

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

ПОЯСНЮВАЛЬНА ЗАПИСКА

(до кваліфікаційної роботи освітнього ступеня «бакалавр»
за спеціальністю 121 «Інженерія програмного забезпечення»
(освітня програма «Інженерія програмного забезпечення»)

на тему:

«Сайт ветеринарної клініки»

Виконала студентка групи ІПЗ-20-4 ЛОСЬ
Ліна Анатоліївна

Керівник роботи:
МОРОЗОВ Андрій Васильович

Рецензент:
САВІЦЬКИЙ Роман Святославович

Житомир – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

«ЗАТВЕРДЖУЮ»

Зав. кафедри інженерії
програмного забезпечення
Тетяна ВАКАЛЮК
“26”лютого 2024 р.

ЗАВДАННЯ
на кваліфікаційну роботу

Здобувач вищої освіти: **ЛОСЬ Ліна Анатоліївна**

Керівник роботи: **МОРОЗОВ Андрій Васильович**

Тема роботи: **«Сайт ветеринарної клініки»**, затверджена Наказом закладу вищої освіти від **“23”лютого 2024 р., №74/с.**

Вихідні дані до роботи: Предмет дослідження – використання технології, моделі, структури, засоби та методи для розробки та взаємодії системи сайту ветеринарної клініки та її модулів управління. Об'єкт дослідження – методи та засоби розробки веб-сайту за визначеними предметними областями.

Консультанти з бакалаврської кваліфікаційної роботи із зазначенням розділів, що їх стосуються:

Розділ	Консультант	Завдання видав	Завдання прийняв
1	Морозов А.В.	02.03.24	02.03.24
2	Морозов А.В.	05.04.24	05.04.24
3	Морозов А.В.	01.05.24	01.05.24

РЕФЕРАТ

Випускна кваліфікаційна робота бакалавра складається з вебсайту за обраною тематикою та пояснювальної записки. Пояснювальна записка до випускної роботи містить 58 сторінок, 21 ілюстрацію та 19 таблиць.

Мета створення веб-сайту ветеринарної клініки полягає в наданні доступу до важливої інформації про послуги клініки, команду фахівців, режим роботи та контактну інформацію. Він також спрямований на створення зручного інтерфейсу для клієнтів, де вони можуть здійснювати онлайн-запис на прийом, переглядати корисні поради щодо догляду за тваринами, а також отримувати актуальні новини та статті з області ветеринарної медицини.

Додаток має зручний інтерфейс, який дозволяє користувачам легко організовувати записи та консультації, а також швидко знаходити необхідну інформацію завдяки зручному пошуковому механізму. Кожна функція доступна всього кількома клацаннями, що дозволяє ефективно використовувати додаток.

КЛЮЧОВІ СЛОВА: ФУНКЦІОНАЛ, МОДЕЛЮВАННЯ, ВЕБ-САЙТ, БАЗА ДАНИХ, МОДЕЛЬ, КЛІНІКА, ЦЕНТР.

					ІПЗ.КР.Б – 121 – 24 – ІПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.	Лось Л.А.				Сайт ветеринарної клініки	Літ.	Арк.	Аркушів
Керівник	Морозов А.В.						3	58
Рецензент	Савіцький Р.С.				Пояснювальна записка	Житомирська політехніка, група ІПЗ-20-4		
Зав.каф.	Вакалюк Т.А.							

ABSTRACT

The graduation qualification of the bachelor's work consists of a website for the selected topic and explanatory note. An explanatory note before graduation contains 58 pages, 21 illustrations and 19 tables.

The website of the veterinary clinic provides access to important information about the clinic's services, team of front-line workers, work hours and contact information. You can also directly access the manual interface for clients, where they can make an online appointment, look through records to keep an eye on animals, and also watch the latest news and articles in the field of veterinary medicine tsini.

The add-on has a user-friendly interface that allows clinicians to easily organize appointments and consultations, as well as quickly find the necessary information via the manual sound mechanism. The skin function is accessible to any number of clicks, which allows you to effectively use the supplement. KEY WORDS: FUNCTIONAL, MODELING, WEBSITE, DATABASE, MODEL, CLINIC, SITE.

					ІПЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

РЕФЕРАТ.....	3
ABSTRACT.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП.....	7
РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ	9
1.1 Постановка задачі	9
1.2 Аналіз вимог до програмного продукту.....	11
1.3 Аналіз аналогів програмного продукту.....	14
1.4 Вибір архітектури веборієнтованої системи ветеринарної клініки	16
1.5 Обґрунтування вибору засобів реалізації та вимоги до апаратного забезпечення	17
Висновки до першого розділу	22
РОЗДІЛ 2 ПРОЄКТУВАННЯ САЙТУ ВЕТЕРИНАРНОЇ КЛІНІКИ	23
2.1 Визначення варіантів використання та об'єктно-орієнтованої структури системи	23
2.2 Розробка бази даних системи.....	28
2.3 Проектування та реалізація алгоритмів роботи системи	32
2.4 Реалізація функціонала системи сайту ветеринарної клініки	36
Висновки до другого розділу.....	39
РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З САЙТОМ ВЕТЕРИНАРНОЇ КЛІНІКИ.....	40
3.1 Порядок встановлення та налаштування системи	40
3.2 Інтерфейс та порядок роботи	42
3.3 Тестування роботи сайту ветеринарної клініки	46
Висновки до третього розділу	49
ВИСНОВКИ	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТКИ	53

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – База даних

ПЗ – Програмне забезпечення

ОС – Операційна система

UML – Уніфікована мова моделювання

MVC – Схема розділення даних додатків і керуючої логіки на три окремі компоненти Model-View-Controller

CRUD – Основні операції, що використовуються для управління даними (Create, Read, Update, Delete)

СУБД – Система управління базами даних

CMS – Content Management System

API – Application Programming Interface (інтерфейс програмування додатків)

HTML – Hypertext Markup Language (мова розмітки гіпертексту)

CSS – Cascading Style Sheets (каскадні таблиці стилів)

QA – Quality Assurance (забезпечення якості)

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

У даній кваліфікаційній роботі буде представлено процес моделювання структури веб-сайту ветеринарного центру, спрямований на покращення доступності інформації та послуг для власників тварин та оптимізацію робочих процесів клініки.

У сучасному світі інформаційні технології стають все більш необхідними для розвитку та вдосконалення різних сфер діяльності. Впровадження інформаційних технологій спричинює постійний потік для розвитку та вдосконалення вебдодатків та сайтів для задоволення висхідних вимог користувачів. Один з ключових аспектів успіху вебпроектів - це правильна структура та організація вебсайту. Ветеринарна медицина, як складова частина охорони здоров'я тварин, також використовує потужність цифрових рішень для покращення якості надання послуг та забезпечення комфорту та безпеки домашніх улюбленців.

Актуальність роботи становить в використанні сучасних технологій проектування структури системи сайту, з врахуванням постійного зростання інтересу до цифрових рішень у сфері ветеринарної медицини.

Мета роботи полягає в розробці та впровадженні вебсайту для ветеринарної клініки з метою полегшення доступу до інформації для власників тварин та оптимізації процесів роботи центру. Для успіху проекту потрібно проаналізувати потреби користувачів, спроектувати зручний інтерфейс, оптимізувати для мобільних пристроїв та забезпечити безпеку даних з використанням сучасних технологій.

Завданням роботи є:

- дослідження та аналіз особливостей моделювання програмного забезпечення за потребами цільової аудиторії;
- проектування об'єктної структури системи за обраною тематикою;
- аналіз та опис вимог, які стосуються функціональності та інтерфейсу програмного продукту;
- використання методів моделювання функцій, взаємодії компонентів та

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

поведінки системи;

- розробка фізичної моделі програмного комплексу, включаючи вибір технологій, архітектурних рішень та визначення інфраструктури потрібних для розгортання.

Об'єкт дослідження – методи та засоби розробки вебсайту за визначеними предметними областями.

Предмет дослідження – використання технології, моделі, структури, засоби та методи для розробки та взаємодії системи сайту ветеринарної клініки та її модулів управління.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1 АНАЛІЗ НАПРЯМКІВ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ РЕАЛІЗАЦІЇ СИСТЕМИ ВЕТЕРИНАРНОЇ КЛІНІКИ

1.1 Постановка задачі

Програмний комплекс для ветеринарної клініки має на меті полегшити та оптимізувати робочі процеси в клініці, а також покращити обслуговування клієнтів та догляд за тваринами. Для забезпечення неперервного виконання необхідних функцій і відповідності встановленим стандартам та специфікаціям, наведено високорівневі вимоги, які повинна задовольняти веборієнтована система планування сайту за обраною тематикою. Ці вимоги гарантують надійність та відповідність всім необхідним стандартам та технічним характеристикам системи.

