КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра технологій управління

Спеціальність 122 «Комп'ютерні науки»
Освітньо-професійна програма «Управління проектами»

КВАЛІФІКАЦІЙНА РОБОТА МАГІСТРА

на тему:

«Дослідження процесів управління проєктом створення мобільного додатку «PetHealth»

Студентки 2-го курсу групи УПз-21

Тетяни ДАНІЛІНОЇ
(Ім'я, ПРІЗВИЩЕ)

_____
(підпис студента)

Науковий керівник:

к.т.н., доцент
(науковий ступінь, вчене звання)

Вадим ЗЮЗЮН
(Ім'я, ПРІЗВИЩЕ)

_____ _____
(дата) *(підпис)*

Попередній захист:

_____
(Висновок: «До захисту в Екзаменаційній комісії»)

Завідувач кафедри
технологій управління _____ Віктор МОРОЗОВ _____
(підпис) *(Ім'я, ПРІЗВИЩЕ)* *(дата)*

Київ – 2025

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій

Кафедра технологій управління
Освітній рівень Магістр
Спеціальність 122 Комп'ютерні науки
Освітньо-професійна програма Управління проектами

ЗАТВЕРДЖУЮ
Завідувач кафедри
професор Віктор МОРОЗОВ

_____
«29» вересня 2025 року

ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Студентка: Даніліна Тетяна Олександрівна
Група: УПз-21

1. Тема кваліфікаційної роботи: «Дослідження процесів управління проєктом створення мобільного додатку «PetHealth»
Затверджена Протоколом № 15 від «16» червня 2025 року.

2. Строк подання студентом готової роботи – «10» грудня 2025 року.

3. Цільова установка та вихідні дані до роботи: дослідження методів і інструментів управління проєктом створення мобільного додатку «PetHealth»; обґрунтування вибору гнучкої методології управління проєктом (Scrum); формування концептуальної та математичної моделей, архітектури програмного забезпечення та структури бази даних; визначення методів планування, управління ризиками та забезпечення якості; результати аналізу ринку PetTech та конкурентного середовища; виявлені потреби та проблеми цільової аудиторії; сформовані функціональні та нефункціональні вимоги до системи; розроблена структурна декомпозиція робіт та організаційна структура команди;

сформований беклог продукту та план спринтів; ідентифіковані ризики, розроблені заходи реагування та розрахований бюджет проєкту.

4. Зміст роботи: обґрунтування актуальності та доцільності проєкту створення мобільного додатку «PetHealth»; комплексний аналіз ринку PetTech, конкурентного середовища та зацікавлених сторін; проведення PEST- і SWOT-аналізу; побудова дерева цілей та формування паспорту проєкту. Розробка концептуальної моделі інформаційної системи, формалізація математичної моделі предметної області та обґрунтування використання методів штучного інтелекту. Визначення функціональних та нефункціональних вимог, обґрунтування вибору методології Scrum, формування ієрархічної структури робіт та організаційної структури команди; створення беклогу продукту на основі користувацьких історій та планування робіт у вигляді спринтів; розробка сценаріїв використання та побудова відповідних діаграм. Проведення ідентифікації та кількісної оцінки ризиків, розробка протиризикових заходів та розрахунок загального бюджету проєкту. Опис архітектури програмного забезпечення та стеку технологій; розробка концептуальної, логічної та фізичної моделей бази даних; проєктування інтерфейсу користувача у Figma та формування плану управління якістю.

5. Перелік графічного матеріалу: таблиці PEST- та SWOT-аналізу; дерево проблем та дерево цілей проєкту; концептуальна модель мобільного додатку; структурна декомпозиція робіт та організаційна структура проєкту; діаграма варіантів використання; ідентифікація, оцінка та протиризикові заходи проєкту; архітектура основних компонентів програмного забезпечення; концептуальна та логічна моделі бази даних; дизайн-макети інтерфейсів користувача мобільного додатку.

6. Календарний план виконання роботи

№ з/п	Назва частин роботи	Виконання роботи
1	Аналіз літературних джерел з предмету дослідження	01.10.25-10.10.25

2	Збір матеріалів для обґрунтування проблеми та доцільності дослідження	13.10.25-17.10.25
3	Складання плану кваліфікаційної роботи	20.10.25-22.10.25
4	Узгодження плану роботи з науковим керівником.	23.10.25-24.10.25
5	Підготовка розділу 1	27.10.25-31.10.25
6	Підготовка розділу 2	03.11.25-07.11.25
7	Підготовка розділу 3	10.11.25-14.11.25
8	Підготовка розділу 4	17.11.25-21.11.25
9	Погодження та внесення змін, рекомендацій від наукового керівника.	24.11.25-26.11.25
10	Оформлення кваліфікаційної роботи	27.11.25-28.11.25
11	Попередній захист кваліфікаційної роботи	10.12.25-14.12.25
12	Передача кваліфікаційної роботи рецензенту для рецензування	15.12.25

Дата видачі завдання «29» вересня 2025 р.

Керівник роботи доцент, Вадим ЗЮЗЮН

_____
(підпис)

Завдання прийняла до виконання
студентка групи УПз-21 Тетяна ДАНІЛІНА

_____
(підпис)

АНОТАЦІЯ

кваліфікаційної роботи магістра на тему:

«Дослідження процесів управління проєктом створення мобільного додатку «PetHealth»

Студентка: Даніліна Тетяна Олександрівна

Науковий керівник: Зюзюн Вадим Ігорович

Рік захисту – 2025.

Мета кваліфікаційної роботи полягає у розробці моделі процесів управління для проєкту створення мобільного додатку «PetHealth», що базується на методології Agile (Scrum), та дозволить оптимізувати строки розробки, підвищити якість кінцевого продукту та ефективно управляти проєктними ризиками.

Ціль проєкту – створення кросплатформного мобільного додатку для автоматизації взаємодії між власниками тварин і ветеринарними клініками, що забезпечує централізоване зберігання медичних даних, онлайн-бронювання послуг та підвищення ефективності лікування.

Практична цінність полягає у розробці універсальної моделі процесів управління та необхідного інформаційного забезпечення, які можуть бути застосовані для ефективної реалізації інноваційних ІТ-проєктів у сфері PetTech. Результатом є сформована система управління, що гарантує прозорий обмін медичними даними та мінімізацію ризиків проєкту.

Кваліфікаційної робота складається з анотації, вступу, основної частини, яка включає чотири розділи, висновків, переліку використаних інформаційних джерел та додатків.

Перший розділ присвячено дослідженню предметної області та теоретичному обґрунтуванню доцільності проєкту. Проведено комплексний аналіз ринку PetTech, конкурентного середовища, PEST- та SWOT-аналіз, а також ідентифікацію та класифікацію зацікавлених сторін. На основі отриманих

даних побудовано дерево проблем і цілей, сформовано паспорт проєкту, що підтверджує його життєздатність та актуальність.

У другому розділі розроблено концептуальну модель системи «PetHealth», що визначає її архітектуру та логічні зв'язки. Виконано формалізацію та математичну постановку задачі за допомогою апарату теорії множин. Обґрунтовано необхідність використання методів штучного інтелекту, зокрема технологій обробки природної мови, для створення інтелектуального помічника VetBot.

Третій розділ містить розробку інформаційного забезпечення та плану управління проєктом. Обґрунтовано вибір гнучкої методології Scrum. Створено структурну та організаційну декомпозиції робіт, сформовано функціональні вимоги у вигляді беклогу продукту та детального планування робіт у спринтах. Проведено ідентифікацію, кількісну оцінку ризиків та розроблено стратегії реагування, а також розраховано бюджет проєкту.

Четвертий розділ описує технічну реалізацію та управління якістю. Обґрунтовано вибір технологічного стеку та архітектури програмного забезпечення. Здійснено проєктування бази даних PostgreSQL та розроблено інтерфейси користувача у Figma. Сформовано план забезпечення якості.

Кваліфікаційна робота складається з 103 сторінок основного тексту, містить 22 рисунки, 32 таблиці, 22 формули та 7 додатків.

Ключові слова: управління проєктами, Agile, Scrum, мобільний додаток, база даних, MVP, ветеринарія.

ЗМІСТ

ВСТУП

Наразі ветеринарна галузь України стрімко набирає оберти в цифровізації. Поштовхом до цього слугує не лише значний обсяг ринку, що налічує мільйони домашніх тварин, але й нові законодавчі ініціативи. Зокрема, впровадження обов'язкової реєстрації, запуск Єдиного державного реєстру домашніх тварин та гармонізація українських ветеринарних стандартів із вимогами ЄС підкреслюють потребу в сучасних, централізованих ІТ-рішеннях.

Наявні на ринку ІТ-рішення не забезпечують комплексного рішення, через що власники тварин стикаються з незручностями, а обмін даними між ветеринарами залишається ускладненим.

Проєкт «PetHealth» пропонує вирішення цієї проблеми шляхом створення єдиної інформаційної системи за аналогією з успішними платформами в медицині для людей, як-от Helsi. Водночас, успіх реалізації такого ІТ-проєкту в умовах високої невизначеності напряму залежить від ефективності обраної моделі управління.

Мета кваліфікаційної роботи магістра полягає у розробці моделі процесів управління для проєкту створення мобільного додатку «PetHealth», що базується на методології Agile (Scrum), та дозволить оптимізувати строки розробки, підвищити якість кінцевого продукту та ефективно управляти проєктними ризиками.

Для досягнення поставленої мети необхідно вирішити такі ***завдання***:

1. Обґрунтувати актуальність проєкту в контексті цифровізації ветеринарних послуг.
2. Розробити концептуальну та формалізовану математичну модель системи.
3. Обґрунтувати вибір методології Scrum, створити структурну декомпозицію робіт, організаційну структуру та сформувати беклог продукту.
4. Провести ідентифікацію та кількісну оцінку проєктних ризиків, розробити стратегії реагування та здійснити розрахунок бюджету проєкту.

5. Розробити архітектуру програмного забезпечення, логічну модель бази даних та дизайн інтерфейсів користувача для ключових функціональних модулів.

Об'єктом дослідження є процеси створення цифрової платформи для взаємодії власників тварин і ветеринарних клінік «PetHealth».

Предметом дослідження є методи, моделі та інструменти управління проєктом створення кросплатформного мобільного додатку «PetHealth» з використанням адаптивної методології Agile.

Методи дослідження. У роботі використано комплекс методів, що включає: аналіз та синтез (для дослідження теоретичних основ управління проєктами, порівняння методологій та систематизації результатів); методи стратегічного аналізу (проведено PEST-аналіз зовнішнього середовища та SWOT-аналіз для визначення життєздатності концепції); системний аналіз (для побудови дерева проблем та дерева цілей, що дозволило чітко сформулювати цілі проєкту); теорія множин (для математичної формалізації моделі системи у вигляді множин, функцій і відношень); методи декомпозиції (для розробки структурної декомпозиції робіт та організаційної структури проєкту; методи об'єктно-орієнтованого проєктування (для моделювання функціоналу та проєктування структури бази даних); а також метод експертних оцінок (для ідентифікації та кількісної оцінки проєктних ризиків).

Практична цінність полягає у розробці універсальної моделі процесів управління та необхідного інформаційного забезпечення, які можуть бути застосовані для ефективної реалізації інноваційних IT-проєктів у сфері PetTech. Результатом є сформована система управління, що гарантує прозорий обмін медичними даними та мінімізацію ризиків проєкту.

Апробація результатів роботи

1. Даніліна Т., Зюзюн В. Теоретичне обґрунтування розробки мобільного додатку "PETHEALTH: ветеринарний помічник" [Theoretical justification for the development of the mobile application "PETHEALTH: Veterinary Assistant"]. Proceedings of the 2st international scientific and practical conference «Information

Systems and Technology: Results and Prospects» (IST 2025), Kyiv, Ukraine, 2025, c. 198-200.

2. Ziuziun V., Danilina T. Conceptual and Mathematical Justification for the «PETHEALTH» Information System Development Project. Taurida Scientific Herald. Series: Technical Sciences, 2025, 4, part 1, pp. 94-102.

3. Ziuziun V., Danilina T. Features of Database Structure Development for the «Pethealth» Mobile Application. Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2025, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv.

4. Зюзюн В., Даніліна Т. Проєктування цифрової системи супроводу власників тварин: від аналізу вимог до реалізації інтерфейсу. Вісник Кременчуцького національного університету імені Михайла Остроградського, 2025, 6(155).

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ТА ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ТА ЖИТТЄЗДАТНОСТІ ПРОЄКТУ

1.1 Теоретичні основи управління проєктами

Ефективність проєкту «PetHealth» напряму залежить від вибору моделі управління, що буде відповідати викликам сучасної розробки програмного забезпечення. Історично в управлінні проєктами сформувалися дві фундаментальні, практично протилежні парадигми: предиктивна (класична) та адаптивна (гнучка).

Класична парадигма управління проєктами, формалізована у фундаментальних зводах знань, таких як PMBOK Guide (англ. A Guide to the Project Management Body of Knowledge), розроблена Інститутом Управління Проєктами (англ. Project Management Institute) [1], базується на предиктивному життєвому циклі. Цей підхід передбачає чітке послідовне виконання фаз: ініціація, планування, виконання, моніторинг та контроль, і закриття.

Найбільш відомою реалізацією цього підходу в IT є класична каскадна модель (англ. Waterfall) [2]. Її суть полягає в тому, що перехід до наступного етапу можливий лише після повного завершення попередньої. Детальне планування, що є ключовим елементом предиктивного підходу, активно використовує низку інструментів для контролю за часом і послідовністю робіт.

Ключовим інструментом серед них є діаграма Ганта [3], яка слугує для візуалізації розкладу проєкту, представляючи роботи у вигляді горизонтальних смуг на часовій шкалі. Для управління складними проєктами залучають методи мережевого планування, зокрема Метод критичного шляху (англ. Critical Path Method, CPM) та PERT (англ. Program Evaluation and Review Technique) [4]. Ці методи застосовуються для розрахунку загальної тривалості проєкту та ідентифікації критичного шляху – найдовшої послідовності завдань, що визначає мінімально необхідний термін його виконання. Таким чином, CPM та PERT забезпечують визначення критичних завдань та мінімізацію ризику зриву

термінів, надаючи керівництву необхідну інформацію для цільового фокусування ресурсів.

Ця модель є ефективною для проєктів зі стабільними, чітко визначеними на старті вимогами, наприклад, у будівництві чи серійному виробництві.

Однак при застосуванні до розробки програмного забезпечення, де вимоги є динамічними та часто невизначеними, цей каскадний підхід виявився джерелом системних проблем. Дослідження IT-галузі, зокрема відомі звіти «CHAOS Report» від Standish Group, протягом десятиліть підтверджували цю кризу. Сучасні аналітичні дані з цих звітів демонструють, що проєкти, керовані за методологією Waterfall, мають втричі нижчі шанси на успіх (13% успішних) порівняно з проєктами, що використовують Agile (42% успішних) [5]. Аналіз причин цих невдач чітко вказує на фундаментальні недоліки предиктивного підходу: нездатність управляти зміною вимог, нереалістичне початкове планування та слабке залучення користувачів. Саме ця нездатність класичних моделей адаптуватися до змін і стала «невирішеною частиною проблеми», що стимулювала пошук нових підходів.

Як відповідь на обмеження каскадної моделі, в IT-галузі набули поширення гнучкі (англ. Agile) методології. Її цінності були формалізовані у 2001 році в «Маніфесті гнучкої розробки» (англ. Agile Manifesto) [6]. Цей документ проголосив революційні на той час цінності: «люди та співпраця важливіші за процеси та інструменти» та «готовність до змін важливіша за слідування плану». Agile – це не одна методологія, а родина ітеративних підходів, що дозволяють команді швидко адаптуватися до змін та надавати цінність замовнику невеликими, але функціональними частинами (інкрементами).

Найпопулярнішим та найбільш структурованим фреймворком у цій родині є Scrum. У своїх основоположних роботах, зокрема «Scrum: Навчись робити вдвічі більше за менший час» Джеффа Сазерленда [7] та офіційному посібнику «The Scrum Guide» (написаному спільно з Кеном Швабером) [8], Scrum описується як фреймворк для управління проєктами зі складною динамікою.

Його ключові елементи – це короткі, фіксовані за часом цикли (спринти), чіткі ролі (власник продукту (англ. Product Owner), Scrum-майстер, команда розробників) та регулярні події (щоденний скрам, огляд спринту, ретроспектива). Саме Scrum дозволяє швидко створювати робочий продукт (англ. Minimum viable product, далі – MVP), отримувати зворотний зв'язок від ринку та негайно коригувати курс.

Іншим популярним гнучким методом є Kanban. На відміну від Scrum, що базується на часових спринтах, Kanban є потоковою моделлю. Його сучасні принципи описані в офіційному посібнику «Essential Kanban Condensed» Девіда Дж. Андерсона та Енді Кармайкла [9]. Цей метод фокусується на візуалізації робочого процесу (за допомогою Kanban-дошки) та обмеженні незавершеної роботи для оптимізації потоку. Це робить його високоефективним для команд, що працюють у режимі постійного потоку завдань, наприклад, технічна підтримка або безперервне вдосконалення вже існуючого продукту.

Варто зазначити, що з часом навіть інститут PMI, який традиційно підтримує класичний підхід, визнав цінність гнучких методологій. У своїх останніх редакціях PMBOK [1] та спеціальному виданні «Agile Practice Guide» [10] PMI описує гібридні моделі, що поєднують стабільність планування Waterfall з гнучкістю Agile.

Гібридний підхід являє собою поєднання предиктивного та адаптивного життєвих циклів. Найпоширеніша його форма передбачає використання класичного планування для фаз, що мають високу визначеність і стабільні вимоги, та гнучкого виконання для фази розробки, де вимоги є динамічними [11]. Цей підхід є високоефективним для великих, складних проєктів, які мають як стабільні компоненти (наприклад, інтеграція зі старими системами), так і компоненти, що вимагають інновацій та швидкого зворотного зв’язку (новітні функції).

Таким чином, для управління інноваційним проєктом «PetHealth» в умовах високої ринкової невизначеності та технічної складності інтеграції, Scrum є найбільш адекватною теоретичною та методологічною базою, оскільки

максимізує здатність проєкту до змін, що підтверджується статистичними даними успішності в IT-галузі.

1.2 Комплексний аналіз предметної області, ринкових тенденцій та конкурентного середовища

1.2.1 Аналіз предметної області та тенденцій ринку PetTech

Предметна область проєкту «PetHealth» знаходиться на перетині двох суміжних, але різних ринків: PetTech та mHealth. PetTech – це технології, орієнтовані на домашніх тварин зокрема, цифровізація догляду, тоді як mHealth – це мобільні технології, спрямовані на здоров’я людей, але їхній досвід є критично важливим для подібних проєктів.

Аналіз глобальних тенденцій, підтверджений низкою аналітичних звітів, свідчить про вибухове зростання ринку PetTech [12; 13]. Ключовим драйвером цього зростання аналітики одностайно називають глобальний тренд на «гуманізацію» тварин [14; 15]. Ця зміна у світогляді, за якої власники все частіше сприймають улюбленців як повноцінних членів родини, безпосередньо стимулює попит на високотехнологічні рішення. Зокрема, на ті, що є основою «PetHealth»: ветеринарну телемедицину, що дозволяє отримувати консультації дистанційно, та цифровізацію медичних карток, яка забезпечує централізований доступ до всієї історії здоров'я тварини. Це підтверджує, що ідея проєкту знаходиться у руслі потужних та довгострокових світових тенденцій.

Водночас, оскільки «PetHealth» є, по суті, медичною інформаційною системою, релевантним є аналіз досліджень із суміжної галузі mHealth. Наукові статті, присвячені розробці та впровадженню додатків для здоров'я людей, виявляють низку спільних «невирішених проблем», які з високою ймовірністю будуть притаманні і «PetHealth». Дослідники вказують на високі бар'єри для входу, пов'язані не стільки з програмуванням, скільки з інтеграцією та безпекою.

Технічна складність полягає в необхідності підключення до різнорідних медичних інформаційних систем (далі – МІС) кожної окремої клініки, що

вимагає розробки гнучкого та універсального API-інтерфейсу [16; 17]. Управлінський виклик пов'язаний із суворими вимогами до безпеки персональних даних, що вимагає від проєкту відповідності міжнародним та національним стандартам конфіденційності (наприклад, вимогам, подібним до GDPR в Європі) [18]. Ці дослідження є критично важливими, оскільки вони висвітлюють основні технічні та управлінські ризики проєкту.

Переходячи від глобальних тенденцій до локального контексту, аналіз українського ринку виявляє унікальну локальну ситуацію. Успішна цифровізація медицини для людей в Україні (Helsi) різко контрастує зі станом ветеринарної галузі. Незважаючи на законодавчі стимули до цифровізації, ринок залишається технологічно фрагментованим. Власники тварин стикаються з незручностями через відсутність єдиної цифрової картки, а обмін даними між ветеринарами залишається ускладненим. Саме цей розрив – між високими очікуваннями користувачів та реальним станом ринку – і є тією невирішеною проблемою та ринковою можливістю, яку має вирішити проєкт «PetHealth» шляхом створення єдиної інформаційної системи [19].

1.2.2 Аналіз зовнішнього середовища

Для глибинного розуміння контексту, в якому розробляється та впроваджується додаток «PetHealth», застосовано метод PEST-аналізу. Цей інструмент дозволяє систематизувати вплив політичних, економічних, соціальних та технологічних факторів на життєздатність проєкту [20].

Аналіз зовнішнього середовища є критично важливим етапом для проєкту, оскільки ринок ветеринарних послуг України наразі перебуває в стані активної трансформації. З одного боку, на нього впливають євроінтеграційні процеси та зміни у законодавстві [21], з іншого боку, проєкт реалізується в умовах воєнного стану, що створює суттєві економічні та безпекові виклики [22].

Результати PEST-аналізу представлені у табл. 1.1.

Таблиця 1.1

PEST-аналіз ветеринарного ринку України

Фактор	Опис	Вплив на проєкт
Політичні	Гармонізація законодавства України з ЄС (реєстрація тварин).	Можливість. Стимулює попит на цифрові ветеринарні паспорти.
	Запуск Єдиного держреєстру (вимога до централізації даних).	Можливість. Необхідність інтеграції з держреєстрами.
	Військово-політична ситуація (високі ризики для інвестицій).	Загроза. Впливає на кінцевий бюджет та терміни реалізації.
Економічні	Загальна економічна нестабільність та інфляція.	Загроза. Знижує купівельну спроможність клієнтів.
	Зростання середніх ставок IT-фахівців в Україні.	Загроза. Підвищує вартість (бюджет) розробки проєкту.
Соціальні	Глобальний тренд на «гуманізацію» тварин.	Можливість. Забезпечує стабільний та зростаючий попит на високоякісні цифрові рішення.
	Високі очікування до сервісу.	Можливість. Готовність аудиторії до використання мобільного додатку.
Технологічні	Масове використання смартфонів.	Можливість. Ідеальна технічна база для мобільного додатку.
	Фрагментованість IT-систем ветеринарних клінік (відсутність єдиного стандарту MIC).	Ризик. Висока складність інтеграції з локальними системами клінік.

Політико-правові фактори виступають драйвером змін. Законодавчі ініціативи щодо обов’язкової реєстрації тварин [23] та створення Єдиного державного реєстру [24] формують нормативну базу, яка робить цифровізацію неминучою. Для проєкту «PetHealth» це відкриває можливість стати зручним інтерфейсом для взаємодії власників тварин з державними реєстрами.

Соціальні фактори компенсують економічні ризики. Попри інфляцію та зниження купівельної спроможності населення (економічна загроза), глобальний тренд на гуманізацію ставлення до тварин забезпечує стійкий попит. Власники готові економити на інших витратах, але продовжують інвестувати в здоров’я своїх улюбленців, що знижує ризики незатребуваності продукту.

Технологічні виклики визначають архітектуру. Фактор фрагментованості наявних систем у ветеринарних клініках є критичним бар'єром. Це означає, що

на етапі проєктування архітектури мобільного додатку необхідно закласти гнучкі механізми інтеграції та можливість автономної роботи додатку без прив'язки до конкретної MIC клініки.

Аналіз PEST демонструє, що попри економічні та безпекові ризики, загальний вектор розвитку ринку (євроінтеграція, цифровізація, соціальні зміни) є сприятливим для запуску проєкту.

1.2.3 Аналіз конкурентного середовища

Для визначення ринкової ніші та формування унікальної торгової пропозиції проведено детальний аналіз конкурентного середовища. Ринок цифрових ветеринарних послуг в Україні є неоднорідним і представлений кількома категоріями гравців.

Одним з основних конкурентів є онлайн-платформа Vethub [25]. Vethub – це онлайн-платформа, спеціалізована на пошуку ветеринарних клінік та запису на приймання до лікарів. Вона є одним з лідерів у своєму сегменті на ринку цифрових рішень для ветеринарії, надаючи власникам домашніх тварин можливість зручно та швидко знаходити необхідні медичні послуги для своїх улюбленців. Основні функції та характеристики платформи:

1. Користувачі можуть знайти ветеринарні клініки в різних регіонах, фільтруючи за типом послуг, локацією та іншими параметрами.

2. Платформа дозволяє записуватися на приймання до лікаря онлайн, що значно спрощує процес організації візитів.

3. Vethub також надає можливість отримання консультацій у режимі онлайн.

4. Платформа має зрозумілий та зручний інтерфейс.

Обмеження: критичним недоліком Vethub є орієнтація переважно на веб-інтерфейс, відсутність повноцінного нативного мобільного додатку з високим рівнем UX/UI, а також обмежене географічне покриття (зосередженість на великих містах).

До непрямих конкурентів можна віднести такі ресурси, як маркетплейси та власні вебсайти ветеринарних клінік, які мають часткову функціональність для пошуку ветеринарних послуг або товарів.

OLX та Kabanchik – це загальні платформи для розміщення оголошень, які також використовується для пошуку послуг, зокрема ветеринарних та зоотоварів [26; 27].

Перевага: велика аудиторія та можливість пошуку за різноманітними категоріями.

Недолік: платформи не мають спеціалізованого інструменту для бронювання приймання, що обмежує зручність користувачів. Крім того, відсутність системи перевірки якості послуг та рейтингу фахівців ускладнює вибір надійного спеціаліста.

Багато ветеринарних клінік мають власні вебсайти, наприклад ветеринарна клініка Зоолюкс [28], де представлена інформація про послуги, лікарів та можливість запису на приймання.

Перевага: дозволяють прямий зв’язок з клінікою, що дає можливість контролювати комунікацію та якість сервісу, а також надають персоналізовану інформацію.

Недолік: не мають централізованого пошуку та об'єднання послуг від різних клінік, що змушує користувачів шукати інформацію на кількох ресурсах. Доступ до додаткових сервісів обмежений, що знижує зручність для кінцевих споживачів.

Результати порівняльного аналізу ключових конкурентів наведено у табл. 1.2.

Таблиця 1.2

Аналіз конкурентного середовища

Конкурент	**Тип**	**Наявність онлайн-запису**	**Єдина медична картка**	**Мобільний додаток**
1	**2**	**3**	**4**	**5**
Vethub	Прямий	Частково (через веб)	Так (у своїй системі)	Відсутній. Слабкий UX.

Закінчення табл. 1.2

1	2	3	4	5
OLX	Непрямий	Ні	Ні	Загальний (не профільний)
Kabanchik	Непрямий	Ні	Ні	Загальний (не профільний)
Сайти окремих клінік	Непрямий	Так (але тільки для себе)	Так (у своїй системі)	Частково

Унікальна торгова пропозиція (УТП) «PetHealth»: На відміну від конкурентів, «PetHealth» позиціонується як платформа з функцією єдиного медичного паспорта. Проєкт пропонує вирішення проблеми фрагментації даних, створюючи незалежну від конкретної клініки цифрову історію тварини, яка завжди доступна власнику у смартфоні.

1.3 Аналіз та класифікація зацікавлених сторін

Управління зацікавленими сторонами (стейкхолдерами) є критично важливим процесом, що визначає успіх проєкту на рівні із технічною реалізацією. Згідно з керівництвом PMBOK, ідентифікація зацікавлених сторін має відбуватися на ранніх етапах ініціації проєкту, щоб своєчасно врахувати їхні очікування, інтереси та потенційний вплив. Неврахування потреб ключових гравців може призвести до опору змінам, юридичних блокувань або створення продукту, що не відповідає ринковим реаліям.

Для систематизації учасників проєкту було використано універсальний підхід, що базується на аналізі трьох аспектів взаємодії: вигоди, внеску та впливу [29]. Це дозволило сформувати список стейкхолдерів та визначити стратегію комунікації з ними.

1. Отримувачі вигоди. Це група, для якої проєкт створює цінність.

– Власники домашніх тварин. Отримують зручний інструмент для догляду за улюбленцями.

– Ветеринарні клініки (Бізнес). Отримують автоматизацію процесів та канал залучення клієнтів.

– Власники продукту. Отримують прибуток та ринкову частку.

2. Учасники реалізації. Особи та групи, які безпосередньо створюють продукт:

– Проєктна команда (англ. Scrum Team). Розробники, QA, дизайнери.

– Технологічні партнери. Провайдери зовнішніх API, без яких неможливий функціонал (карти, оплати).

3. Об'єкти впливу. Групи, на чиє середовище проєкт впливає, змінюючи усталені процеси:

– Персонал клінік (Лікарі, адміністратори). Впровадження системи вимагає від них зміни звичок та навчання.

– Державні регулятори. Проєкт змінює підхід до обліку даних про тварин, що вимагає відповідності законодавству.

Для переходу від ідентифікації до взаємодії було розроблено план стратегій комунікації (табл. 1.3). У ньому для кожної групи визначено ключові очікування, потенційні ризики (що буде, якщо їхні інтереси ігнорувати) та конкретну стратегію залучення.

