

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ПОЛІСЬКИЙ НАЦІОНАЛЬНИ УНІВЕРСИТЕТ

Факультет інформаційних технологій, обліку та фінансів
Кафедра комп'ютерних технологій
і моделювання систем

Кваліфікаційна робота
на правах рукопису

Крошка Анастасія Олександрівна
(прізвище, ім'я, по батькові здобувача освіти)

УДК 636.081.7:061.5

КВАЛІФІКАЦІЙНА РОБОТА

Розробка інформаційної системи ветеринарної клініки

(тема роботи)

126 «Інформаційні системи та технології»

(шифр і назва спеціальності)

Подається на здобуття освітнього ступеня бакалавр

кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання на
відповідне джерело

(підпис, ініціали та прізвище здобувача вищої освіти)

Науковий керівник – старший викладач
кафедри КТіМС Гіваргізов І.Г., к.е.н

Житомир – 2024

Висновок кафедри_____

за результатами попереднього захисту:_____

Протокол засідання кафедри_____

№ _____ від «_____» _____ 20____ р.

Завідувач кафедри

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

«_____» _____ 20____ р.

Результати захисту кваліфікаційної роботи

Здобувач вищої освіти _____ захистив (ла)

(прізвище ,ім'я, по батькові)

кваліфікаційну роботу з оцінкою:

сума балів за 100-бальною шкалою _____

за шкалою ECTS _____

за національною шкалою _____

Секретар ЕК

(науковий ступінь, вчене звання)

(підпис)

(прізвище, ім'я, по батькові)

АНОТАЦІЯ

Крошка А.О. Розробка інформаційної системи ветеринарної клініки. – Кваліфікаційна робота на правах рукопису.

Кваліфікаційна робота присвячена розробці інформаційної системи для ветеринарної клініки з метою поліпшення організації та надання якісних послуг. Основний акцент зроблено на створенні інтегрованої платформи, яка допомагатиме в управлінні медичними записами тварин, складанні графіків прийому, зберіганні інформації про лікування та медичних послуг. Розроблена система спрощує взаємодію між лікарями та власниками тварин, забезпечуючи зручний доступ до медичної інформації та сприяючи оперативній діагностиці та лікуванню.

Робота вирішує актуальну проблему складності доступу до медичної інформації тварин та організації робочих процесів у ветеринарних клініках, що часто призводить до затримок у наданні послуг та неефективного використання ресурсів.

У роботі описано процес аналізу вимог до системи, проектування та розробки інформаційної платформи. Особлива увага приділяється аспектам безпеки даних та забезпечення зручного та ефективного користування системою. Для ветеринарної клініки розроблено додаток на C# у Windows Form. Його інтуїтивний інтерфейс спрощує управління медичними записами тварин та їх власників, забезпечуючи ефективну роботу клініки.

Ключові слова: ветеринарна клініка, інформаційна система, медичні записи, графік прийому, діагностика, лікування, безпека даних.

SUMMARY

Kroshka A.O. Development of the information system of the veterinary clinic.
– Qualification work on manuscript rights.

The qualification work is devoted to the development of an information system for a veterinary clinic in order to improve the organization and provide quality services. The main focus is on creating an integrated platform that will help manage animal medical records, schedule admissions, store information about treatment and medical services. The developed system simplifies the interaction between doctors and animal owners, providing convenient access to medical information and facilitating prompt diagnosis and treatment.

The work solves the urgent problem of the difficulty of accessing medical information of animals and the organization of work processes in veterinary clinics, which often leads to delays in the provision of services and inefficient use of resources.

The work describes the process of analyzing system requirements, designing and developing an information platform. Particular attention is paid to aspects of data security and ensuring convenient and efficient use of the system. A C# Windows Form application was developed for a veterinary clinic. Its intuitive interface simplifies the management of medical records of animals and their owners, ensuring efficient operation of the clinic.

Keywords: veterinary clinic, information system, medical records, appointment schedule, diagnosis, treatment, data security.

ЗМІСТ

АНОТАЦІЯ	3
ВСТУП	7
Розділ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕТЕРИНАРНОЇ КЛІНІКИ	10
1.1 Аналіз інформаційних потреб і визначення предметної області дослідження	10
1.2 Функціональне моделювання інформаційної системи ветеринарної клініки	14
1.3 Структура інформаційної системи ветеринарної клініки	16
Висновки до першого розділу	18
Розділ 2. РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕТЕРИНАРНОЇ КЛІНІКИ	19
2.1. Формалізація моделі інформаційної системи ветеринарної клініки	19
2.2 Прогнозування навантаження на ветеринарну клініку за допомогою чисельного моделювання	20
Висновки до другого розділу	23
Розділ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБКИ ДЛЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ	24
3.1. Проектування інтерфейсу інформаційної системи ветеринарної клініки	24
3.2. Керівництво роботи з додатком користувачу	25
Висновки до третього розділу	25
ВИСНОВКИ ТА ПРОПОЗИЦІЇ	27
ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ	28

Перелік скорочень та умовних позначень

EMR	Electronic medical record (електронні медичні записи)
ТМЦ	Товарно-матеріальні цінності
CRM	Customer relationship management (управління відносинами з клієнтами)
IDEF0	Icam DEFinition for Function Modeling;
ER	Entity Relationship Diagram;
IDEF3	Integrated DEFinition for Process Description Capture Method;
DFD	Data Flow Diagrams

ВСТУП

У сучасному світі, де високотехнологічні рішення стають невід'ємною частиною кожної галузі, інформаційні системи виявляються доречним і ефективним інструментом для поліпшення різноманітних аспектів діяльності. Зокрема, ветеринарна медицина — це галузь, яка постійно розвивається та вдосконалюється, спрямовуючи свої зусилля на покращення якості надання послуг та лікування тварин.

Актуальність теми дослідження. Ветеринарні клініки щодня зіштовхуються із зростаючим потоком інформації та великою кількістю пацієнтів. Завдяки використанню інформаційних систем можна спростити та оптимізувати рутинні процеси, покращити облік медичної інформації та забезпечити ефективну координацію між персоналом та пацієнтами. Важливість забезпечення точної та оперативної обробки даних стає ключовою у забезпеченні якісної ветеринарної допомоги.

Мета кваліфікаційної роботи полягає у розробці інформаційної системи для ветеринарної клініки, яка дозволить автоматизувати основні процеси управління медичною інформацією, запису та планування прийому пацієнтів, а також ведення обліку лікарських препаратів та інших ресурсів клініки.

Реалізація цієї мети досягається вирішенням таких **завдань**:

- дослідити теоретичні і прикладні питання інформаційної системи для ветеринарної клініки;
- розкрити інформаційну структуру інформаційної системи ветеринарної клініки;
- побудувати моделі необхідні для опису поведінки та проєктування інформаційної системи;
- розробити інформаційну систему ветеринарної клініки;
- сформулювати рекомендації для користувачів інформаційної системи ветеринарної клініки.

Предметом дослідження є система управління медичною інформацією, планування та облік ресурсів, а також взаємодія між персоналом та пацієнтами.

Об'єктом дослідження є процес розробки інформаційної системи для ветеринарної клініки.

Для досягнення мети та вирішення задач дослідження використовувалися наступні **методи**: *аналізу* (аналіз наукової та науково-методичної літератури з області дослідження); *порівняння та синтезу* (для розуміння поточних процесів у ветеринарних клініках; дослідження аналогів існуючих ІС); *моделювання* (для побудови моделей структури інформаційних систем та їх складових); *проєктування* (для побудови інтерфейсу ІС); *тестування й аналітики* (для забезпечення ефективності ІС).

Практичне значення отриманих результатів. Отримані результати дозволять впровадити ефективну інформаційну систему ветеринарної клініки, що сприятиме покращенню організації робочих процесів, забезпечить точний облік медичної інформації та підвищить якість надання ветеринарних послуг.

Наукова новизна полягає у впровадженні інтегрованого підходу до розробки інформаційної системи ветеринарної клініки, що включає аналіз бізнес-процесів, проєктування інтерфейсу та розробку програмного забезпечення. Додатково, дослідження враховує потреби як персоналу клініки, так і її клієнтів, сприяючи створенню комплексного інструментарію для оптимізації робочих процесів та поліпшення якості надання ветеринарних послуг.

Унікальність кваліфікаційної роботи полягає в розробці спеціалізованого графіка, який автоматично обчислює та візуалізує кількість клієнтів у ветеринарній клініці протягом певного періоду часу. Цей інструмент сприяє більш точному плануванню робочого часу персоналу та оптимізації надання ветеринарних послуг.

Роботу апробовано на Міжфакультетській науково-практичній інтернет-конференції здобувачів вищої освіти і молодих вчених «Безпека,

технології, інновації: нові горизонти» (м. Житомир, 2023 р.) та Всеукраїнській науково-практичній конференції здобувачів вищої освіти і молодих вчених «Інформаційні технології та моделювання систем» (м. Житомир, 2024 р.), за результатами яких опубліковано тези у збірниках конференцій:

1. Крошка А., Інформаційна система для ветеринарної клініки // Інформаційні технології та моделювання систем : Безпека, технології, інновації: нові горизонти : збірник праць учасників міжфакультетської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих вчених, 15 листопада 2023 р. – Житомир : Поліський національний університет, 2023. – 69 с. – С. 17–18.

2. Крошка А., Інформаційна система для ветеринарної клініки // Інформаційні технології та моделювання систем : збірник праць учасників Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених, 10 квітня 2024 р. Житомир : Поліський національний університет, 2024. 76 с. – С. 51–52.

Структура та обсяг роботи. Кваліфікаційна робота складається з вступу, трьох розділів, висновку та списку використаних джерел. Загальний обсяг роботи становить 31 сторінку.

Розділ 1. ТЕОРЕТИЧНИЙ АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕТЕРИНАРНОЇ КЛІНІКИ

1.1 Аналіз інформаційних потреб і визначення предметної області дослідження

Ветеринарна клініка — це заклад, спеціалізований у наданні медичних послуг тваринам. Робота ветеринарних клінік зорієнтована на збереження здоров'я, діагностику та лікування різних видів домашніх тварин.

Аналіз інформаційних потреб ветеринарної клініки може допомогти визначити, які системи та функції будуть корисними для оптимізації роботи клініки. Ось деякі ключові аспекти, які можуть бути важливими при розробці інформаційної системи для ветеринарної клініки:

Основні функції ветеринарної клініки включають:

1. Медичні огляди та діагностика: здійснення регулярних медичних оглядів та діагностичних процедур для визначення стану здоров'я та виявлення можливих захворювань.
2. Лікування та хірургія: надання лікування для тварин, які стали жертвами травм, захворювань чи інших медичних проблем. Проведення хірургічних операцій, включаючи стерилізацію та кастрацію.
3. Вакцинація та профілактика: надання вакцинацій для запобігання важливим інфекційним хворобам. Консультування власників щодо профілактичних заходів та догляду за тваринами.
4. Догляд за зубами та шкірою: проведення стоматологічних процедур та лікування проблем зубів. Діагностика та лікування шкірних захворювань.
5. Екстрені втручання: надання допомоги в екстрених випадках, таких як травми, отруєння, алергічні реакції тощо.
6. Радіографія та лабораторні дослідження: використання рентгенівських променів для діагностики внутрішніх проблем. Проведення лабораторних тестів для аналізу крові, сечі та інших біологічних рідин.
7. Поради для власників: надання порад щодо правильного годування, утримання та загального догляду за тваринами.

8. Електронна документація: ведення медичних записів та історій лікування тварин для ефективного моніторингу їхнього здоров'я.

Компанії, які спеціалізуються на розробці програмного забезпечення для ветеринарних клінік: Idexx Laboratories, ezyVet, Vetter Software, DVMAX, VetBlue.

Компаній, які працюють на ринку програмного забезпечення є безліч. Я виділила найбільш популярні.

1. CRM Appointer

Компанія Appointer створила просте, зрозуміле і ефективне ПЗ для управління малим бізнесом. Програма для ветеринарної клініки спрощує планування, організацію і розрахунки, економить час і зводить до нуля кількість помилок. Ви отримуєте більше прибутку при менших зусиллях.