Проект створюється з метою не лише реалізації інноваційного середовища, але й забезпечення якісного, зручного обслуговування тварин та задоволення потреб клієнтів, що включає різноманітні аспекти, спрямовані на підвищення якості життя тварин і їхніх власників.

У кваліфікаційній роботі необхідно розробити вебсайт для ветеринарної клініки.

Для реалізації поставленого завдання, основними етапами є:

- Детальний аналіз проєкту, аналогових продуктів та збір необхідних вимог та характеристик.
- Розробка та побудова клієнт-серверної архітектури.
- Розробка системи авторизації та ролей користувачів.
- Реалізація основних функцій та впровадження системи.
- Тестування та забезпечення якості.

Вимоги до програмного забезпечення для ефективного функціонування сайту за заданою тематикою визначаються комплексним підходом до моделі побудови, використанням патернів проєктування, забезпеченням швидкої відповіді сервера та іншими ключовими параметрами. Модель побудови сайту забезпечує:

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- масштабованість (забезпечення можливості легкої масштабованості для врахування зростання трафіку та обсягів контенту);
- модульність (використання модульної архітектури для легкості розширення та модифікації функціонала);
- безпека(впровадження механізмів захисту даних та користувачів, враховуючи основні практики веббезпеки);
- оптимізація зображень та медіафайлів (зменшення розміру зображень та медіафайлів для швидшого завантаження сторінок);
- кешування (використання систем кешування для прискорення обробки запитів та зменшення навантаження на сервер);
- CDN (розподілення контенту та забезпечення швидкого доступу для користувачів з різних регіонів);
- оптимізація запитів до бази даних (забезпечення оптимальних запитів до бази даних для швидкого отримання необхідної інформації);
- інтуїтивний інтерфейс та підтримка мобільної сумісності.

Клієнтська частина повинна бути спроектована та побудована з умовами не високого навантаження на браузер, без додаткових завантажень компонентів. Для її створення потрібно використати стандартні технології веброзробки клієнтської частини. Програмний комплекс повинен бути побудований з використання вебтехнологій.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						10
Змн.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз вимог до програмного продукту

Проект має на меті створення активної та згуртованої спільноти в конкретному напрямку, що сприятиме полегшенню взаємодії між різними групами користувачів.

Представлення проекту буде реалізовано у вигляді сайту, який структуровано та інтуїтивно сприймається та з застосуванням сучасного дизайну та інтерактивних елементів, що дозволить забезпечити привабливий та функціональний інтерфейс для кожного користувача.

Нижче наведено основні вимоги, що будуть гарантують надійність та відповідність всім необхідним стандартам та технічним характеристикам системи.

Для забезпечення неперервного виконання необхідних функцій і відповідності встановленим стандартам та специфікаціям, наведено високорівневі вимоги, які повинна задовольняти веборієнтована система планування сайту клубу веб-новел. Ці вимоги гарантують надійність та відповідність всім необхідним стандартам та технічним характеристикам системи.

Вимоги користувачів:

1. Можливість легко та доступно користуватися сайтом.
2. Можливість користувацької реєстрації і доступу до власного профілю.
3. Можливість використання системи пошуку та категоризації.
4. Можливість додавання до сайту власних коментарів та обговорень.
5. Можливість захистити власну приватність даних та гарантування їх безпеки.
6. Можливість отримання зворотного зв'язку з підтримкою сайту.

Вимоги персоналу сайту:

1. Зручний інтерфейс для внесення даних.
2. Підтримка різних типів контенту. Можливість додавання різних типів контенту, таких як текст, зображення, відео, аудіо та інші мультимедійні елементи.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

3. Можливість редагування та оновлення даних.
4. Можливість побудови системи керівництва з різними правами доступу.
5. Автоматизація процесів верифікації та модерації (перевірка та модерація внесених даних, щоб уникнути небажаних або шкідливих матеріалів).
6. Можливість захисту даних та перегляду історії змін.
7. Можливість наявності системи повідомлень і сповіщень для керівництва про важливі події та оновлення на сайті.

Функціональні вимоги:

1. Реєстрація та авторизація. Можливість користувачам створювати облікові записи, увійти на сайт та керувати своїм профілем.
2. Пошук і фільтрація.
3. Публікація контенту. Можливість додавати, редагувати і видаляти різний контент, такий як статті, відео, фотографії тощо.
4. Коментарі й обговорення.
5. Адміністративний інтерфейс.
6. Звітність.
7. Захист і безпека.
8. Соціальна інтеграція: Можливість обмінюватися контентом або взаємодіяти з іншими соціальними мережами.

Нефункціональні вимоги:

1. Безпека та надійність:
 - Сайт повинен бути доступним 24/7, крім запланованих перерв на технічне обслуговування.
 - Відновлення роботи після можливих збоїв або відмов повинно відбуватися автоматично.
 - Доступність - час, що витрачається на обслуговування системи не повинно перевищувати 3% від загального часу роботи.
 - Середній час безвідмовної роботи - 10 робочих днів.
 - Повинні бути встановлені механізми автентифікації та авторизації користувачів.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						12
Змн.	Арк.	№ докум.	Підпис	Дата		

- Максимальна норма помилок або дефектів - 1 помилка на десять тисяч рядків коду.

2. Зручність використання.

3. Забезпечення конфіденційності.

Інформація користувачів має бути збережена відповідно до законодавства щодо конфіденційності та захисту даних.

4. Продуктивність та застосовність.

- Сайт повинен завантажувати сторінки швидко, з меншими часами очікування.
- Час завантаження сторінки повинен бути менше ніж 5 секунд для забезпечення задоволення користувачів.
- Сайт повинен бути здатний обробляти одночасних запитів від 10 користувачів без втрати продуктивності.
- Час, необхідний для навчання звичайних користувачів - 3 робочих дні (24 години), для навчання просунутих користувачів - 1 робочий день (8 годин).
- Час відгуку для типових задач - не більше 10 секунд, для складних завдань - не більше 25 секунд.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Аналіз аналогів програмного продукту

Вебсайти ветеринарних клінік відображають різноманітність підходів та стратегій у сфері ветеринарної медицини. Кожен з них відіграє важливу роль у сприянні доступності та якості ветеринарних послуг для власників тварин. У даному аналізі ми зосередимося на трьох сайтах аналогів. Кожен з цих сайтів має свої унікальні характеристики та ключові аспекти, які слід ретельно розглянути для вибору оптимальних інструментів, практик та можливостей для покращення нашого власного вебпроекту ветеринарної клініки залежно від конкретних потреб та поставлених вимог.

Назва	АртВет	vet-ua	Авет
Посилання	https://artvet.kiev.ua/	http://vet-ua.com/	https://avet.kiev.ua/
Дизайн та візуальне враження	Сучасний, естетичний, привабливий	Застарілий, бажає кращого	Сучасний, чистий, привабливий
Наявність інформації про послуги та лікарів	Інформативний, відображає всі послуги та лікарів	Обмежений обсяг інформації	Обмежений обсяг інформації
Можливість запису на прийом онлайн	Присутній, зручний та швидкий	Присутній, зручний та швидкий	Присутній, не зручний
Розділи та структура сайту	Добре організовані, легка навігація	Добре організовані	Добре організовані
Наявність блогу або новин	Присутній, активно оновлюється	Присутній, не активний	Присутній, активно оновлюється
Мобільна версія сайту	Присутня, адаптована	Присутня, погано адаптована	Присутня, але може бути кращою
Форма зворотного зв'язку	Присутня	Відсутня	Присутня

Цей аналіз ставить за мету визначити сильні та слабкі сторони кожного вебсайту ветеринарної клініки і вибрати найбільш ефективні підходи для вдосконалення власного вебресурсу. Загалом, сайт АртВет вирізняється якісним дизайном, багатою інформацією та функціоналом, в той час, як vet-ua потребує поліпшення в аспектах дизайну та інформаційного наповнення. Сайт Авет також

є привабливим, але може покращити зручність онлайн-запису та адаптивність для мобільних пристроїв.

