

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Розробка мобільного додатку з використанням технології
штучного інтелекту від Ореп AI для дослідження споживання їжі корінним
населенням Канади та США

Виконав(ла): студент(ка) 4 курсу, групи СП-42
спеціальності 121

«Інженерія програмного забезпечення»

(шифр і назва спеціальності)

		Штекля М. І.
	(підпис)	(прізвище та ініціали)
Керівник		Бойко І. В.
	(підпис)	(прізвище та ініціали)
Нормоконтроль		Стоянов Ю. М.
	(підпис)	(прізвище та ініціали)
Завідувач кафедри		Петрик М. Р.
	(підпис)	(прізвище та ініціали)
Рецензент		
	(підпис)	(прізвище та ініціали)

АНОТАЦІЯ

Розробка мобільного додатку з використанням технології штучного інтелекту від Open AI для дослідження споживання їжі корінним населенням Канади та США // Кваліфікаційна робота освітнього рівня "Бакалавр"

// Штекля Максим Іванович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра програмної інженерії, група СП-42 // Тернопіль, 2024 // с. – 54, рис. – 12, табл. – 0, кресл. – 0, бібліогр. – 14.

Ключові слова: мобільний додаток; штучний інтелект; споживання продуктів.

Дослідження, описане в роботі, концентрується на розробці та оцінці якості мобільної програми, яка має на меті допомогти дослідженню зібрати інформацію про споживання їжі корінним населенням Канади та США (індіанцями).

Загальна інформація про додаток: він використовує мікрофон та диктовку тексту, а далі обробку цієї диктовки і перетворення її у стандартизований формат за допомогою технологій штучного інтелекту від компанії OpenAI для збереження інформації. Диктовка може відбуватися будь-яким стилем мовлення, що дозволяє отримати лише потрібну інформацію для дослідження. Це має підвищити ефективність збору інформації від піддослідних, оскільки знімає з них потребу вручну заповнювати різні форми та бланки.

Воно визначає декілька ключових характеристик, які впливають на залучення користувачів та їх сприйняття програми. Згідно з результатами, важливими елементами для залучення користувачів та стимулювання їх участі є простота у використанні та точність обробки диктовки. Функціональні аспекти програми були визнані важливими, оскільки вона повинна мати корисні та практичні функції, які сприяють збору точних даних про

споживання їжі. Якщо програма спрощує задачу заповнення форм і зменшує ручну роботу, то це збільшує шанси на успіх у зборі даних для дослідження.

Дослідження також підтвердило, що користувачі сприймають досвід використання мобільної програми як корисний і практичний. Це означає, що користувачі не лише отримують задоволення від використання програми, але й вважають її корисною та цінною для досягнення своїх цілей щодо збору інформації про споживання їжі.

ABSTRACT

Mobile App Development using artificial intelligence technology from Open AI to study the food consumption of the indigenous population of Canada and the USA // Qualification work of the educational level "Bachelor" // Maksym Shteklia // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Software Engineering, Group CII-42 // Ternopil, 2024 // p. – 54, fig. – 0, tables – 0, references – 14, posters – 12.

Keywords: mobile application; Artificial Intelligence; consumption of products.

The study described in this paper focuses on the development and evaluation of a mobile application designed to assist in gathering information on food consumption by the indigenous populations of Canada and the United States (Indians).

General information about the application: it uses a microphone and voice dictation, which is then processed and converted into a standardized format using artificial intelligence technologies from OpenAI for information storage. Dictation can occur in any speech style, allowing only the necessary information for the study to be collected. This is expected to enhance the efficiency of data collection from subjects, as it eliminates the need for them to manually fill out various forms and documents.

It identifies several key characteristics that influence user engagement and their perception of the application. According to the results, important elements for attracting users and encouraging their participation include ease of use and the accuracy of dictation processing. The functional aspects of the application were deemed important, as it should have useful and practical features that facilitate the accurate collection of food consumption data. If the application simplifies the task of filling out forms and reduces manual work, it increases the chances of success in data collection for the study.

The study also confirmed that users perceive the experience of using the mobile

application as useful and practical. This means that users not only enjoy using the application but also find it helpful and valuable for achieving their goals related to collecting information on food consumption.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ.....	11
1.1 Характеристика типів мобільних додатків.....	11
1.2 Ознайомлення з проєктом.....	13
1.3 Dynamic Systems Development Method (DSDM) як концепція розробки додатку.....	15
РОЗДІЛ 2. ПРОЄКТУВАННЯ ДОДАТКУ ТА РОЗРОБКА.....	18
2.1 Проєктування.....	18
2.1.1 Життєвий цикл проєкту.....	18
2.1.2 Аналіз вимог до проєкту.....	18
2.1.3 Вибір технологій.....	19
2.1.4 Архітектура та процеси.....	21
2.2 Розробка.....	24
2.2.1 Дизайн.....	24
2.2.2 Серверна частина (backend).....	28
2.2.2.1 Маршрутизація.....	29
2.2.2.2 Контролери (Controllers).....	32
2.2.2.3 Сервіси (Service).....	34
2.2.2.4 Моделі.....	39
2.2.3 Користувацький інтерфейс (frontend).....	40
2.2.3.1 Вхідний файл App.tsx.....	41
2.2.3.2 Екран форми (FormScreens).....	42
2.3 Результати проєктування.....	46

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ.....	48
3.1 Роль центральної нервової системи в трудовій діяльності людини. 48	
3.2 Гігієнічні вимоги до організації та обладнання робочих місць з ВДТ	50
ВИСНОВОК.....	53
ПЕРЕЛІК ПОСИЛАНЬ.....	55

ВСТУП

Дослідження, описане в цій роботі, концентрується на розробці та оцінці якості мобільної програми, яка має на меті допомогти у зборі інформації про споживання їжі корінним населенням Канади та США (індіанцями). Цей проєкт виконується для компанії, яка спеціалізується на дослідженні печінкових захворювань, зокрема неалкогольної жирової хвороби печінки (НАЖХП), яка вражає значну частину населення, особливо серед корінних громад.

Неалкогольна жирова хвороба печінки (НАЖХП) є серйозним медичним станом, при якому жир накопичується в печінці у людей, які мало або зовсім не вживають алкоголю. Якщо це захворювання не лікувати, воно може призвести до важких наслідків, таких як цироз печінки, рак печінки, інсульт або гепатит. За даними дослідження, поширеність НАЖХП серед корінних народів Канади та США значно вища, ніж серед загального населення, що підкреслює необхідність спеціалізованих інтервенцій та досліджень у цій галузі [1].

Розроблена мобільна програма використовує сучасні технології, такі як мікрофон та голосову диктовку, яка потім обробляється і перетворюється у стандартизований формат за допомогою штучного інтелекту від OpenAI для збереження інформації. Користувачі можуть диктувати інформацію будь-яким стилем мовлення, що дозволяє збирати лише необхідні дані для дослідження. Це суттєво підвищує ефективність збору інформації від піддослідних, знімаючи потребу у ручному заповненні численних форм та документів.

Методологія дослідження включає кілька етапів. Спочатку було проведено аналіз вимог користувачів та розроблено прототип додатка. Потім відбувалося тестування прототипу з метою виявлення та усунення можливих недоліків. Учасники дослідження надавали зворотний зв'язок щодо зручності використання та функціональності додатка, що дозволило внести необхідні зміни перед запуском основного етапу збору даних.

Ключовою метою даного дослідження є підвищення ефективності та точності збору даних про споживання їжі серед корінного населення. Це, у свою чергу, дозволить більш точно оцінити масштаби поширення НАЖХП та розробити ефективніші стратегії для її профілактики та лікування. Залучення корінних громад до дослідження є важливим аспектом, оскільки їх активна участь сприяє отриманню більш достовірних та репрезентативних даних.

Крім того, використання мобільного додатка значно спрощує процес збору даних, знімаючи з учасників необхідність у ручному заповненні численних форм. Це також зменшує ймовірність помилок, пов'язаних з людським фактором, що є критично важливим для точності дослідження. Додаток дозволяє здійснювати диктовку будь-яким стилем мовлення, що робить його зручним та доступним для користувачів з різним рівнем технічних навичок.

Очікується, що результати дослідження матимуть значний вплив на медичну практику та політику охорони здоров'я. Отримані дані дозволять розробити більш точні та ефективні рекомендації щодо профілактики та лікування НАЖХП серед корінних народів. Крім того, успішне впровадження мобільного додатка може стати основою для подібних ініціатив в інших сферах медичних досліджень, де необхідно збирати велику кількість даних від різноманітних груп населення.

Важливим аспектом розробки мобільного додатку є забезпечення його інклюзивності та доступності для всіх користувачів, незалежно від їхнього рівня технічної підготовки та мовних особливостей. Для цього програма підтримує багатомовність та адаптивний інтерфейс, що дозволяє користувачам налаштовувати її відповідно до своїх потреб. Це забезпечує ширший доступ до програми та сприяє залученню більшої кількості учасників до дослідження.

Додатково, мобільний додаток розроблений з урахуванням конфіденційності та безпеки даних користувачів. Усі зібрані дані шифруються та зберігаються на захищених серверах, що мінімізує ризики несанкціонованого доступу та втрати інформації. Це особливо важливо в контексті медичних досліджень, де конфіденційність та безпека даних є

пріоритетом.

Окрім цього, дослідження також розглядає вплив традиційних методів харчування корінних народів, таких як самостійно здобута дичина (полювання) та приготування їжі на відкритому вогні, на стан здоров'я та поширеність НАЖХП. Це дозволяє врахувати культурні особливості та традиції корінних народів, що може мати важливе значення для розробки ефективних стратегій профілактики та лікування захворювань печінки в цих громадах.