Можливості CRM Appointer для ветеринарної клініки:

- Повна інформація про господарів і вихованців: дані про клієнтів, облік заявок ветеринарної клініки, всі візити, послуги, історії захворювань, пошук потрібної інформації.
- Автоматизований запис на прийом. Містить зручний алгоритм запису кожного клієнта за рахунок використання електронного календаря. Економія часу співробітників. Мінімальний ризик помилитися і щось упустити;
- Функція онлайн-запису для власників тварин. Є можливість онлайн-запису для власників тварин в зручний для кожного клієнта час.
- Розрахунок заробітної плати. Ви один раз вносите в CRM для ветклініки дані розрахунку заробітних плат. Далі програма щомісяця рахує оклади автоматично.
- Статистика і аналітика ветклініки. Дані по прибутку клініки в різні місяці, історія приходу нових клієнтів/догляду старих, завантаженість ветеринарів і рентабельність процедур. Легко проаналізувати загальну картину підприємства.

- Облік витратних матеріалів на складі. Прихід/догляд витратних матеріалів та медикаментів. Програма для ветклініки розраховує витрату ТМЦ по кожній процедурі після того, як задані початкові значення.

Переваги використання CRM-системи Appointer у ветеринарній клініці:

- **ПЗ працює на будь-якій техніці, браузері.** Використовуйте CRM-систему на ноутбук, комп'ютері, телефоні, виходьте з будь-якого браузера.
- **П'єднується з телефонією IP.** ПЗ інтегрується з будь-яким провайдером IP-телефонії. Нового клієнта можна автоматично створити при дзвінку.
- **Швидке перенесення інформації з інших програм.** Дані без втрат швидко переміщуються в CRM Appointer.
- **Розсилка SMS-нагадувань.** Адміністратор створює текст повідомлення і графік відправки, ПЗ автоматично розсилає нагадування про візит.
- **Запис через соцмережі і сайт клініки.** Клієнти, які не люблять дзвінки, без проблем можуть записатися в клініку віддалено.

Таблиця 1.1-Тарифи CRM Appointer

Термін	Вартість
3 місяці	1 599 ₴ (533 грн./міс.)
6 місяці (+1 міс. безкоштовно)	3 199 ₴ (457 грн./міс.)
1 рік (+3 міс. безкоштовно)	6 399 ₴ (427 грн./міс.)

2. SKYSERVICE POS

Касова система обліку SkyService POS допоможе вести справи ефективно та оперативно. Дозволяє генерувати звіти, контролювати роботу клініки з будь-якої точки світу, працювати через додаток SkyService POS, який можна встановити на будь-який сучасний мобільний телефон чи планшет.

Таблиця 1.2-Тарифи SKYSERVICE POS

Назва тарифу	Вартість	Опис можливостей
BUSINESSMAN	306 грн./міс	Торгові точки 1 Кількість користувачів: ∞ Кількість транзакцій: ∞ Кількість товарних позицій: 600

		Технологічних карт: 60
CORPORATION	536 грн./міс	Торгові точки 1 Кількість користувачів: ∞ Кількість транзакцій: ∞ Кількість товарних позицій: 1000 Технологічних карт: 120
BIGBOSS	765 грн./міс	Торгові точки 1 Кількість користувачів: ∞ Кількість транзакцій: ∞ Кількість товарних позицій: 3500 Технологічних карт: 300

3. EasyWeek

EasyWeek – це програмне забезпечення, створене українцями в Дюссельдорфі. Сервіс підтримує 24 мови та доступний в Україні, Європі, Америці[6].

EasyWeek добре зарекомендували себе як якісне програмне забезпечення, яке пропонує багато корисних функцій за приємну вартість. При цьому ви отримаєте українську технічну підтримку, безкоштовний сайт та ще безліч корисних інструментів. [6]

Переваги CRM-сервісу EasyWeek:

- Програма веде електронний журнал відвідувань клієнтів, зберігає результати аналізів та призначень. Особливості поведінки тварини у різних ситуаціях, харчові звички та хронічні захворювання – все це можна вказати у картці пацієнта. EasyWeek – зручна електронна реєстратура ветклініки.
- У разі нездужань вихованця господареві не потрібно судомно намагатися додзвонитися до ветеринарної клініки, достатньо зайти на сайт або додаток, вибрати спеціаліста та час.
- CRM програма для ветклініки EasyWeek дозволяє згенерувати QR-код для запису онлайн. Код можна розмістити на стійці або двері клініки, також їм зручно ділитися в месенджері.
- EasyWeek дозволяє створити сайт та віджет. Це абсолютно безкоштовно та робиться автоматично, при реєстрації.
- EasyWeek надає більше 4K інтеграцій. Підтримуємо усі сервіси Google, Meta та Google карти.

Програмне забезпечення EasyWeek спрощує життя не лише клієнтам та їхнім вихованцям, але також оптимізує працю ветеринарних лікарів:

Скорочує час на заповнення картки пацієнта. Клієнтська база EasyWeek організована за принципом швидкого пошуку. Так, ви зможете легко знайти потрібні відомості про клієнта за декілька секунд.[]

Зручно зберігає відомості: уся інформація зберігається централізовано та в електронному форматі.

Виключає помилки, пов'язані з ручним обробленням інформації. Система зберігає ваші дані на безпечних серверах, виключаючи людський фактор.

Подає дані наочно: легко вивантажити статистику за конкретним клієнтом – як часто відвідує, які послуги замовляє, яке джерело запису приймання.

CRM-програма EasyWeek підходить як для невеликих ветеринарних кабінетів, так і для великих мережеских центрів, для нових та досвідчених ветклінік. Програма дуже зручна у налаштуванні та проста в управлінні. Порівняно з конкурентами, EasyWeek коштує недорого, а також пропонує систему лояльності для клієнтів будь-якого рівня. Приватним ветеринарним лікарям підійде безкоштовний тариф EasyWeek.

1.2 Функціональне моделювання інформаційної системи ветеринарної клініки

Визначимо мету: проаналізувати процесу діяльності ветеринарної клініки. Потрібно визначити точку зору, з якої будемо розробляти модель.

Перш за все створюємо контекстну діаграму IDEF0 процесу «Діяльність ветеринарної клініки» (див. рис. 1.1, Дод. А).

Для інформаційної системи діяльність ветеринарної клініки вхідними даними є: запис на прийом, медична карта домашнього улюбленця та медичні послуги. В діаграмі роль управління буде виконувати Закон України «Про захист прав споживачів», Закон України «Про ветеринарну медицину» та

внутрішніми правилами клініки. Механізмом процесу функціонування ветеринарної клініки буде менеджер та персонал клініки.

Результатом інформаційної системи є фіксація медичної документації(звітність), інформація про вартість та оплату, перегляд послуг та можливість онлайн-запису.

Проведемо декомпозицію контекстної діаграми на складові процеси (див. рис.1.2, Дод. А). Процес «Діяльність ветеринарної клініки» поділимо на такі підпроцеси: здійснення запису; надання послуг ветеринаром; оплата послуг; надання висновків та рекомендацій.

Розглянемо детально декомпозицію контекстної діаграми, вона складається з 4 процесів. Спочатку відбувається процес «Запису у ветеринарну клініку», під час запису використовуються дані власника тварини. Другий блок – це «Надання послуг», де використовуються медична карта та медичні послуги, що наявні в клініці і відповідають потребі тварини, що захворіла. Після чого клієнту повідомляється про потребу здійснення оплати послуг. Процес «Оплата послуг» відбувається за допомогою менеджера, що працює у клініці. Після оплати послуг та фіксації в медичній карті, можна перейти до наступного блоку «Надання висновків та рекомендацій».

Виконаємо декомпозицію процесу «Надання послуг» на основі побудованої діаграми з використанням стандарту IDEF0 (див. рис. 1.3, Дод. А).

Діаграма декомпозиції у складі має такі процеси: прийом пацієнта; медичне обстеження; визначення діагнозу та лікування; стаціонарне лікування; амбулаторне лікування; функціональна діагностика; хірургія; кастрація, стерилізація; лабораторні дослідження; реєстрація, вакцинація; інші послуги.

Процес надання послуг включає в себе: прийом пацієнта, після чого відбувається підготовка до обстеження, медичне обстеження, визначення діагнозу та сам процес лікування, що може включати в себе стаціонарне

лікування, амбулаторне, функціональну діагностику, хірургію та інші послуги в залежності від визначеного діагнозу.

1.3 Структура інформаційної системи ветеринарної клініки

Діаграма потоку даних (DFD) – це графічний інструмент для моделювання та аналізу систем, який ілюструє потоки даних між процесами, даними, зовнішніми сутностями та джерелами даних[5]. У контексті створення інформаційної системи для ветеринарної клініки DFD може бути корисною з кількох причин:

- DFD дозволяє представити ключові компоненти та їх взаємозв'язки в інформаційній системі для ветеринарної клініки. Це полегшує розуміння загальної архітектури системи.
- діаграма надає можливість визначити, як дані переміщуються через різні частини системи. Це допомагає виявляти можливі точки оптимізації та покращення обробки інформації.
- дозволяє ідентифікувати різні процеси та функції, що відбуваються в системі ветеринарної клініки, такі як запис пацієнтів, назначення лікування тощо.

Застосування DFD допомагає зрозуміти структуру та процеси інформаційної системи для ветеринарної клініки, сприяючи покращенню її функціональності, ефективності та відповідності вимогам користувачів. Виконаємо декомпозиції процесу «Звернення до клініки» наступним чином (див. рис. 1. 4, Дод. А):

Опрацювання звернення виконується за наступною процедурою:

- власник тварини звертається до менеджера клініки;
- звернення обробляється та виконується пошук медичної картки тварини в базі даних тварин.
- аналізується інформація про причину звернення, в залежності від якої обирається ветеринар з бази даних;
- формується заявка, у якій вказано дату, час та кабінет у якому буде проходити прийом.

Виконаємо далі опис процесу «Хірургія» наступним чином (див. рис. 1.5, Дод. А):

Опрацювання звернення виконується за наступною процедурою:

- отримання дозволу власника на хірургічне втручання;
- проведення підготовки тварини до операції;
- проведення операції;
- нагляд після операції.

Виконаємо ще один опис процесу «Оплата послуг» враховуючи процеси, які відбуваються при оплаті наданих послуг. Процес обрахунку розпочинається з визначення ціни наданої послуги. Ціна зберігається в базі даних клініки, тому виконується запит. Після потрібно визначити вартість медикаментів, які були застосовані під час надання послуги. Ціна також зберігається в тій же базі даних. Наступним етапом є визначення загальної вартості наданих послуг. Далі йде процес оплати на касі ветеринарної клініки. Детальний опис складових процесу «Оплата послуг» (див. рис. 1.6, Дод. А).

Для ще більш детального аналізу інформаційної системи було створено IDEF3. IDEF3 (Integrated Definition for Function Modeling) – це мова моделювання, яка використовується для представлення функціональної інформації в системах та процесах. Вона дозволяє визначати та аналізувати функції систем, ідентифікувати взаємозв'язки між ними та відображати їх в структурованій формі[7] (див. рис. 1.7, Дод. А).

Перша дія – здійснення запису власником тварини, власник телефонує у ветеринарну клініку та бронює час та дату візиту, якщо запис підтверджений, то відбувається підтвердження дати та часу доби за день до візиту, якщо ні – запис скасовується. Далі відбувається проведення медичного обстеження, що в свою чергу розбивається на три етапи будовою «І», взяття аналізів, проведення УЗД, виконання рентгену. Після чого власник тварини отримує результати аналізів і вони розгортаються з'єднанням «Ексклюзивне АБО», від якого йде дві дії: провести додаткове обстеження та встановити діагноз. Після

встановлення діагнозу призначається лікування та надаються консультації щодо домашнього догляду.

Висновки до першого розділу

У першому розділі було проаналізовано програмне забезпечення для ветеринарних клінік, включаючи їх інформаційні потреби та організацію роботи. Описано основні поняття та можливості програмного забезпечення для ветеринарних клінік та перелік наданих послуг. Детально розглянуто механізм надання цих послуг.

На основі здійсненого аналізу розроблено моделі діяльності різних процесів ветеринарної клініки, а саме: створено модель процесу діяльності клініки у методології IDEF0 з декомпозицією та глосарієм термінів; побудовано діаграми взаємодії процесів за допомогою DFD та IDEF3.

Це створює базу для подальших розробок та оптимізації системи з метою покращення якості обслуговування клієнтів та ефективності ветеринарної клініки.

Розділ 2. РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ВЕТЕРИНАРНОЇ КЛІНІКИ

2.1. Формалізація моделі інформаційної системи ветеринарної клініки

Створення інформаційної системи для ветеринарної клініки є ключовим елементом оптимізації та підвищення ефективності ветеринарного обслуговування тварин. Аналогічно до інформаційних систем у сфері ветеринарії, ветеринарні клініки можуть використовувати передові технології для забезпечення точної інформації про стан кожного пацієнта. Це дозволяє клініці ефективно планувати та надавати високоякісні ветеринарні послуги, максимально оптимізуючи використання ресурсів та часу лікарів. З урахуванням зростання обсягу клієнтів у ветеринарних клініках та різноманіття видів тварин, використання інформаційних систем стає вирішальним фактором для поліпшення управління та надання високоякісної медичної допомоги у сфері ветеринарії.

