запису має кілька полів для коротких відповідей - про настрій, досягнення, яскравий момент дня, вдячність тощо, а також поле для фото. Всі поля необов’язкові. Якщо вибрано іншу емоцію, фон і відтінки елементів адаптуються.

Після заповнення полів та натискання кнопки "Зберегти" дані запису зберігаються у локальному сховищі AsyncStorage з прив'язкою до дати й логіну користувача. При повторному виборі цієї дати запис відкриється у вигляді "листа". Якщо запису немає, користувач завжди може створити новий. Збереження працює так само і при редагуванні - натискання кнопки "Зберегти" оновлює дані для обраної дати.

```
const handleSave = async () => {
  if (!Object.values(answers).some(x => typeof x === 'string' ? x.trim() : x)) return;
  if (!currentLogin) return;
  const stored = await AsyncStorage.getItem(STORAGE_KEY);
  const allRecords = stored ? JSON.parse(stored) : {};
  const newEntry = { ...answers, date: selectedDate, login: currentLogin };
  allRecords[selectedDate] = newEntry;
  await AsyncStorage.setItem(STORAGE_KEY, JSON.stringify(allRecords));
```

Рисунок 3.14. Збереження нового або оновленого запису у щоденнику через AsyncStorage (DiaryScreen.js).

Після збереження введені дані формують структурований текст щоденника, який підтягується у вигляді готового листа для обраної дати. Усі поля залишаються опційними - якщо користувач заповнив лише останній блок (“А загалом...”), саме ця частина відобразиться у підсумковому тексті, якщо додано фото воно теж підтягується до листа.

```
<Text style={uiStyles.text}>
      {answers.feel ? `Сьогодні я почував(-ла) себе ${answers.feel}, і я ціную цей день таким, яким він є. А
      {answers.made ? `Я хочу нагадати собі, що сьогодні я зміг(-ла) ${answers.made} — і це вже причина пиша
      {answers.emotions ? `Найбільше емоцій мені подарувало ${answers.emotions}, ` : ''}
      {answers.smile ? `а посміхнутися змусило ${answers.smile}. Це зробило мій день кращим.\n` : ''}
      {answers.proud ? `А ще я пишаюсь собою за те, що ${answers.proud}. ` : ''}
      {answers.thanksSelf ? `Дякую собі за те, що я ${answers.thanksSelf}.\n` : ''}
```

Рисунок 3.15. Формування підсумкового тексту щоденника залежно від заповнених полів (DiaryScreen.js).

```
{answers.photo &&
  <Image source={{ uri: answers.photo }} style={styles.letterPhoto} resizeMode="cover" />
}
```

Рисунок 3.16. Додавання фотографії до запису: якщо додано фото, воно автоматично підтягується у фінальний вигляд листа (DiaryScreen.js).

3.7. Реалізація модулю психологічних тестів

Модуль психологічних тестів створений для швидкої й зручної самодіагностики. Всі параметри тестів винесені у конфігураційний файл testConfigs.js, де для кожного тесту задані питання, варіанти відповідей, схема нарахування балів і межі результатів (Рис. 3.17). Таким чином можна

додавати нові опитування без редагування основного коду. Стан екрана (градієнт, фон, колір) адаптується до обраної емоції.

```
export const TESTS_CONFIG = {
  depression: {
    title: "Тест на депресію (PHQ-9)",
    questions: [
      "Мене майже нічого не цікавить або не приносить радості.",
      "Я відчуваю пригнічення, депресію чи безнадійність.",
      "Я маю труднощі зі сном (занадто багато або мало сплю).",
      "Я відчуваю втому або брак енергії.",
      "Маю проблеми з апетитом (або їм забагато, або дуже мало).",
      "Відчуваю себе невдахою або думаю, що підвів(-ла) себе чи свою родину.",
      "Мені важко зосередитися на справах (наприклад, читати газету або дивитися телевізор).",
      "Я рухаюсь або говорю настільки повільно, що це помічають інші (або навпаки — занадто метушливий(-а)).",
      "У мене виникали думки, що краще б мене не було або бажання завдати собі шкоди.",
    ],
    scale: [
      "Зовсім ні",
      "Кілька днів",
      "Більше половини днів",
      "Майже щодня"
    ],
    scoreMap: [0, 1, 2, 3],
    results: [
      { min: 0, max: 4, level: "Відсутність депресії", desc: "Ознак депресії не виявлено." },
      { min: 5, max: 9, level: "Легка депресія", desc: "Можливі легкі ознаки депресії. Варто поспостерігати за
      { min: 10, max: 14, level: "Помірна депресія", desc: "Можливий розвиток депресії. Бажано звернутися до спе
      { min: 15, max: 19, level: "Виражена депресія", desc: "Високий рівень симптомів. Рекомендовано консультаці
      { min: 20, max: 27, level: "Дуже виражена депресія", desc: "Вкрай високий рівень симптомів. Необхідна фахо
```

