

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ АГРОТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
ВІДДІЛЕННЯ «ІНЖЕНЕРНА ІНФРАСТРУКТУРА ТА
КОМП'ЮТЕРНІ НАУКИ»
ЦИКЛОВА КОМІСІЯ СПЕЦІАЛЬНОСТІ «КОМП'ЮТЕРНІ НАУКИ»

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

освітньо-професійний ступінь «фаховий молодший бакалавр»

на тему:

Розробка прикладної програми для ветеринарної лікарні

Виконав студент (ка) 4 курсу, групи П-41
спеціальності 122 «Комп'ютерні науки»

Федорчук Роман Олександрович

Керівник Устименко Ярослав Іванович

Рецензент _____

Житомир – 2024 року

РЕФЕРАТ

Записка: 58 стор., 19 рис., 1 додаток, 20 джерел.

Ключові слова: WINDOWS FORM, C#, АВТОМАТИЗАЦІЯ, ІНТЕРФЕЙС КОРИСТУВАЧА, СИСТЕМА ОБЛІКУ, РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, ІНФОРМАЦІЙНА СИСТЕМА, SQLite.

Дипломна робота присвячена розробці програмного забезпечення для автоматизації процесів у ветеринарних клініках. Основна увага приділяється створенню прикладної програми з інтуїтивним інтерфейсом, яка допомагатиме веденню медичних записів та управлінню даними про тварин.

У роботі проведено аналіз інформаційних потреб ветеринарних клінік, виокремлено основні функції та вимоги до програмного забезпечення. Для забезпечення зручного та ефективного функціонування розроблена база даних на основі SQLite, що забезпечує надійне зберігання та обробку медичної інформації.

Реалізація програмного продукту виконана з використанням технологій Windows Form та мови програмування C#. Результатом роботи є функціональна програма, яка спрощує рутинні процеси у ветеринарній практиці та покращує якість обслуговування тварин та їх власників.

Отже, дипломна робота відображає процес розробки та впровадження програмного забезпечення для ветеринарних клінік, з урахуванням сучасних технологій та вимог користувачів.

Зміст

Вступ.....	5
РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ	8
1.1. Постановка основної задачі розробки	8
1.2. Огляд та порівняння аналогічних систем.	13
1.3. Вибір та обґрунтування технологій розробки	14
Висновки до розділу 1	17
РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА	18
2.1. Розробка бази даних для інформаційної системи	18
2.2. Структура програми	21
2.3 Програмна реалізація	24
2.4 Опис класів	27
Висновки до розділу 2	28
РОЗДІЛ 3. КЕРІВНИЦТВО ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ	30
3.1. Мінімальні вимоги до системи	30
3.2. Інтерфейс користувача	31
3.3 Тестування програмного продукту	37
Висновки до розділу 3	38
ВИСНОВКИ.....	40
Перелік літературних джерел	41
Додаток А.....	44

					ДР.122.041.030.ПЗ					
Змн.	Арк.	№ докум.	Підпис	Дата	Розробка прикладної програми для ветеринарної лікарні			Літ.	Арк.	Аркушів
Розробив	Федорчук Р.О.									
Перевірів	Устименко Я.І.								3	53
Рецензент								ЖАТФК		
Н. Контр.										
Затвердив.	Лавріщев О.О.									

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

GUI - Графічний користувацький інтерфейс.

SQL - Мова структурованих запитів.

CRUD - Створення, читання, оновлення, видалення (Create, Read, Update, Delete).

API - Інтерфейс програмування застосунків.

IDE - Інтегроване середовище розробки.

ORM - Об'єктно-реляційне відображення.

UI - Користувацький інтерфейс.

MVC - Модель-вид-контролер.

DTO - Об'єкти передачі даних.

OOP - Об'єктно-орієнтоване програмування.

					ДР.122.041.030.ПЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

Вступ

В сучасному світі інформаційні технології відіграють важливу роль у різних сферах діяльності, включаючи медицину та ветеринарію. Ветеринарні клініки, як і інші медичні заклади, стикаються з необхідністю автоматизації процесів управління та обліку. Розробка прикладних програм для ветеринарних лікарень дозволяє значно підвищити ефективність їхньої роботи, знизити кількість помилок, поліпшити якість обслуговування пацієнтів та оптимізувати використання ресурсів.

В умовах швидкого розвитку інформаційних технологій, автоматизація бізнес-процесів стає необхідністю для різних галузей, включаючи ветеринарну медицину. Ветеринарні клініки, як і інші медичні заклади, потребують ефективних інструментів для управління даними про пацієнтів, планування та обліку візитів, зберігання історії лікувань, а також для аналізу та звітування. Наявність таких інструментів дозволяє значно підвищити ефективність роботи персоналу, покращити якість обслуговування клієнтів та оптимізувати використання ресурсів.

На сьогоднішній день багато ветеринарних клінік все ще використовують паперовий облік або застарілі програмні рішення, що обмежує їхню здатність до швидкої та точної обробки інформації. Це може призводити до помилок у веденні документації, втрати даних, затримок у процесі обслуговування та, як наслідок, до зниження якості наданих послуг. Введення сучасної прикладної програми, що інтегрується з базою даних, може вирішити ці проблеми, забезпечивши надійне зберігання та швидкий доступ до інформації.

Автоматизація процесів у ветеринарних клініках має низку переваг, серед яких:

- підвищення точності та швидкості обробки даних;

					ДР.122.041.030.ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

- зменшення кількості рутинних завдань для персоналу, що дозволяє їм зосередитися на безпосередньому обслуговуванні пацієнтів;
- поліпшення якості управління клієнтськими відносинами за рахунок швидкого доступу до історії хвороб та лікувань;
- забезпечення аналітичних можливостей для прийняття управлінських рішень на основі точних даних.

У зв'язку з вищезазначеним, розробка прикладної програми для ветеринарної клініки, яка базується на сучасних технологіях та відповідає актуальним вимогам до управління даними, є вкрай важливою та своєчасною. Це дозволить не тільки покращити ефективність роботи ветеринарних закладів, але й підвищити якість медичної допомоги для тварин, що є важливим аспектом у загальному контексті охорони здоров'я.

Метою даної дипломної роботи є розробка прикладної програми для ветеринарної лікарні, яка забезпечуватиме автоматизацію основних процесів управління пацієнтами та їхніми власниками, обліку візитів та лікувань, а також створення звітів та аналітики.

Завданнями даної роботи є:

- аналіз існуючих рішень та визначення основних вимог до програмного забезпечення для ветеринарної лікарні;
- проектування бази даних для зберігання інформації про пацієнтів, власників, візити та лікування;
- розробка користувацького інтерфейсу, який забезпечуватиме зручну роботу з програмою для персоналу клініки;
- реалізація функціональної частини програми на основі вимог та проекту бази даних;
- тестування програми та оцінка її ефективності в умовах реальної ветеринарної клініки.

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Об'єктом дослідження є процеси управління та обліку у ветеринарних клініках. Предметом дослідження є інформаційні системи та програмне забезпечення, які використовуються для автоматизації цих процесів.

В ході роботи використовувалися методи системного аналізу, проектування баз даних, об'єктно-орієнтованого програмування, а також методи тестування програмного забезпечення.

Результати даної роботи можуть бути використані для впровадження у ветеринарних клініках з метою підвищення ефективності їхньої роботи. Розроблена програма дозволить автоматизувати рутинні процеси, зменшити кількість помилок, підвищити якість обслуговування пацієнтів та покращити управління ресурсами клініки.

Дипломна робота складається з вступу, основної частини, висновків, списку використаних джерел та додатків. Основна частина включає розділи, присвячені аналізу існуючих рішень, проектуванню бази даних, розробці користувацького інтерфейсу та функціональної частини програми, а також тестуванню та оцінці ефективності розробленої програми.

					ДР.122.041.030.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЙ РОЗРОБКИ

1.1. Постановка основної задачі розробки

Основною задачею дипломної роботи є розробка прикладної програми для ветеринарної лікарні, яка забезпечить автоматизацію основних процесів управління та обліку, підвищить ефективність роботи персоналу, зменшить кількість помилок і покращить якість обслуговування пацієнтів.

Для успішного виконання основної задачі необхідно виконати такі підзадачі:

Аналіз існуючих рішень і визначення вимог:

Вивчити існуючі програмні рішення для автоматизації ветеринарних клінік.

Провести опитування користувачів (лікарів, адміністраторів) для визначення їхніх потреб і очікувань від програми.

Сформулювати функціональні та нефункціональні вимоги до програмного забезпечення.

Проектування бази даних:

Розробити структуру бази даних для зберігання інформації про пацієнтів, власників, візити та лікування.

Забезпечити інтеграцію між таблицями для підтримки зв'язків та забезпечення цілісності даних.

Створити схему бази даних на основі визначених вимог.

Розробка користувацького інтерфейсу:

Створити інтуїтивно зрозумілий та зручний інтерфейс для користувачів (лікарів, адміністраторів).

Забезпечити легкий доступ до основних функцій програми, таких як реєстрація пацієнтів, планування візитів, ведення історії лікувань.

Реалізувати механізми введення, редагування та видалення даних.

Реалізація функціональної частини програми:

Розробити основні модулі програми, які забезпечуватимуть реалізацію основних функцій (реєстрація пацієнтів, облік візитів, створення звітів).

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

Інтегрувати програму з базою даних для забезпечення зберігання та доступу до даних.

Забезпечити належний рівень безпеки даних, включаючи аутентифікацію користувачів та контроль доступу до функцій програми.

Тестування та впровадження програми:

Провести тестування програми для виявлення та виправлення помилок.

Забезпечити тестування програми в умовах реальної ветеринарної клініки для оцінки її ефективності та зручності використання.

Зібрати зворотний зв'язок від користувачів та внести необхідні корективи для покращення програми.

Провести навчання персоналу клініки для забезпечення ефективного використання програми.

Постановка основної задачі полягає в розробці інтегрованої програмної системи, яка відповідає специфічним потребам ветеринарних клінік. Програма має забезпечити централізоване зберігання даних, зручний доступ до них, автоматизацію рутинних процесів та створення звітів для аналізу діяльності клініки.

Це включає створення бази даних для зберігання інформації про пацієнтів (тварин), їхніх власників, історії візитів та лікувань. Інтерфейс програми має бути розроблений з урахуванням потреб користувачів, забезпечуючи зручну навігацію та доступ до основних функцій.

Функціональна частина програми має реалізовувати основні операції з даними, включаючи реєстрацію нових пацієнтів, планування та облік візитів, ведення історії лікувань, а також генерування звітів для аналізу та управління діяльністю клініки.

Тестування та впровадження програми мають забезпечити її надійну роботу в реальних умовах, а документація та навчання користувачів допоможуть персоналу клініки швидко освоїти нову систему та ефективно її використовувати.