(1)

$$\frac{dl}{dt} = \lambda - \mu l$$

де l – середня кількість клієнтів у системі, λ – інтенсивність потоку клієнтів, μ – інтенсивність обслуговування.

Це диференціальне рівняння враховує приріст клієнтів через потік λ та відтік через обслуговування μ . Модель визначає, як швидко змінюється середня кількість клієнтів у системі відносно часу. Тепер розробим структурну схему цієї формули, вона показана на рисунку (рис. 2.1)

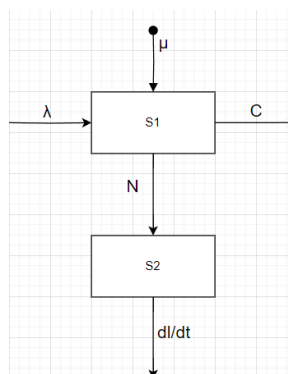


Рисунок 2.1 – Схема числа обслуговуваних клієнтів в клініці

Рівняння $\frac{dl}{dt} = \lambda - \mu l$ описує зміну середньої кількості клієнтів l у ветеринарній клініці відносно часу t . У цьому рівнянні λ представляє інтенсивність потоку нових клієнтів, які приходять в клініку, тоді як μ відображає інтенсивність процесу обслуговування, яке зменшує кількість клієнтів у системі. Змінна l представляє середню кількість клієнтів у системі, а $\frac{dl}{dt}$ визначає темп зміни цієї кількості. Рівняння може бути використане для моделювання і оптимізації процесів обслуговування та очікування клієнтів у ветеринарній клініці.

2.2 Прогнозування навантаження на ветеринарну клініку за допомогою чисельного моделювання

Реалізація моделей була розроблена в редакторі коду Visual Studio Code, на інтерпретованій об'єктно-орієнтованій мові програмування Python.

```
# Функція, яка повертає значення dl/dt
def model(l, t):
    return lambda_val - mu_val * l
```

Рисунок 2.2 – Функція

Функція $model(l, t)$ описує темп зміни середньої кількості клієнтів у ветеринарній клініці за допомогою диференційного рівняння. У цьому рівнянні λ і μ відображають інтенсивність потоку нових клієнтів та процесу обслуговування відповідно. Функція повертає значення, що представляє темп зміни кількості клієнтів відносно часу.

```
# Визначення параметрів
lambda_val = 2.0 # інтенсивність потоку клієнтів
mu_val = 0.5     # інтенсивність обслуговування
```

Рисунок 2.3 – Визначення параметрів

У цьому розділі коду визначаються параметри, які визначають характеристики системи ветеринарної клініки для моделювання. «*lambda_val*» представляє інтенсивність потоку нових клієнтів, що приходять до клініки, а «*mu_val*» вказує на інтенсивність обслуговування, яка визначає те, як швидко клієнти обслуговуються та залишають систему. Ці параметри визначаються

конкретними числовими значеннями (2.0 та 0.5 відповідно), і вони використовуються у функції «*model*» для створення математичної моделі зміни кількості клієнтів у системі відносно часу.

```
# Початкова умова: початкова кількість клієнтів
l0 = 0
```

Рисунок 2.4 – Визначення параметрів

l_0 визначає початкову кількість клієнтів у системі при початку моделювання. У цьому випадку l_0 дорівнює 0, вказуючи, що спочатку немає клієнтів у системі.

```
# Часовий інтервал для розв'язання рівняння (від 0 до 8 годин)
working_hours = 8
t = np.linspace(0, working_hours, 100)
```

Рисунок 2.5 – Часовий інтервал

Зазначається часовий інтервал t , в межах якого проводиться розв'язання диференційного рівняння. У випадку даного коду цей інтервал встановлено від 0 до 8 одиниць часу, що відповідає восьмигодинному робочому дню. Такий часовий проміжок вибирається для відтворення динаміки кількості клієнтів протягом стандартного робочого дня у ветеринарній клініці.

```
# Розв'язок диференційного рівняння
l_solution = odeint(model, 8, t)

# Виведення результатів
for time, clients in zip(t, l_solution):
    print(f"Час: {time:.2f}, Кількість клієнтів: {round(clients[0])}")
```

Рисунок 2.6 – Розв'язок та виведення результатів

На цьому етапі програми використовується чисельний метод розв'язання диференційного рівняння з використанням функції «*odeint*». Результати розв'язку зберігаються в змінній «*l_solution*». Далі, за допомогою циклу, результати виводяться на екран, де для кожного моменту часу вказується час та кількість клієнтів, яка округляється до найближчого цілого числа. Цей процес дозволяє аналізувати, як змінюється кількість клієнтів у ветеринарній клініці протягом восьмигодинного робочого дня.

Powered by  trinket

Час: 0.00 год.,	Кількість клієнтів: 0
Час: 0.08 год.,	Кількість клієнтів: 0
Час: 0.16 год.,	Кількість клієнтів: 0
Час: 0.24 год.,	Кількість клієнтів: 0
Час: 0.32 год.,	Кількість клієнтів: 1
Час: 0.40 год.,	Кількість клієнтів: 1
Час: 0.48 год.,	Кількість клієнтів: 1
Час: 0.57 год.,	Кількість клієнтів: 1
Час: 0.65 год.,	Кількість клієнтів: 1
Час: 0.73 год.,	Кількість клієнтів: 1
Час: 0.81 год.,	Кількість клієнтів: 1
Час: 0.89 год.,	Кількість клієнтів: 1
Час: 0.97 год.,	Кількість клієнтів: 2
Час: 1.05 год.,	Кількість клієнтів: 2
Час: 1.13 год.,	Кількість клієнтів: 2
Час: 1.21 год.,	Кількість клієнтів: 2
Час: 1.29 год.,	Кількість клієнтів: 2
Час: 1.37 год.,	Кількість клієнтів: 2
Час: 1.45 год.,	Кількість клієнтів: 2
Час: 1.54 год.,	Кількість клієнтів: 2
Час: 1.62 год.,	Кількість клієнтів: 2
Час: 1.70 год.,	Кількість клієнтів: 2
Час: 1.78 год.,	Кількість клієнтів: 2
Час: 1.86 год.,	Кількість клієнтів: 2
Час: 1.94 год.,	Кількість клієнтів: 2
Час: 2.02 год.,	Кількість клієнтів: 3
Час: 2.10 год.,	Кількість клієнтів: 3
Час: 2.18 год.,	Кількість клієнтів: 3
Час: 2.26 год.,	Кількість клієнтів: 3
Час: 2.34 год.,	Кількість клієнтів: 3
Час: 2.42 год.,	Кількість клієнтів: 3

Рисунок 2.7 – Результат після компіляції

Ця програма реалізує математичну модель зміни кількості клієнтів у ветеринарній клініці протягом 8-годинного робочого дня. Використовуючи диференціальне рівняння, яке враховує потік нових клієнтів та інтенсивність їх обслуговування, програма розв'язує цю модель та виводить кількість клієнтів для кожного моменту часу протягом заданого періоду. Таким чином, вона дозволяє аналізувати, як змінюється навантаження клініки та скільки часу потрібно для обслуговування клієнтів ветеринарної допомоги.

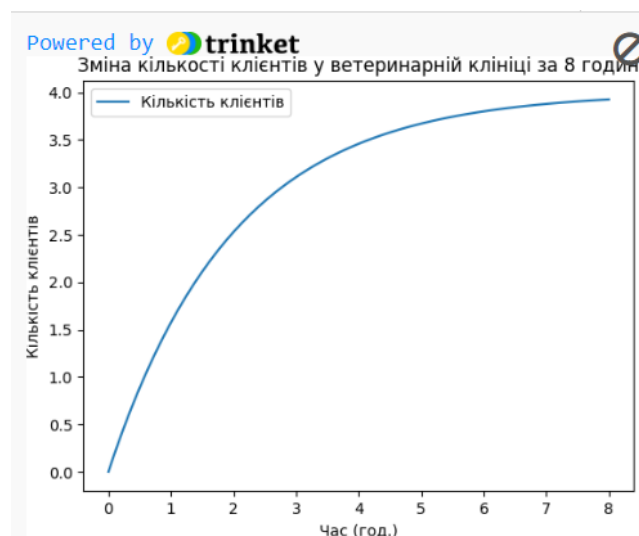


Рисунок 2.8 – Графік зміни кількості клієнтів у клініці

Графік динаміки кількості клієнтів у ветеринарній клініці відображає криву, аналогічну формі S-кривої. Починаючи з початкової кількості клієнтів, процес збільшення спочатку протікає помірно, з послідовним прискоренням. У середині робочого дня може бути досягнуто піку, внаслідок чого темп зростання зменшується. Подальше збільшення клієнтів може сповільнитися або стабілізуватися, а в кінці робочого періоду можливий невеликий спад. Цей динаміка вказує на те, що ресурси клініки могли б бути оптимально використані під час піку, і дозволяє аналізувати ефективність обслуговування клієнтів у ветеринарній клініці.

Висновки до другого розділу

У другому розділі представлено математичну модель зміни кількості клієнтів у ветеринарній клініці протягом робочого дня та її програмну реалізацію для прогнозування навантаження на ветеринарну клініку. Модель ґрунтується на диференціальному рівнянні, яке описує зміну середньої кількості клієнтів у системі відносно часу.

Моделльні параметри, такі як інтенсивність потоку нових клієнтів та інтенсивність обслуговування, можуть бути налаштовані для відображення реальних умов роботи ветеринарної клініки. Чисельне моделювання продемонструвало, що динаміка кількості клієнтів протягом дня має S-подібну форму з піком у середині робочого дня.

Ця модель може бути використана для оптимізації роботи ветеринарної клініки, наприклад, для планування графіків роботи персоналу та розподілу ресурсів тощо.

Розділ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ РОЗРОБКИ ДЛЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

3.1. Проектування інтерфейсу інформаційної системи ветеринарної клініки

Ефективність інформаційної розробки значно залежить від якості її інтерфейсу. Інтерфейс повинен наочно представляти всі можливості програми, не викликаючи в користувача відчуття відстані або невпевненості, і бути легким у використанні і зрозумілим.

Розглянемо зовнішній вигляд додатку. Першим серед них буде головна форма для роботи з власниками тварин. На формі можна лише переглядати власників та виконувати пошук за прізвищем або за адресом проживання.(див. рис.3.1., Дод. Б).

Для роботи з тваринами потрібно в головній формі перейти до вкладки «Тварини» (див. рис.3.2., Дод. Б). Дана вкладка відображає лише тварин, які має вибраний власник на вкладці «Власник». 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» (див. рис.3.3., Дод. Б). У довіднику можна додати новий вид, видалити не потрібний вид або відредагувати наявного.

Якщо є потреба у новій послугі або змінити наявну, потрібно перейти до меню «Довідники» та вибрати «Послуги клініки» (див. рис.3.4., Дод. Б). Даний довідник дозволяє виконувати операції по додаванні нової послуги, редагуванні поточної або видаляти послугу.

З меню можна відкрити форму для перегляду інформації про автора програми (див. рис.3.5., Дод. Б). Опис заповненої лінійної діаграми що обраховує кількість обслуговуваних клієнтів за певний період часу, може виглядати наступним чином: Лінійна діаграма відображає часову динаміку кількості обслуговуваних клієнтів ветеринарної клініки за певний період часу (наприклад, день, тиждень, місяць). Ось основні елементи діаграми: вісь X (ось

абсцис), вісь Y (ось ординат), лінія діаграми, легенда та титул. Ця діаграма дозволяє персоналу клініки швидко оцінити тенденції та пікові навантаження, що допомагає в оптимальному плануванні робочого графіка та ресурсів клініки. (див. рис.3.6., Дод. Б).

3.2. Керівництво роботи з додатком користувачу

Після входу в додаток, користувачу, йому буде запропоновано Windows Form, який включає в себе такі вкладки: «Власник», «Тварини», «Надані послуги», довідник «Види тварин» і довідник «Послуги», користувачі можуть легко взаємодіяти з різними аспектами системи.

Вкладка «Власник» забезпечує можливість перегляду інформації про власників тварин (пацієнтів клініки) та здійснення пошуку за прізвищем чи адресою проживання. Компактний та зручний інтерфейс дозволяє отримувати швидкий доступ до потрібних даних.