В цілому, ця робота має на меті не лише покращити якість збору даних, але й сприяти підвищенню обізнаності про НАЖХП та залученню корінних громад до активної участі в медичних дослідженнях. Досвід і висновки цього дослідження можуть слугувати вказівками для майбутніх проєктів розробки додатків, спрямованих на покращення збору даних для медичних досліджень та інших сфер.

РОЗДІЛ 1. ПРОЄКТУВАННЯ МОБІЛЬНОГО ДОДАТКУ

1.1 Характеристика типів мобільних додатків

Застосування інформаційних технологій створює позитивний вплив у багатьох аспектах, таких як полегшення спілкування та соціалізації в житті людини, прискорення доступу до інформації та зменшення інформаційної асиметрії, а також економія часу та коштів у багатьох роботах у повсякденному житті. З іншого боку, це може мати негативні наслідки, такі як технологічна залежність, порушення конфіденційності та захисту персональних даних, а також створення цифрового розриву. Одним із найпоширеніших явищ збільшення як рівня проникнення, так і частоти та тривалості використання технологій є можливості, які вона пропонує споживачам щодо пошуку, доступу та використання інформації. Завдяки технологічному розвитку та впровадженню мобільні додатки стали одним із інструментів повсякденного життя, що мають два типи корисного впливу: функціональний та емоційний.

Мобільні програми класифікуються залежно від їх природи на дві категорії: гедонічні/експериментальні та утилітарні/цілеспрямовані. Зазвичай існує три альтернативних підходи до мобільного додатку: утилітарна орієнтація, гедонічна орієнтація та гібрид. Наприклад, банківські додатки та додатки для продуктивності підтримують утилітарні/цільові результати, тоді як ігри та додатки для соціальних мереж підтримують гедонічні/експериментальні результати. За допомогою утилітарних програм споживачі здебільшого шукають інформацію, тобто ці програми орієнтовані на завдання, раціональні та вважаються пов'язаними з роботою, тоді як гедонічні мобільні програми використовуються споживачами, щоб провести час у таких заходах, як покупки, розваги та ігри.

Багато мобільних додатків зосереджені на утилітарних аспектах, таких як функціональність і обмін інформацією, особливо як інструмент

онлайн-навчання для ширшої аудиторії. Деякі інші орієнтовані переважно на естетику та взаємодію. Згідно з попередніми дослідженнями, сприйняття користувачами цінностей, таких як емоційні та утилітарні цінності, впливає на задоволеність користувачів, прийняття користувачами та намір оновити, поведінкову взаємодію з мобільними додатками та намір здійснити покупки.

Гібридний підхід також можна використовувати для мобільних додатків, поєднуючи гедонічні та утилітарні цінності. Наприклад, участь в онлайн-навчанні може бути представлена як багатовимірна конструкція, яка включає поведінкові, емоційні та когнітивні елементи. Поведінкова активність пов'язана з навчальною діяльністю та взаємодією з учнями та інструкторами. Афективна залученість стосується реакції на курс або лектора та афектів вищого порядку, таких як надія, гордість або почуття впевненості.

Когнітивне залучення охоплює мислення, планування та саморегуляцію для свого навчання та сприйманого розуміння, знання та майстерності. Намір мобільного залучення визначається через мотивацію, сприйняту цінність і задоволення, які можуть бути пов'язані з афективними елементами. Задоволеність користувачів залежить від високої зручності використання програми. Висока зручність використання призводить до високої задоволеності користувачів, що, своєю чергою, призводить до високого наміру залучення. Крім того, було виявлено, що соціальні мотивації тісно пов'язані з наміром продовжувати співпрацю. Споживачі взаємодіють із мобільними додатками на високому рівні, коли вони мають спільний та інтерактивний досвід роботи зі своїми улюбленими додатками. Зручність часу, інтерактивність і сумісність позитивно впливають на залучення мобільних додатків.

Максимізація залученості учасників є ключовим фактором успіху дослідження. Чим більше людей візьме участь у розробці та використанні додатку, тим більш повною та точною буде інформація, отримана в ході дослідження. Це дозволить виявити відмінності у харчуванні між сучасними людьми, які купують продукти в супермаркетах, та тими, хто змушений полювати та готувати їжу на відкритому вогнищі.

1.2 Ознайомлення з проєктом

Дослідження "Ookwin Study" було ініційовано з метою розгляду проблеми неалкогольної жирової хвороби печінки (НАЖХП) серед корінного населення Канади. Це спільний проєкт кафедри гепатології Університету Манітоби, Four Arrows Regional Health Authority (який представляє чотири громади Island Lake) та клініки у Вінклері, Манітоба.

Команда дослідників, яка включає лікарів, дієтологів, медсестер та корінних координаторів, регулярно подорожує до північних громад, щоб краще зрозуміти проблему НАЖХП. Однією з ключових осіб команди є доктор Джулія Уханов, доцент кафедри гепатології Університету Манітоби [3] та головний дослідник Ookwin Study [1].

Дослідження зосереджене на вивченні поширеності та частоти НАЖХП, стану, при якому жир накопичується в печінці у людей, які мало або зовсім не вживають алкоголю. Якщо його не лікувати, НАЖХП може призвести до важких захворювань печінки, таких як цироз, рак, інсульт або гепатит. Доктор Уханов зазначає, що близько 30% канадців страждають на НАЖХП, причому ця проблема набагато більш поширена серед корінного населення. Тому важливо зрозуміти масштаб проблеми перед тим, як розробляти інтервенції або плани дій [1].

Для участі в дослідженні було залучено 1000 волонтерів з регіону Island Lake та 1000 людей з Вінклера. Протягом п'яти років вони проходять обстеження печінки та контрольні огляди. Ті, у кого діагностують НАЖХП, отримують регулярний моніторинг та працюють з командою для управління захворюванням. Ті, хто не має НАЖХП, проходять УЗД через п'ять років, коли дослідження закінчиться.

Дослідники прагнуть описати захворювання та його поширеність у обох популяціях, щоб розробити інтервенції, специфічні для кожної громади. Участь

корінних координаторів є ключовою для успіху дослідження, оскільки вони допомагають перекладати, відповідати на запитання, заохочувати людей до участі та допомагають у налаштуванні клінік.

Залучення корінних координаторів та волонтерів значно покращує результати дослідження, оскільки це дозволяє встановити довірчі відносини між дослідниками та громадами. Це також допомагає виявити та вирішити напругу, пов'язану з колонізацією, що сприяє автентичному обміну рішеннями для подолання цих проблем.

Раннє виявлення та втручання можуть значно покращити прогноз для пацієнтів з НАЖХП. Зміни в дієті можуть зменшити накопичення жиру в печінці, що, в свою чергу, може зменшити ризик розвитку важких захворювань печінки. Дослідження також підкреслює важливість забезпечення доступу до здорової їжі та підтримки здорових харчових звичок у віддалених північних громадах.

Фінансування дослідження було надано за рахунок гранту у розмірі 2 мільйонів доларів від Канадського інституту досліджень здоров'я. Отримані результати належать громадам відповідно до принципу ОСАР (володіння, контроль, доступ та володіння). Це означає, що громади мають право власності та контроль над своїми даними, що підвищує їхню довіру та участь у дослідженні.

Однією з важливих складових дослідження є активна участь громад у процесі збору даних та моніторингу результатів. Наприклад, Байрон Біарді з Four Arrows Regional Health Authority, який є координатором з безпеки харчових продуктів, зазначає, що залучення місцевих координаторів та волонтерів значно покращує результати дослідження. Це дозволяє встановити довірчі відносини між дослідниками та громадами, що сприяє ефективнішому збору даних та впровадженню інтервенцій.

Дослідження також спрямоване на вивчення традиційних продуктів харчування, які є здоровими та мають культурне значення для корінних народів. Це включає рибу, м'ясо, гриби та березовий сироп. Дослідники разом з

громадами обмінюються знаннями про ці продукти, що сприяє збереженню традицій та покращенню харчових звичок.

Дослідження "Ookwin Study" є прикладом успішної співпраці між дослідниками та корінними громадами. Воно демонструє важливість врахування культурних та соціальних аспектів при розробці медичних інтервенцій. Використання мобільних додатків, таких як розроблений компанією Refuah Solutions Ltd. [4], може значно спростити процес збору даних та збільшити кількість учасників, що сприятиме більш точним та репрезентативним результатам дослідження.

1.3 Dynamic Systems Development Method (DSDM) як концепція розробки додатку

Однією з поширених проблем під час розробки програмного забезпечення або додатків є те, що оцінка їх кінцевими користувачами часто відбувається після завершення розробки, а не в процесі проєктування. Внесення змін на пізніх етапах розробки вимагає значних витрат і може бути неоптимальним з точки зору зручності використання продукту. Тому було прийнято рішення слідувати ітераційному процесу проєктування, створивши прототипну версію програми, яку потім оцінювали за допомогою тестування на зручність використання, а результати враховувалися у вдосконаленій версії перед розгортанням програми.

Цей підхід відповідає концепції DSDM (Dynamic Systems Development Method [5]) як керівництва для розробки програмного забезпечення. Одними з важливих переваг цього методу є:

- орієнтація на бізнес потреби;
- вчасна розробка проєкту;
- співпрацювання команд;
- ітераційна розробка, що включає відгуки користувачів.

- безкомпромісна якість
- демонстрація контролю

Основні принципи DSDM включають залучення користувачів, розширення можливостей команди проєкту, часті релізи, вирішення поточних бізнес-потреб, ітераційну та поетапну розробку, можливість скасування змін, фіксацію масштабу високого рівня до початку проєкту, тестування зручності використання протягом усього життєвого циклу, а також ефективне спілкування. Розробка програми потребує зворотного зв'язку як від фахівців, так і від кінцевих користувачів. Для отримання зворотного зв'язку від кінцевих користувачів було застосовано тестування на зручність використання під час розробки.