Враховуючи інформацію отриману при дослідженні було прийнято рішення для покращення власного вебсайту ветеринарного центру можна використати кращі практики, виявлені на сайті АртВет, такі як сучасний дизайн, інформативність, зручність онлайн-запису та мобільна адаптивність. Також варто враховувати рекомендації щодо поліпшення блогу або розділу новин для підтримки активного оновлення контенту. Таким чином, даний додаток має бути дуже простим у використанні, з інтуїтивним дизайном; буде підтримувати будь-які різноманітні категорії та послуги, що дозволить задовольнити вимоги користувачів.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

1.4 Вибір архітектури веборієнтованої системи ветеринарної клініки

Для стабільної та продуктивної роботи інформаційної системи протягом тривалого періоду необхідно, щоб вона відповідала всім вимогам, мала високу якість та була досить простою у використанні для звичайних користувачів. Ці аспекти прямо залежать від архітектури програмного забезпечення.

Для створення подібних систем варто використовувати клієнт-серверну архітектуру програмного забезпечення з використанням перевірених патернів програмування. Розглянемо ключові концепції, на які варто звернути увагу та надалі використати у даній роботі.

Система буде будуватися з трьох взаємопов'язаних частин: контролери, моделі та представлення. За обробку дій, пов'язаних з користувачем та його діями буде відповідати клієнтська частина системи, яка не тільки взаємодіє з клієнтом та відображає інформацію у зручному вигляді, а й формує запит та надсилає його до серверної частини, а надалі відповідь сервера оброблюється та надається користувачу. Отже, основною метою даного шаблону є гнучка система програмного забезпечення для швидкого та легкого керування над усіма компонентами.

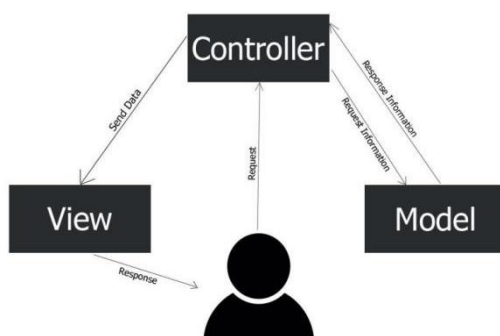


Рисунок.1.1. Діаграма взаємодії між компонентами

Також при розробці діаграми класів планується першочергове включення та використання патерну прототип, який підходить для реалізації сценаріїв з ролями.

1.5 Обґрунтування вибору засобів реалізації та вимоги до апаратного забезпечення

Ключові вимоги:

- Програма має забезпечувати виконання вимог користувачів.
- Можливість легко та доступно користуватися сайтом.
- Можливість користувацької реєстрації і доступу до власного профілю.
- Можливість використання системи пошуку та категоризації.
- Можливість підтримки різних мов.
- Можливість додавання до сайту власних коментарів та обговорень.
- Можливість захистити власну приватність даних та гарантування їх безпеки.
- Можливість отримання зворотного зв'язку з підтримкою сайту.
- Програма має забезпечувати виконання вимог персоналу вебдодатку (власники сайту):
- Зручний інтерфейс для внесення даних.
- Підтримка різних типів контенту. Можливість додавання різних типів елементів наповнення, таких як текст, зображення, відео, аудіо та інші мультимедійні елементи.
- Можливість редагування та оновлення даних.
- Можливість побудови системи керівництва з різними правами доступу.
- Автоматизація процесів верифікації та модерації (перевірка та модерація внесених даних, щоб уникнути небажаних або шкідливих матеріалів).
- Можливість захисту даних та перегляду історії змін.
- Можливість наявності системи повідомлень і сповіщень для керівництва про важливі події та оновлення на сайті.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						17
Змн.	Арк.	№ докум.	Підпис	Дата		

У разі виконання некоректних дій передбачити підказки та пояснення причин через яку виникла помилка.

2. Часові характеристики і розмір пам'яті, необхідної для роботи програми

Час реакції програми на дії користувача (маніпуляції з пристроями введення даних) не повинен перевищувати 0,25 с. Час виконання команд меню не більше 1 с. Відображення масивів даних за запитами не більше 3 хвилин.

Доступність БД – 90% цілодобово. Операції з'єднання з БД не більше 1 хвилини.

Обсяг оперативної пам'яті, необхідний для роботи програми не менше 1Гб. Дисковий простір, необхідний для збереження програми і файлів даних не більше 300 Мб для робочої станції та 20 ГБайт.

Інсталяційний пакет програми, що містить у складі БД не повинні перевищувати 100 Мбайт.

3. Вимоги до надійності

3.1. Вимоги до надійного функціонування

Програма має нормально функціонувати при безперебійній роботі ПК. Доступність БД 90% при одночасному доступі до 20 користувачів. При апаратних збоях, відновлення нормальної роботи програми повинно виконуватись після:

а) апаратні збої сервера – перезавантаження ОС сервера, запуск сервера БД (запуск резервного сервера, використання технологій RAID для збереження даних);

б) апаратні збої робочої станції – перезавантаження ОС ПК, запуск виконуваного файлу програми.

У системі має бути реалізована коректна обробка виняткових ситуацій.

3.2. Контроль вхідної і вихідної інформації

Для контролю коректності вхідної інформації та захисту від помилок оператора:

- перевірка відповідності даних доменам інформаційних атрибутів;

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- використанням механізму авто заповнення та вибору за переліком для зв'язаних даних;

- захист від помилок оператора (залипань, випадкових символів тощо).

Визначені некоректні дії повинні супроводжуватись повідомленнями про помилку і блокуванням операцій оновлення даних. В системі має бути передбачений захист від загального блокування. Для вихідної інформації передбачити:

- відсутність блокування даних через багато користувацький доступ;
- постійне оновлення даних у відображених на екрані звітних формах.
- повідомлення про використання режиму монопольного доступу супервайзерами БД.

3.3. Час відновлення після відмови

Час відновлення після відмови, не пов'язаною з роботою програми, повинен складатися із: часу перезапуску операційної системи; часу запуску сервера БД (підключення до сервера) запуску виконуваного файлу, часу повторного введення або зчитування даних.

3.4. Умови експлуатації й збереження

Регламенті операції проводити за графіком:

- створення резервних копій даних: 1 раз на тиждень;
- збереження резервної копії журналу транзакцій – 1 раз на тиждень;
- архівування даних – 1 раз на місяць; обслуговуючі функції (прибирання “сміття”, де фрагментація тощо) - щоденне;
- оновлення системи ідентифікації та аутентифікації користувачів – 1 раз на місяць та за потребою.

Збереження архівних копій - на визначених носіях. Збереження резервних копій – на визначеному диску без перезапису. Знищення копій за регламентом.

3.5. Вимоги до інформаційних структур на вході і виході

Формат відображення даних має дозволяти імпорт даних в додатки MS Office для редагування та перегляду.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

3.6. Вимоги до методів рішення і мов програмування

Вибір методів рішення здійснюється розробниками без узгодження з замовником. СУБД обирається у відповідності до характеристик визначених в п. 3. З замовником погоджується вибір варіанту за вартісною ознакою.

3.7. Вимоги до системи програмних засобів

Вимоги до програмного забезпечення сервера: ОС – серверна версія ОС Windows не нижче Windows 2000 Server, IIS (або інший сервер), СУБД визначається встановлюється та налаштовується розробником (ліцензування СУБД виконується замовником). Вимоги до програмного забезпечення робочої станції: ОС - родина Windows не нижче Windows XP. Драйвери периферійних пристроїв - введення/виводу визначаються та встановлюються при встановленні ОС в залежності від конфігурації робочої станції. Пакет додатків MS Office. Робоча станція є клієнтом мереж Microsoft.

3.8. Вимоги до складу і параметрів технічних засобів

Вимоги до складу технічних засобів:

1. Сервер:

- сервер у базовій конфігурації із підтримкою RAID;
- з'ємний запам'ятовуючий пристрій для архівування даних;
- принтер для друку;
- засоби для під'єднання до локальної мережі.

2. Робоча станція:

- ПК на базі процесорів Intel, AMD у стандартній комплектації;
- периферійні пристрої друку;
- засоби для під'єднання до локальної мережі;
- Pentium III з тактовою частотою 750 МГц;
- оперативна пам'ять - 128 Мбайт;
- об'єм дискової пам'яті - 20 Гбайт.