Таблиця 1.3

План стратегій комунікації зі стейкхолдерами

Зацікавлена сторона	**Ключове очікування (Інтерес)**	**Потенційний ризик взаємодії**	**Стратегія залучення**
1	**2**	**3**	**4**
Власники тварин (отримувачі вигоди)	Простий інтерфейс, швидкий запис, безпека даних улюбленця.	Низька активність використання через складний UX або недовіру до сервісу.	Активне бета-тестування, збір зворотного зв'язку через вбудовані форми, гейміфікація.
Власники клінік (отримувачі вигоди)	Зменшення «вікон» у розкладі, залучення нових платоспроможних клієнтів.	Відмова від партнерства через складність інтеграції або високу комісію.	Демонстрація економічної вигоди, надання пільгового періоду використання.
Проєктна команда (учасники реалізації)	Чіткі вимоги в беклозі, мінімізація раптових змін, сучасний технологічний стек.	Зрив дедлайнів спринтів, технічний борг, вигорання.	Щоденні стендапи, чітке планування спринтів, ретроспективи.

Закінчення табл. 1.3

1	2	3	4
Технологічні партнери (учасники реалізації)	Коректне використання API, своєчасна оплата тарифних планів.	Відключення сервісів (карт, SMS) через порушення лімітів або оплати.	Контроль SLA (угода) та автоматизація оплат.
Персонал клінік (об'єкти впливу)	Відсутність подвійної роботи (дублювання записів).	Саботаж впровадження, ігнорування системи, скарги на складність.	Проведення тренінгів, створення відеоінструкцій, техпідтримка 24/7.
Державні регулятори (об'єкти впливу)	Відповідність закону про захист персональних даних та ветеринарним нормам.	Блокування роботи сервісу, штрафні санкції, юридичні позови.	Юридичний аудит на етапі ТЗ, впровадження стандартів шифрування.

Аналіз зацікавлених сторін показав, що головне завдання проєкту – знайти баланс між інтересами власників тварин та працівників клінік. Власники тварин хочуть зручності та швидкості, тоді як лікарі та адміністратори часто бояться змін і не хочуть виконувати додаткову роботу в новій програмі. Тому успішність «PetHealth» залежить не лише від якості програмного коду, а й від ефективності навчання персоналу клінік роботі з системою. Без активного використання додатку лікарями він втрачає свою цінність. Також критично важливим є врахування вимог законодавства про захист даних для уникнення ускладнень із державними органами.

1.4 SWOT-аналіз проєкту

Для оцінки життєздатності концепції проєкту «PetHealth» та вибору стратегії його реалізації проведено SWOT-аналіз (англ. Strengths, Weaknesses, Opportunities, Threats). Цей етап базується на результатах попередніх досліджень: аналізі ринкової ніші, макроекономічних факторів, конкурентного середовища та очікувань стейкхолдерів. Метою аналізу є співставлення вимог ринку з потенціалом запропонованої ідеї.

Внутрішнє середовище оцінювалося з точки зору концептуальних переваг майбутнього продукту та ресурсних обмежень команди на старті (табл. 1.4).

Таблиця 1.4

Внутрішні фактори проєкту «PetHealth»

Код	**Сильні сторони (Strengths)**
S01	На відміну від конкурентів, проєкт пропонує централізовану систему обліку даних.
S02	Використання кросплатформного фреймворку React Native дозволяє охопити користувачів iOS та Android з єдиною кодовою базою.
S03	Використання Agile дозволяє команді швидко змінювати пріоритети під ринок.
S04	Архітектура, побудована з урахуванням вимог захисту персональних даних (шифрування, ролі доступу).
Код	**Слабкі сторони (Weaknesses)**
W01	На етапі запуску платформа не має підключених клінік та історії даних.
W02	Функціонування геолокації та оплат залежить від стабільності та тарифів третіх сторін (Google, платіжні шлюзи).
W03	Брак фінансування для масштабних маркетингових кампаній на старті.

Зовнішнє середовище оцінено на основі PEST-аналізу та аналізу стейкхолдерів (табл. 1.5).

Таблиця 1.5

Зовнішні фактори проєкту «PetHealth»

Код	**Можливості (Opportunities)**
O01	Впровадження обов’язкової реєстрації тварин та державних реєстрів створює попит на цифрові інструменти обліку.
O02	Глобальний тренд на гуманізацію тварин та збільшення витрат на їхнє здоров'я.
O03	Високий рівень проникнення смартфонів та звичка користувачів до mobile-first рішень.
O04	Зростання популярності дистанційних консультацій як вільної ніші.
O05	Можливість інтеграції з зоомагазинами та страховими компаніями.
Код	**Загрози (Threats)**
T01	Високий ризик опору персоналу клінік при впровадженні нових програмних продуктів.
T02	Зниження платоспроможності населення та інфляція можуть зменшити попит на платні послуги.
T03	Зростання кількості атак на медичні сервіси та витоків даних.
T04	Ризик копіювання функціонала великими гравцями.

Системне використання SWOT-аналізу переходить від статичної констатації факторів до розробки динамічних стратегічних рішень шляхом попарного зіставлення його елементів. Ця методологія, відома як TOWS-матриця або матриця рішень, дозволяє інтегрувати внутрішні можливості проєкту з зовнішніми умовами. Результатом зіставлення є формування чотирьох груп стратегій, які визначають поведінку проєкту на ринку PetTech та необхідні управлінські дії: стратегія прориву (S+O), стратегія перехідного періоду (S+T та W+O) та стратегія виживання (W+T) (табл. 1.6) [30].

Таблиця 1.6

Матриця рішень: стратегічні напрямки проєкту «PetHealth»

Комбінація	**Стратегічна назва**	**Ключова стратегічна дія для «PetHealth»**
1	**2**	**3**
S01 + S02 + O01	Стратегія прориву (S + O)	SO1. Використати готову централізовану кросплатформну систему для миттєвого задоволення державного попиту на інструменти реєстрації тварин, ставши першим масовим рішенням на ринку.
S01 + O04		SO2. На базі централізованої картки реалізувати функціонал дистанційних консультацій, закриваючи вільну нішу та монетизуючи тренд на турботу про здоров'я тварин.
S03 + T01	Стратегія перехідного періоду (S + T)	ST1. Використати гнучкість Agile для швидкої адаптації інтерфейсу під потреби персоналу клінік, мінімізуючи опір впровадженню через зручність використання.
S04 + T03		ST2. Зробити архітектурну захищеність та шифрування даних ключовою маркетинговою перевагою для протидії кіберзагрозам та завоювання довіри клієнтів.
W01 + W03 + O05	Стратегія перехідного періоду (W + O)	WO1. Компенсувати відсутність бази та маркетингового бюджету через інтеграцію зі страховими компаніями та зоомагазинами, які нададуть доступ до своєї аудиторії.
W01 + O02		WO2. Забезпечити набір первинної бази користувачів за рахунок соціальних механізмів та високої емоційної залученості власників, мінімізуючи витрати на маркетинг.

Закінчення табл. 1.6

1	2	3
W02 + T02	Стратегія виживання (W+T)	WT1. Реліз MVP та жорстке бюджетування. Скласти план заходів для контролю витрат та випустити MVP з мінімальною інтеграцією, щоб забезпечити життєздатність в умовах економічної нестабільності.
W03 + T04		WT2. Замість прямої конкуренції з гігантами, зосередитися на унікальній ніші, поступово нарощуючи функціонал.

Аналіз показав, що проєкт має високий потенціал завдяки відповідності глобальним трендам (O2, O3). Однак критичними зонами є «холодний старт» без бази партнерів (W1) та опір персоналу клінік (T1). Тому стратегією проєкту обрано поетапний запуск із фокусом на створенні цінності для кінцевого користувача (власника тварини), що дозволить згодом залучити клініки через сформований попит.

1.5 Побудова дерева проблем та дерева цілей проєкту

Дерево проблем є інструментом візуалізації, який допомагає команді чітко розрізняти центральну проблему (стовбур), кореневі причини (коріння) та негативні наслідки (гілки). Аналіз починається з ідентифікації основного негативного стану (проблеми), який має бути вирішений [31]. Далі проводиться декомпозиція:

– Рівень 1 (Коріння) – це першопричини. Вони є вихідними точками для подальшого проєктування системи.

– Рівень 2 (Стовбур/Центр) – це безпосередній об'єкт уваги проєкту.

– Рівень 3 (Гілки/Наслідки) – це кінцеві негативні наслідки, яких ми прагнемо уникнути.

Елементи дерева проблем проєкту зображено на рис. 1.1.

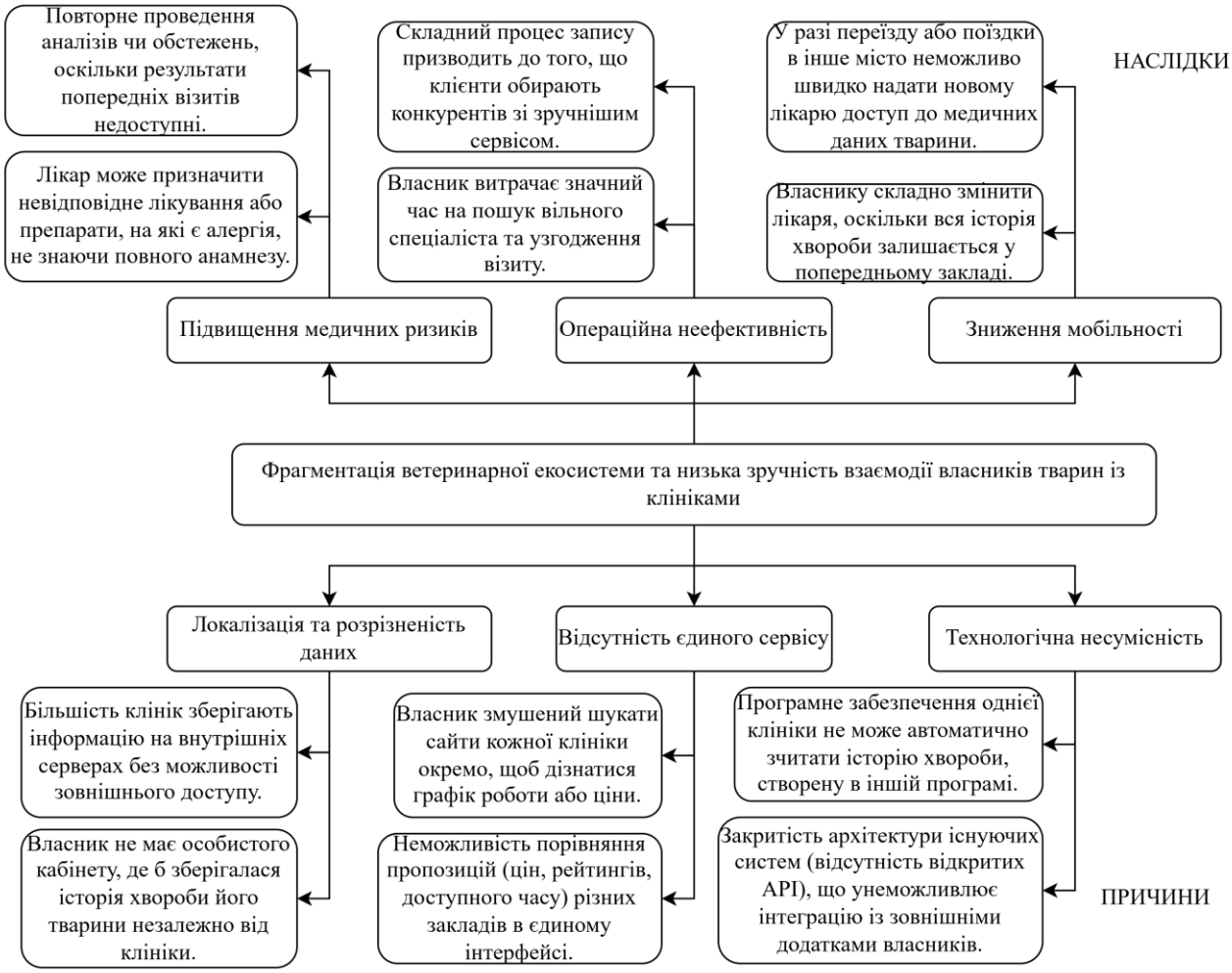

Рис. 1.1. Елементи дерева проблем проєкту «PetHealth»

Візуалізована схема демонструє причинно-наслідкові зв'язки, що визначають необхідність створення системи «PetHealth». Ключовим елементом діаграми визначено фрагментацію ветеринарної екосистеми та низьку зручність взаємодії власників тварин із клініками. Цей негативний стан сформувався під впливом трьох груп фундаментальних причин.

По-перше, спостерігається локалізація та розрізненість даних, оскільки більшість клінік зберігають інформацію виключно на внутрішніх серверах без можливості зовнішнього доступу, через що власник не має особистого кабінету з історією хвороби своєї тварини.

По-друге, ринок характеризується відсутністю єдиного сервісу, що змушує власників шукати сайти кожної клініки окремо та унеможливлює порівняння пропозицій, цін чи рейтингів у єдиному інтерфейсі.

По-третє, існує проблема технологічної несумісності, за якої програмне забезпечення однієї клініки не здатне автоматично зчитати дані з іншої програми через закритість архітектури та відсутність відкритих API .

Сукупність цих причин призводить до низки негативних наслідків, які впливають як на здоров'я тварин, так і на комфорт власників. Це спричиняє підвищення медичних ризиків, коли через відсутність повного анамнезу лікар може призначити невідповідне лікування або повторні аналізи. Також виникає операційна неефективність, коли власник витрачає значний час на узгодження візиту, а складний процес запису змушує клієнтів обирати конкурентів. Крім того, відбувається зниження мобільності, оскільки при переїзді або зміні лікаря неможливо швидко передати новому спеціалісту доступ до медичних даних, що залишаються у попередньому закладі .

Дерево цілей є наступним логічним кроком у системному аналізі. Воно формується шляхом перетворення кожного негативного твердження у дереві проблем на позитивне, досяжне твердження. Цей інструмент демонструє, як усунення кореневих причин призведе до бажаних наслідків (рис. 1.2).

Схема візуалізує ієрархію завдань, необхідних для досягнення бажаного стану системи. Генеральною метою проєкту визначено створення єдиної цифрової екосистеми «PetHealth» для централізованого управління ветеринарними даними. Досягнення цієї мети базується на реалізації трьох стратегічних напрямів, кожен з яких усуває відповідну групу кореневих причин проблем.

Першим напрямом є забезпечення централізації та доступності даних, що передбачає впровадження захищеної бази даних для зберігання інформації незалежно від локальних серверів клінік. Це дозволить надати власнику інструмент для акумуляції повної історії хвороби тварини в єдиному профілі, незалежно від того, де саме здійснювалося обслуговування.

Другий напрям фокусується на створенні єдиного сервісу, що реалізується через можливість пошуку клінік, лікарів та послуг на єдиній платформі без необхідності моніторингу різних сайтів. Крім того, це забезпечить можливість

порівняння пропозицій (цін, рейтингів, часу) та швидкого онлайн-бронювання в єдиному інтерфейсі.

Третім напрямом є забезпечення технологічної сумісності, що досягається шляхом розробки механізмів зчитування та відображення історії хвороби з різних програмних середовищ, а також використання технологій API для інтеграції зовнішнього додатку власника з наявними системами клінік.

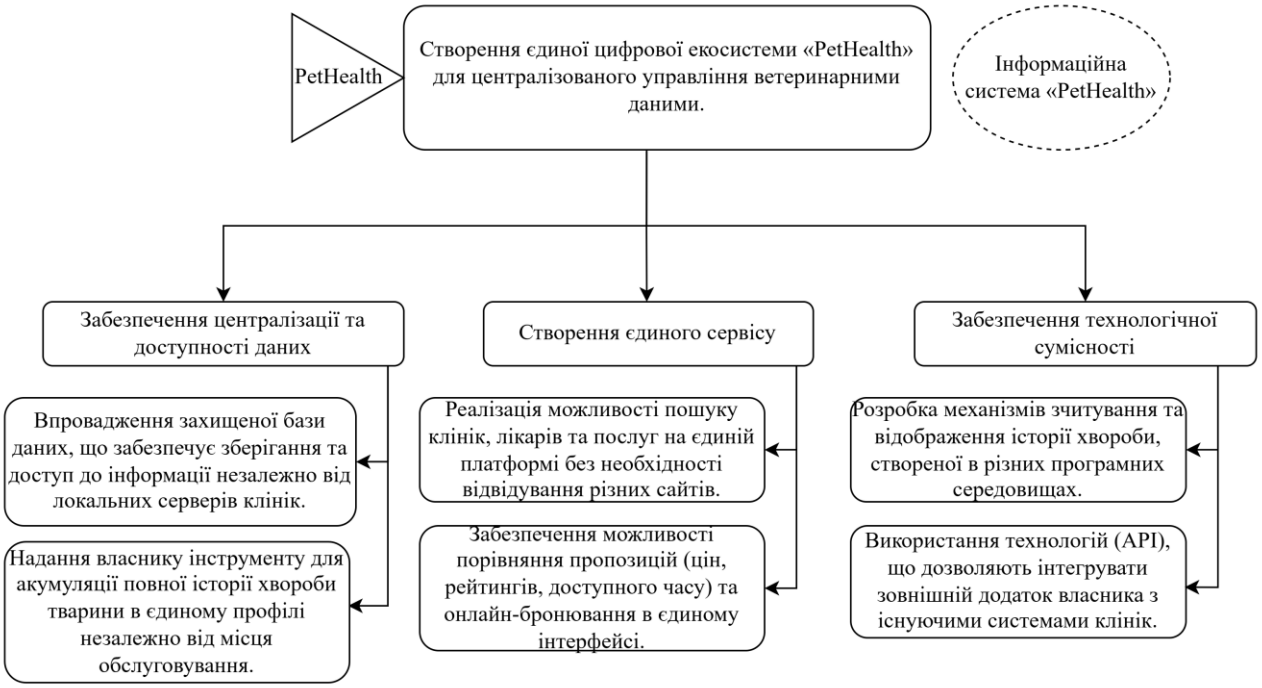

Рис. 1.2. Елементи дерева цілей проєкту «PetHealth»

Підсумовуючи, проведений аналіз підтвердив актуальність створення єдиної екосистеми та дозволив виокремити пріоритетні напрями розробки. Визначені причинно-наслідкові зв’язки гарантують, що реалізація проєкту буде спрямована на усунення кореневих причин неефективності ринку, а не лише на боротьбу з їх наслідками, що є основою для успішного планування проєкту.

1.6 Формулювання технічного завдання на розробку у вигляді паспорту проєкту

На основі проведених досліджень сформовано паспорт проєкту, який визначає ключові параметри, зміст та очікувані результати розробки [32].

Назва проєкту: Розробка мобільного додатку «PetHealth».

Мета проєкту: Створення кросплатформного мобільного додатку для автоматизації взаємодії між власниками тварин і ветеринарними клініками, що забезпечує централізоване зберігання медичних даних, онлайн-бронювання послуг та підвищення ефективності лікування.

Зацікавлені сторони: Власники домашніх тварин (користувачі), ветеринарні клініки та приватні лікарі (партнери), інвестори.

Цілі проєкту: Розробити та запустити MVP мобільного додатку (iOS/Android) з функціями медкарти та онлайн-запису протягом 6 місяців; Залучити 1000 активних користувачів та 20 клінік-партнерів у м. Києві протягом першого кварталу після релізу; Скоротити середній час запису на прийом до 3 хвилин та забезпечити доступність історії хвороби 24/7.

Задачі проєкту:

1. Виконати аналіз існуючих рішень, сформувати вимоги та беклог продукту.

2. Спроєктувати архітектуру бази даних та серверної частини.

3. Розробити UX/UI дизайн та клієнтську частину.

4. Реалізувати інтеграцію з платіжними системами та картографічними сервісами.

5. Провести комплексне тестування та розмістити додаток у App Store/Google Play.

Результат проєкту: Функціонуюча екосистема, що забезпечує прозорий обмін медичними даними, мінімізацію лікарських помилок завдяки повному анамнезу та оптимізацію завантаженості ветеринарних лікарів.

Життєвий цикл: Проєкт реалізується за ітеративно-інкрементальною моделлю (Scrum) і складається з 5 фаз:

Першою фазою є ініціація та концептуалізація, яка закладає фундамент усього проєкту. На цьому етапі проводиться глибокий аналіз предметної області та ринку ветеринарних послуг, вивчаються існуючі рішення конкурентів та ідентифікуються ключові проблеми користувачів. Паралельно відбувається

визначення зацікавлених сторін, формування проєктної команди з розподілом ролей, а також створення початкового беклогу продукту, який містить пріоритезований список функціональних вимог.

Наступною є фаза планування та проєктування, метою якої є технічна підготовка до розробки та мінімізація ризиків. Цей етап включає визначення архітектури системи, вибір технологічного стеку, а також детальне проєктування бази даних, включаючи розробку концептуальної, логічної та фізичної моделей. Важливою складовою є UI/UX дизайн, що передбачає створення інформаційної архітектури та інтерактивних прототипів інтерфейсу для перевірки зручності користування ще до написання коду.

Центральне місце займає фаза ітеративної розробки, яка проходить циклічно у вигляді спринтів. Кожна ітерація включає планування завдань, безпосередню реалізацію програмного коду серверної та клієнтської частин, а також проведення регулярних нарад для синхронізації команди. Завершується кожен спринт оглядом готового інкременту продукту та ретроспективою, що дозволяє команді аналізувати ефективність роботи та вносити необхідні корективи в процес.

Після завершення основної розробки настає фаза впровадження та стабілізації. Вона передбачає проведення комплексного тестування системи, включаючи функціональні та навантажувальні тести, для виявлення та виправлення критичних помилок. На цьому ж етапі здійснюється розгортання серверної частини на хмарній інфраструктурі, публікація мобільного додатку в магазинах Google Play та App Store, а також підготовка документації для користувачів та адміністраторів.

Завершальною є фаза підтримки та розвитку, яка триває протягом усього періоду експлуатації системи. Вона включає постійний моніторинг стабільності роботи серверів, забезпечення технічної підтримки користувачів та збір зворотного зв'язку. На основі аналізу отриманих даних та метрик ефективності приймаються рішення щодо подальшого вдосконалення функціоналу та випуску оновлених версій програмного продукту.

Проведений системний аналіз виявив, що існуюча фрагментація ринку та технологічна несумісність систем вимагають створення єдиної цифрової екосистеми. Для досягнення мети проєкту та успішної технічної реалізації системи «PetHealth» необхідно вирішити комплекс взаємопов'язаних завдань, які визначають структуру подальшого дослідження:

1. Розробити концептуальну модель системи для визначення її меж та зв'язків із зовнішнім середовищем.
2. Виконати математичну постановку задачі на основі теорії множин для формального опису інформаційних потоків.
3. Обґрунтувати вибір методології управління проєктом.
4. Розробити структурну декомпозицію робіт та організаційну структуру команди.
5. Сформувати детальні функціональні вимоги та беклог продукту.
6. Розробити стратегію управління ризиками та розрахувати бюджет проєкту для забезпечення його економічної ефективності.
7. Спроєктувати архітектуру програмного забезпечення та розробити структуру бази даних.
8. Розробити дизайн інтерфейсів користувача, що відповідає вимогам зручності.
9. Сформувати план забезпечення якості.

Сформоване технічне завдання у вигляді паспорту проєкту та визначені задачі дослідження створюють логічну основу для переходу до етапів математичного моделювання, організаційного планування та безпосередньої програмної розробки.

РОЗДІЛ 2. МАТЕМАТИЧНА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

2.1. Розробка концептуальної моделі мобільного додатку

Розробка будь-якого програмного забезпечення, зокрема мобільного додатку, починається з побудови концептуальної моделі, яка дозволяє узагальнено представити структуру системи, її основні компоненти, зв’язки між ними та взаємодію із зовнішнім середовищем. Концептуальна модель є важливою складовою етапу проєктування, оскільки забезпечує цілісне бачення системи ще до безпосередньої реалізації функціоналу [32].

Розроблену концептуальну модель мобільного додатку «PetHealth», який призначений для полегшення взаємодії між власниками домашніх тварин і ветеринарними клініками, зображено на рис. 2.1. Система дозволяє здійснювати запис на приймання, вести електронні медичні картки тварин, отримувати онлайн-консультації, а також здійснювати інші дії, пов’язані з доглядом за домашніми улюбленцями [33].

У процесі розробки концептуальної моделі було виконано аналіз системи, визначено її структуру, надсистему, підсистеми, зовнішні зв’язки та функціональні компоненти

1. Система:

Мобільний додаток «PetHealth».

2. Аналіз системи з фізичної точки зору:

Надсистема: власники тварин, ветеринарні клініки, платіжни системи.

Підсистеми:

– Мобільний додаток: основна підсистема для запису на приймання, пошуку клінік, взаємодії з ветеринарами та отримання медичних рекомендацій.

– Власники тварин: особи, що реєструють своїх тварин, шукають ветклініки, записуються на приймання.

– Ветеринарні клініки: приймають записи, ведуть медичні карти тварин, надають консультації.

– Адміністративна панель: спеціалізована підсистема для адміністраторів системи, яка дозволяє здійснювати керування користувачами, контролювати зміст додатку, здійснювати моніторинг технічного стану системи, а також налаштовувати загальні параметри роботи сервісу.

3. Визначення зовнішніх зв’язків системи з надсистемою:

Зовнішні фактори впливу:

– Власники тварин: оформляють онлайн-запис до лікаря, задають питання, отримують консультації через додаток.

– Ветеринарні клініки: приймають онлайн-записи, ведуть записи про тварин.

4. Перелік підсистем:

Мобільний додаток:

- Інтерфейс користувача.
- Система онлайн-запису.
- Медичні картки тварин.
- Модуль онлайн-консультацій з лікарем.
- Пошук лікарів та ветклінік.
- Функція оплати послуг.

Клієнтський акаунт:

- Реєстрація тварин.
- Перегляд медичної картки.
- Створення та управління онлайн-записами.
- Налаштування обліковим записом.

Ветеринарні клініки:

- Система управління записами.
- Ведення медичних карток тварин.
- Модуль взаємодії з користувачами.
- Інтерфейс управління.

Адміністративна панель:

- Управління користувачами.
- Управління контентом.
- Моніторинг системи.
- Управління налаштуваннями.

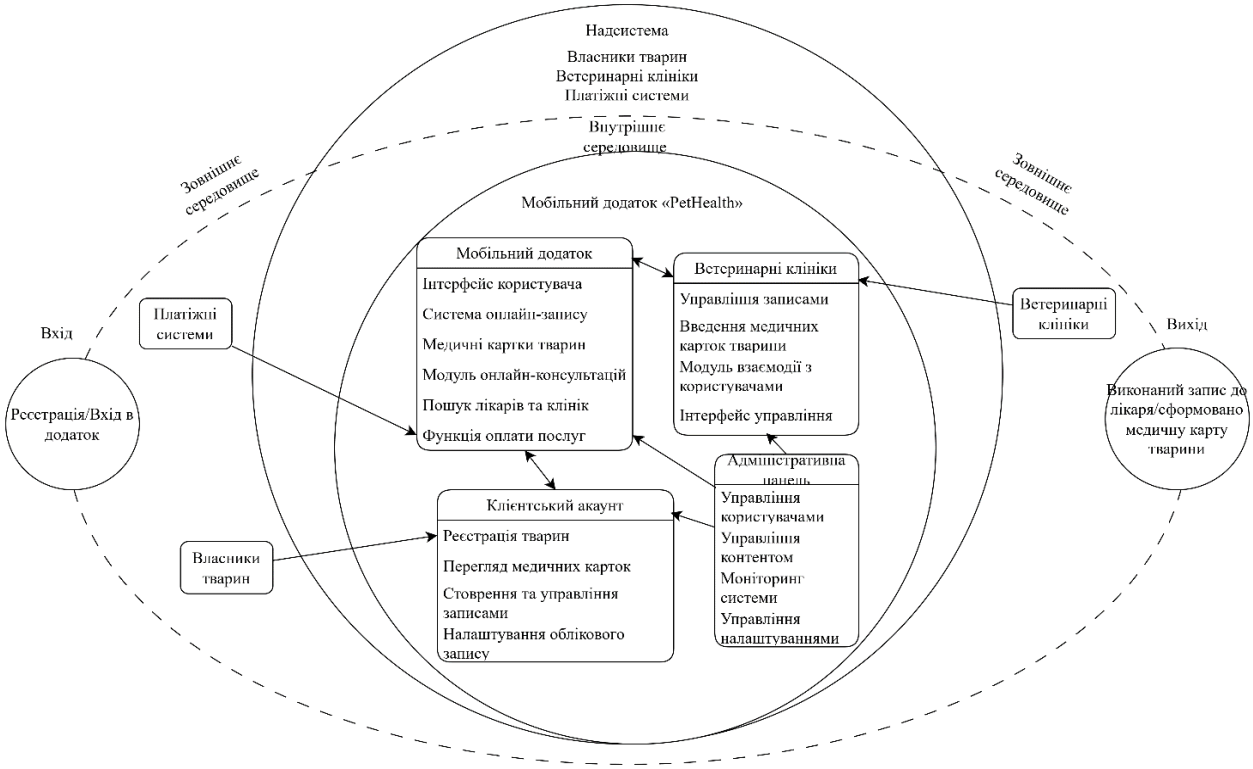

Рис. 2.1. Концептуальна модель мобільного додатку «PetHealth»

Як видно з рис. 2.1, функціонування системи «PetHealth» реалізується через взаємодію внутрішнього середовища з надсистемою. Процес починається з входу (реєстрації або авторизації користувача), що надає доступ до функціональних модулів мобільного додатку. Центральним елементом моделі виступає підсистема «Мобільний додаток», яка забезпечує двосторонній обмін даними між клієнтським акаунтом та системами ветеринарних клінік, а також інтеграцію із зовнішніми платіжними сервісами.

Результатом роботи системи (виходом) є сформований запис до лікаря та оновлена медична картка тварини. Запропонована концептуальна модель дозволяє чітко розмежувати зони відповідальності кожної підсистеми та візуалізувати інформаційні потоки, що є необхідною умовою для подальшої деталізації вимог.

2.2 Формалізація математичних моделей та постановка задачі в математичному вигляді

Формалізація математичних моделей для мобільного додатку «PetHealth» передбачає опис системи у вигляді множин, функцій і зв’язків між її компонентами. Такий підхід базується на методах теорії множин та моделювання інформаційних систем, що дає змогу чітко визначити структуру даних, логіку взаємодії та основні функціональні залежності.

Система «PetHealth» розглядається як сукупність взаємодіючих об'єктів. Для коректного моделювання необхідно виділити базові множини, що відповідають реальним сутностям процесу надання ветеринарних послуг [33].