Вкладки «Тварини», довідник «Види тварин» та довідник «Послуги» дозволяють керувати інформацією про тварин, види тварин та послуги відповідно. Додавання нових записів, видалення та редагування існуючих здійснюється легко та ефективно.

У вкладці «Про програму» можна отримати детальну інформацію про розробника ПЗ.

Загалом, інструкція дозволяє користувачам ознайомитися повною мірою із функціоналом та усіма необхідними інструментами для зручного та продуктивного використання інформаційної системи ветеринарної клініки.

Висновки до третього розділу

Третій розділ присвячено опису спроектованого інтерфейсу розробленої інформаційної системи ветеринарної клініки та огляду функціоналу системи. Інтерфейс готового продукту є зручним, зрозумілим та наочним, щоб користувачі могли легко взаємодіяти з системою та отримувати доступ до необхідної інформації.

Важливим аспектом є створена інструкція користувача, яка надає чіткі та конкретні вказівки щодо використання основних функцій системи. Ця

інструкція має на меті спростити робочий процес користувачів та полегшити доступ до необхідної інформації.

Загалом, інтерфейс та інструкція користувача інформаційної системи ветеринарної клініки розроблені таким чином, щоб забезпечити зручне та ефективне використання системи для всіх користувачів.

ВИСНОВКИ ТА ПРОПОЗИЦІЇ

Мета кваліфікаційної роботи полягала в дослідженні, проектуванні та створенні програмного забезпечення, яке відповідає потребам ветеринарної клініки. Було досліджено функціональність ветеринарної клініки, вдосконалення робочих місць та процесів, покращення якості обслуговування тварин та їх власників. Крім того, при написанні кваліфікаційної роботи збільшила розуміння сучасних технологій в області інформаційних систем, а також вдосконалила навички аналізу, проектування та моделювання.

Було проведено розробку моделей процесів діяльності ветеринарної клініки, з метою систематизації та оптимізації її роботи.

Дослідила програмне забезпечення інших компаній, зокрема: Idexx Laboratories, ezyVet, Vetter Software, DVMAX, VetBlue.

Спроектowano модель процесу діяльності ветеринарної клініки у методології IDEF0. Зроблено декомпозицію. При написанні кваліфікаційної роботи було використано DFD для організації документообігу та обробки інформації. Було зроблено декілька декомпозицій. Продemonстровано використання нотації IDEF3 для опису структурних аспектів процесів.

Було сформульовано вимоги для програмного та апаратного забезпечення клініки. На які спиралась при розробці своєї інформаційної системи. Детально описала інтерфейс та зробила опис роботи з нею.

ПЕРЕЛІК ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. CRM Appointer [Електронний ресурс] – Режим доступу до ресурсу:
<https://appointer.ua/>.
2. EasyWeek [Електронний ресурс] – Режим доступу до ресурсу:
<https://easyweek.com.ua/programa-dlya-vetklinik.html>.
3. SkyService POS [Електронний ресурс] – Режим доступу до ресурсу:
<https://skyservice.pro/>.
4. 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>.
5. 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.
6. Кастрація та стерилізація тварин [Електронний ресурс] – Режим доступу до ресурсу: http://drug.zp.ua/p_kastraciya-sterilizaciya.html.
7. Цілодобовий стаціонар [Електронний ресурс] – Режим доступу до ресурсу: http://drug.zp.ua/p_stacionar.html.
8. Лабораторні дослідження. Лабораторні дослідження для тварин [Електронний ресурс] – Режим доступу до ресурсу:
http://drug.zp.ua/p_laboratornye-issledovaniya.html.
9. Універсальна Система Обліку [Електронний ресурс] – Режим доступу до ресурсу:
10. https://shop.as-service.com.ua/specialized_software/veterinary.
11. Юрчук, Н. П. Інформаційні системи в управлінні діяльністю підприємства [Текст] / Н. П. Юрчук // Агросвіт. - 2015. - № 19. - С. 53–58.
12. Криворучко, О. В., Дитинюк, О. В. Інформаційні технології моделювання виробничих процесів як інструмент прийняття управлінських рішень [Текст] / О. В. Криворучко, О. В. Дитинюк // Інформаційні технології управління. - 2017. - № 31. - С. 65–70.

13. Kornienko, G., Chabanenko, M., Leheza, Y. Assessment of the economic efficiency of it application at enterprises [Текст] / G. Kornienko, M. Chabanenko, Y. Leheza // *Baltic Journal of Economic Studies*. - 2018. - Vol. 4, No. 3. - P. 123–132. - DOI: <https://doi.org/10.30525/2256-0742/2018-4-3-123-132>.

14. Дорохов, О. В. Критерії та методи оцінки ефективності інформаційних систем [Текст] / О. В. Дорохов // *Системи обробки інформації*. - 2010. - № 1. - С. 219–222.

15. Писарчук, О. О. Оцінювання ефективності інформаційних систем за вектором критеріїв [Текст] / О. О. Писарчук // *Журнал Вінницького національного аграрного університету*. - 2018. - Вип. 3. - С. 117-123.

16. Верескун, М. В. Методи оцінки ефективності впровадження інформаційних систем на промислових підприємствах [Текст] / М. В. Верескун // *Теоретичні і практичні аспекти економіки та інтелектуальної власності*. - ПДТУ. - Маріуполь. - 2015. - Вип. 1, Т. 1. - С. 21–26.

17. Сафонов, М. С., Яковенко, О. Є. Прогнозування стану показників об'єктно-орієнтованої моделі в інформаційній системі [Текст] / М. С. Сафонов, О. Є. Яковенко // *Інформаційні технології в освіті, науці та виробництві*. - Одеса. - 2013. - № 3(4). - С. 92 – 98.

18. Beskorovajnyj, V. V. Systemological analysis of the problem of structural synthesis of geographically distributed systems [Текст] / V. V. Beskorovajnyj // *Automated control systems and automation devices*. - 2002. - No. 120. - P. 29–37.

19. Буй, Ф. Л., Беляєв, Л. В. Методи і технології реінжинірингу інформаційних систем [Текст] / Ф. Л. Буй, Л. В. Беляєв // *Інформаційні технології в економіці і управлінні: зб. наук. студ. праць*. - Одеса: ОНЕУ. - 2019. - Вип. 1. - С. 145–152.

20. Іванова, Т. В., Баранов, В. В. Сучасний стан розвитку інформаційних систем [Електронний ресурс]. - URL: www.kntu.kr.ua/doc/nauk_zap_10_1/stat_10_1/64.doc.

21. ДСТУ ISO/IEC 2382:2017 (ISO/IEC 2382:2015, IDT). Інформаційні технології. Словник термінів [Текст]. - К.: УкрНДНЦ. - 2017.

22. Кисіль, Н. М., Гаталяк, З. П., Горбаль, Н. І. Класифікація інформаційних систем [Електронний ресурс]. - URL: base.dnsgb.com.ua/files/journal/Lisove-gospodarstvo-l-p.../242_Kysil_LG_29.pdf.
23. Литовченко, Д. М. Реінжиніринг інформаційних систем економічних об'єктів. Моделювання регіональної економіки. 2010. № 1. С. 176–184.
24. Черняк, Н.І. Моделювання структури інформаційної системи управління агропромислового комплексу регіону [Текст] / Н.І. Черняк // Оптико-електронні інформаційно-енергетичні технології. - 2008. - № 2. - С. 83-88.
25. Яремко, С.А., Бевз, С.В. Розробка критеріїв оцінювання сучасних інформаційних систем обліку та управління бізнес-процесами підприємств [Текст] / С.А. Яремко, С.В. Бевз // Вісник Хмельницького національного університету. - 2015. - № 1 35 університету. - С. 158–163.
26. Філінюк, М. А., Багацький, В. О., Ліщинська, Л. Б., Войцеховська, О. В. Критеріальне оцінювання ефективності інформаційних пристроїв та систем: навчальний посібник [Текст] / М. А. Філінюк, В. О. Багацький, Л. Б. Ліщинська, О. В. Войцеховська. - Вінниця: ВНТУ, 2014. - 143 с.
27. Zhang Yi Fei, Che Ruhana Isa. Factors Influencing Activity-Based Costing Success: A Research Framework [Текст] / Yi Fei Zhang, Ruhana Isa Che // International Journal of Trade, Economics and Finance. - 2010. - V.1, no.2. - P. 144-150.
28. Evaluation of effectiveness of information systems implementation in organization (by example of ERP-systems) [Текст] / O V Demyanova and other // Journal of Physics Conference Series. - 2018.05. - Vol. 1015. - P. 1-5.
29. Muhamed Saif Q., Mohammed Mohammed Q., Nayl T., Chyrkova K. The Concept of Building a Model of the National Blood Information System [Текст] / Muhamed Saif Q., Mohammed Mohammed Q., Nayl T., K. Chyrkova // Iraqi Journal for Computers and Informatics (IJCI). - 2017. - 43 (1). - P. 17 – 21.
30. Чиркова, К.С. Особливості функціонування інформаційних систем служби крові [Текст] / К.С. Чиркова // Матеріали 19-ого міжнародного

молодіжного форуму «Радіоелектроніка і молодь в XXI в.». - Харків, 2015. - С. 160–161.

31. Beskorovajnyj, V. V. Systemological analysis of the problem of structural synthesis of geographically distributed systems [Текст] / V. V. Beskorovajnyj // Automated control systems and automation devices. - 2002. - No. 120. - P. 29–37.

32. Sander T. Modeling Object-Oriented Software for Reverse Engineering and Refactoring [Текст] / T. Sander // Thesis. - 2001. - University of Bern, Bern, Switzerland. - P. 142 – 149.

33. Ducasse S. Rretro-Conception d'Application `a Objets Reengineering ObjectOriented Applications [Текст] / S. Ducasse // Universitre Pierre et Marie Curie. - Paris. - 2001. - P. 81–93.

34. Крошка А., Інформаційна система для ветеринарної клініки // Інформаційні технології та моделювання систем : Безпека, технології, інновації: нові горизонти : збірник праць учасників міжфакультетської науково-практичної інтернет-конференції здобувачів вищої освіти і молодих вчених, 15 листопада 2023 р. – Житомир : Поліський національний університет, 2023. – 69 с. – С. 17–18.

35. Крошка А., Інформаційна система для ветеринарної клініки // Інформаційні технології та моделювання систем : збірник праць учасників Всеукраїнської науково-практичної конференції здобувачів вищої освіти і молодих вчених, 10 квітня 2024 р. Житомир : Поліський національний університет, 2024. 76 с. – С. 51–52.

Додаток А

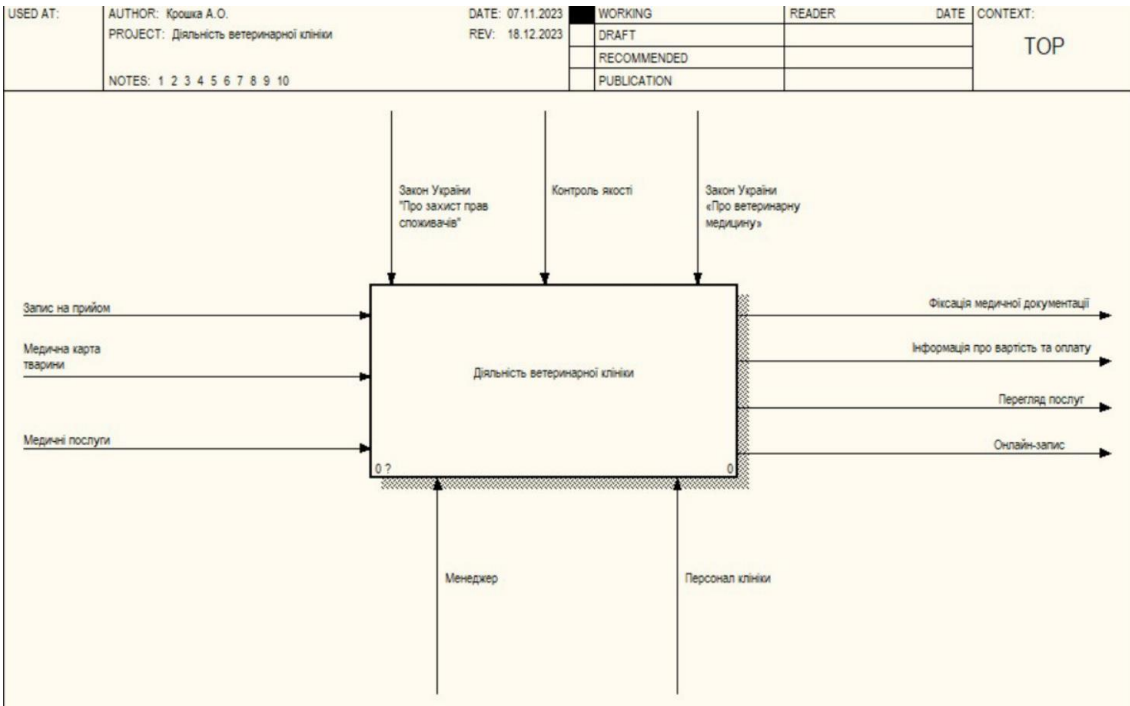


Рисунок 1.1 – IDEF0 «Діяльність ветеринарної клініки»

