

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Автоматизації і інформаційних технологій
(факультет)

Управління проектами
(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»
на тему:**

«Управління розробкою ІТ-продукту: застосунок "Pet"»

Беренок Максим Віталійович

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

Автоматизації і інформаційних технологій
(факультет)

Управління проектами
(кафедра)

ЗАТВЕРДЖУЮ
Завідувач кафедри УП
Сергій БУШУЄВ

„___” _____ 2024 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

на тему:

Управління розробкою ІТ-продукту: застосунок "Pet"

Виконав: студент 4-го курсу
групи ІСТ-УП-20
Спеціальність: 126 Інформаційні
системи та технології
ОПП Управління проектами

Максим БЕРЕНОК

Керівник Олександр ВОЙТЕНКО

Рецензент _____

Київ 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: управління проектами

Освітній рівень: бакалавр

Спеціальність: 126 Інформаційні системи та технології

Освітня програма: Управління проектами

ЗАТВЕРДЖУЮ
Завідувач кафедри УП
Сергій БУШУЄВ

„___” _____ 2024 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВР**

Беренок Максим Віталійович

1. Тема роботи: Управління розробкою ІТ-продукту: застосунок "Pet"
затверджена наказом ректора № 399/2 від ____ 2024 р.
2. Керівник роботи: Войтенко Олександр Степанович, канд. техн. наук, доцент,
доцент кафедри управління проектами КНУБА.
3. Термін подання студентом роботи до захисту: ____ 2024 р.
4. Зміст пояснювальної записки за розділами:
 - Р.1. Аналіз предметної області та постановка завдання.
 - Р.2. Розробка інформаційного забезпечення.
 - Р.3. Розробка програмного забезпечення.
 - Р.4. Управління проектом створення додатку.
5. Інформаційні слайди:
 - С.1. Цілі та задачі.
 - С.2. Організаційна структура системи.
 - С.3. Архітектура програми.
 - С.4. Концептуальна модель БД.
 - С.5. Даталогічна модель бази даних.
 - С.6. Структура підсистем програми.

6. Календарний план виконання кваліфікаційної роботи

Види робіт та їх зміст	Дата виконання
Р.1. Аналіз предметної області та постановка задачі	___ ___ 2023
Р.2. Розробка інформаційного забезпечення	___ ___ 2023
Р.3. Розробка програмного забезпечення	___ ___ 2023
Р.4. Управління проектом створення додатку	___ ___ 2023
Остаточне оформлення роботи	___ ___ 2023
Попередній захист роботи на кафедрі	___ ___ 2023

7. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис
Управління створення проектом	д.т.н. проф. Терентьев О.О.		
Прийом програмного продукту	к.т.н., доц. Бойко Є.Г.		

8. Дата видачі завдання: 15 лютого 2024 року

Керівник	_____	Олександр ВОЙТЕНКО
	(підпис)	
Студент	_____	Максим БЕРЕНОК
	(підпис)	

АНОТАЦІЯ

Беренок М.В. Управління розробкою ІТ-продукту: застосунок "Pet".

Кваліфікаційна випускна робота бакалавра за спеціальністю: 126 «Інформаційні системи і технології», освітня програма «Управління проектами». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Дипломна робота присвячена розробці медичної інформаційної системи для ветеринарних клінік та медичного порталу для власників домашніх тварин в Україні. У роботі розглядаються питання автоматизації процесів медичного обслуговування домашніх тварин, що включає облік клієнтів, планування та облік візитів, доступ до медичних записів, фінансовий облік та генерацію звітів.

Основною метою розробки є створення зручного та інтуїтивно зрозумілого програмного продукту, який забезпечить високу ефективність роботи та конфіденційність даних. У процесі розробки додатку використовуються сучасні технології, зокрема веб-технології, фреймворки та бази даних.

Методи дослідження включають аналіз предметної області, порівняльний аналіз існуючих рішень та методів управління проектами, а також використання інструментальних засобів для реалізації програмного забезпечення.

Результатом роботи є розроблена інформаційна модель автоматизованої системи у вигляді програми, яка забезпечить ефективну взаємодію між ветеринарними клініками та власниками домашніх тварин, сприятиме підвищенню якості медичного обслуговування та задоволеності клієнтів.

Ключові слова: медична інформаційна система, ветеринарна клініка, автоматизація, управління проектами, інформаційні технології.

SUMMARY

Berenok M.V. IT product development management: "Pet" application.

Bachelor's thesis in specialty: 126 “Information Systems and Technologies”, educational program “Project Management.” – Kyiv National University of Construction and Architecture - Kyiv, 2024.

The thesis is dedicated to the development of a medical information system for veterinary clinics and a medical portal for pet owners in Ukraine. The work addresses the automation of pet healthcare processes, including client accounting, scheduling and recording of visits, access to medical records, financial accounting, and report generation.

The main goal of the development is to create a user-friendly and intuitive interface that will ensure high work efficiency and data confidentiality. In the process, modern technologies are used, including web technologies, frameworks, and databases.

Research methods include subject area analysis, comparative analysis of existing solutions and project management methodologies, and the use of software implementation tools.

The result of the work is a developed information model of an automated program that will ensure effective interaction between veterinary clinics and pet owners, improve the quality of medical care and customer satisfaction.

Keywords: medical information system, veterinary clinic, automation, project management, information technology.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП.....	10
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ	12
1.1. Загальний аналіз предметної області щодо розробки програмного застосунку	12
1.2. Аналіз рішень та існуючих сервісів та їх функцій щодо надання медичного супроводу для домашніх тварин та їх власників.....	14
1.3. Вибір структурних і функціональних особливостей застосунку, що розробляється.....	28
Висновки до розділу 1	30
2. АНАЛІЗ І ВИБІР МЕТОДУ УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ ЗАСТОСУНКУ.....	31
2.1. Застосування методології управління проектами в галузі інформаційних систем та технологій	31
2.2. Класичний підхід до розробки програмних продуктів	32
2.3. Застосування гнучкого підходу в розробці ІТ-рішень	34
2.3.1. Scrum	35
2.3.2. Kanban	37
2.3.3. Змішані моделі, DSDM.....	39
2.3.4. Масштабовані моделі Scrum.....	41
Висновки до розділу 2	43
3. ПРОЕКТНІ ТА ТЕХНІЧНІ РІШЕННЯ ЗАСТОСУНКУ РЕТ.....	45
3.1 Концептуальний дизайн	45
3.1.1 Діаграма EER.....	45
3.1.2 Взаємозв'язки між сутностями.....	46

3.2	Реляційна схема.....	48
3.2.1	Формат даних для кожного відношення.....	49
3.3	Нормалізація	52
3.4	Оператори SQL.....	54
3.4.1	Створення бази даних за допомогою операторів SQL.....	54
3.4.1.1	Створення таблиці	54
3.4.1.2	Стан бази даних.....	55
3.5	Пропонований GUI	56
	Висновки до розділу 3	61
4. ІНФОРМАЦІЙНА МОДЕЛЬ ПРОЕКТУ РОЗРОБКИ ЗАСТОСУНКУ		
РЕТ	62
4.1.	План управління проектом.....	62
4.2.	Управління розподілом ресурсів	71
4.3.	Управління вартістю проекту	76
4.4.	Управління комунікаціями в проекті.....	80
4.5.	Управління зацікавленими сторонами проекту.....	82
	Висновки до розділу 4	86
ЗАГАЛЬНІ ВИСНОВКИ.....		87
СПИСОК ДЖЕРЕЛ		88

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ

МІС – медична інформаційна система

ПЗ – програмне забезпечення

СУ – системи управління

СУБД – системи управління базами даних

СУЛ – система управління лікарнею

AWS - Amazon Web Services

GCP - Google Cloud Platform

RBS – Resource breakdown structure

ВСТУП

На сучасному ринку програмного забезпечення для потреб медичних закладів, що надають послуги лікування домашніх тварин, розробка відповідних сервісів або програмних продуктів стає все більш актуальною та важливою. Постійне зростання клієнтів та підвищення вимог до їхнього медичного обслуговування вимагає наявності ефективних програмних засобів, які забезпечують якісне та оперативне обслуговування, навіть в умовах турбулентного зовнішнього середовища. З огляду на сформовані тенденції розвитку надання медичних послуг з обслуговування такого роду клієнтів постає необхідність розробки відповідного програмного продукту.

Мета кваліфікаційної роботи – опис та застосування відповідних методів щодо розробки програмного застосунку "Pet" та управління проектом його розробки.

Об'єкт дослідження кваліфікаційної роботи – інформаційна система, спрямована на забезпечення потреб закладів охорони здоров'я домашніх тварин та медичний портал для їх власників в Україні.

Предметом дослідження кваліфікаційної роботи є процеси розробки та інформаційна модель предметної області розроблюваної інформаційної системи, а також її програмний інтерфейс та організація робіт з її розробки – управління проектом.

Основними **задачами дослідження** є:

- Проведення аналізу предметної області застосування програмного додатку.
- Проведення аналізу сучасних рішень та сервісів, що надають відповідні послуги з обслуговування клієнтів медичних інформаційних систем.
- Вибір структурних та функціональних елементів додатку, що розробляється
- Аналіз та вибір методів управління проектами задля ефективної

реалізації проєкту розробки програмного додатку

- Розробка інформаційної моделі проєкту розробки програмного додатку

Pet

Методи дослідження – у роботі використовуються сучасні методи планування, прогнозу, аналізу та контролю.

Актуальність даної роботи полягає в тому, що розробка даного програмного продукту дасть змогу зайняти нішу на ринку з надання послуг медичним закладам та клієнтам, які мають домашніх улюбленців, що дозволить компаніям, які будуть використовувати цей додаток отримати конкурентну перевагу та покращувати задоволеність та лояльність клієнтів.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ЗАСТОСУВАННЯ

1.1. Загальний аналіз предметної області щодо розробки програмного застосунку

На сучасному ринку України розробка програмного забезпечення (далі – ПЗ) для медичних потреб домашніх тварин стає все більш актуальною та важливою. Постійне зростання кількості домашніх тварин та підвищення вимог до їхнього медичного обслуговування вимагає наявності ефективних інформаційних систем, які забезпечують якісне та оперативне обслуговування.

Управління розробкою програмного застосунку "Pet" виходить за рамки простої автоматизації процесів і має на меті створення повноцінної медичної інформаційної системи (далі – МІС) для закладів ветеринарної медицини та медичного порталу для власників домашніх тварин в Україні.

Спрощене управління клієнтською базою та наданням медичних послуг для тварин вимагає зручного та інтуїтивно зрозумілого програмного інтерфейсу, а також високого рівня безпеки та конфіденційності даних. Ця система має включати в себе широкий функціонал, який об'єднує в собі облік клієнтів, планування та облік візитів, доступ до інформації з обмеженим рівнем доступу залежно від ролі працівника, а також можливість пошуку клієнтів та перегляду їхніх медичних записів.

Розробка даного застосунку передбачає використання сучасних технологій, зокрема, Web-технологій, фреймворків та баз даних для забезпечення надійної та ефективної роботи. Крім того, реалізація програми передбачає використання елементів сучасного графічного інтерфейсу для зручності роботи користувачів та оптимізації їхнього робочого процесу.

Об'єктом дослідження для проекту "Управління розробкою ІТ-продукту: застосунок "Pet" є медична інформаційна система, спрямована на забезпечення потреб закладів охорони здоров'я домашніх тварин та медичний портал для їх власників в Україні.

В рамках дослідження передбачено визначення основних інформаційних об'єктів, характеристик, встановлення зв'язків між ними, а також виявлення джерел надходження даних та обмежень, які можуть бути застосовані до цих об'єктів. Предметом дослідження є основні або базові процеси а також інтерфейс користувача та взаємодія з даними, необхідними для забезпечення функціональності програмного проекту. Також розглядається складна структура бази даних, яка має відповідати вимогам, встановленим для цього проекту.

Методи дослідження включають вивчення та узагальнення матеріалів з предметної області, порівняльний аналіз існуючих технологічних та проектних підходів, а також використання інструментальних засобів для реалізації програмного проекту.

Результатом дослідження є розроблена інформаційна модель автоматизованої системи у вигляді програми, яка буде використовуватися для інформування клієнтів клініки та надання переліку послуг. Ця модель базується на використанні сучасного графічного інтерфейсу та відповідає підходам та методам управління, що використовуються в ІТ-проектах.

Для організації, що спеціалізується на наданні послуг для домашніх тварин, ключовими вимогами до програмного забезпечення є простота, швидкість та зручність в користуванні. Наявність інтуїтивно зрозумілого інтерфейсу є надзвичайно важливою, оскільки це забезпечить зв'язок з клієнтами та оптимізує процес надання послуг.

Також важливо забезпечити широкий функціонал для різних категорій користувачів, включаючи як звичайних клієнтів, так і адміністраторів. Задачею розробки є створення програмного продукту, який здатний виконувати різноманітні функції та забезпечити потреби клієнтів у наданні медичних послуг для їх домашніх улюбленців.

Перед усім потрібно створити невеликий зріз по основним функціям, які може включати в себе мій програмний продукт "Pet".

Для реалізації програмного продукту можна виділити такі необхідні функції як:

- Облік клієнтів та їх даних.

- Планування та облік візитів.
- Пошук клієнтів за прізвищем, кличкою домашньої тварини або номером картки.
- Перегляд попередніх записів у картці клієнта.
- Забезпечення можливості роздруковувати дані з бази та результати запитів у зручному вигляді.
- Надання переліку послуг та інші функції, що можуть забезпечити комфортне обслуговування клієнтів.

Розглядаючи аналіз предметної області, можна зробити висновок, що завдання, що стосується автоматизації, стосується стандартних аналітичних і облікових процесів. Це завдання можна успішно вирішити за допомогою програмного забезпечення та інструментів, що надаються технічним завданням, зокрема елементів файлових баз даних та підходів до управління проектами.

Цей аналіз допоможе розкрити контекст і потреби перед реалізацією проекту "Pet" в галузі здоров'я тварин. Детальне вивчення цих аспектів дозволить не тільки визначити напрямки подальшого розвитку, але і забезпечить максимальну відповідність проекту реальним потребам і можливостям цільової аудиторії.

1.2. Аналіз рішень та існуючих сервісів та їх функцій щодо надання медичного супроводу для домашніх тварин та їх власників.

На основі попереднього аналізу, приведеному в розділі 1.1. очевидно, що прогрес інформаційних технологій викликає значні перетворення в різних областях, включаючи ветеринарну медицину, що є предметом дослідження в цій дипломній роботі. Віртуальні інструменти та програмні рішення стають незамінними у повсякденній роботі ветеринарів та власників домашніх тварин. Ці технології підвищують доступність охорони здоров'я, забезпечують простоту

та ефективність у управлінні медичними даними, а також сприяють координації клінічних зустрічей.

У цій частині буде детально розглянуто два широко використовувані програмні застосунки, які допомагають ветеринарним клінікам автоматично та ефективно надавати медичний супровід для домашніх тварин та їх власників.

Сервіси, які будуть порівняні, є провідними постачальниками ПЗ у сфері ветеринарної допомоги. Вони мають добре вибудовану та міцну структуру в МІС. Кожне з цих програмних забезпечень мають свої унікальні характеристики та функціональні можливості, які можуть відповідати різним потребам ветеринарних клінік та власників домашніх улюбленців. Ось деякі з них:

- IDEXX Cornerstone
- eVetPractice

Дані програмні рішення допоможуть у проведенні глибокого порівняльного аналізу цих сервісів, зосереджуючись на таких аспектах:

- Функціональні можливості
- Інтерфейс користувача
- Використані технології
- Зручність використання

Цей аналіз допоможе зрозуміти переваги та недоліки кожного сервісу та визначити найкращий вибір характеристик та функціоналу, які будуть використовуватися у застосунку "Pet" та покращать надання медичної допомоги для домашніх тварин та їх власників.

IDEXX Cornerstone – це програмне забезпечення, розроблене спеціально для ветеринарних клінік для управління їхньою діяльністю, покращення комунікації з власниками домашніх тварин та підвищення якості догляду за тваринами. Він функціонує як цифровий хаб для всього, що пов'язано з управлінням ветеринарною клінікою.

Аналізуючи цей сервіс було виділено основні функціональні можливості цього сервісу:

1. *Облік тварин та клієнтів.*

Cornerstone дозволяє створювати та зберігати профілі клієнтів і їх тварин, включаючи базову інформацію про власника, контактні дані, медичну історію тварини та інші важливі дані.

2. Планування візитів до ветеринара.

Система дозволяє планувати та реєструвати візити клієнтів до клініки за допомогою календаря або планера, що спрощує організацію робочого часу та управління графіком прийому.

3. Медична книга.

Cornerstone дозволяє ветеринарам вести детальні медичні записи та створювати історію лікування для кожної тварини, включаючи відомості про проведені процедури, призначення, результати обстежень тощо.

4. Обробка платежів та фінансовий облік.

Система дозволяє здійснювати обробку платежів від клієнтів за надані послуги, виставляти рахунки, вести облік фінансових транзакцій та генерувати фінансові звіти.

5. Звіти та аналітика.

Cornerstone надає можливість генерувати різноманітні звіти та аналітичні дані про роботу клініки, включаючи фінансові звіти, звіти про навантаження, статистику про клієнтів та тварин, аналіз ефективності роботи та інше.

6. Інтеграція з іншими системами.

Система може бути інтегрована з іншими програмними рішеннями, такими як системи лабораторної діагностики, медичні апарати, платіжні шлюзи тощо.

Користуючись програмою та звернувши увагу на інтерфейс користувача можна сказати, що Cornerstone має на меті забезпечити простоту, зручність та ефективність в роботі для ветеринарів та персоналу клініки. Основні елементи інтерфейсу включають:

Головне меню. Меню розташоване у верхній частині екрана і містить основні розділи програми, такі як "File", "Edit", "Activities", "Lists" тощо.

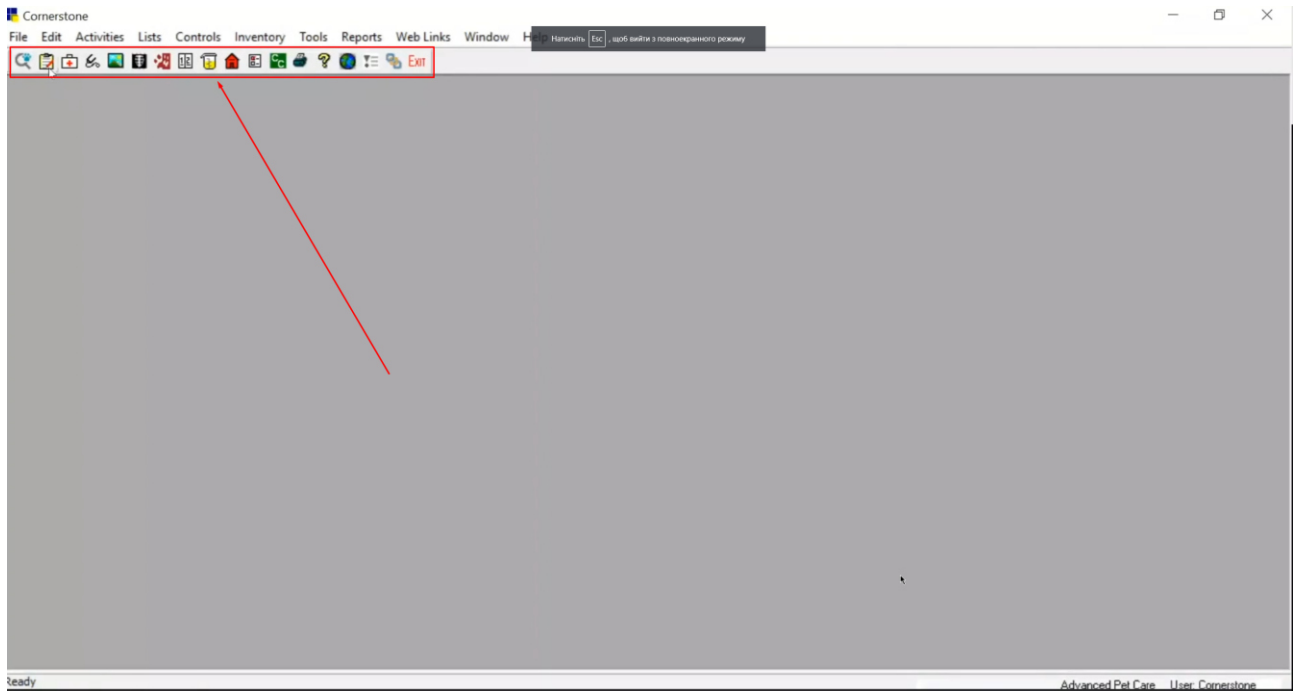


Рис. 1.2.1 Головне меню в IDEXX Cornerstone

Панель навігації. Також розташована у верхній частині екрана і містить швидкий доступ до основних розділів або функцій програми такі як "Patient Clipboard", "Search Clients/Patient", "Check-in", "Appointment Scheduler" тощо.

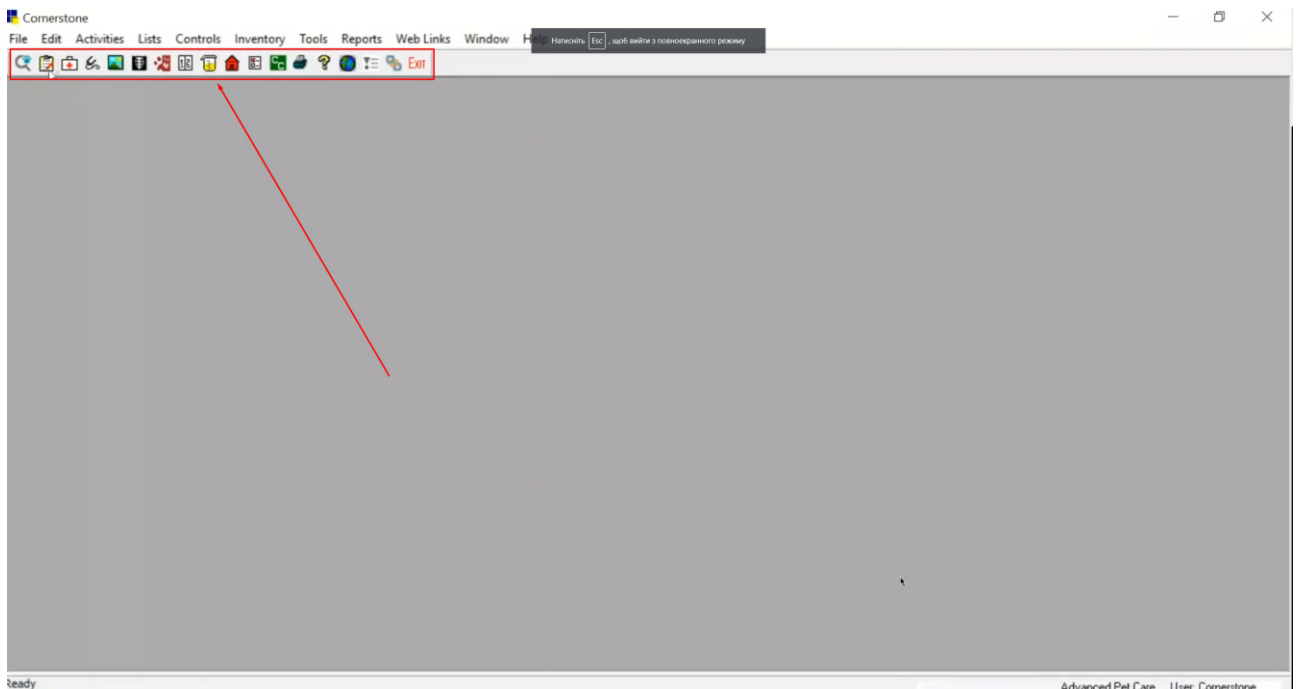


Рис. 1.2.2 Панель навігації в IDEXX Cornerstone

Карта пацієнта. Дозволяє переглядати та редагувати інформацію про клієнтів та їх тварин, включаючи персональні дані, контактну інформацію, історію лікування тощо.