					ДР.122.041.030.ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

Аналіз інформаційних потреб ветеринарної клініки є ключовим етапом у розробці прикладної програми. Він включає визначення основних процесів, які повинні бути автоматизовані, а також інформаційних потоків, які потребують обробки та зберігання. Це дозволить забезпечити ефективну роботу клініки, оптимізувати використання ресурсів та підвищити якість обслуговування пацієнтів.

Основні інформаційні потреби ветеринарної клініки можна розділити на кілька категорій:

Управління даними про пацієнтів (тварин): ідентифікаційні дані (ім'я, вид, порода, вік, стать); історія хвороб та лікувань; вакцинації та профілактичні заходи; відомості про алергії та особливості догляду;
Управління даними про власників тварин: ідентифікаційні дані (ім'я, контактна інформація, адреса).

Управління даними про візити: записи про заплановані та минулі візити; причини візитів та надані послуги; рекомендації щодо подальшого лікування та догляду;

Управління даними про лікування та процедури: опис наданих лікувальних процедур та їх результати;

Реєстрація пацієнтів та власників: поточний спосіб збору та зберігання даних; частота оновлення інформації та проблеми, пов'язані з цим;

Запис на візити та облік візитів: процес запису на прийом; системи обліку візитів та надання послуг; проблеми, що виникають через недостатню автоматизацію;

Ведення історії хвороб та лікувань: поточний метод зберігання медичних записів; доступ до історії хвороб під час прийомів; проблеми, пов'язані з пошуком та доступом до інформації.

На основі проведеного аналізу, можна сформулювати основні вимоги до інформаційної системи для ветеринарної клініки:

Централізоване зберігання даних: єдина база даних для зберігання всієї інформації про пацієнтів, власників, візити та лікування.

					ДР.122.041.030.ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

Інтуїтивно зрозумілий користувацький інтерфейс: зручний інтерфейс для введення, пошуку та редагування даних; можливість швидкого доступу до потрібної інформації;

Автоматизація рутинних процесів: автоматичний облік візитів та лікувань; генерація нагадувань про заплановані процедури та візити.

Аналіз інформаційних потреб ветеринарної клініки показав необхідність розробки сучасної інформаційної системи, яка забезпечить автоматизацію основних процесів управління та обліку. Впровадження такої системи дозволить підвищити ефективність роботи клініки, зменшити кількість помилок, покращити якість обслуговування пацієнтів та забезпечити зручний доступ до необхідної інформації для персоналу клініки.

Ветеринарна клініка виконує широкий спектр функцій, спрямованих на забезпечення здоров'я та добробуту тварин. До основних функцій ветеринарної клініки належать медичні послуги а саме:

Діагностика та лікування захворювань: проведення клінічних обстежень; встановлення діагнозу на основі симптомів та лабораторних досліджень; призначення та проведення лікування різних захворювань.

Хірургічні втручання: планові та екстрені операції; стерилізація та кастрація; оперативне лікування травм та інших ургентних станів.

Вакцинація та профілактика захворювань: проведення планових щеплень; профілактичні огляди та консультації; профілактика паразитарних інвазій.

Стоматологічні послуги: чистка зубів; лікування захворювань ротової порожнини; видалення зубів.

Консультації з утримання та догляду: рекомендації щодо харчування; поради з догляду за шерстю, шкірою та кігтями; консультації з питань поведінки та тренувань.

Консультації з розведення: поради щодо планування розведення; обстеження тварин перед спарюванням; допомога під час вагітності та пологів.

					ДР.122.041.030.ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Лабораторні дослідження: аналіз крові, сечі та калу; бактеріологічні та вірусологічні дослідження. цитологічні та гістологічні дослідження.

Діагностичні дослідження: ультразвукова діагностика (УЗД); рентгенографія; ендоскопія.

Стаціонар для хворих тварин: нагляд за тваринами, що потребують інтенсивної терапії; післяопераційний догляд; лікування важких випадків, що потребують постійного медичного контролю.

Реабілітація та фізіотерапія: відновлення після травм та операцій; фізіотерапевтичні процедури (електротерапія, масаж, гідротерапія).

Управління записами та документацією: ведення медичних карток пацієнтів; облік візитів, процедур та лікування; зберігання історії хвороб та інших важливих документів.

Ветеринарна клініка виконує широкий спектр функцій, спрямованих на забезпечення здоров'я та добробуту тварин. Від ефективності реалізації цих функцій залежить не тільки якість надання медичної допомоги, але й загальний рівень задоволеності клієнтів клініки. Автоматизація процесів та використання сучасних інформаційних систем дозволяють оптимізувати роботу клініки, підвищити її ефективність та забезпечити високий рівень обслуговування пацієнтів.

					ДР.122.041.030.ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2. Огляд та порівняння аналогічних систем.

На сучасному ринку існує багато програмних рішень для ветеринарних клінік, кожне з яких має свої унікальні функції та переваги. Порівняння п'яти популярних систем допоможе визначити їхні сильні та слабкі сторони і зрозуміти, які можливості вони надають користувачам.

Vetstation є потужною системою для управління ветеринарними клініками, яка пропонує широкий спектр функцій для адміністрування та медичних записів. Вона дозволяє вести облік пацієнтів та їхніх власників, планувати візити, зберігати історію хвороб і лікування, а також управляти фінансами та запасами. Vetstation підтримує інтеграцію з лабораторним обладнанням, що спрощує проведення діагностичних тестів. Недоліком системи може бути її складність у налаштуванні та освоєнні, що може вимагати додаткового навчання персоналу.

AVImark є однією з найбільш популярних систем у США. Вона забезпечує широкий функціонал, включаючи управління медичними записами, планування візитів, фінансовий облік та звітність, а також управління запасами. AVImark має зручний інтерфейс, який легко освоїти, і підтримує інтеграцію з різними сторонніми програмами. Проте, висока вартість ліцензії може бути перешкодою для невеликих клінік з обмеженим бюджетом.

Cornerstone є ще однією відомою системою, яка пропонує широкий спектр функцій для ветеринарних клінік. Вона дозволяє вести детальний облік пацієнтів, управляти візитами та лікуваннями, а також проводити фінансовий облік. Cornerstone підтримує інтеграцію з лабораторними та діагностичними системами, що робить її зручною для комплексного медичного обслуговування. Однак, через високий рівень функціональності система може бути занадто складною для невеликих клінік.

ezyVet є хмарною системою управління, яка пропонує гнучкість і доступність з будь-якої точки з доступом до Інтернету. Вона підтримує широкий спектр функцій, включаючи облік пацієнтів, управління візитами,

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

фінансовий облік та звітність. ezyVet також інтегрується з іншими хмарними сервісами, що забезпечує додаткові можливості для користувачів. Проте, залежність від Інтернету може бути недоліком у випадку нестабільного з'єднання.

Hippo Manager є доступною та легкою у використанні системою, яка пропонує основні функції для управління ветеринарною клінікою. Вона включає управління пацієнтами та візитами, фінансовий облік та звітність, а також основні функції управління запасами. Hippo Manager має простий інтерфейс, що робить її підходящою для невеликих клінік з обмеженими потребами. Однак, вона може не мати деяких розширених функцій, які доступні у більш дорогих системах.

Таким чином, кожна з цих систем має свої унікальні переваги та недоліки. Вибір відповідної системи залежить від специфічних потреб клініки, її бюджету та вимог до функціональності. Vetstation і AVImark надають широкий спектр можливостей для великих клінік, тоді як ezyVet забезпечує гнучкість завдяки хмарним технологіям, а Hippo Manager є доступним рішенням для невеликих клінік.

1.3. Вибір та обґрунтування технологій розробки

C# є однією з найбільш популярних мов програмування для розробки десктопних додатків на платформі Windows завдяки її інтеграції з .NET Framework. Вона забезпечує високу продуктивність, багатий набір бібліотек та інструментів, а також простоту та зручність у використанні. Крім того, C# підтримує об'єктно-орієнтоване програмування, що спрощує розробку, тестування та підтримку програмного забезпечення.

.NET є потужною платформою для розробки десктопних додатків на Windows. Вона пропонує широкий набір класів та бібліотек для роботи з графічним інтерфейсом, базами даних, мережею та багатьма іншими аспектами. .NET забезпечує високу продуктивність та безпеку, а також інтеграцію з іншими технологіями Microsoft, що робить його ідеальним вибором для розробки корпоративних додатків.

					ДР.122.041.030.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Windows Forms є однією з основних технологій для створення графічних інтерфейсів на платформі .NET. Вона забезпечує простоту та швидкість розробки, підтримує багатий набір елементів управління та дозволяє створювати зручні та інтуїтивно зрозумілі інтерфейси для користувачів. Windows Forms також інтегрується з іншими компонентами .NET, що спрощує розробку комплексних додатків.

SQLite є легковажною і вбудованою реляційною базою даних, яка не вимагає налаштування сервера. Вона підходить для додатків, які потребують зберігання даних локально на пристрої. SQLite забезпечує високу продуктивність, надійність та простоту у використанні. Вона підтримує основні SQL-запити та має невеликий обсяг, що робить її ідеальною для невеликих та середніх додатків, таких як ветеринарні клініки.

Entity Framework (EF) є потужним ORM (Object-Relational Mapping) інструментом для .NET, який спрощує роботу з базами даних, дозволяючи розробникам працювати з базою даних як з об'єктами. Використання EF дозволяє зменшити кількість написаного коду, підвищити продуктивність розробки та забезпечити підтримку різних баз даних без зміни коду додатка. Крім того, EF інтегрується з C# та .NET, що робить його зручним вибором для цього проекту.

Dapper є легким ORM для .NET, який забезпечує високу продуктивність та простоту використання. Він менш складний, ніж Entity Framework, і підходить для тих випадків, коли необхідна швидка робота з базою даних. Dapper забезпечує швидке виконання SQL-запитів та дозволяє використовувати прості об'єктні моделі, що робить його ідеальним для невеликих проектів з високими вимогами до продуктивності.

NUnit є популярним фреймворком для модульного тестування у .NET. Він забезпечує потужні можливості для створення та виконання тестів, підтримує різні типи тестів (наприклад, юніт-тести, інтеграційні тести) та має хорошу інтеграцію з інструментами розробки, такими як Visual Studio.

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

Використання NUnit дозволяє забезпечити високу якість коду та зменшити кількість помилок на етапі розробки.

Git є популярною системою контролю версій, яка дозволяє ефективно керувати змінами в коді, підтримувати спільну роботу над проектом та забезпечувати історію змін. Використання Git забезпечує гнучкість у розробці, можливість легко відслідковувати помилки та забезпечує надійний спосіб зберігання та відновлення коду.

Вибір технологій для розробки прикладної програми для ветеринарної клініки базується на їх продуктивності, простоті використання, інтеграції та відповідності потребам проекту. Використання C#, .NET, Windows Forms, SQLite, Entity Framework, Dapper, NUnit та Git дозволяє створити надійну, ефективну та зручну у використанні систему, яка відповідає вимогам сучасних ветеринарних клінік.