Рисунок 3.17. Фрагмент конфігурації одного з тестів у testConfigs.js.

Під час проходження тесту кожне питання виводиться окремо з варіантами відповідей, які підсвічуються при виборі.

Після завершення тесту і натискання кнопки "Завершити", додаток автоматично підраховує бали, визначає рівень стану та коротко пояснює результат. Далі ці дані зберігаються у локальному сховищі AsyncStorage під унікальним ключем, який містить логін користувача та ідентифікатор тесту.

```
const calcScore = () => {
  let total = 0;
  answers.forEach((val, i) => {
    if (val == null) return;
    let idx = test.reverse && test.reverse.includes(i)
      ? test.scoreMap.length - 1 - val
      : val;
    total += test.scoreMap[idx];
  });
  return total;
};
```

Рисунок 3.18. Підрахунок загального балу за тестом із урахуванням зворотних питань (TestScreen.js)

Також була реалізована статистика проходжень тестів безпосередньо у головному меню модуля самодіагностики. На екрані вибору тесту користувач бачить іконку статистики у правому верхньому куті. При її натисканні відкривається модальне вікно з історією останніх проходжень для кожного тесту — депресії, тривоги та стресу.

У цьому вікні відображаються до трьох останніх результатів для кожного тесту. При завантаженні екрану через цикл підтягуються масиви результатів для кожного тесту з унікальними ключами типу SELFTEST_{login}_{testKey}, після чого історія записується у локальний state для відображення статистики.

```
useEffect(() => {
   let isActive = true;
   (async () => {
     const loginVal = await AsyncStorage.getItem('CURRENT_LOGIN');
     if (!loginVal) return;
     setLogin(loginVal);
     let data = {};
     for (let test of TESTS) {
       const item = await AsyncStorage.getItem(`SELFTEST_${loginVal}_${test.key}`);
       data[test.key] = item ? JSON.parse(item) : [];
     }
     if (isActive) setHistory(data);
     console.log("Loaded history for login:", loginVal, data);
   })();
   return () => { isActive = false; };
 }, [isFocused]);
```

Рисунок 3.19. Завантаження історії результатів проходження тестів для поточного користувача з локального сховища.

3.8 Реалізація модуля списку приємностей

Модуль списку приємностей розроблено для того, щоб користувачі могли швидко знаходити й керувати діями для покращення настрою, адаптованими до їхнього емоційного стану. Основний акцент зроблено на простоті та можливості персоналізації відповідно до принципів емпатичного дизайну.

Дані про приємності зберігаються в локальному сховищі AsyncStorage під унікальним ключем. Кожна приємність має поле назва, статус («Працює», «Потребує перевірки», «Не працює»), дата, та належність до користувача. Додавання нової приємності відбувається через простий модальний інтерфейс (Рис. 3.20).

```
const handleAdd = () => {
   const trimmed = newPleasure.trim();
   if (!trimmed || pleasures.includes(trimmed)) return;
   setPleasures(list => [...list, trimmed]);
   setRatings(r => ({ ...r, [trimmed]: 1 })); // нова — статус "Працює"
   setNewPleasure('');
 };
```

Рисунок 3.20. Додавання та збереження нової приємності

Користувач може редагувати чи видаляти приємності (Рис. 3.21).

```
// Зберігає всі зміни
  const handleSave = async () => {
    if (!currentLogin) return;
    const key = `PLEASURES_SETUP_${currentLogin}`;
    await AsyncStorage.setItem(key, JSON.stringify({ pleasures, ratings }));
    navigation.goBack();
  };

  // Змінює статус приємності по натисканню кнопки (по колу)
  const handleCycleStatus = (item) => {
    const current = ratings[item] || 1;
    const idx = STATUS.findIndex(s => s.key === current);
    const next = STATUS[(idx + 1) % STATUS.length].key;
    setRatings(r => ({ ...r, [item]: next }));
```

Рисунок 3.21. Збереження змін у списку приємностей і циклічна зміна статусу дії за допомогою кнопки.