DSDM поділяє проєкти на три фази: передпроєктну, життєвий цикл проєкту та післяпроєктну. На рисунку 1.1 показано процес розробки програми.

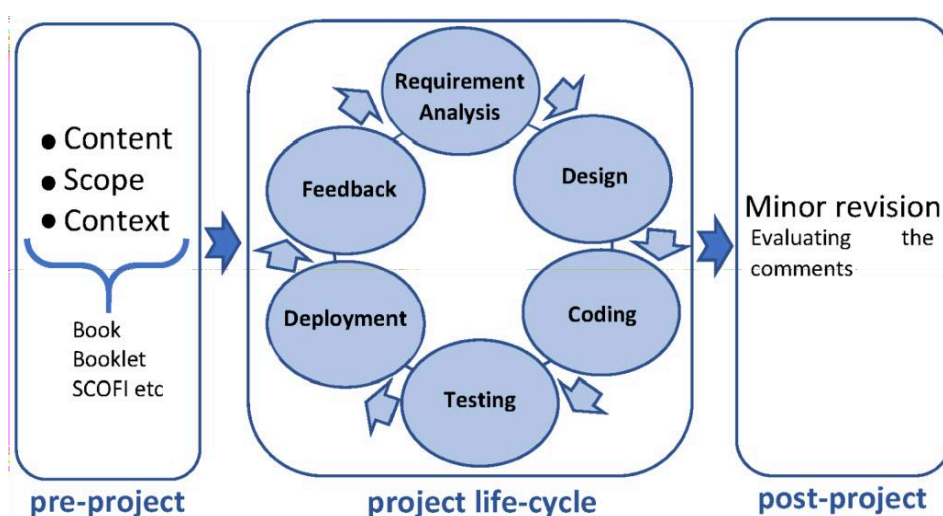


Рисунок 1.1 – Процес реалізації проєкту

На першому кроці було визначено основну структуру програми, її обсяг, контекст і вміст. Метою програми є спрощення процесу збору даних для дослідження неалкогольної жирової хвороби печінки (НАЖХП) серед корінного населення Канади та США. Контекст програми полягає в тому, щоб зробити процес збору даних більш зручним та доступним для користувачів, що сприятиме залученню більшої кількості учасників до дослідження.

Для контенту використовували інструкції, керівництва та відеоматеріали, які допомагають користувачам правильно заповнювати форми та надавати необхідну інформацію. Ця інформація була поєднана з простими інструкціями та прикладами, щоб користувачі могли легко орієнтуватися в додатку та зрозуміти, як правильно використовувати його функції.

Було створено контент, який підтримується покроковими інструкціями, що показують, які дані потрібно вводити на кожному етапі – від збору первинної інформації до завершення процесу збору даних. Цей вміст у додатку складається з трьох основних етапів (збір даних, перевірка даних, надсилання даних) та різних підетапів.

Також було реалізовано ідеї для підвищення продуктивності використання програми. Спонсори дослідження виплачують грошову компенсацію за правильне та своєчасне заповнення форми, що сприяє додатковій мотивації заповнення форми та прийнятті участі в дослідженні.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ДОДАТКУ ТА РОЗРОБКА

2.1 Проєктування

Проєктування програмного продукту є процесом створення детального плану реалізації, який буде використовуватися під час подальшої розробки. Основною метою цього етапу є забезпечення відповідності інформаційної системи попередньо сформованим вимогам та виконання нею своїх функцій відповідно до заданих параметрів. На основі описаних у попередньому розділі пунктів формується вектор потенційних технологій та шляхів для реалізації ідеї.

2.1.1 Життєвий цикл проєкту

Життєвий цикл проєкту включає шість основних етапів: аналіз вимог, проєктування, розробку коду, тестування, розгортання та збір відгуків. Кожен з цих етапів відіграє важливу роль у створенні якісного та ефективного мобільного додатку, забезпечуючи його відповідність потребам користувачів та вимогам бізнесу.

2.1.2 Аналіз вимог до проєкту

В першу чергу була проведена онлайн-зустріч з представниками спонсорської компанії. Мета першого мітингу заключалась у визначенні вимог, очікуваних від розробки, обговорення та пропозиції щодо можливих варіантів вирішення задач. Результатом обговорення стало рішення розробки мобільного додатку та виявлення наступних функціональних вимог:

- міграція бази даних з учасниками дослідження
- авторизація користувача в системі

- створення мобільного додатку
- створення форми для заповнення піддослідним
- функція голосового запису з подальшою обробкою запису та конвертуванням у формат потрібний для дослідження
- функція завантаження фотографій “до” та “після” прийому їжі
- відео з інструктажем вимірювання порцій

2.1.3 Вибір технологій

Наступним кроком було вирішенням технічної складової проєкту. Frontend і backend є двома найважливішими аспектами будь-якої програми. Інтерфейс (frontend) – це те, що бачать користувачі та містить візуальні елементи, як-от кнопки, прапорці, графіку та текстові повідомлення. Це дозволяє користувачам взаємодіяти з програмою. Сервер (backend) — це дані та інфраструктура, завдяки яким програма працює. Він зберігає та обробляє дані програми для користувачів [6].

Для розробки frontend частини було прийняте рішення використувати React Native, що надало змогу написати один код, який буде скомпільовано та портовано на IOS та Android девайси, без потреби писати рішний код під кожен з вищеперечислених платформ.



Рис. 2.1 – Логотип React Native

React Native — це платформа програмного забезпечення інтерфейсу користувача з відкритим кодом, створена Meta Platforms, Inc. Вона використовується для розробки додатків для Android, Android TV, iOS, macOS, tvOS, Web, Windows і UWP, дозволяючи розробникам використовувати рамку React разом із власні можливості платформи. Він використовується для розробки додатків Android та iOS у Facebook, Microsoft і Shopify. Він також використовується для розробки програм віртуальної реальності в Oculus [7].

Для backend частини було вирішено використовувати фреймворк Laravel на мові програмування PHP у зв'язці з базою даних MySQL. Перевагою даного фреймворку є грамотна архітектура проєкту та велика кількість бібліотек. “Laravel — це фреймворк веб-додатків із виразним елегантним синтаксисом. Ми вже заклали фундамент, дозволяючи вам творити, не переймаючись дрібницями.” [8]



Рис. 2.2 – Логотип Laravel

Laravel — безкоштовний, з відкритим кодом PHP-фреймворк, створений Тейлором Отвелом (англ. Taylor Otwell) і призначений для розробки вебдодатків відповідно до шаблону model–view–controller (MVC) [9].

Для обробки голосової диктовки та пост-обробки даних було вирішено використовувати технологію штучного інтелекту від компанії OpenAI, а саме модель gpt-3.5-turbo, оскільки за невелику ціну отримується потужність та швидкість.



Рис. 2.2 – Логотип OpenAI (ChatGPT)

Порóджувальний попередньо тренóваний трансфóрмер 3 (англ. Generative Pre-trained Transformer 3, GPT-3) — це авторегресійна модель мови, яка використовує глибоке навчання, щоби виробляти текст, подібний до людського. Вона є мовною передбачувальною моделлю третього покоління в серії GPT-п, створеній OpenAI, лабораторією досліджень штучного інтелекту в Сан-Франциско. Повна версія GPT-3 має ємність у 175 мільярдів параметрів машинного навчання. GPT-3, яку було представлено в травні 2020 року і яка перебуває в бета-тестуванні станом на липень 2020 року, є частиною тенденції попереднього тренування представлень мови в системах обробки природної мови (ОПМ). Перед випуском GPT-3 найбільшою мовною моделлю була Turing NLG Microsoft, представлена в лютому 2020 року, з ємністю в 17 мільярдів параметрів, або менш ніж 10 % у порівнянні з GPT-3. [10]

2.1.4 Архітектура та процеси

Додаток було побудовано на основі двох ключових технологій: React Native та Laravel. React Native був використаний як основний інструмент для створення користувацького інтерфейсу та відображення інформації на мобільних пристроях. Завдяки React Native вдалося забезпечити кросплатформенну розробку додатку,

що дозволяє запускати одну і ту ж саму версію додатку як на iOS, так і на Android, що значно спрощує підтримку та оновлення програми.

Laravel, у свою чергу, був використаний для розробки бекенду додатку. Вся комунікація між користувачем та додатком здійснюється через API, створене за допомогою Laravel. Це дозволяє розділити функціональність додатку на дві незалежні частини: фронтенд (користувацький інтерфейс) та бекенд (серверна логіка та управління даними). Такий підхід має кілька важливих переваг.

React Native забезпечує швидкий і зручний інтерфейс для користувачів, дозволяючи створювати динамічні та інтерактивні елементи. Ця технологія надає можливість використовувати одні і ті ж компоненти для різних платформ, що забезпечує однорідність інтерфейсу та зручність для користувачів. Всі елементи додатку, включаючи форми для введення даних, інтерфейси для перегляду інформації та навігацію, були створені за допомогою React Native.

Laravel був обраний для створення бекенд-частини додатку через його гнучкість, надійність та масштабованість. Вся логіка додатку, включаючи управління користувачами, обробку та зберігання даних, реалізована на базі Laravel. Основна функція бекенду - це обробка запитів від користувачів через API, яке дозволяє взаємодіяти з додатком у реальному часі.

Вся комунікація між фронтендом та бекендом здійснюється через API, створене на базі Laravel. Це дозволяє забезпечити високу швидкість обробки запитів та гнучкість у розробці нових функцій. API відповідає за обробку даних, збереження інформації в базі даних, а також за надання необхідної інформації фронтенду для відображення.

Для зберігання даних використовується база даних MySQL. Laravel надає можливість легко інтегруватися з MySQL, що забезпечує надійне та ефективне управління даними. Використання каркасу сутностей дозволяє при необхідності інтегрувати додаток з іншими базами даних, що робить додаток більш гнучким та масштабованим.