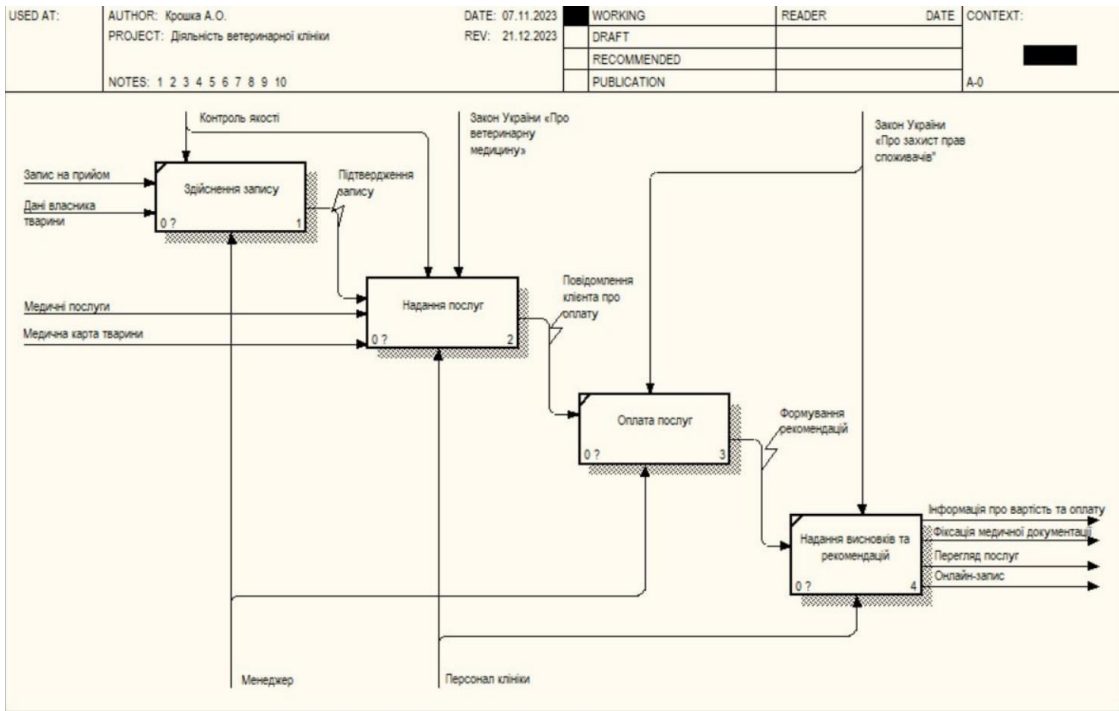


Рисунок 1.1 – Декомпозиція контекстної діаграми процесу «Діяльність ветеринарної клініки»

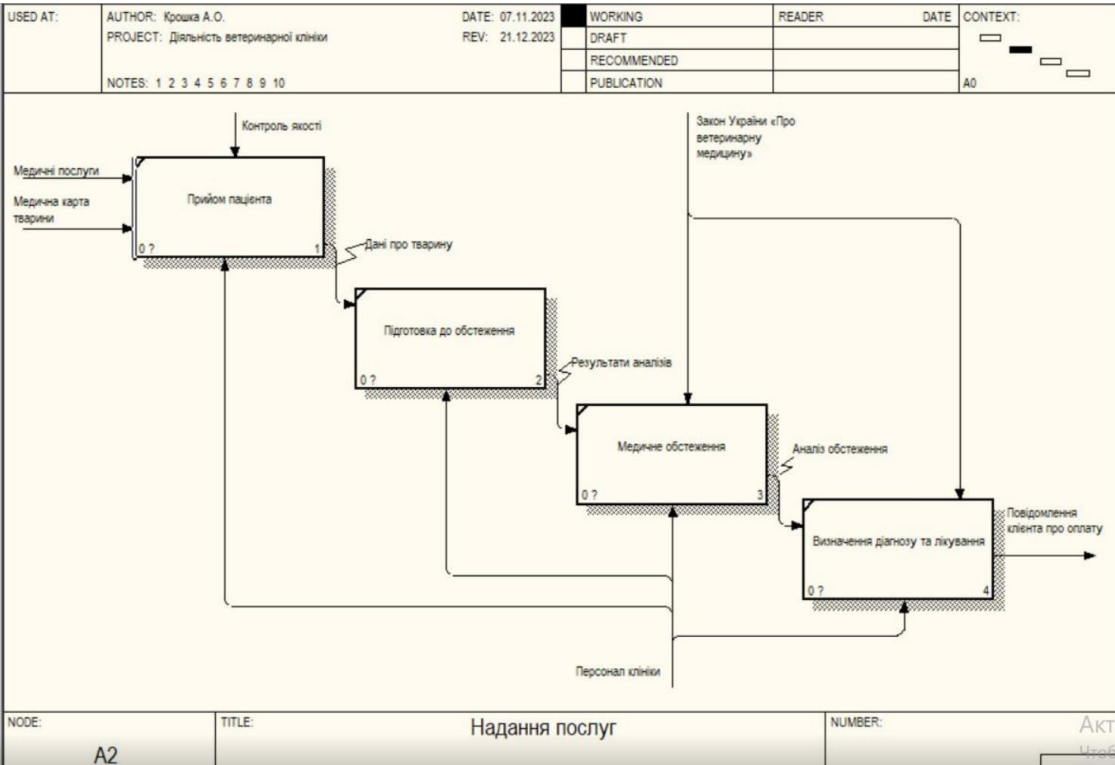


Рисунок 1.2 – Декомпозиція процесу «Надання послуг»

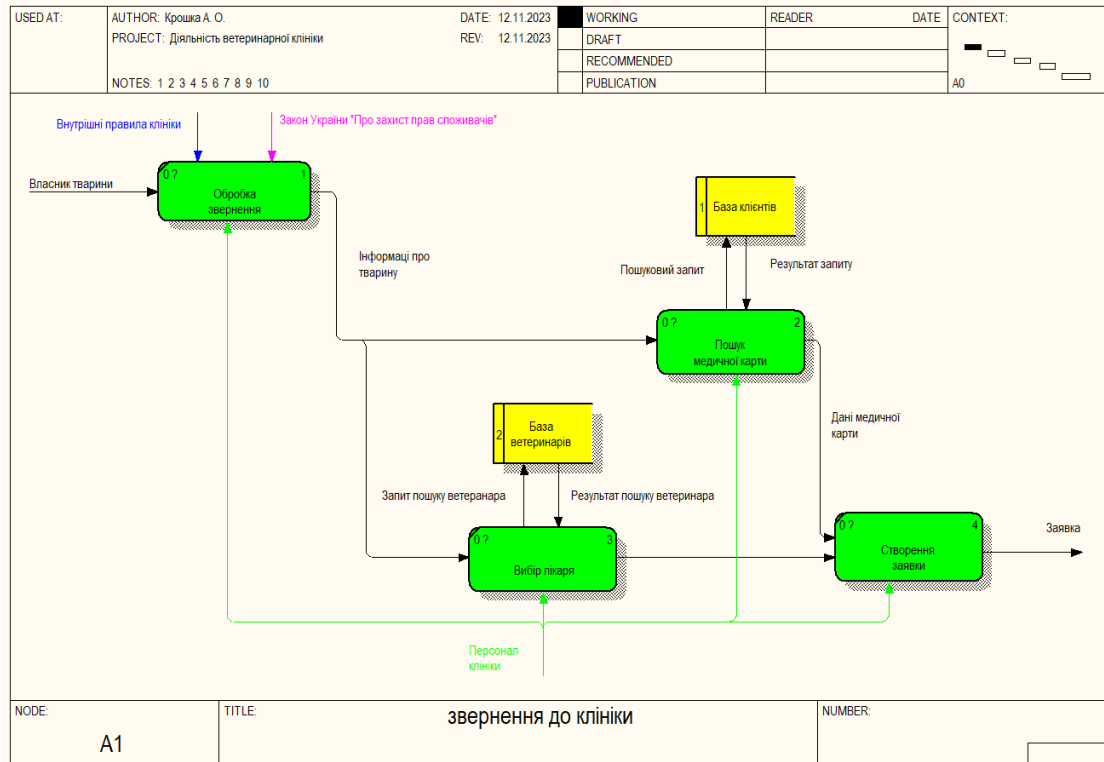


Рисунок 1.4 – Опис процесу «Звернення до клініки»

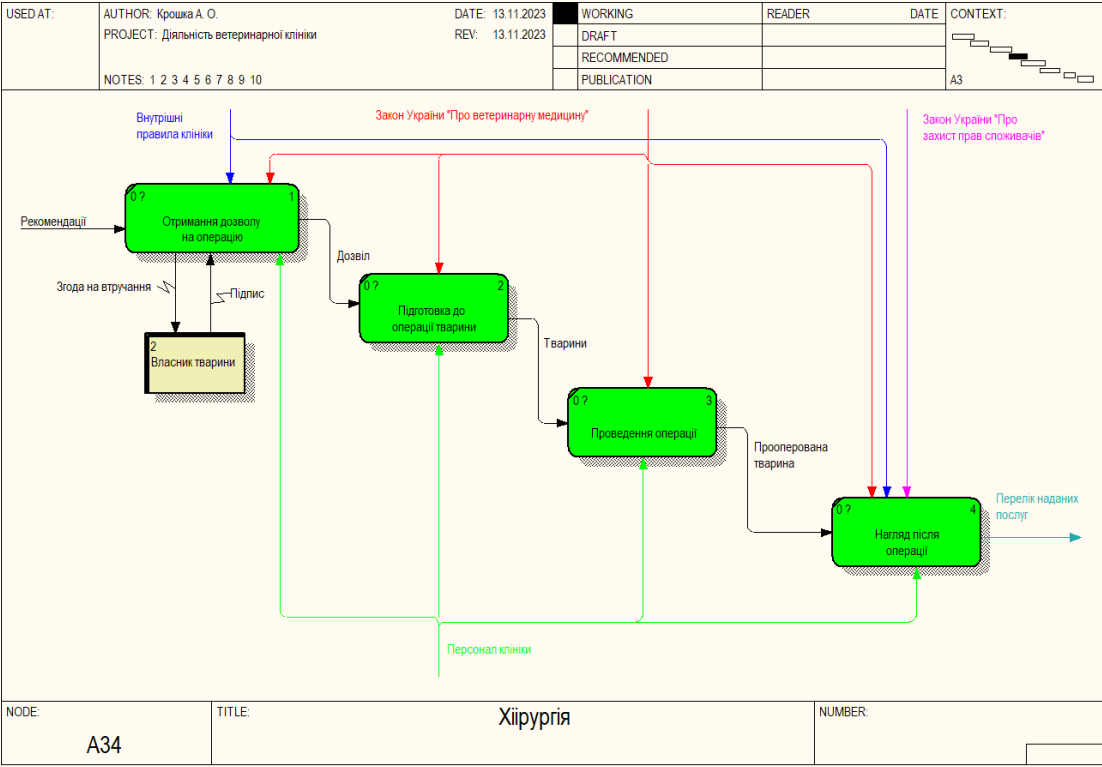


Рисунок 1.5 – Опис процесу «Хірургія»