					ДР.122.041.030.ПЗ	Арк.
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

Висновки до розділу 1

В рамках розділу було проведено детальний аналіз інформаційних потреб ветеринарної клініки та вибрано відповідні технології для розробки прикладної програми. Розглянуті функціональні вимоги дозволили визначити основні напрямки автоматизації процесів клініки, що включають управління даними про пацієнтів, власників, візити, лікування, фінансовий облік та управління запасами.

На основі аналізу сучасних програмних рішень для ветеринарних клінік, було порівняно п'ять популярних систем. Це дозволило виявити їхні сильні та слабкі сторони, що допомогло сформулювати вимоги до майбутньої системи та визначити ключові аспекти для її розробки.

Вибір технологій для реалізації прикладної програми обґрунтований їхньою відповідністю вимогам проекту, простотою інтеграції та використання, а також надійністю та продуктивністю. Використання мови програмування C#, платформи .NET Framework та Windows Forms забезпечує створення ефективного та зручного інтерфейсу користувача. База даних SQLite та ORM інструменти, такі як Entity Framework та Dapper, гарантують надійне зберігання та обробку даних. Тестування з використанням NUnit та система контролю версій Git забезпечують високу якість коду та надійність розробки.

Таким чином, проведений аналіз та вибір технологій створюють міцну основу для розробки прикладної програми, яка відповідатиме потребам ветеринарної клініки, покращить якість обслуговування та оптимізує внутрішні процеси. Наступним кроком буде реалізація даного проекту з використанням вибраних технологій, що дозволить досягти поставлених цілей та вирішити визначені завдання.

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

РОЗДІЛ 2. ПРОЕКТУВАННЯ І РОЗРОБКА

2.1. Розробка бази даних для інформаційної системи

Для забезпечення роботи прикладної програми ветеринарної клініки було розроблено базу даних на основі SQLite. Вибір цієї бази даних обумовлений її легкістю, простотою інтеграції та високою продуктивністю, що робить її ідеальним рішенням для невеликих та середніх проектів. Нижче описано структуру бази даних та основні таблиці, необхідні для забезпечення функціональності програми.

База даних ветеринарної клініки включає кілька основних таблиць, які зберігають інформацію про пацієнтів (тварин), власників, візити, лікування, процедури, медикаменти та фінансові операції. Нижче наведено опис кожної таблиці та її ключові поля.

Таблиці бази даних

Таблиця Owners

Зберігає інформацію про власників тварин.

OwnerID INTEGER Унікальний ідентифікатор власника

FirstName TEXT Ім'я власника

LastName TEXT Прізвище власника

PhoneNumber TEXT Номер телефону

Email TEXT Адреса електронної пошти

Address TEXT Адреса проживання

Таблиця Patients

Зберігає інформацію про пацієнтів (тварин).

PatientID INTEGER Унікальний ідентифікатор пацієнта

OwnerID INTEGER Ідентифікатор власника (зовнішній ключ)

Name TEXT Ім'я тварини

Species TEXT Вид тварини

Breed TEXT Порода

Age INTEGER Вік

Gender TEXT Стать

Таблиця Appointments

Зберігає інформацію про візити до клініки.

AppointmentID INTEGER Унікальний ідентифікатор візиту

PatientID INTEGER Ідентифікатор пацієнта (зовнішній ключ)

Date TEXT Дата та час візиту

Reason TEXT Причина візиту

Таблиця Treatments

Зберігає інформацію про лікування та процедури, проведені під час візитів.

TreatmentID INTEGER Унікальний ідентифікатор лікування

AppointmentID INTEGER Ідентифікатор візиту (зовнішній ключ)

Description TEXT Опис лікування або процедури

Cost REAL Вартість лікування

Таблиця Medications

Зберігає інформацію про медикаменти, призначені пацієнтам.

MedicationID INTEGER Унікальний ідентифікатор медикаменту

Name TEXT Назва медикаменту

Dosage TEXT Дозування

Quantity INTEGER Кількість

Таблиця Billing

Зберігає інформацію про фінансові операції та виставлені рахунки.

BillingID INTEGER Унікальний ідентифікатор рахунку

AppointmentID INTEGER Ідентифікатор візиту (зовнішній ключ)

TotalCost REAL Загальна вартість

Paid REAL Сплачена сума

Due REAL Заборгованість

Розроблена база даних на основі SQLite забезпечує надійне та ефективне зберігання всіх необхідних даних для роботи ветеринарної клініки. Вона включає ключові таблиці для обліку пацієнтів, власників, візитів, лікувань, медикаментів та фінансових операцій. Вибрані технології дозволяють швидко

обробляти інформацію, легко інтегрувати базу даних з прикладною програмою та забезпечують зручний доступ до необхідних даних для медичного персоналу клініки.

Було створено шість таблиць та пов'язано їх між собою за допомогою зв'язку один до багатьох (Рисунок 2.1.1). Детальна структура кожної таблиці описана в додатку А.

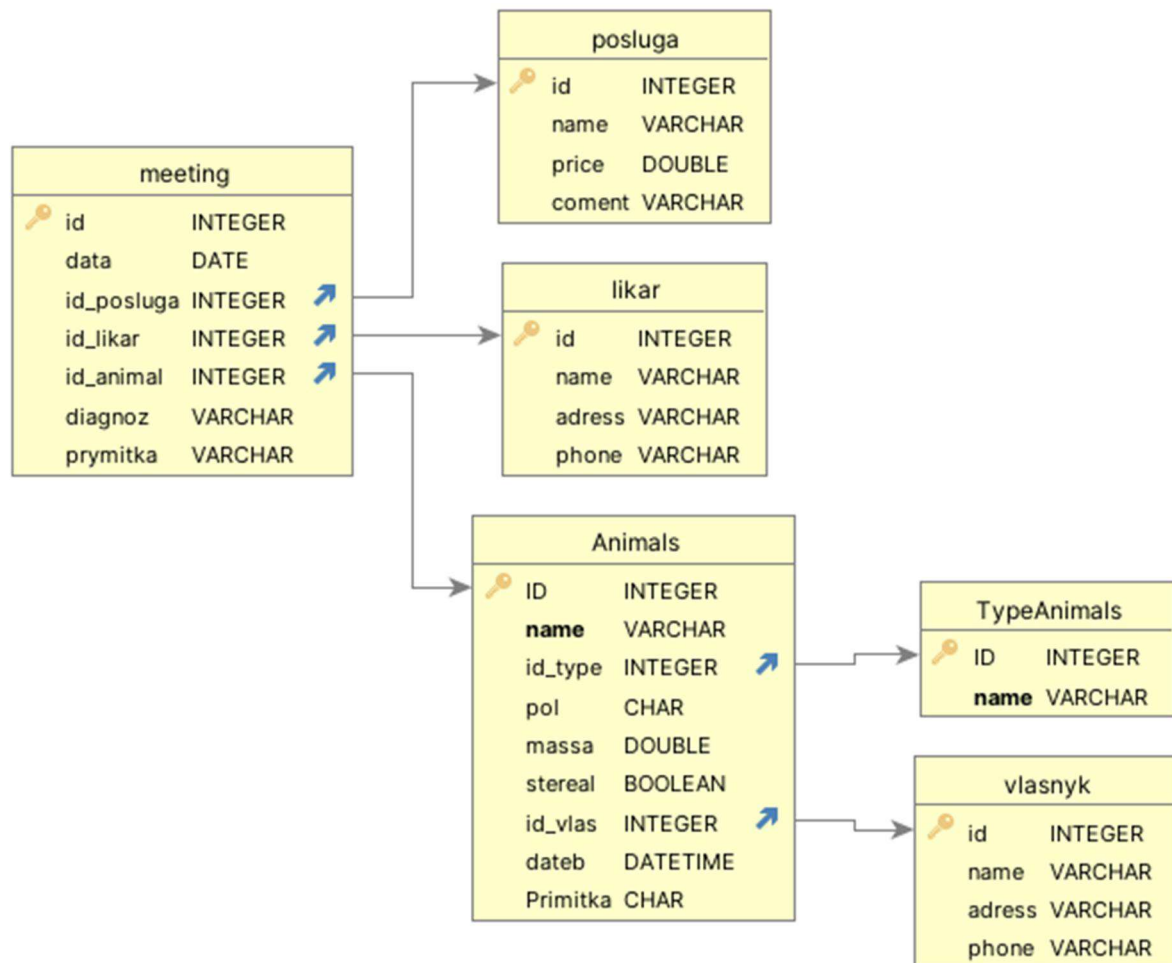


Рисунок 2.1.1 Схема бази даних ветеринарної клініки

2.2. Структура програми

Діаграма класів програмного продукту виглядає таким чином як показано на рисунку 2.2.1.

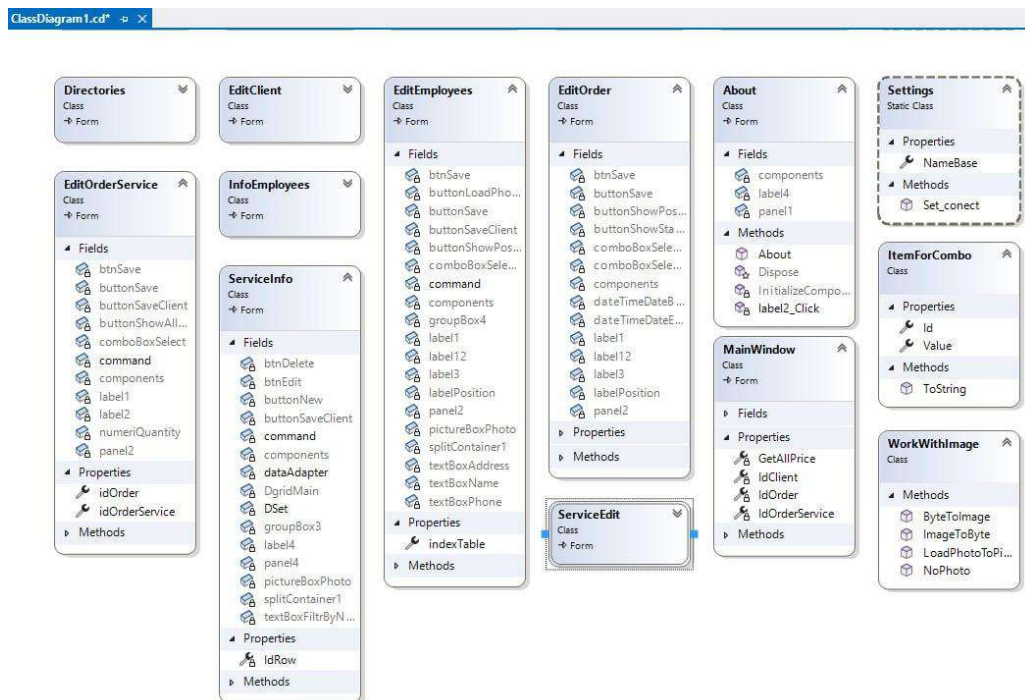


Рис. 2.2.1 – Діаграма класів

На рисунку 2.2.2. зображена структура проєкту.