Користувачі додатку можуть заповнювати форми, що сприяють покращенню якості зібраних даних. Вся ця інформація передається на сервер

через API, де обробляється та зберігається в базі даних. Це дозволяє забезпечити точність та актуальність даних, зібраних у ході дослідження.

Проектування додатку з використанням React Native та Laravel дозволило створити гнучку, надійну та масштабовану систему, яка забезпечує зручний інтерфейс для користувачів та ефективне управління даними. Розділення функцій між фронтендом та бекендом через API забезпечує високу швидкість обробки запитів та легкість у розширенні функціональності додатку у майбутньому.

Сам процес роботи додатку виглядає наступним чином:

1. Запис диктовки: Користувач, використовуючи додаток на базі React Native, записує голосову диктовку свого прийому їжі. Інтерфейс додатку забезпечує зручний спосіб для запису та управління диктовками.
2. Передача даних на сервер: Після завершення запису диктовка передається на сервер через API, розроблене на базі Laravel. Сервер обробляє цей запит і зберігає аудіозапис у базі даних.
3. Обробка даних за допомогою OpenAI: Laravel API здійснює запит до API від OpenAI для обробки запису голосу. Запис передається до OpenAI, де здійснюється транскрипція голосу у текстовий формат.
4. Отримання транскрипту: Після обробки OpenAI повертає транскрипт запису голосу у текстовому форматі. Laravel зберігає цей транскрипт у базі даних.
5. Форматування даних: Laravel робить ще один запит до API OpenAI, у якому просить відформатувати отримані дані відповідно до заданого формату. Це дозволяє позбутися зайвої інформації та структурувати дані у зручному вигляді.
6. Перевірка та редагування: Відформатовані дані повертаються користувачу через інтерфейс додатку. Користувач перевіряє їх на помилки та, за необхідності, редагує інформацію.
7. Підтвердження та збереження: Після перевірки та редагування користувач підтверджує правильність введених даних. Відформатовані дані зберігаються на сервері.

Ця архітектура роботи додатку була ретельно обговорена з керівництвом

спонсорської компанії. Після детального аналізу та консультації, було досягнуто згоди щодо її доцільності та ефективності. Керівництво спонсорської компанії уважно розглянуло запропоновані рішення та технології, включаючи використання React Native для фронтенду та Laravel для бекенду, а також механізми взаємодії через API.

Після детального розгляду та обговорення всіх аспектів архітектури, включаючи її потенційні переваги та ризики, керівництво надало свою підтримку та офіційно затвердило запропоновану архітектуру. Було отримано всі необхідні дозволи на початок розробки, що підтвердило повну готовність до реалізації проекту згідно з узгодженим планом.

Такий підхід забезпечив впевненість у тому, що архітектура відповідає стратегічним цілям спонсорської компанії та вимогам кінцевих користувачів, що в свою чергу дозволяє розпочати розробку додатку з чітким розумінням його майбутніх етапів і кінцевого результату.

2.2 Розробка

2.2.1 Дизайн

Процес розробки дизайну для додатку був ретельно спланований і реалізований з урахуванням потреб користувачів та цілей проєкту. Основною метою було створення інтуїтивно зрозумілого та візуально привабливого інтерфейсу, який забезпечить комфортну взаємодію з додатком на всіх пристроях та не буде складним для різних груп лиць з різною ознайомленістю із сучасними технологіями.

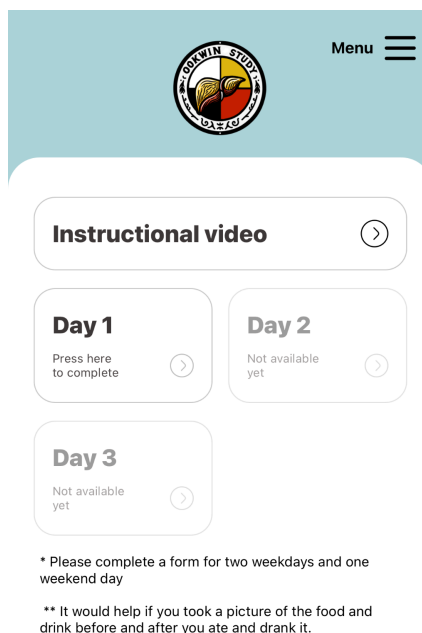


Рис. 2.3 – Головний екран

На головному екрані (рис. 2.3) користувачі бачать п'ять основних кнопок, які дозволяють швидко переходити до різних розділів додатку. Інтерфейс містить такі основні елементи:

- логотип Ookwin Study розташований у верхній частині екрану та виконує роль ідентифікації додатку;
- на головному екрані розташовані кнопки, що дозволяють переходити до різних розділів додатку, таких як “Інструкційне відео” та “Дні заповнення форм”;
- нижче основний екран містить кнопки для навігації до різних днів заповнення форм (Day 1, Day 2, Day 3), що дозволяє користувачам зручно заповнювати необхідну інформацію по мірі потреб.

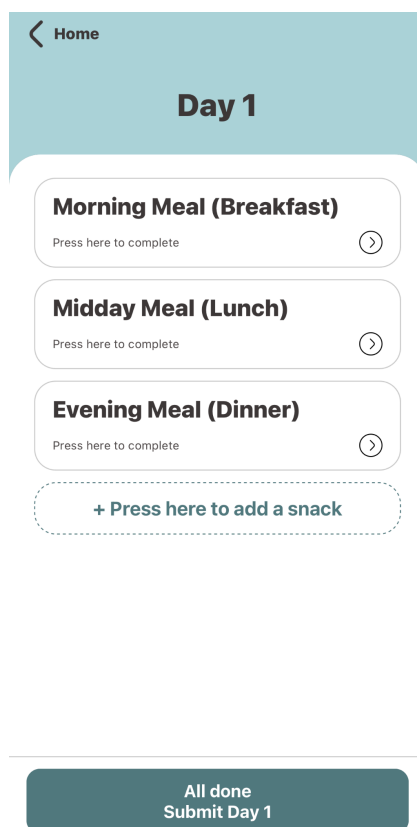


Рис. 2.4 – Екран “День”

При переході в один із доступних днів, відображається екран “День” (рис. 2.4) користувачі бачать 3 кнопки, які дозволяють швидко переходити до різних прийомів їжі. Інтерфейс містить такі основні елементи:

- інформацію про номер дня (День 1, День 2, День 3);
- кнопки вибору пори прийому їжі;
- кнопка для додавання снеку (ще одного прийому їжі)
- кнопка відправки форми при завершенні заповнення форми за один із днів.

< Day 1

1/3

Tell us what you ate for breakfast

Please include as many details as possible, such as what you ate, how it was prepared (cooked, fried, etc), whether there were any leftovers (25%, 50%, etc. leftover), and if you added sugar, etc. Also, include information about any beverages and alcohol consumed

[How to measure your food](#)

Tap to Start Recording audio

Skip recording Proceed audio

Рис. 2.5 – Екран “Форма”

На екрані форми (рис. 2.5) користувачі бачать такі елементи інтерфейсу:

- прогрес форми (3 етапи);
- інструкції що саме потрібно зазначити в голосову записі;
- кнопка запису аудіо;
- кнопки пропуску запису аудіо та обробки аудіо.

01:53 43%

< Day 1 2 / 3

Consumed at:

Jun 20, 2024 1:53am

Food

+ Tap here to add photos

(Add images before and after the meal)

Toasts

mayo	15 grams	Remove
red fish	12 grams	Remove

Edit dish Remove Dish

+ Add food

☐ No food

Beverages

To add a beverage, click the button below.

< Go back Save form

Рис. 2.6 – Екран заповненої форми

На екрані заповненої форми (рис. 2.6) користувачі бачать введені дані та можуть їх редагувати. Також є можливість завантажити фото “до” та “після” прийому їжі.

2.2.2 Серверна частина (backend)

Використання бекенду для реалізації API є критично важливим аспектом сучасних додатків, що забезпечує ефективну взаємодію між користувачем і серверною частиною системи. API забезпечило обробку всіх запитів від фронтенд-частини додатку, виконаної на React Native, і дозволяє взаємодіяти з

базою даних та сторонніми сервісами. Структура API включає наступні компоненти:

- Маршрутизація: Всі запити до API маршрутизуються через Laravel, що дозволяє легко керувати маршрутизацією та забезпечувати належну обробку запитів.
- Контролери: Контролери обробляють вхідні запити, взаємодіють з моделями та повертають відповіді користувачам. Це забезпечує чистий та організований код.
- Моделі: Моделі використовуються для взаємодії з базою даних, що дозволяє легко маніпулювати даними та забезпечувати їх цілісність.
- Сервіси: Laravel надає можливість створення сервісів для обробки логіки додатку, що забезпечує гнучкість та можливість повторного використання коду.