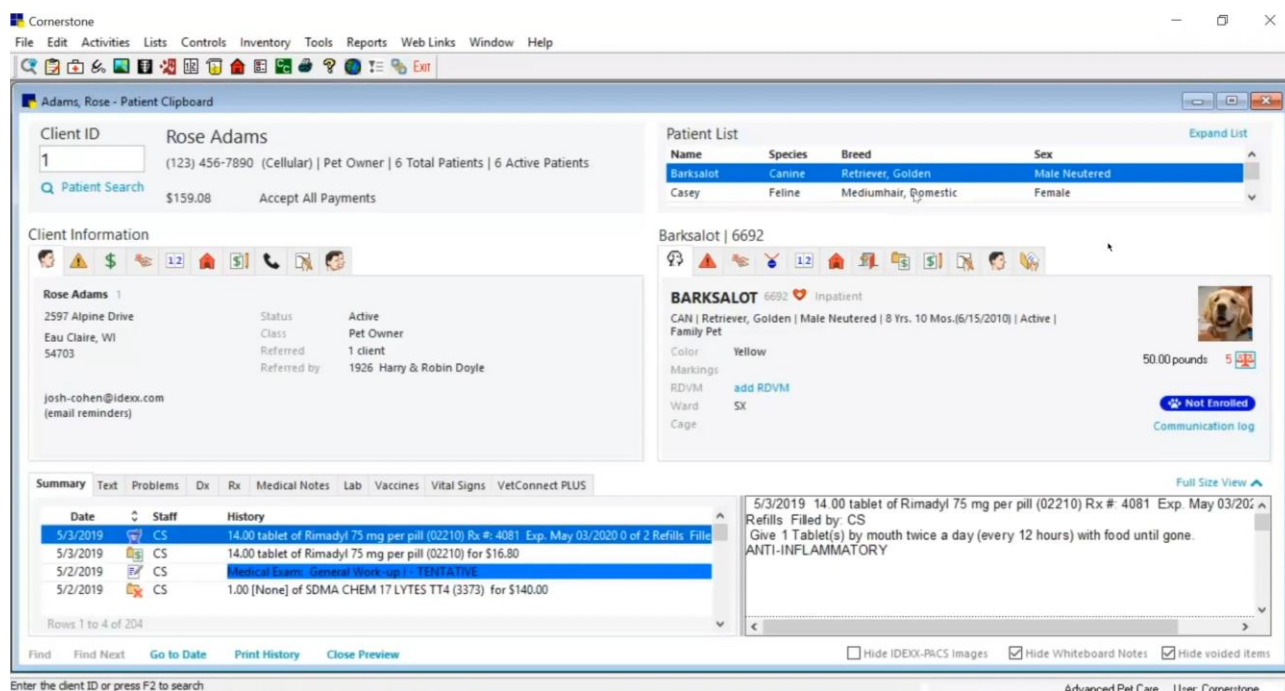


Рис. 1.2.3 Карта пацієнта в IDEXX Cornerstone

Календар. Показує графік прийому клієнтів та планування майбутніх візитів. Можна побачити, що записи відрізняються кольором це означає, що записи мають різний статус та попередження.

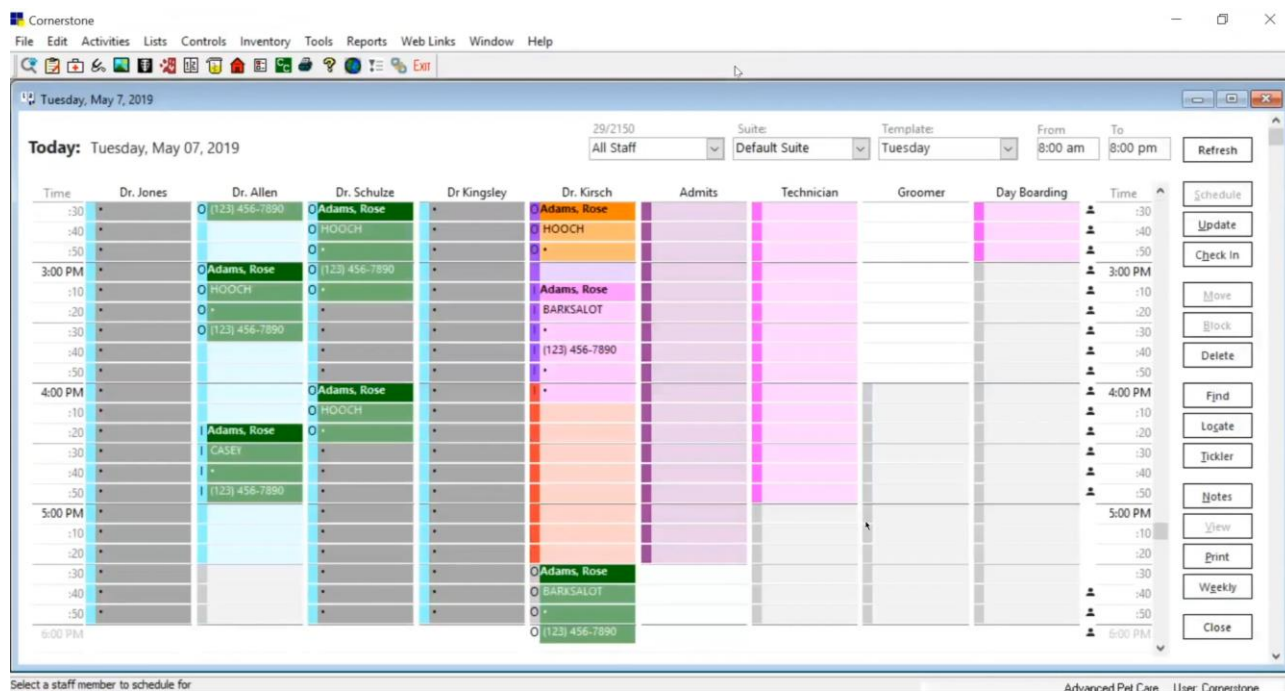


Рис. 1.2.4 Календар в IDEXX Cornerstone

Медичні записи пацієнта. Дозволяють переглядати та редагувати історію медичного лікування тварин, включаючи проведені процедури, результати обстежень, призначення тощо.

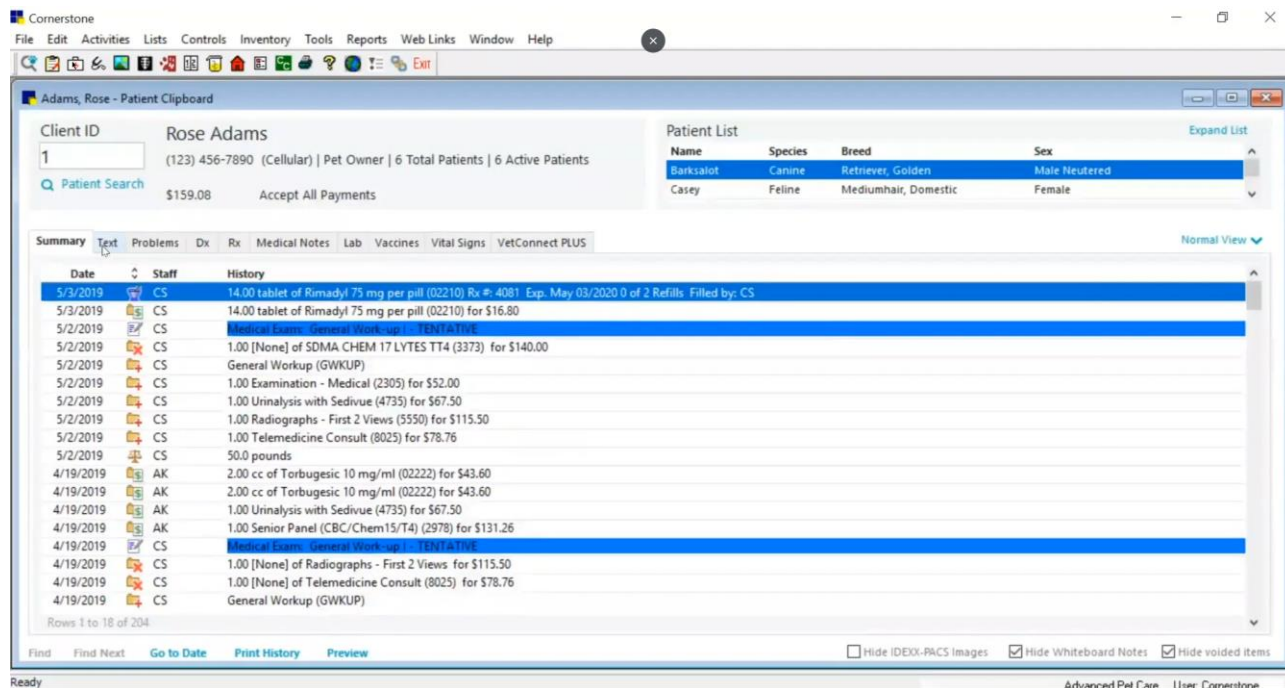


Рис. 1.2.5 Медичні записи пацієнта в IDEXX Cornerstone

Форми та документи. Включають форми для створення нових медичних записів, рецептів, медичних карток, рахунків тощо.

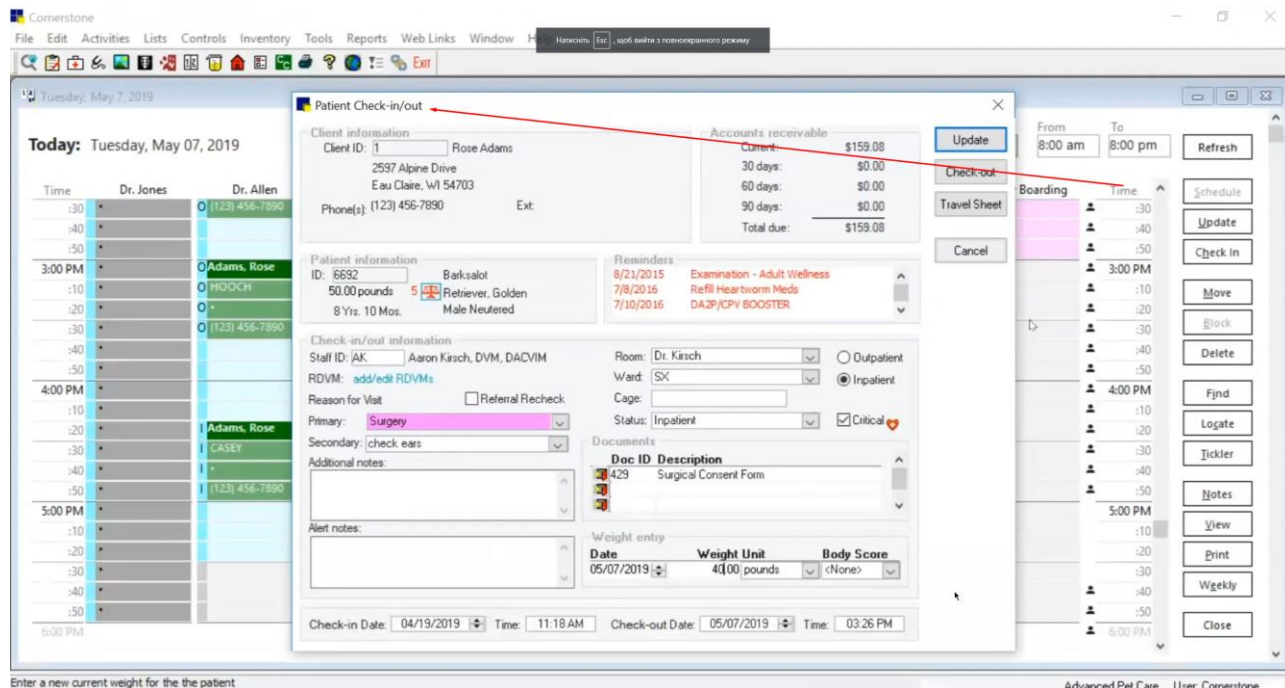


Рис. 1.2.6 Форми та документи в IDEXX Cornerstone

Інші функції. Залежно від налаштувань та ролі користувача, інтерфейс може містити інші корисні елементи, такі як фінансові звіти, аналітика, інструменти адміністрування тощо.

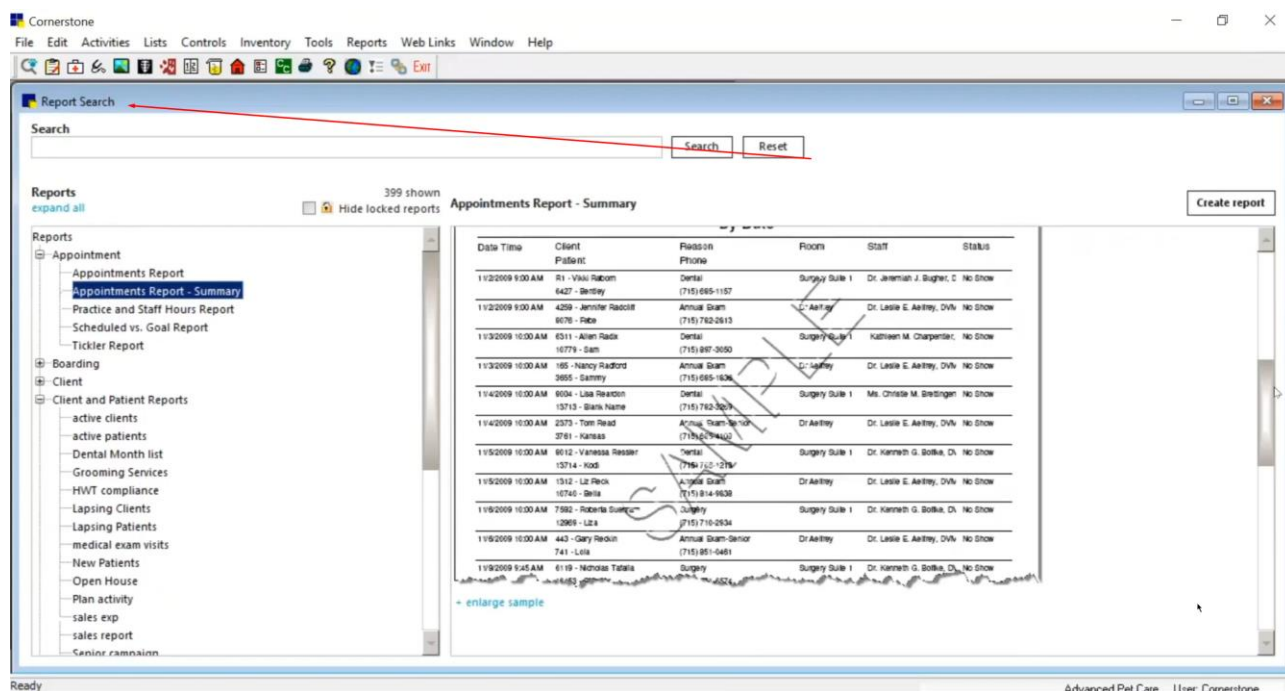


Рис. 1.2.7 Звіти в IDEXX Cornerstone

Дізнатися, які саме технології та платформи були використані в цьому сервісі доволі складно, бо не можуть бути публічно розголошені, можна зробити деякі обґрунтовані припущення, засновані на галузевих стандартах і тенденціях:

- **База даних:** IDEXX Cornerstone використовує реляційну систему управління базами даних для зберігання та управління даними. Найпоширенішими системами управління базами даних (далі —СУБД) для корпоративних додатків є Microsoft SQL Server, Oracle Database або PostgreSQL.
- **Веб-технології:** Враховуючи необхідність веб-доступу та функціональності, IDEXX Cornerstone, ймовірно, використовує веб-технології, такі як HTML, CSS та JavaScript для свого користувацького інтерфейсу. Крім того, він може використовувати веб-фреймворки та бібліотеки, такі як Angular, React або Vue.js для створення інтерактивних веб-додатків.
- **Інтеграції:** ПЗ може включати інтеграційні технології та протоколи для зв'язку із зовнішніми системами та сервісами. Це може включати

використання API (інтерфейсів прикладного програмування) для взаємодії зі службами лабораторних досліджень, системами візуалізації або платформами електронних медичних записів.

- **Платформи розгортання:** IDEXX Cornerstone можна розгорнути локально або розмістити в хмарі. При локальному розгортанні він буде працювати на серверах. Хмарне розгортання передбачає використання постачальників хмарної інфраструктури, таких як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform (GCP).

Проаналізувавши функціональні можливості та інтерфейс користувача, можна виділити такі аспекти в зручність використання як:

1. **Інтуїтивний інтерфейс користувача:** Cornerstone має дружній та легко зрозумілий інтерфейс користувача. Меню, кнопки та інші елементи керування організовані логічно та інтуїтивно, щоб користувачі швидко знаходили необхідні функції.

2. **Швидкість роботи:** Ефективна робота програми, швидка відповідь на команди користувача та мінімальний час очікування були помічені під час використання Cornerstone.

3. **Наявність документації та навчання:** На сайті наявна чітка та доступна документація, яка значно полегшила процес використання.

Наступним сервісом для проведення аналізу буде:

eVetPractice — це також один з провідних ПЗ для ветеринарних клінік, яке використовується по всьому світу. Завдяки своїй широкій функціональності та високому рівню надійності, eVetPractice допомагає ветеринарним клінікам підвищити ефективність своєї роботи та покращити якість надання медичних послуг тваринам.

Функціонал сервісу eVetPractice охоплює широкий спектр можливостей, спрямованих на полегшення ветеринарної практики та поліпшення обслуговування клієнтів. Основні функції включають:

1. **Облік тварин та клієнтів:** Система дозволяє створювати та зберігати профілі клієнтів і їх тварин, включаючи базову інформацію про тварину, історію медичного обслуговування, вакцинації, тестування та інше.

2. **Планування візитів:** Можливість створення та керування розкладом візитів клієнтів до клініки, включаючи можливість призначення часу, а також нагадування клієнтам про майбутні візити.

3. **Медичні записи:** Функціонал для ведення повних та докладних медичних записів про кожну тварину, включаючи діагнози, призначення, результати лабораторних тестів, рентгенівські знімки та інші медичні дані.

4. **Фінансовий облік:** Можливість створення та ведення фінансових записів про оплату послуг, створення рахунків для клієнтів, відстеження оплат та невиплачених рахунків.

5. **Звітність:** Функціонал для створення різноманітних звітів та аналізу даних, таких як звіти про фінансову діяльність, звіти про відвідуваність та інші.

6. **Комерційні можливості:** Деякі версії сервісу можуть включати можливості інтеграції з іншими платформами, такими як онлайн-магазини для продажу ветеринарних препаратів, аксесуарів для тварин та інше.

Інтерфейс користувача eVetPractice має дуже зручну та інтуїтивно зрозумілу структуру. Основний екран містить такі елементи:

Головне меню: при відкритті застосунку нас зустрічає розділ "Home page". В якій можна побачити актуальну інформацію та новини.

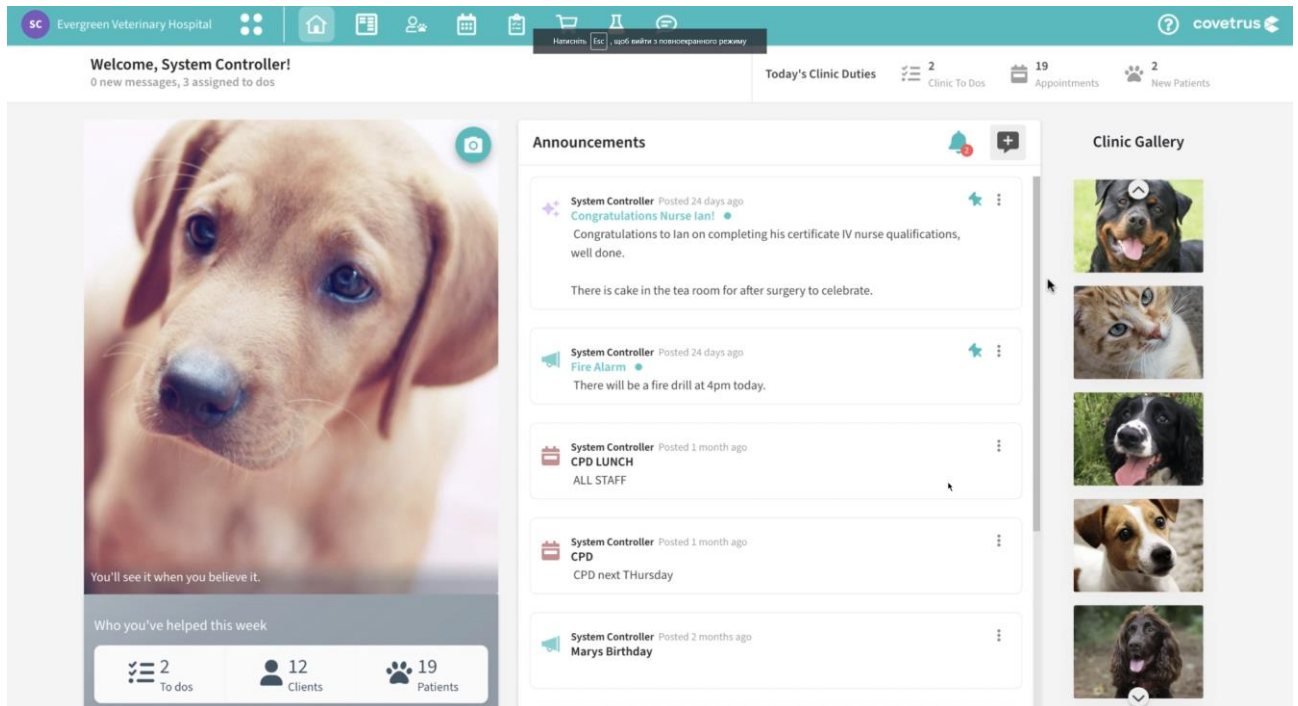


Рис. 1.2.8 Головне меню в eVetPractice

Панель навігації: знаходиться зверху екрану і містить розділи, такі як "Home page", "Whiteboard", "Task", "Clients", "Reports", "Labs" тощо. Це дозволяє користувачам швидко переходити до потрібного розділу.

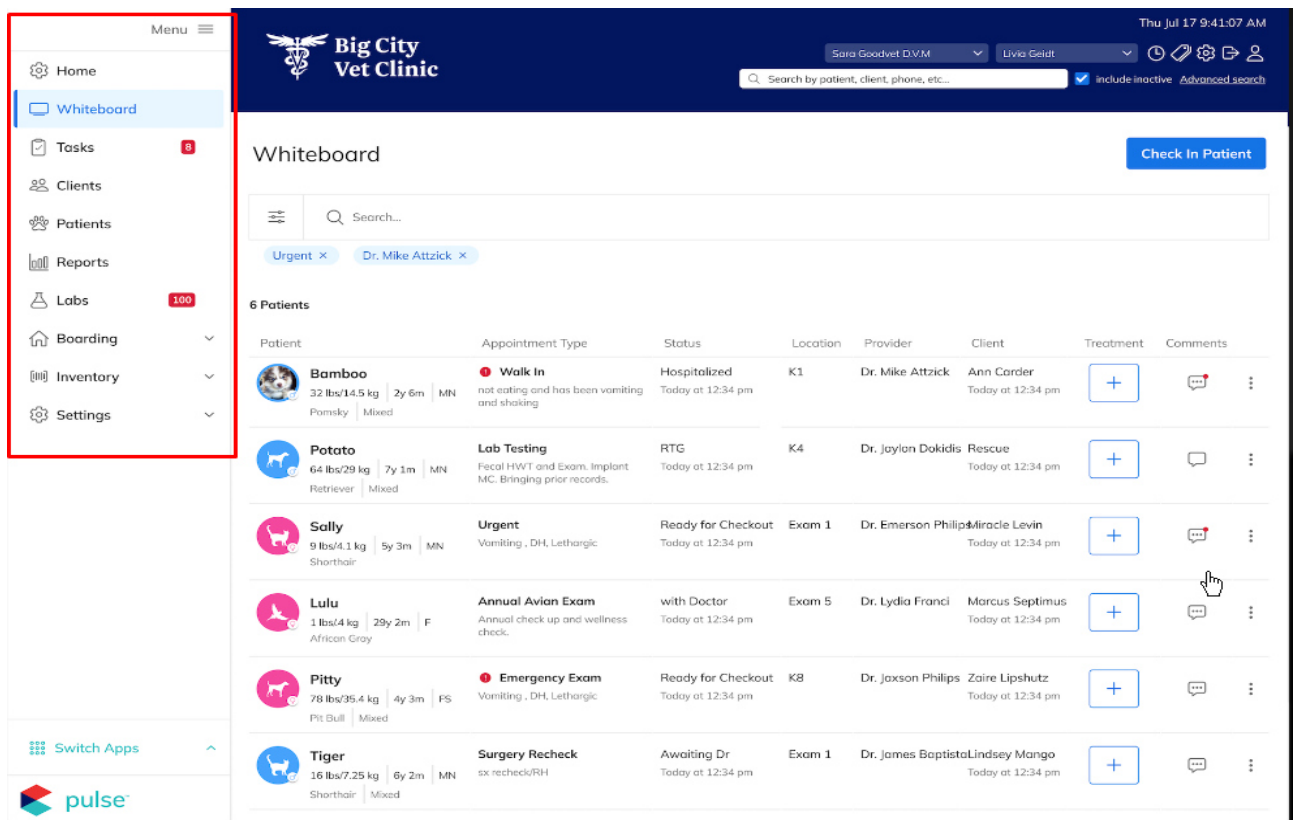


Рис. 1.2.9 Панель навігації в eVetPractice

Карта пацієнта: Дозволяє переглядати та редагувати інформацію про клієнтів та їх тварин, включаючи персональні дані, контактну інформацію, історію лікування тощо.