Модуль адаптується до поточного емоційного стану користувача. Для позитивних емоцій (наприклад, радість) першою відкривається вкладка «Потребує перевірки», щоб мотивувати пробувати нові приємності. Для інших емоцій пріоритетно відображаються перевірені приємності зі статусом «Працює». Весь інтерфейс адаптується під вибрану емоцію.

```
    // Визначення емоції
    const em = getActiveEmotion && getActiveEmotion();
    setEmotion(em);

    // Вибір вкладки за емоцією
    const isPositive = em && ['радість'].includes(em.toLowerCase());
    setActiveFilter(isPositive ? 3 : 1);
  })();
  return () => { isActive = false; };
}, []);
```

Рисунок 3.22. Адаптація інтерфейсу під вибрану емоцію

Усі приємності відображаються у фільтрованому списку. Для зручності поруч із кожною категорією відображається кількість дій у цій категорії (Рис. 3.23).

```
// Підрахунок по категоріях
const counts = { 1: 0, 2: 0, 3: 0 };
pleasures.forEach(item => {
  const rate = ratings[item];
  if (counts[rate]) counts[rate]++;
  else if (rate) counts[rate] = 1;
});

// Фільтрований список
const filteredPleasures = pleasures.filter(item => ratings[item] === activeFilter);
```

Рисунок 3.23. Виведення кількості приємностей у кожній категорії

Інтерфейс модуля інтуїтивний, користувач може швидко змінити статус приємності, додати нову або видалити зайву. Усі ключові дії з приємностями виконуються в один клік.

ВИСНОВКИ

У рамках кваліфікаційної роботи створено прототип мобільного застосунку, спрямованого на підтримку психоемоційного благополуччя молоді, який інтегрує принципи емоційного дизайну, персоналізації та сучасні підходи до проектування користувацького досвіду (UX) і інтерфейсу (UI).

У першому розділі здійснено аналіз проблеми психоемоційної нестабільності молоді, оцінено наявні цифрові рішення та обґрунтовано необхідність розробки застосунку з адаптивним інтерфейсом, що враховує емоційний стан користувача. Встановлено, що індивідуалізований підхід до дизайну та сценаріїв взаємодії є ключовим для підвищення залученості й ефективності психоемоційної підтримки.

У другому розділі розроблено концепцію застосунку, центральним елементом якої є вибір емоційного стану як основа для персоналізації. Запропоновано систему адаптивного інтерфейсу, що використовує візуальні та текстові елементи, натхненні стилістикою мультфільму «Головоломка». Визначено основні функціональні модулі - реєстрація, вибір емоції, рефлексія, щоденник, самодіагностика та створення списку позитивних активностей.

У третьому розділі реалізовано прототип у середовищі React Native. Інтерфейс і логіка застосунку динамічно адаптуються до обраного емоційного стану, а для зберігання даних використано локальне сховище AsyncStorage, що забезпечує швидкодію без серверної інфраструктури. Розроблений прототип продемонстрував ефективність у вирішенні поставлених завдань і відповідає потребам молоді в доступних і адаптивних інструментах.

Перспективи подальшого розвитку включають інтеграцію серверної бази даних, розширення функціональних можливостей і поглиблене UX-дослідження в реальних умовах експлуатації. Мета роботи досягнута, а

отримані результати мають потенціал для практичного впровадження та вдосконалення цифрових рішень для психоемоційної підтримки.