Основні множини:

1. Множина зареєстрованих користувачів мобільного додатку:

$$U = \{u_1, u_2, \ldots, u_i\}, \tag{2.1}$$

де u_i – окремий користувач додатку (власник тварини), який пройшов процедуру автентифікації та має унікальний ідентифікатор у системі. Кожен елемент множини характеризується набором персональних даних (ім'я, контакти, налаштування профілю).

2. Множина тварин, зареєстрованих у системі:

$$P = \{p_1, p_2, \ldots, p_j\}, \tag{2.2}$$

де p_j – унікальна картка тварини, що містить дані про вид, породу, вік та фізіологічні особливості. Кожна тварина закріплена за конкретним користувачем, що формує ієрархію володіння.

3. Множина ветеринарних клінік:

$$C = \{c_1, c_2, \ldots, c_k\}, \tag{2.3}$$

де c_k – сутність, що описує конкретний медичний заклад. Атрибутами елементів цієї множини є геолокація, графік роботи, перелік доступного обладнання та контактна інформація.

4. Множина лікарів-ветеринарів:

$$V = \{v_1, v_2, \ldots, v_m\}, \tag{2.4}$$

де v_m – кваліфікований спеціаліст, який надає послуги. Кожен ветеринар асоційований з певною клінікою та має визначену спеціалізацію, що впливає на типи послуг, які він може надавати.

5. Множина записів на приймання:

$$A = \{a_1, a_2, \ldots, a_t\}, \tag{2.5}$$

де $a_i = (u_i, p_j, c_k, v_m, t)$ – факт реєстрації візиту користувача до лікаря на фіксований часовий слот t.

6. Множина медичних карток по кожній тварині:

$$M = \{m_1, m_2, \ldots, m_z\}, \tag{2.6}$$

де m_z – електронна історія хвороби, що акумулює результати оглядів, вакцинацій та призначень.

7. Множина онлайн-консультації:

$$O = \{o_1, o_2, \ldots, o_q\}, \tag{2.7}$$

де $o_i = (u_i, v_m, t, dialog)$ – запис про проведену або заплановану дистанційну консультацію, яка включає текстовий чат або відеозв'язок.

8. Множина транзакцій оплати:

$$S = \{s_1, s_2, \ldots, s_r\}, \tag{2.8}$$

де s_r – фінансова операція, що фіксує факт оплати послуг. Кожна транзакція має статус (успішна, відхилена, очікується) та часову мітку.

9. Множина відгуків:

$$R = \{r_1, r_2, \ldots, r_n\}, \tag{2.9}$$

де r_n – суб'єктивна оцінка якості послуг, залишена користувачем. Вона впливає на рейтинг лікаря та клініки в системі.

10. Множина адміністраторів:

$$Adm = \{adm_1, adm_2, \ldots, adm_h\}, \quad (2.10)$$

де adm_h – користувачі з розширеними правами доступу, які відповідають за модерацію контенту та керування довідниками системи.

Елементи системи пов'язані між собою логічним відношеннями, які визначають їх взаємодію та логіку роботи додатку. Ці зв'язки забезпечують цілісність і функціональність системи.

1. Зв'язок між користувачем та тваринами полягає у тому, що кожен користувач $u_i \in U$ може мати одну або декілька тварин $p_j \in P$, які закріплені за його обліковим записом:

$$u_i \to \{p_{j1}, p_{j2}, \ldots p_{jn}\} \quad (2.11)$$

2. Зв'язок між ветеринаром та клінікою визначає, що кожен ветеринар $v_m \in V$ працює у певній клініці $c_k \in C$:

$$v_m \to c_k \quad (2.12)$$

3. Зв'язок між твариною та медичною карткою описує відповідність кожної тварини $p_j \in P$ до однієї медичної картки $m_j \in M$:

$$p_j \leftrightarrow m_j \quad (2.13)$$

4. Зв'язок між записом на приймання та лікарем визначає, що кожен запис $a_i \in A$ прив'язаний до конкретного ветеринара $v_m \in V$:

$$a_i \to v_m \quad (2.14)$$

Система виконує набір ключових функцій, які забезпечують її основний функціонал. Ці функції представлені у вигляді математичних операцій, що описують дії користувачів, ветеринарів і адміністраторів.

1. Запис на приймання.

Функція f_{app} створює новий запис у множині A за умови доступності слоту:

$$f_{app}(u_i, p_j, v_m, t) \rightarrow a_{new} \in A, \quad (2.15)$$

де t – обране користувачем значення часу візиту.

2. Функція оплати.

Функція f_{pay} генерує транзакцію s для оплати запису a або консультації o:

$$f_{pay}(a_i, amount) \rightarrow s_{new} \in S, \quad (2.16)$$

3. Функція модерації.

Функція f_{mod} змінює статус відгуку r (публікація або видалення):

$$f_{mod}(adm_h, r_n) \rightarrow status \in \{active, deleted\}, \quad (2.17)$$

4. Функція консультації.

Функція f_{chat} ініціює сесію чату між власником та лікарем:

$$f_{chat}(u_i, v_m) \rightarrow o_{new} \in O, \quad (2.18)$$

Для забезпечення цілісності даних, уникнення конфліктів та дотримання бізнес-логіки функціонування системи «PetHealth» накладаються такі логічні обмеження:

1. Уникнення перетину в записах.

Один ветеринар не може мати більше одного запису на той самий квант часу t. Це забезпечує унікальність слотів у розкладі:

$$a_i = (u_1, p_1, c_1, v_m, t) \Rightarrow \nexists a_k \neq a_i : a_k = (u_2, p_2, c_2, v_m, t) \quad (2.19)$$

2. Умова здійснення оплати.

Фінансова транзакція s_r може бути ініційована лише для існуючого запису a_i який має статус підтвердженого:

$$s_r \in S \Leftrightarrow (a_i \in A \wedge Status(a_i) = 'Confirmed') \quad (2.20)$$

3. Конфіденційність медичних даних.

Перегляд медичної картки m_j дозволений лише власнику тварини u_{owner} або лікуючому ветеринару v_{doctor}:

$$Access(m_j) \in \{u_{owner}, v_{doctor}\} \quad (2.21)$$

4. Валідація відгуків.

Користувач може залишити відгук r_n про лікаря лише за умови наявності завершеного візиту a_i до цього спеціаліста:

$$Create(r_n) \Rightarrow \exists a_i \in A : (User(a_i) = User(r_n) \wedge Status(a_i) = 'Completed') \quad (2.22)$$

Описана математична модель дозволяє чітко систематизувати логіку роботи додатку та взаємодію його компонентів. Таке представлення є необхідною базою для переходу до етапу програмування, оскільки воно спрощує проектування бази даних, допомагає уникнути логічних помилок та створює умови для подальшої оптимізації системи.

2.3. Використання методів ШІ та моделювання розроблених моделей

У сучасних інформаційних системах все частіше застосовуються методи штучного інтелекту, зокрема технології обробки природної мови (англ. Natural Language Processing, далі – NLP), для покращення взаємодії користувача з додатком [34]. У рамках розвитку мобільного додатку «PetHealth» пропонується запровадити інтелектуальний чат-бот [35] VetBot, який базується на NLP. Цей підхід забезпечує гнучку обробку запитів, природну взаємодію та персоналізований досвід, що зменшує навантаження на службу підтримки та підвищує залученість користувачів.

Для подальшого розвитку VetBot передбачається використання сучасних NLP-рішень, зокрема таких як Rasa [36] або Dialogflow [37], що забезпечують аналіз запитів користувача та генерацію відповідей у режимі реального часу.

Основні функції VetBot реалізуються через такі механізми:

1. Розпізнавання намірів (англ. Intent Recognition) – виявлення мети користувача на основі аналізу змісту повідомлення.

2. Виділення сутностей (англ. Entity Extraction) – виявлення імен, дат, назв клінік, ідентифікаторів тварин тощо.

3. Аналіз контексту – урахування попередніх звернень у рамках сесії для формування більш релевантної відповіді.

Для реалізації VetBot пропонується використати попередньо навчену модель, наприклад, BERT [38], адаптовану до потреб додатку через додаткове навчання на даних про догляд за тваринами, ветеринарні послуги та взаємодію з користувачами. База знань VetBot включає інформацію про здоров'я тварин, графіки вакцинації, рекомендації ветеринарів і дані з профілів користувачів.

VetBot на основі NLP пропонує розширені функції, які роблять його ефективним інструментом для користувачів додатку «PetHealth»:

1. *Обробка різноманітних запитів*. Чат-бот може відповідати на складні та нестандартні запитання, наприклад, «Мій собака чухається, що це може бути?» чи «Як підготувати цуценя до візиту до ветеринара?». Він пропонує відповідні дії, такі як звернення до ветеринара або базові поради.

2. *Персоналізовані рекомендації*. Аналізуючи дані профілю тварини (порода, вік, історія здоров'я), VetBot надає індивідуальні поради, наприклад, рекомендації щодо годування для конкретної породи чи нагадування про дегельмінтизацію.

3. *Інтерактивність і залучення*. Чат-бот пропонує вікторини, цікаві факти про тварин, гумористичні повідомлення чи міні-ігри, адаптовані до уподобань користувача.

4. *Багатомовна підтримка*. VetBot може обробляти запити різними мовами, такими як українська чи англійська, що робить додаток доступним для ширшої аудиторії.

5. *Інтеграція з профілем*. Доступ до даних користувача дозволяє надавати контекстно релевантні відповіді, наприклад, нагадування про щеплення для конкретної тварини.

Використання технологій обробки природної мови в чат-боті VetBot дозволяє значно покращити взаємодію з користувачем. Однією з основних

переваг є здатність системи розуміти різні формулювання запитів. Це забезпечує гнучкість у спілкуванні, оскільки бот може реагувати не лише на заздалегідь визначені фрази, а й на нестандартні чи неструктуровані повідомлення.

Ще однією перевагою є природність відповідей. Завдяки NLP користувач отримує інформацію у звичній формі, що створює відчуття живого спілкування. Крім того, система може покращувати свої результати на основі накопичених даних, що дає змогу адаптувати VetBot до нових ситуацій та підвищувати точність відповідей з часом.

Разом із тим, реалізація подібного підходу вимагає значних ресурсів. Створення та підтримка NLP-моделей передбачає використання потужних обчислювальних систем, а також залучення фахівців зі спеціальними знаннями. Іншою проблемою є можливість виникнення помилок у випадках, коли запит сформульовано нечітко або містить кілька значень. У таких ситуаціях система може неправильно зрозуміти намір користувача, тому важливо передбачити механізми уточнення або повторного запиту.

Зважаючи на вищезазначене, впровадження NLP у VetBot є перспективним напрямом, що потребує комплексного підходу до проєктування, технічної підтримки та поступового розвитку.

Для ілюстрації функціональності інтелектуального чат-бота VetBot створено концептуальний макет, що демонструє базову логіку взаємодії користувача з віртуальним помічником (див. рис. 2.2).

У запропонованому дизайні передбачено, що чат-бот VetBot вітатиме користувача у дружній, напівігровій формі та пропонуватиме низку основних опцій: відповіді на поширені запитання, персоналізовані рекомендації з догляду за тваринами, цікаві факти, міні-ігри, а також можливість прямого зв’язку з ветеринаром. Інтерфейс VetBot спроєктовано з урахуванням принципів доступності, інтуїтивної навігації та емоційної залученості користувача. Особливу увагу приділено адаптивності під різні типи пристроїв і візуальній гармонії з основним дизайном додатку «PetHealth». Такий підхід забезпечує

комфортну взаємодію незалежно від технічного рівня користувача, сприяє довірі до системи та підвищує загальну ефективність цифрового сервісу.

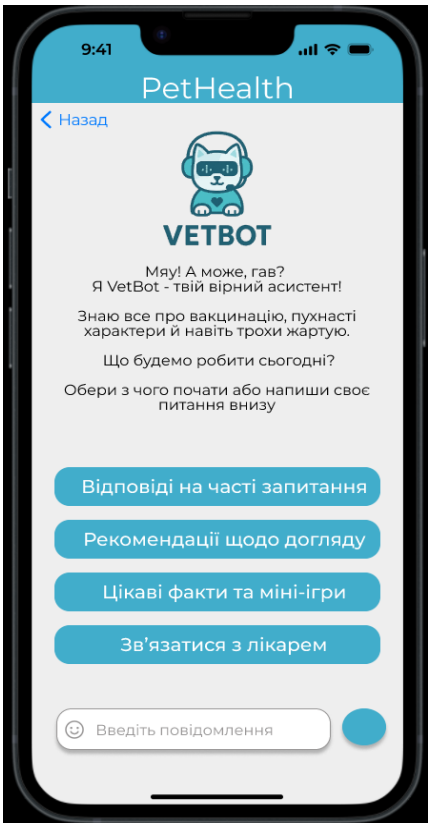

Рис. 2.2. Інтерфейс чат-бота VetBot у додатку «PetHealth»

У розділі розроблено концептуальну модель системи «PetHealth», яка визначає структуру мобільного додатку, його підсистеми та характер взаємодії із зовнішнім середовищем. Здійснено математичну постановку задачі на основі теорії множин: формалізовано ключові об'єкти (користувачі, тварини, лікарі, записи) та логічні обмеження, що забезпечують цілісність даних.

Окрему увагу приділено обґрунтуванню використання методів штучного інтелекту. Запропоновано інтеграцію інтелектуального асистента VetBot на базі NLP, що дозволяє автоматизувати первинну консультацію та підвищити зручність користування сервісом. Розроблені концептуальна та математична моделі створюють необхідне теоретичне підґрунтя для переходу до наступного етапу – розробки інформаційного забезпечення та планування управління проєктом.

РОЗДІЛ 3. РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ПРОЄКТУ

3.1 Вибір методології управління проєктом

З огляду на необхідність роботи в умовах високої невизначеності, для реалізації проєкту обрано гнучку методологію розробки, а саме фреймворк Scrum.

Організація процесу розробки:

1. *Спринти*. Весь процес розбито на часові відрізки по 2 тижні. В кінці кожного спринту команда має показати робочу частину функціонала (наприклад, готовий екран реєстрації).

2. *Управління задачами*. Для контролю виконання робіт використовується електронна дошка (наприклад, Trello або Jira). Всі завдання проходять шлях: «Треба зробити» – «В роботі» – «Готово».

3. *Комунікація*. Проводяться щоденні короткі наради (англ. Daily Scrum) по 15 хвилин, щоб обговорити прогрес і проблеми, а також підсумкові зустрічі в кінці спринту для аналізу помилок.

Такий підхід забезпечує прозорість роботи та дозволяє створити MVP у найкоротші терміни.

3.2 Структурна декомпозиція робіт та організаційна структура проєкту

3.2.1 Структурна декомпозиція робіт

При гнучкій моделі структурна декомпозиція робіт (англ. Work Breakdown Structure, далі – WBS) формується шляхом ітеративної декомпозиції, де високорівневі результати розбиваються на менші, більш детальні пакети робіт, які можуть бути адаптовані та переглянуті на кожній ітерації. На відміну від традиційного підходу, де структура фіксується на початку проєкту, тут ключовим є гнучкість у внесенні змін та адаптація до нових вимог у ході роботи.

Структура WBS (рис. 3.1) складається з п'яти основних рівнів, кожен з яких відповідає за певний етап створення продукту

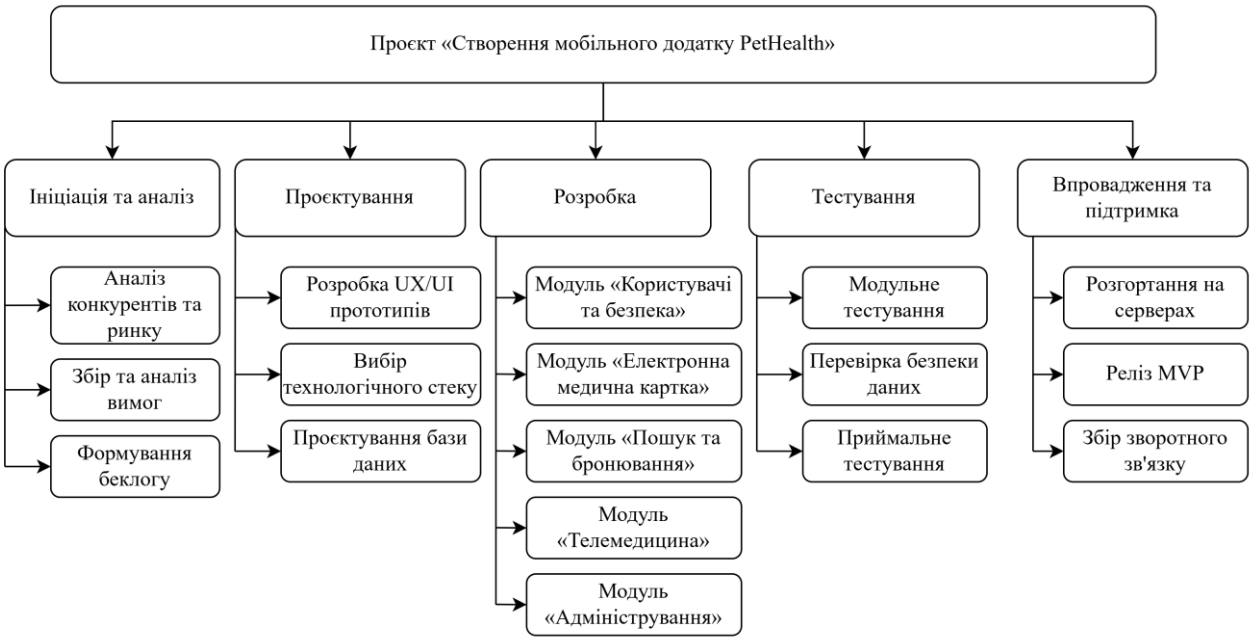

Рис. 3.1. Структурна декомпозиція робіт

Етап ініціації та аналізу включає підготовчі роботи, необхідні для старту розробки. Сюди входять аналіз ринку та конкурентів для визначення ніші, збір та формалізація вимог, а також формування початкового беклогу продукту, що є основою для подальшого планування спринтів.

Проєктування передбачає створення технічного та візуального фундаменту системи. Ключовими задачами є розробка UX/UI прототипів для забезпечення зручності користувачів, вибір технологічного стеку (React Native, Node.js) та проєктування структури бази даних.

Розробка – найбільш трудомісткий етап, декомпозований на незалежні функціональні модулі.

Тестування включає забезпечення якості продукту через модульне тестування окремих компонентів, перевірку безпеки даних та фінальне приймальне тестування перед релізом.

Фінальний етап, впровадження та підтримка, включає розгортання серверної частини, реліз MVP у магазинах додатків та збір першого зворотного зв'язку від користувачів для планування наступних ітерацій.

3.2.2 Організаційна структура проєкту

Організаційна структура проєкту (англ. Organizational Breakdown Structure, далі – OBS) є фундаментальною управлінською моделлю, що визначає архітектуру команди, ієрархію підпорядкування та зони відповідальності для успішної реалізації проєкту «PetHealth». Враховуючи високу динаміку ринку PetTech та необхідність швидкої адаптації до відгуків користувачів, для проєкту обрано крос-функціональну структуру, що базується на принципах Agile/Scrum. Такий підхід дозволяє відійти від жорсткої класичної ієрархії на користь гнучкості, самоорганізації та максимальної швидкості прийняття рішень, що є критичним для випуску конкурентного MVP.

Структура команди проєкту «PetHealth» логічно поділена на рівні, кожен з яких виконує специфічну функцію (рис. 3.2):

На стратегічному рівні, що відповідає за бізнес-цілі та ресурси, ключову роль відіграє керівник проєкту (англ. Project Manager). Він забезпечує адміністративний контроль, планування бюджету, управління ризиками та звітність перед замовником. Його доповнює власник продукту, який фокусується на змістовному наповненні додатку, формуючи беклог та пріоритизуючи User Stories для максимізації цінності продукту [39].

Тактичний рівень забезпечує ефективність робочих процесів. Тут центральною фігурою є скрам-майстер, який виступає фасилітатором команди. Його завдання полягає в організації Scrum-подій (планування, щоденні зустрічі, ретроспективи), усуненні перешкод для розробників та гарантуванні дотримання принципів Agile. Це забезпечує безперебійний ритм роботи та постійне вдосконалення процесів [40].

Безпосереднє створення продукту відбувається на виконавчому рівні, де працює крос-функціональна команда. Ядро розробки складають UI/UX дизайнер, який відповідає за інтуїтивність інтерфейсу згідно зі стандартами проєктування iOS та Android, а також Frontend та Backend розробники, що реалізують мобільний клієнт, серверну логіку та інтеграції. Критично важливою для медичного додатку є роль фахівця із забезпечення якості та спеціаліста із

кібербезпеки, які гарантують стабільність роботи, відповідність GDPR та захист персональних даних [41; 42]. Замикає цей рівень системний аналітик, який трансформує бізнес-вимоги у чіткі технічні завдання.

На консультативному рівні до проєкту залучаються вузькопрофільні експерти, які забезпечують відповідність продукту галузевим стандартам. Ветеринарний консультант валідує медичні алгоритми та коректність даних у картках тварин. Юридичний консультант відповідає за правовий супровід, розробку політики конфіденційності та регулювання відносин із партнерами.

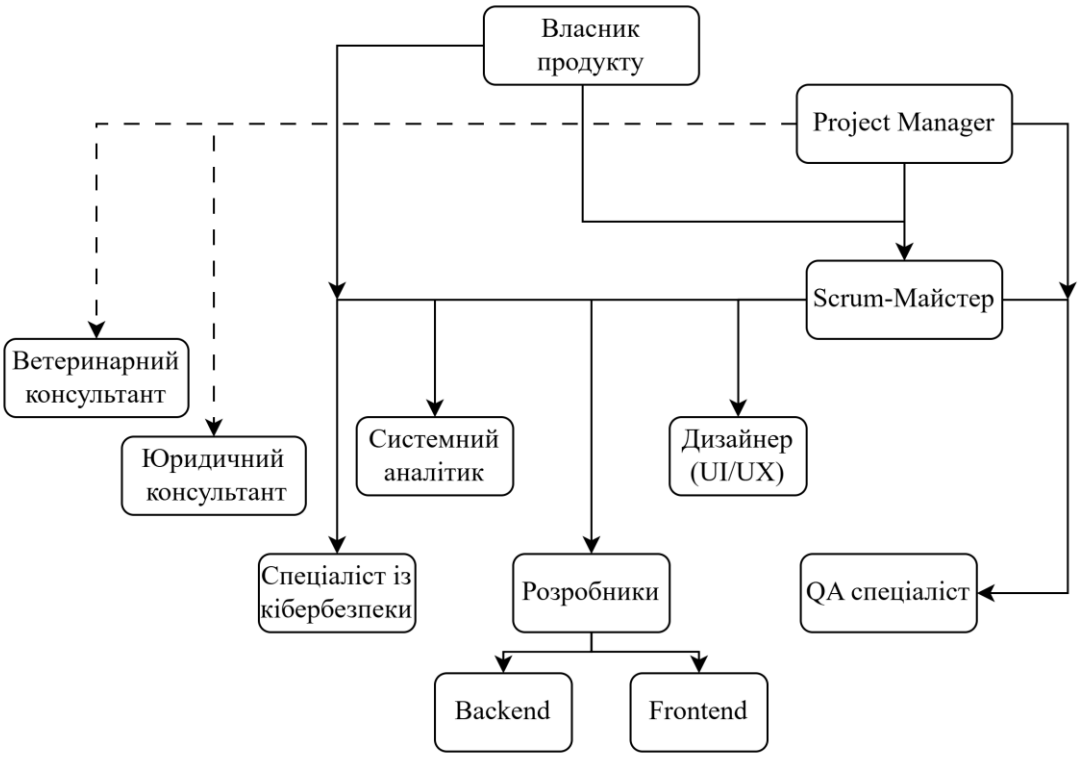

Рис. 3.2. Схема організаційної структури проєкту

Для забезпечення ефективності процесів розроблено детальну схему комунікації, яка визначає потоки інформації та зони відповідальності (рис. 3.3).

На схемі візуалізовано як прямі управлінські зв'язки, так і функціональну взаємодію між учасниками, що дозволяє уникнути комунікаційних розривів.

На стратегічному рівні ключова взаємодія відбувається між керівником проєкту та власником продукту. Вони узгоджують бюджетні обмеження та загальні терміни реалізації MVP, що дозволяє синхронізувати бізнес-очікування з ресурсними можливостями. Скрам-майстер у цій моделі виступає центральним

координаційним вузлом: він отримує стратегічні вказівки від керівництва, але взаємодіє з командою реалізації на принципах фасилітації, забезпечуючи безперешкодний робочий процес та дотримання методології Scrum без прямого адміністративного тиску.

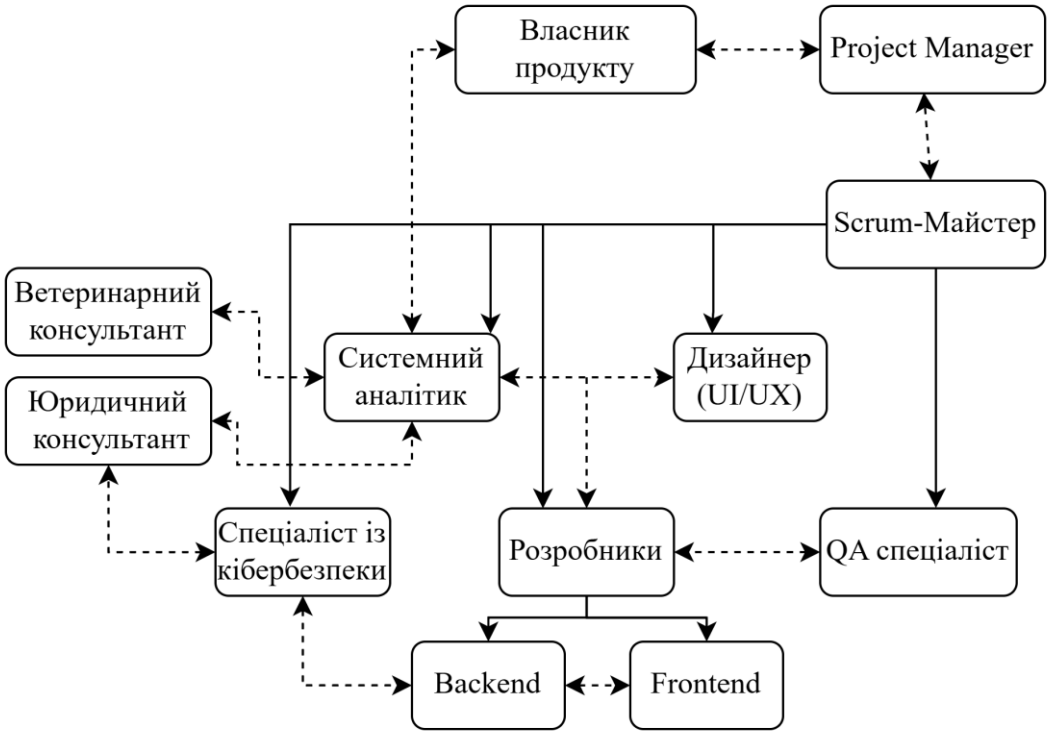

Рис. 3.3. Схема комунікаційних зв’язків в команді

Потік технічних вимог реалізується через логічний ланцюжок трансформації інформації. Власник продукту передає візію функціоналу системному аналітику, який формалізує її у вигляді технічних специфікацій та User Stories. Далі аналітик комунікує ці вимоги дизайнеру для візуалізації інтерфейсу та розробникам для написання коду. Важливо зазначити, що ці комунікаційні канали є двосторонніми, що забезпечує постійний зворотний зв'язок: технічна команда може уточнювати вимоги або пропонувати альтернативні рішення ще на етапі планування.

На виконавчому рівні забезпечено тісну співпрацю. Backend та Frontend розробники постійно координують свої дії для інтеграції серверної частини з мобільним додатком, а контроль якості реалізується через їхню циклічну взаємодію з QA-спеціалістом.

Окремий контур комунікації вибудувано для експертної підтримки: ветеринарний консультант валідує медичні дані на етапі аналітики, а юридичний консультант взаємодіє із системним аналітиком та спеціалістом із кібербезпеки. Це забезпечує відповідність продукту не лише медичним стандартам, а й правовим нормам захисту даних ще на етапі проєктування архітектури.

Така організація комунікаційних потоків дозволяє мінімізувати ризики викривлення інформації при передачі технічних завдань та забезпечує прозорість прийняття рішень.

3.3 Визначення функціональних та нефункціональних вимог до продукту ІТ проєкту

Розробка мобільного додатку «PetHealth» передбачає формування чітких вимог, які визначають очікувану поведінку системи та її ключові характеристики. Вимоги класифікуються на функціональні та нефункціональні, що дозволяє систематизувати підхід до реалізації проєкту [43].

Функціональні вимоги (англ. Functional Requirements) визначають, які саме дії має виконувати система. Вони описують основну поведінку додатку: які сервіси надаються, як користувач взаємодіє з інтерфейсом, які процеси виконує система у відповідь на запити. Наприклад, це можуть бути функції реєстрації, пошуку клінік, онлайн-запису чи ведення профілю тварини.

У таблиці 3.1 наведено перелік ключових функціональних вимог до мобільного додатку «PetHealth».

Таблиця 3.1

Функціональні вимоги до мобільного додатку «PetHealth»

ID	Тип вимоги	Опис вимоги
1	**2**	**3**
FR001	Реєстрація	Система повинна забезпечувати можливість створення облікового запису користувача шляхом введення персональних даних (ім'я, email, телефон) з обов'язковою верифікацією.