The screenshot shows the patient card for 'Bamboo' in the eVetPractice system. The card is divided into several sections:

- Header:** Big City Vet Clinic logo, search bar, and user information.
- Patient Information:** Name (Bamboo), ID (000021), status (Active), breed (Cavalier King Charles Spaniel / Poodle), and other details.
- Owner Information:** Allie Rogers, contact details, and company information.
- Visits:** A list of visits including Surgery (In Progress), Wellness Visit, Boarding, and Standard.
- Medical Record:** A section with tabs for Medical Record, Highlights, and Online Activity.
- Weight Trend:** A line graph showing weight changes over time.
- Core Vaccines:** A list of vaccines with their expiration dates and status (e.g., Overdue).
- Reminders:** A list of upcoming reminders for various procedures.

Рис. 1.2.10 Карта пацієнта в eVetPractice

Календар: показує графік прийому клієнтів та планування майбутніх візитів. Можна побачити, що записи відрізняються кольором це означає, що записи мають різний статус та попередження.

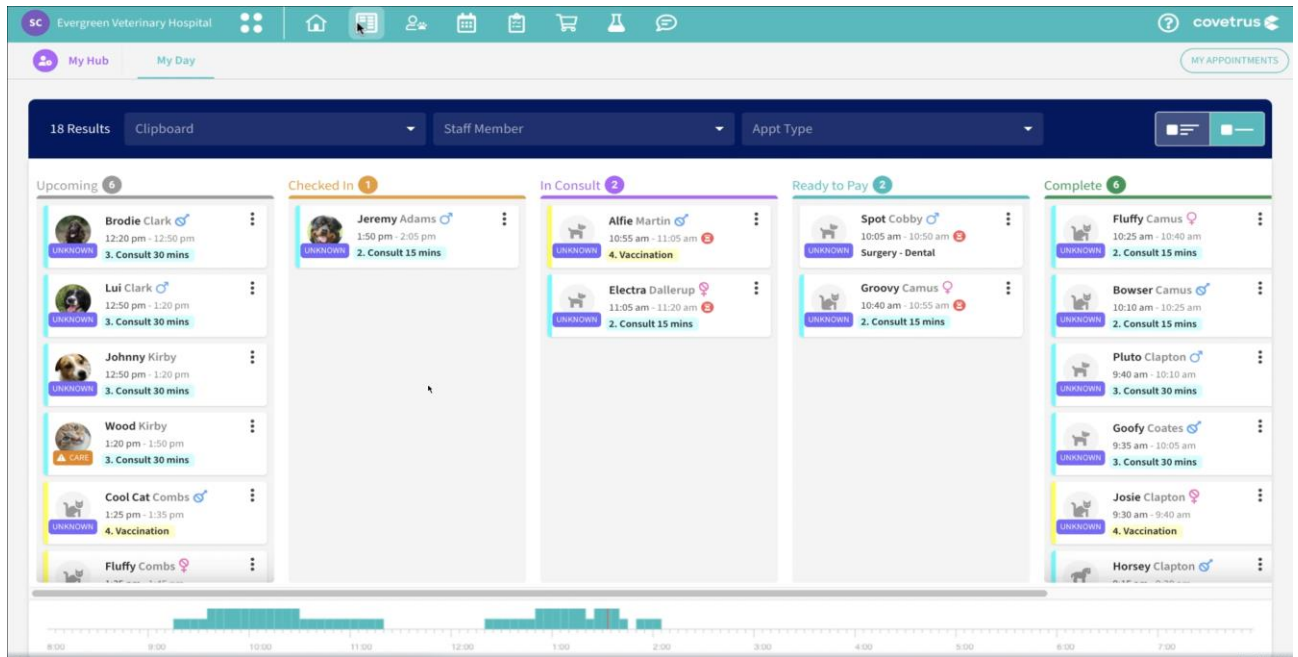


Рис. 1.2.11 Календар в eVetPractice

Медичні записи пацієнта: дозволяють переглядати та редагувати історію медичного лікування тварин, включаючи проведені процедури, результати обстежень, призначення тощо.

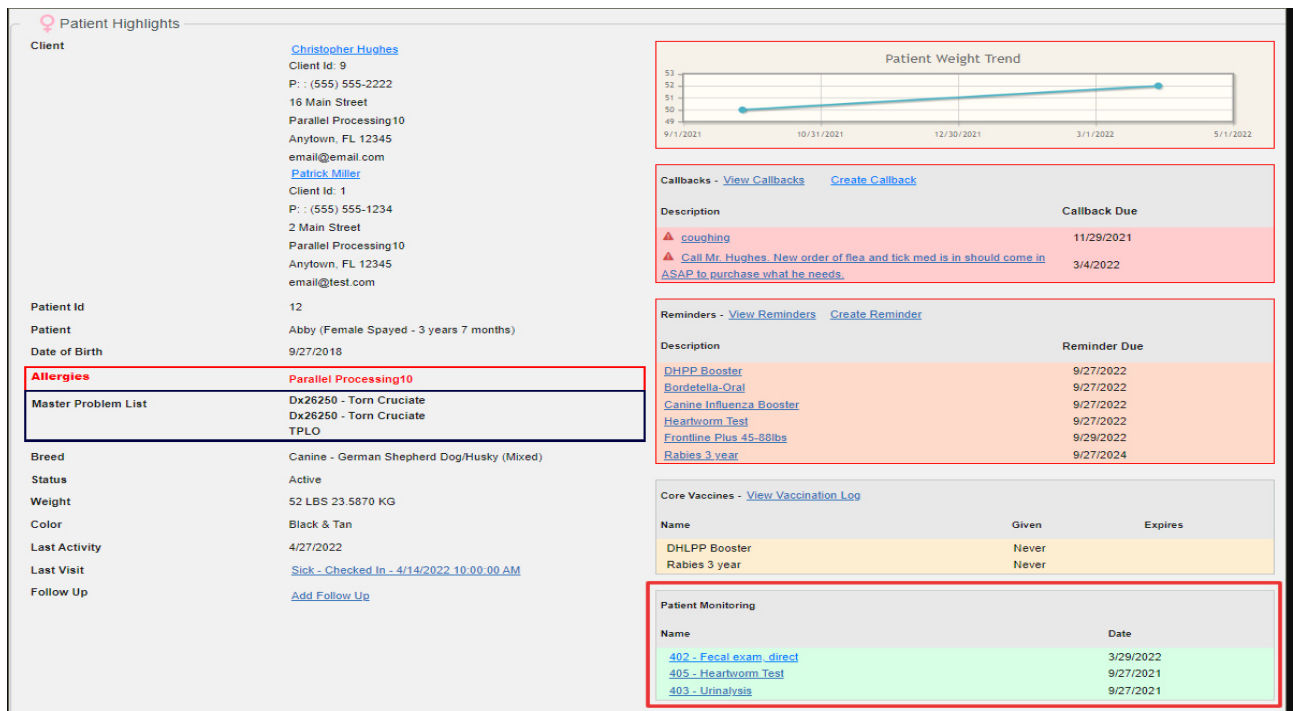


Рис. 1.2.12 Медичні записи пацієнта в eVetPractice

Включають форми для створення нових медичних записів, рецептів, медичних карток, рахунків тощо.

Cash Summary										
Payment Type	Number of Payments					Amount	Percentage of All			
Cash	158					\$146,078.95	99.80 %			
Check	3					\$293.20	0.20 %			
Invoice Post	2					\$0.00				
Total Payments	163					\$146,372.15				
Total Transactions:	163					Total Patient Transactions:	44			
Average Client Transaction:	\$897.99					Average Patient Transaction:	\$3,326.64			
Total Nontaxable Sales *:	\$4,547.03									
Total Taxable Sales * (incl. tax collected):	\$138,772.14									
Sales Tax Collected (minus returns):	\$25,305.65									
Services Tax Collected (minus returns):	\$892.25									
Other Tax Collected (minus returns):	\$17.42									
Number of transactions with Discounts:	99					Total of Discounts:	\$1,729.30			
Number of Coupons:	20					Total of Coupons:	(\$951.02)			
Held Checks:	\$71.59					Total Returns:	\$3,052.98			
<ul style="list-style-type: none"> • (Nontaxable Sales + Taxable Sales) + Total Returns = Total of Cash Summary • Total Taxable Sales includes tax collected • Gross - Exempt + Taxable • Exempt - Nontaxable Sales + Sales Tax Collected • Taxable - taxable sales excluding tax collected • Discounts - given at time of checkout which is usually some percentage off of entire transaction. • Coupons - given at time of treatment and is determined by negative price on a billable item. • ** ACT - Average Client Transaction ("total revenue" / "# of transactions") • ** APT - Average Patient Transaction ("total revenue" / "# of unique patients") 										

Tax Summary										
Tax Rate Type	Number of Transactions	Sales Tax	Sales Pre-Tax Amount	Services Tax	Services Pre-Tax Amount	Other Tax	Other Pre-Tax Amount	Pre-Tax Amount	Post-Tax Amount	Percentage of All
PRACTICE LEVEL	91	\$25,305.65	\$421,760.83	\$892.25	\$22,306.25	\$17.42	\$348.40	\$112,556.82	\$138,772.14	100.00 %
Totals	91	\$25,305.65	\$421,760.83	\$892.25	\$22,306.25	\$17.42	\$348.40	\$112,556.82	\$138,772.14	

Рис. 1.2.13 Форми та документи в eVetPractice

eVetPractice публічно не розголошує конкретні технології та платформи, які лежать в основі їх програмного забезпечення. Ця інформація вважається конфіденційною.

Однак, ґрунтуючись на функціях, які пропонує eVetPractice, можна зробити деякі обґрунтовані припущення щодо потенційних базових технологій:

- **Програмне рішення:** eVetPractice, ймовірно, є хмарним програмним рішенням, тобто воно працює на віддалених серверах, доступ до яких здійснюється через Інтернет. Це позбавляє ветеринарних лікарів необхідності встановлювати та підтримувати програмне забезпечення на окремих комп'ютерах. Потенційними варіантами можуть бути такі популярні хмарні

платформи, як Amazon Web Services (AWS), Microsoft Azure або Google Cloud Platform.

- **Веб-технології:** інтерфейс користувача eVetPractice, ймовірно, побудовано з використанням таких веб-технологій, як HTML, CSS і JavaScript. Це забезпечує зручний інтерфейс, доступний через веб-браузер на різних пристроях.

- **База даних:** для зберігання й керування великою кількістю медичних записів домашніх тварин, даних про зустрічі та іншої інформації eVetPractice, ймовірно, використовує надійну СУБД, як-от MySQL, Oracle або Microsoft SQL Server.

- **Можливості інтеграції:** eVetPractice може інтегруватися з іншим ветеринарним програмним або апаратним забезпеченням, потенційно використовуючи API (інтерфейси прикладного програмування) для полегшення обміну даними.

Користуючись eVetPractice можна сказати, що це ПЗ створене з урахуванням зручності використання для ветеринарних клінік та їх клієнтів. Ось деякі ключові аспекти, які можна виділити :

1. **Інтуїтивний інтерфейс:** eVetPractice пропонує користувачам інтуїтивний та легкий у використанні інтерфейс, який дозволяє швидко засвоїти функціонал системи без додаткових навчань.

2. **Функціональність "в один клік":** Система може надавати швидкий доступ до ключових функцій та операцій через прості інтерфейси та мінімізувати кількість кроків для виконання завдань.

3. **Мобільний доступ:** В залежності від придбаної підписки eVetPractice може бути доступний через мобільний додаток або мобільну версію веб-сайту, що дозволяє користувачам отримувати доступ до системи з будь-якого пристрою, що має підключення до Інтернету.

Висновок:

У порівнянні між IDEXX Cornerstone і eVetPractice видно, що обидва сервіси спрямовані на ветеринарні клініки та пропонують різноманітні функціональність для автоматизації їхньої діяльності. Обидва сервіси мають інтуїтивний інтерфейс та можливість доступу через мобільні пристрої, що полегшує користування ними.

Однак важливо зазначити, що жоден з цих сервісів не має спеціального додатку для самого клієнта, через який він може самостійно переглядати та записуватися на прийом. Це може бути обмеженням для клінік, які хочуть надати клієнтам більше можливостей самостійного взаємодії зі своїм сервісом.

1.3. Вибір структурних і функціональних особливостей застосунку, що розробляється

Основаючись на проведеному аналізі, який був у попередньому підрозділі та порівнянні сервісів IDEXX Cornerstone і eVetPractice, можна виділити наступні структурні і функціональні особливості, які можуть бути корисними для розробки застосунку "Pet":

1. Облік клієнтів:

- Створення профілю клієнта з основними даними.
- Можливість додавання та редагування візитів та медичних даних клієнтів та їх історії.
- Детальний опис попередніх візитів та медичних рекомендацій.

2. Планування та календар:

- Модуль для планування прийому клієнтів та їх домашніх тварин.
- Інтерактивний календар для легкого перегляду та управління розкладом.
- Різний колір під записи (різний статус та попередження про запис).

3. Запис на прийом:

- Онлайн система запису на прийом для клієнтів через веб-портал або мобільний додаток.

4. Онлайн консультації:

- Внутрішній чат або система повідомлень для спілкування з клієнтами та іншими працівниками клініки.

5. Історія візитів:

- Зберігання повної історії медичного обслуговування кожної тварини з можливістю швидкого доступу до неї ("Data backup" автоматично створить резервні копії даних і контролюватиме їх).

6. Фінансові операції:

- Модуль для проведення обліку фінансових операцій (виставлення рахунків та прийому платежів).

7. Звітність:

- Генерація звітів та аналітики щодо різної діяльності клініки.

8. Мобільний додаток:

- Розробити мобільний додаток для зручного доступу клієнтів до всіх основних функцій системи.

Висновки до розділу 1

Під час дослідження теми та порівняння сучасних програмних рішень для ветеринарних клінік було визнано багато вирішальних факторів, які необхідно враховувати при розробці програмного застосунку "Pet". Результати дослідження показують, що основними критеріями медичного програмного забезпечення для домашніх тварин є ефективність, надійність і зручність для користувача. Досліджуючи ці два сервіси: IDEXX Cornerstone та eVetPractice, було проаналізовано їхні переваги та недоліки. Цей аналіз допоміг встановити фундаментальні критерії для функціонування програмного забезпечення "Pet".

Зокрема, було вирішено, що програма "Pet" повинна мати функції для керування інформацією про клієнтів і тварин, планування та відстеження візитів, доступу до медичних записів, обробки фінансових транзакцій, а також створення звітів і аналітики. Важливими аспектами є інтеграція з іншими системами та наявність мобільного додатку для легкого доступу споживачів. Забезпечення надійних заходів безпеки та суворих протоколів конфіденційності даних також є головною відповідальністю.

Враховуючи всі поставлені цілі та вимоги, створення програмного додатку "Pet" значно підвищить рівень лікування домашніх тварин, гарантує зручність та ефективність роботи ветеринарних клінік, а також підвищить рівень задоволеності клієнтів.

2. АНАЛІЗ І ВИБІР МЕТОДУ УПРАВЛІННЯ ПРОЄКТОМ РОЗРОБКИ ЗАСТОСУНКУ

2.1. Застосування методології управління проектами в галузі інформаційних систем та технологій

Ефективне управління проектами в галузі інформаційних систем та технологій є критичним аспектом для успішної реалізації ІТ-проектів, які можуть включати розробку ПЗ, впровадження інформаційних систем, розробку веб-додатків, мобільних додатків тощо. Відповідно до специфіки ІТ-проектів, ефективне управління проектами забезпечує досягнення цілей проекту в межах визначених часу, бюджету та якості. У ІТ-проектах використання методологій управління проектами дозволяє структурувати роботу, встановлювати чіткі цілі та вимоги, забезпечувати ефективне використання ресурсів і контролювати хід реалізації проекту.

Управління ІТ-проектами завжди супроводжується високим рівнем невизначеності. У цій галузі проекти часто характеризуються швидкими змінами у вимогах і технологіях, що вимагає від команди високої гнучкості та здатності швидко адаптуватися до нових умов. Технологічні інновації та зміни в ІТ-індустрії можуть суттєво вплинути на перебіг проекту, тому управління ризиками є одним з ключових аспектів. Важливим в цьому сенсі фактором є швидкість зміни технологій. ІТ-сфера постійно еволюціонує, що вимагає від кожного з команд не лише вчасно освоювати нові технології, але й інтегрувати їх у поточні проекти. Це значно підвищує складність управління проектами, оскільки необхідно постійно стежити за останніми трендами та впроваджувати інноваційні рішення в проекти.

ІТ-проекти потребують високого рівня якості, адже забезпечення якості програмного забезпечення є надзвичайно важливим. Помилки чи недоліки можуть спричинити серйозні фінансові та репутаційні втрати. Ефективне управління якістю вимагає ретельного планування, ретельного тестування та пильного контролю якості протягом усього життєвого циклу проекту. Цей підхід

мінімізує помилки і гарантує вищий кінцевий продукт. Ефективне спілкування є ще одним важливим аспектом. Ефективне спілкування є ключовим для успішного управління ІТ-проектами. Важливо встановити чіткі лінії комунікації між усіма зацікавленими сторонами (внутрішніми та зовнішніми): командами розробників, замовниками, користувачами, керівництвом тощо. Працюючи з думкою керівника проекту, важливо надати пріоритет прозорості та підтримувати регулярне спілкування, щоб сприяти послідовності та запобігати можливим непорозумінням. Прозорість та регулярний обмін інформацією допомагають забезпечити узгодженість і уникнути непорозумінь.

Для досягнення цих цілей використовуються різні методології управління проектами. Про них і піде розповідь у даному розділі

2.2. Класичний підхід до розробки програмних продуктів

Фреймворк Waterfall, або "водоспад", є одним з традиційних і широко використовуваних підходів до розробки програмних продуктів. Вона була вперше сформована у 1970 році Вінстоном Ройсом і залишається популярною в багатьох галузях, де вимоги до проекту можуть бути чітко визначені на початку.

Waterfall отримав свою назву через лінійну послідовність етапів, кожен з яких витікає з попереднього подібно до каскаду водоспаду. Основна ідея цього підходу полягає у тому, що проект проходить через кілька чітко визначених фаз, які виконуються послідовно одна за одною. Кожен етап має завершитися повністю перед тим, як розпочнеться наступний.

Основні етапи Waterfall включають:

1. Визначення вимог. Цей крок зосереджений на зборі всіх системних вимог від замовника або кінцевих користувачів. Зазначені потреби мають бути задокументовані та затверджені. Далі ці документи слугують основою для наступних етапів розробки. Важливість цього етапу полягає в тому, що зміни, внесені на наступних етапах, можуть виявитися дуже складними і дорогими.

2. Аналіз системи. На цьому етапі попередні вимоги формують в точні технічні специфікації, які визначають архітектуру системи та її складові елементи.

3. *Проектування.* На цьому етапі розробляється детальний дизайн системи, включаючи інтерфейси користувача, архітектуру програмного забезпечення, модулі та їхні взаємозв'язки.

4. *Реалізація.* На цьому етапі розробники втілюють вимоги в код та створюють зовнішній вигляд застосунку на основі наданих специфікацій та дизайну.

5. *Тестування.* Далі система проходить серію тестів, щоб перевірити, чи відповідає вона заданим критеріям і виявити будь-які потенційні проблеми.

6. *Впровадження.* Наступним кроком буде розгортання продукту для завантаження у різні маркети для користування.

7. *Супровід.* Після інсталяції система переходить у фазу обслуговування, що включає усунення помилок, які не виявили під час тестування, модифікацію, оновлення продукту та допомогу користувачам з різним типом питань.

Переваги Waterfall включають в себе чіткість і структурованість процесу, це дозволяє легко планувати і контролювати проект. Кожен етап має визначений початок і кінець, а також вихідні дані, які можуть бути використані для оцінки прогресу і прийняття рішень. Це корисно для тих проектів, які заздалегідь попіклувалися про вимоги та є мала ймовірності того, що вони будуть змінені.

Однак Waterfall має і недоліки. Основна проблема стосується його жорсткості і відсутності гнучкості. У реальних проектах вимоги часто можуть змінюватися протягом усього життєвого циклу, і в цьому випадку, Waterfall не може забезпечити ефективний механізм для адаптації до таких змін. Внесення змін на пізніх етапах може бути дуже складним і дорогим, оскільки це може вимагати значного перегляду усіх етапів у проекті. Це може призвести до затримок і перевищення бюджету. Крім того, кінцевий продукт стає доступним для користувачів лише на фінальних етапах проекту, що означає, що зворотній зв'язок від користувачів надходить занадто пізно для внесення значущих змін.

Дослідження і практичний досвід показують, що фреймворк Waterfall (водоспад) ефективний для проектів з чітко визначеними вимогами та низьким рівнем невизначеності.

Приклади використання Waterfall:

- **Проект розробки процесора Intel:** Розробка нових процесорів Intel, таких як серія Intel Core або Intel Xeon, зазвичай виконується з використанням Waterfall підходу. Це пов'язано з чітко визначеними етапами розробки, такими як дослідження, розробка архітектури, проектування схем, виробництво та тестування. Кожен етап потребує завершення перед переходом до наступного, що допомагає забезпечити високу якість і надійність кінцевого продукту.
- **Розробка відеокарт NVIDIA** Проекти з розробки відеокарт, такі як серія NVIDIA GeForce або Quadro, також слідують Waterfall підходу. Вимоги до продуктивності, енергоефективності та сумісності з іншими компонентами системи визначаються на початкових етапах, після чого проект проходить через послідовні фази проектування, виробництва і тестування. Це дозволяє уникнути критичних помилок, які можуть виникнути через невідповідність специфікаціям.

Ці приклади та дослідження показують, що Waterfall є ефективним підходом для проектів з чітко визначеними вимогами і стабільним середовищем. Однак для більшості ІТ-проектів, які характеризуються високою динамічністю і частими змінами, більш ефективними є гнучкі методології, такі як Agile.

2.3. Застосування гнучкого підходу в розробці ІТ-рішень

У сучасному світі інформаційних технологій дедалі частіше використовується гнучкий підхід (Agile) для розробки ІТ-рішень. На відміну від класичних методів, які орієнтовані на створення завершених продуктів, гнучкі методології зосереджені на постійному вдосконаленні та адаптації продукту відповідно до змінних вимог і потреб користувачів. Це зумовлено тим, що ІТ-рішення за своєю природою мають бути гнучкими та швидко реагувати на зміни ринку, технологій і очікувань клієнтів.