Список використаних джерел:

1. VoxUkraine. Чи переживуть вони війну неушкодженими? Дослідження психічного здоров'я студентів київських вишів у листопаді 2023 року [Електронний ресурс]. – URL: https://voxukraine.org/chy-perezhyvut-vony-vijnu-neushkodzhenymy-doslidzhennya-psyhichnogo-zdorov-ya-studentiv-kyyivskyh-vyshiv-u-lystopadi-2023-roku
2. Психічне здоров'я та повсякденне життя студентів під час війни в Україні [Електронний ресурс] // Практична сімейна психологія. – 2023. – №4. – URL: https://uk.e-medjournal.com/index.php/psp/article/view/596?utm_source=chatgpt.com
3. U-Report Europe. Ментальне здоров'я молодих українців за кордоном: результати опитування, лютий 2024 року [Електронний ресурс]. – URL: https://europe.ureport.in/opinion/3705/
4. Cyr D., Head M., Larios H. Colour appeal in website design within and across cultures: A multi-method evaluation // International Journal of Human-Computer Studies. – 2010. – Т. 68, № 1. – С. 1–21. – URL: https://www.sciencedirect.com/science/article/abs/pii/S1071581918301824
5. Emotional Design [Електронний ресурс] // Interaction Design Foundation. – URL: https://www.interaction-design.org/literature/topics/emotional-design
6. Flow State in UX: Designing for Engagement [Електронний ресурс] // PeepalDesign. – URL: https://peepaldesign.com/flow-state-in-ux-designing-for-engagement/
7. Wüllner S. та ін. Mobile applications in adolescent psychotherapy during the COVID-19 pandemic: a systematic review // Frontiers in Public Health. – 2024. – Т. 12. – № статті 1345808. – URL: https://www.frontiersin.org/journals/public-health/articles/10.3389/fpubh.2024.1345808/full
8. В Машкіна, ВО Абрамов, ТІ Носенко, ЄВ Іваніченко.Навчально-методичний посібник до виконання і захисту кваліфікаційної роботи бакалавра спеціальностей 122 Комп'ютерні науки для здобувачів вищої освіти денної форми навчання, Київський столичний університет імені Бориса Грінченка. Київ, 2024.- 43 с.

9. Теоретичні та практичні аспекти використання математичних методів та інформаційних технологій в освіті й науці:моногр. / за заг. ред. О. Литвин. — К.: Київ. ун-т ім. Б. Грінченка,2021. — 332 с.
10. Яскевич, Владислав Олександрович (2023) *JavaScriptбібліотека React Навчальний посібник для студентів спеціальності Комп'ютерні науки* Київський університет імені Бориса Грінченка, Україна, Київ. https://elibrary.kubg.edu.ua/id/eprint/48587/
11. Designing Emotional UX: Lessons from Riley's Mind in Inside Out [Електронний ресурс] // Medium, Design Bootcamp. – URL: https://medium.com/design-bootcamp/designing-emotional-ux-lessons-from-rileys-mind-in-inside-out-a94a2b525061
12. Inside Out 2 (2024): Summary and Box Office Data [Електронний ресурс] // The Numbers. – URL: https://www.the-numbers.com/movie/Inside-Out-2-(2024)#tab=summary
13. How Calm Disrupted the Wellness Industry with a Smart App Marketing Strategy [Електронний ресурс] // Medium, Ronn Torossian. – URL: https://ronntorossian.medium.com/how-calm-disrupted-the-wellness-industry-with-a-smart-app-marketing-strategy-2063bca3f203
14. Katz A. Inside Out 2 Is Already Breaking Box Office Records for Pixar [Електронний ресурс] // TIME. – 2024. – URL: https://time.com/6989419/inside-out-2-box-office-pixar/
15. Jalali M., Azimi R., Neshati E. Combined Effect of Color and Shape on Cognitive Performance // Journal of Behavioral and Brain Science. – 2022. – Т. 12, № 8. – С. 372–383. – URL: https://www.researchgate.net/publication/362433031_Combined_Effect_of_Color_and_Shape_on_Cognitive_Performance
16. DynamicColorIOS [Електронний ресурс] // React Native Documentation. – URL: https://reactnative.dev/docs/dynamiccolorios
17. React Native library for image upload and cropping supporting both Android and iOS [Електронний ресурс] // Stack Overflow. – URL: https://stackoverflow.com/questions/45299287/react-native-library-for-image-upload-and-cropping-supporting-both-android-and
18. What Is React Native Async Storage and Why Should You Use It? [Електронний ресурс] // Monterail Blog. – URL: https://www.monterail.com/blog/what-is-react-native-async-storage

ДОДАТКИ