Закінчення табл. 3.1

1	2	3
FR002	Авторизація	Система має надавати механізми безпечного входу до особистого кабінету за допомогою логіна/пароля або через зовнішні сервіси (Google, Apple ID), а також функцію відновлення доступу.
FR003	Управління профілями	Система повинна дозволяти користувачу створювати, редагувати та видаляти цифрові профілі тварин, включаючи завантаження фото та збереження демографічних даних.
FR004	Пошук та фільтрація	Система має здійснювати пошук ветеринарних клінік та лікарів за заданими критеріями (геолокація, рейтинг, спеціалізація) та відображати результати списком або на карті.
FR005	Онлайн-запис	Система повинна забезпечувати функціонал вибору вільного слоту в розкладі лікаря, оформлення запису на приймання та управління статусом запису (скасування, перенесення).
FR006	Доступ до медкарти	Система має відображати історію хвороби, результати досліджень, призначення та календар вакцинацій у структурованому вигляді в профілі тварини.
FR007	Оплата послуг	Система повинна підтримувати обробку безпечних онлайн-платежів за ветеринарні послуги через інтеграцію з платіжними шлюзами.
FR008	Управління розкладом	Система повинна надавати інструменти для формування робочого графіка, налаштування тривалості приймання та блокування часових проміжків.
FR009	Ведення медзаписів	Система має забезпечувати можливість створення та збереження нових записів у медичній картці пацієнта, включаючи текстові описи діагнозів та прикріплення файлів.
FR010	Управління пацієнтами	Система повинна відображати актуальний список запланованих візитів та дозволяти змінювати статус виконання запису (наприклад, «Завершено», «Не з'явився»).
FR011	Онлайн-консультації	Система має підтримувати функціонал для проведення віддалених консультацій у форматі чату або відеозв'язку.
FR012	Модерація контенту	Система повинна надавати інтерфейс для перегляду та модерації відгуків користувачів і даних про клініки для забезпечення достовірності.
FR013	Управління користувачами	Система має забезпечувати повний контроль над обліковими записами, включаючи можливість їх блокування або видалення у разі порушення правил.

Нефункціональні вимоги (англ. Non-Functional Requirements) встановлюють загальні властивості системи, які не пов’язані з конкретною функціональністю, але впливають на її якість. До таких вимог належать продуктивність, безпека, надійність, доступність, зручність інтерфейсу, сумісність з різними платформами тощо.

У табл. 3.2 наведено перелік ключових нефункціональних вимог до мобільного додатку «PetHealth».

Таблиця 3.2

Нефункціональні вимоги до мобільного додатку «PetHealth»

ID	Тип вимоги	Опис вимоги
1	**2**	**3**
NFR001	Сумісність	Мобільний додаток повинен коректно функціонувати на пристроях під управлінням ОС Android (версії 9.0 і вище) та iOS (версії 14.0 і вище).
NFR002	Безпека	Система повинна забезпечувати шифрування персональних даних користувачів (AES-256) та передачу даних через захищений протокол HTTPS (TLS 1.3) відповідно до вимог GDPR.
NFR003	Зручність	Інтерфейс має бути спроєктований так, щоб користувач міг виконати ключову дію (запис на приймання) не більше ніж за 2 хвилини після авторизації.
NFR004	Локалізація	Додаток повинен підтримувати українську та англійську мови інтерфейсу з можливості зміни мови в налаштуваннях профілю.
NFR005	Доступність	Елементи інтерфейсу повинні відповідати стандарту WCAG 2.1 для забезпечення доступу користувачам з порушеннями зору (контрастність, підтримка скрін-рідерів).
NFR006	Надійність	Система повинна забезпечувати рівень доступності не менше 99.9% протягом місяця.
NFR007	Ефектив-ність	Додаток повинен споживати не більше 200 МБ оперативної пам’яті пристрою та займати не більше 100 МБ дискового простору після встановлення.
NFR008	Масштабо-ваність	Архітектура серверної частини повинна витримувати одночасне навантаження до 1000 активних користувачів без зниження продуктивності.

Закінчення табл. 3.2

1	2	3
NFR009	Резервне копіювання	Система повинна автоматично створювати повні резервні копії бази даних щодня.
NFR010	Відновлення після збоїв	Час повного відновлення працездатності системи після критичного збою не повинен перевищувати 60 хвилин.

Сформульовані вимоги безпосередньо впливають на вибір технологічного стеку та архітектурних рішень для проекту «PetHealth». Зокрема, високі вимоги до безпеки даних та масштабованості системи диктують необхідність використання надійних протоколів шифрування та гнучких серверних рішень. Таким чином, цей етап є перехідною ланкою від аналізу предметної області до безпосередньої технічної реалізації програмного продукту.

3.4 Створення беклогу продукту

Для реалізації функціональності мобільного додатку «PetHealth» необхідно сформувати чіткий перелік задач, які будуть виконуватись командою розробників на різних етапах проєкту. Такий перелік називається беклогом продукту (англ. Product backlog). Він є ключовим інструментом управління проєктом, який дозволяє впорядкувати задачі, визначити їхню важливість і черговість виконання, а також забезпечити реалізацію усіх очікуваних можливостей системи та технічних характеристик, необхідних для її стабільної, безпечної та зручної роботи. Беклог також сприяє ефективній комунікації між командою розробників, замовником і зацікавленими сторонами, гарантуючи, що всі вимоги будуть враховані в процесі розробки.

Беклог продукту формується на основі користувацьких історій (англ. User Stories) – коротких описів дій, які користувачі очікують виконати за допомогою системи, – та відповідних завдань і підзадач, що деталізують реалізацію кожної функції. Користувацькі історії створюються у форматі: «Як [роль користувача], я хочу [функція], щоб [мета]», що допомагає чітко визначити потреби користувачів і цілі функціоналу [44].

Перехід до User Stories є ключовим у Scrum, оскільки він зміщує фокус з абстрактної системи на кінцевий результат для користувача. Кожна історія проходить процес обговорення командою, наприклад планувального покеру (англ. Planning Poker), де визначаються критерії прийняття (англ. Acceptance Criteria) [45; 46]. Ці критерії, виходячи з функціональних та нефункціональних вимог, слугують чіткими умовами, за яких власник продукту може підтвердити завершення роботи над елементом беклогу. Це також забезпечує прозорість і зменшує ризик «розповзання» змісту.

Фрагмент переліку User Story з відповідними критеріями прийняття представлено в табл. 3.3. Повний перелік User Story представлено в табл. А.1 Додатку А кваліфікаційної роботи.

Таблиця 3.3

Фрагмент переліку User Story

Код US	**Формулювання US**	**Критерії прийняття**
1	**2**	**3**
US001 (FR001)	Як власник тварини, я хочу зареєструватися в додатку, щоб створити власний профіль та отримати доступ до послуг.	1. Користувач може ввести ім'я, email/телефон та пароль. 2. Система перевіряє валідність даних (формат email, складність пароля). 3. Користувач отримує код підтвердження (SMS/Email). 4. Після підтвердження створюється обліковий запис.
US002 (FR002)	Як зареєстрований користувач, я хочу авторизуватися в системі, щоб отримати доступ до своїх даних.	1. Вхід можливий через логін/пароль або Google/Apple ID. 2. При помилці виводиться повідомлення «Невірні дані». 3. Доступна кнопка «Забули пароль?» для відновлення доступу. 4. Після успішного входу відкривається Головний екран.

Закінчення табл. 3.3

1	2	3
US003 (FR003)	Як власник тварини, я хочу створити профіль свого улюбленця, щоб зберігати його дані в одному місці.	1. Форма містить поля: фото, кличка, вид, порода, дата народження, стать. 2. Можливість редагувати дані профілю. 3. Можливість видалити профіль тварини. 4. Профіль відображається у списку «Мої тварини».
US004 (FR004)	Як власник тварини, я хочу шукати клініки та лікарів, щоб обрати найкращий варіант поблизу.	1. Пошук за назвою, містом або геолокацією. 2. Фільтрація за рейтингом (наприклад, 4+ зірки) та спеціалізацією. 3. Відображення результатів списком та пінами на карті. 4. Картка лікаря містить фото, спеціалізацію та відгуки.
…	…	…
US011 (FR011)	Як ветеринар, я хочу проводити онлайн-консультації, щоб допомагати пацієнтам дистанційно.	1. Запуск відеодзвінка або чату з картки запису. 2. Можливість обміну текстовими повідомленнями та фото. 3. Таймер тривалості консультації.
US012 (FR012)	Як адміністратор системи, я хочу модерувати відгуки, щоб запобігти спаму та некоректній поведінці.	1. Список нових відгуків зі статусом «На перевірці». 2. Кнопки «Опублікувати» та «Видалити». 3. Можливість переглянути автора відгуку.
US013 (FR013)	Як адміністратор системи, я хочу керувати користувачами, щоб блокувати порушників правил сервісу.	1. Пошук користувача за email або ID. 2. Кнопка «Заблокувати» з вказанням причини. 3. Відображення статусу акаунта (Активний/Заблокований).

Таким чином, розроблений перелік User Story забезпечує повну фіксацію функціональних вимог до MVP. Наявність критеріїв прийняття для кожної історії гарантує прозорість процесу розробки та створює чіткі умови для підтвердження завершення робіт власником продукту.

Після формулювання користувацьких історій здійснюється їх подальша структуризація у форматі беклогу продукту. На цьому етапі для кожної User Story визначається відповідна задача, яка відображає загальний функціональний компонент, необхідний для реалізації очікуваної поведінки системи. Кожна

задача, у свою чергу, деталізується через підзадачі, що конкретизують послідовність технічних або організаційних дій, необхідних для її виконання.

Така побудова беклогу дозволяє перейти від узагальнених вимог користувача до чітко визначених робочих одиниць, що підлягають реалізації в межах проєкту. Структура таблиці забезпечує логічну ієрархію – від потреб користувача до задач і підзадач – і створює основу для подальшого планування, оцінювання та контролю ходу виконання робіт.

Структурований фрагмент беклогу продукту проєкту «PetHealth» наведено в табл. 3.4. Повний беклог продукту проєкту продемонстровано в табл. Б.1 Додатку Б кваліфікаційної роботи.

Таблиця 3.4

Фрагмент беклогу продукту проєкту «PetHealth»

Код US	Задача	Підзадача
US001	T01.1 Реалізація бекенду реєстрації	ST01.1.1 Проєктування схеми бази даних для користувачів.
		ST01.1.2 Створення API для реєстрації.
		ST01.1.3 Реалізація валідації вхідних даних та хешування паролів.
	T01.2 Реалізація фронтенду реєстрації	ST01.2.1 Верстка екрану реєстрації (поля вводу, кнопки).
		ST01.2.2 Інтеграція з API реєстрації та обробка помилок.
		ST01.2.3 Реалізація екрану підтвердження коду.
US002	T02.1 Механізм авторизації	ST02.1.1 Створення API для логіну та генерації токенів.
		ST02.1.2 Реалізація входу через Google/Apple ID.
		ST02.1.3 Верстка екрану входу та логіка збереження сесії.
	T02.2 Відновлення доступу	ST02.2.1 Реалізація API для скидання пароля (надсилання посилання).
		ST02.2.2 Створення екранів «Забули пароль» та «Новий пароль».
...	...	…
US010	T10.1 Панель адміністратора клініки	ST10.1.1 Верстка дашборду зі списком записів на день.
		ST10.1.2 Логіка фільтрації записів за лікарями та статусами.
		ST10.1.3 Функціонал зміни статусу візиту («Завершено», «Не з'явився»).
US011	T11.1 Телемедицина	ST11.1.1 Налаштування WebSocket для чату в реальному часі.
		ST11.1.2 Інтеграція сервісу для відеодзвінків.
		ST11.1.3 Інтерфейс чату з можливістю надсилання фото.
US012	T12.1 Модерація	ST12.1.1 Адмін-панель для перегляду нових відгуків.
		ST12.1.2 API для затвердження або видалення відгуків.
		ST12.1.3 Логіка розрахунку рейтингу клініки на основі відгуків.

Побудова структурованого беклогу продукту на основі користувацьких історій, задач і підзадач забезпечує чітке бачення проєкту як для команди розробників, так і для зацікавлених сторін. Такий підхід дозволяє ефективно розподіляти завдання, контролювати виконання, вчасно виявляти проблеми та адаптувати функціональність відповідно до потреб користувачів.

3.5 Планування спринтів

Планування спринту є ключовою подією в методології Scrum, яка визначає обсяг робіт на найближчу ітерацію. Процес планування включає декомпозицію задач із беклогу продукту на конкретні технічні роботи, їх оцінку в годинах та розподіл між виконавцями. Це дозволяє точно розрахувати навантаження команди та уникнути ризиків невиконання зобов'язань [47].

Нижче наведено фрагмент планування спринту №1 (табл. 3.5). Мета спринту №1: реалізувати захищений доступ користувачів до системи (реєстрація, вхід, відновлення пароля) та підготувати базову архітектуру бази даних. До спринту увійшли задачі, що відповідають користувацьким історіям US001 (Реєстрація) та US002 (Авторизація).

Повний план виконання робіт спринту №1 продемонстровано в табл. В.1 Додатку В кваліфікаційної роботи.

Таблиця 3.5

Фрагмент плану виконання робіт спринту №1

Задача/ Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
1	**2**	**3**	**4**	**5**
ST01.1.1	Виконання SQL-скрипта створення таблиці Users у БД	3	Backend Developer	PostgreSQL
ST01.1.2	Написання коду API маршруту/register	2	Backend Developer	Node.js, Express
ST01.1.2	Програмування контролера створення користувача	4	Backend Developer	JavaScript

Закінчення табл. 3.5

1	2	3	4	5
ST01.1.3	Кодування схеми валідації даних (email/pass)	3	Backend Developer	Joi Library
ST01.1.3	Підключення бібліотеки хешування паролів	2	Backend Developer	Bcrypt
ST02.1.1	Програмна реалізація генерації JWT токена	4	Backend Developer	JSON Web Token
ST02.1.1	Написання Middleware для перевірки токена	3	Backend Developer	Node.js
ST02.1.2	Налаштування стратегії Google OAuth у коді	5	Backend Developer	Passport.js
ST02.2.1	Конфігурація поштового клієнта (SMTP)	3	Backend Developer	Nodemailer
ST02.2.1	Кодування ендпоінту для скидання пароля	3	Backend Developer	Node.js
ST01.2.1	Верстка екрану реєстрації за готовим макетом	4	Frontend Developer	React Native
ST01.2.1	Стилізація компонентів вводу (Input, Button)	4	Frontend Developer	CSS/ Stylesheet
…	…	…	…	…
РАЗОМ	Загальний обсяг робіт	71	Команда	

Як видно з табл. 3.5, загальна трудомісткість робіт спринту №1 становить 71 годину. Левова частка часу відведена на розробку серверної логіки та клієнтського інтерфейсу, що виконуються паралельно для оптимізації процесу. Задачі з тестування заплановані на фінальну частину ітерації, що дозволяє перевірити вже готовий функціонал реєстрації та авторизації перед його релізом. Використання чіткої декомпозиції дозволяє контролювати прогрес виконання кожної технічної підзадачі.

Для візуалізації реального часового розподілу завдань було побудовано діаграму Ганта [48] за допомогою програмного забезпечення ProjectLibre. Діаграма демонструє, що роботи над серверною та клієнтською частинами розпочинаються одночасно (паралельно) в перший день спринту, а інтеграція та тестування відбуваються послідовно після готовності відповідних компонентів (рис. 3.4).

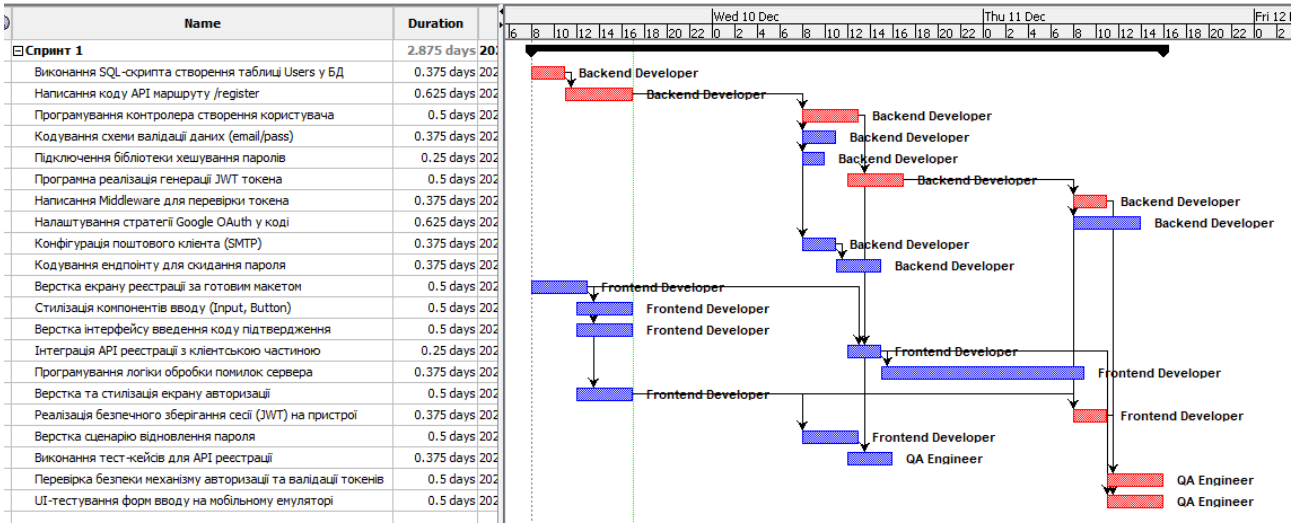

Рис. 3.4. Візуалізація спринту 1 за допомогою діаграми Ганта

Окрім цього, діаграма візуалізує ключові точки синхронізації між учасниками команди. На графіку чітко визначено моменти передачі результатів роботи, коли завершений Backend-компонентом функціонал стає доступним для інтеграції Frontend-розробнику. Це дозволяє уникнути комунікаційних розривів та гарантує, що розробники клієнтської частини не розпочнуть роботу над модулем раніше, ніж буде готова відповідна серверна логіка.

3.6 Формування Use Case елементів до функціональних вимог та побудова Use Case Diagram

Для детального опису функціональних можливостей мобільного додатку «PetHealth» здійснюється побудова сценаріїв використання (англ. Use Case), що відображають типові взаємодії користувачів із системою. Такі сценарії забезпечують структуроване бачення того, як різні типи акторів ініціюють функції додатку для досягнення конкретних цілей.

Елементи Use Case сформовано на основі функціональних вимог (табл. 3.1) та користувацьких історій (див. Додаток А). Для забезпечення повноти моделі в системі виділено чотири ключових акторів, кожен з яких має специфічні права доступу та цілі:

1. Власник тварини – основний користувач додатку, який здійснює пошук клінік, онлайн-запис на приймання, управління профілями тварин, перегляд медичних карток, оплату послуг та отримання онлайн-консультацій.

2. Ветеринар – медичний спеціаліст, який використовує систему для налаштування власного робочого розкладу, ведення медичних записів пацієнтів та надання дистанційних консультацій.

3. Адміністратор клініки – співробітник закладу, відповідальний за операційну діяльність, зокрема моніторинг записів та управління потоком пацієнтів.

4. Адміністратор системи – технічний спеціаліст, який забезпечує модерацію контенту (відгуків) та управління обліковими записами користувачів для підтримання безпеки платформи.

Така рольова модель дозволяє чітко розмежувати зони відповідальності та реалізувати необхідні механізми контролю доступу.

Детальний перелік Use Case елементів наведено в табл. 3.6.

Аналіз наведених сценаріїв свідчить про те, що архітектура додатку є клієнтоорієнтованою: найбільша кількість функціональних можливостей (UC001–UC007) зосереджена навколо актора «Власник тварини». Це вимагає від системи створення максимально зручного та інтуїтивного інтерфейсу для цієї ролі.

Таблиця 3.6

Перелік Use Case елементів для мобільного додатку «PetHealth»

ID Use Case	Назва	Пов'язана вимога	Актор	Опис
1	**2**	**3**	**4**	**5**
UC001	Реєстрація нового користувача	FR001 (US001)	Користувач	Користувач створює новий обліковий запис, вводячи персональні дані та проходячи верифікацію через SMS/Email.
UC002	Авторизація в системі	FR002 (US002)	Зареєстро-ваний користувач	Користувач входить у свій особистий кабінет за допомогою логіна/пароля або через зовнішні сервіси (Google, Apple ID).

Закінчення табл. 3.6

1	2	3	4	5
UC003	Управління профілем тварини	FR003 (US003)	Власник тварини	Власник створює, редагує або видаляє цифрову картку свого улюбленця, додає фото та дані.
UC004	Пошук клінік та лікарів	FR004 (US004)	Власник тварини	Власник здійснює пошук ветеринарних послуг за геолокацією, рейтингом або спеціалізацією та переглядає результати на мапі/списком.
UC005	Онлайн-запис на приймання	FR005 (US005)	Власник тварини	Власник обирає вільний слот у розкладі лікаря та оформляє запис на візит.
UC006	Перегляд медичної картки	FR006 (US006)	Власник тварини	Власник переглядає історію хвороби, призначення, результати аналізів та календар вакцинацій свого улюбленця.
UC007	Оплата послуг	FR007 (US007)	Власник тварини	Власник здійснює безготівкову оплату за надані ветеринарні послуги через інтегрований платіжний шлюз.
UC008	Управління робочим розкладом	FR008 (US008)	Ветеринар	Лікар налаштовує свої робочі години, тривалість візиту та блокує час для перерв/відпусток.
UC009	Ведення медичних записів	FR009 (US009)	Ветеринар	Лікар створює новий запис у картці пацієнта, вносить діагноз, скарги та прикріплює файли/документи.
UC010	Моніторинг записів клініки	FR010 (US010)	Адміністратор клініки	Адміністратор переглядає список візитів на день, фільтрує їх та змінює статуси («Завершено», «Не з'явився»).
UC011	Проведення онлайн-консультації	FR011 (US011)	Ветеринар, Власник тварини	Лікар та власник тварини зв'язуються дистанційно через відеозв'язок або чат для надання консультації.
UC012	Модерація відгуків	FR012 (US012)	Адміністратор системи	Адміністратор перевіряє нові відгуки, публікує їх або видаляє у разі порушення правил (спам, некоректна поведінка).
UC013	Управління користувачами	FR013 (US013)	Адміністратор системи	Адміністратор здійснює пошук користувачів та блокує порушників правил сервісу.

Водночас сценарії, закріплені за ветеринарами та адміністраторами (UC008–UC013), відіграють роль підтримуючих процесів, без яких неможливе коректне функціонування основних сервісів. Зокрема, сценарії управління розкладом та модерації контенту забезпечують актуальність та валідність даних у системі.

На основі сформованих Use Case описів побудовано діаграму варіантів використання (англ. Use Case Diagram), яка візуалізує основні сценарії роботи системи та їх зв’язки [49]. Це значно полегшує розробку, тестування, впровадження й узгодження вимог між усіма учасниками проєкту – розробниками, замовником та іншими зацікавленими сторонами. На рис. 3.5 зображено Use Case Diagram для мобільного додатку «PetHealth», яка охоплює всі функціональні вимоги, передбачені проєктом.

Аналіз побудованої діаграми дозволяє виділити ключові архітектурні особливості системи. По-перше, центральним елементом взаємодії є прецедент «Авторизація в системі». Він пов'язаний відношенням включення з більшістю функціональних блоків (управління профілем, пошук, перегляд медкарти, моніторинг записів). Це графічно підкреслює вимогу щодо захисту персональних даних: доступ до функцій можливий виключно після успішної ідентифікації користувача.

По-друге, діаграма візуалізує умовні розширення базових процесів. Зокрема, сценарій «Оплата послуг» розширює сценарій «Онлайн-запис на приймання», що означає ініціацію процесу оплати лише за певних умов (наприклад, якщо обрана послуга є платною або вимагає передплати), не будучи обов’язковим для всіх типів записів. Крім того, сценарій «Проведення онлайн-консультації» також розширює функціонал оплати, вказуючи на те, що доступ до телемедичних послуг надається виключно після успішної фінансової транзакції.

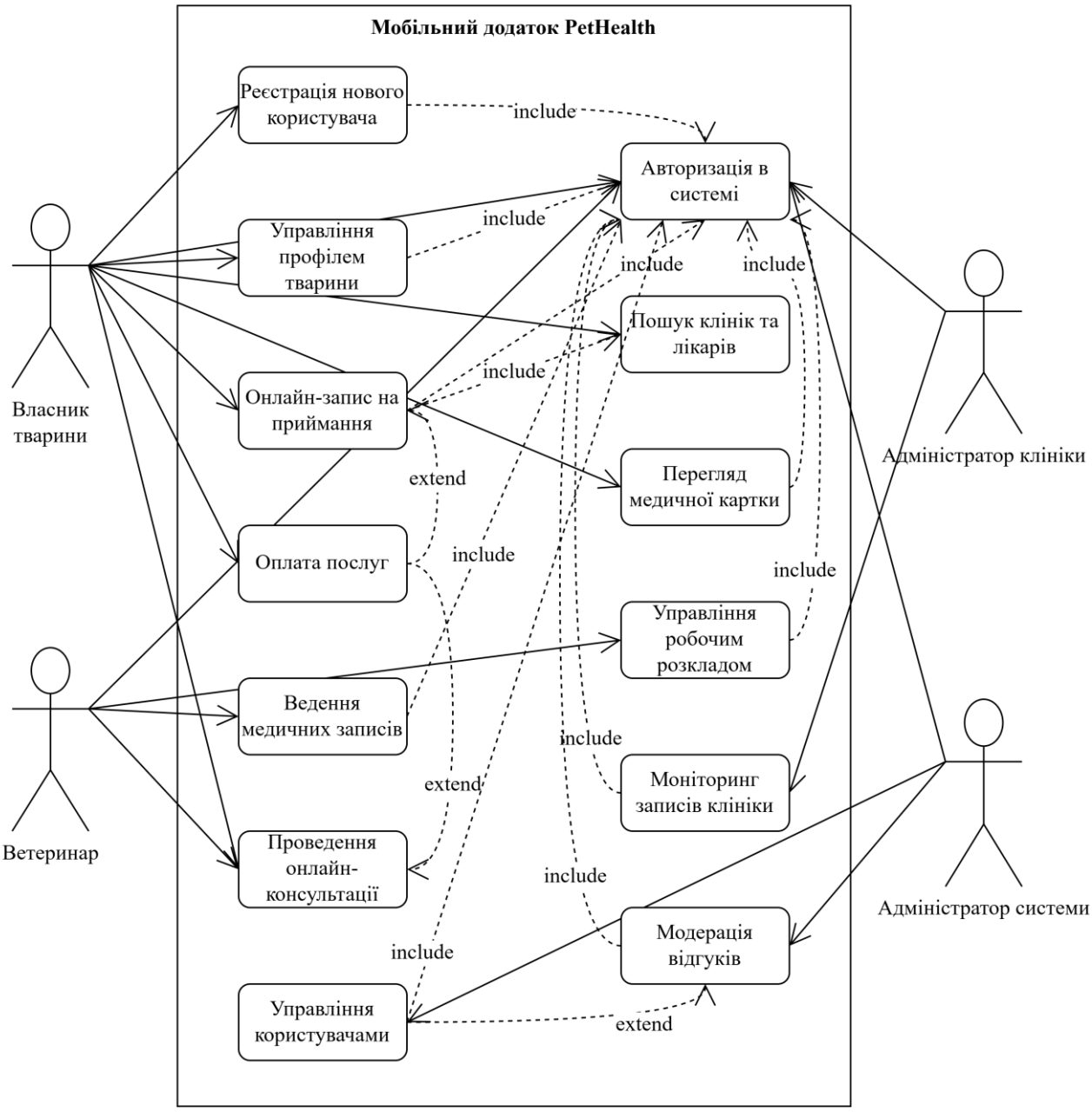

Рис. 3.5. Use Case Diagram мобільного додатку «PetHealth»

Таким чином, побудова Use Case моделі дозволила трансформувати функціональні вимоги у чіткі сценарії взаємодії користувачів із системою «PetHealth». Розроблена діаграма фіксує межі проекту, чітко розподіляє зони відповідальності між акторами та визначає критичні залежності між процесами.

3.7 Управління ризиками

Успішна реалізація ІТ-проєкту, зокрема створення мобільного додатку «PetHealth», неможлива без системного підходу до управління ризиками.

Специфіка ветеринарної сфери, використання хмарних технологій AWS та кросплатформної розробки на React Native створюють унікальний набір загроз, які можуть вплинути на терміни, бюджет та якість кінцевого продукту.

Процес управління ризиками в даній роботі базується на підході, описаному в роботі [50], що включає три етапи: ідентифікацію, кількісну оцінку та розробку стратегій реагування.

На першому етапі було проведено аналіз предметної області, в результаті чого ідентифіковано 30 потенційних ризикових подій. Для зручності аналізу вони класифіковані за сферами впливу:

1. Технічні: пов’язані з обраним стеком технологій (React Native, AWS) та інтеграціями.

2. Управлінські: стосуються процесів планування та роботи команди.

3. Операційні та юридичні: загрози стабільності роботи сервісу та правові аспекти (GDPR).

4. Фінансові: бюджетні обмеження та валютні коливання.

5. Взаємодія: ризики, пов'язані з користувачами та партнерами (ветклініками).

6. Форс-мажорні: зовнішні обставини непереборної сили, зокрема воєнний стан в Україні.

Фрагмент ідентифікованих ризиків наведено в табл. 3.7. Повний реєстр ідентифікованих ризиків продемонстровано в табл. Г.1 Додатку Г кваліфікаційної роботи.

Таблиця 3.7

Фрагмент ідентифікації ризиків проекту

№	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	**2**	**3**	**4**	**5**
1	Технічні	Несумісність із застарілим ПЗ ветклінік	Висока	Середня
2	Технічні	Критичні помилки (баги) в коді мобільного додатку	Висока	Висока

Закінчення табл. 3.7

1	2	3	4	5
3	Технічні	Проблеми інтеграції з Google Maps API	Середня	Середня
4	Технічні	Низька продуктивність додатку на старих смартфонах	Середня	Висока
…	…	…	…	…
12	Управлінські	Відсутність технічної документації	Середня	Висока
13	Операційні	Відмова в публікації App Store / Google Play	Висока	Низька
14	Операційні	Збій хмарного хостингу (AWS)	Висока	Низька
…	…	…	…	…
16	Юридичні	Порушення законодавства про захист даних (GDPR)	Висока	Висока
17	Юридичні	Відсутність ліцензій на сторонні бібліотеки	Середня	Висока
…	…	…	…	…
26	Взаємодія (Користувачі)	Негативні відгуки через баги MVP	Висока	Висока
…	…	…	…	…
29	Форс-мажорні	Безпекові ризики (війна)	Висока	Низька
30	Форс-мажорні	Різке коливання валютного курсу	Середня	Низька

Аналіз реєстру ідентифікованих ризиків (табл. 3.7) демонструє, що проєкт реалізується в умовах високої невизначеності. Найбільшу питому вагу серед загроз складають технічні ризики, зумовлені складністю інтеграції мобільного додатку з різнорідними обліковими системами ветеринарних клінік та використанням хмарної архітектури.