4. Вимоги до програмної документації

1. Програмна документація повинна включати наступні відомості:

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						20
Змн.	Арк.	№ докум.	Підпис	Дата		

"Інструкція по інсталяції (встановленню) програми", складається з опису інсталяційного пакета, переліку етапів інсталяції та їх послідовності, опису встановлених програмних компонентів та режимів їх роботи після інсталяції.

2. Керівництво користувача" складається з опису послідовності завантаження програми, основних режимів роботи, опису основних екранних форм, переліку виняткових ситуацій та реакції користувача на них, порядку виконання завдань в системі. Під час оформлення пояснювальної записки до курсової роботи дані відомості містяться в 3 розділі курсової роботи.

3. "Керівництво адміністратора баз даних", складається з опису складу таблиць бази даних та правил доступів до них, опису та послідовності робіт по обслуговуванню бази даних (архівування, резервне копіювання, з вказівкою періодичності виконання та засобів, що для цього використовуються), порядку налаштування серверу та клієнтських додатків.

4. Середовище програмування PHPStorm є відмінним вибором для етапу кодування, оскільки воно надає широкі можливості для розробки веб-додатків на мові програмування PHP [15]. Його інтегровані інструменти, підтримка PHP, HTML, CSS, JavaScript та інших технологій веб-розробки дозволяють ефективно створювати та налагоджувати програми. Середовище являє собою інтелектуальний редактор з можливостями швидкого аналізу коду, запобіганню помилок на початковому етапі коду і автоматизованими засобами рефакторинга. Також було обрано OpenServer. Основні особливості OpenServer включають:

- інтуїтивний інтерфейс дозволяє швидко встановити та налаштувати сервер;
- підтримка різних технологій(зокрема OpenServer підтримує PHP, MySQL, Apache та інші технології, що часто використовуються у веброботці);
- модульність (можливість встановлення та використання різних модулів та компонентів в залежності від потреб проекту);
- локальний доступ (робота над проектами локально, не підключаючись до зовнішнього сервера, що сприяє збереженню приватності та швидкістю розробки).

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						21
Змн.	Арк.	№ докум.	Підпис	Дата		

В загальному, OpenServer – це зручний інструмент для розробки веб-додатків та швидкого налаштування.

База даних відіграє важливу роль для кожного сучасного вебдодатку. Завдяки динамічній природі вебзастосунків зараз, навіть найпростіші програми вимагають деяких механізмів зберігання, доступу та зміни даних. СУБД — це програмні та мовні засоби, які виступають посередником між БД та її численними користувачами. Завдяки сукупності мовних та програмних засобів, СУБД сприяють створенню, веденню та спільного використання БД різними користувачами. Сучасна програма СУБД складаються з ядра, процесору мови БД, підсистеми підтримки часу виконання та сервісних програм, які надають додаткові можливості для обслуговування інформаційних систем. На даний момент існує багато систем управління даними з різним функціоналом. Правильний вибір СУБД є одним з важливих етапів при розробці сайту чи додатку баз даних, а також він має задовольняти потреби як теперішні, так і в майбутньому приносячи вигоду. Важливо також звертати увагу яку структуру (реляційну чи нереляційну) містить обрана СУБД, адже вони мають ряд своїх переваг та особливостей. Тому для виконання поставлених завдань було обрано використовувати в даному проєкті MySQL [10].

Висновки з першого розділу

Проведено аналіз предметної області за допомогою якого були визначені основні вимоги до платформи, яка спрямована на створення зручного середовища для клієнтів та сприяння взаємодії між клінікою та її основними клієнтами. В процесі дослідження аналогічних і схожих за концепцією продуктів були проаналізовані та порівняні різні платформи та їх принципи роботи. На основі відповідних вимог та критеріїв був сформований набір основних технологій і засобів реалізації для розробки вебсайту.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2 ПРОЕКТУВАННЯ САЙТУ ВЕТЕРИНАРНОЇ КЛІНІКИ

2.1. Визначення варіантів використання та об'єктно-орієнтованої структури системи

Для наочної взаємодії між елементами додатку було створено діаграму варіантів використання (див. рис. 2.1). Вона відображає основні функціональні можливості сайту та взаємодію користувача з системою. Основні актори на діаграмі - це користувач сайту та адміністратор. Вони взаємодіють з системою через різні випадки використання.

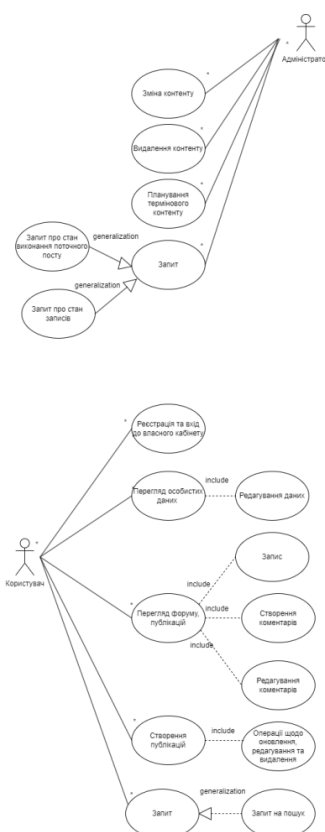


Рисунок 2.1 – Діаграма варіантів використання

Опис діаграми варіантів використання наведено в таблицях.

Таблиця 2.1

Опис прецеденту «Зміна контенту»

Назва варіанту використання	Зміна контенту
Актори	Адміністратор
Короткий опис	Адміністратор може відкоригувати та вносити правки у інформацію в різні блоки на сайті.

Таблиця 2.2

Опис прецеденту «Видалення контенту»

Назва варіанту використання	Видалення контенту
Актори	Адміністратор
Короткий опис	Адміністратор може видаляти контент з сайту.

Таблиця 2.3

Опис прецеденту «Планування термінового контенту»

Назва варіанту використання	Планування термінового контенту
Актори	Адміністратор
Короткий опис	Адміністратор розміщує новий контент в необхідний час.

Таблиця 2.4

Опис прецеденту «Запит»

Назва варіанту використання	Запит
Актори	Адміністратор
Короткий опис	Адміністратор має можливість переглядати наявні запити та записи.

Таблиця 2.5

Опис прецеденту «Реєстрація та вхід»

Назва варіанту використання	Реєстрація та вхід
Актори	Адміністратор, користувач
Короткий опис	Користувачі можуть зареєструватися та авторизуватись на сайті, також є можливість авторизації для адміністраторів, щоб керувати та переглядати активності на сайті.

Таблиця 2.6

Опис прецеденту «Перегляд особистих даних»

Назва варіанту використання	Перегляд особистих даних
-----------------------------	--------------------------

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						24
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 2.6

Актори	Адміністратор, авторизований користувач
Короткий опис	Користувачі сайту мають змогу переглядати власні дані в профілі та вносити зміни до них.

Таблиця 2.7

Опис прецеденту «Перегляд форму, публікацій»

Назва варіанту використання	Перегляд форму, публікацій
Актори	Користувач
Короткий опис	Користувачі мають можливість переглядати контент сайту, створювати запис на курс та консультацію.

Таблиця 2.8

Опис прецеденту «Створення публікацій»

Назва варіанту використання	Створення публікацій
Актори	Авторизований користувач
Короткий опис	Користувачі мають можливість створювати власні відгуки на сайті та керувати ними.

Таблиця 2.9

Опис прецеденту «Запит»

Назва варіанту використання	Запит
Актори	Користувач
Короткий опис	Користувачі можуть здійснити пошук необхідної інформації за допомогою функціоналу на сайті.

Попередній аналіз функціональних вимог дозволив виділити наступні сутності, які забезпечать програмну реалізацію додатка. Наведено узагальнену діаграму об'єктно-орієнтованої структури системи, яка ілюструє взаємозв'язки та взаємодію між різними об'єктами системи. Кожна сутність представляє певний компонент або модуль програмного продукту, з власними атрибутами та

методами. Діаграма слугує візуальним представленням архітектури системи, сприяючи кращому розумінню її загальної структури.