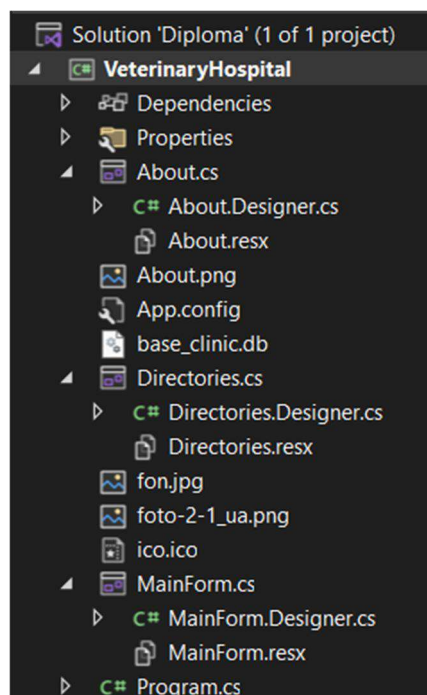


Рис. 2.2.2 – Структура проєкту

Структура таблиць бази даних

На рисунку 2.2.3. зображена структура таблиці Тварини.

base_clinic

Table name: Animals

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	ID	INTEGER								NULL
2	name	VARCHAR (50)								NULL
3	id_type	INTEGER								NULL
4	pol	CHAR								NULL
5	massa	DOUBLE (3, 3)								NULL
6	stereal	BOOLEAN								false
7	id_vlas	INTEGER								NULL
8	dateb	DATETIME								NULL
9	Primitka	CHAR (200)								NULL

Рисунок 2.2.3. – Структура таблиці Animals

На рисунку 2.2.4. зображена структура таблиці Лікар.

base_clinic

Table name: likar

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	id	INTEGER								NULL
2	name	VARCHAR (100)								NULL
3	adress	VARCHAR (100)								NULL
4	phone	VARCHAR (15)								NULL

Рисунок 2.2.4. – Структура таблиці likar

На рисунку 2.2.5. зображена структура таблиці Запис.

base_clinic

Table name: meeting

☐ WITHOUT ROWID

☐ STRICT

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	id	INTEGER								NULL
2	data	DATE								date('now')
3	id_posluga	INTEGER								NULL
4	id_likar	INTEGER								NULL
5	id_animal	INTEGER								NULL
6	diagnoz	VARCHAR (100)								NULL
7	prymitka	VARCHAR (100)								NULL

Рисунок 2.2.5. – Структура таблиці meeting

На рисунку 2.2.6. зображена структура таблиці Послуга.

base_clinic		Table name: posluga	☐ WITHOUT ROWID		☐ STRICT					
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	id	INTEGER								NULL
2	name	VARCHAR (500)								NULL
3	price	DOUBLE (3, 2)								0
4	coment	VARCHAR (250)								NULL

Рисунок 2.2.6. – Структура таблиці posluga

На рисунку 2.2.7. зображена структура таблиці Тип тварини.

base_clinic		Table name: TypeAnimals	☐ WITHOUT ROWID		☐ STRICT					
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	ID	INTEGER								NULL
2	name	VARCHAR (50)								NULL

Рисунок 2.2.7. – Структура таблиці TypeAnimals

На рисунку 2.2.8. зображена структура таблиці Власник.

base_clinic		Table name: vlasnyk	☐ WITHOUT ROWID		☐ STRICT					
	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	Default value
1	id	INTEGER								NULL
2	name	VARCHAR (100)								NULL
3	adress	VARCHAR (100)								NULL
4	phone	VARCHAR (15)								NULL

Рисунок 2.2.8. – Структура таблиці vlasnyk

2.3 Програмна реалізація

Розробка прикладної програми для ветеринарної клініки включає створення графічного інтерфейсу користувача (GUI), обробку даних та інтеграцію з базою даних SQLite. Для реалізації проекту використовуються такі основні технології: мова програмування C#, платформа .NET Framework, Windows Forms для GUI та ORM Entity Framework для роботи з базою даних.

Основні компоненти програми: інтерфейс користувача (GUI); логіка бізнес-процесів; робота з базою даних; інтерфейс користувача (GUI).

Інтерфейс користувача розроблений за допомогою Windows Forms і складається з кількох форм, які дозволяють взаємодіяти з різними даними в базі. Головна форма (MainForm)

```
public partial class MainForm : Form {  
    public MainForm() {  
        InitializeComponent(); }  
    private void btnOwners_Click(object sender, EventArgs e) {  
        OwnersForm ownersForm = new OwnersForm();  
        ownersForm.Show(); }  
    private void btnPatients_Click(object sender, EventArgs e) {  
        PatientsForm patientsForm = new PatientsForm();  
        patientsForm.Show(); }  
    private void btnAppointments_Click(object sender, EventArgs e) {  
        AppointmentsForm appointmentsForm = new AppointmentsForm();  
        appointmentsForm.Show(); }  
    private void btnBilling_Click(object sender, EventArgs e) {  
        BillingForm billingForm = new BillingForm();  
        billingForm.Show(); }  
}
```

Форма власників (OwnersForm)

```

public partial class OwnersForm : Form {
    private VetClinicContext db;
    public OwnersForm() {
        InitializeComponent();
        db = new VetClinicContext();
        LoadOwners(); }
    private void LoadOwners() {
        var owners = db.Owners.ToList();
        dataGridViewOwners.DataSource = owners; }
    private void btnAddOwner_Click(object sender, EventArgs e) {
        Owner owner = new Owner {
            FirstName = txtFirstName.Text,
            LastName = txtLastName.Text,
            PhoneNumber = txtPhoneNumber.Text,
            Email = txtEmail.Text,
            Address = txtAddress.Text };
        db.Owners.Add(owner);
        db.SaveChanges();
        LoadOwners(); } }

```

Основна логіка бізнес-процесів реалізована у вигляді методів, які забезпечують обробку даних та взаємодію між формами і базою даних.

Контекст бази даних (VetClinicContext)

```

public class VetClinicContext : DbContext{
    public DbSet<Owner> Owners { get; set; }
    public DbSet<Patient> Patients { get; set; }
    public DbSet<Appointment> Appointments { get; set; }
    public DbSet<Treatment> Treatments { get; set; }
    public DbSet<Billing> Billings { get; set; }
    protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder) { optionsBuilder.UseSqlite("Data Source=VetClinic.db");}}

```

Модель даних (Owner)

```

public class Owner {
    public int OwnerID { get; set; }
    public string FirstName { get; set; }
    public string LastName { get; set; }
    public string PhoneNumber { get; set; }
    public string Email { get; set; }
    public string Address { get; set; }
    public ICollection<Patient> Patients { get; set; }
}

```

Робота з базою даних забезпечується через Entity Framework, що дозволяє автоматизувати багато аспектів роботи з даними, таких як збереження, оновлення та видалення записів. Додавання даних до бази (метод в OwnersForm)

```

private void btnAddOwner_Click(object sender, EventArgs e) {
    Owner owner = new Owner {
        FirstName = txtFirstName.Text,
        LastName = txtLastName.Text,
        PhoneNumber = txtPhoneNumber.Text,
        Email = txtEmail.Text,
        Address = txtAddress.Text
    };
    db.Owners.Add(owner);
    db.SaveChanges();
    LoadOwners();
}

```

Завантаження даних з бази (метод в OwnersForm)

```

private void LoadOwners() {
    var owners = db.Owners.ToList();
    dataGridViewOwners.DataSource = owners;
}

```

Розробка прикладної програми для ветеринарної клініки включає створення зручного інтерфейсу користувача за допомогою Windows Forms, реалізацію логіки бізнес-процесів для обробки даних та інтеграцію з базою даних SQLite через Entity Framework. Такий підхід забезпечує високу продуктивність, надійність та зручність використання програми, що дозволяє ефективно керувати інформацією у ветеринарній клініці та покращити якість обслуговування пацієнтів.

2.4 Опис класів

В кодї програми зустрічаються наступні класи:

Клас Owner зберігає інформацію про власників тварин, включаючи їх ім'я, прізвище, номер телефону, електронну адресу та адресу проживання.

Клас Patient представляє тварин, які відвідують клініку. Зберігає інформацію про ім'я тварини, вид, породу, вік, стать та ідентифікатор власника.

Appointment цей клас зберігає дані про візити пацієнтів до клініки. Включає дату візиту, причину візиту та ідентифікатор пацієнта.

Treatment в цьому класі зберігається інформація про проведене лікування під час візитів. Включає опис лікування, вартість та ідентифікатор візиту.

Billing цей клас відповідає за зберігання даних про фінансові операції, пов'язані з візитами та лікуванням. Містить загальну вартість, сплачену суму, суму, що залишилася до сплати, та ідентифікатор візиту.

VetClinicContext тут зберігається контекст бази даних, який використовується для взаємодії з базою даних SQLite. Містить набори даних для кожної моделі (Owners, Patients, Appointments, Treatments, Billings) і налаштування з'єднання з базою даних.

OwnerService це клас виступає як сервіс для обробки даних, пов'язаних з власниками тварин. Виконує операції додавання, видалення, оновлення та отримання інформації про власників.

MainForm клас що відповідає за головну форму програми, яка забезпечує навігацію між різними секціями програми (власники, пацієнти, візити, лікування, рахунки).

OwnersForm клас що відповідає за форму для взаємодії з даними про власників. Дозволяє переглядати, додавати, редагувати та видаляти інформацію про власників.

Ці класи забезпечують основні функціональні можливості програми для ветеринарної клініки, включаючи зберігання даних, обробку бізнес-логіки та

					ДР.122.041.030.ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

надання графічного інтерфейсу для користувачів. Кожен клас має чітко визначене призначення, що допомагає організувати код і підтримувати його зрозумілість та зручність у використанні.

Висновки до розділу 2

Висновки до розділу проектування та розробки мають ключове значення, оскільки вони підсумовують весь процес створення програмного продукту та відображають досягнуті результати.

В процесі проектування було ретельно розглянуто структуру програми, визначено основні класи, їх взаємозв'язки та функціональні обов'язки. Це дозволило створити гнучку та масштабовану архітектуру, яка легко піддається змінам та розширенням.

Процес розробки функціональності був систематичним та послідовним.

Дизайн графічного інтерфейсу користувача був розроблений з урахуванням простоти використання та зручності для кінцевих користувачів. Чітка структура та інтуїтивно зрозумілі елементи керування забезпечують зручну роботу з програмою.

Проектування і розробка прикладної програми для ветеринарної лікарні є важливим етапом, що передбачає аналіз потреб користувачів, проектування архітектури програмного забезпечення, розробку функціональності та тестування системи.

На початковому етапі проводиться аналіз потреб ветеринарних клінік та їх клієнтів. Визначаються основні функції програми, такі як облік власників тварин, візити, лікування та фінансовий облік.

Створюється детальний план структури програми, включаючи розподіл функцій між різними класами та модулями, визначення бази даних та інтерфейсу користувача.