Водночас є наявність групи форс-мажорних ризиків високого ступеня впливу, пов’язаних із воєнним станом в Україні (блекаути, безпека команди). Хоча ці ризики є зовнішніми і важкокерованими, їх ідентифікація на ранньому етапі є критично важливою для адаптації процесів розробки та комунікації в команді.

Для кількісного оцінювання ризиків проєкту «PetHealth» обрано такі параметри: затримки у часі, фінансові втрати, ймовірність настання ризикової

події та частота її прояву у межах проєкту. Кожен параметр оцінювався двома способами: якісно та кількісно.

Затримки у часі характеризують можливий вплив ризику на тривалість проєкту. Якісна оцінка визначає рівень затримки: «низька», «середня», «висока». Кількісна оцінка відповідає числовому значенню за квазікількісною шкалою, що дозволяє враховувати її у комплексному показнику важливості ризику.

Фінансові втрати відображають очікувані додаткові витрати при реалізації ризику. Якісна оцінка визначає, наскільки значними будуть ці витрати, а кількісна перетворює оцінку у числовий показник для подальших розрахунків.

Ймовірність показує, наскільки висока ймовірність настання події. Якісна оцінка визначає рівень ймовірності, а кількісна дозволяє врахувати її при розрахунку важливості ризику.

Частота характеризує, скільки разів ризик може реалізуватися протягом проєкту. Аналогічно, якісна оцінка описує рівень частоти, а кількісна перетворює її на числовий показник.

Важливість ризику (комплексний показник) розраховується на основі кількісних оцінок усіх параметрів, що дозволяє ранжувати ризики та виділити три класи: найбільш впливові, середнього впливу та найменш впливові. Такий підхід забезпечує системне управління ризиками та підтримує прийняття обґрунтованих управлінських рішень у процесі реалізації проєкту.

Фрагмент результатів оцінювання представлено в табл. 3.8. Повний перелік результатів оцінювання ризиків проєкту продемонстровано в табл. Д.1 Додатку Д кваліфікаційної роботи.

Таблиця 3.8

Фрагмент результатів оцінювання ризиків проєкту

№	Ризикова подія	Затримки у часі		Фінансові витрати		Ймовір-ність		Частота		Важливість ризику
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
1	Несумісність із застарілим ПЗ ветклінік	ВС	8	ВН	7	ВС	8	ВС	8	56
2	Критичні помилки (баги) в коді мобільного додатку	СС	5	СС	5	ВН	7	ВС	8	35
...	...	...	...	...	...	...	...	...	...	...
7	Конфлікт синхронізації даних	СВ	6	ВВ	9	ВН	7	СВ	6	63
...	...	...	...	...	...	...	...	...	...	...
14	Збій хмарного хостингу (AWS)	ВВ	9	ВВ	9	НС	2	НН	1	18
15	Збій платіжного шлюзу під час оплати послуг	НВ	3	ВВ	9	НВ	3	СН	4	27
...	...	...	...	...	...	...	...	...	...	...
18	Претензії від користувачів через некоректні поради	НС	2	ВС	8	СН	4	СС	5	32
19	Перевищення бюджету на хмарні сервіси	СН	4	ВВ	9	СВ	6	НН	1	54
...	...	...	...	...	...	...	...	...	...	...
23	Відмова клінік від партнерства	СВ	6	ВВ	9	СС	5	НВ	3	45
...	...	...	...	...	...	...	...	...	...	...
27	Відключення електроенергії (блекаути)	ВС	8	СВ	6	ВВ	9	ВС	8	54
...	...	...	...	...	...	...	...	...	...	...

Закінчення табл. 3.8

1	2	3	4	5	6	7	8	9	10	11
...	...	...	...	...	...	...	...	...	...	...
29	Безпекові ризики (війна)	ВВ	9	ВВ	9	ВВ	9	ВВ	9	81
30	Різке коливання валютного курсу	НС	2	ВН	7	ВН	7	СС	5	49

На основі розрахунку важливості виділено 5 найбільш критичних ризиків, які потребують першочергової уваги:

Безпекові ризики (війна) – 81 бал. Найвищий пріоритет через постійну загрозу життю команди та стабільності інфраструктури.

Конфлікт синхронізації даних – 63 бали. Специфічний технічний ризик, критичний для цілісності медичних карток.

Несумісність із застарілим ПЗ ветклінік – 56 балів. Основний бар'єр для інтеграції з партнерами.

Перевищення бюджету та Блекаути – по 54 бали. Загрози, що можуть зупинити розробку через брак ресурсів.

Для мінімізації впливу виявлених загроз розроблено протиризикові заходи проєкту, фрагмент яких показано в табл. 3.9. Для кожного ризику визначено:

Профілактичні заходи – це дії, що виконуються заздалегідь для зниження ймовірності.

Симптоми (тригери) – це ознаки того, що ризик починає реалізовуватися.

Реактивні заходи – це план дій при появі симптомів та при настанні самої проблеми.

Повний перелік розробки протиризикових заходів проєкту продемонстровано в табл. Е.1 Додатку Е кваліфікаційної роботи.

Таблиця 3.9

Фрагмент розробки протиризикових заходів проєкту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		Профілактика		При симптомі	При проблемі
1	Несумісність із застарілим ПЗ ветклінік	Розробка веб-кабінету адміністратора, що не потребує інтеграції.	Клініка повідомляє про використання паперових журналів або Excel.	Навчання персоналу роботі через браузер/планшет.	Надання планшета з ПЗ в оренду для ручного введення.
…	…	…	…	…	…
7	Конфлікт синхронізації даних	Налаштування автозбереження даних.	Поява різних даних на різних пристроях.	Повідомлення користувачу про помилку.	Завантаження актуальних даних із сервера.
…	…	…	…	…	…
19	Перевищення бюджету на хмарні сервіси	Моніторинг витрат (AWS Budgets).	Витрачено 80% бюджету, а зроблено 50% роботи.	Скорочення функціоналу MVP.	Пошук додаткового фінансування або заморозка проєкту.
…	…	…	…	…	…
27	Відключення електроенергії (блекаути)	Закупівля EcoFlow/Starlink, локальне середовище.	Графіки відключень, втрата зв'язку.	Перехід на асинхронний графік, офлайн-робота.	Зсув дедлайнів, передача задач колегам зі світлом.
…	…	…	…	…	…
29	Безпекові ризики (війна)	Дистанційна робота, сервери за кордоном.	Оголошення повітряної тривоги.	Перехід в укриття, зупинка нарад.	Призупинення проєкту, евакуація команди.
30	Різке коливання валютного курсу	Фіксація курсу в договорах, валютні резерви.	Різка зміна курсу НБУ.	Перегляд бюджету, скорочення витрат.	Підняття цін на послуги сервісу.

Розроблена матриця реагування на ризики відображає перехід від реактивного до проактивного управління проєктом. Для кожного критичного ризику визначено не лише план дій на випадок кризи, а й набір превентивних

заходів, які дозволяють знизити ймовірність настання загрози або зменшити її наслідки.

3.8 Бюджет проєкту

Економічне планування проєкту здійснюється на основі моделі Time&Material (T&M) [51]. Цей підхід дозволяє гнучко керувати витратами, сплачуючи за фактично відпрацьований час фахівців, що є оптимальним для етапу створення MVP в умовах можливих змін вимог. Бюджет розраховано для річного життєвого циклу (12 місяців), який включає етапи ініціації, активної розробки MVP, релізу та технічної підтримки.

Першочерговим завданням є визначення складу команди та ступеня завантаженості кожного учасника. Для оптимізації витрат застосовано гібридну модель залучення (табл. 3.10), де технічне ядро має змінне навантаження, адміністративний персонал – часткову зайнятість, а експерти – проєктну.

Таблиця 3.10

План залучення та навантаження людських ресурсів

Роль	**Тип зайнятості**	**Наванта-ження**	**Тривалість участі**	**Примітка**
1	**2**	**3**	**4**	**5**
Backend Розробник	Змінна	100% – 30%	12 міс.	6 міс. – розробка API/БД, 6 міс. – підтримка.
Frontend Developer	Змінна	100% – 30%	12 міс.	6 міс. – мобільна розробка, 6 міс. – оновлення.
QA Спеціаліст	Часткова	50%	12 міс.	Постійний контроль якості, регресійне тестування.
Керівник проєкту (PM)	Часткова	50%	12 міс.	Координація, звітність, управління ризиками.

Закінчення табл. 3.10

1	2	3	4	5
Власник продукту (PO)	Часткова	20%	12 міс.	Пріоритезація беклогу, приймання робіт.
Скрам-майстер (SM)	Часткова	20%	12 міс.	Фасилітація подій, усунення перешкод.
Системний аналітик	Проєктна	100%	2 міс.	Етап збору вимог та формування ТЗ.
UI/UX Дизайнер	Проєктна	100%	1.5 міс.	Проєктування інтерфейсів.
Спеціаліст із кібербезпеки	Проєктна	-	1 міс.	Аудит безпеки перед релізом.
Ветеринарний консультант	Проєктна	-	2 тижні	Валідація медичних довідників.
Юридичний консультант	Проєктна	-	2 тижні	Розробка оферти та політики конфіденційності.

Запропонований план розподілу трудових ресурсів є основою для подальшого фінансового планування. Чітке розмежування типів зайнятості (постійна, часткова, проєктна) та фіксація тривалості участі кожного фахівця дозволяють перейти до точного розрахунку основної статті витрат бюджету – фонду оплати праці, який наведено в наступній таблиці.

Витрати на команду є основною статтею бюджету. Розрахунок базується на середніх ринкових ставках IT-спеціалістів в Україні [52] (табл. 3.11).

Таблиця 3.11

Розрахунок фонду оплати праці

Роль у команді	Середня ЗП (грн/міс)	Формула розрахунку	Всього (грн)
1	2	3	4
Backend Розробник	70 000	(70000 × 6) + (21000 × 6)	546 000
Frontend Developer	42 500	(42500 × 6) + (12750 × 6)	331 500
QA Спеціаліст	34 000	34000 × 0.5 × 12	204 000
Керівник проєкту (PM)	42 500	42500 × 0.5 × 12	255 000

Закінчення табл. 3.11

1	2	3	4
Власник продукту (PO)	45 000	45000 × 0.2 × 12	108 000
Скрам-майстер (SM)	42 500	42500 × 0.2 × 12	102 000
Системний аналітик	40 000	40000 × 2	80 000
UI/UX Дизайнер	38 000	38000 × 1.5	57 000
Спеціаліст із кібербезпеки	22 500	22500 × 1	22 500
Ветеринарний консультант	26 000	26000 × 0.5$ (2 тижні)	13 000
Юридичний консультант	20 000	20000 × 0.5$ (2 тижні)	10 000
РАЗОМ	1 729 000		

Аналіз структури фонду оплати праці показує, що домінуючою статтею є витрати на безпосередню технічну реалізацію продукту – оплата послуг Backend та Frontend розробників. Це є закономірним для IT-проєктів із складною архітектурою, де основна цінність створюється саме на етапі кодування. Водночас витрати на консультантів та вузькопрофільних спеціалістів складають незначну частку бюджету завдяки їх короткостроковому залученню, що підтверджує економічну доцільність проєктної форми зайнятості.

Окрім витрат на персонал, бюджет включає операційні витрати на технічну інфраструктуру (оренда серверів, доменів), ліцензії на програмне забезпечення для командної роботи, а також маркетинговий бюджет для просування продукту на ринку [53] (табл. 3.12).

Представлені розрахунки підтверджують економічну доцільність обраної стратегії мінімізації капітальних вкладень. Замість створення власної серверної інфраструктури та розширення адміністративного штату, проєкт спирається на використання хмарних сервісів AWS та аутсорсинг супровідних послуг.

Таблиця 3.12

Операційні та маркетингові витрати

Витрата	Опис	Вартість (грн/рік)
Хмарна інфраструктура (AWS)	Оренда серверів (EC2), бази даних (RDS), сховища (S3) та сервісів сповіщень (SNS)	150 000
Публікація в магазинах додатків	Річна підписка Apple Developer Program ($99) та одноразовий внесок Google Play Console ($25).	5 500
Ліцензії ПЗ	Оплата інструментів для дизайну (Figma), управління проєктами (Jira/Trello) та IDE.	45 000
Домен та SSL	Оренда доменного імені pethealth.ua та сертифікатів безпеки.	2 500
Маркетинг та просування	Таргетована реклама в соцмережах (Instagram, Facebook), друк промо-матеріалів для клінік.	250 000
Бухгалтерський супровід	Аутсорсинг ведення бухгалтерського обліку, подання звітності та розрахунок податків.	50 000
РАЗОМ	Загальні операційні витрати	503 000

Для забезпечення фінансової стабільності проєкту та мінімізації впливу ідентифікованих ризиків, до зведеного бюджету закладено резервний фонд у розмірі 10% від суми основних витрат [54] (табл. 3.13).

Таблиця 3.13

Загальний бюджет проєкту «PetHealth»

Стаття витрат	Сума (грн)	Частка у бюджеті, %
Фонд оплати праці	1 729 000	70,4%
Операційні та маркетингові витрати	503 000	20,5%
Всього основні витрати	2 232 000	90,9%
Резервний фонд (10% від основних витрат)	223 200	9,1%
ЗАГАЛЬНИЙ БЮДЖЕТ ПРОЄКТУ	2 455 200	100%

Зведена структура бюджету чітко відображає пріоритети проєкту. Левова частка фінансування (понад 70%) спрямована на людський капітал, що є характерною ознакою інтелектуальних IT-продуктів. Операційні витрати, хоч і

становлять меншу частину (20,5%), забезпечують критично важливий фундамент для маркетингового старту та технічної стійкості. Закладений резервний фонд у розмірі 223 200 грн (9,1%) є фінансовим буфером, який гарантує, що проєкт не зупиниться у випадку непередбачуваних обставин, таких як коливання курсу валют або необхідність додаткових годин розробки.

Отже, розроблений кошторис проєкту «PetHealth» загальною вартістю 2 455 200 грн повністю покриває потреби всіх етапів життєвого циклу продукту протягом першого року. Бюджет враховує ринкові реалії оплати праці, необхідні витрати на просування та технічну підтримку, а також містить механізми захисту від ризиків. Таке фінансове планування дозволяє команді зосередитися на технічній реалізації продукту, маючи гарантії ресурсного забезпечення.

РОЗДІЛ 4. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ РЕАЛІЗАЦІЇ ПРОЄКТУ

4.1 Опис структури та алгоритмів програмного забезпечення

Програмне забезпечення мобільного додатку «PetHealth» реалізується з використанням сучасних технологій, що забезпечують високу ефективність, масштабованість, безпеку та зручність користування. Архітектура системи є клієнт-серверною і включає мобільний клієнт (додаток), серверну частину (бекенд), реляційну базу даних і хмарну інфраструктуру для зберігання та обробки інформації.

Мобільний додаток створюється з використанням React Native – кросплатформного фреймворку, що дозволяє одночасно підтримувати Android та iOS. Це рішення забезпечує єдиний кодовий простір, адаптацію інтерфейсу до різних типів пристроїв, а також спрощує інтеграцію з нативними модулями за допомогою Native Modules. Такий підхід дозволяє зменшити витрати часу на розробку і підтримку, зберігаючи при цьому повну функціональність нативних додатків [55].

Серверна частина системи реалізована за допомогою середовища Node.js з використанням фреймворку Express. Вона відповідає за обробку запитів користувачів, виконання бізнес-логіки, управління авторизацією та взаємодію з базою даних. API, побудовані за принципом REST, дозволяють клієнту ефективно взаємодіяти із сервером, використовуючи стандартизовані HTTP-запити [56].

Для зберігання даних обрано систему управління базами даних PostgreSQL, що дозволяє ефективно організовувати та пов’язувати реляційні об'єкти, такі як профілі користувачів, медичні картки тварин, історію бронювань, транзакції, а також повідомлення. Реляційна модель дозволяє підтримувати цілісність даних і зручно масштабувати структуру за потреби.

Інфраструктурна частина рішення реалізується з використанням Amazon Web Services. Хмарна платформа забезпечує надійне зберігання інформації,

масштабованість під час зростання навантаження, резервне копіювання критичних даних і доступ до сервісів реального часу. Зокрема, використовується сервіс Amazon SNS для реалізації push-сповіщень, що інформують користувача про майбутні події, оновлення в медичній картці чи новини клініки [57].

Інтеграція з зовнішніми сервісами також є важливою частиною архітектури. Для відображення клінік на карті та побудови маршрутів використовується Google Maps API [58]. Для обробки онлайн-платежів інтегруються платіжні системи Portmone, EasyPay і PayPal. Крім того, система підтримує двосторонню комунікацію між користувачами та клініками за допомогою технології WebSocket, яка забезпечує передачу повідомлень у реальному часі.

Таким чином, запропонована структура програмного забезпечення дозволяє ефективно реалізувати функціональні вимоги проєкту, забезпечуючи при цьому стабільність, масштабованість і високий рівень безпеки.

Структуру основних компонентів програмного забезпечення, а також взаємозв'язки між ними й зовнішніми сервісами, відображено на рис. 4.1.

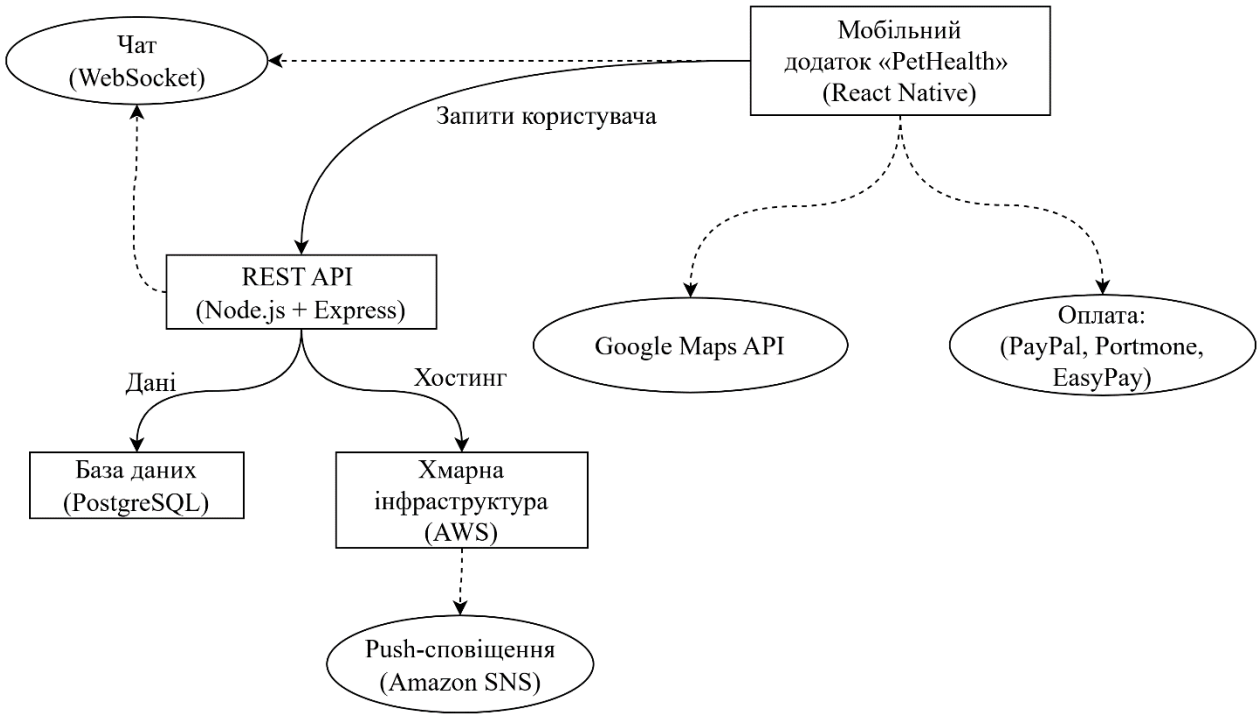

Рис. 4.1. Структура основних компонентів ПЗ

Окрім загальної архітектури, критично важливим для стабільності системи є коректна реалізація внутрішніх алгоритмів обробки даних.

Оскільки додаток працює з персональними даними власників тварин та історією медичних записів, процедура авторизації реалізована з підвищеними вимогами до безпеки. Для забезпечення захищеної передачі даних без необхідності зберігати сесію на сервері використано стандарт JWT (англ. JSON Web Token). Це рішення дозволяє масштабувати систему та забезпечує надійну автентифікацію мобільних клієнтів.

Логіка роботи алгоритму авторизації складається з наступних кроків (рис. 4.2):

1. Користувач вводить облікові дані (електронну пошту та пароль) у мобільному додатку.

2. Додаток передає дані захищеним каналом (HTTPS) на сервер. Пароль ніколи не передається у відкритому вигляді.

3. Сервер здійснює пошук користувача в базі даних PostgreSQL. Якщо користувача знайдено, система порівнює хеш введеного пароля з хешем, що зберігається в базі, використовуючи криптографічну бібліотеку.

4. У разі успішної перевірки сервер генерує пару токенів:

– Access Token (короткостроковий) – для авторизації поточних запитів до API.

– Refresh Token (довгостроковий) – для безпечного оновлення доступу без повторного введення пароля.

5. Токени повертаються клієнту і зберігаються в захищеному сховищі пристрою (SecureStore для iOS або Keystore для Android), що унеможливлює їх викрадення сторонніми програмами.

Процес запису до лікаря є центральною функцією системи. Алгоритм розроблено таким чином, щоб забезпечити цілісність даних (уникнення подвійного бронювання одного часу) та надати користувачеві гнучкість у виборі способу оплати.

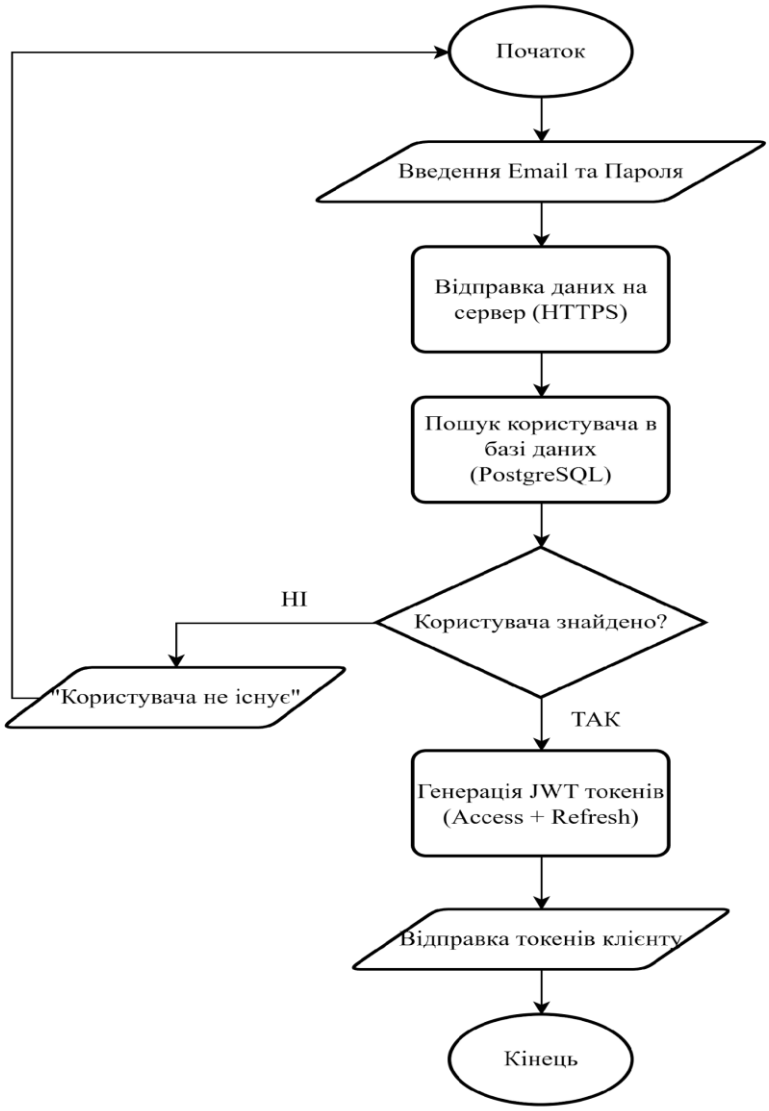

Рис. 4.2. Блок-схема алгоритму авторизації користувача

Логіка роботи алгоритму бронювання (рис. 4.3):

1. Користувач обирає клініку, лікаря, необхідну послугу та бажану дату візиту. Система звертається до бази даних (таблиця Schedule) та відображає лише доступні часові слоти.

2. При виборі конкретного часу система миттєво перевіряє його актуальний статус:

– Якщо слот вже зайнятий іншим користувачем, система пропонує обрати інший час.

– Якщо слот вільний, система встановлює «Тимчасове блокування» (Soft Lock) на 10 хвилин, резервуючи цей час за поточним користувачем на період оформлення заявки.

3. Користувач заповнює інформацію про тварину та причину звернення.

4. Враховуючи специфіку медичних послуг, користувачеві надається вибір:

– Онлайн-оплата: миттєве списання коштів через платіжний шлюз.

– Оплата в клініці: бронювання візиту з оплатою по факту надання послуг.

5. Фіналізація та збереження:

– Система змінює статус слоту в розкладі на «Заброньовано» (is_booked = TRUE).

– У таблиці Appointment створюється новий запис із відповідним статусом оплати.

– Користувачеві та лікарю надсилаються push-сповіщення з підтвердженням успішного запису.

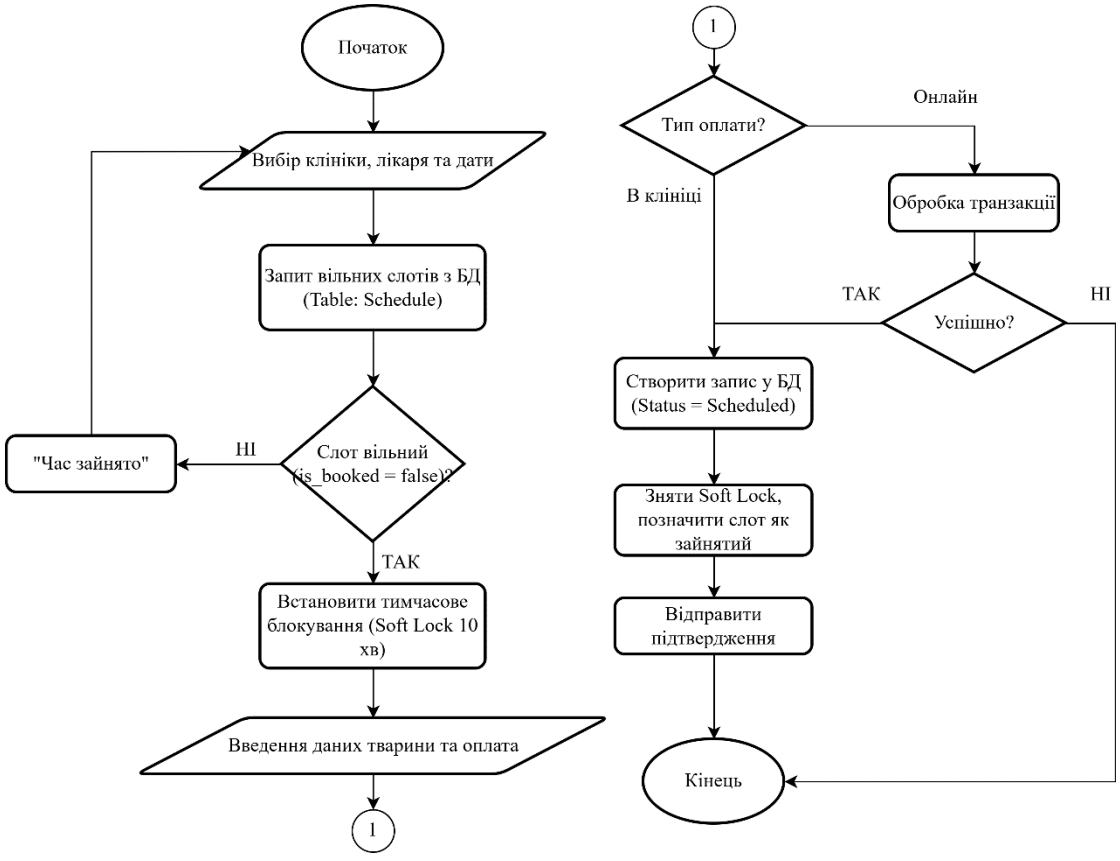

Рис. 4.3. Блок-схема алгоритму онлайн-бронювання

Таким чином, спроєктована архітектура та розроблені алгоритми створюють цілісну основу для функціонування мобільного додатку «PetHealth». Клієнт-серверна модель із використанням хмарних технологій AWS забезпечує необхідну масштабованість та доступність сервісів. Реалізація процедури авторизації через JWT-токени гарантує захист персональних даних, а логіка бронювання запобігає конфліктам у розкладі лікарів.

4.2 Розробка бази даних проєкту

Проєктування бази даних (далі – БД) є важливим етапом розробки інформаційної системи, оскільки забезпечує структуроване зберігання даних, їхню цілісність та ефективну обробку. Основними етапами розробки бази даних є створення концептуальної, логічної та фізичної моделей.

Для реалізації бази даних була обрана система управління БД PostgreSQL [59 - 61]. Це потужна об’єктно-реляційна система керування базами даних (далі – СКБД) з відкритим вихідним кодом, яка широко використовується як у комерційних, так і в наукових проєктах завдяки своїй надійності, розширюваності та високому рівню підтримки стандарту SQL.

Критично важливою перевагою PostgreSQL є її сумісність із сучасними інструментами розробки на базі Node.js [62]. Це дозволяє використовувати технологію ORM (англ. Object-Relational Mapping) – підхід, що перетворює дані з бази безпосередньо в об'єкти JavaScript/TypeScript. Використання ORM-бібліотек (таких як TypeORM, Prisma або Sequelize) дозволяє розробникам абстрагуватися від написання складних SQL-запитів вручну. Замість цього взаємодія з даними відбувається через методи класів, що значно прискорює розробку, робить код чистішим та автоматично захищає систему від поширених вразливостей, таких як SQL-ін'єкції.