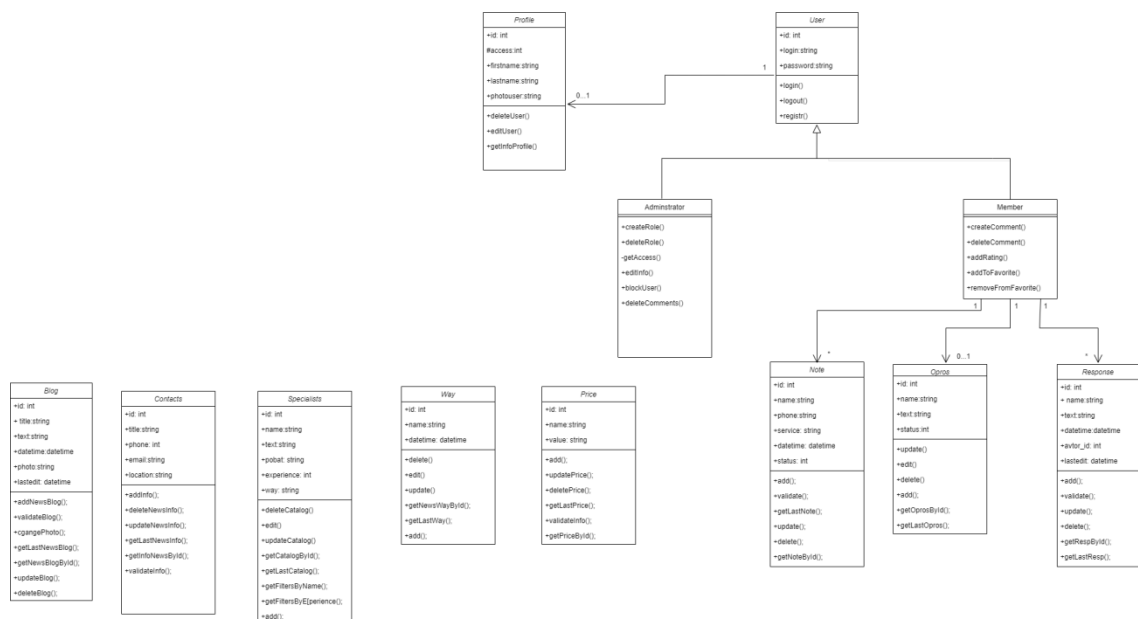


Рисунок 2.2 – Діаграма класів

Внаслідок проєктування діаграми до веб-орієнтованої системи сайту ветеринарної клініки було включено такі класи:

- User – клас для реєстрації та авторизації користувача;
- Profile – клас для процесу перегляду особистих даних;
- Administrator – клас для дій адміністратора сайту;
- Member – клас для дій авторизованих користувачів;
- Contacts – клас для перегляду актуального місцезнаходження та контактних даних клініки сайту;
- Specialists – клас для процесу перегляду спеціалістів клініки сайту;
- Way – клас для процесу перегляду актуальних послуг клініки;
- Note – клас для процесу запису на консультації в клініці;
- Opros – клас для процесу запису на курси клініки;
- Price – клас для перегляду послуг та цін на сайті;
- Blog – клас для процесу перегляду останніх нових клініки сайту;
- Response – клас для процесу залишення відгуків користувачами.

В діаграмі можна виділити класи, які спроектовано, використовуючи

шаблон “Прототип”. Це породжувальний патерн, який дає змогу копіювати об’єкти не вдаючись у подробиці самої реалізації. Він підходить для реалізації сценаріїв з ролями. Прототип – клас User, а класи ролей його клонують – Admin з включення власного функціонала для реалізації поставлених завдань [1].

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						27
Змн.	Арк.	№ докум.	Підпис	Дата		

2.2. Розробка бази даних системи

Для збереження файлів та інформації у додатку використовуються MySQL, це система управління базами даних, яка базується на реляційної моделі.

База даних системи складається з дев'яти таблиць:

- Users – таблиця, яка зберігає дані користувачів;
- Way – таблиця, яка зберігає дані про напрямки клініки;
- Specialists – таблиця, яка зберігає дані про спеціалістів клініки;
- Response – таблиця, яка зберігає дані про відгуки клієнтів;
- Price – таблиця, яка зберігає дані про ціну послуг;
- Opros – таблиця, яка зберігає дані про записи на консультацію;
- Note – таблиця, яка зберігає дані про записи на курси;
- Contacts – таблиця, яка зберігає дані про контактні дані клініки;
- Blog – таблиця, яка зберігає дані про новини на сайті.

Опис кожної з таблиць бази даних наведений у таблицях.

Таблиця 2.10

Опис таблиці Users

Поле	Тип	Призначення
id	int	Ідентифікатор
login	varchar	Логін користувача
password	varchar	Пароль користувача
access	int	Доступ користувача
lastname	varchar	Прізвище
firstname	varchar	Ім'я
photo_user	varchar	Фото користувача

Таблиця 2.11

Опис таблиці Way

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Назва послуги
datetime	datetime	Дата та час

Таблиця 2.12

Опис таблиці Specialists

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Ім'я
surname	varchar	Прізвище
pobat	varchar	По батькові
experience	int	Стаж
way	varchar	Спеціалізація

Таблиця 2.13

Опис таблиці Response

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Назва
text	varchar	Текст посту
datetime	datetime	Дата та час
avtor_id	int	Автор посту

Таблиця 2.14

Опис таблиці Price

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Назва послуги
value	varchar	Значення
datetime	datetime	Дата та час

Таблиця 2.15

Опис таблиці Opros

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Ім'я

Продовження таблиці 2.15

text	varchar	Електронна пошта
datetime	datetime	Дата та час
status	int	Статус обробки

Таблиця 2.16

Опис таблиці Note

Поле	Тип	Призначення
id	int	Ідентифікатор
name	varchar	Ім'я
phone	varchar	Номер телефону
datetime	datetime	Дата та час
status	int	Статус обробки
service	varchar	Назва послуги

Таблиця 2.17

Опис таблиці Contacts

Поле	Тип	Призначення
id	int	Ідентифікатор
title	varchar	Назва
phone	int	Номер телефону
email	varchar	Електронна пошта
location	varchar	Місцезнаходження

Таблиця 2.18

Опис таблиці Blog

Поле	Тип	Призначення
id	int	Ідентифікатор
title	varchar	Заголовок
text	text	Тест посту
datetime	datetime	Дата та час
photo	varchar	Фото

lastedit	datetime	Дата та час внесення змін
----------	----------	---------------------------

2.3. Проектування та реалізація алгоритмів роботи системи

Додаток має інтуїтивний і легкий у використанні інтерфейс, що включає всі основні функції платформи, яка у свою чергу дозволяє користувачам зручно та просто взаємодіяти з нею. Було створено декілька діаграм активності для візуалізації та представлення потоку діяльності в системі (див. рис. 2.2 та рис. 2.3). Вони пропонують чіткий спосіб зображення послідовних і паралельних дій, які відбуваються під час виконання системних процесів.

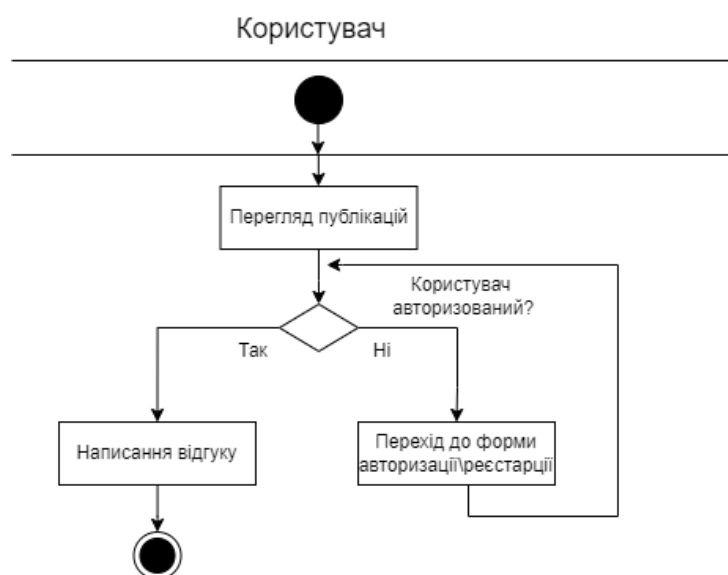


Рисунок 2.2 – Діаграма активності процесу написання відгуку

Початковий стан: Користувач у момент перегляду публікації та початку етапу написання відгуку.

Дія: Перегляд публікації та написання відгуку.

Стан прийняття рішення: користувач сайту є авторизованим?

- Так:
 - Дія: Написання відгуку та публікація матеріалу на сайт.
 - Кінцевий стан.
- Ні:
 - Дія: Перехід до форми авторизації або реєстрації для входу на сайт.
 - Кінцевий стан.