					ДР.122.041.030.ПЗ	Арк.
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

На основі проектування розробляється функціональність програми. Цей етап включає написання коду, реалізацію бази даних, створення графічного інтерфейсу користувача та інші технічні аспекти.

Після успішного завершення тестування програма готова до впровадження в експлуатацію. Важливо також забезпечити підтримку програмного забезпечення, включаючи виправлення помилок та вдосконалення функціональності з часом.

В цілому, проектування і розробка прикладної програми для ветеринарної лікарні є складним, але дуже важливим процесом, який дозволяє створити ефективний та корисний інструмент для ветеринарних клінік та їх клієнтів.

Отже, висновки до розділу проектування та розробки підкреслюють успішність процесу створення програмного продукту, а також вказують на досягнення поставлених цілей та відповідність розробленої системи вимогам та очікуванням користувачів.

					ДР.122.041.030.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. КЕРІВНИЦТВО ВИКОРИСТАННЯ ПРОГРАМНОГО ПРОДУКТУ

3.1. Мінімальні вимоги до системи

Для ефективної роботи прикладної програми для ветеринарної лікарні мінімальні вимоги до системи має включати:

Операційна система Windows 7 або новіша версія, або альтернативна операційна система, яка підтримує використання .NET. Мінімум двоядерний процесор з тактовою частотою 1.6 ГГц або вище. Рекомендується мінімум 4 ГБ оперативної пам'яті.

Вільне місце на диску: Потрібно достатньо вільного місця на диску для встановлення програми та збереження даних. Рекомендується мінімум 1 ГБ вільного простору. Мінімально рекомендований роздільна здатність екрану 1024x768 пікселів.

					ДР.122.041.030.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2. Інтерфейс користувача

На рисунку зображено діаграму за допомогою якої можна описати взаємодію користувача з інформаційною системою (Рисунок 3.2.1.)

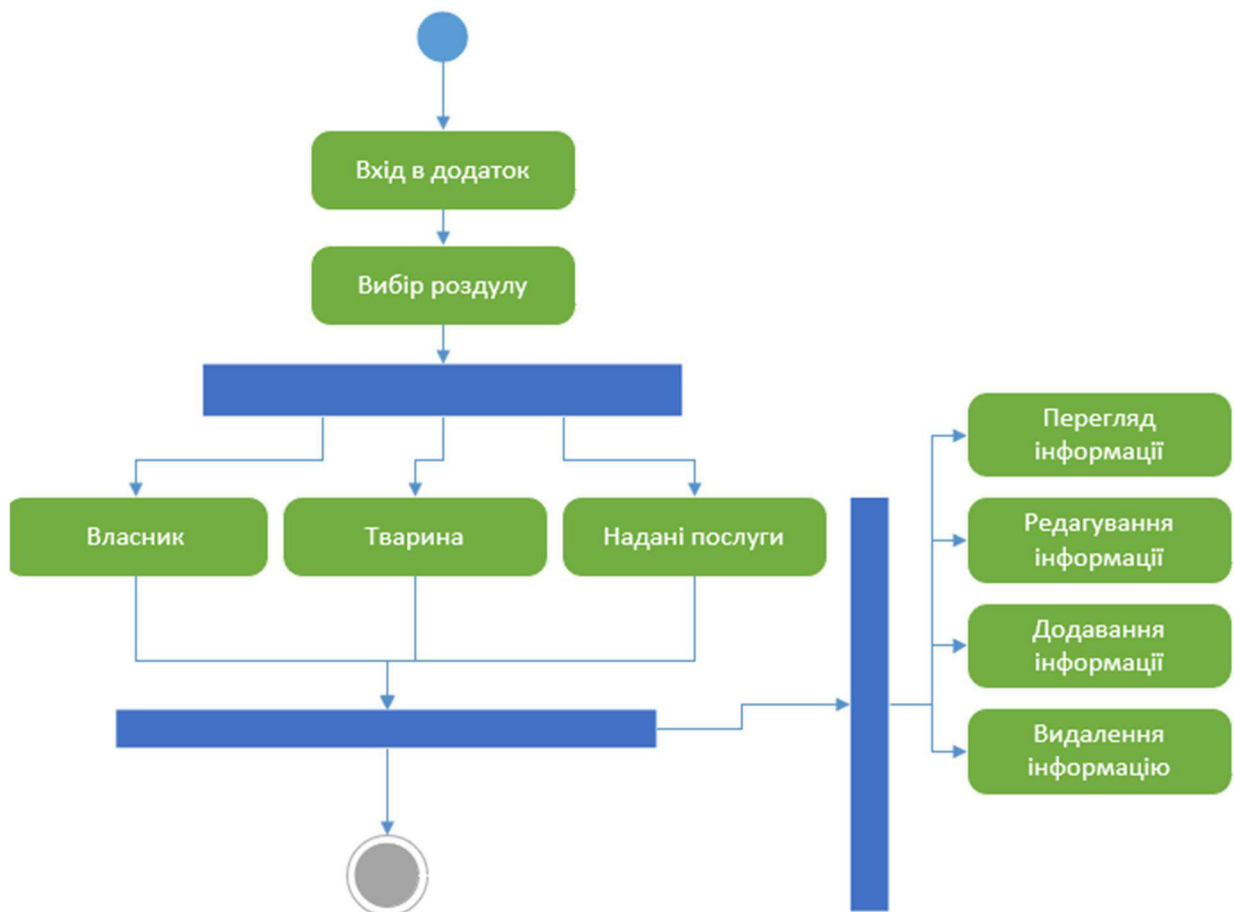


Рисунок 3.2.1 – Діаграма активності.

Для реалізації інтерфейсу програми було обрано середовище розробки Microsoft Visual Studio та технологію Windows Forms. Вона дозволяє створювати графічний інтерфейс користувача і є частиною Microsoft .NET, .NET Framework. Даний інтерфейс спрощує доступ до елементів інтерфейсу Microsoft Windows за допомогою створення обгортки для Win32 API в керованому коді.

При запуску програми відкривається вікно для роботи з власниками тварин (Рисунок 3.2.2). На вкладці можна лише переглядати власників та виконувати пошук за прізвищем або за адресом проживання.

Код	ПІБ	Адреса	Телефон
1	Іваненко Іван Іванович	Житомир, Наливайка 135	+380963692585
2	Петренко Петро Петрович	Київ, Хрещатик 26	239638
3	Тарарай Микола Іванович	Житомирю вул. Перемоги 145	362536
4	Марченко Олександр Олександрович	Житомир, Мануїльського 36	+380663595689
5	Степаненко Степан Степанович	Львів	103

Рисунок 3.2.2 – Інтерфейс програми для роботи з власниками

Для роботи з власниками я створила додаткову форму (Рисунок 3.2.3 Рисунок 3.2.3). Вона має назву «Довідник «Власники тварин»». Вона дозволяє:

- додавання нового;
- редагувати наявного;
- видаляти власника тварини.

Код	ПІБ	Адреса	Телефон
1	Іваненко Іван Іванович	Житомир, Наливайка 135	+380963692585
2	Петренко Петро Петрович	Київ, Хрещатик 26	239638
3	Тарарай Микола Іванович	Житомирю вул. Перемоги 145	362536
4	Марченко Олександр Олександрович	Житомир, Мануїльського 36	+380663595689
*			

Рисунок 3.2.3 – Довідник «Власники тварин»

Для роботи з тваринами потрібно в головному вікні перейти до вкладки «Тварини» (Рисунок 3.2.4.). Дана вкладка відображає лише тварин, які має

вибраний власник на вкладці «Власник». На вкладці можна додати нову, видалити або змінити тварину.

Рисунок 3.2.4 – Інтерфейс програми для роботи з тваринами

На вкладці можна додати нову, видалити або змінити тварину. При додаванні нової тварини потрібно вибрати вид тварини зі списку що розкривається. Якщо в списку немає необхідного вид, то слід піти в меню і вибрати довідник «Види тварин» (Рисунок 3.2.5.). В довіднику ви можете додати новий вид, видалити не потрібний вид або відредагувати наявного.

Рисунок 3.2.5 – Довідник «Види тварин»

Для роботи з власниками я створила додаткову форму (Рисунок 3.2.6.). Вона має назву «Довідник «Власники тварин»». Вона дозволяє:

- додавання нового;
- редагувати наявного;
- видаляти власника тварини.

Довідник "Власники"				
Новий запис		Видалити запис		Зберегти зміни
	Код	ПІБ	Адреса	Телеф
▶	1	Іваненко Іван Іванович	Житомир. Наливайка 135	+380...
	2	Петренко Петро Петрович	Київ, Хрещатик 26	2396...
	3	Тарарай Микола Іванович	Житомирю вул. Перемоги 145	3625...
	4	Марченко Олександр Олександрович	Житомир. Мануїльського 36	+380...
	5	Степаненко Степан Степанович	Львів	103
*				

Рисунок 3.2.6 – Довідник «Власники тварин»

Якщо потрібно надати послуги або просто переглянути надані послуги потрібно в головному вікні перейти до вкладки «Надані послуги» (Рисунок 3.2.7.). Дана вкладка дозволяє виконувати дії з послугами: змінити, видалити, редагувати.

Ветеринарна клініка																	
Довідники ?																	
																	
Власник		Тварини		Надані послуги													
Тварина		миша Кличка "Сіра" (Д)		323,85 грн.		Пошук за місяцем											
						Пошук послуги											
Код	Дата	Діагноз	Послуга	Ціна	Лікар	Примітка											
1	21.07.2023	Укуси кліщів	Видалення кліщів	23,85	Хомутовський Андрій Борисович	Видалено 5 штук											
2	04.10.2023	Отруєння	Пероральна дача препаратів	300	Бугай Петро Федотович												
Виберіть дію ☒ Нова послуга ☐ Редагувати вибрану Зберегти Видалити ДАНІ ПРО ПОСЛУГУ Лікар Діагноз Дата прийому 18.11.2023 Перелік можливих послуг <table border="1"> <tr> <th>Послуга</th> <th>Ціна</th> </tr> <tr> <td>Вакцинація від сибірки</td> <td>25,36</td> </tr> <tr> <td>Вакцинація від сказу</td> <td>100</td> </tr> <tr> <td>Видалення кліщів</td> <td>23,85</td> </tr> <tr> <td>Виклик ветеринарного лікаря</td> <td>100</td> </tr> </table>								Послуга	Ціна	Вакцинація від сибірки	25,36	Вакцинація від сказу	100	Видалення кліщів	23,85	Виклик ветеринарного лікаря	100
Послуга	Ціна																
Вакцинація від сибірки	25,36																
Вакцинація від сказу	100																
Видалення кліщів	23,85																
Виклик ветеринарного лікаря	100																

Рисунок 3.2.7 – Інтерфейс програми для роботи з послугами

При наданні послуги потрібно вибрати зі списку що розкривається ветеринара. Якщо ветеринара немає в списку, його потрібно додати через

меню «Довідники» вибрати «Ветеринари клініки» (Рисунок 3.2.8.). Довідник ветеринарів дозволяє виконувати операції по додаванні нового чи то редагувати вибраного ветеринара.