2.2.2.1 Маршрутизація

Маршрутизація у Laravel є одним з ключових аспектів, що дозволяє контролювати потік запитів до додатку. Вона забезпечує відповідність між URL-запитами і відповідними контролерами та методами, які їх обробляють. В цьому проєкті маршрутизація реалізована у файлі `routes/api.php`.

```

3  use App\Http\Controllers\Api\DayController;
4  use App\Http\Controllers\Api\MealController;
5  use App\Http\Controllers\Api\AuthController;
6  use App\Http\Controllers\Api\UserController;
7  use Illuminate\Support\Facades\Route;
8
9  Route::post( uri: 'login', [AuthController::class, 'login']);
10
11 Route::group(['middleware' => 'auth:sanctum'], function () {
12     Route::get( uri: 'users/me', [UserController::class, 'me']);
13     Route::put( uri: 'users/update-fcm', [UserController::class, 'updateFcm']);
14
15     Route::post( uri: 'submit-day/{day}', [DayController::class, 'submitDay']);
16     Route::resource( name: 'days', controller: DayController::class)->only(['index', 'show']);
17     Route::post( uri: 'proceed-audio', [MealController::class, 'proceedAudio']);
18     Route::post( uri: 'create-meal', [MealController::class, 'createMeal']);
19     Route::post( uri: 'confirm-meal/{meal}', [MealController::class, 'confirmMeal']);
20     Route::put( uri: 'user/notifications/enable', [UserController::class, 'enableNotifications']);
21
22     Route::middleware( middleware: 'admin')->group(function () {
23         Route::resource( name: 'users', controller: UserController::class);
24     });
25 });

```

Рис. 2.7 – Реалізація маршрутизації

Основні пункти маршрутизації:

- Аутентифікація: Маршрут для аутентифікації користувачів. Запити до цього маршруту обробляються контролером, що відповідає за вхід користувачів у систему;
- Група маршрутів з middleware: Всі маршрути в певній групі захищені middleware, що означає, що до них можуть звертатися тільки аутентифіковані користувачі. Це забезпечує безпеку додатку, дозволяючи доступ лише авторизованим користувачам;
- Маршрути користувачів:
 - Отримання інформації про поточного користувача;
 - Оновлення токена для сповіщень;
 - Увімкнення повідомлень для користувача;
 - Маршрути для роботи з днямиж
- Відправка даних за конкретний день;
- Перегляд списку днів та деталей конкретного дня;
- Маршрути для роботи з прийомами їжі;
- Обробка аудіодиктовки;
- Створення запису про прийом їжі;
- Підтвердження даних про прийом їжі;
- Адміністративні маршрути. Всі маршрути в цій групі захищені middleware, що означає, що до них мають доступ лише користувачі з правами адміністратора.

Маршрутизація у Laravel забезпечує ефективну організацію та обробку запитів, дозволяючи чітко розподіляти функції між різними контролерами та методами.

2.2.2.2 Контроллери (Controllers)

Контроллери у Laravel є основними компонентами, що обробляють запити користувачів, взаємодіють з моделями та повертають відповіді. Вони забезпечують логіку додатку та виконують різні операції, такі як створення, оновлення, обробка даних та інше. Контроллери дозволяють організувати код додатку у більш чистий та структурований спосіб. Основним паттерном фреймворку Laravel є MVC (Model - View - Controller), що дозволяє підтримувати правильну та зрозумілу архітектуру. Завдяки перенесенню всієї бізнес-логіки в сервіси (Service), в контроллерах було дотримано принципу SRP (принцип єдиної відповідальності), де контроллер відповідає за виконання дії в сервісі та виведенню результатів у форматі JSON, який вважається одним із основних форматів, що використовуються при побудові API.

```

15 class MealController extends Controller
16 {
17     public function __construct(
18         private readonly MealService $mealService
19     ) {}
20
21     public function proceedAudio(AudioRequest $request)
22     {
23         if ($request->hasFile( key: 'audio')) {
24             return response()->json([
25                 'data' => $this->mealService->proceedMealDetailsByAudio($request->validated())
26             ]);
27         }
28
29         return response()->json(['message' => 'No file uploaded.'], status: Response::HTTP_BAD_REQUEST);
30     }
31
32     public function createMeal(CreateMealRequest $request)
33     {
34         return response()->json($this->mealService->createOrConfirmMeal($request->validated()));
35     }
36
37     public function confirmMeal(ConfirmMealRequest $request, Meal $meal)
38     {
39         return response()->json($this->mealService->createOrConfirmMeal($request->validated(), $meal));
40     }
41 }

```

Рис. 2.8 – MealController.php

Одним з основних контролерів в проєкті Ookwin є MealController.php (рис 2.8). Він забезпечує запитів з даними про спожиту їжу, введених користувачем. Першим методом є construct, який в свою чергу ініціює сервіс MealService, в

якому містяться основні методи для обробки даних. Прикладом методу обробки та виводу даних може слугувати наступний метод `proceedAudio`, який забезпечує звернення до сервісу, який обробляє запис голосу користувача та видає оброблені та структуровані дані про прийом їжі у JSON форматі. Також Laravel підтримує Dependency Injections [11]. В даному випадку це `AudioRequest` (рис. 2.9). Цей тип класів у фреймворках дозволяє валідувати надіслані користувачом дані.

```

7  class AudioRequest extends FormRequest
8  {
9      /**
10     * Determine if the user is authorized to make this request.
11     */
12     public function authorize(): bool
13     {
14         return true;
15     }
16
17     /**
18     * Get the validation rules that apply to the request.
19     *
20     * @return array<string, \Illuminate\Contracts\Validation\ValidationRule|array<mixed>|string>
21     */
22     public function rules(): array
23     {
24         return [
25             'audio' => ['required', 'file', 'mimes:mp3,wav,aac,m4a'],
26             'day_id' => ['required', 'exists:days,id'],
27             'daytime' => ['required', 'string'],
28         ];
29     }
30 }

```

Рис. 2.9 – `AudioRequest.php`

Клас `AudioRequest.php` дозволяє валідувати такі поля:

- `audio`: обов'язкове поле, в яке може бути передано виключно файл та розширення дозволених файлів: `mp3,wav,aac,m4a`;
- `day_id`: обов'язкове поле з ідентифікатором дня, в яке можна передати лише ідентифікатор існуючої моделі в базі даних дня;
- `daytime`: обов'язкове поле з типом стрічки, в яке передається значення `enum` з константою часу дня (наприклад `breakfast, lunch` тощо).

2.2.2.3 Сервіси (Service)

Основним сервісом для обробки даних про прийом їжі є MealService.php (рис. 2.10).

В конструкторах MealService ініціює два додаткових сервіси:

- OpenAIService: відповідає за API запити до сервісів компанії OpenAI через які використовуються для обробки штучним інтелектом голосового введення та даних;
- ImageService: відповідає за взаємодію із зображеннями (завантаження на сервер чи конвертацію формату зображення).

Методи сервісу:

- proceedMealDetailsByAudio

Цей метод обробляє аудіозаписи прийомів їжі. Основні кроки:

- Перетворення аудіофайлу в формат MP3 та його збереження;
 - Використання сервісу OpenAiService для транскрипції аудіозапису в текст
 - Обробка транскрипту для отримання деталей прийому їжі у форматі JSON;
 - Створення запису про прийом їжі в базі даних та збереження отриманих даних;
 - Повернення ідентифікатора прийому їжі та детальної інформації.
- getMealDetailsByTranscript:

Цей метод обробляє транскрипт, використовуючи OpenAiService для генерації структури даних про прийом їжі у форматі JSON. Він передає системні інструкції та текст транскрипту до сервісу, який повертає структуровану інформацію про страви та напої.

- `convertToMP3AndSave`

Цей метод перетворює завантажений аудіофайл у формат MP3 та зберігає його. Основні кроки:

- Збереження тимчасового файлу аудіо.
- Конвертація файлу в формат MP3 за допомогою бібліотеки FFmpeg.
- Видалення тимчасового файлу.
- Повернення імені конвертованого файлу.

- `createOrConfirmMeal`

Цей метод створює новий запис про прийом їжі або підтверджує існуючий. Основні кроки:

- Перевірка та обробка зображень, якщо вони надані.
- Використання транзакції для забезпечення цілісності даних.
- Створення або оновлення запису про прийом їжі з відповідним статусом.
- Збереження зображень, асоційованих з прийомом їжі.
- Повернення результату операції.

```

class MealService
{
    public function __construct(
        private readonly OpenAiService $openAiService,
        private readonly ImageService $imageService
    ) {}

    public function proceedMealDetailsByAudio(mixed $data): array
    {
        $file = $data['audio'];

        $filename = $this->convertToMP3AndSave($file);

        $transcript = $this->openAiService->speechToText($file);

        $details = $this->getMealDetailsByTranscript($transcript['text']);

        $meal = Meal::create([
            'day_id' => $data['day_id'],
            'daytime' => $data['daytime'],
            'transcript' => $transcript['text'],
            'audio' => $filename,
            'data' => $details,
        ]);

        return [
            'meal_id' => $meal->id,
            'details' => $details
        ];
    }
}

```

Рис. 2.10 – MealService.php

Після обробки та отримання транскрипту голосового запису, створюється запит до API від OpenAI зі спеціальними чіткими інструкціями для штучного інтелекту, як саме повинні обробитися дані та в якому форматі їх повернути. Ці інструкції виглядають наступним чином:

You are tasked with organizing meal intake details into JSON format. Follow these guidelines:

1. Only include information provided by the user, without inferring any details.
2. Generate a unique ID for each dish and beverage.

3. For each dish:

- Include the dish name, which must include the amount if provided.
- List the products used with their names and amounts. If an amount is not provided, leave it blank.
- Add any specific cooking details or additional notes (e.g., how it was cooked).
- Exclude beverages and their additions from the dish details.

4. If a dish name is provided more than once with identical products, do not duplicate the dish entry. Instead, update the existing dish by modifying the product amounts accordingly.

5. For products within the same dish that have identical names, combine their amounts rather than listing them separately.

6. For beverages:

- List all beverages consumed, including their names, amounts, and any additions (e.g., sugar) in the notes.

7. Separate meal notes should include any extra meal-related information, excluding products, which should be listed under dishes or beverages.

Use this template (all fields MUST be included even if they are empty):

```
{
  "dishes": [
    {
      "id": "",
      "name": "",
      "products": [
        {
          "name": "",
          "amount": ""
        }
      ],
      "notes": ""
    }
  ],
  "notes": "",
  "beverages": [
    {
```

```

        "id": "",
        "name": "",
        "amount": "",
        "notes": ""
    }
]
}

```

Ensure beverages and their additions are listed only in the beverages section (e.g., juice, coffee, tea) with relevant notes, not in dishes.