На відміну від класичних реляційних баз, PostgreSQL підтримує роботу з неструктурованими даними у форматі JSON. Це надає системі унікальну гібридну гнучкість: основні сутності (користувачі, платежі) зберігаються у вигляді суворих реляційних таблиць для забезпечення цілісності, тоді як

специфічні атрибути (наприклад, деталі медичної картки, які можуть відрізнятися для різних видів тварин) можуть зберігатися у вигляді JSON-документів без необхідності зміни схеми БД.

PostgreSQL розповсюджується за вільною ліцензією. Це дозволяє використовувати інструментарій корпоративного рівня, що підтримує високі навантаження, без жодних фінансових витрат на ліцензування. Для стартап-проєкту це означає можливість оптимізації бюджету та перенаправлення коштів на розробку функціонала та маркетинг.

4.2.1 Концептуальна модель бази даних

Концептуальна модель бази даних є першим кроком у розробці структури даних, що відображає основні об'єкти та їхні взаємозв’язки в межах предметної області. Вона служить для того, щоб зрозуміти, які дані будуть зберігатися в системі і як ці дані повинні бути організовані з точки зору бізнес-логіки. Концептуальна модель не залежить від конкретної СКБД або технологій, і її мета – надати загальне уявлення про предметну область для подальшого переходу до логічної моделі.

Вона відображається у вигляді ER-діаграми, що містить такі основні компоненти:

1. Сутності (англ. Entities) – об'єкти, що зберігаються в базі даних, зображені у вигляді прямокутників.

2. Атрибути (англ. Attributes) – характеристики сутностей, які зберігаються у вигляді овалів, пов’язаних із відповідними сутностями.

3. Зв’язки (англ. Relationships) – взаємозв’язки між сутностями, що зображуються ромбами.

Для мобільного застосунку «PetHealth» концептуальна модель включає такі сутності:

1. Власник (англ. Owner) – описує власника тварини.

2. Тварина (англ. Pet) – містить інформацію про домашню тварину.

3. Клініка (англ. Clinic) – описує ветеринарні клініки.

4. Запис на приймання (англ. Appointment) – містить інформацію про заплановані приймання.

5. Послуга (англ. Service) – представляє доступні послуги в клініках.

6. Лікар (англ. Veterinarian) – описує медичного спеціаліста.

7. Розклад (англ. Schedule) – описує доступні слоти часу.

8. Оплата (англ. Payment) – описує фінансову транзакцію.

9. Відгук (англ. Review) – описує оцінку та коментар клієнта.

Зв’язки між сутностями:

1. Owner – Pet (один-до-багатьох): один власник може мати кілька тварин.

2. Clinic – Service (один-до-багатьох): одна клініка може надавати багато послуг.

3. Clinic – Veterinarian (один-до-багатьох): в одній клініці працює багато лікарів.

4. Veterinarian – Schedule (один-до-багатьох): один лікар має багато слотів у розкладі.

5. Veterinarian – Appointment (один-до-багатьох): лікар проводить багато приймань.

6. Pet – Appointment (один-до-багатьох): тварина може мати історію візитів.

7. Appointment – Payment (один-до-одного): один візит оплачується одним чеком.

8. Appointment – Review (один до одного): один запис може мати один відгук.

Концептуальна модель БД для мобільного додатку «PetHealth» зображена на рис. 4.4.

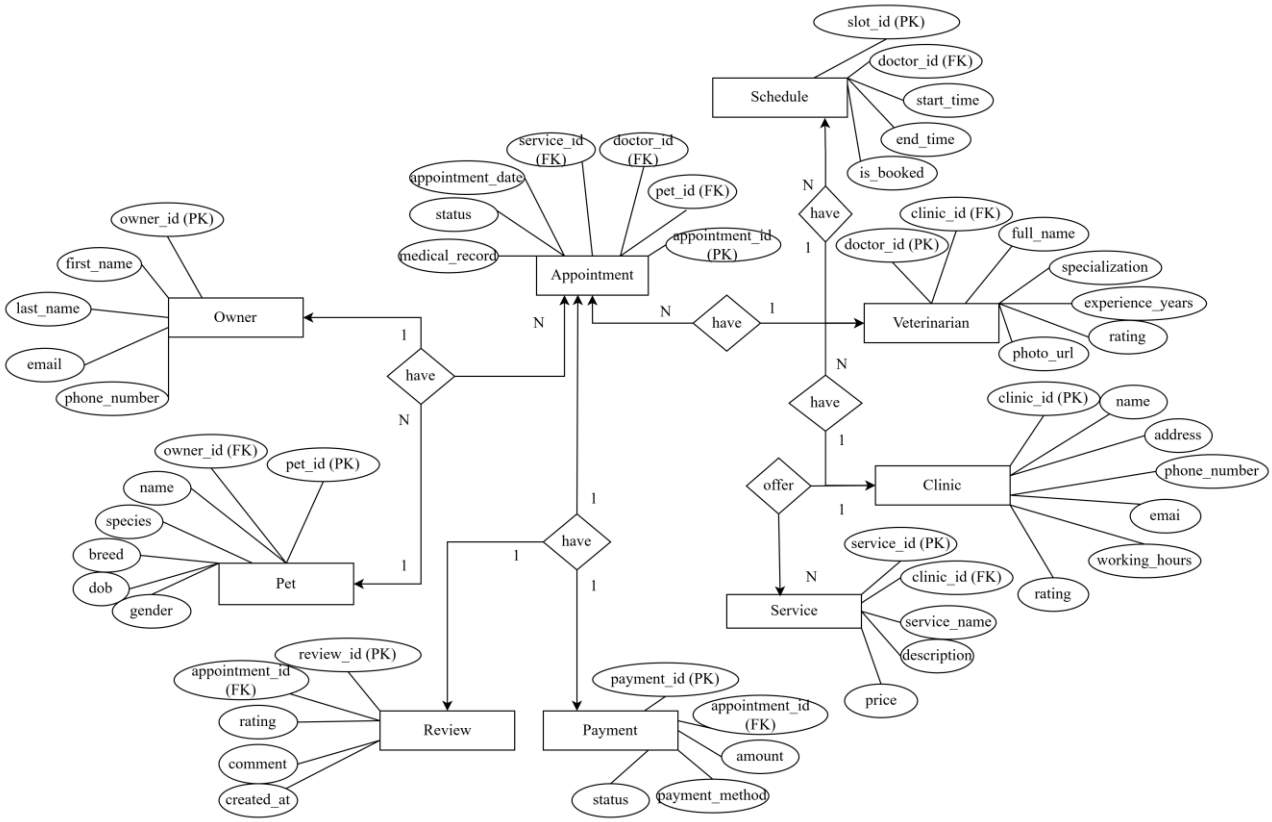

Рис. 4.4. Концептуальна модель БД для мобільного додатку

Представлена концептуальна модель відображає повний набір інформаційних об'єктів, необхідних для реалізації функціональних вимог додатку «PetHealth». Вона чітко фіксує бізнес-правила предметної області, такі як підпорядкованість лікарів клінікам та історію хвороби тварин. Ця схема слугує фундаментом для наступного етапу проектування – розробки логічної моделі та фізичної реалізації бази даних у середовищі PostgreSQL, що передбачає визначення типів даних, первинних ключів та індексів.

4.2.2 Логічна модель бази даних

Логічна модель БД є проміжним етапом між концептуальною моделлю та фізичною реалізацією. Вона деталізує структуру даних, визначаючи склад таблиць, набір атрибутів для кожної сутності, типи зв'язків та ключові поля, необхідні для забезпечення цілісності даних.

Структура логічної моделі для системи «PetHealth» включає наступні сутності та їх атрибути:

1. Owner: owner_id (PK), first_name, last_name, email, phone_number.

2. Pet: pet_id (PK), owner_id (FK), name, species, breed, dob, gender.

3. Clinic: clinic_id (PK), name, address, phone_number, email, working_hours, rating, photo_url.

4. Veterinarian: doctor_id (PK), clinic_id (FK), full_name, specialization, experience_years, rating.

5. Service: service_id (PK), clinic_id (FK), service_name, description, price.

6. Schedule: slot_id (PK), doctor_id (FK), start_time, end_time, is_booked.

7. Appointment: appointment_id (PK), pet_id (FK), doctor_id (FK), service_id (FK), appointment_date, status, medical_notes.

8. Payment: payment_id (PK), appointment_id (FK), amount, payment_method, status.

9. Review: review_id (PK), appointment_id (FK), rating, comment, created_at.

Графічне зображення логічної моделі БД [63] представлено на рис. 4.5.

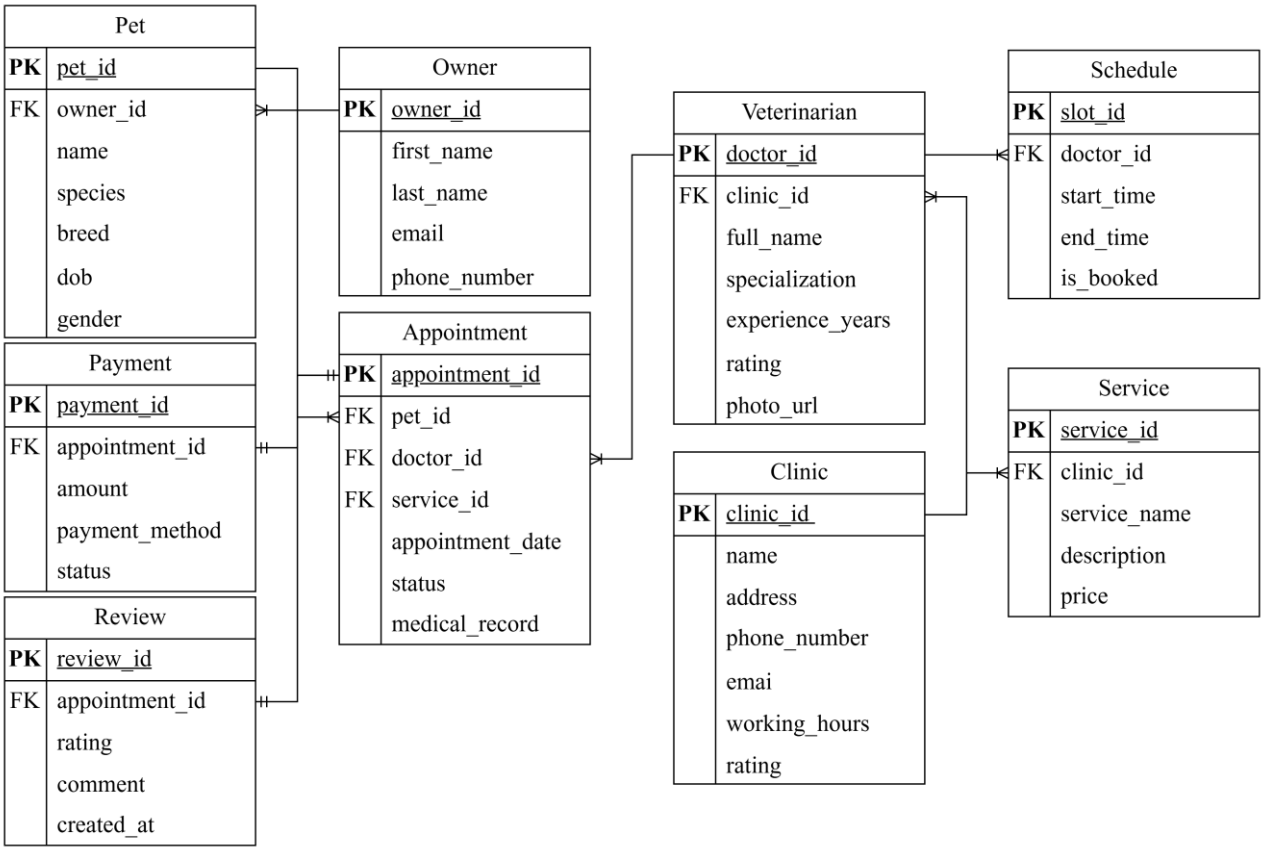

Рис. 4.5. Логічна модель БД для мобільного додатку

Визначена система первинних та зовнішніх ключів гарантує посилальну цілісність, створюючи надійну основу для фізичної реалізації бази даних у середовищі PostgreSQL.

4.2.3 Фізична модель бази даних

Фізична модель бази даних є технічною реалізацією логічної моделі засобами мови SQL для обраної системи управління базами даних СКБД PostgreSQL. На цьому етапі для кожної сутності визначено фізичні параметри: типи даних полів, первинні та зовнішні ключі, обмеження цілісності та налаштування автоінкременту (SERIAL).

Графічне зображення фізичної моделі ЬД наведено на рис. 4.6.

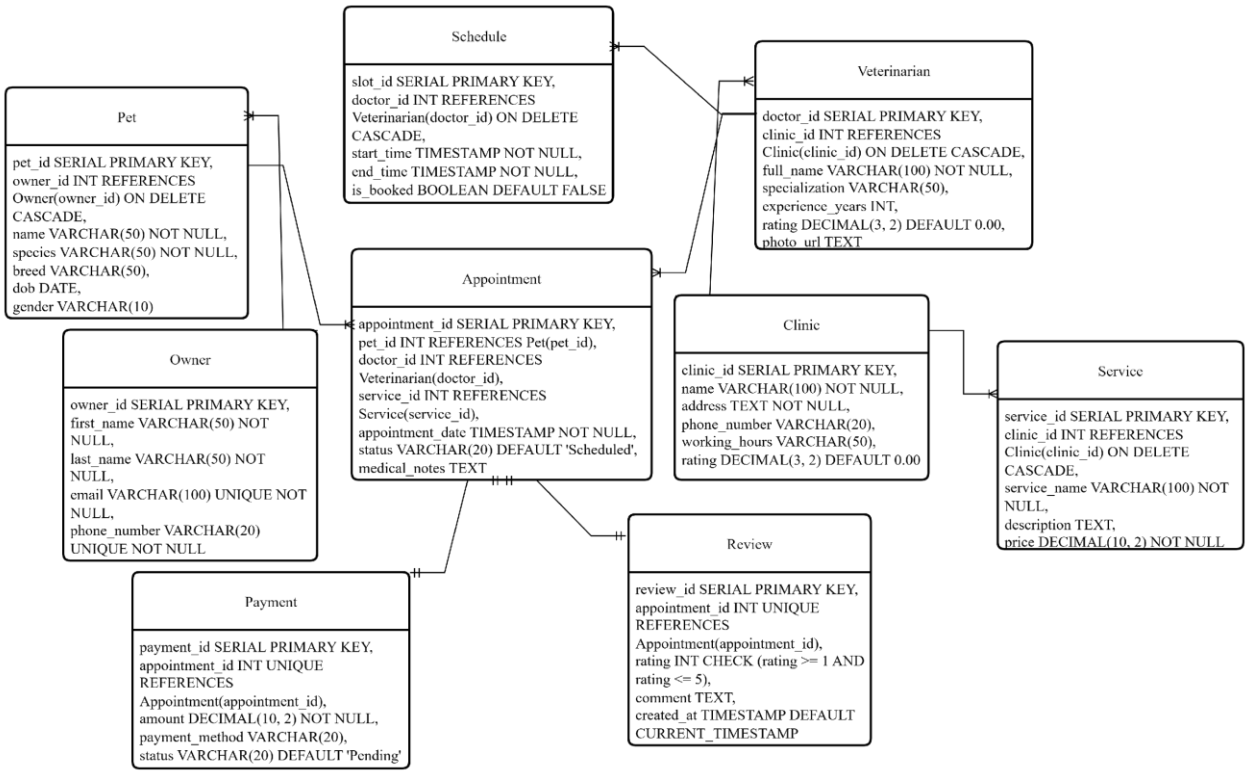

Рис. 4.6. Фізична модель БД для мобільного додатку

Розроблена фізична модель слугує безпосередньою основою для генерації DDL-скриптів (англ. Data Definition Language). Визначення точних типів даних та обмежень (NOT NULL, DEFAULT) дозволяє оптимізувати використання дискового простору та забезпечити високу швидкість виконання запитів у середовищі PostgreSQL.

Нижче наведено детальний опис фізичної структури кожної таблиці.

Таблиця Owner зберігає облікові дані користувачів. Для полів email та phone_number встановлено обмеження унікальності (табл. 4.1).

Таблиця 4.1

Структура таблиці Owner

Стовпець	Тип даних	Опис
owner_id	SERIAL	Унікальний ідентифікатор власника тварини
first_name	VARCHAR(50)	Ім'я власника (макс. 50 символів)
last_name	VARCHAR(50)	Прізвище власника (макс. 50 символів)
email	VARCHAR(100)	Унікальний email власника (макс. 100 символів)
phone_number	VARCHAR(20)	Унікальний номер телефону власника (макс. 20 символів)

Таблиця Pet зберігає профілі тварин. Налаштовано каскадне видалення: при видаленні власника видаляються і його тварини (табл. 4.2).

Таблиця 4.2

Структура таблиці Pet

Стовпець	Тип даних	Опис
pet_id	SERIAL	Унікальний ідентифікатор тварини
owner_id	INT	Зовнішній ключ на Owner
name	VARCHAR(50)	Кличка тварини (макс. 50 символів)
species	VARCHAR(50)	Вид (макс. 50 символів)
breed	VARCHAR(50)	Порода (макс. 50 символів)
dob	DATE	Дата народження
gender	VARCHAR(10)	Стать (перелік: Male/Female/Other)

Таблиця Clinic містить реєстр ветеринарних закладів (табл. 4.3).

Таблиця 4.3

Структура таблиці Clinic

Стовпець	Тип даних	Опис
1	**2**	**3**
clinic_id	SERIAL	Унікальний ідентифікатор закладу
name	VARCHAR(100)	Назва клініки (макс. 100 символів)
address	TEXT	Повна адреса
phone_number	VARCHAR(20)	Контактний телефон (макс. 20 символів)

Закінчення табл. 4.3

1	2	3
email	VARCHAR(100)	Email для зв'язку (макс. 100 символів)
working_hours	VARCHAR(50)	Робочі години (макс. 50 символів)
rating	DECIMAL(3,2)	Рейтинг (за замовчуванням 0.00)

Таблиця Veterinarian описує профілі лікарів. Поле photo_url зберігає посилання на фотографію у хмарному сховищі (табл. 4.4).

Таблиця 4.4

Структура таблиці Veterinarian

Стовпець	Тип даних	Опис
doctor_id	SERIAL	Унікальний ідентифікатор лікаря
clinic_id	INT	Зовнішній ключ на Clinic
full_name	VARCHAR(100)	ПІБ лікаря
specialization	VARCHAR(50)	Спеціалізація
experience	INT	Стаж роботи (років)
rating	DECIMAL(3,2)	Персональний рейтинг
photo_url	TEXT	URL фотографії

Таблиця Service описує послуги та їх вартість. Тип DECIMAL забезпечує точність фінансових даних (табл. 4.5).

Таблиця 4.5

Структура таблиці Service

Стовпець	Тип даних	Опис
service_id	SERIAL	Унікальний ідентифікатор послуги
clinic_id	INT	Зовнішній ключ на Clinic
service_name	VARCHAR(100)	Назва послуги
description	TEXT	Опис процедури
price	DECIMAL(10,2)	Вартість послуги (грн)

Таблиця Schedule містить слоти для запису (табл. 4.6).

Таблиця 4.6

Структура таблиці Schedule

Стовпець	Тип даних	Опис
1	2	3
slot_id	SERIAL	Унікальний ідентифікатор слота
doctor_id	INT	Зовнішній ключ на Veterinarian
slot_id	SERIAL	Унікальний ідентифікатор слота
start_time	TIMESTAMP	Час початку слоту

Закінчення табл. 4.6

1	2	3
end_time	TIMESTAMP	Час закінчення слоту
is_booked	BOOLEAN	Статус зайнятості (True/False)

Таблиця Appointment центральна таблиця, що пов'язує клієнта, лікаря та послугу (табл. 4.7).

Таблиця 4.7

Структура таблиці Appointment

Стовпець	Тип даних	Опис
appointment _id	SERIAL	Унікальний ідентифікатор візиту
pet_id	INT	Зовнішній ключ на Pet
doctor_id	INT	Зовнішній ключ на Veterinarian
service_id	INT	Зовнішній ключ на Service
appointment_date	TIMESTAMP	Дата та час візиту
status	VARCHAR(20)	Статус запису
medical_notes	TEXT	Результати огляду

Таблиця Payment має зв'язок «один-до-одного» із записом (UNIQUE constraint) (табл. 4.8).

Таблиця 4.8

Структура таблиці Payment

Стовпець	Тип даних	Опис
payment_id	SERIAL	Унікальний ідентифікатор оплати
appointment_id	INT	Зовнішній ключ
amount	DECIMAL(10,2)	Сума транзакці
payment_method	VARCHAR(20)	Метод оплати
status	VARCHAR(20)	Статус оплати

Таблиця Review дозволяє залишати оцінку (1-5) після завершення візиту (табл. 4.9).

Таблиця 4.9

Структура таблиці Review

Стовпець	Тип даних	Опис
review_id	SERIAL	Унікальний ідентифікатор відгука
appointment_id	INT	Зовнішній ключ
rating	INT	Оцінка (Check 1-5)
comment	VARCHAR(20)	Текст відгуку
created_at	VARCHAR(20)	Час створення

Повний програмний код (SQL-скрипт) для створення бази даних наведено у Додатку Ж кваліфікаційної роботи.

4.3 Розробка інтерфейсів програмного забезпечення

Однією з ключових складових розробки мобільного додатку «PetHealth» є створення зручного, інтуїтивно зрозумілого та візуально привабливого інтерфейсу користувача. Інтерфейс має забезпечувати логічну навігацію, швидкий доступ до основних функцій додатку, а також адаптацію до різних типів мобільних пристроїв і розмірів екранів [64].

Для розробки інтерфейсу було використано Figma – сучасний онлайн-інструмент для проєктування графічних макетів та створення інтерактивних прототипів [65]. Його перевагами є підтримка спільної роботи в реальному часі, кросплатформеність, можливість створення адаптивного дизайну, а також швидке тестування логіки взаємодії між елементами інтерфейсу ще до початку програмної реалізації.

У рамках цієї роботи у Figma було реалізовано інтерактивний прототип мобільного додатку, що відображає основну структуру сервісу «PetHealth». Прототип охоплює всі ключові процеси – від реєстрації й створення профілю тварини до перегляду медичної картки, пошуку клініки та запису на приймання.

Створення прототипу відбувалося з урахуванням основних принципів зручності користування та адаптивного дизайну. Кожен екран має чітку структуру, яка допомагає користувачеві швидко орієнтуватися в системі. Дизайн побудований з дотриманням ієрархії інформації, використано піктограми, кнопки зі зрозумілими написами, а також візуальні акценти, які спрощують сприйняття. При цьому окрему увагу приділено базовим принципам доступності інтерфейсу, зокрема контрастності кольорів і розміру шрифтів.

У структурі прототипу реалізовані наступні ключові екрани: (1) Екран входу з автентифікацією через пошту. (2) Головний екран з доступом до основних розділів. (3) Екран профілю користувача. (4) Екран додавання домашньої тварини. (5) Медична картка тварини з історією щеплень і візитів.

(6) Екран пошуку клініки з фільтрами за рейтингом, місцем розташування, спеціалізацією. (7) Екран запису на приймання із вибором дати, часу та лікаря.

На рис. 4.7-4.13 наведено приклади основних елементів інтерфейсу, які реалізовано у Figma [66].

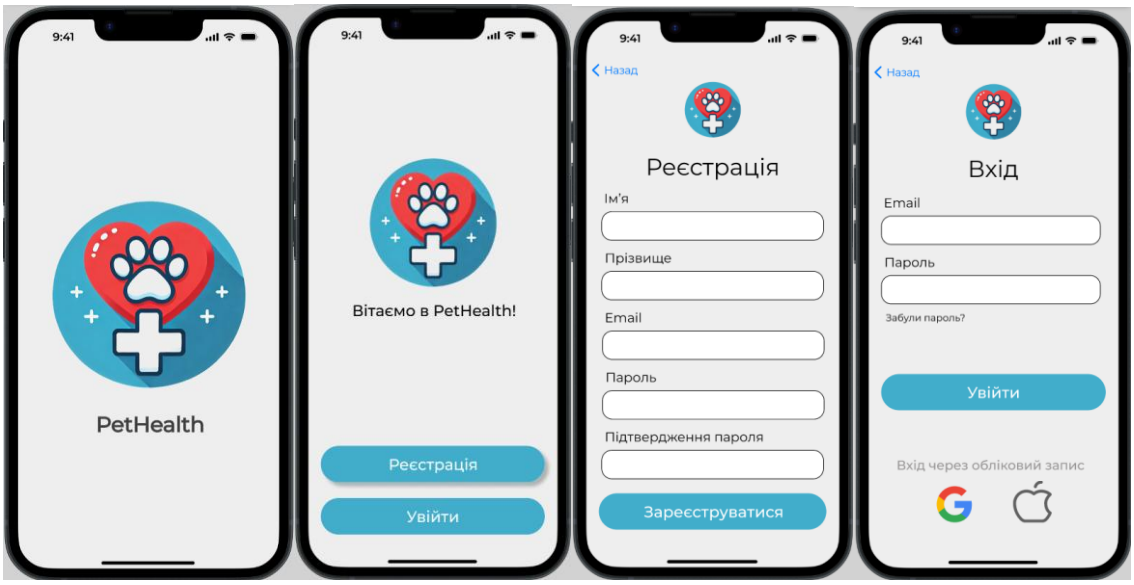

Рис. 4.7. Екран входу з автентифікацією через пошту

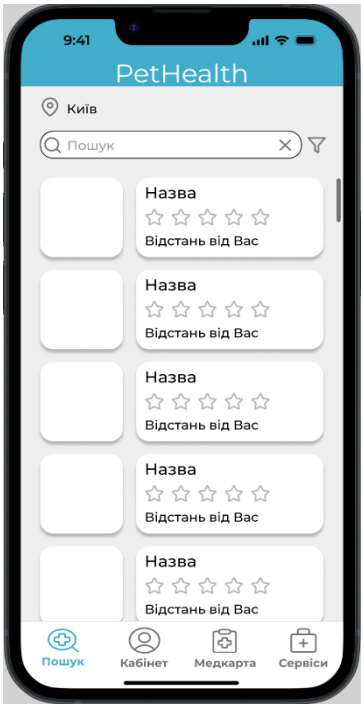

Рис. 4.8. Головний екран з доступом до основних розділів

Рис. 4.9. Екрани профілю користувача та додавання домашньої тварини

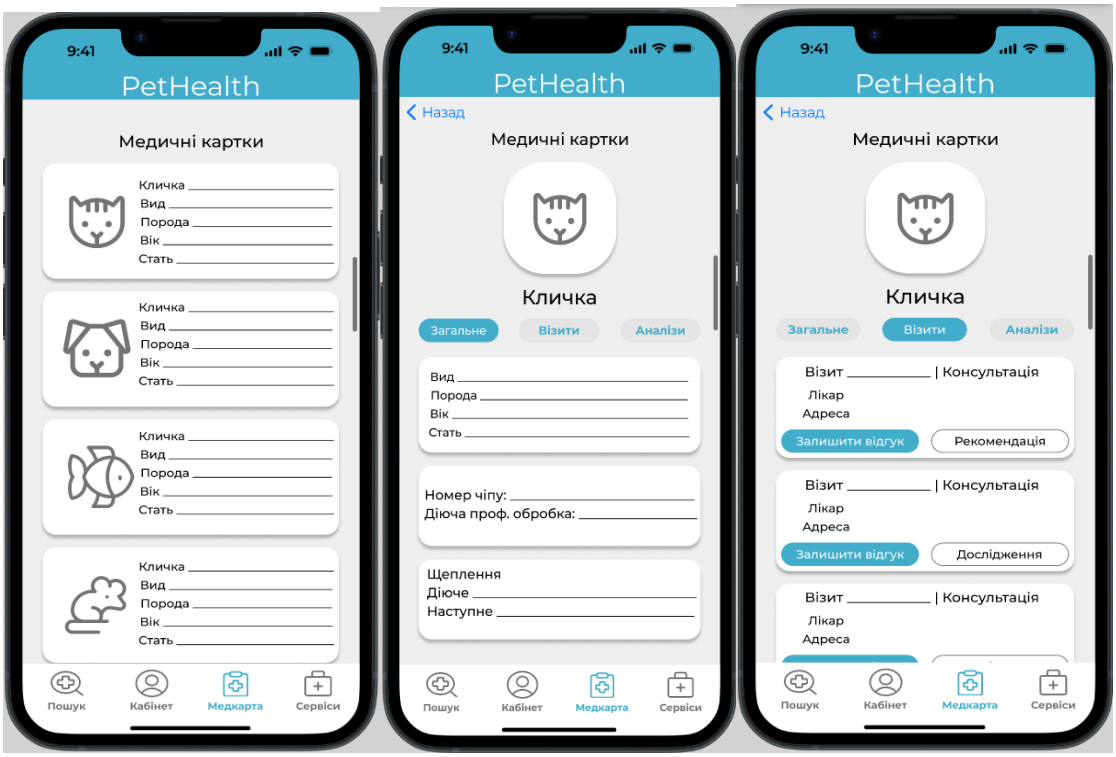

Рис. 4.10. Медична картка тварини з історією щеплень і візитів

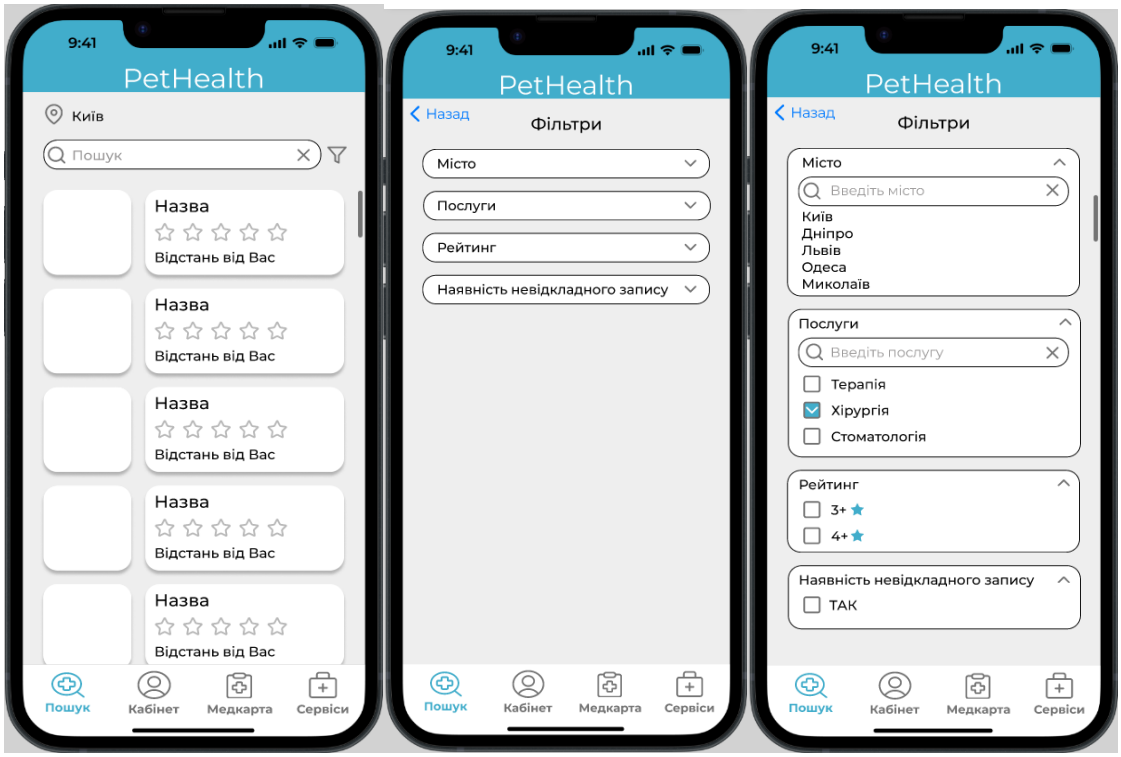

Рис. 4.11. Екран пошуку клініки з фільтрами за рейтингом, місцем розташування, спеціалізацією