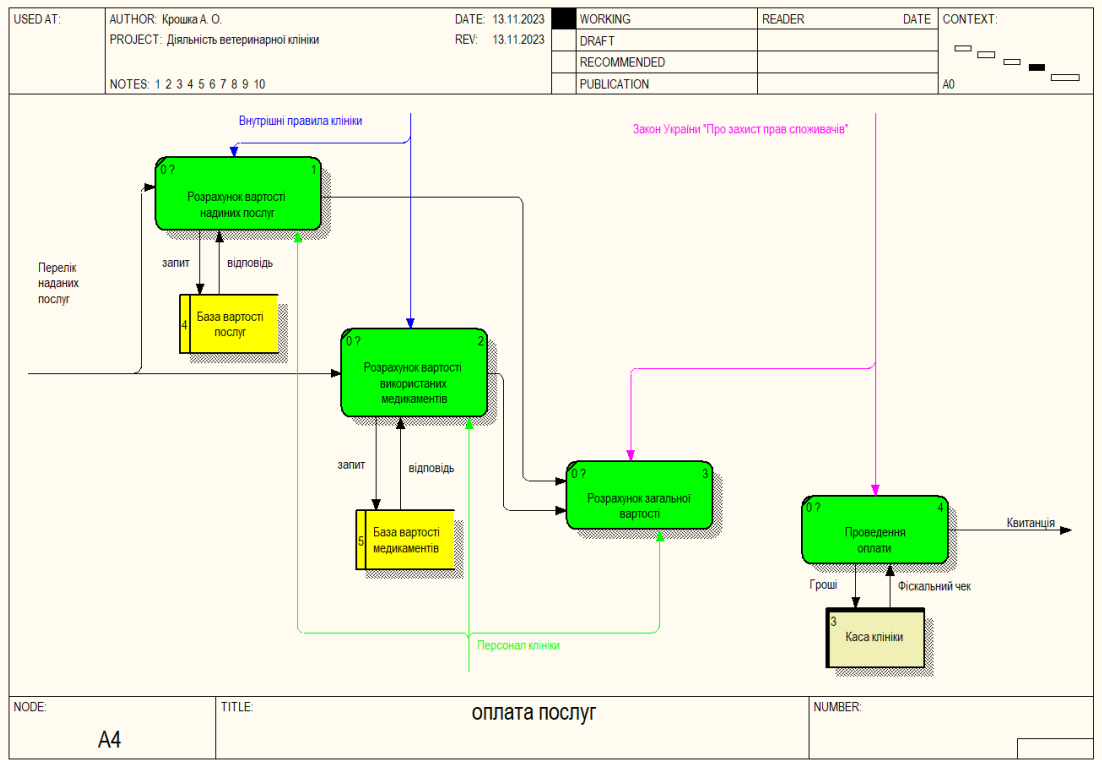


Рисунок 1.6– Опис процесу «Оплата послуг»

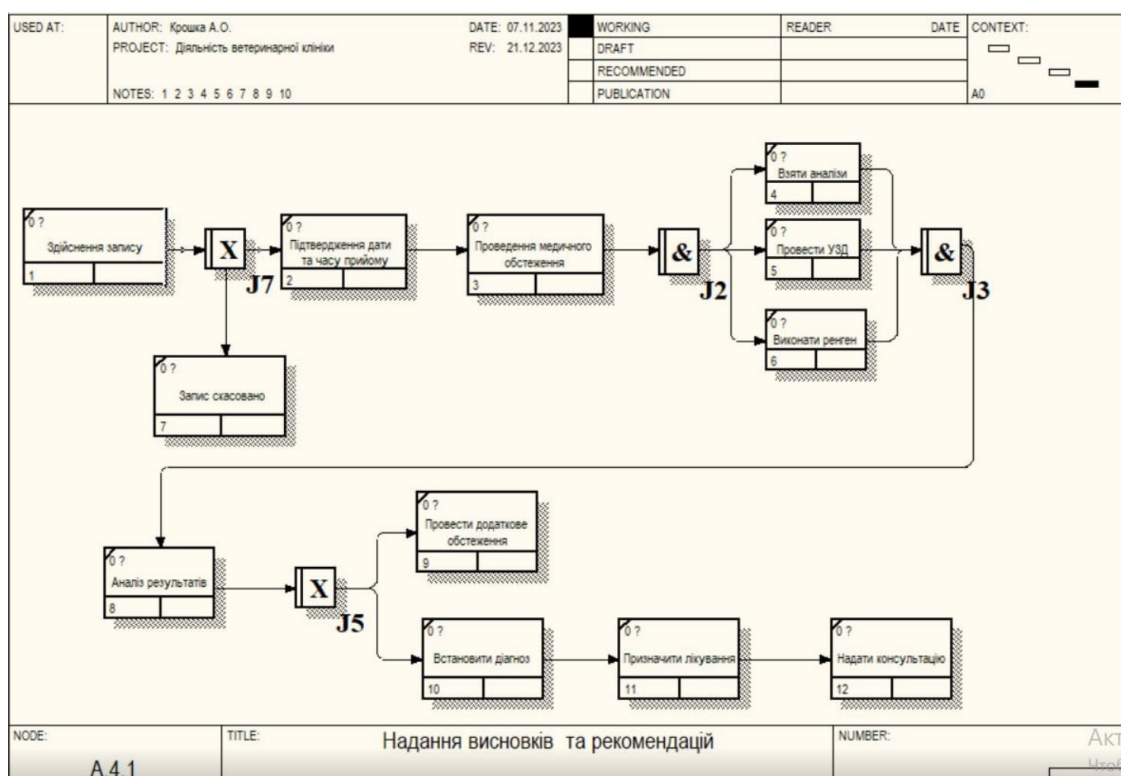


Рисунок 1.7 – IDEF3 «Надання висновків та рекомендацій»

Додаток Б

Ветеринарна клініка

Довідники ?

 **ВЕТЕРИНАРНА КЛІНІКА**

Власник | **Тварини** | **Надані послуги**

Пошук за ПІБ Пошук за адресою

	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 3.1 – Інтерфейс програми для роботи з власниками

Ветеринарна клініка

Власник | **Тварини** | **Надані послуги**

Власник тварин **Крошка Анастасія Олександрівна** **Всіх тварин 2**

Код	Вид тварини	Кличка	Дата народження	Стать	Маса	Стерелізація
1.	Кіт	Барон	13.05.2010	Х	6	☐
2.	Хом'як	Білий	25.09.2023	Д	1	☐

Виберіть дію

☒ Нова тварина

☐ Редагувати вибрану

Дані по тварині

Вид тварини

Кличка

Дата народження

Маса

Стерелізація ☐

Стать ☐ х ☐ д

Активация Windows
Чтобы активировать Windows, зайдите на сайт Windows.com

Рисунок 3.2 – Інтерфейс програми для роботи з тваринами

Довідник "Види тварин"

Новий запис Видалити запис Зберегти зміни

	Код	Назва
▶	2	кішка
	3	собака
	5	миша
	10	єнот
	11	крокодил
	20	Хом'ячок
*		

Рисунок 3.3 – Довідник «Види тварин»

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	Інекція підшкірна або внутрішньо мязова
	6	Внутрішньо венна інекція
	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 3.4 – Довідник «Послуги клініки»



Рисунок 3.5 – Вікно «Про програму»

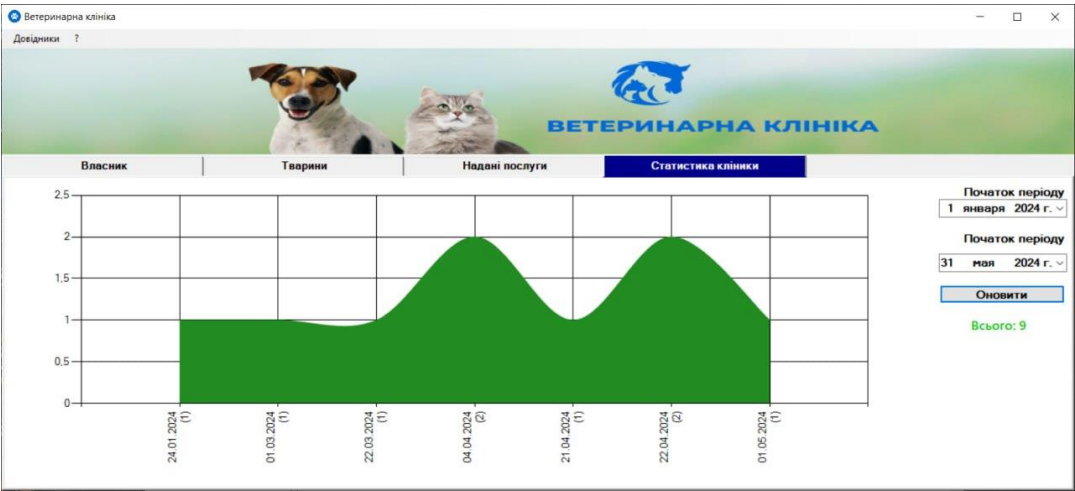


Рисунок 3.6 – Вікно «Статистика клініки»

Додаток В

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using workfile;
using System.Text;
using System.Data.SQLite;
using System.Windows.Forms;
using Ветеринарна_клініка;
using System.Runtime.Versioning;
using System.Linq;
using System.Windows.Forms.DataVisualization.Charting;

namespace VeterinaryClinic
{
    [SupportedOSPlatform("windows")]
    public partial class MainForm : Form
    {
        SQLiteConnection conn;
        SQLiteCommand comand = new SQLiteCommand();
        DataSet DSet = new DataSet();
        DataTable DTable = new DataTable();
        SQLiteDataAdapter dataAdapter;

        SQLiteDataAdapter dataAdapterAnimals;
        DataTable DTableAn = new DataTable();
        DataSet DSetAn = new DataSet();

        SQLiteDataAdapter dataAdapterPos1;
        DataTable DTablePos = new DataTable();
        DataSet DSetPos = new DataSet();
        class ItemForCombo
        {
            public int Id { get; set; }
            public string Value { get; set; }
            public override string ToString()
            {
                return Value;
            }
        }

        public MainForm()
        {
            InitializeComponent();
            DGVanimals.ColumnHeadersDefaultCellStyle.Alignment =
DataGridviewContentAlignment.MiddleCenter;
            DGVanimals.AutoSizeColumnsMode =
DataGridviewAutoSizeColumnsMode.AllCells;

            DGVvlasnyk.ColumnHeadersDefaultCellStyle.Alignment =
DataGridviewContentAlignment.MiddleCenter;
            DGVvlasnyk.AutoSizeColumnsMode =
DataGridviewAutoSizeColumnsMode.AllCells;

```

```

        DGVposlugi.ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        DGVposlugi.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;

        dateTimePickerStart.Value = new DateTime(DateTime.Now.Year,
DateTime.Now.Month, 1);

        DateTime firstDayOfNextMonth = new DateTime(DateTime.Now.Year,
DateTime.Now.Month, 1).AddMonths(1);
        DateTime lastDayOfThisMonth = firstDayOfNextMonth.AddDays(-1);
        dateTimePickerEnd.Value = lastDayOfThisMonth;

        chart1.Series[0].Color = Color.ForestGreen;
    }

    int CurPoz(DataGridView Grid)
    {
        int row = -1;
        if (Grid.DataSource != null)
            row = Grid.BindingContext[Grid.DataSource].Position;
        return row;
    }
    int IdVlas
    {
        get
        {
            int id = -1;
            int row = CurPoz(DGVvlasnyk);
            if (row > -1)
                id
                =
int.Parse(DGVvlasnyk.Rows[row].Cells[0].Value.ToString());
            return id;
        }
    }
    int IdAnimals
    {
        get
        {
            int id = -1;
            int row = CurPoz(DGVanimals);
            if (row > -1)
                id
                =
int.Parse(DGVanimals.Rows[row].Cells[0].Value.ToString());
            return id;
        }
    }
    int IdPosl
    {
        get
        {
            int id = -1;
            int row = CurPoz(DGVposlugi);
            if (row > -1)
                id
                =
int.Parse(DGVposlugi.Rows[row].Cells[0].Value.ToString());
            return id;
        }
    }

```

```

    }
}
int IdMeetings
{
    get
    {
        int id = -1;
        int row = CurPoz(DGVposlugi);
        if (row > -1)
            id
            =
int.Parse(DGVposlugi.Rows[row].Cells[0].Value.ToString());
        return id;
    }
}

private void LikarMenuItem_Click(object sender, EventArgs e)
{
    Directories likar = new Directories();
    likar.Text = "Довідник \"Ветеринари\"";
    likar.Query = "SELECT " +
        "id as 'Код'," +
        "name as 'ПІБ'," +
        "adress as 'Адресса'," +
        "phone as 'Телефон'" +
        " FROM likar";
    likar.ShowDialog();
    LoadDataComboBox(comboBoxLikar, "select id as 'Код', name as 'Назва'
from likar");
}
private void VlasnykMenuItem_Click(object sender, EventArgs e)
{
    Directories vlasnyk = new Directories();
    vlasnyk.Text = "Довідник \"Власники\"";
    vlasnyk.Query = "SELECT " +
        "id as 'Код'," +
        "name as 'ПІБ'," +
        "adress as 'Адресса'," +
        "phone as 'Телефон'" +
        " FROM vlasnyk";
    vlasnyk.ShowDialog();
    LoadDataVlasnyk();
}
private void VydyAnimalsMenuItem_Click(object sender, EventArgs e)
{
    Directories animal = new Directories();
    animal.Text = "Довідник \"Види тварин\"";
    animal.Query = "select id as 'Код', name as 'Назва'      from
TypeAnimals";

    animal.ShowDialog();
    LoadDataComboBox(comboBoxTypAn, animal.Query);
}
private void PoslugyMenuItem_Click(object sender, EventArgs e)
{
    Directories poslugy = new Directories();

```