Метод `textGeneration` (рис. 2.11) відповідає за генерацію тексту за допомогою моделі GPT-3.5 від OpenAI. Він використовує API запит до сервісу OpenAI для отримання відповіді на основі вхідних повідомлень. Нижче наведено детальний опис функціональності цього методу.

Параметри:

- `array $messages`: Масив повідомлень, які передаються до API для генерації тексту. Кожне повідомлення включає роль (наприклад, 'system', 'user') та контент (текст повідомлення);
- `string $model`: Назва моделі, яка використовується для генерації тексту. За замовчуванням використовується 'gpt-3.5-turbo-0125'.

Процес роботи:

1. Формування запиту:
 - 1.1. Метод формує запит до API OpenAI, використовуючи маршрут 'chat/completions'.
 - 1.2. У запиті вказується модель, масив повідомлень та формат відповіді (у даному випадку JSON-об'єкт).

2. Відправка запиту:
 - 2.1. Використовується метод `request` для відправки запиту до API OpenAI з переданими даними.
3. Обробка відповіді:
 - 3.1. Метод перевіряє, чи містить відповідь API ключ `'choices'` з першим елементом, який має ключ `'message'` та підключ `'content'`;
 - 3.2. Якщо такий елемент присутній, метод декодує JSON-строку у масив та повертає результат.
4. Обробка помилок:
 - 4.1. Якщо відповідь API не містить очікуваних даних, метод повертає масив з повідомленням про помилку.

```

public function textGeneration(array $messages, string $model = 'gpt-3.5-turbo-0125'): array
{
    $response = $this->request( uri: 'chat/completions', data: [
        'model' => $model,
        'messages' => $messages,
        'response_format' => ['type' => 'json_object']
    ]);

    if (isset($response['choices'][0]['message']['content'])) {
        return json_decode($response['choices'][0]['message']['content'], associative: true);
    }

    return ['error' => 'Something went wrong. Try again'];
}

```

Рис. 2.11 – метод відправки API запиту на генерацію тексту

2.2.2.4 Моделі

Модель у Laravel представляє собою клас, який слугує абстракцією для взаємодії з базою даних, дозволяючи розробникам працювати з даними як з об'єктами. Вона містить визначення структури даних, що зберігаються в таблиці, методи для маніпуляції цими даними, а також встановлює зв'язки з іншими моделями, забезпечуючи таким чином логічну цілісність та зручність роботи з даними у додатку.

Модель Meal у Laravel представляє собою ORM (Object-Relational Mapping) клас, який використовується для взаємодії з таблицею бази даних, що зберігає інформацію про прийоми їжі. Ця модель визначає структуру даних, які можуть бути збережені в таблиці, та забезпечує зв'язки з іншими моделями.

Одна з основних моделей Meal використовує властивість `$fillable` для визначення масиву атрибутів, які можуть бути масово призначені. Це включає поля `day_id`, `daytime`, `transcript`, `audio`, `data` та `status`. Таке визначення дозволяє захистити модель від масового присвоєння небажаних атрибутів, підвищуючи безпеку додатку.

Властивість `$casts` використовується для автоматичного приведення атрибутів до визначених типів. У цьому випадку атрибут `data` буде автоматично конвертуватися в масив при отриманні з бази даних і назад в JSON при збереженні.

Модель також визначає два зв'язки з іншими моделями. Метод `images` встановлює зв'язок "один до багатьох" з моделлю `Image`, що означає, що кожен прийом їжі може мати багато асоційованих зображень. Це досягається за допомогою методу `hasMany`, який повертає інстанцію класу `HasMany`.

Метод `day` визначає зв'язок "багато до одного" з моделлю `Day`. Це означає, що кожен прийом їжі належить до конкретного дня. Зв'язок встановлюється за допомогою методу `belongsTo`, який повертає інстанцію класу `BelongsTo`.

Таким чином, модель Meal забезпечує структуровану взаємодію з таблицею бази даних, дозволяючи легко зберігати, отримувати та управляти даними про прийоми їжі, а також забезпечує необхідні зв'язки з іншими моделями для підтримки цілісності даних.

2.2.3 Користувацький інтерфейс (frontend)

Як було зазначено вище, для фронтенду було використано React Native. Основна увага приділялася створенню зручного та інтуїтивно зрозумілого інтерфейсу користувача з привабливим дизайном. Було розроблено інтерфейс,

який дозволяє користувачам легко взаємодіяти з додатком, виконувати необхідні дії та отримувати важливу інформацію.

Дизайн додатку був створений з урахуванням найкращих практик UI/UX дизайну. Інтерфейс включає чітко структуровані меню та екрани, що дозволяють користувачам швидко знаходити потрібні функції та дані. Основний екран додатку містить зручні кнопки навігації, які ведуть до ключових розділів додатку, таких як дні заповнення форм, інструкційні відео та інші необхідні функції.

Були розроблені форми для введення даних, що сприяють зручності використання додатку. Користувачі можуть легко записувати свої прийоми їжі, переглядати інструкції та вносити необхідні корективи. Інтерфейс також забезпечує можливість перегляду та редагування даних перед їх остаточним збереженням, що підвищує точність та якість зібраної інформації.

Особлива увага приділялася інтерактивності та візуальній привабливості інтерфейсу. Було використано сучасні візуальні елементи, які роблять додаток не тільки функціональним, але й приємним у використанні. Всі елементи інтерфейсу були оптимізовані для забезпечення швидкої та плавної роботи додатку на різних пристроях.

2.2.3.1 Вхідний файл App.tsx

Файл App.tsx є основним файлом додатку, написаного на React Native. Він відповідає за налаштування та управління навігацією між екранами, а також за авторизацію користувачів. Спочатку імпортуються необхідні компоненти та функції, такі як різні екрани додатку (HomeScreen, LoadingScreen, LoginScreen, FormScreen, DayScreen, MenuScreen), а також контейнери для навігації (NavigationContainer і createNativeStackNavigator) та контексти для авторизації (AuthProvider і useAuth). Додатково імпортуються функції для роботи з FCM (Firebase Cloud Messaging) та бібліотека для управління сплеш-екраном.

У функціональному компоненті App використовується контекст useAuth,

щоб отримати інформацію про поточного користувача та стан завантаження. Використовуються два хука `useEffect`: перший ховає сплеш-екран і отримує FCM-токен під час першого завантаження додатку, другий реєструє слухача для FCM і очищає його при розмонтуванні компонента.

Основний функціонал `App` перевіряє, чи знаходиться додаток в стані завантаження. Якщо так, то відображається екран завантаження. Якщо завантаження завершено, а користувач авторизований, відображаються основні екрани додатку (головна сторінка, форми, дні, меню). Якщо користувач не авторизований, відображається екран входу.

Компонент `Root` є обгорткою для `App`, забезпечуючи контекст авторизації через `AuthProvider`. Це дозволяє передавати стан авторизації і функції для управління нею по всьому додатку. Таким чином, файл `App.js` забезпечує структуровану і логічну навігацію, а також управління станом авторизації, що робить додаток зручним і функціональним для користувачів.

2.2.3.2 Екран форми (FormScreens)

Екран `FormScreen` є центральним компонентом додатку, який дозволяє користувачам записувати аудіо, обробляти його, а також додавати і редагувати дані про прийом їжі. Він реалізований за допомогою `React Native`, що дозволяє створювати зручний і інтуїтивно зрозумілий інтерфейс для мобільних пристроїв. Кожен компонент у `React Native` відповідає за певну частину функціональності додатку. Наприклад, на `FormScreen` є компоненти для запису аудіо, відображення та редагування інформації про страви і напої, а також для завантаження фотографій їжі.

Основна функціональність `FormScreen` полягає в записі аудіо користувачами. Для цього використовуються методи `onStartRecord` і `onStopRecord` (рис. 2.12), які починають та завершують запис, відповідно. Записане аудіо зберігається і передається на сервер для обробки. Після обробки аудіо

користувачі можуть бачити і редагувати транскрибовані дані про прийом їжі. Крім того, користувачі можуть додавати фотографії страв, що допомагає забезпечити точність зібраної інформації.

Компоненти в React Native є будівельними блоками інтерфейсу користувача. Вони можуть бути простими, такими як текстові поля або кнопки, або складними, такими як форми для введення даних чи модальні вікна. На FormScreen використовуються компоненти DishItem та BeverageItem, які відображають інформацію про страви та напої, відповідно. Модальні вікна DishModal та BeverageModal дозволяють користувачам додавати або редагувати інформацію про страви та напої.

Для управління станом додатку використовуються хуки, такі як useState та useEffect. Хук useState дозволяє зберігати і оновлювати стан компонента, наприклад, чи є аудіо в процесі запису, або які страви і напої були додані користувачем. Хук useEffect використовується для виконання побічних ефектів, таких як налаштування слухачів подій або запуск таймерів.

Стилі в React Native визначаються за допомогою об'єктів JavaScript. Вони подібні до CSS, але мають деякі відмінності в синтаксисі та можливостях. Стилі визначають зовнішній вигляд компонентів, такі як кольори, шрифти, відступи та розміри. На FormScreen використовуються стилі для кнопок, текстових полів, контейнерів та інших елементів інтерфейсу, щоб забезпечити їх привабливий вигляд і зручність використання.

Метод proceedAudio відповідає за обробку аудіо, відправляючи його на сервер для транскрипції. Після отримання результатів обробки, дані про прийом їжі відображаються на екрані, де користувач може їх редагувати та доповнювати. Метод saveForm відповідає за збереження введених даних на сервері. Він обробляє форми, додає завантажені зображення і відправляє всі дані на сервер для зберігання.