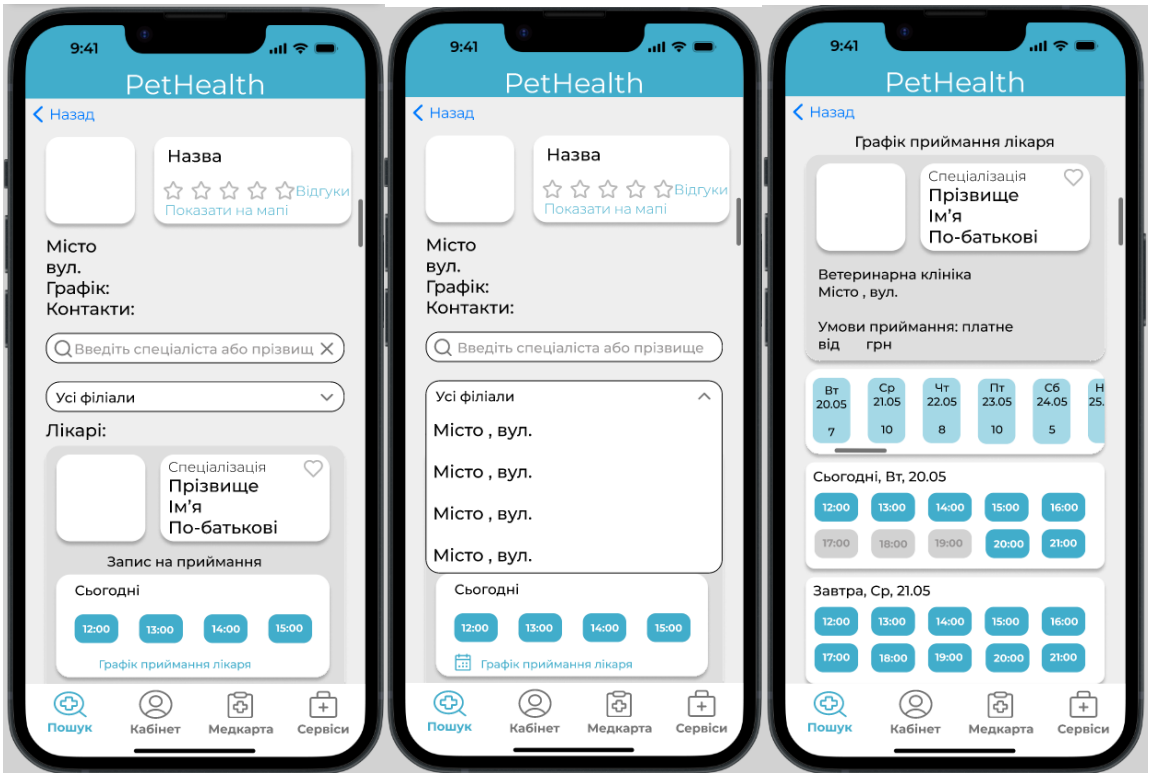

Рис. 4.12. Екран запису на приймання із вибором дати, часу та лікаря

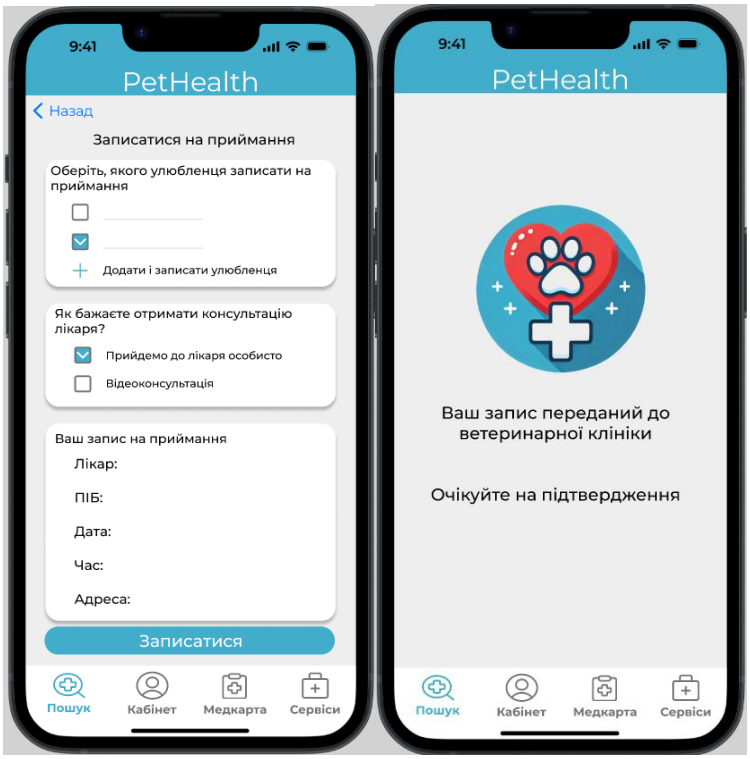

Рис. 4.13. Екран запису на приймання із вибором дати, часу та лікаря

Представлені інтерфейси створюють основу для реалізації зручного та доступного мобільного додатку відповідно до поставлених функціональних вимог.

4.4 Управління якістю

Забезпечення якості програмного забезпечення проєкту «PetHealth» реалізується через комплексний підхід, що включає планування цільових показників якості майбутнього програмного продукту та контроль якості процесів проєктування на етапі створення прототипу. Такий розподіл дозволяє гарантувати відповідність продукту технічним вимогам ще до початку написання коду.

Для оцінки якості мобільного додатку «PetHealth» обрано модель якості згідно з міжнародним стандартом ISO/IEC 25010. З огляду на специфіку проєкту (медична інформаційна система, робота з персональними даними, мобільний інтерфейс), ключовими характеристиками визначено: функціональну придатність, безпеку, надійність та зручність використання.

На основі нефункціональних вимог, визначених у розділі 3 (табл. 3.2), сформовано план оцінки якості з конкретними метриками та критеріями прийнятності (табл. 4.10).

Таблиця 4.10

План оцінки якості продукту

Характе-ристика	Метрика	Метод збору даних	Критерій прийнятності
Безпека	Рівень шифрування даних	Аудит безпеки коду	Відповідність AES-256 (дані), TLS 1.3 (канал)
Надійність	Доступність системи	Моніторинг серверів	≥ 99.9% протягом місяця
Зручність використа-ння	Час виконання ключової операції (запис)	Юзабіліті-тестування	≤ 2 хвилин для нового користувача
Продуктив-ність	Споживання оперативної пам'яті	Профілювання додатку	≤ 200 МБ на клієнтському пристрої
Сумісність	Підтримка ОС	Тестування на емуляторах	Android (версії 9.0 і вище) та iOS (версії 14.0 і вище)

Оскільки розробка проєкту знаходиться на етапі створення інтерактивного прототипу, забезпечення якості фокусується на верифікації інтерфейсних рішень. Для цього використано метод трасування вимог та оцінювання зрілості процесів згідно зі стандартом ISO/IEC 33063.

Для гарантування того, що всі вимоги знайшли своє відображення у візуальному прототипі, побудовано трасувальну матрицю (табл. 4.11). Вона встановлює зв'язок: «Вимога – Екран інтерфейсу – Тест-кейс». Це дозволяє переконатися, що функціонал не був пропущений на етапі дизайну.

Таблиця 4.11

Трасувальна матриця вимог проєкту

Вимога (ID)	Елемент інтерфейсу	Тест-кейс	Статус
1	**2**	**3**	**4**
FR001 (Реєстрація)	Екран Sign Up, форма введення даних	ТС-01: Перевірка наявності всіх полів та переходу на «Пошук» після кліку «Реєстрація».	Виконано

Закінчення табл. 4.11

1	2	3	4
FR002 (Авторизація)	Екран Login, кнопки соцмереж	ТС-02: Симуляція переходу при виборі «Увійти через Google» та валідація полів логіну.	Виконано
FR003 (Профілі)	Екран My Pets, форма Add Pet	ТС-03: Перевірка зручності додавання фото тварини та заповнення картки (кличка, порода).	Виконано
FR004 (Пошук)	Екран Search, фільтри	ТС-04: Тестування роботи фільтрів (за рейтингом/послугою) та відображення результатів.	Виконано
FR005 (Онлайн-запис)	Екран Booking, віджет календаря	ТС-05: Перевірка сценарію (User Flow) вибору вільного слоту часу та лікаря.	Виконано
FR006 (Медкарта)	Екран Pet History	ТС-06: Візуальна перевірка читабельності історії вакцинацій та попередніх візитів.	Виконано
FR007 (Оплата)	Екран Checkout	ТС-07: Перевірка наявності кнопок вибору платіжної системи (Apple Pay/Google Pay).	В макеті
FR011 (Телемедицина)	Екран Chat, відеодзвінок	ТС-11: Перевірка розташування кнопки відеодзвінка та вікна чату на мобільному екрані.	В макеті
FR012 (Модерація)	Веб-панель адміністратора	ТС-12: Перевірка інтерфейсу списку відгуків та кнопок модерації.	Заплановано
FR013 (Користувачі)	Веб-панель адміністратора	ТС-13: Перевірка доступності функції блокування користувача в адмін-панелі.	Заплановано

Складання трасувальної матриці забезпечило наскрізну верифікацію проєктних рішень. Встановлений взаємозв'язок між ідентифікаторами функціональних вимог та елементами графічного інтерфейсу підтверджує повноту покриття технічного завдання на етапі прототипування, що мінімізує ризик виникнення дефектів проєктування.

Для визначення ефективності управління проєктом було проведено самооцінку процесів життєвого циклу розробки (на етапі проєктування) за шкалою від 0 до 5 (0 – процес відсутній, 5 – оптимізований). Оцінювалися

процеси управління вимогами, валідації дизайну та забезпечення якості UX (табл. 4.12).

Таблиця 4.12

Оцінка зрілості процесів проєктування

Процес	Мета процесу	Рівень	Коментар
Управління вимогами	Узгодження функціоналу до початку проєктування	4	Вимоги детально формалізовані (User Stories, див. Додаток А) та стали основою для структури екранів у Figma.
Валідація (Дизайн)	Перевірка прототипу на відповідність вимогам	2	Усі екрани намальовані у Figma, але вони статичні.
Забезпечення якості (QA)	Контроль юзабіліті та стандартів	2	Проведена візуальна перевірка кольорів та шрифтів. Тестування на реальних користувачах ще не робили.

На основі проведеної оцінки виявлено необхідність перетворення статичних макетів на інтерактивний прототип для проведення повноцінного тестування з користувачами (табл. 4.13).

Таблиця 4.13

Оцінка зрілості процесів проєктування

№	Процес	Проблема	Рекомендація	Відповідальний /Термін
1	Прототипування	Статичні макети не дозволяють перевірити зручність переходів між екранами.	Налаштувати інтерактивні зв'язки у Figma для симуляції роботи додатку.	UI/UX Designer / 3 тижні
2	Передача в розробку	Розробникам важко зрозуміти логіку поведінки кнопок без клікабельного прототипу.	Створити UI Kit та карту екранів з описом логіки переходів.	UI/UX Designer / 2 тижні
3	Забезпечення якості	Ризик проблем із доступністю тексту.	Перевірити макети через плагін «Contrast Checker» на відповідність стандарту WCAG 2.1.	QA Engineer / 3 дні

У розділі реалізовано програмну архітектуру системи на основі технологій React Native та Node.js, а також спроєктовано реляційну базу даних PostgreSQL для надійного зберігання інформації. Розроблено дизайн інтерфейсів у середовищі Figma, що забезпечує зручну взаємодію користувачів із функціоналом додатку Також сформовано систему управління якістю. Розроблено трасувальну матрицю вимог та план тестування, що гарантує відповідність розроблених рішень технічним завданням та високий рівень безпеки продукту.

Узагальнюючи результати виконаної роботи, можна стверджувати, що спроєктований мобільний додаток «PetHealth» є актуальним та затребуваним рішенням на ринку ветеринарних послуг України. Його впровадження допоможе користувачам суттєво спростити догляд за домашніми тваринами: власники отримають інструмент для миттєвого онлайн-запису та доступу до електронної медкарти, а клініки зможуть автоматизувати адміністративні процеси та зменшити навантаження на персонал. Комплексне опрацювання технічних, фінансових та управлінських аспектів проєкту підтверджує його готовність до реалізації, а ціль кваліфікаційної роботи магістра, присвячена проєктуванню цього додатку, досягнута в повному обсязі.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи на тему «Дослідження процесів управління проєктом створення мобільного додатку «PetHealth» було досягнуто поставлену мету та послідовно реалізовано всі визначені завдання.

Обґрунтовано актуальність створення цифрового рішення для ветеринарної сфери. Проведений PEST-аналіз та SWOT-аналіз підтвердив високу доцільність проєкту «PetHealth» в умовах цифровізації. Визначено ключову конкурентну перевагу – створення незалежної цифрової медичної картки, що вирішує проблему фрагментації даних.

Розроблено концептуальну модель додатку, яка структурує систему на ключові підсистеми та визначає взаємодії із зовнішніми сервісами. Сформульовано формалізовану математичну модель на основі теорії множин, що включає множини ключових об'єктів та функції, забезпечуючи логічне представлення процесів обробки медичних даних.

Обґрунтовано вибір методології Scrum як найбільш адаптивної та гнучкої для ІТ-проєктів з високою невизначеністю вимог. Створено структурну декомпозицію робіт для візуалізації повного обсягу робіт, а також організаційну структуру проєкту на базі крос-функціональної Scrum-команди. Сформовано беклог продукту на основі деталізованих користувацьких історій та критеріїв приймання, що дозволило перейти до фази планування спринтів.

Проведено ідентифікацію 30 потенційних ризиків, з яких за допомогою методу експертних оцінок виділено 5 найбільш критичних загроз. Для цих ризиків розроблено протиризикові заходи. Здійснено розрахунок бюджету проєкту на річний життєвий цикл, що включає резервний фонд, необхідний для нівелювання фінансових загроз.

Розроблено архітектуру програмного забезпечення та логічну модель бази даних, що забезпечує цілісність, масштабованість та ефективне зберігання медичних записів. Спроектовано ключові інтерфейси користувача для функціональних модулів, які повністю відповідають функціональним вимогам, принципам зручності та інтуїтивності навігації.

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. PMI Ukraine Chapter | PMBOK 7. PMI Ukraine Chapter | Управління проєктами та сертифікації PMI. URL: https://pmiukraine.org/pmbok7 (дата звернення: 07.11.2025).

2. Singh R. Top 16 Project Management Methodologies [2025] | IPM. Institute of Project Management (IPM) Ireland. URL: https://projectmanagement.ie/blog/project-management-methodologies/ (дата звернення: 07.11.2025).

3. Діаграма Ганта: що це таке, як побудувати + приклади. Worksection. URL: https://worksection.com/ua/blog/what-is-gantt-chart.html (дата звернення: 07.11.2025).

4. Use critical path method (CPM) for project management [2025] asana. Asana. URL: https://asana.com/resources/critical-path-method (дата звернення: 11.11.2025).

5. Standish Chaos Report 2021 – SUCCESS THROUGH SAFE. SUCCESS THROUGH SAFE. URL: https://www.successthroughsafe.com/blog-1/2021/11/13/standish-chaos-report-2021 (дата звернення: 07.11.2025).

6. Manifesto for Agile Software Development. Manifesto for Agile Software Development. URL: https://agilemanifesto.org/ (дата звернення: 07.11.2025).

7. Сазерленд Дж. Scrum: навчись робити вдвічі більше за менший час / пер. з англ. Я. Лебеденко. Харків : Клуб сімейн. дозвілля, 2022. 280 с.

8. Home | Scrum Guides. Home | Scrum Guides. URL: https://scrumguides.org/ (дата звернення: 07.11.2025).

9. Carmichael A., Anderson D. J. Essential Kanban Condensed. Blue Hole Press, 2016. 100 с.

10. Agile Practice Guide. Project Management Institute, 2018. 210 с.

11. Predictive, adaptive and hybrid approaches - mudassir iqbal. Mudassir Iqbal. URL: https://mudassiriqbal.net/predictive-adaptive-and-hybrid-approaches/ (дата звернення: 10.11.2025).

12. Pet Tech Market Size, Share & 2030 Growth Trends Report. Mordor Intelligence. URL: https://www.mordorintelligence.com/industry-reports/pet-tech-market (дата звернення: 07.11.2025).

13. Pet Tech Market Size and Outlook 2030. Market Research Reports, Marketing Research Company USA, Market Size | TechSci Research. URL: https://www.techsciresearch.com/report/pet-tech-market/21999.html (дата звернення: 07.11.2025).

14. (MRFR) M. R. F. Exploring the Growth of the Pet Tech Market: Trends, Drivers, and Future Outlook (2025-2034). openPR.com. URL: https://www.openpr.com/news/4099320/exploring-the-growth-of-the-pet-tech-market-trends-drivers (дата звернення: 07.11.2025).

15. Pet Tech Market Size & Share, Statistics Report 2032. Global Market Insights Inc. URL: https://www.gminsights.com/industry-analysis/pet-tech-market (дата звернення: 07.11.2025).

16. Interoperability of heterogeneous health information systems: a systematic literature review - PMC. PMC Home. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9875417/ (дата звернення: 07.11.2025).

17. Problems and Barriers Related to the Use of Digital Health Applications: Scoping Review - PMC. PMC Home. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC10221513/ (дата звернення: 07.11.2025).

18. Data Privacy Concerns Using mHealth Apps and Smart Speakers: Comparative Interview Study Among Mature Adults - PMC. PMC Home. URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC9237761/ (дата звернення: 07.11.2025).

19. Даніліна Т., Зюзюн В. Теоретичне обґрунтування розробки мобільного додатку “PETHEALTH: ветеринарний помічник”. 2st international scientific and practical conference «Information Systems and Technology: Results and Prospects» (IST 2025), Kyiv, Ukraine, 2025, С. 198-200.

20. What Is a PEST Analysis?. Business News Daily. URL: https://www.businessnewsdaily.com/5512-pest-analysis-definition-examples-template.html (дата звернення: 19.11.2025).

21. Interfax-Ukraine. Україна в 2026р. перейде на нові європейські правила у ветеринарній медицині - Держпродспоживслужба. Інтерфакс-Україна. URL: https://interfax.com.ua/news/general/1080337.html (дата звернення: 19.11.2025).

22. Трекер економіки України під час війни - Центр економічної стратегії. Центр економічної стратегії. URL: https://ces.org.ua/tracker-economy-during-the-war/ (дата звернення: 19.11.2025).

23. Реєстрація домашніх тварин стане доступною у ЦНАПах Києва. Офіційний портал КМДА - Головна. URL: https://kyivcity.gov.ua/news/reyestratsiya_domashnikh_tvarin_stane_dostupnoyu_u_tsnapakh_kiyeva/ (дата звернення: 07.11.2025).

24. VisitUkraine Today. Державний реєстр домашніх тварин запрацював в Україні: у чому його суть. Visit Ukraine - SERVICE PORTAL ABOUT UKRAINE 24/7. URL: https://surl.li/nyruih (дата звернення: 07.11.2025).

25. VetHub.pet. VETHUB - Сервіс пошуку та запису до ветеринара у твоєму місті. URL: https://vethub.pet/ (дата звернення: 07.11.2025).

26. Оголошення olx.ua. OLX.ua. URL: https://www.olx.ua/uk/uslugi/uslugi-dlya-zhivotnyh/kiev/q-ветеринар/?srsltid=AfmBOooG8gkQNpIsHJSD9aKZTKkILBcr6EZofKp9mZQ04vPdvTTczrKW (дата звернення: 07.11.2025).

27. Онлайн консультація ветеринара київ – ціни, вартість - kabanchik.ua. Kabanchik.ua. URL: https://kabanchik.ua/ua/kyiv/category/onlain-konsultatsiia-veterynara (дата звернення: 19.11.2025).

28. Ветклініка Zoолюкс. Зоолюкс. URL: https://zoolux.clinic/ (дата звернення: 07.11.2025).

29. Хіґні, Джозеф Основи управління проектами [Електронний ресурс] / Джозеф Хіґні ; пер. з англ. Я. Машико. – 5-те вид. – Харків : Фабула, 2020. – 272 с. – режим доступу: http://elib.chdtu.edu.ua/e-books/4229

30. SWOT аналіз як інструмент створення Стратегії. Простий Брендинг. URL: https://biletska.dp.ua/swot-analiz-iak-instrument-stvorennia-stratehii/ (дата звернення: 17.11.2025).

31. Using problem and objective tree for community development projects. grassrootscollective. URL: https://www.thegrassrootscollective.org/problem-objective-tree-development (дата звернення: 17.11.2025).

32. Морозов В.В., Коломієць А.С. Методи розробки концепцій іт проєктів [Текст]: методичні вказівки для виконання практичних, лабораторних та самостійних робіт з навчальної дисципліни / Морозов В.В., Коломієць А.С. – К. : КНУ імені Тараса Шевченка, 2024. – 60 с.

33. Ziuziun V., Danilina T. Conceptual and Mathematical Justification for the “PETHEALTH” Information System Development Project. Taurida Scientific Herald. Series: Technical Sciences, 2025, 4, T.1, P. 94-102. https://doi.org/10.32782/tnv-tech.2025.4.1.10

34. What is NLP? - Natural Language Processing Explained - AWS. Amazon Web Services, Inc. URL: https://aws.amazon.com/what-is/nlp/#:~:text=Natural%20language%20processing%20(NLP)%20is,manipulate,%20and%20comprehend%20human%20language. (дата звернення: 05.06.2025).

35. Створення чат-бота з нуля: посібник для початківців. Unite.AI. URL: https://www.unite.ai/uk/creating-a-chatbot-from-scratch-a-beginners-guide/ (дата звернення: 05.06.2025).

36. Conversational AI with Language Models | Rasa Documentation. Conversational AI Platform | Superior Customer Experiences Start Here. URL: https://rasa.com/docs/learn/concepts/calm (дата звернення: 06.06.2025).

37. Dialogflow Documentation | Google Cloud. Google Cloud. URL: https://cloud.google.com/dialogflow/docs (дата звернення: 06.06.2025).

38. Maverick A. BERT–Bidirectional Encoder Representations. Medium. URL:https://samanemami.medium.com/bert-bidirectional-encoder-representations-e98833f9dfcd (date of access: 06.06.2025).

39. Tips for A product owner to build A world class healthcare product. Agile & Scrum Courses Training - Professional Certification Training. URL: https://premieragile.com/tips-for-a-product-owner-to-build-a-healthcare-product/ (дата звернення: 17.11.2025).

40. Agile scrum methodology for app development. Appinventiv. URL: https://appinventiv.com/blog/agile-scrum-methodology-in-mobile-app-development/ (дата звернення: 17.11.2025).

41. Top 10 essential cybersecurity tips for veterinary practices | ezyVet. Cloud Veterinary Software | ezyVet. URL: https://www.ezyvet.com/blog/cybersecurity-tips-for-veterinary-practices (дата звернення: 17.11.2025).

42. OWASP mobile top 10 | OWASP foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/www-project-mobile-top-10/ (дата звернення: 17.11.2025).

43. GeeksforGeeks. Functional and non functional requirements - geeksforgeeks. GeeksforGeeks. URL: https://www.geeksforgeeks.org/software-engineering/functional-vs-non-functional-requirements/ (дата звернення: 08.11.2025).

44. User stories and user story examples by mike cohn. Mountain Goat Software. URL: https://www.mountaingoatsoftware.com/agile/user-stories (дата звернення: 17.11.2025).

45. Planning poker: an agile estimating and planning technique. Mountain Goat Software. URL: https://www.mountaingoatsoftware.com/agile/planning-poker (дата звернення: 17.11.2025).

46. Scrum Alliance. Acceptance criteria: everything you need to know plus examples. Home. URL: https://resources.scrumalliance.org/Article/need-know-acceptance-criteria (дата звернення: 17.11.2025).

47. Sprint planning meeting: what it is and how to do it. Mountain Goat Software. URL: https://www.mountaingoatsoftware.com/agile/scrum/meetings/sprint-planning-meeting (дата звернення: 09.12.2025).

48. Guide to using agile gantt charts. Smartsheet. URL: https://www.smartsheet.com/content/agile-gantt#how-gantt-charts-work-with-agile-methodology (дата звернення: 09.12.2025).

49. Use case diagram best practices and examples - justinmind. Justinmind |. URL: https://www.justinmind.com/blog/use-case-diagramming-examples/ (дата звернення: 09.12.2025).

50. Тімінський О.Г., Коломієць А.С. Методи управління ризиками в IT проєктах [Текст]: методичні вказівки до виконання практичних, лабораторних робіт та самостійної роботи для студентів освітньої програми «Управління проєктами» спеціальності 122 «Комп'ютерні науки» для денної і заочної форм навчання / Тімінський О.Г., Коломієць А.С. – К. : КНУ імені Тараса Шевченка, 2021. – 40 с.

51. Understanding time & material vs. fixed price projects in software development: what's best for your needs?. Inspeerity. URL: https://inspeerity.com/blog/understanding-time-material-vs-fixed-price-projects-in-software-development-whats-best-for-your-needs/ (дата звернення: 09.12.2025).

52. Середня зарплата в Україні. Work.ua – сервіс пошуку роботи №1 в Україні. URL: https://www.work.ua/salary-all/ (дата звернення: 09.12.2025).

53. App development costs 2025: complete pricing guide & calculator. Topflight. URL: https://topflightapps.com/ideas/app-development-costs/#32 (дата звернення: 09.12.2025).

54. Contingency reserve in project management. ProjectManager. URL: https://www.projectmanager.com/blog/contingency-reserve (дата звернення: 09.12.2025).

55. The benefits of using react native for mobile development. IEEE Computer Society. URL: https://www.computer.org/publications/tech-news/trends/benefits-of-react-native (дата звернення: 10.06.2025).

56. Srikanth A. How to create a REST API with Node.js and Express. blog.postman.com. URL: https://blog.postman.com/how-to-create-a-rest-api-with-node-js-and-express/ (дата звернення: 11.06.2025).

57. Sending mobile push notifications with Amazon SNS - Amazon Simple Notification Service. URL: https://docs.aws.amazon.com/sns/latest/dg/sns-mobile-application-as-subscriber.html (дата звернення: 11.06.2025).

58. Blog: Three ways to add a map implementation to your app and when to use each Google Maps Platform. Google Maps Platform. URL: https://mapsplatform.google.com/resources/blog/three-ways-add-map-implementation-your-app-and-when-use-each/ (дата звернення: 11.06.2025).

59. PostgreSQL: About. PostgreSQL: The world's most advanced open-source database. URL: https://www.postgresql.org/about/ (дата звернення: 01.11.2025).

60. Vadlamani V. Introduction to PostgreSQL Database Management. PostgreSQL Skills Development on Cloud. Apress, Berkeley, CA, 2024, P. 1-39. https://doi.org/10.1007/979-8-8688-0817-3_1

61. Ravi Kumar Y.V., Samayam A.K., Kadambari P. PostgreSQL System Architecture. Mastering PostgreSQL Administration. Apress, Berkeley, CA, 2025, P. 1-18. https://doi.org/10.1007/979-8-8688-1507-2_1

62. Top 6 orms for modern node.js app development. Amplication Blog. URL: https://amplication.com/blog/top-6-orms-for-modern-nodejs-app-development (дата звернення: 10.12.2025).

63. Ziuziun V., Danilina T. Features of Database Structure Development for the «Pethealth» Mobile Application. Information Technology and Implementation (Satellite): Conference Proceedings, November 21, 2025, Kyiv, Ukraine / Taras Shevchenko National University of Kyiv.

64. The ultimate guide to mobile UX design: principles, best practices, and examples. Thoughts about Product Adoption, User Onboarding and Good UX | Userpilot Blog. URL: https://userpilot.com/blog/mobile-ux-design/ (дата звернення: 09.12.2025).

65. High-Fidelity prototyping: what is it and how can it help? | figma. Figma. URL: https://www.figma.com/resource-library/high-fidelity-prototyping/ (дата звернення: 09.12.2025).