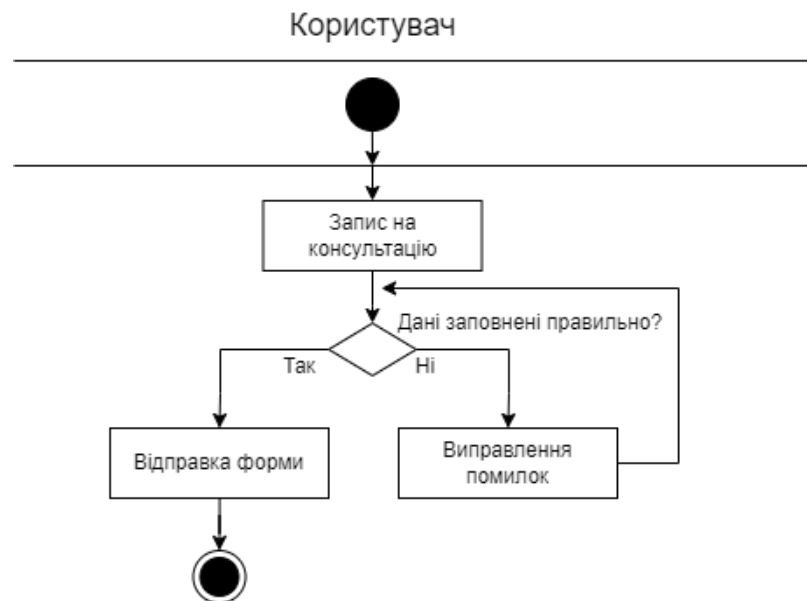


Рисунок 2.3 – Діаграма активності процесу запису на консультацію
 Початковий стан: Користувач у момент запису на консультацію.
 Дія: Запис на консультацію.
 Стан прийняття рішення: дані заповнені правильно?

- Так:
 - Дія: Відправлення форми та запис на консультацію.
 - Кінцевий стан.
- Ні:
 - Дія: Повторний перехід до форми з уточненням правильності її заповнення.
 - Кінцевий стан.

Діаграми послідовності є важливим інструментом під час моделювання в області програмного забезпечення, спрямованим на візуалізацію послідовності виконання різноманітних операцій та взаємодії об'єктів у системі. Вони дозволяють аналізувати та документувати стани між різними компонентами програми чи системи [7]. Одна діаграма послідовності описує один із варіантів використання, тому на рисунку (див. рис. 2.4) побудовано діаграму послідовності для демонстрації виконання процесу написання відгуку.

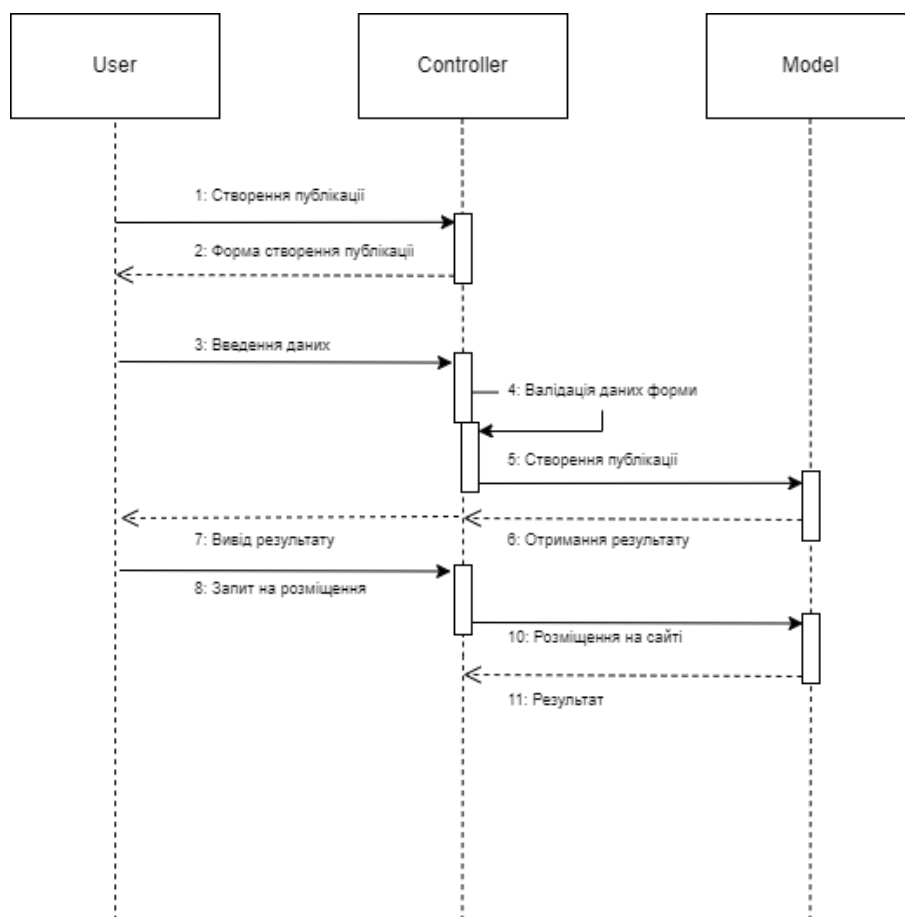


Рисунок 2.4 – Діаграма послідовності

Для відображення взаємодії між різними елементами композитної структури було побудовано діаграму кооперації. На діаграмі показано об'єкти та їх повідомлення, якими вони обмінюються між собою в рамках визначеного варіанту використання заданого процесу (див. рис. 2.5).

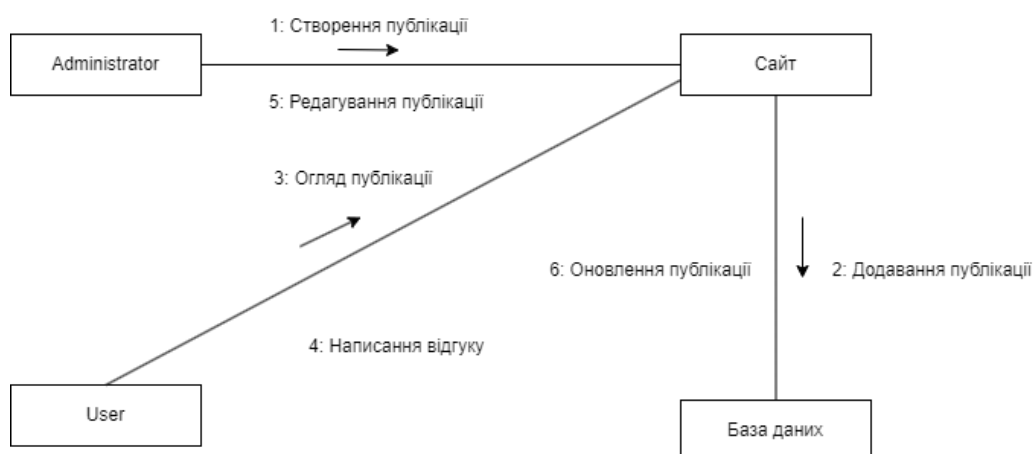


Рисунок 2.5. – Діаграма кооперації написання відгуку

Загальний огляд послідовності процесу дій діаграми:

1. Адміністратор створює публікацію: Адміністратор ініціює створення нової публікації, передає її на сайт.
2. Сайт зберігає публікацію: Сайт зберігає отримані дані публікації в базі даних.
3. Користувач переглядає публікації: Користувач ініціює запит до сайту для перегляду публікацій.
4. Користувач залишає відгук: Користувач пише та надсилає відгук через сайт.
5. Адміністратор редагує публікацію: Адміністратор вносить зміни до наявної публікації, передає зміни на сайт.
6. Сайт зберігає зміни: Сайт зберігає оновлені дані публікації в базі даних.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						35
Змн.	Арк.	№ докум.	Підпис	Дата		

2.4. Реалізація функціонала системи сайту ветеринарної клініки

Маючи загальний алгоритм веб-додатку, який був отриманий з попередніх підрозділів, можна почати безпосередню роботу над проектом.

Для прикладу взято роботу над модулем “Користувач” в якому було створено контролер Users, який керує всією роботою у цьому модулі (під назвою Users), забезпечує перехід між сторінками, які пов'язані у даному розділі, дозволяє додавати та редагувати наявну інформацію публікацій, що пов'язаний з користувачем та його діями безпосередньо на сайті.