```

        poslugy.Text = "Довідник \"Послуги\"";
        poslugy.Query = "SELECT " +
            "id as 'Код'," +
            "name as 'ПІБ', " +
            "price as 'Ціна'," +
            "coment as 'Примітка' " +
            "FROM posluga";
        poslugy.ShowDialog();
        LoadDataListView();
        LoadDataPos(IdAnimals, textBox7.Text, Month.SelectedIndex > -1 ?
        (Month.SelectedIndex + 1).ToString("d2") : "");
    }
    private void Set_conect()
    {
        try
        {
            conn = new SQLiteConnection("Data Source=base_clinic.db;
Version=3;");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK,
            MessageBoxIcon.Error);
            Close();
        }
    }
    private void Ex_quary(string quary)
    {
        Set_conect();
        conn.Open();
        comand = conn.CreateCommand();
        comand.CommandText = quary;
        comand.ExecuteNonQuery();
        conn.Close();
    }
    private void LoadDataVlasnyk(string filtrPIB = "", string filtrAdress =
    "")
    {
        Set_conect();
        conn.Open();
        comand = conn.CreateCommand();
        comand.CommandText = "SELECT " +
            "id as 'Код'," +
            "name as 'ПІБ', " +
            "adress as 'Адресса'," +
            "phone as 'Телефон'" +
            $" FROM vlasnyk LIKE WHERE name Like '{filtrPIB}%' and adress
            Like '{filtrAdress}%'";
        dataAdapter = new SQLiteDataAdapter(comand.CommandText, conn);
        DSet.Reset();
        dataAdapter.Fill(DSet);
        DTable = DSet.Tables[0];
        DGVvlasnyk.DataSource = DTable;
        conn.Close();
    }

```

```

        DGVvlasnyk.Columns[0].ReadOnly = true;
        DGVvlasnyk.Columns[DTable.Columns.Count - 1].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;

        LoadDataAnimals(IdVlas);

    }
    private void LoadDataAnimals(int id = 0)
    {
        Set_conect();
        conn.Open();
        comand = conn.CreateCommand();
        comand.CommandText = @"SELECT Animals.ID as 'Код',
            TypeAnimals.name as 'Вид тварини',
            Animals.name as 'Кличка',
            dateb as 'Дата народження',
            pol as 'Стать',
            massa as 'Маса',
            stereal as 'Стерілізація', Primitka as 'Примітка'" +
            $"FROM Animals INNER JOIN TypeAnimals on Animals.id_type =
TypeAnimals.id where id_vlas = {id}";
        dataAdapterAnimals = new SQLiteDataAdapter(comand.CommandText,
conn);

        DSetAn.Reset();
        dataAdapterAnimals.Fill(DSetAn);
        DTableAn = DSetAn.Tables[0];
        DGVanimals.DataSource = DTableAn;
        conn.Close();

        DGVanimals.Columns[0].ReadOnly = true;
        DGVanimals.Columns[DTableAn.Columns.Count - 1].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;

        SetText();
        LoadDataPos(IdAnimals, textBox7.Text, Month.SelectedIndex > -1 ?
(Month.SelectedIndex + 1).ToString("d2") : "");

    }
    private void LoadDataPos(int id = 0, string findpos = "", string datap
= "")
    {
        Set_conect();
        conn.Open();
        comand = conn.CreateCommand();

        comand.CommandText = @"SELECT meeting.ID as 'Код',
            data as 'Дата',
            diagnoz as 'Діагноз',
            posluga.name as 'Послуга',
            posluga.price as 'Ціна',
            likar.name as 'Лікар',
            posluga.id as 'код_послуги',
            prymitka as 'Примітка'
            FROM meeting
            INNER JOIN posluga on meeting.id_posluga = posluga.id
            INNER JOIN likar on meeting.id_likar = likar.id " +

```

```

        $" where id_animal = {id} AND posluga.name LIKE
'#{findpos}%'" +
        $"AND strftime('%m', data) Like ('#{datap}%')";
dataAdapterPos1 = new SQLiteDataAdapter(comand.CommandText, conn);
DSetPos.Reset();
dataAdapterPos1.Fill(DSetPos);
DTablePos = DSetPos.Tables[0];
DGVposlugi.DataSource = DTablePos;
conn.Close();
double sum = 0;

DGVposlugi.Columns[0].ReadOnly = true;
DGVposlugi.Columns[DTablePos.Columns.Count - 2].Visible = false;
DGVposlugi.Columns[DTablePos.Columns.Count - 1].AutoSizeMode =
DataGridViewAutoSizeColumnMode.Fill;

for (int i = 0; i < DGVposlugi.Rows.Count; i++)
{
    double.TryParse(DGVposlugi.Rows[i].Cells[4].Value.ToString(),
out double rez);
    sum += rez;
}
labelSUM.Text = sum.ToString("N2") + " грн.";
SetText();
}
private void LoadDataComboBox(ComboBox combo, string q)
{
    Set_conect();
    conn.Open();
    comand = conn.CreateCommand();
    comand.CommandText = q;
    var rdr = comand.ExecuteReader();
    combo.Items.Clear();
    while (rdr.Read())
    {
        combo.Items.Add(new ItemForCombo { Id = rdr.GetInt32(0), Value
= rdr.GetString(1) });
    }
    conn.Close();
}
private void LoadDataListView()
{
    Set_conect();
    conn.Open();
    comand = conn.CreateCommand();
    comand.CommandText = "SELECT " +
        "id as 'Код'," +
        "name as 'ПІБ'," +
        "price as 'Ціна' " +
        "FROM posluga";
    var rdr = comand.ExecuteReader();
    listView1.Items.Clear();
    while (rdr.Read())

```

```

    {
        ListViewItem lvi = new ListViewItem();
        lvi.Text = rdr.GetString(1);
        lvi.Tag = rdr.GetInt32(0);
        lvi.SubItems.Add(rdr.GetDouble(2).ToString());
        listView1.Items.Add(lvi);
    }
    conn.Close();

}

private void MainForm_Load(object sender, EventArgs e)
{
    TestTime.TestNow(1724962184);
    LoadDataVlasnyk();
    LoadDataComboBox(comboBoxTypAn, "select id as 'Код', name as 'Назва'
from TypeAnimals");
    LoadDataComboBox(comboBoxLikar, "select id as 'Код', name as 'Назва'
from likar");
    LoadDataListView();
}
private void textBox1_TextChanged(object sender, EventArgs e)
{
    LoadDataVlasnyk(textBox1.Text, textBox2.Text);
}
private void SetText()
{
    int OldPozVlas = CurPoz(DGVvlasnyk);
    int OldPozAnimal = CurPoz(DGVanimals);
    if (OldPozVlas > -1)
    {
        label3.Text
DGVvlasnyk.Rows[OldPozVlas].Cells[1].Value.ToString() + $"          Всіх тварин
{DGVanimals.Rows.Count}";
    }
    else
        label3.Text = "Невідомо";
    if (OldPozAnimal > -1)
    {
        label15.Text
DGVanimals.Rows[OldPozAnimal].Cells[1].Value.ToString();
        label15.Text += "          Кличка      \"\"\"      +
DGVanimals.Rows[OldPozAnimal].Cells[2].Value.ToString() + "\"\" \"";
        label15.Text += "          (          \"          +
DGVanimals.Rows[OldPozAnimal].Cells[4].Value.ToString() + \" ) \"";
    }
    else
    {
        label15.Text = "Невідомо";
    }
}
private void SaveAnimal_Click(object sender, EventArgs e)
{
    if (comboBoxTypAn.SelectedItem == null)

```

```

        {
            MessageBox.Show("Виберіть вид тварини", "Інформація",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            return;
        }
        if (IdVlas == -1)
        {
            MessageBox.Show("Не вибрано власника", "Інформація",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            return;
        }

        int idtyp = ((ItemForCombo)comboBoxTypAn.SelectedItem).Id;
        char pol = radioButton1.Checked ? radioButton1.Text[0] :
radioButton2.Text[0];

        int curPozAnimals = CurPoz(DGVAnimals);
        Set_conect();
        conn.Open();
        if (radioEdit.Checked)
        {
            if (curPozAnimals != -1)
            {
                comand = conn.CreateCommand();
                if
(!int.TryParse(DGVAnimals.Rows[curPozAnimals].Cells[0].Value.ToString(), out int ID))
                    ID = 0;
                comand.CommandText = $"UPDATE Animals SET name =
'{{namek1.Text}}', id_type = '{{idtyp}}', pol = '{{pol}}', " +
                    $" massa = '{{massatb.Text.ToString().Replace(',', ' '}}',
stereal = '{{checkBox1.Checked}}', id_vlas = '{{IdVlas}}', dateb =
'{{dateTimePicker1.Value.ToString("yyyy-MM-dd")}}', " +
                    $" Primitka = '{{textBox5.Text}}' WHERE ID = {ID}";
                comand.ExecuteScalar();
                conn.Close();
                LoadDataAnimals(IdVlas);
                DGVAnimals.BindingContext[DGVAnimals.DataSource].Position
= curPozAnimals;
                MessageBox.Show("Дані оновлено", "Інформація",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
        }
        else
        {
            comand = conn.CreateCommand();
            comand.CommandText = "INSERT INTO Animals(name, id_type, pol,
massa, stereal, id_vlas, dateb, Primitka )" +
                $"VALUES
('{{namek1.Text}}', '{{idtyp}}', '{{pol}}', '{{massatb.Text.ToString().Replace(',', ' '}}', " +
                $" '{{checkBox1.Checked}}', '{{IdVlas}}', '{{dateTimePicker1.Value.ToString("yyyy-MM-
dd")}}', '{{textBox5.Text}}')";
            comand.ExecuteScalar();
            conn.Close();
            Vanimals_Clear();
        }
    }
}

```

```

        LoadDataAnimals(IdVlas);
        MessageBox.Show("Дані          добавлено",          "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

}
private void SavePos_Click(object sender, EventArgs e)
{
    if (IdVlas == -1)
    {
        MessageBox.Show("Не          вибрано          власника",          "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (IdAnimals == -1)
    {
        MessageBox.Show("Не          вибрано          тварину",          "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (comboBoxLikar.SelectedItem == null)
    {
        MessageBox.Show("Виберіть          вид          лікаря",          "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    if (listView1.SelectedItems == null)
    {
        MessageBox.Show("Виберіть          вид          послугу",          "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }

    int idlikar = ((ItemForCombo)comboBoxLikar.SelectedItem).Id;
    int idpos = (int)listView1.SelectedItems[0].Tag;

    int curPozPos = CurPoz(DGVposlugi);
    Set_conect();
    conn.Open();
    if (radioB_pos.Checked)
    {
        comand = conn.CreateCommand();

        if (IdMeetings < 0)
        {
            MessageBox.Show("Виберіть послугу яку потрібно редагувати",
            "Інформація", MessageBoxButtons.OK, MessageBoxIcon.Information);
            return;
        }
        comand.CommandText = $"UPDATE meeting  " +
            $"SET  " +
            $"data = '{dateTimePicker2.Value.ToString("yyyy-MM-dd")}' , "
+

```

```

        $"id_posluga = '{idpos}', " +
        $"id_likar = '{idlikar}', " +
        $"diagnoz = '{textBoxDiagnoz.Text}'," +
        $"prymitka = '{textBoxPosPrim.Text}' WHERE ID =
{IdMeetings}";

        comand.ExecuteScalar();
        conn.Close();
        LoadDataPos(IdAnimals, textBox7.Text, Month.SelectedIndex > -1
? (Month.SelectedIndex + 1).ToString("d2") : "");
        DGVposlugi.BindingContext[DGVposlugi.DataSource].Position =
curPozPos;