Довідник "Ветеринари клініки" — □ ×

	Код	ПІБ	Адреса	Телефон
▶	1	Бугай Петро Федотович	Коростень	1036665
	2	Теля Марія Антонівна	Переброди 45	412563
	3	Макаренко Іван Олександрович	Житомир Селецька 147	+380972516705
	4	Хомутовський Андрій Борисович	Житомир, Київська 32	+380661234587
*				

Новий запис

Видалити запис

Зберегти зміни

Рисунок 3.2.8 – Довідник «Ветеринари клініки»

Якщо є потреба у новій послугі потрібно або потрібно змінити наявну, то потрібно піти до меню «Довідники» та вибрати «Послуги клініки» (Рисунок 3.2.9.). Даний довідник дозволяє виконувати операції по додаванні нової послуги, редагуванні поточної або видаляти послугу.

Довідник "Послуги"

Новий запис Видалити запис Зберегти зміни

	Код	ПІБ
▶	1	Вакцинація від сибірки
	2	Видалення кліщів
	3	Первинний прийом та клінічний огляд
	4	Прийом повторний
	5	Інєкція підшкірна або внутрішньо мязова
	6	Внутрішньо венна інєкція
	7	Пероральна дача препаратів
	8	Ректальне введення препаратів
	9	Виклик ветеринарного лікаря
	10	Комплексне щеплення проти чуми, аденовірозу (I, II типів), парвовірозу, па
	11	Комплексна вакцинація собак без антирабічного компонента
	12	Щеплення проти чуми, аденовірозу (I, II типів), парвовірозу, парагрипу та л
	13	Щеплення проти чуми, аденовірозу (I, II типів), парвовірозу, парагрипу та л

Рисунок 3.2.9 – Довідник «Послуги клініки»

З меню можна відкрити вікно для перегляду інформації про автора програми (Рисунок 3.2.10.).

Про програму

**Дипломна робота
студента групи П-41
Федорчука Роман Олександровича
на тему:
"Розробка прикладної програми для
ветеринарної лікарні"
2024**



Рисунок 3.2.10 – Вікно «Про програму»

3.3 Тестування програмного продукту

Ось кілька прикладів тестів для додатку Windows Form для ветеринарної клініки з використанням бази даних:

Тест додавання нового клієнта:

Крок 1: Відкрити форму додавання нового клієнта.

Крок 2: Ввести дані нового клієнта в поля форми.

Крок 3: Натиснути кнопку "Зберегти".

Очікуваний результат: Новий клієнт доданий до бази даних та відображений у списку клієнтів.

Тест видалення існуючої тварини:

Крок 1: Відкрити форму перегляду інформації про тварину.

Крок 2: Вибрати тварину, яку потрібно видалити.

Крок 3: Натиснути кнопку "Видалити".

Очікуваний результат: Тварина видалена з бази даних та більше не відображається у списку тварин.

Тест пошуку клієнта за ім'ям:

Крок 1: Відкрити форму пошуку клієнта.

Крок 2: Ввести ім'я або прізвище клієнта в поле пошуку.

Крок 3: Натиснути кнопку "Пошук".

Очікуваний результат: Список клієнтів з ім'ям або прізвищем, що введені, відображений у вікні пошуку.

Тест оновлення медичної історії тварини:

Крок 1: Відкрити форму перегляду медичної історії тварини.

Крок 2: Вибрати тварину та переглянути її медичну історію.

Крок 3: Внести зміни до медичної історії (наприклад, додати новий візит або змінити діагноз).

Крок 4: Натиснути кнопку "Зберегти".

Очікуваний результат: Зміни внесені до медичної історії тварини збережені в базі даних.

Тест безпеки: SQL-ін'єкція:

Крок 1: Спробувати ввести в поле пошуку спеціальний SQL-запит.

Крок 2: Натиснути кнопку "Пошук".

Очікуваний результат: Система має відхилити запит і показати повідомлення про помилку, а не виконати небезпечний SQL-запит.

Ці приклади тестів допоможуть перевірити різні аспекти функціональності та безпеки вашого додатку та його взаємодії з базою даних.

Ці приклади тестів допоможуть перевірити різні аспекти функціональності та безпеки вашого додатку та його взаємодії з базою даних.

Висновки до розділу 3

У цьому розділі дипломної роботи ми розглянули процес розробки прикладної програми для ветеринарної клініки з використанням десктопної платформи Windows та бази даних SQLite. На основі аналізу інформаційних потреб ветеринарних клінік були сформульовані вимоги до програмного забезпечення та визначені його основні функції.

Для досягнення поставленої мети було проведено огляд та порівняння аналогічних систем, вибрано найбільш підходящі технології розробки, а також розроблено базу даних на основі SQLite для зберігання інформації про клієнтів, тварин та їх медичних записів.

Програмна реалізація була проведена за допомогою мови програмування C# та платформи Windows Form, що дозволило створити зручний та інтуїтивно зрозумілий інтерфейс користувача. Були розроблені різні класи для взаємодії з базою даних, включаючи класи для додавання, оновлення та видалення записів.

Для забезпечення якості програмного забезпечення було проведено тестування додатку, включаючи тести функціональності, тести взаємодії з базою даних та тести безпеки. Результати тестування дозволили переконатися в коректності та надійності програми.

					ДР.122.041.030.ПЗ	Арк.
						38
Змн.	Арк.	№ докум.	Підпис	Дата		

Завершальний етап роботи полягав у підготовці керівництва з використання програмного продукту, в якому надано інструкції щодо встановлення програми, навігації в інтерфейсі, роботи з базою даних та інші важливі аспекти використання програми.

У результаті цієї роботи було успішно розроблено та протестовано програмне забезпечення для ветеринарної клініки, яке дозволить зручно та ефективно вести медичну документацію про тварин та їх власників.

					ДР.122.041.030.ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У цій дипломній роботі була проведена розробка прикладної програми для ветеринарної клініки на платформі Windows з використанням бази даних SQLite. Процес розробки включав аналіз інформаційних потреб ветеринарних клінік, вибір технологій розробки, проектування та розробку програмного продукту, а також тестування та створення керівництва з використання.

В результаті виконаної роботи були досягнуті такі ключові результати:

Розроблено програмний продукт, який відповідає вимогам та потребам ветеринарних клінік. Програма дозволяє зберігати та управляти інформацією про клієнтів, їх тварин та медичні записи.

Використано сучасні технології розробки, такі як мова програмування C#, платформа Windows Form та база даних SQLite, що дозволило створити надійний та ефективний програмний продукт.

Проведено тестування програми з використанням різних видів тестів, що дозволило перевірити його функціональність, взаємодію з базою даних.

Розроблено керівництво з використання програмного продукту, яке надає користувачам зрозумілі інструкції щодо встановлення та використання програми.

У цілому, розроблений програмний продукт може стати корисним інструментом для ветеринарних клінік, що дозволить їм ефективно вести медичну документацію та покращити якість надання ветеринарних послуг.

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

Перелік літературних джерел

1. ARIS [Електронний ресурс] – Режим доступу до ресурсу:
<https://www.kpms.ru/Automatization/ARIS.htm>.
2. CA ERwin Data Modeler [Електронний ресурс] – Режим доступу до ресурсу: https://www.administrator-pro.ru/cat/soft/ca_technologies/ca_erwin_data_modeler/.
3. CA ERwin Process Modeler [Електронний ресурс] – Режим доступу до ресурсу: <https://www.architect-design.ru/item.1011.html>.
4. CASE-технологии [Електронний ресурс] – Режим доступу до ресурсу: <https://piter-soft.ru/knowledge/glossary/process/case-tehnology.html>.
5. CRM Appointer [Електронний ресурс] – Режим доступу до ресурсу: <https://appointer.ua/>.
6. EasyWeek [Електронний ресурс] – Режим доступу до ресурсу: <https://easyweek.com.ua/programa-dlya-vetklinik.html>.
7. Rational Rose [Електронний ресурс] – Режим доступу до ресурсу: https://www.kpms.ru/Automatization/Rational_Rose.htm.
8. SkyService POS [Електронний ресурс] – Режим доступу до ресурсу: <https://skyservice.pro/>.
9. Все виды ветеринарных услуг [Електронний ресурс] – Режим доступу до ресурсу: <https://zoovet74.ru/kontakty/o-nas/chelyabinskaya-veterinarnaya-klinika-zoovet>.
10. Закон України «Про ветеринарну медицину» від 25.06.1992 р. №36. Відомості Верховної Ради України. Дата оновлення: 21.03.2021 р. [Електронний ресурс]. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/2498-12#Text>.
11. Інформаційні технології та моделювання бізнес-процесів / О. М. Томашевський, Г. Г. Цегелик, М. Б. Вітер, В. І. Дубук. – Київ: Видавництво «Центр учбової літератури», 2012. – 296 с.

					<i>ДР.122.041.030.ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

- 12.Кастрация и стерилизация животных [Электронный ресурс] – Режим доступа до ресурсу: http://drug.zp.ua/p_kastraciya-sterilizaciya.html.
- 13.Круглосуточный стационар [Электронный ресурс] – Режим доступа до ресурсу: http://drug.zp.ua/p_stacionar.html.
- 14.Лабораторные исследования. Лабораторные исследования для животных [Электронный ресурс] – Режим доступа до ресурсу: http://drug.zp.ua/p_laboratornye-issledovaniya.html.
- 15.МЕТОДОЛОГИЯ DFD [Электронный ресурс] – Режим доступа до ресурсу: https://www.sites.google.com/site/anisimovkhv/learning/pris/lecture/tema6/tema6_3#p68.
- 16.Методология моделирования процессов IDEF3 [Электронный ресурс] – Режим доступа до ресурсу: <http://asu.ugatu.ac.ru/library/56c1644b436bb/80ecddf06ad1fd912d8d973b0ec29b40.pdf>.
- 17.Основные методологии обследования организаций. Стандарт IDEF0. [Электронный ресурс] – Режим доступа до ресурсу: <https://www.cfin.ru/vernikov/idef/idef0.shtml>.
- 18.Прием ветеринара [Электронный ресурс] – Режим доступа до ресурсу: http://drug.zp.ua/p_priem-veterinara.html.
- 19.Регистрация и вакцинация [Электронный ресурс] – Режим доступа до ресурсу: http://drug.zp.ua/p_registraciya-i-vakcinaciya.html.
- 20.Рентген для животных [Электронный ресурс] – Режим доступа до ресурсу: http://drug.zp.ua/p_rentgen.html.