Для реєстрації використано форму, яка забезпечує початкову перевірку та відправлення інформації до вказаної бази даних:

```
public function Validate($formRow)
{
    $errors = [];
    if (empty($formRow['login']))
        $errors[] = 'Поле "Логін" не може бути порожнім!';
    $user = $this->GetUserByLogin($formRow['login']);
    if (!empty($user))
        $errors[] = 'Користувач із вказаним логіном вже зареєстрований!';
    if (empty($formRow['password']))
        $errors[] = 'Поле "Пароль" не може бути порожнім!';
    if ($formRow['password'] != $formRow['password2'])
        $errors[] = 'Паролі не співпадають!';
    if (strlen($formRow['password']) <= 3)
        $errors[] = 'Поле "Пароль" має містити більше 3-ьох символів!';
    if (empty($formRow['firstname']))
        $errors[] = 'Поле "Прізвище" не може бути порожнім!';
    if (empty($formRow['lastname']))
        $errors[] = 'Поле "Ім'я" не може бути порожнім!';
    if (count($errors) > 0)
        return $errors;
    else
        return true;
}
```

Також відбувається шифрування пароля користувача і проходить перевірка наявності фотографії, відповідно до якої надалі буде виведено певне фото профілю, а також відфільтровано введені поля тільки для тих значень, що потрібні в базі даних.

```
public function AddUser($userRow)
{
    $valRez = ($this->Validate($userRow));
    if (is_array($valRez)) {
        $rez = [
            'error' => true,
            'messages' => $valRez
        ];
        return $rez;
    }
    $fields = ['login', 'password', 'lastname', 'firstname'];
    $userRowFilter = Utils::ArrFilter($userRow, $fields);
    $userRowFilter['photo user'] = '...photo';
    $userRowFilter['password'] = md5($userRowFilter['password']);
}
```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        $id = \Core\core::getInstance()->getDB()->insert('users', $userRowFilter);
        return [
            'error' => false,
            'id' => $id
        ];
    }
}

```

При авторизації відбувається перевірка введених даних користувачем з інформацією з бази даних по відповідним полям:

```

    public function AuthUser($login, $password)
    {
        $password = md5($password);
        $users = \Core\core::getInstance()->getDB()->select('users', '*', [
            'login' => $login,
            'password' => $password
        ]);
        if (count($users) == 1) {
            $user = $users[0];
            return $user;
        } else {
            return false;
        }
    }
}

```

При повторній авторизації виведеться повідомлення, що користувач вже на сайті. Кнопки «Авторизація» та «Вхід» приховуються, натомість з'являються пункти «Профіль» та «Вийти» та фото користувача. Також реалізовано пункт меню власний профіль, в якому користувач має змогу переглянути, видалити профіль та редагувати дані й можливість виходу з профілю:

```

    function actionProfile()
    {
        $title = 'Профіль';
        $id = $_GET['id'];
        $user = $this->userModel->getUserById($id);
        if (isset($_SESSION['user']))
            return $this->render('profile', ['model' => $user], [
                'MainTitle' => $title
            ]);
    }
}

```

При кожній операції виводиться відповідне сповіщення, яке показує стан виконаної операції (успіх або помилку з її уточненням).

```

    function actionLogin()
    {
        $title = 'Вхід';
        $title2 = 'Вхід на сайт';
        if (isset($_SESSION['user']))
            return $this->renderMessage('success', 'Ви вже увійшли на сайт!', null, [
                'MainTitle' => $title,
                'PageTitle' => $title2
            ]);

        if ($this->isPost()) {
            $user = $this->userModel->AuthUser($_POST['login'], $_POST['password']);
            if (!empty($user)) {
                $_SESSION['user'] = $user;

                return $this->renderMessage('success', 'Успішний вхід на сайт!', null, [
                    'MainTitle' => $title,
                    'PageTitle' => $title2
                ]);
            }
        }
    }
}

```

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    } else {
        return $this->render('login', null, [
            'MainTitle' => $title,
            'PageTitle' => $title2,
            'MessageText' => 'Не коректно заповнено поле "Логін" чи "Пароль"!',
            'MessageClass' => 'danger'
        ]);
    }
} else {
    $params = [
        'MainTitle' => $title,
        'PageTitle' => $title2
    ];
    return $this->render('login', null, $params);
}
}

```

Варто зазначити, що в даній системі наявний поділ доступу до деяких пунктів меню на звичайного користувача, авторизованого користувача та адміністратора. А доступ до маніпуляцій з інформацією до певного блоку меню мають лише авторизовані користувачі.

```

    $userModel = new \Models\Users();
    $user = $userModel->getCurrentUser();
    if ($user == null)
    {
        $rez = [
            'error'=> true,
            'messages'=> ['Користувач не аутентифікований!']
        ];
        return $rez;
    }

```

Для забезпечення роботи над додаванням, редагуванням та видаленням інформації на сайті було створено спеціальні форми, які після проходження валідації, підтвердженням прав доступу та id користувача дають змогу проводити різні операції над вмістом сайту певного розділу:

```

    public function Validate($newsRow)
    {
        $errors = [];
        if (empty($newsRow['text']))
            $errors[] = 'Поле "Текст" не може бути порожнім!';
        if (empty($newsRow['title']))
            $errors[] = 'Поле "Заголовок" не може бути порожнім!';
        if (count($errors) > 0)
            return $errors;
        else
            return true;
    }

```

Приклад коду для видалення:

```

    public function deleteForum($id)
    {
        $news = $this->getNewsById($id);
        $userModel = new \Models\Users();

        $user = $userModel->getCurrentUser();
        if (empty($news) || empty($user) || $user['access'] != 1)
            if ($user['id'] != $news['user_id'])

```

```

        return false;
    \Core\core::getInstance()->getDB()->delete('forum', ['id' => $id]);
    return true;
}

```

Для зручності було створено окремий клас DB, в який було винесено підключення та основні CRUD операції, які допомагають працювати над даними в базі даних:

```

        protected $pdo;

    public function __construct($server, $login, $password, $database)
    {
        $this->pdo = new \PDO("mysql:host={$server};dbname={$database};charset=UTF8", $login,
        $password);
        $this->pdo->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);
    }

    public function insert($table, $row)
    {
        $fieldsStr = implode(', ', array_keys($row));
        $valueParts = [];
        foreach ($row as $key => $value) {
            $valueParts[] = '?';
        }
        $valueStr = implode(', ', $valueParts);
        $sql = "INSERT INTO {$table} ({$fieldsStr}) VALUES ({$valueStr})";
        $sth = $this->pdo->prepare($sql);
        $sth->execute(array_values($row));
        return $this->pdo->lastInsertId();
    }

```

Висновки з другого розділу

У ході виконання другого розділу було розроблено основні класи для функціонування системи майбутнього сайту та спроектовано базу даних з детальним описом таблиць, також розроблено діаграми активності, послідовності та кооперації для опрацювання можливих взаємозв'язків між компонентами системи сайту відповідно до зазначеної теми. Було описано та проаналізовано варіанти використання програмного продукту відповідно до вимог програмного забезпечення.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3 ІНТЕРФЕЙС ТА ПОРЯДОК РОБОТИ З САЙТОМ ВЕТЕРИНАРНОЇ КЛІНІКИ

3.1 Порядок встановлення та налаштування системи

Веб-орієнтована система сайту ветеринарної клініки – це вебсайт, який створений для можливості доступу до нього через різні браузер. Для використання даного сайту необхідно мати пристрій (телефон, планшет, ноутбук), який відповідає наступним мінімальним вимогам:

- Операційна система Windows 7 і вище, MacOS 10.11 і вище.
- Процесор з тактовою частотою 1,3 – 3,8 ГГц.
- Android 6.0.
- 2 Гб ОЗУ.
- Будь-який браузер.

Таким чином розроблений додаток не вимагає спеціалізованого забезпечення, додаткових налаштувань або засобів розгортання крім стандартних для більшості сучасних сайтів на будь-яких операційних системах. Тому користуватися сайтом буде максимально зручно та легко.