Agile дозволяє командам працювати більш ефективно, реагувати на зворотній зв'язок від користувачів у режимі реального часу і постійно покращувати кінцевий продукт. Основна ідея полягає у тому, щоб створювати невеликі інкременти робочого продукту, які можна швидко тестувати і впроваджувати, а не чекати завершення всього проекту. Це допомагає

мінімізувати ризики і забезпечити, що кінцеве рішення буде максимально відповідати вимогам користувачів.

У відповідь на усі ці виклики все більше організацій переходять до гнучких підходів (Agile), таких як Scrum, Kanban та інші. Про них і піде слово далі.

2.3.1. Scrum

Scrum – один з найпопулярніших і найпоширеніших методів гнучкого управління проектами, особливо в сфері розробки ПЗ. Scrum допомагає організаціям ефективно управляти складними проектами і швидко адаптуватися до змін. Цей метод, або набір методів та методик, ґрунтується на чітких принципах і структурованих процесах, що дозволяє досягати високої продуктивності і якості продукту.

Команда Scrum складається з трьох основних ролей: Product Owner, Scrum Master і Development Team.

- Product Owner відповідає за максимізацію цінності продукту, створеного командою. Він керує беклогом продукту, визначає пріоритети і забезпечує, щоб команда розуміла вимоги і цілі.

- Scrum Master відповідає за дотримання принципів і практик Scrum, допомагає команді працювати ефективно і усуває перешкоди, що заважають роботі команди.

- Development Team – це невелика крос-функціональна група професіоналів, відповідальна за створення інкрементів робочого продукту на кожному спринті.

Підхід Scrum також включає три основні артефакти: Product Backlog, Sprint Backlog і Increment.

- Product Backlog – це динамічний список всіх функцій, змін, виправлень і вимог, які потрібно розробити для продукту. Він постійно оновлюється і пріоритизується Product Owner'ом.

- Sprint Backlog – це підмножина Product Backlog, обрана для роботи на поточному спринті. Включає задачі, які команда зобов'язується виконати протягом спринту.

- Increment – це робочий продукт або результат спринту, який повинен бути завершеним і готовим до релізу. Кожен інкремент додається до попередніх, забезпечуючи поступове створення кінцевого продукту.

Scrum визначає п'ять основних подій, які структурують процес розробки і забезпечують регулярний зворотний зв'язок.

- Спринт – основний цикл розробки, тривалістю від одного до чотирьох тижнів. У кожному спринті команда створює інкремент робочого продукту.

- Планування спринту – зустріч на початку спринту, де команда планує роботу на спринт, визначає цілі і задачі.

- Daily Scrum – щоденна коротка зустріч (до 15 хвилин), де команда обговорює прогрес, плани на день і можливі перешкоди.

- Sprint Review – зустріч в кінці спринту, де команда демонструє завершену роботу зацікавленим сторонам і обговорює зворотний зв'язок.

- Sprint Retrospective – зустріч після завершення спринту, де команда аналізує процес роботи, обговорює успіхи і проблеми, шукає способи покращення роботи в наступних спринтах.

Scrum забезпечує високу гнучкість і адаптивність, дозволяючи швидко реагувати на зміни вимог і отримувати зворотний зв'язок від користувачів. Це особливо важливо в умовах швидко змінюваного ІТ-ринку, де потреби користувачів можуть змінюватися дуже швидко. Scrum також забезпечує підвищену якість продукту завдяки регулярним оглядам і тісній співпраці з замовником. Прозорість процесу розробки, яку забезпечують постійні зустрічі і чіткі вимоги, дозволяє всім зацікавленим сторонам бути в курсі прогресу проекту. Самоорганізовані команди і активна участь у прийнятті рішень підвищують мотивацію і задоволеність роботою.

Дослідження підтверджують ефективність Scrum. Згідно зі звітом Standish Group CHAOS, проекти, що використовують гнучкі методології, мають значно

вищий рівень успіху в порівнянні з традиційними методами. Дослідження також показують, що Scrum допомагає зменшити час на розробку і збільшити якість кінцевого продукту.

Наприклад, компанія Microsoft використовує Scrum для управління розробкою багатьох своїх продуктів, таких як Windows і Office. Це дозволяє їм швидко впроваджувати нові функції і адаптуватися до зворотного зв'язку користувачів. У Google багато команд застосовують Scrum для управління проектами, такими як Google Search і Google Ads, що забезпечує гнучкість і швидку адаптацію до змінних ринкових умов. Розробка ігор в Electronic Arts, таких як FIFA і Battlefield, також використовує Scrum, що дозволяє командам швидко впроваджувати нові ідеї, тестувати їх і отримувати зворотний зв'язок від гравців.

У висновку: Scrum є потужним інструментом для управління проектами в ІТ-сфері, дозволяючи командам працювати ефективніше, швидше адаптуватися до змін і створювати високоякісні продукти. Його гнучкість і орієнтація на командну роботу роблять його ідеальним вибором для багатьох сучасних ІТ-проектів.

2.3.2. Kanban

Kanban – це метод управління проектами, який має свої корені в системі виробництва Toyota. Він був створений для підвищення ефективності виробничих процесів, але згодом став широко застосовуватися в сфері ІТ, зокрема в розробці ПЗ. Kanban допомагає командам візуалізувати робочі процеси, зменшувати втрати і забезпечувати безперервний потік роботи.

Основна ідея Kanban полягає у використанні візуальної системи для управління роботою. Дошка Kanban – це інструмент, який використовує картки і колонки для відображення етапів робочого процесу і стану завдань. Дошка складається з колонок, кожна з яких відповідає певному етапу процесу, наприклад, "Завдання", "В процесі", "Готово" ("To Do", "In Progress", "Done").

). Завдання або елементи роботи представлені картками, які переміщуються між колонками в міру їх виконання. Це дозволяє команді легко бачити, на якому етапі знаходиться кожне завдання і які етапи потребують уваги.

Kanban має кілька ключових принципів, які допомагають оптимізувати робочий процес:

Візуалізація роботи: Усі завдання та їх стан візуально відображаються на дошці Kanban. Це дозволяє легко та ефективно управляти проектом.

Обмеження незавершеної роботи (WIP): Kanban задає обмеження на кількість завдань, які можуть знаходитися на кожному етапі процесу одночасно. Загалом це допомагає запобігати переробкам команди і забезпечує стабільний процес роботи.

Управління потоком: Kanban наголошує увагу на безперервній роботі, з метою уникнення затримок і підвищення ефективності. Команда постійно аналізує і вдосконалює процеси, щоб всі етапи мали плавний потік завдань від початку до кінця.

Явні правила процесу: Всі правила, які були встановлені, повинні бути чітко визначені і зрозумілі для всіх членів команди.

Регулярне вдосконалення: Kanban передбачає постійний аналіз і вдосконалення процесу роботи над проектом. Команда регулярно проводить зустрічі, де обговорює проблеми, які виникають, і шукає способи їх вирішення.

Kanban забезпечує високу гнучкість і дозволяє командам швидко адаптуватися до змін. Завдяки візуалізації робочих процесів і обмеженням на незавершену роботу, Kanban допомагає зменшити втрати часу і підвищити продуктивність.

Дослідження показують, що використання Kanban може значно підвищити ефективність роботи команд.

Наприклад, компанія Hewlett-Packard використовувала Kanban для управління розробкою програмного забезпечення і змогла зменшити час виконання завдань на 37% і підвищити продуктивність на 50%. Інший приклад – компанія Spotify, яка використовує Kanban для управління своїми командами, що

дозволяє їм швидко реагувати на зміни в вимогах і забезпечувати високий рівень якості продукту.

У висновку: Kanban є ефективним інструментом для управління проектами в IT-сфері, дозволяючи командам працювати більш організовано і ефективно. Завдяки візуалізації робочих процесів, обмеженням на незавершену роботу і постійному вдосконаленню, Kanban допомагає командам забезпечувати безперервний потік роботи і досягати високих результатів.

2.3.3. Змішані моделі, DSDM

Змішані моделі розробки програмного забезпечення, зокрема DSDM (Dynamic Systems Development Method), є підходом, який поєднує елементи різних методологій для створення ефективного процесу управління проектами. DSDM є однією з перших гнучких методологій і була створена в 1994 році як відповідь на зростаючу потребу в гнучких методах розробки систем. Основна мета DSDM полягає в забезпеченні якісного і швидкого створення інформаційних систем, які відповідають потребам бізнесу.

DSDM ґрунтується на дев'яти принципах, які визначають її філософію і підхід до управління проектами:

Фокус на бізнес-потребах: Проекти DSDM орієнтовані на задоволення конкретних бізнес-потреб і забезпечення цінності для замовника. Всі рішення в процесі розробки повинні базуватися на бізнес-пріоритетах.

Встановлення жорстких термінів: Час і ресурси є фіксованими, а функціональність системи може бути адаптована відповідно до цих обмежень. Це забезпечує, що проекти завершуються вчасно і в межах бюджету.

Постійна співпраця: Тісна співпраця між розробниками, замовниками і користувачами є ключовим елементом DSDM. Це забезпечує постійний зворотний зв'язок і дозволяє швидко реагувати на зміни вимог.

Поступова розробка: Система створюється поступово через ітерації, кожна з яких додає нову функціональність і покращує існуючу. Це дозволяє постійно оцінювати прогрес і забезпечувати якість на кожному етапі.

Інкрементальна доставка: Продукт доставляється частинами, кожна з яких представляє собою працюючу версію системи. Це дозволяє замовнику бачити результати роботи на ранніх етапах і надавати зворотний зв'язок.

Інтегроване тестування: Тестування є невід'ємною частиною процесу розробки і виконується на всіх етапах. Це забезпечує виявлення і виправлення помилок на ранніх етапах і підвищує якість кінцевого продукту.

Контроль за процесом: DSDM використовує контрольні точки і регулярні зустрічі для моніторингу прогресу проекту і внесення коректив. Це дозволяє підтримувати високу якість і своєчасність виконання завдань.

Розподіл обов'язків: Всі учасники проекту мають чітко визначені ролі і обов'язки, що забезпечує ефективну комунікацію і координацію роботи.

Постійне вдосконалення: Процес розробки постійно вдосконалюється на основі зворотного зв'язку і накопиченого досвіду. Це дозволяє підвищувати ефективність і якість роботи з кожним проектом.

DSDM також передбачає використання певних інструментів і технік для забезпечення ефективного управління проектом. Одним з основних інструментів є модель проекту, яка включає чотири основні фази: дослідження, функціональна модель, проектування та реалізація, перехід. Кожна з цих фаз включає в себе ітерації, що дозволяють поступово розробляти і вдосконалювати продукт.

Дослідження показують, що використання DSDM може значно підвищити ефективність і якість розробки ПЗ.

Наприклад, дослідження, проведене у Великобританії, показало, що компанії, які використовують DSDM, досягають вищих показників задоволеності замовників і якості продукту в порівнянні з тими, які використовують традиційні методи розробки.

У висновку: Змішані моделі, такі як DSDM, поєднують переваги різних підходів і забезпечують високу гнучкість і адаптивність. Вони дозволяють командам швидко реагувати на зміни вимог, забезпечувати високу якість продукту і задовольняти бізнес-потреби замовників. DSDM є ефективним інструментом для управління складними ІТ-проектами і може бути успішно застосований в різних галузях і контекстах.

2.3.4. Масштабовані моделі Scrum

Масштабовані моделі Scrum розроблені для застосування принципів та практик Scrum у великих організаціях і на проектах, що потребують роботи декількох команд одночасно. Оскільки Scrum спочатку створювався для невеликих команд, масштабування його до рівня, коли в проекті беруть участь сотні або навіть тисячі розробників, вимагає адаптації. Існує кілька популярних фреймворків для масштабування Scrum, серед яких найбільш відомі - це LeSS (Large-Scale Scrum), SAFe (Scaled Agile Framework) та Nexus.

LeSS (Large-Scale Scrum) – це фреймворк, який дозволяє масштабувати Scrum на кілька команд, що працюють над одним продуктом. LeSS базується на принципах Scrum, зберігаючи простоту і орієнтованість на доставку цінності. LeSS передбачає використання єдиного Product Backlog для всіх команд, а також спільних заходів, таких як спільне планування спринтів та ретроспективи. Важливою особливістю LeSS є його фокус на усуненні відходів та оптимізації всього процесу розробки.

SAFe (Scaled Agile Framework) – це один із найпопулярніших фреймворків для масштабування Scrum і інших гнучких методологій. SAFe дозволяє інтегрувати принципи Agile та Lean у великі організації, забезпечуючи координацію роботи декількох команд та управління великими програмами і портфелями проектів. SAFe включає чотири рівні: Team, Program, Large Solution та Portfolio, кожен з яких має свої ролі, артефакти та події. Це дозволяє великим організаціям підтримувати гнучкість на рівні команд і забезпечувати стратегічне управління на вищих рівнях.

Nexus – це фреймворк, розроблений засновниками Scrum, Кеном Швабером і Джеффом Сазерлендом, для масштабування Scrum на кілька команд. Nexus додає мінімум додаткових ролей, артефактів та подій до стандартного Scrum, що забезпечує простоту і ефективність. Основним артефактом Nexus є Nexus Integration Team, яка відповідає за координацію роботи декількох команд і інтеграцію їх роботи в єдиний продукт. Nexus також використовує Nexus Sprint

Backlog для відстеження прогресу роботи і забезпечення узгодженості між командами.

Дослідження показують, що використання масштабованих моделей Scrum може значно підвищити ефективність великих проєктів.

Наприклад, дослідження, проведене компанією VersionOne, показало, що організації, які використовують SAFe, досягають кращих показників продуктивності і задоволеності замовників порівняно з тими, які використовують традиційні методи управління проєктами.

Застосування масштабованих моделей Scrum вимагає не тільки адаптації процесів, але й зміни організаційної культури. Команди повинні навчитися працювати разом, дотримуючись принципів прозорості, інспекції і адаптації. Це може включати регулярні зустрічі для обговорення прогресу, виявлення проблем і пошуку шляхів їх вирішення. Важливим аспектом є також забезпечення безперервної інтеграції та автоматизованого тестування, що дозволяє зменшити ризики і підвищити якість продукту.

Таким чином, масштабовані моделі Scrum, такі як LeSS, SAFe і Nexus, забезпечують ефективний підхід до управління великими проєктами, дозволяючи командам зберігати гнучкість і продуктивність навіть у великих організаціях. Вони забезпечують координацію роботи, оптимізацію процесів і підвищення якості продукту, що є ключовими факторами успіху в сучасному світі IT.

Висновки до розділу 2

Після детального аналізу різних підходів до управління проектами, включаючи класичний Waterfall, а також сучасні гнучкі методології, такі як Scrum, Kanban, DSDM та масштабовані моделі Scrum, стає очевидним, що для нашого проекту розробки медичної інформаційної системи "Pet" найкраще підходить саме Agile.

Вибір Agile методології обґрунтований наступними причинами:

1. **Гнучкість та адаптивність.** Agile забезпечує високу гнучкість та здатність швидко реагувати на зміни вимог або зовнішніх умов. У цьому IT-проекті, де потреби користувачів та технологічні вимоги можуть змінюватися, можливість швидкої адаптації є критично важливою.

2. **Ітеративний підхід.** Agile базується на ітераційному процесі, що дозволяє регулярно переглядати та покращувати продукт на основі зворотного зв'язку. Це дозволяє поступово додавати нові функціональності, що є важливим для складних проектів, таких як МІС, де потрібні регулярні оновлення та удосконалення.

3. **Співпраця та залучення зацікавлених сторін.** Agile методології, зокрема Scrum, передбачають активну участь замовника та інших зацікавлених сторін у всіх етапах проекту. Це забезпечує краще розуміння вимог та очікувань, що дозволяє створювати продукт, який максимально відповідає потребам користувачів.

4. **Підвищена продуктивність та якість.** Використання Agile сприяє підвищенню продуктивності команд через регулярні зустрічі, чіткі завдання та постійне вдосконалення процесів. Інтегроване тестування на кожному етапі створення застосунку "Pet" забезпечує високу якість кінцевого продукту.

5. **Зменшення ризиків.** Завдяки регулярним випускам та постійній інтеграції, Agile дозволяє виявляти та вирішувати проблеми на ранніх етапах, що значно знижує ризики невдачі проекту. Це особливо важливо для проекту в сфері охорони здоров'я, де будь-які помилки можуть мати серйозні наслідки.

6. **Ефективне використання ресурсів.** Agile підходи дозволяють більш ефективно використовувати ресурси, забезпечуючи правильний розподіл

завдань та оптимізацію часу роботи команди. Це дозволить зменшити витрати та підвищити ефективність у проекті.

7. **Підтримка інновацій.** Agile сприяє творчості та інноваціям завдяки відкритій комунікації та експериментуванню. Це особливо корисно для проекту "Pet", що включають нові технології та методи.

З огляду на ці переваги, було прийнято рішення використовувати Agile методологію для управління проектом розробки медичної інформаційної системи "Pet". Цей підхід дозволить створити якісний продукт, який буде відповідати потребам користувачів, забезпечить гнучкість в розробці та дозволить ефективно управляти ресурсами та часом.

3. ПРОЕКТНІ ТА ТЕХНІЧНІ РІШЕННЯ ЗАСТОСУНКУ РЕТ

Хоча існують успішні та комплексні системи охорони здоров'я, які покращують результати лікування чотириногих пацієнтів, краща система управління лікарнею (далі — СУЛ) завжди може підвищити ефективність. Цей проект має на меті зібрати приклади минулих спроб, розробити високорівневе представлення обсягу роботи і забезпечити глибокий аналіз даних, що використовуються в системі управління лікарнею. Це представлення буде включати різні моделі даних, які розглядають представлення даних, необхідних для планування прийому, роботи лікарні та відстеження фінансів.

3.1 Концептуальний дизайн

Даний звіт почнеться з логічного проектування та моделювання вказаних нижче набору даних. По-перше, потрібно розробити діаграму ER/EER разом з усіма основними припущеннями, яка візуально представляє сутності та зв'язки між ними.

3.1.1 Діаграма EER

Розширені діаграми сутності-зв'язку (EER) є важливою частиною інтерфейсу моделювання зокрема в MySQL Workbench, який буде використаний в даній роботі. Діаграми EER надають візуальне представлення зв'язків між таблицями у моделі. Зміни, внесені за допомогою редактора моделей, показано на відповідній діаграмі. Подібним чином зміни, які вносяться в діаграму, легко реєструються у відповідному редакторі моделей.

На рисунку 3.1.1 представлена діаграма EER, створена на основі опису проекту:

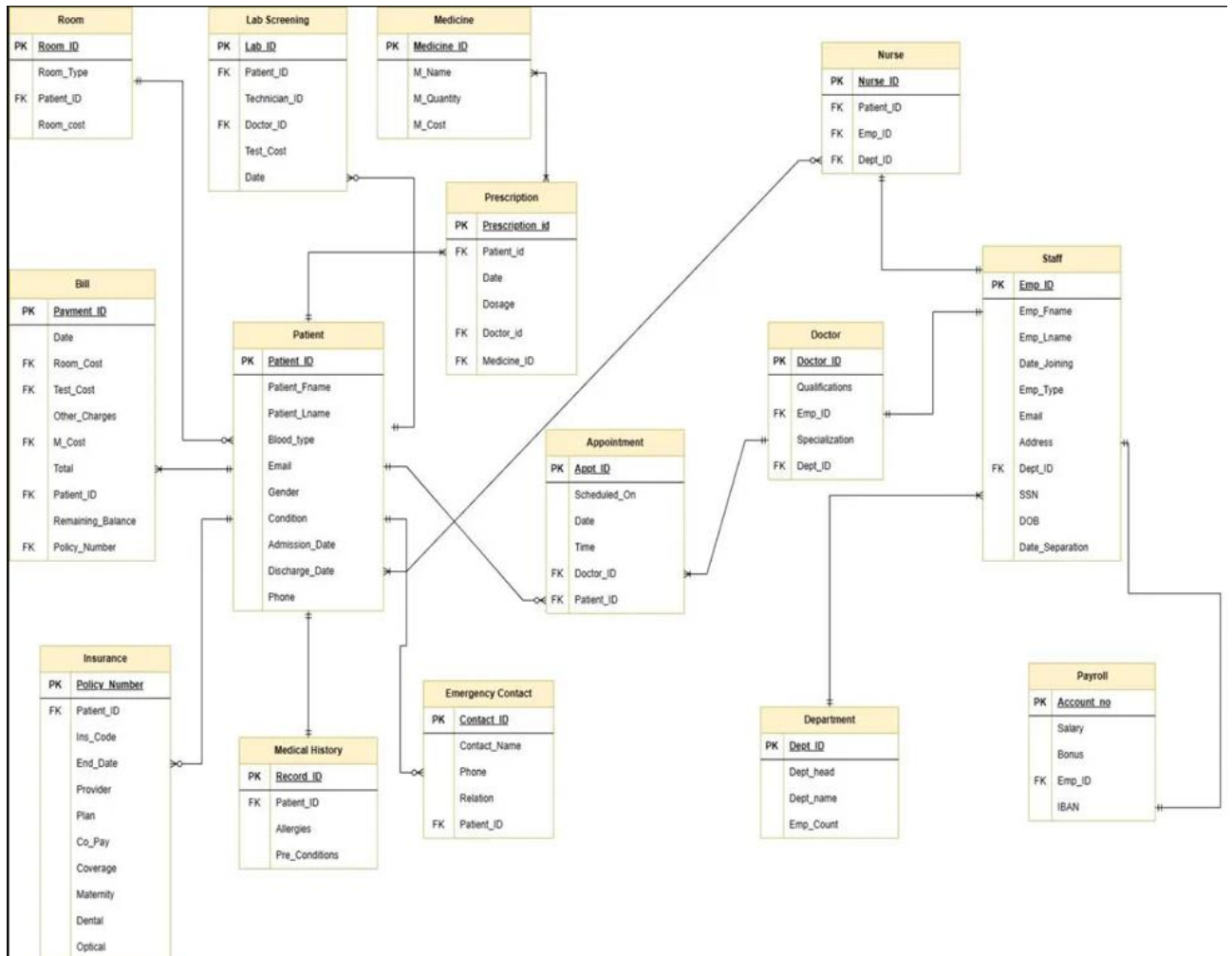


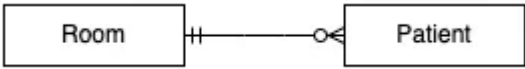

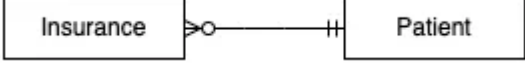

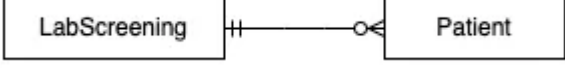
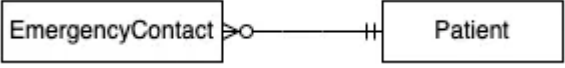
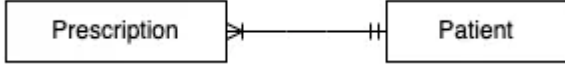

Рис. 3.1.1 Діаграма EER для СУЛ

3.1.2 Взаємозв'язки між сутностями

У цьому розділі розглядаються позначення min-max, що використовуються на діаграмі EER для зображення зв'язків між сутностями.

Нотації "Crow Foot" детально описані в Таблиці 3.1, щоб полегшити розуміння та інтерпретацію зв'язків, які існують в реляційній базі даних.