        MessageBox.Show("Дані оновлено", "Інформація",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else
    {
        comand = conn.CreateCommand();
        comand.CommandText = "INSERT INTO meeting( id_posluga, id_likar,
id_animal, diagnoz,data, prymitka )" +
        $"VALUES
('{idpos}','{idlikar}','{IdAnimals}','{textBoxDiagnoz.Text}'," +
        $"'{dateTimePicker2.Value.ToString("yyyy-MM-
dd")}','{textBoxPosPrim.Text}')";
        comand.ExecuteScalar();
        conn.Close();
        Vanimals_Clear();
        LoadDataPos(IdAnimals, textBox7.Text, Month.SelectedIndex > -1
? (Month.SelectedIndex + 1).ToString("d2") : "");
        MessageBox.Show("Дані добавлено", "Інформація",
MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
}

private void Vanimals_Clear()
{
    comboBoxTypAn.SelectedIndex = -1;
    namek1.Text = "";
    dateTimePicker1.Value = DateTime.Now;
    radioButton1.Checked = true;
    radioButton2.Checked = false;
    massatb.Text = "0";
    checkBox1.Checked = false;
    textBox5.Text = "";
}

private void VPos_Clear()
{
    dateTimePicker2.Value = DateTime.Now;
    textBoxDiagnoz.Text = "";
    comboBoxLikar.SelectedIndex = -1;
    listView1.SelectedIndices.Clear();
    textBoxPosPrim.Text = "";
}

```

```

private void DGVanimals_Edit()
{
    if (DGVanimals.SelectedCells.Count > 0)
    {
        int row = DGVanimals.SelectedCells[0].RowIndex;
        comboBoxTypAn.Text =
DGVanimals.Rows[row].Cells[1].Value.ToString();
        namekl.Text = DGVanimals.Rows[row].Cells[2].Value.ToString();
        dateTimePicker1.Text =
DGVanimals.Rows[row].Cells[3].Value.ToString();

        radioButton1.Checked =
(DGVanimals.Rows[row].Cells[4].Value.ToString() == radioButton1.Text);
        radioButton2.Checked =
(DGVanimals.Rows[row].Cells[4].Value.ToString() == radioButton2.Text);

        massatb.Text = DGVanimals.Rows[row].Cells[5].Value.ToString();

        checkBox1.Checked =
(DGVanimals.Rows[row].Cells[6].Value.ToString() == "True");
        textBox5.Text = DGVanimals.Rows[row].Cells[7].Value.ToString();
    }
    else
        MessageBox.Show("Не вибрано кого редагувати", "Редагувати",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
private void DGPosluga_Edit()
{
    try
    {
        if (DGVposlugi.SelectedCells.Count > 0)
        {
            int row = DGVposlugi.SelectedCells[0].RowIndex;
            dateTimePicker2.Text =
DGVposlugi.Rows[row].Cells[1].Value.ToString();
            textBoxDiagnoz.Text =
DGVposlugi.Rows[row].Cells[2].Value.ToString();

            comboBoxLikar.Text =
DGVposlugi.Rows[row].Cells[5].Value.ToString();

            int.TryParse(DGVposlugi.Rows[row].Cells[6].Value.ToString(), out int id_typ);

            int i;
            for (i = 0; i < listView1.Items.Count; i++)
            {
                if (listView1.Items[i].Tag is int a)
                {
                    if (a == id_typ)

```

```

        break;
    }
}
if (listView1.Items.Count != i)
{
    this.listView1.Items[i].Focused = true;
    this.listView1.Items[i].Selected = true;
    listView1.EnsureVisible(i);
}
textBoxPosPrim.Text =
DGVposlugi.Rows[row].Cells[7].Value.ToString();
}
else
    MessageBox.Show("Не вибрано кого редагувати", "Редагувати",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Помилка", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}

}
private void radioEdit_CheckedChanged(object sender, EventArgs e)
{
    if (radioEdit.Checked)
    {
        SaveAnimal.Text = "Зберегти зміни";
        DGVanimals_Edit();
    }
    else
    {
        Vanimals_Clear();
        SaveAnimal.Text = "Зберегти тварину";
    }
}
private void radioEditPos_CheckedChanged(object sender, EventArgs e)
{
    if (radioB_pos.Checked)
    {
        SaveAnimal.Text = "Зберегти зміни";
        DGPosluga_Edit();
    }
    else
    {
        VPos_Clear();
        SaveAnimal.Text = "Зберегти послугу";
    }
}
private void DeleteAnimal_Click(object sender, EventArgs e)
{
    if (DGVanimals.DataSource == null)
    {

```

```

        MessageBox.Show("Не вибрано кого видаляти", "Видалення",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    int curPozAnimals = CurPoz(DGVAnimals);
    if (curPozAnimals != -1)
    {
        if (MessageBox.Show("Видалити запис", "Видалення",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2) ==
        DialogResult.No)
            return;
        Set_conect();
        conn.Open();
        comand = conn.CreateCommand();
        comand.CommandText = $"DELETE FROM Animals WHERE ID =
        '{IdAnimals}'";
        comand.ExecuteScalar();
        conn.Close();
        LoadDataAnimals(IdVlas);
        MessageBox.Show("Запис видалено", "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else MessageBox.Show("Не вибрано рядок для видалення", "Видалення",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    private void textBox7_TextChanged(object sender, EventArgs e)
    {
        LoadDataPos(IdAnimals, textBox7.Text, Month.SelectedIndex > -1 ?
        (Month.SelectedIndex + 1).ToString("d2") : "");
    }
    private void DGVvlasnyk_RowValidating(object sender,
    DataGridViewCellCancelEventArgs e)
    {
        LoadDataAnimals(IdVlas);
    }
    private void DGVAnimals_RowValidating(object sender,
    DataGridViewCellCancelEventArgs e)
    {
        LoadDataPos(IdAnimals, textBox7.Text, Month.SelectedIndex > -1 ?
        (Month.SelectedIndex + 1).ToString("d2") : "");
    }
    private void listView1_DrawColumnHeader(object sender,
    DrawListViewColumnHeaderEventArgs e)
    {
        e.DrawDefault = true;
    }
    /// <summary>
    /// Оформлення ListView
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void listView1_DrawItem(object sender,
    DrawListViewItemEventArgs e)
    {
        ListView listView = (ListView)sender;

        // Check if e.Item is selected and the ListView has a focus.

```

```

        if (!listView.Focused && e.Item.Selected)
        {
            Rectangle rowBounds = e.Bounds;
            int leftMargin =
e.Item.GetBounds(ItemBoundsPortion.Label).Left;
            Rectangle bounds = new Rectangle(leftMargin, rowBounds.Top,
rowBounds.Width - leftMargin, rowBounds.Height);
            e.Graphics.FillRectangle(SystemBrushes.Highlight, bounds);
        }
        else
            e.DrawDefault = true;
    }
    /// <summary>
    /// Отрисовка элементов ListView
    /// </summary>
    /// <param name="sender"></param>
    /// <param name="e"></param>
    private void listView1_DrawSubItem(object sender,
DrawListViewSubItemEventArgs e)
    {
        const int TEXT_OFFSET = 1; // I don't know why the text is located
at 1px to the right. Maybe it's only for me.

        ListView listView = (ListView)sender;

        // Check if e.Item is selected and the ListView has a focus.
        if (!listView.Focused && e.Item.Selected)
        {
            Rectangle rowBounds = e.SubItem.Bounds;
            Rectangle labelBounds =
e.Item.GetBounds(ItemBoundsPortion.Label);
            int leftMargin = labelBounds.Left - TEXT_OFFSET;
            Rectangle bounds = new Rectangle(rowBounds.Left + leftMargin,
rowBounds.Top, e.ColumnIndex == 0 ? labelBounds.Width : (rowBounds.Width - leftMargin
- TEXT_OFFSET), rowBounds.Height);
            TextFormatFlags align;
            switch (listView.Columns[e.ColumnIndex].TextAlign)
            {
                case HorizontalAlignment.Right:
                    align = TextFormatFlags.Right;
                    break;
                case HorizontalAlignment.Center:
                    align = TextFormatFlags.HorizontalCenter;
                    break;
                default:
                    align = TextFormatFlags.Left;
                    break;
            }
            TextRenderer.DrawText(e.Graphics, e.SubItem.Text,
listView.Font, bounds, SystemColors.HighlightText,
align | TextFormatFlags.SingleLine |
TextFormatFlags.GlyphOverhangPadding | TextFormatFlags.VerticalCenter |
TextFormatFlags.WordEllipsis);
        }
        else
            e.DrawDefault = true;
    }
}

```

```

private void buttonDeletePosl_Click(object sender, EventArgs e)
{
    if (DGVposlugi.DataSource == null)
    {
        MessageBox.Show("Не вибрано кого видаляти", "Видалення",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
    int curPozpos = CurPoz(DGVposlugi);
    if (curPozpos != -1)
    {
        if (MessageBox.Show("Видалити запис", "Видалення",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2) ==
        DialogResult.No)
            return;
        Set_conect();
        conn.Open();
        comand = conn.CreateCommand();
        comand.CommandText = $"DELETE FROM meeting WHERE ID =
        '{IdMeetings}'";
        comand.ExecuteScalar();
        conn.Close();
        LoadDataPos(IdAnimals);
        MessageBox.Show("Запис видалено", "Інформація",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    else MessageBox.Show("Не вибрано рядок для видалення", "Видалення",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
}

private void Month_TextChanged(object sender, EventArgs e)
{
    Month.Text = "";
    textBox7_TextChanged(sender, e);
}

private void tabControl1_DrawItem(object sender, DrawItemEventArgs e)
{
    e.Graphics.SetClip(e.Bounds);
    string text = tabControl1.TabPages[e.Index].Text;
    SizeF sz = e.Graphics.MeasureString(text, e.Font);
    var tabTextArea = (RectangleF)tabControl1.GetTabRect(e.Index);

    bool bSelected = (e.State & DrawItemState.Selected) ==
    DrawItemState.Selected;
    using (SolidBrush b = new SolidBrush(bSelected ? Color.DarkBlue :
    SystemColors.Control))
        e.Graphics.FillRectangle(b, e.Bounds);

    using (SolidBrush b = new SolidBrush(bSelected ? Color.White :
    Color.Black))
    {
        StringFormat stringFormat = new StringFormat();
        stringFormat.Alignment = StringAlignment.Center;
    }
}

```

```

        stringFormat.LineAlignment = StringAlignment.Center;
        e.Graphics.DrawString(text, e.Font, b, tabTextArea,
stringFormat);
    }
    if (tabControl1.SelectedIndex == e.Index)
        e.DrawFocusRectangle();

    e.Graphics.ResetClip();
}

private void toolStripMenuItem1_Click(object sender, EventArgs e)
{
    About about = new About();
    about.ShowDialog();
}

private int GetId(string query)
{
    int id = -1;
    Set_conect();
    conn.Open();
    if (conn == null) { return id; }
    SQLiteCommand command = conn.CreateCommand();
    command.CommandText = query;
    // Виконання SQL-запиту і отримання результату
    object result = command.ExecuteScalar();
    // Перевірка на null та конвертація в int
    if (result != null && result != DBNull.Value)
        id = Convert.ToInt32(result);
    conn.Close();
    return id;
}

private void buttonRefreshStatistic_Click(object sender, EventArgs e)
{
    Set_conect();
    conn.Open();
    if (conn == null) { return; }
    SQLiteCommand command = conn.CreateCommand();
    string query = "SELECT data as date, count(*) as couter " +
        "FROM meeting " +
        $"WHERE data BETWEEN '{dateTimePickerStart.Value:yyyy-MM-dd}'
and '{dateTimePickerEnd.Value:yyyy-MM-dd}' " +
        "group by data ";
    command.CommandText = query;
    var reader = command.ExecuteReader();
    int i = 0;
    chart1.Series[0].Points.Clear();
    while (reader.Read())
    {
        DateTime date = reader.GetDateTime(0);
        Int64 counter = reader.GetInt64(1);
        DataPoint point = new DataPoint(i, counter);
        point.AxisLabel = $"{date:d}\n({counter})";
        //point.AxisLabel = $"123456666 -{counter}";
        chart1.Series[0].Points.Add(point);
        i++;
    }
}

```

```

    }
    conn.Close();
    chart1.ChartAreas[0].AxisX.LabelStyle.Angle = -90;
    chart1.ChartAreas[0].AxisX.LabelStyle.Font = new Font("Arial", 10);

    //
    labelAllStatistic.Text = "Всего: " + GetId($"SELECT count(*) as
couter FROM meeting " +
        $"WHERE data BETWEEN '{dateTimePickerStart.Value:yyyy-MM-dd}'
and '{dateTimePickerEnd.Value:yyyy-MM-dd}' ");
    }

    private void tabControl1_SelectedIndexChanged(object sender, EventArgs
e)
    {

        if (tabControl1.SelectedTab == tabPageStatistics)
        {
            buttonRefreshStatistic_Click(sender, e);
        }
    }
}

```