66. Зюзюн В., Даніліна Т. Проєктування цифрової системи супроводу власників тварин: від аналізу вимог до реалізації інтерфейсу. Вісник Кременчуцького національного університету імені Михайла Остроградського, 2025, 6(155).

ДОДАТКИ

Додаток А

Таблиця А.1

User Story мобільного застосунку «PetHealth»

Код US	Формулювання US	Критерії прийняття
1	**2**	**3**
US001 (FR001)	Як власник тварини, я хочу зареєструватися в додатку, щоб створити власний профіль та отримати доступ до послуг.	1. Користувач може ввести ім'я, email/телефон та пароль. 2. Система перевіряє валідність даних (формат email, складність пароля). 3. Користувач отримує код підтвердження (SMS/Email). 4. Після підтвердження створюється обліковий запис.
US002 (FR002)	Як зареєстрований користувач, я хочу авторизуватися в системі, щоб отримати доступ до своїх даних.	1. Вхід можливий через логін/пароль або Google/Apple ID. 2. При помилці виводиться повідомлення «Невірні дані». 3. Доступна кнопка «Забули пароль?» для відновлення доступу. 4. Після успішного входу відкривається Головний екран.
US003 (FR003)	Як власник тварини, я хочу створити профіль свого улюбленця, щоб зберігати його дані в одному місці.	1. Форма містить поля: фото, кличка, вид, порода, дата народження, стать. 2. Можливість редагувати дані профілю. 3. Можливість видалити профіль тварини. 4. Профіль відображається у списку «Мої тварини».
US004 (FR004)	Як власник тварини, я хочу шукати клініки та лікарів, щоб обрати найкращий варіант поблизу.	1. Пошук за назвою, містом або геолокацією. 2. Фільтрація за рейтингом (наприклад, 4+ зірки) та спеціалізацією. 3. Відображення результатів списком та пінами на карті. 4. Картка лікаря містить фото, спеціалізацію та відгуки.

Продовження табл. А.1

1	2	3
US005 (FR005)	Як власник тварини, я хочу записатися на прийом онлайн, щоб зекономити час на телефонні дзвінки.	1. Відображення календаря з доступними слотами лікаря. 2. Вибір типу послуги при записі. 3. Підтвердження запису кнопкою «Записатися». 4. Можливість скасувати запис у «Мої візити».
US006 (FR006)	Як власник тварини, я хочу переглядати медичну картку, щоб бачити історію хвороби та вакцинацій.	1. Список візитів відсортований за датою (від нових до старих). 2. Деталі візиту містять діагноз та призначення. 3. Можливість завантажити результати аналізів (PDF/JPEG). 4. Окремий блок для календаря вакцинацій.
US007 (FR007)	Як власник тварини, я хочу оплачувати послуги через додаток, щоб розраховуватися швидко та безконтактно.	1. Вибір методу оплати (картка, Google Pay, Apple Pay). 2. Відображення суми до сплати перед підтвердженням. 3. Отримання електронного чека після успішної транзакції. 4. Обробка помилок оплати (наприклад, «Недостатньо коштів»).
US008 (FR008)	Як ветеринар, я хочу керувати своїм розкладом, щоб клієнти бачили актуальний час для запису.	1. Інтерфейс календаря з можливістю вибору робочих днів/годин. 2. Можливість блокувати час (обід, відпустка). 3. Налаштування тривалості прийому (наприклад, 30 хв, 1 год).
US009 (FR009)	Як ветеринар, я хочу створювати записи в медкарті, щоб фіксувати хід лікування пацієнта.	1. Форма нового запису містить поля: скарги, діагноз, лікування. 2. Можливість прикріпити файли (фото, документи). 3. Автоматичне підтягування даних пацієнта із запису. 4. Збереження запису робить його видимим для власника.

Закінчення табл. А.1

1	2	3
US010 (FR010)	Як адміністратор клініки, я хочу бачити список записів на день, щоб контролювати потік пацієнтів.	1. Таблиця записів з фільтром по лікарях та датах. 2. Зміна статусу візиту («Очікується», «В роботі», «Завершено», «Не з'явився»). 3. Повідомлення відображаються у центрі сповіщень пристрою.
US011 (FR011)	Як ветеринар, я хочу проводити онлайн-консультації, щоб допомагати пацієнтам дистанційно.	1. Запуск відеодзвінка або чату з картки запису. 2. Можливість обміну текстовими повідомленнями та фото. 3. Таймер тривалості консультації.
US012 (FR012)	Як адміністратор системи, я хочу модерувати відгуки, щоб запобігти спаму та некоректній поведінці.	1. Список нових відгуків зі статусом «На перевірці». 2. Кнопки «Опублікувати» та «Видалити». 3. Можливість переглянути автора відгуку.
US013 (FR013)	Як адміністратор системи, я хочу керувати користувачами, щоб блокувати порушників правил сервісу.	1. Пошук користувача за email або ID. 2. Кнопка «Заблокувати» з вказанням причини. 3. Відображення статусу акаунта (Активний/Заблокований).

Додаток Б

Таблиця Б.1

Беклог продукту проєкту «PetHealth»

Код US	Задача	Підзадача
1	**2**	**3**
US001	T01.1 Реалізація бекенду реєстрації	ST01.1.1 Проєктування схеми бази даних для користувачів.
		ST01.1.2 Створення API для реєстрації.
		ST01.1.3 Реалізація валідації вхідних даних та хешування паролів.
	T01.2 Реалізація фронтенду реєстрації	ST01.2.1 Верстка екрану реєстрації (поля вводу, кнопки).
		ST01.2.2 Інтеграція з API реєстрації та обробка помилок.
		ST01.2.3 Реалізація екрану підтвердження коду.
US002	T02.1 Механізм авторизації	ST02.1.1 Створення API для логіну та генерації токенів.
		ST02.1.2 Реалізація входу через Google/Apple ID.
		ST02.1.3 Верстка екрану входу та логіка збереження сесії.
	T02.2 Відновлення доступу	ST02.2.1 Реалізація API для скидання пароля (надсилання посилання).
		ST02.2.2 Створення екранів «Забули пароль» та «Новий пароль».
US003	T03.1 Управління профілем тварини	ST03.1.1 Створення таблиці Pets у БД та зв'язків з Users.
		ST03.1.2 Розробка API для профілів тварин.
		ST03.1.3 Реалізація завантаження та збереження фото тварини.
	T03.2 Інтерфейс «Мої тварини»	ST03.2.1 Верстка списку карток тварин.
		ST03.2.2 Створення форм додавання та редагування профілю.
US004	T04.1 Пошукова система	ST04.1.1 Розробка алгоритму пошуку з фільтрами.
		ST04.1.2 Інтеграція API карт (Google Maps/Mapbox) для геолокації.
	T04.2 Інтерфейс пошуку	ST04.2.1 Верстка екрану пошуку з фільтрами (рейтинг, тип лікаря).
		ST04.2.2 Реалізація відображення клінік пінами на карті.

Продовження табл. Б.1

1	2	3
US005	T05.1 Логіка бронювання	ST05.1.1 Створення структури БД для записів.
		ST05.1.2 Розробка алгоритму перевірки вільних слотів лікаря.
		ST05.1.3 API для створення, скасування та перенесення запису.
	T05.2 Інтерфейс запису	ST05.2.1 Верстка календаря та вибору часового слоту.
		ST05.2.2 Екран підтвердження бронювання та список «Мої візити».
US006	T06.1 Перегляд медкарти	ST06.1.1 API для отримання історії хвороби конкретної тварини.
		ST06.1.2 Верстка екрану «Медична картка» з хронологією візитів.
	T06.2 Робота з файлами	ST06.2.1 Логіка завантаження та відображення результатів аналізів.
		ST06.2.2 Візуалізація календаря вакцинацій.
US007	T07.1 Платіжна система	ST07.1.1 Інтеграція платіжного шлюзу (наприклад, LiqPay/Stripe).
		ST07.1.2 Реалізація безпечної передачі платіжних даних.
		ST07.1.3 Обробка статусів оплати (успіх/помилка) та генерація чека.
US008	T08.1 Графік лікаря	ST08.1.1 Створення таблиці розкладу у БД.
		ST08.1.2 Інтерфейс календаря для налаштування робочих годин.
		ST08.1.3 Логіка блокування часу (відпустка, перерва).
US009	T09.1 Створення медзаписів	ST09.1.1 Форма лікаря для внесення діагнозу та призначень.
		ST09.1.2 API для збереження запису та прив'язки до пацієнта.
		ST09.1.3 Функціонал прикріплення фото/документів до візиту.
US010	T10.1 Панель адміністратора клініки	ST10.1.1 Верстка дашборду зі списком записів на день.
		ST10.1.2 Логіка фільтрації записів за лікарями та статусами.
		ST10.1.3 Функціонал зміни статусу візиту («Завершено», «Не з'явився»).
US011	T11.1 Телемедицина	ST11.1.1 Налаштування WebSocket для чату в реальному часі.
		ST11.1.2 Інтеграція сервісу для відеодзвінків.
		ST11.1.3 Інтерфейс чату з можливістю надсилання фото.

Закінчення табл. Б.1

1	2	3
US012	T12.1 Модерація	ST12.1.1 Адмін-панель для перегляду нових відгуків.
		ST12.1.2 API для затвердження або видалення відгуків.
		ST12.1.3 Логіка розрахунку рейтингу клініки на основі відгуків.
US013	T13.1 Управління користувачами	ST13.1.1 Таблиця всіх користувачів з пошуком та фільтрацією.
		ST13.1.2 Реалізація механізму блокування користувачів.
		ST13.1.3 Відображення статусу блокування в системі.

Додаток В

Таблиця В.1

План виконання робіт спринту №1

Задача /Підзадача	Назва роботи	Тривалість (год)	Виконавець	Матеріал, технології
1	**2**	**3**	**4**	**5**
ST01.1.1	Виконання SQL-скрипта створення таблиці Users у БД	3	Backend Developer	PostgreSQL
ST01.1.2	Написання коду API маршруту/register	2	Backend Developer	Node.js, Express
ST01.1.2	Програмування контролера створення користувача	4	Backend Developer	JavaScript
ST01.1.3	Кодування схеми валідації даних (email/pass)	3	Backend Developer	Joi Library
ST01.1.3	Підключення бібліотеки хешування паролів	2	Backend Developer	Bcrypt
ST02.1.1	Програмна реалізація генерації JWT токена	4	Backend Developer	JSON Web Token
ST02.1.1	Написання Middleware для перевірки токена	3	Backend Developer	Node.js
ST02.1.2	Налаштування стратегії Google OAuth у коді	5	Backend Developer	Passport.js
ST02.2.1	Конфігурація поштового клієнта (SMTP)	3	Backend Developer	Nodemailer
ST02.2.1	Кодування ендпоінту для скидання пароля	3	Backend Developer	Node.js
ST01.2.1	Верстка екрану реєстрації за готовим макетом	4	Frontend Developer	React Native
ST01.2.1	Стилізація компонентів вводу (Input, Button)	4	Frontend Developer	CSS/Stylesheet
ST01.2.3	Верстка інтерфейсу введення коду підтвердження	4	Frontend Developer	React Native

Закінчення табл. В.1

1	2	3	4	5
ST01.2.2	Інтеграція API реєстрації з клієнтською частиною	2	Frontend Developer	Axios
ST01.2.2	Програмування логіки обробки помилок сервера	3	Frontend Developer	JavaScript
ST02.1.3	Верстка та стилізація екрану авторизації	4	Frontend Developer	React Native
ST02.1.3	Реалізація безпечного зберігання сесії (JWT) на пристрої	3	Frontend Developer	Async Storage
ST02.2.2	Верстка сценарію відновлення пароля	4	Frontend Developer	React Native
ST01.1.2	Виконання тест-кейсів для API реєстрації	3	QA Engineer	Postman
ST02.1.1	Перевірка безпеки механізму авторизації та валідації токенів	4	QA Engineer	Postman
ST01.2.1	UI-тестування форм вводу на мобільному емуляторі	4	QA Engineer	Android Studio
РАЗОМ	Загальний обсяг робіт	71	Команда	

Додаток Г

Таблиця Г.1

Ідентифікація ризиків проекту

№	Тип ризику	Ризикова подія	Сила впливу	Керованість
1	**2**	**3**	**4**	**5**
1	Технічні	Несумісність із застарілим ПЗ ветклінік	Висока	Середня
2	Технічні	Критичні помилки (баги) в коді мобільного додатку	Висока	Висока
3	Технічні	Проблеми інтеграції з Google Maps API	Середня	Середня
4	Технічні	Низька продуктивність додатку на старих смартфонах	Середня	Висока
5	Технічні	Вразливість даних тварин та власників (кіберзагрози)	Середня	Середня
6	Технічні	Збій системи Push-сповіщень (нагадування про візит)	Висока	Висока
7	Технічні	Конфлікт синхронізації даних	Висока	Висока
8	Управлінські	Розповзання меж проєкту	Висока	Висока
9	Управлінські	Нечітко сформульовані вимоги до функціоналу	Висока	Висока
10	Управлінські	Втрата ключового розробника	Висока	Середня
11	Управлінські	Низька мотивація команди через перепрацювання	Середня	Висока
12	Управлінські	Відсутність технічної документації	Середня	Висока
13	Операційні	Відмова в публікації App Store/Google Play	Висока	Низька
14	Операційні	Збій хмарного хостингу (AWS)	Висока	Низька

Закінчення табл. Г.1

1	2	3	4	5
15	Операційні	Збій платіжного шлюзу під час оплати послуг	Середня	Низька
16	Юридичні	Порушення законодавства про захист даних (GDPR)	Висока	Висока
17	Юридичні	Відсутність ліцензій на сторонні бібліотеки	Середня	Висока
18	Юридичні	Претензії від користувачів через некоректні поради	Середня	Середня
19	Фінансові	Перевищення бюджету на хмарні сервіси	Середня	Висока
20	Фінансові	Зростання вартості платних API (Google, SMS)	Середня	Низька
21	Фінансові	Затримка фінансування наступного етапу	Висока	Низька
22	Взаємодія (Клініки)	Саботаж персоналу клінік	Висока	Середня
23	Взаємодія (Клініки)	Відмова клінік від партнерства	Висока	Середня
24	Взаємодія (Користувачі)	Складний інтерфейс для старшої аудиторії	Середня	Висока
25	Взаємодія (Користувачі)	Низька активність користувачів на старті	Висока	Середня
26	Взаємодія (Користувачі)	Негативні відгуки через баги MVP	Висока	Висока
27	Форс-мажорні	Відключення електроенергії (блекаути)	Висока	Середня
28	Форс-мажорні	Втрата інтернет-зв'язку у команди	Висока	Середня
29	Форс-мажорні	Безпекові ризики (війна)	Висока	Низька
30	Форс-мажорні	Різке коливання валютного курсу	Середня	Низька

Додаток Д

Таблиця Д.1

Результати оцінювання ризиків проєкту

№	Ризикова подія	Затримки у часі		Фінансові витрати		Ймовір-ність		Частота		Важливість ризику
		ЯО	КО	ЯО	КО	ЯО	КО	ЯО	КО	
1	2	3	4	5	6	7	8	9	10	11
1	Несумісність із застарілим ПЗ ветклінік	ВС	8	ВН	7	ВС	8	ВС	8	56
2	Критичні помилки (баги) в коді мобільного додатку	СС	5	СС	5	ВН	7	ВС	8	35
3	Проблеми інтеграції з Google Maps API	СС	5	СС	5	СН	4	СС	5	20
4	Низька продуктив-ність додатку на старих смартфонах	НВ	3	СС	5	ВН	7	ВС	8	35
5	Вразливість даних тварин та власників (кіберзагрози)	ВВ	9	ВВ	9	НВ	3	НН	1	27
6	Збій системи Push-сповіщень (нагадування про візит)	СН	4	СВ	6	СС	5	ВН	7	30
7	Конфлікт синхронізації даних	СВ	6	ВВ	9	ВН	7	СВ	6	63

Продовження табл. Д.1

1	2	3	4	5	6	7	8	9	10	11
8	Розповзання меж проєкту	ВС	8	СС	5	ВС	8	СВ	6	40
9	Нечітко сформульовані вимоги до функціоналу	ВС	8	СС	5	ВН	7	СС	5	35
10	Втрата ключового розробника	ВВ	9	ВС	8	СН	4	НС	2	32
11	Низька мотивація команди через перепрацювання	СВ	6	СН	4	СС	5	СН	4	20
12	Відсутність технічної документації	СН	4	НВ	3	ВС	8	ВС	8	24
13	Відмова в публікації App Store / Google Play	ВС	8	СВ	6	СС	5	НС	2	30
14	Збій хмарного хостингу (AWS)	ВВ	9	ВВ	9	НС	2	НН	1	18
15	Збій платіжного шлюзу під час оплати послуг	НВ	3	ВВ	9	НВ	3	СН	4	27
16	Порушення законодавства про захист даних (GDPR)	СВ	6	ВВ	9	НС	2	НН	1	18

Продовження табл. Д.1

1	2	3	4	5	6	7	8	9	10	11
17	Відсутність ліцензій на сторонні бібліотеки	СС	5	ВН	7	НВ	3	НН	1	21
18	Претензії від користувачів через некоректні поради	НС	2	ВС	8	СН	4	СС	5	32
19	Перевищення бюджету на хмарні сервіси	СН	4	ВВ	9	СВ	6	НН	1	54
20	Зростання вартості платних API (Google, SMS)	НС	2	ВН	7	СВ	6	СС	5	42
21	Затримка фінансування наступного етапу	ВВ	9	ВС	8	СН	4	НС	2	32
22	Саботаж персоналу клінік	СВ	6	ВН	7	ВН	7	СС	5	49
23	Відмова клінік від партнерства	СВ	6	ВВ	9	СС	5	НВ	3	45
24	Складний інтерфейс для старшої аудиторії	НВ	3	СС	5	ВН	7	ВС	8	35
25	Низька активність користувачів на старті	НВ	3	ВС	8	СВ	6	СС	5	48

Закінчення табл. Д.1

1	2	3	4	5	6	7	8	9	10	11
26	Негативні відгуки через баги MVP	СН	4	ВН	7	СВ	6	СС	5	42
27	Відключення електроенергії (блекаути)	ВС	8	СВ	6	ВВ	9	ВС	8	54
28	Втрата інтернет-зв'язку у команди	ВН	7	СВ	6	СВ	6	НВ	3	36
29	Безпекові ризики (війна)	ВВ	9	ВВ	9	ВВ	9	ВВ	9	81
30	Різке коливання валютного курсу	НС	2	ВН	7	ВН	7	СС	5	49

Додаток Е

Таблиця Е.1

Розробка протиризикових заходів проєкту

№	Ризикова подія	ПРЗ 1	Симптом (рання ознака)	ПРЗ 2	ПРЗ 3
		Профілактика		При симптомі	При проблемі
1	2	3	4	5	6
1	Несумісність із застарілим ПЗ ветклінік	Розробка веб-кабінету адміністратора, що не потребує інтеграції.	Клініка повідомляє про використання паперових журналів або Excel.	Навчання персоналу роботі через браузер/планшет.	Надання планшета з ПЗ в оренду для ручного введення.
2	Критичні помилки (баги) в коді мобільного додатку	Автотести та перевірка коду.	Зростання кількості баг-репортів за спринт.	«Code Freeze» (зупинка розробки нових функцій), фокус на багах.	Випуск термінового хотфіксу (Hotfix), відкат релізу.
3	Проблеми інтеграції з Google Maps API	Вивчення документації API, встановлення лімітів запитів.	Помилки геокодування, затримка відображення карти.	Перехід на кешовані дані карт.	Тимчасова заміна на OpenStreetMap, звернення в підтримку.
4	Низька продуктивність додатку на старих смартфонах	Оптимізація зображень та коду.	Скарги тестувальників на «лаги» при скролінгу.	Відключення важких анімацій.	Випуск полегшеної «Lite» версії інтерфейсу.
5	Вразливість даних тварин та власників (кіберзагрози)	Шифрування БД, HTTPS, двофакторна аутентифікація.	Аномальний трафік, спроби перебору паролів.	Блокування IP-адрес, примусове скидання сесій.	Відновлення з чистого бекапу, повідомлення користувачів.

Продовження табл. Е.1

1	2	3	4	5	6
6	Збій системи Push-сповіщень (нагадування про візит)	Використання надійного провайдера.	Логи помилок доставки, скарги «не прийшло нагадування».	Повторна відправка черги повідомлень.	Відправка дублюючих SMS через резервний шлюз.
7	Конфлікт синхронізації даних	Налаштування автозбереження даних.	Поява різних даних на різних пристроях.	Повідомлення користувачу про помилку.	Завантаження актуальних даних із сервера.
8	Розповзання меж проєкту	Чітке ТЗ, фіксація обсягу робіт на старті спринту.	Поява нових «термінових» ідей під час розробки.	Перенесення нових задач у Беклог (на майбутнє).	Перегляд термінів релізу, відмова від другорядних функцій.
9	Нечітко сформульовані вимоги до функціоналу	Деталізація User Stories, уточнення беклогу з командою.	Велика кількість питань від розробників під час спринту.	Проведення воркшопу для уточнення деталей.	Зупинка розробки модуля до повного затвердження макетів.
10	Втрата ключового розробника	Парне програмування, документація коду.	Зниження активності, чутки про звільнення.	Розмова 1-на-1, перегляд умов праці/бонусів.	Терміновий найм аутстаф-фахівця, перерозподіл задач.
11	Низька мотивація команди через перепрацювання	Дотримання work-life balance, відсутність овертаймів.	Падіння швидкості, мовчання на дейлі-мітингах.	Тімбілдінг, надання додаткового вихідного.	Ротація кадрів, залучення нових мотивованих учасників.
12	Відсутність технічної документації	Обов'язкові коментарі в коді.	Новий програміст довго розбирається в задачах.	Зупинка розробки для написання інструкцій.	Створення опису для найважливіших частин коду.

Продовження табл. Е.1

1	2	3	4	5	6
13	Відмова в публікації App Store/Google Play	Вивчення правил магазинів до початку роботи.	Статус «Відхилено» в кабінеті розробника.	Листування з підтримкою магазину.	Виправлення помилок, на які вказав магазин.
14	Збій хмарного хостингу (AWS)	Розгортання в декількох зонах доступності.	Збільшення часу відгуку сервера.	Перемикання на резервний регіон/ сервер.	Відновлення з бекапу на іншому хостингу.
15	Збій платіжного шлюзу під час оплати послуг	Підключення резервної платіжної системи.	Зростання кількості помилок транзакцій.	Переключення на резервний шлюз.	Обробка платежів вручну через виставлення рахунків (IBAN).
16	Порушення законодавства про захист даних (GDPR)	Збір лише необхідних даних, юридичний аудит.	Запити користувачів на видалення даних.	Автоматизація вивантаження/видалення даних.	Юридичний захист, сплата штрафів, виправлення процесів.
17	Відсутність ліцензій на сторонні бібліотеки	Використання Open Source з перевіреними ліцензіями.	Попередження від аудиторів коду.	Заміна бібліотеки на безкоштовний аналог.	Купівля комерційної ліцензії.
18	Претензії від користувачів через некоректні поради	Додавання дисклеймера «Не є медичною порадою».	Негативні відгуки про «погане лікування».	Оновлення умов використання.	Юридичне врегулювання, повернення коштів.
19	Перевищення бюджету на хмарні сервіси	Моніторинг витрат (AWS Budgets).	Витрачено 80% бюджету, а зроблено 50% роботи.	Скорочення функціоналу MVP.	Пошук додаткового фінансування або заморозка проєкту.

Продовження табл. Е.1

1	2	3	4	5	6
20	Зростання вартості платних API (Google, SMS)	Оптимізація запитів, кешування відповідей.	Отримання рахунку з перевище-нням лімітів.	Обмеження кількості запитів на користувача.	Зміна провайдера послуг на дешевшого.
21	Затримка фінансування наступного етапу	Створення фінансового резерву на 1-2 місяці.	Повідомле-ння інвестора про затримку траншу.	Скорочення витрат на інфраструк-туру/марке-тинг.	Тимчасове призупинення робіт.
22	Саботаж персоналу клінік	Максималь-но простий UI, система нагадувань.	Пусті графіки лікарів, хоча клініка працює.	Дзвінок менеджера, введення бонусів за записи.	Ескалація до власника клініки, відключення онлайн-запису.
23	Відмова клінік від партнерства	Демонстра-ція вигоди, безкоштовн ий період.	Велика кількість відмов на етапі переговорів.	Зміна комерційної пропозиції, покращення умов.	Зміна цільової аудиторії.
24	Складний інтерфейс для старшої аудиторії	UX-тестування на віковій аудиторії.	Питання в підтримку: «Як записатися?»	Додавання відео-інструкцій та підказок.	Редизайн інтерфейсу на основі фідбеку.
25	Низька активність користувачів на старті	Маркетин-гова стратегія.	Мала кількість реєстрацій у перший тиждень.	Запуск додаткової реклами, акції для нових клієнтів.	Зміна каналів просування.
26	Негативні відгуки через баги MVP	Бета-тестування перед публічним релізом.	Поява оцінок 1-2 зірки в сторах.	Швидкі відповіді на відгуки, обіцянка виправлень.	Випуск оновлення з виправленнями та бонусами для невдоволених.

Закінчення табл. Е.1

1	2	3	4	5	6
27	Відключення електроенергії (блекаути)	Закупівля EcoFlow/Starlink, локальне середовище.	Графіки відключень, втрата зв'язку.	Перехід на асинхронний графік, офлайн-робота.	Зсув дедлайнів, передача задач колегам зі світлом.
28	Втрата інтернет-зв'язку у команди	Дублювання каналів (оптика + мобільний).	Високий пінг, нестабільне з'єднання.	Робота через мобільний хотспот.	Робота в офлайн-режимі з подальшою синхронізацією
29	Безпекові ризики (війна)	Дистанційна робота, сервери за кордоном.	Оголошення повітряної тривоги.	Перехід в укриття, зупинка нарад.	Призупинення проєкту, евакуація команди.
30	Різке коливання валютного курсу	Фіксація курсу в договорах, валютні резерви.	Різка зміна курсу НБУ.	Перегляд бюджету, скорочення витрат.	Підняття цін на послуги сервісу.

Додаток Ж

Лістинг SQL-скрипта для створення бази даних мобільного додатку «PetHealth»:

```
-- 1. Створення таблиці Власників (Користувачів)
CREATE TABLE Owner (
    owner_id SERIAL PRIMARY KEY,
    first_name VARCHAR(50) NOT NULL,
    last_name VARCHAR(50) NOT NULL,
    email VARCHAR(100) UNIQUE NOT NULL,
    phone_number VARCHAR(20) UNIQUE NOT NULL,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- 2. Створення таблиці Клінік
CREATE TABLE Clinic (
    clinic_id SERIAL PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    address TEXT NOT NULL,
    phone_number VARCHAR(20),
    email VARCHAR(100),
    working_hours VARCHAR(50),
    rating DECIMAL(3, 2) DEFAULT 0.00
);

-- 3. Створення таблиці Лікарів
CREATE TABLE Veterinarian (
    doctor_id SERIAL PRIMARY KEY,
    clinic_id INT NOT NULL,
    full_name VARCHAR(100) NOT NULL,
    specialization VARCHAR(50),
    experience_years INT,
    rating DECIMAL(3, 2) DEFAULT 0.00,
    photo_url TEXT,
    FOREIGN KEY (clinic_id) REFERENCES Clinic(clinic_id) ON DELETE
CASCADE
);

-- 4. Створення таблиці Послуг
CREATE TABLE Service (
    service_id SERIAL PRIMARY KEY,
    clinic_id INT NOT NULL,
    service_name VARCHAR(100) NOT NULL,
    description TEXT,
```

```
    price DECIMAL(10, 2) NOT NULL,
    duration_min INT DEFAULT 30,
    FOREIGN KEY (clinic_id) REFERENCES Clinic(clinic_id) ON DELETE
CASCADE
);

-- 5. Створення таблиці Тварин
CREATE TABLE Pet (
    pet_id SERIAL PRIMARY KEY,
    owner_id INT NOT NULL,
    name VARCHAR(50) NOT NULL,
    species VARCHAR(50) NOT NULL,
    breed VARCHAR(50),
    dob DATE,
    gender VARCHAR(10) CHECK (gender IN ('Male', 'Female', 'Other')),
    FOREIGN KEY (owner_id) REFERENCES Owner(owner_id) ON DELETE
CASCADE
);

-- 6. Створення таблиці Розкладу лікарів
CREATE TABLE Schedule (
    slot_id SERIAL PRIMARY KEY,
    doctor_id INT NOT NULL,
    start_time TIMESTAMP NOT NULL,
    end_time TIMESTAMP NOT NULL,
    is_booked BOOLEAN DEFAULT FALSE,
    FOREIGN KEY (doctor_id) REFERENCES Veterinarian(doctor_id) ON
DELETE CASCADE
);

-- 7. Створення таблиці Записів на прийом
CREATE TABLE Appointment (
    appointment_id SERIAL PRIMARY KEY,
    pet_id INT NOT NULL,
    doctor_id INT NOT NULL,
    service_id INT,
    appointment_date TIMESTAMP NOT NULL,
    status VARCHAR(20) DEFAULT 'Scheduled',
    medical_notes TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (pet_id) REFERENCES Pet(pet_id) ON DELETE
CASCADE,
    FOREIGN KEY (doctor_id) REFERENCES Veterinarian(doctor_id) ON
DELETE CASCADE,
    FOREIGN KEY (service_id) REFERENCES Service(service_id)
```

```
);

-- 8. Створення таблиці Оплат
CREATE TABLE Payment (
    payment_id SERIAL PRIMARY KEY,
    appointment_id INT UNIQUE NOT NULL,
    amount DECIMAL(10, 2) NOT NULL,
    payment_method VARCHAR(20),
    status VARCHAR(20) DEFAULT 'Pending',
    transaction_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (appointment_id) REFERENCES
Appointment(appointment_id)
);

-- 9. Створення таблиці Відгуків
CREATE TABLE Review (
    review_id SERIAL PRIMARY KEY,
    appointment_id INT UNIQUE NOT NULL,
    rating INT CHECK (rating >= 1 AND rating <= 5),
    comment TEXT,
    is_moderated BOOLEAN DEFAULT FALSE,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (appointment_id) REFERENCES
Appointment(appointment_id)
);
```