Нотації "Crow Foot"

Представлення	Опис
	<p>Пацієнт має бути закріплений за однією, і лише в одну кімнату. У кімнати може бути нуль або багато пацієнтів</p>
	<p>Пацієнт може мати від одного до багатьох рахунків. Рахунок повинен бути виставлений одному і тільки одному пацієнту</p>
	<p>Пацієнт може мати кілька страхових полісів. Страховий поліс повинен бути пов'язаний з тим самим пацієнтом</p>
	<p>Пацієнт повинен мати медичну історію хвороби в лікарні. Історія хвороби повинна бути пов'язана з пацієнтом</p>
	<p>Пацієнт може мати від нуля до багатьох лабораторних скринінгових процедур. Запис про лабораторний скринінг пов'язаний з одним і тільки з одним пацієнтом</p>
	<p>Пацієнт може мати від нуля до багатьох екстрених контактів. Екстрений контакт повинен бути пов'язаний з одним і тільки одним пацієнтом</p>
	<p>Пацієнт може мати від одного до багатьох рецептів. Рецепт повинен бути пов'язаний з одним і тільки одним пацієнтом</p>
	<p>Рецепт може містити від одного до багато ліків. Ліки можуть бути виписані в одному або декількох рецептах</p>

	<p>Пацієнт може бути записаний на декілька прийомів. Прийом повинен бути пов'язаний з одним і тільки одним пацієнтом.</p>
	<p>Лікар може мати від одного до багатьох прийомів. Прийом повинен бути запланований з одним і тільки одним лікарем</p>
	<p>Медсестра може бути закріплена за одним або багатьма пацієнтами. Пацієнт може мати від нуля до багатьох медсестер.</p>
	<p>Лікар повинен мати один і тільки один запис під стафом. Запис про персонал може бути пов'язаний тільки з одним і тільки одним лікарем.</p>
	<p>Медсестра повинна мати один і тільки один запис у розділі «Персонал». Запис про персонал може бути пов'язаний тільки з однією і тільки однією медсестрою,</p>
	<p>Платіжна відомість повинна мати один і тільки один запис у розділі «Персонал». Штатний запис може бути пов'язаний лише з одним і тільки одним записом у платіжній відомості записом.</p>
	<p>Співробітник повинен працювати в одному і тільки одному відділі. У підрозділі може бути від одного до багатьох співробітників</p>

3.2 Реляційна схема

Діаграма підкреслює зв'язки між кожною таблицею, показуючи, як кожен зовнішній ключ пов'язаний з первинним ключем батьківської таблиці. Це схема, яка показує, як дані співвідносяться та вибираються з бази даних. Наприклад,

розміщення Emp_ID в таблиці Nurse і Doctor дозволяє отримати відповідні дані, коли Emp_ID викликається в запиті.

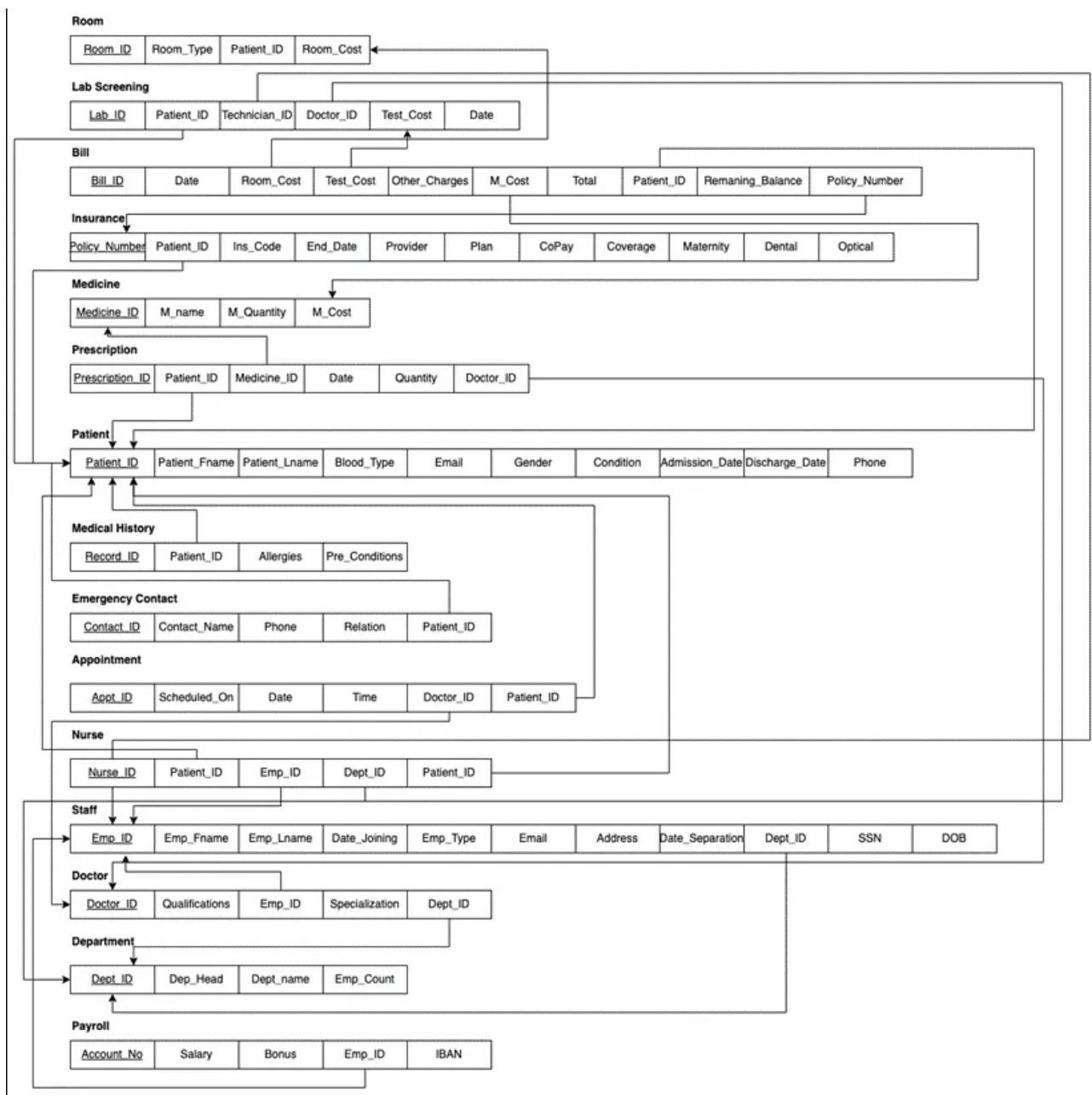


Рис.3.2.1 Реляційна схема для бази даних

3.2.1 Формат даних для кожного відношення

В таблицях нижче представлені формати даних атрибутів розроблених таблиць бази даних.

Таблиця Кабінет

Relation Names	Attributes	Data Type
Room	Room_ID	INT
	Room_Type	VARCHAR(50)
	Patient_ID	INT
	Room_Cost	Decimal(10,2)

```
CREATE TABLE Room (
  Room_ID INT NOT NULL,
  Room_Type VARCHAR(50) NOT NULL,
  Patient_ID INT NOT NULL,
  Room_Cost DECIMAL(10,2),
  PRIMARY KEY (Room_ID),
  FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID)
);
```

Таблиця 3.2.2

Таблиця Рахунок

Relation Names	Attributes	Data Type
Bill	Bill_ID	INT
	Date	DATA
	Room_Cost	INT
	Test_Cost	Decimal(10,2)
	Other_Charges	Decimal(10,2)
	M_Cost	Decimal(10,2)
	Total	Decimal(10,2)
	Patient_ID	INT
	Remaining_Balance	Decimal(10,2)
	Policy_Number	VARCHAR(20)

```
CREATE TABLE Bill (
  Bill_ID INT NOT NULL,
  Date DATE NOT NULL,
  Room_Cost DECIMAL(10,2),
  Test_Cost DECIMAL(10,2),
  Other_Charges DECIMAL(10,2),
```

```

M_Cost DECIMAL(10,2),
Total DECIMAL(10,2),
Patient_ID INT NOT NULL,
Remaining_Balance DECIMAL(10,2),
Policy_Number VARCHAR(20) NOT NULL,
PRIMARY KEY (Payment_ID),
FOREIGN KEY (Room_Cost) REFERENCES Room (Room_Cost),
FOREIGN KEY (Test_Cost) REFERENCES Lab_Screening (Test_Cost),
FOREIGN KEY (M_Cost) REFERENCES Medicine (M_Cost),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
FOREIGN KEY (Policy_Number) REFERENCES Insurance (Policy_Number)
);

```

Таблиця 3.2.3

Таблиця Пацієнт

Relation Names	Attributes	Data Type
Patient	Patient_ID	INT
	Patient_FName	VARCHAR(20)
	Patient_LName	VARCHAR(20)
	Phone	VARCHAR(12)
	Blood_Type	VARCHAR(5)
	Email	VARCHAR(50)
	Gender	VARCHAR(10)
	Condition	VARCHAR(30)
	Admission_Data	DATA
	Discharge_Data	DATA

```

CREATE TABLE Patient (
  Patient_ID INT NOT NULL,
  Patient_FName VARCHAR(20) NOT NULL,
  Patient_LName VARCHAR(20) NOT NULL,
  Phone VARCHAR(12) NOT NULL,
  Blood_Type VARCHAR(5) NOT NULL,
  Email VARCHAR(50),
  Gender VARCHAR(10),
  Condition_ VARCHAR(30),
  Admission_Date DATE,
  Discharge_Date DATE,
  PRIMARY KEY (Patient_ID)
);

```

3.3 Нормалізація

У цій частині застосовуємо принципи нормалізації, щоб переконатися, що всі таблиці відповідають 3NF.

Таблиця 3.2.4

Таблиця Пацієнт (нормалізація)

Patient
Patient_ID:int
Patient_FName:string
Patient_LName:string
Phone:int
Blood_Type:string
Email:string
Gender:string
Condition:string
Admission_Data:timestamp
Discharge_Data:timestamp

Відповідність 1NF: Таблиця відповідає першій нормальній формі (1NF), оскільки містить лише атомарні (неподільні) значення в кожному стовпчику, і немає груп, що повторюються.

Відповідність 2NF: Таблиця також відповідає другій нормальній формі (2NF), оскільки має первинний ключ «patient_id», який однозначно ідентифікує кожен рядок, а всі неключові атрибути повністю функціонально залежать від первинного ключа.

Відповідність 3NF: Таблиця відповідає Третій нормальній формі (3NF), оскільки не має транзитивних залежностей. Всі неключові атрибути безпосередньо залежать від первинного ключа «patient_id».

Функціональні залежності:

patient_id → patient_fname, patient_lname, Phone_no, blood_group, email, gender, condition, admission_date, discharge_date

Цей набір функціональних залежностей означає, що ім'я, номер телефону, група крові, електронна пошта, стать, стан, дата госпіталізації та дата виписки кожного пацієнта безпосередньо визначаються його унікальним ідентифікатором пацієнта. В результаті таблиця «Пацієнт» знаходиться у 3NF.

Таблиця 3.2.5

Таблиця Запис (нормалізація)

Appointment
App_ID:int
Schedules_on:Date
Data:Date
Time:timestamp
Doctor_ID:int
Patient_ID:int

Відповідність 1NF: Таблиця відповідає першій нормальній формі (1NF), оскільки містить атомарні значення в кожному стовпчику і не містить груп, що повторюються.

Відповідність 2NF: Таблиця відповідає другій нормальній формі (2NF), оскільки має первинний ключ «Appt_id», який однозначно ідентифікує кожен рядок, а всі неключові атрибути повністю функціонально залежать від первинного ключа.

Відповідність 3NF: Таблиця відповідає Третій нормальній формі (3NF), оскільки не містить транзитивних залежностей. Всі неключові атрибути безпосередньо залежать від первинних ключів «Appt_id», «Doctor_ID» та «Patient_id».

Функціональні залежності:

Appt_id → Scheduled_on, Date, Time, Doctor_ID, Patient_id
 Doctor_ID → Doctor_name, Doctor_specialization
 Patient_id → Patient_name, Patient_date_of_birth

Цей набір функціональних залежностей вказує на те, що для кожного запису, ідентифікованого «Appt_id», безпосередньо визначаються атрибути «Scheduled_on», «Date», «Time», «Doctor_ID» та «Patient_id». Немає жодних транзитивних залежностей, що гарантує, що таблиця «записів» добре структурована і відповідає принципам 3NF.

Примітка: Всі таблиці, що використовуються в Розділі 3 та Розділі 4, спочатку нормалізуються, а потім використовуються.

3.4 Оператори SQL

Цей звіт описує етап реалізації проекту створення бази даних, зосереджуючись на створенні бази даних, налаштуванні таблиць, наповненні даних та SQL-запитах. Цей проект використовує систему управління базами даних MySQL. Частина 1 - це створення бази даних, включаючи таблиці, всі інші структури, а також обмеження, тип і формат даних, Частина 2 - це розробка сценарію запиту та його реалізація, а також результати. Частина 3 - це висновок про те, що включено до цього звіту.

3.4.1 Створення бази даних за допомогою операторів SQL

3.4.1.1 Створення таблиці

Table Creation (Створення таблиць)

```
CREATE TABLE Staff (  
    Emp_ID INT NOT NULL,  
    Emp_FName VARCHAR(20) NOT NULL,  
    Emp_LName VARCHAR(20) NOT NULL,  
    Date_Joining DATE,  
    Date_Seperation DATE,  
    Emp_Type VARCHAR(15) NOT NULL,  
    Email VARCHAR(50),  
    Address VARCHAR(50) NOT NULL,  
    Dept_ID INT NOT NULL,  
    SSN INT NOT NULL,  
    PRIMARY KEY (Emp_ID),  
    FOREIGN KEY (Dept_ID) REFERENCES Department (Dept_ID)  
);
```

```
CREATE TABLE Doctor (  
    Doctor_ID INT NOT NULL,  
    Qualifications VARCHAR(15) NOT NULL,  
    Emp_ID INT NOT NULL,  
    Specialization VARCHAR(20) NOT NULL,
```

```

Dept_ID INT NOT NULL,
PRIMARY KEY (Doctor_ID),
FOREIGN KEY (Emp_ID) REFERENCES Staff (Emp_ID),
FOREIGN KEY (Dept_ID) REFERENCES Department (Dept_ID)
);

CREATE TABLE Nurse (
Nurse_ID INT NOT NULL,
Patient_ID INT NOT NULL,
Emp_ID INT NOT NULL,
Dept_ID INT NOT NULL,
PRIMARY KEY (Nurse_ID),
FOREIGN KEY (Patient_ID) REFERENCES Patient (Patient_ID),
FOREIGN KEY (Emp_ID) REFERENCES Staff (Emp_ID),
FOREIGN KEY (Dept_ID) REFERENCES Department (Dept_ID)
);

```

3.4.1.2 Стан бази даних

Щоб забезпечити заповнення бази даних для цілей тестування та розробки, в кожену таблицю були вставлені зразки фіктивних даних. Наступні записи були додані до кожної таблиці, щоб забезпечити узгодженість та достовірність даних. Тут буде показано лише частину даних, оскільки в кожній таблиці є кілька рядків.

Запис таблиці «Пацієнт»

Patient_ID	Patient_FName	Patient_LName	Phone	Blood_Type	Email	Gender	Condition_	Admission_Date	Discharge_Date
1	John	Doe	555-1234	A+	john.doe@email.com	Male	Injury	2023-01-01	2023-01-10
2	Jane	Smith	555-5678	O-	jane.smith@email.com	Female	Flu	2023-02-05	2023-02-15
3	Michael	Johnson	555-8765	B+	michael.johnson@email.com	Male	Allergies	2023-03-10	2023-03-20
4	Emily	Williams	555-2345	AB-	emily.williams@email.com	Female	Headache	2023-04-15	2023-04-25
5	Robert	Brown	555-5432	A-	robert.brown@email.com	Male	Fracture	2023-05-20	2023-05-30
6	Alice	Davis	555-7890	O+	alice.davis@email.com	Female	Respiratory Infection	2023-06-25	2023-07-05
7	Christopher	Miller	555-4321	B-	chris.miller@email.com	Male	Back Pain	2023-07-30	2023-08-09
8	Olivia	Jones	555-9876	AB+	olivia.jones@email.com	Female	Concussion	2023-09-04	2023-09-14
9	William	Wilson	555-8765	A+	william.wilson@email.com	Male	Appendicitis	2023-10-09	2023-10-19
10	Sophia	Moore	555-6543	O-	sophia.moore@email.com	Female	Diabetes	2023-11-14	2023-11-24

Рисунок 3.4.1 Інформація в таблиці “Пацієнт”.

Запис таблиці «Рахунок»

Bill_ID	Date	Room_Cost	Test_Cost	Other_Charges	M_Cost	Total	Patient_ID	Remaining_Balance	Policy_Number
60	2023-12-05	150.00	75.00	20.00	5.99	250.99	1	150.00	A123456
61	2023-12-06	200.00	50.00	25.00	12.75	287.75	2	200.00	B789012
62	2023-12-07	200.00	80.00	15.00	6.25	301.25	3	175.00	C345678
63	2023-12-08	250.00	60.00	20.00	9.75	339.75	4	180.00	D901234
64	2023-12-09	200.00	70.00	25.00	14.99	309.99	5	190.00	E567890
65	2023-12-10	250.00	65.00	20.00	22.50	357.50	6	220.00	F123789
66	2023-12-11	150.00	55.00	15.00	18.75	238.75	7	160.00	G456321
67	2023-12-12	200.00	60.00	20.00	12.75	292.75	8	200.00	H987654

Рисунок 3.4.2 Інформація в таблиці "Рахунок"

Розробка сценарію запиту

Запит_01: Керівництво лікарні хоче розрахувати загальний дохід, отриманий лікарнею, включаючи плату за палати, плату за лабораторні обстеження та інші різні платежі, за певний період часу.

```
SELECT DATE_FORMAT(b.Date, '%Y-%m-%d') AS Billing_Date,  
SUM(b.Room_Cost + b.Test_Cost + b.Other_Charges + b.M_Cost) AS Total_Revenue  
FROM bill b  
WHERE b.Date BETWEEN '2023-12-01' AND '2023-12-31'  
GROUP BY DATE_FORMAT(b.Date, '%Y-%m-%d')  
ORDER BY DATE_FORMAT(b.Date, '%Y-%m-%d');
```

Результат для запиту_01:

	Billing_Date	Total_Revenue
▶	2023-12-05	250.99
	2023-12-06	287.75
	2023-12-07	301.25
	2023-12-08	339.75
	2023-12-09	309.99
	2023-12-10	357.50
	2023-12-11	238.75
	2023-12-12	292.75
	2023-12-13	364.99
	2023-12-14	230.49

Рис. 3.4.3 Результат для запиту_01

3.5 Пропонований GUI

Графічний інтерфейс користувача (GUI) є важливим елементом будь-якої програми, який полегшує зручну взаємодію користувача з системою. У цьому розділі буде детально розглянуто графічний інтерфейс користувача "Pet",

зокрема його складові елементи, функціональні можливості та атрибути, які роблять його зручним та ефективним у користуванні.

Загальний огляд інтерфейсу:

Графічний інтерфейс "Pet" ретельно розроблено з використанням найсучасніших принципів дизайну інтерфейсів, що забезпечує як візуальну привабливість, так і практичне використання. Інтерфейс розроблений з використанням концепцій простоти і логічного групування частин, що дозволяє користувачам ефективно знаходити необхідні функції і без особливих зусиль освоювати систему.

На головному екрані (рисунок 3.5.1) розташовані основні елементи, такі як логотип ветеринарної клініки, назва додатку та контактні іконки для доступу до налаштувань і повідомлень. У верхній частині екрану знаходиться профіль користувача, який включає аватар, ім'я користувача, місто проживання та контактний номер телефону.

Основний функціонал додатку представлений у вигляді списку опцій, таких як запис на прийом до ветеринара, зв'язок з ветеринаром, створення списку справ, можливість придбати ліки та інформація про притулки для тварин. Кожна з цих опцій надає доступ до відповідних сервісів: інтеграція з календарем для бронювання візитів, чат або дзвінок з ветеринарним спеціалістом, створення та управління завданнями, пов'язаними з доглядом за тваринами, та інтеграція з онлайн-аптекою для замовлення ветеринарних препаратів.

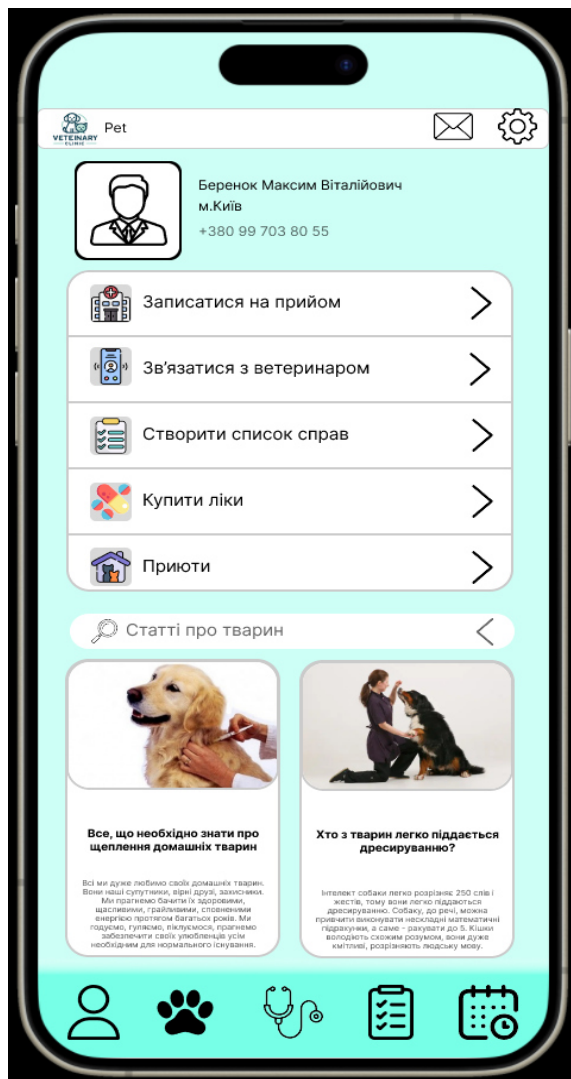


Рис. 3.5.1 Головне меню застосунку "Pet"

На рисунку 3.5.2 показаний профіль домашнього улюбленця в додатку "Pet". Верхня частина екрану містить кілька важливих елементів: у лівому верхньому куті знаходиться іконка улюбленця, поруч з якою розташована кнопка "Додати улюбленця". Праворуч від цієї кнопки знаходяться іконка для доступу до повідомлень та кнопка "Назад".

У центрі екрана розміщена фотографія собаки. Під фотографією наведено базову інформацію про тварину: стать (чоловіча), дата народження (15.05.2022), порода (пітбуль) і вага (20 кг). Над фотографією є мітка "Тварина застрахована".

Нижче інформаційного блоку розташована кнопка "Змінити дані", яка дозволяє редагувати інформацію про улюбленця.

Під цією кнопкою є дві основні функціональні кнопки: "Записи та нагадування" і "Рецепти". Перша кнопка надає доступ до записів про здоров'я та

нагадувань щодо догляду за твариною, друга - до рецептів та рекомендацій ветеринара.

Функціонал цього екрану дозволяє власникам домашніх тварин зручно переглядати та керувати інформацією про своїх улюбленців, редагувати дані, отримувати доступ до важливих записів та рецептів, а також додавати нових улюбленців до профілю.

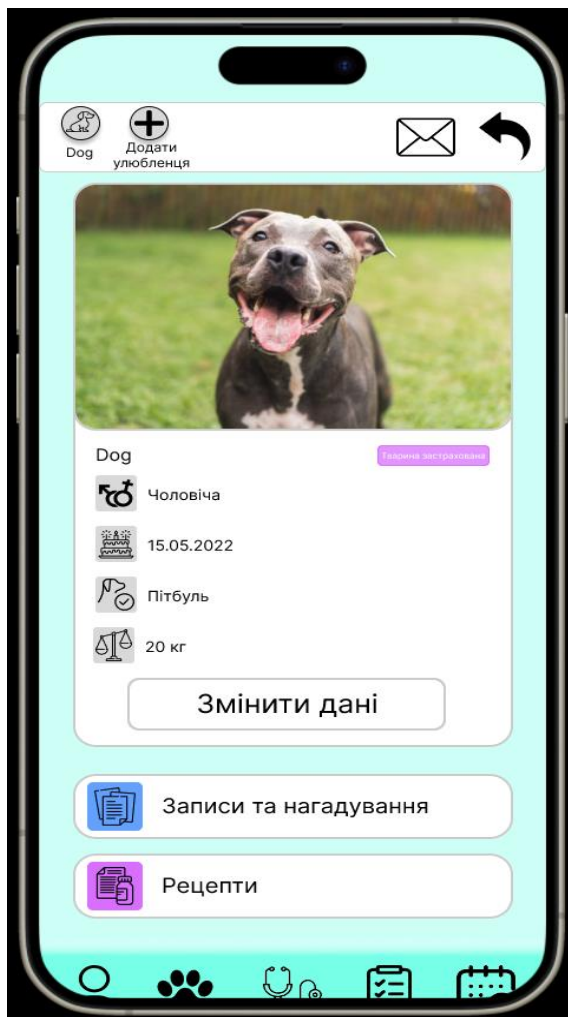


Рис. 3.5.2 Профіль тварини в застосунку "Pet"

На наступному рисунку 3.5.3 зображено інформацію про ветеринарну клініку в додатку "Pet". У верхній частині екрану знаходиться фотографія клініки. Під фотографією вказана назва клініки - "Ветеринарна клініка" та її адреса: вулиця Велика Васильківська, 63, Київ, 03150. Також є контактний номер телефону клініки з зеленою іконкою телефону для швидкого дзвінка.