ДОДАТКИ

Додаток А

Програмный код модуля EditClient

```
namespace VeterinaryClinic
{
    partial class MainForm
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        /// <summary>
        /// Требуемый метод для поддержки конструктора — не изменяйте
        /// содержимое этого метода с помощью редактора кода.
        System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle1
        = new System.Windows.Forms.DataGridViewCellStyle();
        System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle2
        = new System.Windows.Forms.DataGridViewCellStyle();
        System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle3
        = new System.Windows.Forms.DataGridViewCellStyle();
        System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle4
        = new System.Windows.Forms.DataGridViewCellStyle();
        System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle5
        = new System.Windows.Forms.DataGridViewCellStyle();
        System.Windows.Forms.DataVisualization.Charting.ChartArea chartArea1
        = new System.Windows.Forms.DataVisualization.Charting.ChartArea();
        System.Windows.Forms.DataVisualization.Charting.Legend legend1 = new
```

					ДР.122.041.030.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

```

textBox1 = new System.Windows.Forms.TextBox();
tabPage2 = new System.Windows.Forms.TabPage();
panel4 = new System.Windows.Forms.Panel();
groupBox2 = new System.Windows.Forms.GroupBox();
radioButton1 = new System.Windows.Forms.RadioButton();
radioButton2 = new System.Windows.Forms.RadioButton();
textBox5 = new System.Windows.Forms.TextBox();
groupBox1 = new System.Windows.Forms.GroupBox();
label9 = new System.Windows.Forms.Label();
comboBoxTypAn = new System.Windows.Forms.ComboBox();
label8 = new System.Windows.Forms.Label();
label7 = new System.Windows.Forms.Label();
label5 = new System.Windows.Forms.Label();
checkBox1 = new System.Windows.Forms.CheckBox();
dateTimePicker1 = new System.Windows.Forms.DateTimePicker();
radioEdit = new System.Windows.Forms.RadioButton();
radioInsert = new System.Windows.Forms.RadioButton();
SaveAnimal = new System.Windows.Forms.Button();
DGVanimals = new System.Windows.Forms.DataGridView();
panel3 = new System.Windows.Forms.Panel();
label3 = new System.Windows.Forms.Label();
label4 = new System.Windows.Forms.Label();
tabPage3 = new System.Windows.Forms.TabPage();
panel5 = new System.Windows.Forms.Panel();
groupBox4 = new System.Windows.Forms.GroupBox();
tabControl2 = new System.Windows.Forms.TabControl();
tabPage4 = new System.Windows.Forms.TabPage();
textBoxPosPrim = new System.Windows.Forms.TextBox();
label10 = new System.Windows.Forms.Label();
comboBoxLikar = new System.Windows.Forms.ComboBox();

```

```

label13 = new System.Windows.Forms.Label();
dateTimePicker2 = new System.Windows.Forms.DateTimePicker();
label14 = new System.Windows.Forms.Label();
textBoxDiagnoz = new System.Windows.Forms.TextBox();
groupBox6 = new System.Windows.Forms.GroupBox();
button2 = new System.Windows.Forms.Button();
radioB_pos = new System.Windows.Forms.RadioButton();
radioButton6 = new System.Windows.Forms.RadioButton();
button3 = new System.Windows.Forms.Button();
DGVposlugi = new System.Windows.Forms.DataGridView();
panel6 = new System.Windows.Forms.Panel();
tabPageStatistics = new System.Windows.Forms.TabPage();
chart1 = new System.Windows.Forms.DataVisualization.Charting.Chart();
panel1 = new System.Windows.Forms.Panel();
labelAllStatistic = new System.Windows.Forms.Label();
labelEnd = new System.Windows.Forms.Label();
labelStart = new System.Windows.Forms.Label();
dateTimePickerEnd = new System.Windows.Forms.DateTimePicker();
dateTimePickerStart = new System.Windows.Forms.DateTimePicker();
label7.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label7.Location = new System.Drawing.Point(14, 118);
label7.Name = "label7";
label7.Size = new System.Drawing.Size(34, 13);
label7.TabIndex = 26;
label5.AutoSize = true;
label5.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label5.Location = new System.Drawing.Point(14, 58);
label5.Name = "label5";
label5.Size = new System.Drawing.Size(43, 13);
label5.TabIndex = 23;

```

```

checkBox1.AutoSize = true;
checkBox1.CheckAlign
System.Drawing.ContentAlignment.MiddleRight;
checkBox1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
checkBox1.Location = new System.Drawing.Point(178, 120);
checkBox1.Name = "checkBox1";
checkBox1.Size = new System.Drawing.Size(90, 17);
checkBox1.TabIndex = 29;
checkBox1.Text = "Стерелізація";
dateTimePicker1.Font = new System.Drawing.Font("Microsoft Sans Serif",
8F);
dateTimePicker1.Format
System.Windows.Forms.DateTimePickerFormat.Short;
dateTimePicker1.Location = new System.Drawing.Point(161, 81);
dateTimePicker1.Name = "dateTimePicker1";
dateTimePicker1.Size = new System.Drawing.Size(132, 20);
massatb.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
massatb.Location = new System.Drawing.Point(76, 118);
massatb.Name = "massatb";
massatb.Size = new System.Drawing.Size(93, 20);
label6.AutoSize = true;
label6.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label6.Location = new System.Drawing.Point(14, 84);
label6.Name = "label6";
groupBox3.Location = new System.Drawing.Point(10, 10);
groupBox3.Name = "groupBox3";
groupBox3.Size = new System.Drawing.Size(216, 179);
groupBox3.TabIndex = 24;
groupBox3.TabStop = false;
DeleteAnimal.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F,

```

```

System.Drawing.FontStyle.Bold);
DeleteAnimal.Location = new System.Drawing.Point(32, 112);
DeleteAnimal.Name = "DeleteAnimal";
DeleteAnimal.Size = new System.Drawing.Size(127, 29);
DeleteAnimal.TabIndex = 20;
DeleteAnimal.Text = "Видалити вибрану";
DeleteAnimal.UseVisualStyleBackColor = true;
radioEdit.AutoSize = true;
radioEdit.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
radioEdit.Location = new System.Drawing.Point(6, 45);
radioEdit.Name = "radioEdit";
radioEdit.Size = new System.Drawing.Size(127, 17);
dataGridViewCellStyle3.SelectionBackColor =
System.Drawing.SystemColors.Highlight;
dataGridViewCellStyle3.SelectionForeColor =
System.Drawing.SystemColors.HighlightText;
dataGridViewCellStyle3.WrapMode =
System.Windows.Forms.DataGridViewTriState.False;
DGVanimals.DefaultCellStyle = dataGridViewCellStyle3;
DGVanimals.Dock = System.Windows.Forms.DockStyle.Fill;
DGVanimals.Location = new System.Drawing.Point(3, 34);
DGVanimals.Name = "DGVanimals";
DGVanimals.ReadOnly = true;
DGVanimals.RowHeadersWidth = 51;
DGVanimals.Size = new System.Drawing.Size(970, 367);
DGVanimals.TabIndex = 5;
panel3.Controls.Add(label3);
panel3.Controls.Add(label4);
panel3.Dock = System.Windows.Forms.DockStyle.Top;
panel3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);

```

```

label3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label3.ForeColor = System.Drawing.Color.Red;
label3.Location = new System.Drawing.Point(124, 7);
label3.Name = "label3";
label3.Size = new System.Drawing.Size(0, 13);
label4.AutoSize = true;
label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label4.Location = new System.Drawing.Point(5, 7);
label4.Name = "label4";
label4.Size = new System.Drawing.Size(88, 13);
label4.TabIndex = 1;
tabPage3.Controls.Add(panel5);
tabPage3.Controls.Add(DGVposlugi);
tabPage3.Controls.Add(panel6);
tabPage3.Location = new System.Drawing.Point(4, 29);
tabPage3.Name = "tabPage3";
tabPage3.Size = new System.Drawing.Size(976, 404);
tabPage3.TabIndex = 2;
panel5.Controls.Add(groupBox4);
panel5.Controls.Add(groupBox6);
panel5.Dock = System.Windows.Forms.DockStyle.Bottom;
panel5.Location = new System.Drawing.Point(0, 194);
listView1.GridLines = true;
listView1.HeaderStyle
System.Windows.Forms.ColumnHeaderStyle.Nonclickable;
listView1.Location = new System.Drawing.Point(3, 3);
listView1.Sorting = System.Windows.Forms.SortOrder.Ascending;
listView1.TabIndex = 35;
listView1.UseCompatibleStateImageBehavior = false;
listView1.View = System.Windows.Forms.View.Details;

```

```

listView1.DrawColumnHeader += listView1_DrawColumnHeader;
listView1.DrawItem += listView1_DrawItem;
columnHeader1.Text = "Послуга";
columnHeader2.Text = "Ціна";
columnHeader2.TextAlign =
System.Windows.Forms.HorizontalAlignment.Center;
tabPage5.Name = "tabPage5";
tabPage5.Padding = new System.Windows.Forms.Padding(3);
tabPage5.Size = new System.Drawing.Size(467, 130);
tabPage5.TabIndex = 1;
tabPage5.Text = "Примітка до наданої послуги";
tabPage5.UseVisualStyleBackColor = true;
textBoxPosPrim.Multiline = true;
textBoxPosPrim.Name = "textBoxPosPrim";
textBoxPosPrim.ScrollBars = System.Windows.Forms.ScrollBars.Both;
textBoxPosPrim.Size = new System.Drawing.Size(461, 124);
textBoxPosPrim.TabIndex = 34;
label10.AutoSize = true;
label10.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label10.Location = new System.Drawing.Point(13, 34);
label10.Name = "label10";
label10.Size = new System.Drawing.Size(35, 13);
label10.TabIndex = 31;
comboBoxLikar.FormattingEnabled = true;
comboBoxLikar.Location = new System.Drawing.Point(70, 31);
comboBoxLikar.Name = "comboBoxLikar";
comboBoxLikar.Size = new System.Drawing.Size(188, 21);
comboBoxLikar.TabIndex = 30;
label13.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label13.Location = new System.Drawing.Point(13, 62);

```

```

label13.Name = "label13";
label13.Size = new System.Drawing.Size(47, 13);
label13.TabIndex = 23;
label13.Text = "Діагноз";
System.Windows.Forms.DateTimePickerFormat.Short;
dateTimePicker2.Location = new System.Drawing.Point(121, 98);
dateTimePicker2.Name = "dateTimePicker2";
dateTimePicker2.Size = new System.Drawing.Size(137, 20);
dateTimePicker2.TabIndex = 27;
label14.AutoSize = true;
label14.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
label14.Location = new System.Drawing.Point(13, 98);
label14.Name = "label14";
label14.Size = new System.Drawing.Size(79, 13);
label14.TabIndex = 24;
label14.Text = "Дата прийому";
textBoxDiagnoz.Font = new System.Drawing.Font("Microsoft Sans Serif",
8F);
textBoxDiagnoz.Location = new System.Drawing.Point(70, 59);
textBoxDiagnoz.Name = "textBoxDiagnoz";
textBoxDiagnoz.Size = new System.Drawing.Size(188, 20);
textBoxDiagnoz.TabIndex = 22;
groupBox6.Controls.Add(button2);
groupBox6.Controls.Add(radioButton_pos);
groupBox6.Controls.Add(radioButton6);
groupBox6.Controls.Add(button3);
groupBox6.Dock = System.Windows.Forms.DockStyle.Left;
groupBox6.Location = new System.Drawing.Point(10, 10);
groupBox6.Name = "groupBox6";
groupBox6.Size = new System.Drawing.Size(194, 190);