FormScreen використовує кілька бібліотек для забезпечення додаткових функцій, таких як react-native-audio-recorder-player для запису і відтворення аудіо, react-native-date-picker для вибору дати і часу. Інтерфейс цього екрану було

розроблено з урахуванням зручності користувачів, забезпечуючи інтуїтивно зрозумілу навігацію і легкий доступ до всіх необхідних функцій. Це дозволяє користувачам швидко і ефективно вводити дані про прийом їжі, що значно полегшує процес збору інформації для дослідження.

```
const [isProcessing, setIsProcessing] = useState( initialState: false);
const [datetime, setDatetime] = useState(new Date());
const [isOpenDatePicker, setIsOpenDatePicker] = useState( initialState: false);
const [instructionalVideoModal, setInstructionalVideoModal] = useState( initialState: false);
const [noFood, setNoFood] = useState( initialState: false);
const [uploadedImages, setUploadedImages] = useState( initialState: []);
const [imagesUploadModal, setImagesUploadModal] = useState( initialState: false);

const timeoutRef = useRef( initialValue: null);
const scrollViewRef = useRef( initialValue: null);
const inputRef = useRef( initialValue: null);

1 usage  🧑 Maksym Shteklia
const onStartRecord = async () => {
  const result = await audioRecorderPlayer.startRecorder();
  audioRecorderPlayer.addRecordBackListener( callback: e => {
    console.log("recordBack", e);
  });
  console.log(result);
  setIsRecording( value: true);
  setStartTime(Date.now());
  setRecordingDuration( value: 0);
};

1 usage  🧑 Maksym Shteklia
const onStopRecord = async () => {
  const result = await audioRecorderPlayer.stopRecorder();
  audioRecorderPlayer.removeRecordBackListener(e => {
    console.log("stop");
  });
  console.log(result);
  setIsRecording( value: false);

  setAudioPath(result);
};
```

Рис. 2.12 – Основні функції компоненту FormScreen

Не менш важливою частиною додатку є обробка даних та взаємодія з бекендом через API. На екрані FormScreen це здійснюється за допомогою функцій `proceedAudio` та `createMeal`, які відправляють дані на сервер для подальшої обробки.

Функція `proceedAudio` (рис. 2.13) відповідає за відправку аудіофайлу на сервер. Вона використовує метод `post` клієнта `API` для надсилання запиту з форм-даними, які містять аудіофайл. Сервер обробляє аудіофайл, виконує його транскрипцію та повертає деталі прийому їжі. Ці деталі потім відображаються на екрані, де користувач може їх переглянути та за потреби відредагувати. У випадку виникнення помилки, функція `proceedAudio` логує її в консоль для подальшого аналізу.

Функція `createMeal` (рис. 2.13) використовується для збереження даних про прийом їжі після їх перевірки та редагування користувачем. Вона також використовує метод `post` для відправки форм-даних на сервер. Ці дані включають інформацію про страви, напої та додаткові нотатки користувача. Якщо запит успішно оброблений сервером, функція повертає отримані дані. У випадку виникнення помилки, функція повертає відповідне повідомлення про помилку.

Використання таких функцій дозволяє забезпечити ефективну та надійну взаємодію між фронтендом та бекендом додатку. Завдяки цьому користувачі можуть швидко та зручно передавати дані на сервер, отримувати необхідну інформацію та зберігати результати. Це сприяє підвищенню точності та ефективності збору даних, що є критично важливим для успішного проведення дослідження та отримання достовірних результатів.

2.3 Результати проектування

Результати проектування додатку виявилися досить успішними, забезпечивши зручність використання та високу ефективність збору даних. Основним досягненням стало створення інтуїтивно зрозумілого інтерфейсу, який дозволяє користувачам легко записувати та надсилати аудіофайли, переглядати та редагувати деталі прийомів їжі, а також зберігати їх у системі. Завдяки використанню `React Native` для фронтенду та `Laravel` для бекенду, вдалося

досягти високої швидкості роботи додатку та надійності обробки даних.

Процес проєктування включав кілька ключових етапів, серед яких виділяються: аналіз вимог, проєктування архітектури системи, розробка користувацького інтерфейсу, інтеграція з серверною частиною та тестування функціональності. Під час аналізу вимог були визначені основні функції та завдання додатку, що дозволило сформулювати чіткі технічні вимоги та критерії успішності проєкту. Проєктування архітектури системи передбачало створення логічної структури додатку, яка забезпечує оптимальну взаємодію між різними його компонентами.

Розробка користувацького інтерфейсу з використанням компонентного підходу React Native дозволила створити зручний та інтуїтивно зрозумілий інтерфейс. Кожен компонент – це окремий елемент інтерфейсу, який виконує певну функцію, наприклад, запис аудіо, введення тексту, відображення списків страв та напоїв тощо. Компонентний підхід забезпечує високу гнучкість та повторне використання коду, що значно спрощує розробку та підтримку додатку. Крім того, використання хуків React дозволило оптимізувати управління станом компонентів та обробку побічних ефектів, таких як завантаження даних або анімації.

Інтеграція з серверною частиною, розробленою на базі Laravel, дозволила забезпечити надійне та ефективне управління даними. Вся комунікація між фронтендом та бекендом здійснюється через API, що дозволяє легко розширювати функціональність додатку без значних змін у його архітектурі. Laravel забезпечує високу швидкість обробки запитів та надійність збереження даних у базі даних MySQL. Використання каркасу сутностей у Laravel дозволяє при необхідності інтегрувати додаток з іншими базами даних, що робить додаток більш гнучким та масштабованим.

Стилі додатку були розроблені з урахуванням сучасних тенденцій у дизайні мобільних додатків, що забезпечує естетично привабливий та зручний інтерфейс. Кожен екран додатку, включаючи головний екран, екран форми та екран з детальною інформацією про прийоми їжі, був ретельно спроектований з

урахуванням потреб користувачів. Використання CSS-in-JS підходу дозволило забезпечити високу гнучкість у налаштуванні стилів та швидке внесення змін за потреби.

Значну увагу було приділено забезпеченню зручності користування додатком на різних пристроях. Адаптивний дизайн дозволяє користувачам комфортно використовувати додаток як на мобільних телефонах, так і на планшетах. Це досягається завдяки використанню відсоткових значень для ширини та висоти елементів, а також медіазапитів для налаштування стилів під різні розміри екранів.

Функції додатку, такі як запис аудіо, транскрипція голосових записів, обробка даних та їх відправка на сервер, працюють бездоганно завдяки ефективній інтеграції з OpenAI та використанню надійних технологій для обробки аудіо та тексту. Це дозволяє забезпечити точність та надійність зібраних даних, що є критично важливим для проведення дослідження. Зокрема, було реалізовано можливість автоматичного перетворення аудіозаписів у текстовий формат, що значно спрощує процес введення даних для користувачів.

Загалом, результати проєктування додатку показують, що вдалося досягти всіх поставлених цілей: створити зручний та ефективний інструмент для збору та обробки даних, який відповідає вимогам користувачів та забезпечує високу якість результатів дослідження. Важливим аспектом є те, що всі процеси, від запису аудіо до зберігання відформатованих даних, були автоматизовані та оптимізовані, що забезпечує зручність використання додатку та високу швидкість обробки інформації.

РОЗДІЛ 3. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

3.1 Роль центральної нервової системи в трудовій діяльності людини

Центральна нервова система (ЦНС) відіграє ключову роль у регуляції трудової діяльності людини, виконуючи функції координації, контролю та інтеграції різних видів активності організму. Вона забезпечує координацію рухів і моторики, що є необхідним для виконання фізичних завдань. Мозок і спинний мозок отримують інформацію від рецепторів, обробляють її та відправляють відповідні сигнали до м'язів, що дозволяє людині виконувати точні і складні рухи. Це особливо важливо для таких професій, які вимагають високої точності та координації, як хірургія, музика чи спорт.

ЦНС також відповідає за розумову діяльність, таку як планування, аналіз та прийняття рішень. Вона забезпечує здатність до обробки великої кількості інформації, що надходить з навколишнього середовища, і прийняття на її основі обґрунтованих рішень. Це особливо важливо для виконання складних та багатозадачних завдань на роботі, таких як управління проектами, фінансовий аналіз чи наукові дослідження. Кора головного мозку, яка є частиною ЦНС, відповідає за обробку сенсорної інформації, що надходить з зовнішнього середовища, і за генерацію адекватних відповідей. Ця здатність до інтеграції і аналізу інформації дозволяє людині адаптуватися до змінних умов праці, прогнозувати результати своїх дій і вносити корективи у свої стратегії.

Емоційний стан людини, який також регулюється ЦНС, суттєво впливає на продуктивність праці. Стрес, тривога та інші емоційні стани можуть впливати на здатність зосереджуватися та виконувати завдання. Важливу роль у цьому контексті відіграє лімбічна система, яка відповідає за обробку емоцій та формування мотивації. Наприклад, в умовах високого стресу, як під час кризових ситуацій або термінових проектів, здатність ЦНС ефективно керувати емоціями і мотивацією може визначати успішність виконання завдань.

Здатність до творчого мислення і вирішення проблем також залежить від

ЦНС. Префронтальна кора головного мозку відповідає за функції, пов'язані з плануванням, інноваціями і вирішенням складних задач. Люди, які можуть генерувати нові ідеї і пропонувати нетривіальні рішення, часто стають цінними співробітниками в будь-якій сфері діяльності, від наукових досліджень до бізнесу та інженерії. ЦНС забезпечує адаптацію до змін навколишнього середовища та умов праці. Вона допомагає організму пристосовуватися до нових вимог та стресових ситуацій, підтримуючи гомеостаз та ефективність діяльності.

Завдяки здатності до навчання та запам'ятовування, людина може вдосконалювати свої навички та підвищувати кваліфікацію, що є важливим аспектом професійного розвитку. Гіпокамп, частина лімбічної системи, відіграє ключову роль у формуванні довготривалої пам'яті. Це дозволяє людині накопичувати професійні знання і навички, які можуть бути використані для підвищення ефективності праці. В умовах швидкої зміни технологій і методів роботи здатність до швидкого навчання і адаптації стає особливо важливою. Наприклад, у сфері інформаційних технологій, де інновації відбуваються дуже швидко, здатність ЦНС до навчання і адаптації визначає успіх працівника.