Нижче розташовано інформаційний блок з послугами, які надає ця клініка. Заголовок "Цей заклад надає послуги:" підкреслює доступні опції. Послуги представлені у вигляді іконок з підписами:

- Прийом (синя іконка зі стетоскопом)
- Грумінг (червона іконка з ножицями)
- Притулити (фіолетова іконка з будиночком для тварин)
- Пошук (жовта іконка з лупою)

У нижній частині екрана знаходиться навігаційна панель додатку з п'ятьма іконками, що забезпечують швидкий доступ до різних розділів додатку: профіль користувача, сторінка тварин, консультації з ветеринаром, список справ та календар. Функціонал цього екрану дозволяє користувачам швидко знайти і зв'язатися з ветеринарною клінікою, дізнатися про доступні послуги, а також скористатися зручною навігацією для переходу до інших розділів додатку.

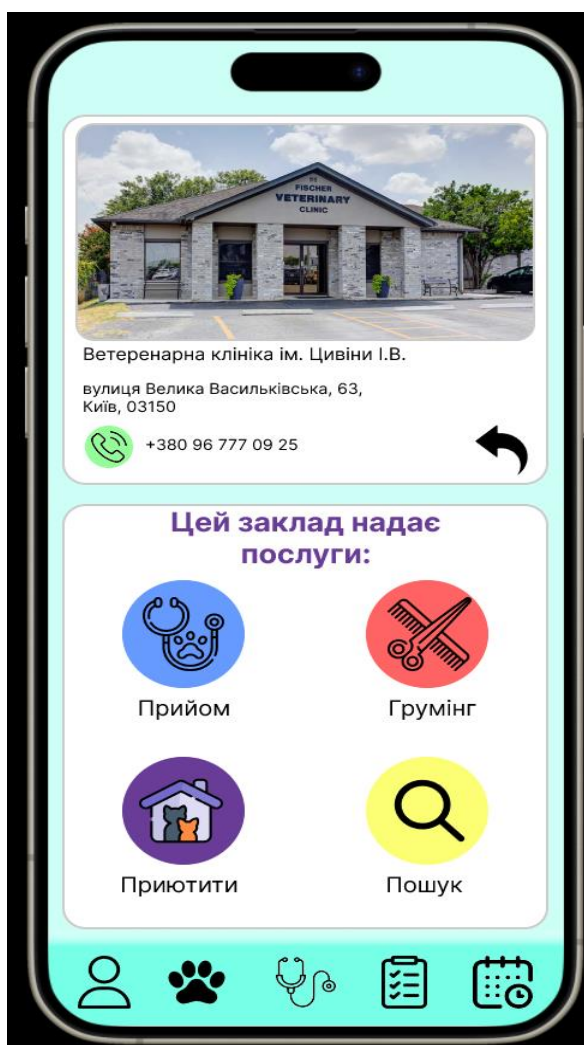


Рис. 3.5.3 Ветеринарна клініка в застосунку "Pet"

Висновки до розділу 3

Отже, ця модель даних була створена для того, щоб допомогти співробітникам лікарні зберігати інформацію та покращити доступ до неї, полегшивши процес пошуку. Управління базою даних лікарні має бути вдосконалене або модернізоване, щоб відповідати будь-якій ситуації. У цьому звіті розроблено реляційну схему бази даних системи управління лікарнею. Наша діаграма EER та пов'язана з нею реляційна схема показують концептуальний та логічний дизайн системи. Також визначені типи даних, припущення та обмеження для кожного атрибуту у відношеннях.

Наступним кроком є реалізація цієї бази даних і відповідна зміна дизайну. Розроблена система та її оцінка має бути проведена з метою вдосконалення системи баз даних та процесів управління в лікарнях. Вона здатна зберігати різноманітні та великі за обсягом бази даних. Крім того, ПЗ було розроблено з урахуванням програмних модулів для обробки інформації медичного центру, такої як дані про пацієнтів, управління постачанням, рахунки пацієнтів тощо. Таким чином, це програмне забезпечення містить файли баз даних пацієнтів, лікарів, медсестер та відділень лікарні і повинно надавати необхідну інформацію, яка буде сумісною, точною, гнучкою, захищеною та ефективною для виконання поставлених цілей.

4. ІНФОРМАЦІЙНА МОДЕЛЬ ПРОЕКТУ РОЗРОБКИ ЗАСТОСУНКУ PET

Розробка застосунку "Pet" є складним процесом, який вимагає чіткого планування та координації для досягнення успішного результату. Інформаційна модель проекту охоплює всі аспекти управління проектом, включаючи планування, управління часом, вартістю, комунікаціями та зацікавленими сторонами. Цей розділ детально розгляне ключові елементи управління проектом, які забезпечать ефективне виконання завдань та досягнення поставлених цілей. Використання сучасних методологій та інструментів дозволить оптимізувати процес розробки та забезпечити високу якість кінцевого продукту.

Для управлінням свого проекту "Pet" розглянемо ряд програмних рішень, які допоможуть ефективно планувати, координувати та контролювати всі аспекти проекту. Ось деякі з них, їх переваги та недоліки.

Для управлінням своїм проектом "Pet" було обрано програмне забезпечення Microsoft Project. Це один з найпотужніших і найвідоміших інструментів для управління проектами. Це рішення забезпечить комплексний підхід до планування, контролю та моніторингу проекту, що є критично важливим для успішної реалізації нашої ініціативи. Microsoft Project пропонує широкий спектр можливостей, які допоможуть ефективно організувати роботу команди, розподілити ресурси та контролювати витрати.

4.1. План управління проектом

План управління проектом – це комплексний документ, який визначає, яким чином буде виконуватися, контролюватися і завершуватися проект. Це один із ключових документів у проектному менеджменті, який слугує дорожньою картою для всіх членів команди і забезпечує узгодженість у всіх аспектах проекту.

Основними компонентами (складовими) плану управління проектом є:

1. Опис Проекту

- Визначення мети проекту та його ключових завдань.
 - Опис очікуваних результатів та продуктів проекту.
2. **Обсяг Проекту**
- Визначення меж проекту, включаючи те, що буде включено і виключено з проекту.
 - Список основних задач та етапів проекту.
3. **Розклад Проекту**
- Графік виконання задач з зазначенням початкових та кінцевих дат.
 - Визначення віх проекту для контролю прогресу.
4. **Управління Ресурсами**
- Планування ресурсів, включаючи персонал, обладнання та матеріали.
 - Опис ролей та відповідальностей членів команди.
5. **Бюджет Проекту**
- Оцінка витрат на виконання проекту.
 - Контроль та управління бюджетом для запобігання перевитратам.
6. **Управління Ризиками**
- Виявлення можливих ризиків для проекту.
 - Розробка плану дій для запобігання та мінімізації ризиків.
7. **Забезпечення Якості**
- Визначення стандартів якості для продуктів проекту.
 - Планування заходів для забезпечення відповідності продукту вимогам якості.
8. **Комунікаційний План**
- Визначення каналів та методів комунікації між членами команди та іншими стейкхолдерами.
 - Регулярні звіти про прогрес та статус проекту.

Оскільки більшість з цих компонентів можна легко виконати, використовуючи Microsoft Project, то можна розпочати створювати дорожню карту мого проекту.

Слід розпочати з роз'яснень важливих аспектів під час побудови проєкту. Фази реалізації, віхи та задачі є основними компонентами управління проєктом. Вони відіграють критично важливу роль у плануванні, виконанні, контролі та завершенні проєктів.

Розберемось, що таке фази реалізації, віхи та задачі?

Фаза реалізації проєкту — це етап у життєвому циклі проєкту, під час якого розробляються та впроваджуються всі заплановані дії та задачі з метою досягнення кінцевих результатів проєкту. Цей етап включає виконання всіх технічних, управлінських та організаційних завдань, необхідних для досягнення визначених цілей проєкту.

Фаза реалізації є ключовою, оскільки саме на цьому етапі проєктна команда перетворює плани, створені на попередніх етапах, у конкретні продукти, послуги або результати. Основними завданнями цієї фази є управління ресурсами, виконання робіт згідно з графіком, контроль якості, вирішення проблем і внесення необхідних коректив для забезпечення успішного завершення проєкту.

Віха - це ключова подія або важлива точка в проєкті, яка позначає завершення певного етапу або досягнення важливого результату. Віхи допомагають відслідковувати прогрес проєкту та можуть використовуватись як контрольні точки. Віха не має тривалості (тобто тривалість віхи дорівнює нулю), а служить лише для позначення важливих моментів у графіку проєкту.

Задача - це конкретне завдання або частина роботи, яка повинна бути виконана в рамках проєкту. Задачі мають тривалість, ресурси, що їх виконують, і можуть мати залежності від інших задач. Задачі складають основу плану проєкту та допомагають структурувати і організувати всю необхідну роботу для досягнення кінцевих цілей проєкту.

Задачі складають основний робочий план проєкту і можуть бути пов'язані між собою залежностями (наприклад, одна задача не може початися, поки не завершиться інша). Віхи, в свою чергу, використовуються для маркування завершення важливих етапів або досягнення ключових результатів, до яких проєкт прямує через виконання задач.

Розпочнемо з введення фаз, віх та задач для проекту "Pet". Для кожної фази проекту вкажемо відповідні задачі та їх тривалість.

Режим завдання	Task Name	Тривалість	Крайній термін	Початок	Завершення
	Створення застосунку	670,5 днів	НД	Сб 01.06.24	Чт 31.12.26
	Початок проекту	0 днів	НД	Сб 01.06.24	Сб 01.06.24
	Визначення проекту	35 днів	НД	Пн 03.06.24	Пт 19.07.24
	Проектування	54 днів	НД	Пн 22.07.24	Чт 03.10.24
	Ітерація 1	200 днів	НД	Пт 04.10.24	Пн 14.07.25
	Ітерація 2	160 днів	НД	Пн 14.07.25	Вт 24.02.26
	Навчання	19 днів	НД	Вт 24.02.26	Пн 23.03.26
	Впровадження	11 днів	НД	Пн 23.03.26	Вт 07.04.26
	Закриття роботи над продуктом	18 днів	НД	Вт 07.04.26	Пт 01.05.26
	Завершення проекту	0 днів	НД	Чт 31.12.26	Чт 31.12.26

Рис.4.1.1 Фази проекту "Pet"

Перша фаза: **Визначення проекту** (35 днів) див. рис 4.1.2. Цей етап охоплює початкове планування та узгодження основних аспектів проекту. Завдання включають проведення початкових сесій щодо визначення обсягу проекту та збору вимог із зацікавленими сторонами (8 днів), визначення цілей проекту, обсягу і результатів (7 днів), створення статуту проекту або початкового документа (10 днів), визначення ключових зацікавлених сторін проекту (5 днів) та отримання схвалення обсягу та статуту проекту від відповідних зацікавлених сторін (5 днів). Визначення проекту є критичним етапом, що забезпечує узгодження між усіма зацікавленими сторонами та створює основу для подальшого планування та виконання проекту.

Також потрібно звернути увагу на те, що в саме ПЗ Microsoft Project дозволяє робити примітки на створених задачах.

Для прикладу візьмемо одну з задач першої фази "Створення статуту проекту або початковий документ" рис 4.1.1.

Режим завдання	Task Name	Тривалість	Крайній термін	Початок	Завершення
	Створення застосунку	670,5 днів	НД	Сб 01.06.24	Чт 31.12.26
	Початок проекту	0 днів	НД	Сб 01.06.24	Сб 01.06.24
	Визначення проекту	35 днів	НД	Пн 03.06.24	Пт 19.07.24
	Проведення початкової сесії щодо визначення обсягу проекту та збору вимог із зацікавленими сторонами	8 днів	НД	Пн 03.06.24	Ср 12.06.24
	Визначення цілі проекту, обсяг і результати проекту	7 днів	НД	Чт 13.06.24	Пт 21.06.24
	Створення статуту проекту або початковий документ	10 днів	НД	Пн 24.06.24	Пт 05.07.24
	Визначення ключових зацікавлених сторін проекту	5 днів	НД	Пн 08.07.24	Пт 12.07.24
	Отримання схвалення обсягу та статуту проекту від відповідних зацікавлених сторін	5 днів	НД	Пн 15.07.24	Пт 19.07.24

Рис 4.1.2 Фаза "Визначення проекту" для проекту "Pet"

Як можна побачити ліворуч від обраної задачі з'явилась примітка. Яку можна відкрити на ознайомитися в разі необхідності. У самій примітці описаний статут проекту "Pet" див рис 4.1.1.

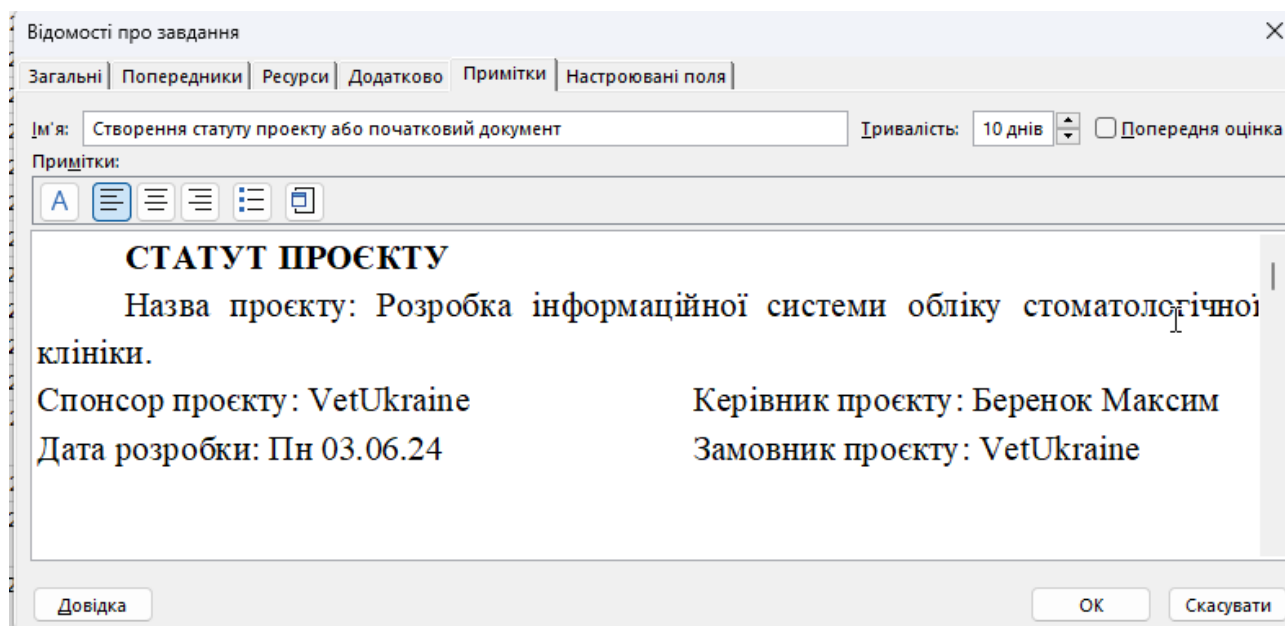


Рис 4.1.3 Статут проекту "Pet"

Друга фаза: Проектування (54 дні). Цей етап деталізує технічні аспекти та розробляє конкретні плани для виконання проекту. Завдання включають узгодження обсягу проекту (7 днів), розробку плану ресурсів (7 днів), розробку плану якості (7 днів), узгодження бюджету проекту (9 днів), розробку концепції та архітектури (30 днів), проектування інтерфейсу користувача (10 днів), визначення технічних вимог (10 днів) та розробку плану управління проектом (10 днів). Остання задача завершення етапу проектування відбувається без тривалості (фіха), що служить контрольним пунктом для переходу до наступного етапу. Етап проектування є ключовим для забезпечення детального планування та підготовки до реалізації проекту, що забезпечує чітке розуміння ресурсів, бюджету, якості та технічних вимог, необхідних для успішного виконання проекту.

	➤	4 Створення застосунку	496 днів	НД	Сб 01.06.24	Чт 30.04.26
	➤	Початок проекту	0 днів	НД	Сб 01.06.24	Сб 01.06.24
	➤	4 Визначення проекту	35 днів	НД	Пн 03.06.24	Пт 19.07.24
	➤	Проведення початкової сесії щодо визначення обсягу проекту та збору вимог із зацікавленими сторонами	8 днів	НД	Пн 03.06.24	Ср 12.06.24
	➤	Визначення цілі проекту, обсяг і результати проекту	7 днів	НД	Чт 13.06.24	Пт 21.06.24
	➤	Створення статуту проекту або початковий документ	10 днів	НД	Пн 24.06.24	Пт 05.07.24
	➤	Визначення ключових зацікавлених сторін проекту	5 днів	НД	Пн 08.07.24	Пт 12.07.24
	➤	Отримання схвалення обсягу та статуту проекту від відповідних зацікавлених сторін	5 днів	НД	Пн 15.07.24	Пт 19.07.24
	➤	4 Проектування	54 днів	НД	Пн 22.07.24	Чт 03.10.24
	➤	Узгодження обсягу проекту	7 днів	НД	Пн 22.07.24	Вт 30.07.24
	➤	Розробка плану ресурсів	7 днів	НД	Ср 31.07.24	Чт 08.08.24
	➤	Розробка плану якості	7 днів	НД	Пт 09.08.24	Пн 19.08.24
	➤	Узгодження бюджету проекту	9 днів	НД	Пт 09.08.24	Ср 21.08.24
	➤	Розробка концепції та архітектури	30 днів	НД	Пт 09.08.24	Чт 19.09.24
	➤	Проектування інтерфейсу користувача	10 днів	НД	Пт 09.08.24	Чт 22.08.24
	➤	Визначення технічних вимог	10 днів	НД	Пт 09.08.24	Чт 22.08.24
	➤	Розробка плану управління проектом	10 днів	НД	Пт 20.09.24	Чт 03.10.24
	➤	Проектування завершено	0 днів	НД	Чт 03.10.24	Чт 03.10.24

Рис 4.1.3 Фаза "Проектування" для проекту "Pet"

У розділі два "Аналіз і вибір методу управління проектом розробки застосунку" було обрано методологію Agile. Використання Agile дозволяє розбити проект на ітерації, кожна з яких включає всі етапи розробки, від аналізу вимог до тестування. Це забезпечує гнучкість і адаптивність процесу розробки, дозволяє швидко реагувати на зміни вимог та постійно вдосконалювати продукт на основі зворотного зв'язку від користувачів. Вибір Agile також сприяє тісній співпраці між командою розробників та зацікавленими сторонами, що допомагає забезпечити високу якість кінцевого продукту та задоволення потреб користувачів.

Розпочнемо з **першої ітерації**. Вона включає в себе: аналіз вимог, створення UI/UX-дизайну, розробку та тестування.

Аналіз/вимоги до програмного забезпечення триватиме 30 днів. На цьому етапі будуть виконані наступні завдання: проведення аналізу потреб (3 дні), розробка попередніх специфікацій програмного забезпечення (10 днів), розробка попереднього бюджету (2 дні), перегляд специфікацій і бюджету програмного забезпечення з командою (4 дні), отримання зворотного зв'язку щодо специфікацій програмного забезпечення (3 дні), розробка графіку роботи (2 дні), отримання схвалення для продовження роботи (3 дні) та забезпечення необхідних ресурсів (3 дні). Завершення аналізу позначається як нульовий день.

Створення UI/UX-дизайну займе 52 дні. Завдання включають проведення дослідження та аналіз користувачів (3 дні), розробку каркасів та прототипів для ключових екранів і функцій (5 днів), проведення воркшопів з користувачами для збору відгуків (20 днів), створення високоякісних макетів дизайну (15 днів), розробку інструкції щодо інтерфейсу користувача та UX, стандарти дизайну для узгодженості (5 днів), закінчення розробки початкового UI/UX-дизайну (2 дні) та отримання дозволу на продовження (2 дні). Завершення розробки дизайну позначається як нульовий день, тому це рахується віхою.

Розробка триватиме 56 днів. Завдання включають налаштування середовища розробки, включаючи засоби контролю версій і управління проектами (10 днів), розробку базової серверної інфраструктури, включаючи бази даних, початкові API та архітектуру сервера (5 днів), розширення серверної інфраструктури (2 дні), написання коду для ключових функцій користувача інтерфейсу (30 днів), інтеграцію сторонніх служб і API (9 днів) і віхою цього буде завершення розробки (0 днів).

Тестування займе 62 дні. На цьому етапі будуть виконані наступні завдання: розробка планів модульного тестування на основі специфікацій продукту (10 днів), розробка планів інтеграційних тестів на основі специфікацій продукту (4 дні), модульне тестування (29 днів), що включає перегляд модульного коду (7 днів), тестування модулів компонентів на відповідність специфікаціям продукту (5 днів), виявлення аномалій у специфікаціях продукту (2 дні), модифікацію коду (10 днів) та повторне тестування зміненого коду (5 днів) та завершення модульного тестування (0 днів). Тестування інтеграції триватиме 23 дні, включаючи тестування інтеграції модулів (5 днів), виявлення аномалій до специфікацій (3 дні), модифікацію коду (10 днів) та повторне тестування модифікованого коду (5 днів) та завершення тестування інтеграції та загального тестування (0 днів) це дві віхи, які позначають кінець тестувань.

Друга ітерація проекту включає в себе: створення UI/UX-дизайну, розробку та тестування.

Створення UI/UX-дизайну триватиме 33 дні. На цьому етапі будуть виконані наступні завдання: допрацювання дизайну на основі зворотного зв'язку

користувачів (10 днів), уточнення каркасів та прототипів для нових функцій (5 днів), адаптація високоякісних макетів дизайну до нових вимог (7 днів), оновлення інструкцій та стандартів дизайну (6 днів), фінальне коригування дизайну (3 дні) та перевірка узгодженості дизайну для всіх екранних розмірів та пристроїв (2 дні). Завершення створення дизайну (0 днів).

Розробка триватиме 65 днів. Завдання включають початкову інтеграцію клієнтської та серверної частин (15 днів), завершення програмування всіх функцій застосунку (25 днів), остаточну інтеграцію всіх компонентів системи (10 днів) та оптимізацію продуктивності та безпеки застосунку (15 днів). Завершення розробки (0 днів)

Тестування займе 62 дні. На цьому етапі будуть виконані наступні завдання: розробка планів модульного тестування на основі специфікацій продукту (10 днів), розробка планів інтеграційних тестів на основі специфікацій продукту (12 днів), модульне тестування (29 днів), що включає перегляд модульного коду (7 днів), тестування модулів компонентів на відповідність специфікаціям продукту (5 днів), виявлення аномалій у специфікаціях продукту (2 дні), модифікацію коду (10 днів) та повторне тестування зміненого коду (5 днів). Завершення модульного тестування позначається як нульовий день. Тестування інтеграції триватиме 23 дні, включаючи тестування інтеграції модулів (5 днів), виявлення аномалій до специфікацій (3 дні), модифікацію коду (10 днів) та повторне тестування модифікованого коду (5 днів). Завершення тестування інтеграції та загального тестування (0 днів) це дві віхи, які позначають кінець тестувань.