```

```

groupBox6.TabIndex = 24;
groupBox6.TabStop = false;
groupBox6.Text = "Виберіть дію";
button2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F,
System.Drawing.FontStyle.Bold);
button2.Location = new System.Drawing.Point(21, 116);
button2.Name = "button2";
button2.Size = new System.Drawing.Size(132, 27);
button2.TabIndex = 20;
button2.Text = "Видалити вибрану";
button2.UseVisualStyleBackColor = true;
button2.Click += buttonDeletePosl_Click;
radioB_pos.AutoSize = true;
radioB_pos.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
radioB_pos.Location = new System.Drawing.Point(8, 64);
radioB_pos.Name = "radioB_pos";
radioB_pos.Size = new System.Drawing.Size(127, 17);
radioB_pos.TabIndex = 19;
radioB_pos.Text = "Редагувати вибрану";
radioB_pos.UseVisualStyleBackColor = true;
radioB_pos.CheckedChanged += radioEditPos_CheckedChanged;
radioButton6.AutoSize = true;
radioButton6.Checked = true;
radioButton6.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
radioButton6.Location = new System.Drawing.Point(8, 37);
radioButton6.Name = "radioButton6";
radioButton6.Size = new System.Drawing.Size(94, 17);
radioButton6.TabIndex = 18;
radioButton6.TabStop = true;
radioButton6.Text = "Нова послуга";

```

```

radioButton6.UseVisualStyleBackColor = true;
button3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F,
System.Drawing.FontStyle.Bold);
button3.Location = new System.Drawing.Point(21, 91);
button3.Name = "button3";
button3.Size = new System.Drawing.Size(131, 24);
button3.TabIndex = 5;
button3.Text = "Зберегти послугу";
button3.UseVisualStyleBackColor = true;
button3.Click += SavePos_Click;
DGVposlugi.AllowUserToAddRows = false;
DGVposlugi.AllowUserToDeleteRows = false;
DGVposlugi.AllowUserToResizeRows = false;
dataGridViewCellStyle4.Font = new System.Drawing.Font("Microsoft Sans
Serif", 8.25F);
DGVposlugi.AlternatingRowsDefaultCellStyle = dataGridViewCellStyle4;
DGVposlugi.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
dataGridViewCellStyle5.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft;
dataGridViewCellStyle5.BackColor =
System.Drawing.SystemColors.Window;
dataGridViewCellStyle5.Font = new System.Drawing.Font("Microsoft Sans
Serif", 8F);
dataGridViewCellStyle5.ForeColor =
System.Drawing.SystemColors.ControlText;
dataGridViewCellStyle5.SelectionBackColor =
System.Drawing.SystemColors.Highlight;
dataGridViewCellStyle5.SelectionForeColor =
System.Drawing.SystemColors.HighlightText;

```

```

DGVposlugi.DefaultCellStyle = dataGridViewCellStyle5;
DGVposlugi.Dock = System.Windows.Forms.DockStyle.Fill;
DGVposlugi.Location = new System.Drawing.Point(0, 41);
DGVposlugi.Margin = new System.Windows.Forms.Padding(10);
DGVposlugi.Name = "DGVposlugi";
DGVposlugi.ReadOnly = true;
DGVposlugi.RowHeadersWidth = 51;
DGVposlugi.Size = new System.Drawing.Size(976, 363);
panel6.Controls.Add(labelSUM);
panel6.Dock = System.Windows.Forms.DockStyle.Top;
panel6.Font = new System.Drawing.Font("Microsoft Sans Serif", 8F);
panel6.Location = new System.Drawing.Point(0, 0);
panel6.Name = "panel6";
panel6.Size = new System.Drawing.Size(976, 41);
labelSUM.Anchor = System.Windows.Forms.AnchorStyles.Top
System.Windows.Forms.AnchorStyles.Bottom
System.Windows.Forms.AnchorStyles.Left;
label17.AutoSize = true;
label17.Font = new System.Drawing.Font("Verdana", 10.2F,
System.Drawing.FontStyle.Bold);
System.Windows.Forms.AnchorStyles.Right;
textBox7.Font = new System.Drawing.Font("Verdana", 8F,
System.Drawing.FontStyle.Bold);
textBox7.Location = new System.Drawing.Point(803, 8);
textBox7.Name = "textBox7";
textBox7.Size = new System.Drawing.Size(159, 20);
textBox7.TabIndex = 4;
label15.Font = new System.Drawing.Font("Verdana", 8F,
System.Drawing.FontStyle.Bold);
label15.ForeColor = System.Drawing.Color.Green;

```

```

label15.Location = new System.Drawing.Point(102, 12);
label15.Name = "label15";
label15.Size = new System.Drawing.Size(70, 13);
label15.TabIndex = 3;
label16.AutoSize = true;
label16.Font = new System.Drawing.Font("Verdana", 10.2F,
System.Drawing.FontStyle.Bold);
label16.Location = new System.Drawing.Point(5, 9);
label16.Name = "label16";
label16.Size = new System.Drawing.Size(74, 17);
label16.TabIndex = 1;
tabPageStatistics.Controls.Add(panel1);
tabPageStatistics.Location = new System.Drawing.Point(4, 29);
tabPageStatistics.Name = "tabPageStatistics";
tabPageStatistics.Size = new System.Drawing.Size(976, 404);
tabPageStatistics.TabIndex = 3;
tabPageStatistics.Text = "Статистика клініки";
chart1.ChartAreas.Add(chartArea1);
chart1.Dock = System.Windows.Forms.DockStyle.Fill;
legend1.Name = "Legend1";
chart1.Legends.Add(legend1);
panel1.Controls.Add(labelStart);
panel1.Controls.Add(dateTimePickerEnd);
panel1.Controls.Add(dateTimePickerStart);
panel1.Controls.Add(button1);
panel1.Dock = System.Windows.Forms.DockStyle.Right;
panel1.Location = new System.Drawing.Point(805, 0);
panel1.Name = "panel1";
panel1.Size = new System.Drawing.Size(171, 404);
panel1.TabIndex = 4;

```

```

labelAllStatistic.AutoSize = true;
labelAllStatistic.Font = new System.Drawing.Font("Segoe UI Semibold",
12F, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, 204);
labelAllStatistic.ForeColor = System.Drawing.Color.FromArgb(0, 192, 0);
labelAllStatistic.Location = new System.Drawing.Point(42, 173);
labelAllStatistic.Name = "labelAllStatistic";
labelAllStatistic.Size = new System.Drawing.Size(79, 21);
labelAllStatistic.TabIndex = 6;
labelAllStatistic.Text = "Всього: 0";
labelEnd.AutoSize = true;
labelEnd.Location = new System.Drawing.Point(34, 67);
labelEnd.Name = "labelEnd";
labelEnd.Size = new System.Drawing.Size(133, 17);
labelEnd.TabIndex = 5;
labelEnd.Text = "Початок періоду";
labelStart.AutoSize = true;
labelStart.Location = new System.Drawing.Point(34, 9);
labelStart.Name = "labelStart";
dateTimePickerEnd.Value = new System.DateTime(2024, 4, 30, 0, 0, 0, 0);
// dateTimePickerStart
menuStrip1.ResumeLayout(false);
menuStrip1.PerformLayout();
tabControl1.ResumeLayout(false);
tabPage1.ResumeLayout(false);
((System.ComponentModel.ISupportInitialize)DGVVlasnyk).EndInit();
panel2.ResumeLayout(false);
panel2.PerformLayout();
tabPage2.ResumeLayout(false);
panel4.ResumeLayout(false);
groupBox2.ResumeLayout(false);

```

```

groupBox2.PerformLayout();
groupBox3.ResumeLayout(false);
groupBox3.PerformLayout();
((System.ComponentModel.ISupportInitialize)DGVanimals).EndInit();
panel3.ResumeLayout(false);
panel3.PerformLayout();
tabPage3.ResumeLayout(false);
panel5.ResumeLayout(false);
groupBox4.ResumeLayout(false);
groupBox4.PerformLayout();
tabControl2.ResumeLayout(false);
tabPage4.ResumeLayout(false);
tabPage5.ResumeLayout(false);
tabPage5.PerformLayout();
groupBox6.ResumeLayout(false);
groupBox6.PerformLayout();
((System.ComponentModel.ISupportInitialize)DGVposlugi).EndInit();
panel6.ResumeLayout(false);
panel6.PerformLayout();
tabPageStatistics.ResumeLayout(false);
((System.ComponentModel.ISupportInitialize)chart1).EndInit();
private System.Windows.Forms.ToolStripMenuItem vlasnykStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem PoslugyMenuItem;
private System.Windows.Forms.ToolStripMenuItem VydyAnimalsMenuItem;
private System.Windows.Forms.TabControl tabControl1;
private System.Windows.Forms.TabPage tabPage1;
private System.Windows.Forms.Panel panel2;
private System.Windows.Forms.Panel panelHeader;
private System.Windows.Forms.TabPage tabPage2;
private System.Windows.Forms.TabPage tabPage3;

```

```

private System.Windows.Forms.DataGridView DGVvlasnyk;
private System.Windows.Forms.TextBox textBox1;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox textBox2;
private System.Windows.Forms.DataGridView DGVanimals;
private System.Windows.Forms.GroupBox groupBox2;
private System.Windows.Forms.TextBox textBox5;
private System.Windows.Forms.GroupBox groupBox1;
private System.Windows.Forms.Label label14;
private System.Windows.Forms.TextBox textBoxDiagnoz;
private System.Windows.Forms.GroupBox groupBox6;
private System.Windows.Forms.Button button2;
private System.Windows.Forms.ColumnHeader columnHeader1;
private System.Windows.Forms.ColumnHeader columnHeader2;
private System.Windows.Forms.TabPage tabPage5;
private System.Windows.Forms.TextBox textBoxPosPrim;
private System.Windows.Forms.Label labelSUM;
private System.Windows.Forms.ToolStripMenuItem toolStripMenuItem1;
private System.Windows.Forms.TabPage tabPageStatistics;
private System.Windows.Forms.DataVisualization.Charting.Chart chart1;
private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.Label labelAllStatistic;
private System.Windows.Forms.Label labelEnd;
private System.Windows.Forms.Label labelStart;
private System.Windows.Forms.DateTimePicker dateTimePickerEnd;
private System.Windows.Forms.DateTimePicker dateTimePickerStart;
private System.Windows.Forms.Button button1;
}
}

```

					ДР.122.041.030.ПЗ	Арк.
						58
Змн.	Арк.	№ докум.	Підпис	Дата		