Мотивація і емоційний стан суттєво залежать від функціонування ЦНС. Вона регулює виділення нейромедіаторів, таких як допамін і серотонін, які впливають на почуття задоволення і благополуччя. Це, у свою чергу, впливає на мотивацію до праці, рівень залученості в роботу і здатність досягати поставлених цілей. Люди, які мають стабільну і збалансовану роботу ЦНС, зазвичай демонструють вищу продуктивність і більшу стійкість до стресу. Наприклад, стабільний рівень допаміну сприяє підвищеній мотивації та відчуттю задоволення від досягнень, що може бути ключовим фактором у кар'єрному зростанні та професійному розвитку[12].

ЦНС також впливає на фізіологічні аспекти трудової діяльності, такі як цикл сну і неспання, який регулюється циркадними ритмами. Адекватний сон є критично важливим для відновлення мозку і підтримки його функціональної активності. Недосипання може призвести до зниження когнітивних функцій, зниження продуктивності і підвищення ризику професійних помилок.

Наприклад, у професіях, які вимагають високої концентрації і точності, як-от лікарі, пілоти або водії, недостатній сон може призвести до катастрофічних наслідків.

ЦНС також відіграє важливу роль у підтриманні фізичної витривалості та енергійності. Спинний мозок і різні частини мозку працюють разом, щоб регулювати роботу м'язів, серцево-судинної системи та інших органів, що дозволяє людині працювати протягом тривалого часу без втоми. Наприклад, фізична праця, така як будівництво або спорт, вимагає високої координації між ЦНС і м'язовою системою для ефективного виконання завдань[12].

Таким чином, центральна нервова система є комплексним механізмом, який забезпечує не лише базові функції, необхідні для фізичної діяльності, але й складні когнітивні та емоційні процеси, що впливають на загальну продуктивність і ефективність людини в трудовій діяльності.

3.2 Гігієнічні вимоги до організації та обладнання робочих місць з ВДТ

Гігієнічні вимоги до організації та обладнання робочих місць користувачів комп'ютерів спрямовані на створення комфортних, безпечних та здорових умов праці. Приміщення повинні мати відповідні об'ємно-планувальні рішення, що включають заборону розміщення робочих місць у підвальних приміщеннях та на цокольних поверхах.

Площа на одне робоче місце повинна бути не менше 6 м², а об'єм – не менше 20 м³. Відстань між робочими столами повинна бути щонайменше 2,5 м у рядку та 1,2 м між рядами. Стіни приміщень слід фарбувати у пастельні тони з коефіцієнтом відбиття 0,5-0,6, що сприяє комфортному зоровому сприйняттю. Приміщення мають бути оснащені шафами для зберігання документів, полицями, стелажми та іншими меблями, враховуючи вимоги до площі.

Приміщення з ВДТ повинні бути оснащені аптечками першої допомоги, а

також побутовими приміщеннями для відпочинку під час роботи, включаючи кімнату психологічного розвантаження. У кімнаті психологічного розвантаження слід передбачити місця для приготування та роздачі тонізуючих напоїв, а також місця для занять фізичною культурою.

Обладнання і організація робочого місця повинні забезпечувати ергономічність конструкції всіх елементів робочого місця з урахуванням характеру та особливостей трудової діяльності. Конструкція робочого місця повинна забезпечувати підтримання оптимальної робочої пози. Робочі місця повинні бути розташовані відносно вікон так, щоб природне світло падало збоку, переважно зліва.

Висота робочої поверхні столу має регулюватися в межах 680-800 мм, а ширина та глибина повинні забезпечувати зручність виконання операцій у зоні досяжності моторного поля. Робочий стіл повинен мати простір для ніг заввишки не менше 600 мм, завширшки не менше 500 мм і завглибшки не менше 450 мм на рівні колін та 650 мм на рівні простягнутої ноги.

Робочий стілець має бути підйомно-поворотним, регульованим за висотою, кутом нахилу сидіння та спинки, а також відстанню від спинки до переднього краю сидіння. Поверхня сидіння має бути плоскою, передній край – заокругленим. Висота поверхні сидіння має регулюватися в межах 400-500 мм, а ширина та глибина становити не менше 400 мм. Кут нахилу сидіння – до 15° вперед і до 5° назад. Підставка для ніг повинна мати ширину не менше 300 мм, глибину не менше 400 мм, регулюватися за висотою до 150 мм і кутом нахилу до 20°.

Екран ВДТ повинен бути розташований на відстані 600-700 мм від очей користувача і забезпечувати зручність зорового спостереження під кутом +30° до нормальної лінії погляду. Клавіатура має розташовуватися на поверхні столу на відстані 100-300 мм від краю, звернутого до працюючого, і повинна мати опорний пристрій, що дозволяє змінювати кут нахилу поверхні клавіатури в межах 5-15°.

Для захисту від комп'ютерних випромінювань необхідно застосовувати

приєкранні фільтри та локальні світлофільтри. Робочі місця з ВДТ повинні бути обладнані пюпітрами для документів, що легко переміщуються.

З метою зниження втоми та підтримання працездатності, організація праці передбачає внутрішньозмінні регламентовані перерви для відпочинку, а також комплекси гімнастичних вправ. Сеанси психологічного розвантаження можуть включати елементи аутогенного тренування, що сприяють розслабленню та активізації працездатності. Для користувачів ПК проводяться обов'язкові медичні огляди, включаючи попередні при влаштуванні на роботу та періодичні протягом трудової діяльності.

Дотримання цих вимог допомагає запобігти професійним захворюванням, таким як остеохондроз, сколіоз, простатит, гінекологічні хвороби, астенопія, синдром карпального каналу та інші. Вагітні жінки повинні бути відсторонені від роботи з ПК через ризики для здоров'я матері та дитини.

ВИСНОВКИ

Проект розробки мобільного додатку для збору та обробки даних про прийоми їжі в рамках дослідження неалкогольної жирової хвороби печінки серед корінних народів Канади виявився успішним та досяг усіх поставлених цілей. Використання сучасних технологій, таких як React Native для фронтенду та Laravel для бекенду, дозволило створити зручний, інтуїтивно зрозумілий та ефективний інструмент для користувачів.

Завдяки компонентному підходу React Native вдалося забезпечити високу гнучкість у розробці інтерфейсу, що дозволяє легко адаптувати додаток до різних платформ та розмірів екранів. Використання хуків React дозволило оптимізувати управління станом компонентів та обробку побічних ефектів, що значно підвищило продуктивність додатку. Інтеграція з Laravel забезпечила надійне управління даними та швидку обробку запитів через API, що дозволило досягти високої швидкості роботи додатку.

Особливу увагу було приділено процесу збору та обробки даних. Можливість запису аудіо, його автоматична транскрипція за допомогою OpenAI, а також подальша обробка та форматування даних значно спростили процес введення інформації для користувачів. Це дозволило забезпечити точність та актуальність зібраних даних, що є критично важливим для успішного проведення дослідження.

Проектування додатку також включало розробку адаптивного дизайну, який забезпечує зручність користування на різних пристроях. Інтерфейс був спроектований з урахуванням сучасних тенденцій у дизайні мобільних додатків, що забезпечує естетично привабливий вигляд та інтуїтивну зрозумілість для користувачів.

В цілому, розроблений додаток повністю відповідає вимогам дослідження та надає зручний інструмент для збору та обробки даних про харчування. Проект продемонстрував високу ефективність застосованих технологій та підходів, забезпечивши надійність, швидкість та зручність роботи з додатком. Це дозволяє

з упевненістю стверджувати, що розроблений додаток є важливим кроком у напрямку підвищення ефективності досліджень у галузі охорони здоров'я та сприяє поліпшенню здоров'я населення через більш точний збір та аналіз даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Study brings clinical research on growing problem of liver disease to northern Manitoba communities. URL:
<https://www.cbc.ca/news/canada/manitoba/manitoba-first-nations-liver-disease-study-1.5128684>
2. Indigenous Health Conference. URL:
<https://www.cpd.utoronto.ca/indigenoushealth/files/2020/11/IHC-Abstract-Book.pdf>
3. UOM official website. URL: <https://umanitoba.ca/>
4. Refuah SOLUTIONS LTD. URL: <https://refuahsolutions.com/>
5. What is DSDM? URL:
<https://www.agilebusiness.org/business-agility/what-is-dsdm.html>
6. Frontend and backend. URL:
<https://aws.amazon.com/compare/the-difference-between-frontend-and-backend/#:~:text=Frontend%20and%20backend%20are%20two,that%20make%20your%20application%20work.>
7. React Native. Wiki. URL: https://wikipedia.org/wiki/React_Native
8. Laravel Website. URL: <https://laravel.com/>
9. MVC in Laravel. URL: [https:// wikipedia.org/wiki/Laravel](https://wikipedia.org/wiki/Laravel)
10. GPT-3. URL: <https://uk.wikipedia.org/wiki/GPT-3>
11. Dependency Injection. URL:
https://en.wikipedia.org/wiki/Dependency_injection
12. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ ТА ОХОРОНА ПРАЦІ / Я. О. СЕРІКОВ, Л. Ф. КОЖЕНЕВСЬКІ, М. В. ХВОРОСТ – ХНУМГ ім. О. М. Бекетова, 2021. URL:
https://elartu.tntu.edu.ua/bitstream/lib/42437/1/dyplom_Yamko_2023.pdf
13. НАВЧАЛЬНИЙ ПОСІБНИК «БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ»
 URL: https://os.nure.ua/wp-content/uploads/2019/05/posibnik-bgd_2018_p.1.pdf