П'ята фаза: Навчання (19 днів): Ця фаза охоплює підготовку та проведення навчання для кінцевих користувачів та персоналу служби підтримки, що забезпечить ефективне використання розробленого застосунку.

Розробка специфікацій навчання для кінцевих користувачів триватиме 4 дні, а для персоналу служби підтримки - ще 4 дні. Потім, за 3 дні, буде визначено методику проведення навчання, яка може включати комп'ютерне навчання, аудиторні заняття тощо. Розробка навчальних матеріалів займе 1 день. Після цього проведуть юзабіліті дослідження тренінгу (3 дні), щоб перевірити

ефективність навчальних матеріалів. На основі результатів дослідження протягом 1 дня відбудеться доопрацювання навчальних матеріалів. Завершальним етапом стане розробка механізму проведення тренінгу (3 дні), після чого навчальні матеріали будуть готові.

Шоста фаза: Впровадження (11 днів): Ця фаза охоплює підготовку та впровадження застосунку у виробниче середовище.

Проведення підготовчих робіт для встановлення застосунку у маркетах триватиме 2 дні. Після цього за 3 дні буде підготовлено та розповсюджено документацію щодо використання застосунку. Запуск рекламної кампанії для сповіщення клієнтів про наявність нового застосунку займе 4 дні. Викладення програми у маркети триватиме 2 дні. І віха Завершення впровадження (0 днів).

Сьома фаза: Закриття роботи над продуктом (18 днів): Ця фаза охоплює завершення проекту та документування отриманих результатів.

Проведення заключної зустрічі із зацікавленими сторонами займе 3 дні. Після цього протягом 5 днів буде задокументовано уроки, отримані під час проекту. Завершення та архівування всієї проектної документації займе ще 5 днів. Заключним етапом стане закриття облікових записів проекту (5 днів).

План управління проектом передбачає успішне виконання всіх фаз реалізації проекту, включаючи визначення проекту, аналіз вимог, розробку, тестування, навчання, впровадження та закриття роботи над продуктом. Завершення проекту заплановано на 31 грудня 2026 року, в той час як остання віха завершується 1 травня 2026 року. Це забезпечує значний запас часу для вирішення будь-яких непередбачених проблем і гарантує, що всі задачі будуть виконані вчасно. Такий підхід до управління проектом підвищує шанси на успішну реалізацію проекту, задовольняючи всі вимоги та очікування зацікавлених сторін.

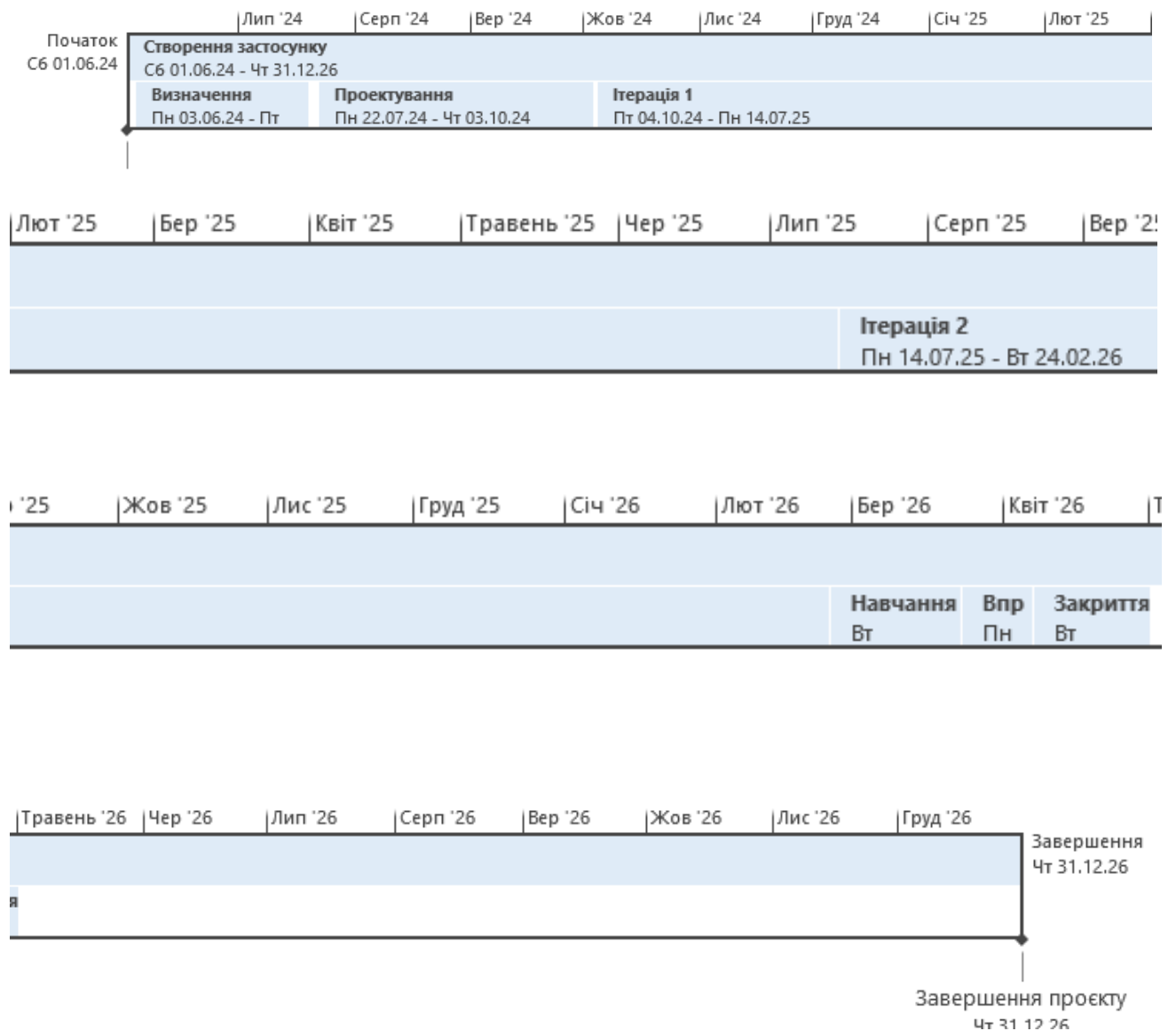


Рис 4.1.4 Календар роботи для проекту "Pet"

4.2. Управління розподілом ресурсів

Розподіл ресурсів — це процес призначення і використання робочих ресурсів, матеріальних ресурсів і обладнання для виконання завдань в проекті. В Microsoft Project можливо встановлювати ресурси для кожного завдання, визначати їх робочий час, призначати пріоритети та відстежувати використання ресурсів у часі.

Через інтерфейс Microsoft Project також можна легко переглядати, редагувати і видаляти ресурси, а також встановлювати їхні властивості. Крім того, програма дозволяє створювати різні види звітів і діаграм, що відображають використання ресурсів в проекті.

Розподіл ресурсів в Microsoft Project допомагає забезпечити ефективне використання ресурсів, уникнення перевантаження ресурсів та вчасне виконання завдань в рамках проекту.

Для того, щоб створити та вписати всі необхідні дані потрібно перейти у вкладку "Аркуш ресурсів" див рис 4.2.1

№	Режим завдання	Task Name	Тривалість	Крайній термін	Початок	Завершення	Попередники
0		Створення застосунку	670,5 днів	НД	Сб 01.06.24	Чт 31.12.26	
1		Початок проекту	0 днів	НД	Сб 01.06.24	Сб 01.06.24	
2		Визначення проекту	35 днів	НД	Пн 03.06.24	Пт 19.07.24	
3		Проведення початкової сесії щодо визначення обсягу проекту та збору вимог із зацікавленими сторонами	8 днів	НД	Пн 03.06.24	Ср 12.06.24	
4		Визначення цілі проекту, обсяг і результати проекту	7 днів	НД	Чт 13.06.24	Пт 21.06.24	3
5		Створення статуту проекту або початкової документації	10 днів	НД	Пн 24.06.24	Пт 05.07.24	4
6		Визначення ключових зацікавлених сторін проекту	5 днів	НД	Пн 08.07.24	Пт 12.07.24	5
7		Отримання схвалення обсягу та статуту проекту від відповідних зацікавлених сторін	5 днів	НД	Пн 15.07.24	Пт 19.07.24	6
		Проекткування	54 днів	НД	Пн 22.07.24	Чт 03.10.24	
		Узгодження обсягу проекту	7 днів	НД	Пн 22.07.24	Вт 30.07.24	7
		Розробка плану ресурсів	7 днів	НД	Ср 31.07.24	Чт 08.08.24	9
		Розробка плану якості	7 днів	НД	Пт 09.08.24	Пн 19.08.24	10
		Узгодження бюджету проекту	9 днів	НД	Пт 09.08.24	Ср 21.08.24	10
		Розробка концепції та архітектури	30 днів	НД	Пт 09.08.24	Чт 19.09.24	10
		Проекткування інтерфейсу користувача	10 днів	НД	Пт 09.08.24	Чт 22.08.24	10
		Визначення технічних вимог	10 днів	НД	Пт 09.08.24	Чт 22.08.24	10
		Розробка плану управління проектом	10 днів	НД	Пт 20.09.24	Чт 03.10.24	10;11;12;13;14;
		Проекткування завершено	0 днів	НД	Чт 03.10.24	Чт 03.10.24	16
		Ітерація 1	200 днів	НД	Пт 04.10.24	Пн 14.07.25	
		Аналіз/вимоги до програмного забезпечення	30 днів	НД	Пт 04.10.24	Чт 14.11.24	
		Проведення аналізу потреб	3 днів	НД	Пт 04.10.24	Вт 08.10.24	16
		Розробка попередніх специфікацій програмного забезпечення	10 днів	НД	Ср 09.10.24	Вт 22.10.24	20
		Розробка попереднього бюджету	2 днів	НД	Ср 23.10.24	Чт 24.10.24	21
		Перегляд специфікацій/бюджету програмного забезпечення з командою	4 днів	НД	Пт 25.10.24	Ср 30.10.24	22
		Отримання зворотного зв'язку щодо специфікацій програмного забезпечення	3 днів	НД	Чт 31.10.24	Пн 04.11.24	23
		Розробити графік роботи	2 днів	НД	Вт 05.11.24	Сб 06.11.24	24

Рис 4.2.1 Аркуш ресурсів в Microsoft Project для проекту "Pet"

На даній сторінці можливо побачити такі стовпчики: ім'я ресурсу, тип ресурсу, одиниці виміру, група ресурсів, базові календарі, вартість за одиницю ресурсу, об'єм роботи, вартість ресурсів за завдання та додаткові поля (див рис 4.2.2). Також можливо додавати або видаляти стовпчики відповідно до потреб проекту, щоб відображати різні аспекти ресурсів, такі як розміщення, контактна інформація тощо.

Ім'я ресурсу	Тип	Одиниця виміру/матеріалів	Ініціали	РБС	Макс. одиниць	Звич. ставка
Понад. ставка	Витрати/використань	Нарахування	Основний календар	Бюджет		

Рис 4.2.2 Поля в Аркуші ресурсів в Microsoft Project для проекту "Pet"

Надалі слід розглянути ресурси, які будуть застосовані у проекті та описати

їх.

В роботі буде використаний **Resource breakdown structure (далі RBS)** — ієрархічний перелік ресурсів, пов'язаних за функцією та типом ресурсів, який використовується для полегшення планування та контролю роботи над проектом.

"Керівник проекту" є типу "Робота" і відповідає за керування проектом. Одиниця вимірювання матеріалів в даному випадку не вказана, оскільки це робочий ресурс. Ресурс віднесений до категорії "Спеціалісти". Максимальна одиниця ресурсного використання - 100%. Звичайна ставка складає 50 000,00 € на місяць, а понад норму ставка - 50 500,00 € на місяць. Нарахування здійснюється пропорційно використанню ресурсу. Основний календар для цього ресурсу є "Standard".

Проектний менеджер відноситься до типу "Робота" та відповідає за керування проектом. Так само, як і керівник проекту, він класифікується як "Спеціалісти". Максимальна одиниця ресурсного використання - 100%. Звичайна ставка для нього становить 30 000,00 € на місяць, а понаднормативна - 30 500,00 € на місяць. Нарахування здійснюється пропорційно використанню ресурсу, а основний календар - "Standard".

Бізнес-аналітик, також типу "Робота", відноситься до "Спеціалістів" та відповідає за аналіз бізнес-потреб. Його максимальне ресурсне використання також становить 100%. Звичайна ставка - 25 000,00 € на місяць, понаднормативна - 25 500,00 € на місяць. Нарахування проводиться пропорційно використанню ресурсу, а основний календар - "Standard".

Архітектор ПЗ також відноситься до типу "Робота" та класифікується як "Спеціалісти". Максимальна одиниця ресурсного використання - 100%. Звичайна ставка - 40 000,00 € на місяць, а понаднормативна - 40 500,00 € на місяць. Нарахування здійснюється пропорційно використанню ресурсу, а основний календар - "Standard".

Команда UI/UX дизайнерів, команда розробників та команда QA інженерів також є типу "Робота" та відносяться до категорії "Спеціалісти". Їх максимальне ресурсне використання становить 300%, 400% та 400% відповідно. Звичайні ставки для цих команд складають 20 000,00 € на місяць, а понаднормативні - 20

500,00 € на місяць. Нарахування проводиться пропорційно використанню ресурсу, а основний календар - "Standard".

Менеджер з впровадження, менеджер з управління ризиками, менеджер з фінансів, менеджер з управління ресурсами та менеджер з управління якістю також відносяться до типу "Робота" та класифікуються як "Спеціалісти". Для них максимальна одиниця ресурсного використання також становить 100%. Звичайні ставки складають 25 000,00 € на місяць, а понаднормативні - 25 500,00 € на місяць. Нарахування здійснюється пропорційно використанню ресурсу, а основний календар - "Standard".

Технічна підтримка є також типу "Робота" та класифікується як "Спеціалісти". Її максимальне ресурсне використання становить 500%. Звичайна ставка складає 15 000,00 € на місяць, а понаднормативна - 15 500,00 € на місяць. Нарахування проводиться пропорційно використанню ресурсу, а основний календар - "Standard".

Сервери для розробки та тестування, робочі станції та папір, ручки та інше є типу "Матеріал" і мають одиницю вимірювання "шт.". Нарахування здійснюється пропорційно використанню ресурсу. Ресурси були описані в "Аркуш ресурсів" та використані в проекті див рис 4.2.3 та рис 4.2.4

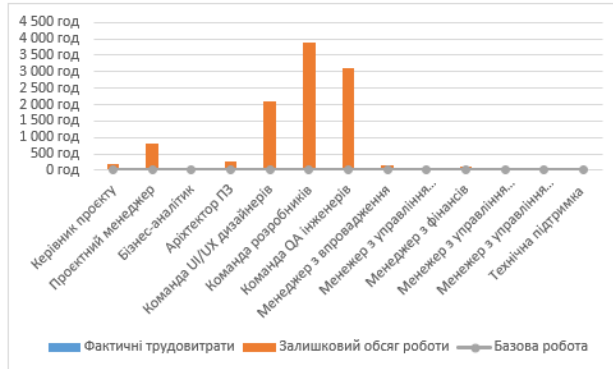
Ім'я ресурсу	Тип	Одиниця вимірювання матеріалу	Ініціали	РБС	Макс. одиниці	Звич. ставка	Понад. ставка	Витрати/використання	Нарахування	Основний календар
Керівник проекту	Робота		К	Спеціалісти	100%	50 000,00 €/місяць	50 500,00 €/місяць	0,00 €	Пропорційне	Standard
Проектний менеджер	Робота		П	Спеціалісти	100%	30 000,00 €/місяць	30 500,00 €/місяць	0,00 €	Пропорційне	Standard
Бізнес-аналітик	Робота		Б	Спеціалісти	100%	25 000,00 €/місяць	25 500,00 €/місяць	0,00 €	Пропорційне	Standard
Архітектор ПЗ	Робота		А	Спеціалісти	100%	40 000,00 €/місяць	40 500,00 €/місяць	0,00 €	Пропорційне	Standard
Команда UI/UX дизайнерів	Робота		К	Спеціалісти	300%	20 000,00 €/місяць	20 500,00 €/місяць	0,00 €	Пропорційне	Standard
Команда розробників	Робота		К	Спеціалісти	400%	25 000,00 €/місяць	25 500,00 €/місяць	0,00 €	Пропорційне	Standard
Команда QA інженерів	Робота		К	Спеціалісти	400%	20 000,00 €/місяць	20 500,00 €/місяць	0,00 €	Пропорційне	Standard
Менеджер з впровадження	Робота		М	Спеціалісти	100%	25 000,00 €/місяць	25 500,00 €/місяць	0,00 €	Пропорційне	Standard
Менеджер з управління ризиками	Робота		М	Спеціалісти	100%	20 000,00 €/місяць	20 500,00 €/місяць	0,00 €	Пропорційне	Standard
Менеджер з фінансів	Робота		М	Спеціалісти	100%	20 000,00 €/місяць	20 500,00 €/місяць	0,00 €	Пропорційне	Standard
Менеджер з управління ресурсів	Робота		М	Спеціалісти	100%	20 000,00 €/місяць	20 500,00 €/місяць	0,00 €	Пропорційне	Standard
Менеджер з управління якістю	Робота		М	Спеціалісти	100%	20 000,00 €/місяць	20 500,00 €/місяць	0,00 €	Пропорційне	Standard
Технічна підтримка	Робота		Т	Спеціалісти	500%	15 000,00 €/місяць	15 500,00 €/місяць	0,00 €	Пропорційне	Standard
Бюджет проекту	Витрати		Б	Бюджет					Пропорційне	
Сервери для розробки та тестування	Матеріал	шт.	С	Обладнання		12 000,00 €		0,00 €	Пропорційне	
Робочі станції	Матеріал	шт.	Р	Обладнання		15 000,00 €		0,00 €	Пропорційне	
Папір, ручки та інше	Матеріал	шт.	П	Обладнання		3 000,00 €		0,00 €	Пропорційне	

Рис 4.2.3 Доступні ресурси в Microsoft Project для проекту "Pet"

ОГЛЯД РЕСУРСІВ

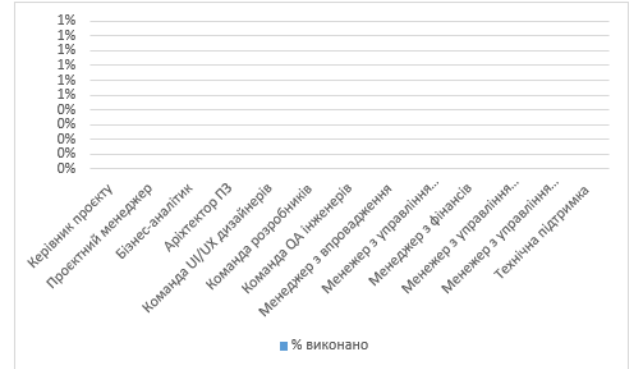
СТАТИСТИКА ДЛЯ РЕСУРСІВ

Стан трудовитрат для всіх трудових ресурсів.



СТАН ТРУДОВИТРАТ

Відсоток роботи, виконаної всіма трудовими ресурсами.



СТАН РЕСУРСУ

Залишковий обсяг роботи для всіх трудових ресурсів.

Ім'я	Початок	Завершення	Залишковий обсяг роботи
Керівник проекту	Пн 03.06.24	Пт 10.04.26	176 год
Проектний менеджер	Пн 03.06.24	Пт 01.05.26	824 год
Бізнес-аналітик	Пн 24.06.24	Пт 05.07.24	40 год
Архітектор ПЗ	Пт 09.08.24	Ср 06.11.24	288 год
Команда UI/UX дизайнерів	Пт 09.08.24	Чт 28.08.25	2 088,02 год
Команда розробників	Ср 29.01.25	Вт 07.04.26	3 904 год
Команда QA інженерів	Чт 17.04.25	Вт 24.02.26	3 104 год
Менеджер з впровадження	Вт 24.02.26	Пт 17.04.26	167,6 год
Менежер з управління ризиками	НД	НД	0 год
Менежер з фінансів	Пт 09.08.24	Ср 30.10.24	120 год
Менежер з управління ресурсів	Ср 31.07.24	Чт 08.08.24	56 год
Менежер з управління якості	Пт 09.08.24	Пн 19.08.24	56 год

Рис 4.2.4 Звіт по огляду ресурсів в Microsoft Project для проекту "Pet"

Управління розподілу ресурсів — це складний процес, який вимагає уважного планування і ефективного використання доступних ресурсів. Вищезгадані ресурси включають робочі зусилля спеціалістів, матеріали та бюджети. Цей підрозділ надав детальну інформацію про кожен ресурс, включаючи його тип, одиницю вимірювання, максимальні та звичайні ставки, нарахування та основний календар. Він допомагає забезпечити ефективне управління проектом шляхом збалансованого використання ресурсів і забезпечення їх належної доступності на кожному етапі. Такий підхід дозволяє оптимізувати витрати, підтримувати високу продуктивність команди та досягати успішних результатів у проекті.

4.3. Управління вартістю проекту

Управління вартістю проекту - це ключовий аспект успішного управління будь-яким проектом. Воно включає в себе планування, контроль і оптимізацію витрат для забезпечення відповідності бюджету проекту. Основні етапи управління вартістю проекту включають установлення бюджету, відстеження витрат, ідентифікацію ризиків та пошук можливостей з економії коштів. Крім того, ефективне управління вартістю проекту передбачає постійний моніторинг фінансових показників і вчасні коригування стратегії, якщо це необхідно. Послідовний підхід до управління вартістю проекту допомагає забезпечити фінансову стабільність і досягнення поставлених цілей проекту.

Планування бюджету повинно бути прозорим і деталізованим, щоб уникнути непорозумінь і недоліків у майбутньому. Крім того, слід враховувати можливі ризики та шукати способи їх зменшення або управління, щоб уникнути перевищення бюджету.

Microsoft Project пропонує широкий спектр використання звітів. Тому в цьому підрозділі будуть проаналізовані звіти на рахунок витрат проекту "Pet".

Звіт "**Огляд витрат на завдання**" містить детальну інформацію про витрати, пов'язані з різними етапами та завданнями проекту. Він включає дані про фактичні витрати, базову вартість, відхилення витрат та залишкові витрати. Основна мета звіту – надати прозору картину фінансових витрат на кожному етапі проекту та виявити відхилення від запланованого бюджету.

Стан витрат: Графік показує розподіл витрат на проект, включаючи фактичну вартість, залишкові витрати та базову вартість. Витрати розбиті по різних категоріях, зокрема на початок проекту, визначення проекту, проектування, ітерації, навчання, впровадження, закриття роботи над продуктом і завершення проекту.

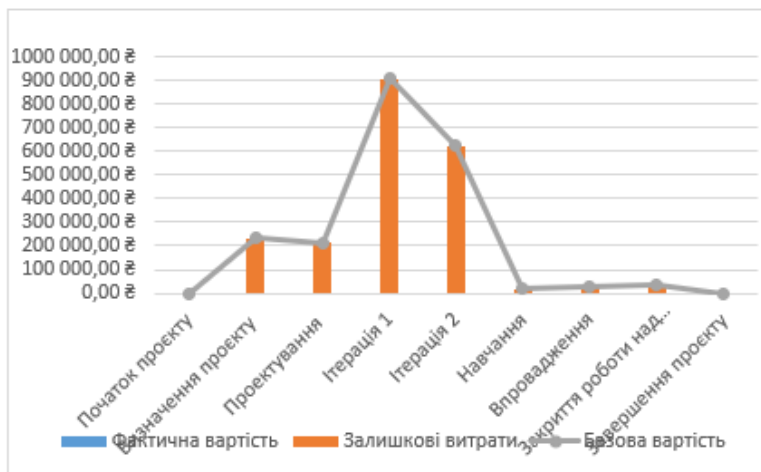
Аналіз даних: Більшість витрат припадає на ітерації 1 та 2, що є типовим для проектів з розробки, де більшість витрат виникає під час активної фази реалізації. Відхилення витрат мінімальні, що свідчить про ретельне планування та управління бюджетом. Залишкові витрати за всіма етапами відповідають

фактичним витратам, що вказує на повну відповідність запланованих і фактичних витрат.

ОГЛЯД ВИТРАТ НА ЗАВДАННЯ

СТАН ВИТРАТ

Стан витрат для завдань верхнього рівня.



ВІДОМОСТІ ПРО ВАРТІСТЬ

Відомості про вартість для всіх завдань верхнього рівня.

Ім'я	Витрати	Базова вартість	Відхилення витрат	Залишкові витрати
Початок проекту	0,00 €	0,00 €	0,00 €	0,00 €
Визначення проекту	232 750,00 €	232 750,00 €	0,00 €	232 750,00 €
Проектування	214 000,00 €	214 000,00 €	0,00 €	214 000,00 €
Ітерація 1	909 002,08 €	909 100,00 €	-97,92 €	909 002,08 €
Ітерація 2	622 000,00 €	622 000,00 €	0,00 €	622 000,00 €
Навчання	17 187,50 €	17 200,00 €	-12,50 €	17 187,50 €
Впровадження	29 750,00 €	30 000,00 €	-250,00 €	29 750,00 €
Закриття роботи над продуктом	33 250,00 €	34 000,00 €	-750,00 €	33 250,00 €
Завершення проекту	0,00 €	0,00 €	0,00 €	0,00 €

Рис 4.3.1 Звіт по огляду витрат на завдання в Microsoft Project для проекту "Pet"

Звіт "Огляд витрат" містить детальну інформацію про витрати на різних етапах проекту, а також надає аналіз відхилень від запланованого бюджету. Основна мета звіту – забезпечити прозорість фінансових витрат на проект і виявити можливі перевищення бюджету або економію коштів. Стан витрат: У звіті представлений графік, який демонструє розподіл витрат на проект. Витрати розподілені на фактичну вартість, залишкові витрати та базову вартість.

Фактична вартість проекту становить 2 057 939,59 ₴, залишкові витрати становлять 2 057 939,59 ₴. Базовий план проекту відсутній, що може свідчити про необхідність налаштування базового плану для подальшого контролю витрат.

ОГЛЯД ВИТРАТ

СБ 01.06.24 - ЧТ 31.12.26



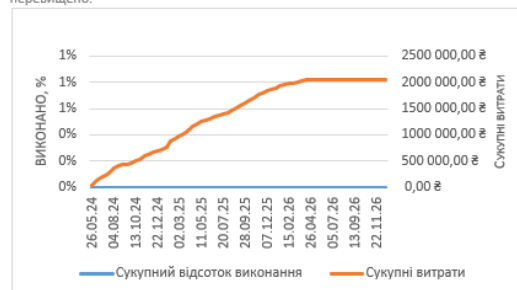
СТАН ВИТРАТ

Стан витрат для завдань верхнього рівня.

Ім'я	Фактична вартість	Залишкові витрати	Базова вартість	Витрати	Відхилення витрат
Початок проекту	0,00 ₴	0,00 ₴	0,00 ₴	0,00 ₴	0,00 ₴
Визначення проекту	0,00 ₴	232 750,00 ₴	232 750,00 ₴	232 750,00 ₴	0,00 ₴
Проектування	0,00 ₴	214 000,00 ₴	214 000,00 ₴	214 000,00 ₴	0,00 ₴
Ітерація 1	0,00 ₴	909 002,08 ₴	909 100,00 ₴	909 002,08 ₴	-97,92 ₴
Ітерація 2	0,00 ₴	622 000,00 ₴	622 000,00 ₴	622 000,00 ₴	0,00 ₴
Навчання	0,00 ₴	17 187,50 ₴	17 200,00 ₴	17 187,50 ₴	-12,50 ₴
Впровадження	0,00 ₴	29 750,00 ₴	30 000,00 ₴	29 750,00 ₴	-250,00 ₴
Закриття роботи над продуктом	0,00 ₴	33 250,00 ₴	34 000,00 ₴	33 250,00 ₴	-750,00 ₴
Завершення проекту	0,00 ₴	0,00 ₴	0,00 ₴	0,00 ₴	0,00 ₴

ПЕРЕБІГ ВИКОНАННЯ ТА ВАРТІСТЬ

Досягнутий обсяг виконаних робіт порівняно з витраченими коштами за цей час. Якщо рядок "Виконано %" опиниться під рядком сукупних витрат, можливо, бюджет проекту перевищено.



СТАН ВИТРАТ

Стан витрат для всіх завдань верхнього рівня. Ваш основний план нульовий?

[Спробуйте налаштувати як базовий план](#)

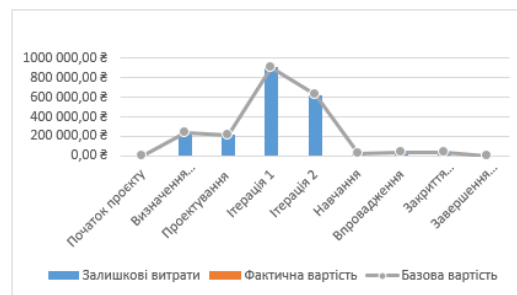


Рис 4.3.2 Звіт по огляду витрат в Microsoft Project для проекту "Pet"

Аналіз даних: Більшість витрат припадає на ітерації 1 та 2, що є типовим для проектів з розробки, де більшість витрат виникає під час активної фази реалізації. Відхилення витрат мінімальні, що свідчить про ретельне планування та управління бюджетом. Залишкові витрати за всіма етапами відповідають фактичним витратам, що вказує на повну відповідність запланованих і фактичних витрат.

Звіт надає всебічну інформацію про стан витрат на проект, дозволяючи виявити потенційні області для покращення та забезпечити ефективне управління бюджетом.

Звіт "**Огляд вартості ресурсів**" містить детальну інформацію про витрати на різні ресурси, необхідні для виконання проекту. Він включає дані про фактичну вартість, базову вартість, залишкові витрати, а також відхилення витрат для різних категорій ресурсів.

Стан витрат: У звіті представлені дані про фактичну вартість, яка становить 1,601,939.59 ₴, та залишкові витрати в розмірі 456,000.00 ₴. Базова вартість не вказана.

Витрати поділені за типами: робота, матеріали та інші витрати. Графік демонструє розподіл витрат за цими категоріями.

Аналіз даних: Основні витрати припадають на команди розробників та QA інженерів, що є типовим для проектів, пов'язаних з розробкою програмного забезпечення.

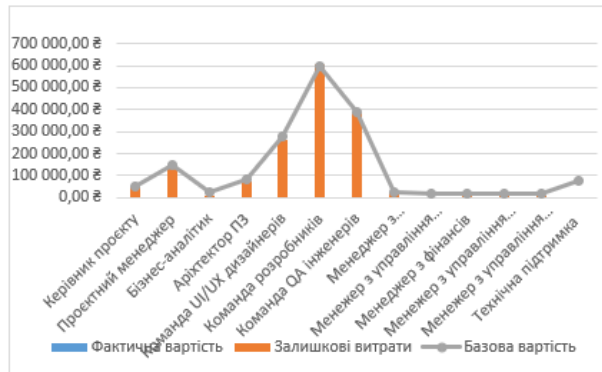
Відхилення витрат у багатьох випадках є негативними, що вказує на перевищення бюджету.

Звіт надає всебічну інформацію про стан витрат на проект, дозволяючи виявити потенційні області для покращення та забезпечити ефективне управління бюджетом.

ОГЛЯД ВАРТОСТІ РЕСУРСІВ

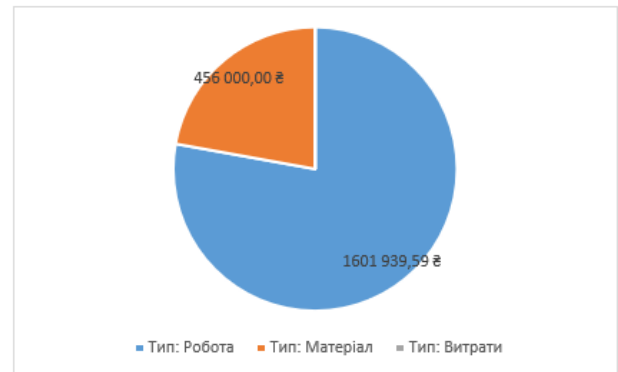
СТАН ВИТРАТ

Стан витрат для трудових ресурсів.



РОЗПОДІЛ ВИТРАТ

Розподіл вартості серед ресурсів різних типів.



ВІДОМОСТІ ПРО ВАРТІСТЬ

Відомості про вартість усіх трудових ресурсів.

Ім'я	Звичайна ставка	Базова вартість	Відхилення витрат
Керівник проекту	50 000,00 ₪/місяць	50 000,00 ₪	5 000,00 ₪
Проектний менеджер	30 000,00 ₪/місяць	150 000,00 ₪	4 500,00 ₪
Бізнес-аналітик	25 000,00 ₪/місяць	25 000,00 ₪	-18 750,00 ₪
Архітектор ПЗ	40 000,00 ₪/місяць	80 000,00 ₪	-8 000,00 ₪
Команда UI/UX дизайнерів	20 000,00 ₪/місяць	280 000,00 ₪	-18 997,92 ₪
Команда розробників	25 000,00 ₪/місяць	600 000,00 ₪	10 000,00 ₪
Команда QA інженерів	20 000,00 ₪/місяць	390 000,00 ₪	-2 000,00 ₪
Менеджер з впровадження	25 000,00 ₪/місяць	25 000,00 ₪	1 187,50 ₪
Менеджер з управління ризиками	20 000,00 ₪/місяць	20 000,00 ₪	-20 000,00 ₪
Менеджер з фінансів	20 000,00 ₪/місяць	20 000,00 ₪	-5 000,00 ₪
Менеджер з управління ресурсів	20 000,00 ₪/місяць	20 000,00 ₪	-13 000,00 ₪
Менеджер з управління якістю	20 000,00 ₪/місяць	20 000,00 ₪	-13 000,00 ₪
Технічна підтримка	15 000,00 ₪/місяць	75 000,00 ₪	-75 000,00 ₪

Рис 4.3.3 Звіт по огляду вартості ресурсів для проекту "Pet"

4.4. Управління комунікаціями в проекті

Управління комунікаціями в проекті є ключовим аспектом для забезпечення ефективного спілкування між усіма учасниками проекту. Це включає в себе не лише передачу інформації, але й забезпечення взаєморозуміння, мотивації команди та вирішення конфліктів.

Планування комунікацій починається з ідентифікації зацікавлених сторін, їх потреб та очікувань. Далі визначаються методи та інструменти комунікації, які найбільш підходять для конкретного проекту. Це можуть бути щоденні звіти, зустрічі, відеоконференції, електронна пошта, чати тощо.

З метою забезпечення ефективності комунікацій, встановлюються чіткі ролі та відповідальності учасників проекту, а також розробляється розклад комунікаційних заходів. Крім того, важливо створювати відкрите середовище, де кожен учасник має можливість висловити свої ідеї та внести внесок у розвиток проекту.

Під час реалізації проекту, необхідно постійно відстежувати ефективність комунікацій та вчасно реагувати на виникаючі проблеми або недорозуміння. Також важливо проводити оцінку задоволеності учасників проекту комунікаційним процесом і вносити корективи в план комунікацій при необхідності.

У проекті будуть використовуватись Microsoft Teams та Discord для організації комунікації та співпраці між учасниками. Обидва ці інструменти мають свої переваги:

1. **Microsoft Teams:**

- **Інтегрована платформа:** 1. Teams надає широкий спектр інструментів для комунікації, співпраці та організації робочого процесу, включаючи чат, відеоконференції, обмін файлами, завдання та календар.

- **Зручність від Microsoft 365:** Як частина екосистеми Microsoft 365, Teams інтегрується з іншими програмами, такими як Outlook, SharePoint, OneDrive, Word, Excel, PowerPoint, що полегшує співпрацю та обмін даними.

- **Безпека та керування доступом:** Teams забезпечує різні рівні безпеки та можливості керування доступом, що важливо для захисту конфіденційної інформації та даних проекту.

2. **Discord:**

- **Фокус на спільноті:** Discord відомий своєю спрямованістю на спільноти геймерів, але також добре підходить для комунікації у різних професійних командах.
- **Гнучкість та налаштування:** Discord пропонує широкі можливості налаштування серверів та каналів, що дозволяє створювати зручні простори для обговорень, спільної роботи та спілкування.
- **Голосові та відеодзвінки:** У Discord є можливість проводити групові голосові та відеодзвінки, що дозволяє зручно спілкуватися в реальному часі.

Використання Teams та Discord разом дозволить ефективно організувати комунікацію, співпрацю та обмін інформацією в межах проекту, забезпечуючи гнучкість та доступність для всіх учасників.

Використання Microsoft Teams та Discord в нашому проекті демонструє нашу готовність до забезпечення ефективної комунікації та співпраці між усіма учасниками. Обидва інструменти надають широкі можливості для організації робочих процесів, спільної роботи та взаємодії в реальному часі. Teams від Microsoft інтегрується з іншими продуктами Microsoft 365 та надає широкий спектр інструментів, тоді як Discord відомий своєю гнучкістю та спрямованістю на спільноту користувачів. Залучення обох цих платформ забезпечить ефективний обмін інформацією, організацію комунікації та сприятиме успішному виконанню проекту.

4.5. Управління зацікавленими сторонами проекту

Управління зацікавленими сторонами проекту - це ключовий аспект успішного виконання будь-якого проекту. У нашому випадку, це означає виявлення, аналіз та ефективне взаємодію з усіма групами зацікавлених сторін, які можуть бути впливовими на проект або впливати на його результати.

Однією з переваг управління зацікавленими сторонами є попередня ідентифікація цих сторін та їх потреб. Після проведення ретельного аналізу

різноманітних зацікавлених сторін, таких як клієнти, користувачі, власники бізнесу, управління, команда проекту та інші стейкхолдери. Це дозволило розробити стратегію комунікації та взаємодії з кожною з цих груп.

Ще однією перевагою є можливість вирішувати конфлікти та виробляти компроміси між різними сторонами. Це робиться шляхом активного слухання, врахування їхніх потреб та пошуку оптимальних рішень, які задовольняють усіх учасників.

Управління зацікавленими сторонами також передбачає постійний моніторинг їхніх потреб та реакцій на проект, щоб швидко реагувати на будь-які зміни в їхніх вимогах чи очікуваннях.

Для проекту буде важливим проаналізувати та скласти список зацікавлених сторін для подальшого успішного розвитку проекту.

Таблиця 4.5.1

Карта зацікавлених сторін для проекту "Pet"

Зацікавлена сторона	Опис
Керівник проекту	Зацікавлений у успішному завершенні проекту, досягненні поставлених цілей та забезпеченні його ефективного управління
Проектний менеджер	Має інтерес у забезпеченні вчасності та якості виконання завдань, відповідальний за планування та виконання проекту.
Команда проекту	Це розробники, тестувальники та інші учасники, зацікавлені в успішному впровадженні та розробці ІТ-продукту
Інвестори	Мають інтерес у вкладенні коштів у проект та отриманні прибутку або інших вигод від його успішності.
Конкуренти	Можуть бути зацікавлені в реакції на ваш проект та впливі його успіху на їхні позиції на ринку

ЗМІ/Соціальні мережі	Можуть відігравати роль у поширенні інформації про ваш проєкт серед широкої аудиторії
Ветеринарні клініки	Як потенційні користувачі, вони зацікавлені у впровадженні та використанні вашого застосунку для полегшення своєї роботи
Держава	Може мати інтерес у регулюванні та контролі за використанням медичних інформаційних систем у сфері домашніх тварин
Клієнт	Власники домашніх тварин, які будуть користувачами застосунку "Pet" і мають інтерес у його ефективності та зручності

Таблиця 4.5.2

Таблиця інтересів зацікавлених сторін проєкту "Pet"

Зацікавлена сторона	Інтереси
Керівник проєкту	Успішне завершення проєкту. Досягнення поставлених цілей. Забезпечення ефективного управління проєктом.
Проектний менеджер	Вчасне та якісне виконання завдань. Планування та координація робіт згідно з графіком. Забезпечення ресурсів для успішного виконання проєкту.
Команда проєкту	Реалізація проєкту з використанням передових технологій. Створення високоякісного та функціонального ІТ-продукту. Успішне впровадження та підтримка застосунку "Pet".
Інвестори	Отримання прибутку або інших вигод від успішного проєкту.

	Забезпечення відповідного рівня ризику та повернення інвестицій.
Конкуренти	Можливе бажання обмежити конкурентоспроможність проєкту. Реакція на конкурентні переваги або недоліки вашого застосунку.
ЗМІ/Соціальні мережі	Поширення інформації про проєкт серед широкої аудиторії. Публікація позитивних відгуків та реклама продукту.
Ветеринарські клініки	Оптимізація робочих процесів через використання застосунку "Pet". Покращення якості та швидкості надання медичних послуг.
Держава	Забезпечення відповідності проєкту законодавству. Підтримка розвитку та використання сучасних технологій у медичній галузі.
Клієнт	Якість медичного обслуговування Зручність та ефективність Безпека даних

Висновки до розділу 4

Розділ детально описує інформаційну модель проекту розробки застосунку "Pet", яка охоплює всі ключові аспекти управління проектом, включаючи планування, управління часом, вартістю, комунікаціями та зацікавленими сторонами. В цьому розділі були розроблені:

1. План управління проектом
2. Управління часом і розподіл ресурсів
3. Управління вартістю проекту
4. Управління комунікаціями
5. Управління зацікавленими сторонами

Цей розділ надав комплексне розуміння управління проектом розробки застосунку "Pet", підкресливши важливість планування, ефективного використання ресурсів, контролю витрат, організації комунікацій та взаємодії з зацікавленими сторонами для досягнення успішних результатів.

ЗАГАЛЬНІ ВИСНОВКИ

Робота присвячена розробці медичної інформаційної системи "Pet", яка спрямована на підвищення якості медичного обслуговування домашніх тварин в Україні. В процесі виконання роботи було проведено детальний аналіз предметної області, що дозволило виявити головні вимоги та особливості, необхідні для створення ефективного програмного забезпечення.

Було здійснено аналіз існуючих рішень та методологій управління проєктами, що допомогло визначити оптимальний підхід для реалізації проєкту. Основною методологією, обраною для розробки, став Agile, яка забезпечує гнучкість та адаптивність у процесі розробки, дозволяє швидко реагувати на зміни вимог та забезпечує постійне покращення продукту.

Розроблено концептуальний дизайн системи, що включає детальні діаграми EER та опис зв'язків між сутностями, які забезпечують ефективне управління даними та їх взаємодію. Вибір сучасних технологій, таких як: веб-технології, фреймворки та бази даних, забезпечив надійність, масштабованість та ефективність роботи системи.

Особлива увага приділялася розробці функціональних компонентів системи, що включають модулі обліку клієнтів, планування та обліку візитів, доступ до медичних записів, фінансовий облік та генерацію звітів. Було створено інтуїтивно зрозумілий інтерфейс користувача, що забезпечує зручність використання та високу продуктивність.

У результаті виконаної роботи було розроблено інформаційну модель автоматизованої програми, яка забезпечує ефективну взаємодію між ветеринарними клініками та власниками домашніх тварин, сприяє підвищенню якості медичного обслуговування та задоволеності клієнтів. Розробка застосунку "Pet" стала важливим кроком у напрямку вдосконалення процесів надання медичних послуг для домашніх тварин, що відповідає сучасним вимогам та тенденціям у сфері інформаційних технологій.

СПИСОК ДЖЕРЕЛ

1. How to Choose the Best Veterinary Software for Your Clinic. [Онлайновий] 2024 p. <https://vetsuccess.com/how-to-choose-the-best-veterinary-software/>.
2. Федоров, А. В. *Методи автоматизації ветеринарної медицини: огляд сучасних програмних рішень*. 2022.
3. Коваленко, І. О. *Інформаційні системи в медицині: приклад застосування для домашніх тварин*. 2021.
4. Watson, J. D. *Veterinary Information Systems: A Review of the Current Market*,. 2021.
5. Smith, L. M. *The Role of User Interface Design in Veterinary Software*. 2022.
6. Wilson, D. R. *Comparative Analysis of Veterinary Software Solutions*. 2021.
7. Veterinary Information Systems: A Comprehensive Review. [Онлайновий] 2024 p. <https://www.vetcompanion.com/articles/veterinary-information-systems-review>.
8. IDEXX Cornerstone Veterinary Software. [Онлайновий] 2024 p. <https://www.idexx.com/en/veterinary/software/idexx-cornerstone/>.
9. eVetPractice: Veterinary Practice Management Software. [Онлайновий] 2024 p. <https://www.covetrus.com/products/evetpractice>.
10. The Future of Veterinary Software. [Онлайновий] 2024 p. <https://www.veterinarypracticenews.com/the-future-of-veterinary-software/>.
11. Introduction to Agile Project Management,. [Онлайновий] 2024 p. <https://www.agilealliance.org/introduction-to-agile>.
12. Waterfall vs. Agile: Which Methodology is Right for Your Project? [Онлайновий] 2024 p. <https://www.pmi.org/learning/library/waterfall-vs-agile-right-methodology-10254>.
13. Scrum Guide. [Онлайновий] 2024 p. <https://www.scrum.org/resources/scrum-guide>.
14. Kanban: A Brief Introduction. [Онлайновий] 2024 p. <https://leankit.com/learn/kanban/kanban-introduction>.

15. The Scaled Agile Framework (SAFe). [Онлайновый] 2024 p. <https://www.scaledagileframework.com/>.
16. Large-Scale Scrum (LeSS): Overview. [Онлайновый] 2024 p. <https://less.works/less/framework/index.html>.
17. Royce, Winston W. *Managing the Development of Large Software Systems*. 1970.
18. Agile Project Management: Best Practices and Methodologies. [Онлайновый] 2024 p. <https://www.smartsheet.com/agile-project-management>.
19. The Benefits of Agile Pr. [Онлайновый] 2024 p. <https://www.forbes.com/sites/forbestechcouncil/2022/05/05/the-benefits-of-agile-project-management/>.
20. Agile Project Management: Best Practices and Methodologies. [Онлайновый] 2024 p. <https://www.smartsheet.com/agile-project-management>.
21. Silberschatz, Abraham, Henry F. Korth, and S. Sudarshan. *Database System Concepts*. 2019.
22. Introduction to Database Systems. [Онлайновый] 2024 p. <https://www.oracle.com/database/what-is-database/>.
23. Database Management System (DBMS). [Онлайновый] 2024 p. <https://www.techopedia.com/definition/24381/database-management-system-dbms>.
24. The Relational Model for Database Management. [Онлайновый] 2024 p. <https://cs186.berkeley.edu/the-relational-model-for-database-management/>.
25. SQL Basics: Learn SQL with this Basic SQL Tutorial. [Онлайновый] 2024 p. <https://www.datacamp.com/community/tutorials/sql-tutorial-query>.
26. Microsoft Project Overview. [Онлайновый] 2024 p. <https://www.microsoft.com/en-us/microsoft-365/project/project-management-software>.
27. Project Management Guide. [Онлайновый] 2024 p. <https://www.pmi.org/pmbok-guide-standards/foundational/project-management-guide>.
28. Project Management Phases. [Онлайновый] 2024 p. <https://www.smartsheet.com/project-management-phases>.

29. Stakeholder Management. [Онлайновый] 2024 р.
https://www.mindtools.com/pages/article/newPPM_07.htm.
30. Resource Management. [Онлайновый] 2024 р.
<https://www.wrike.com/project-management-guide/faq/what-is-resource-management/>.
31. Cost Management in Project Management. [Онлайновый] 2024 р.
<https://www.projectmanager.com/blog/cost-management>.
32. Communications Management. [Онлайновый] 2024 р.
<https://www.projectmanagementdocs.com/template/project-documents/communications-management-plan/>.
33. Risk Management in Projects. [Онлайновый] 2024 р.
<https://www.projectmanager.com/blog/risk-management>.
34. Using Discord for Project Management. [Онлайновый] 2024 р.
<https://discord.com/blog/using-discord-for-project-management>.
35. Creating a Project Plan with Gantt Chart. [Онлайновый] 2024 р.
<https://www.teamgantt.com/guide-to-gantt-charts/project-plan>.