Перейдемо до наступного етапу, а саме до розробки діаграми розгортання, що описує архітектуру програмного комплексу. Діаграма розгортання використовується для розуміння та моделювання взаємодії між програмним та апаратним забезпеченням системи. Вона дозволяє розглядати, як компоненти програмного забезпечення і саме забезпечення взаємодіють один з одним у середовищі. Для виконання поставленого завдання було змодельовано діаграму розгортання, де зображено головні вузли та їх взаємозв'язок. Результат побудованої схеми показано на рисунку 3.1.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						40
Змн.	Арк.	№ докум.	Підпис	Дата		

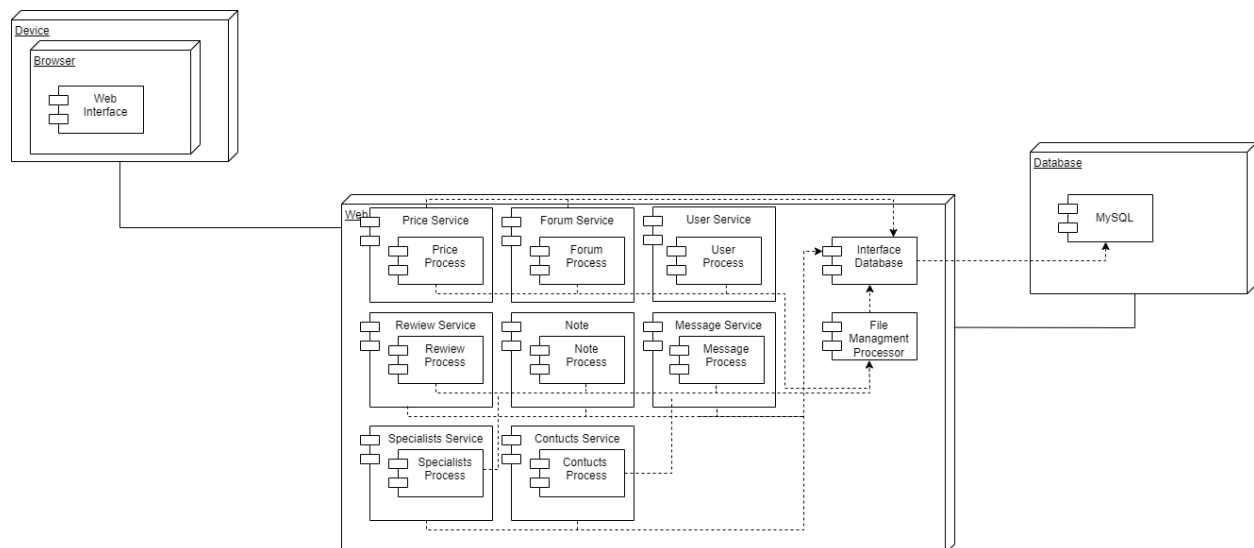


Рисунок 3.1. – Діаграма розгортання

На рисунку представлено вузол користувача (Device), який в свою чергу з'єднується із головним вузлом API (Web). Даний головний вузол спрямовує свою основну роботу для взаємодії з іншими наявними вузлами системи для коректного опрацювання надісланих запитів від користувача та правильного відображення кінцевих результатів.

3.2 Інтерфейс та порядок роботи

Для огляду та робочого процесу розглянемо рисунки на яких буде наведено та продемонстровано основні елементи користувацького інтерфейсу, з якими користувачі будуть мати взаємодію безпосередньо під час роботи. При відкритті сайту користувач має змогу побачити та взаємодіяти з пунктами головного меню, основною сторінкою сайту та кнопками для авторизації чи реєстрації на даному сайті, рисунок 3.2:

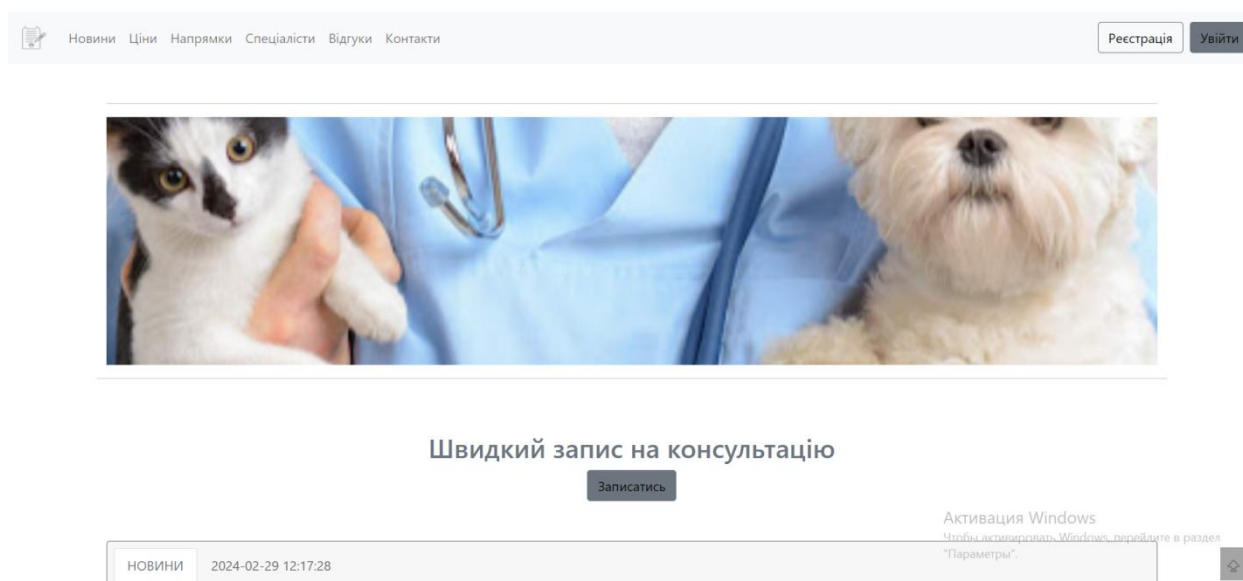


Рисунок 3.2. – Основна сторінка сайту

Після реєстрації та авторизації користувачу відкриваються додаткові можливості на сайті, а саме власний профіль та доступ до написання відгуків.

Рисунок 3.3. – Форма реєстрації

					ІПЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						42
Змн.	Арк.	№ докум.	Підпис	Дата		

Також авторизований користувач сайту може редагувати коментарі, які він написав раніше.

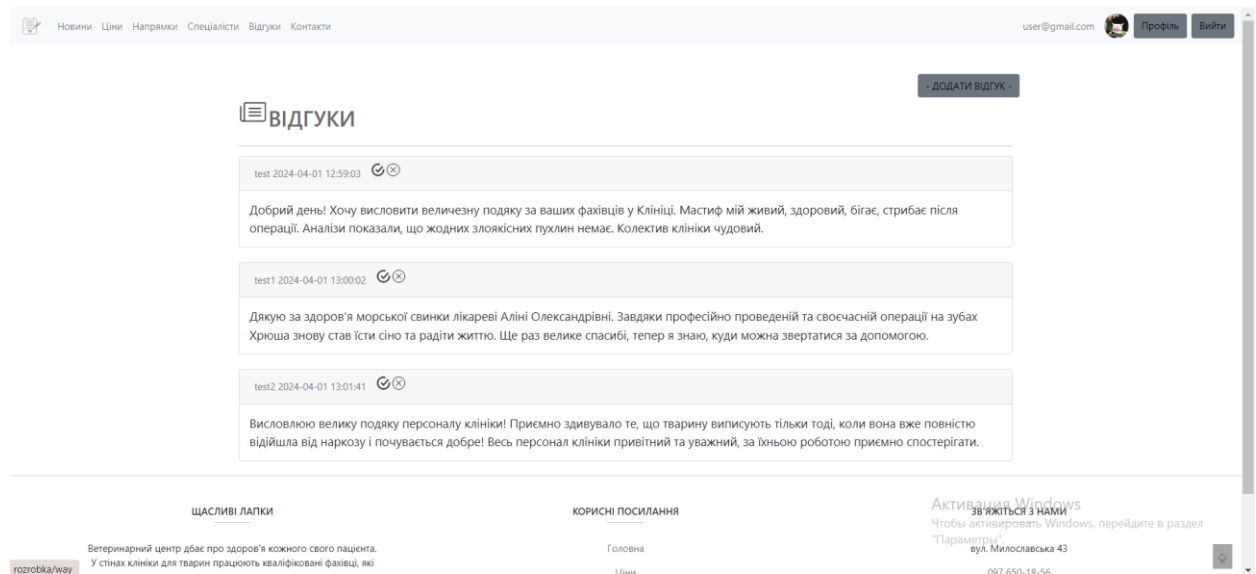


Рисунок 3.4. – Відгуки на сайті

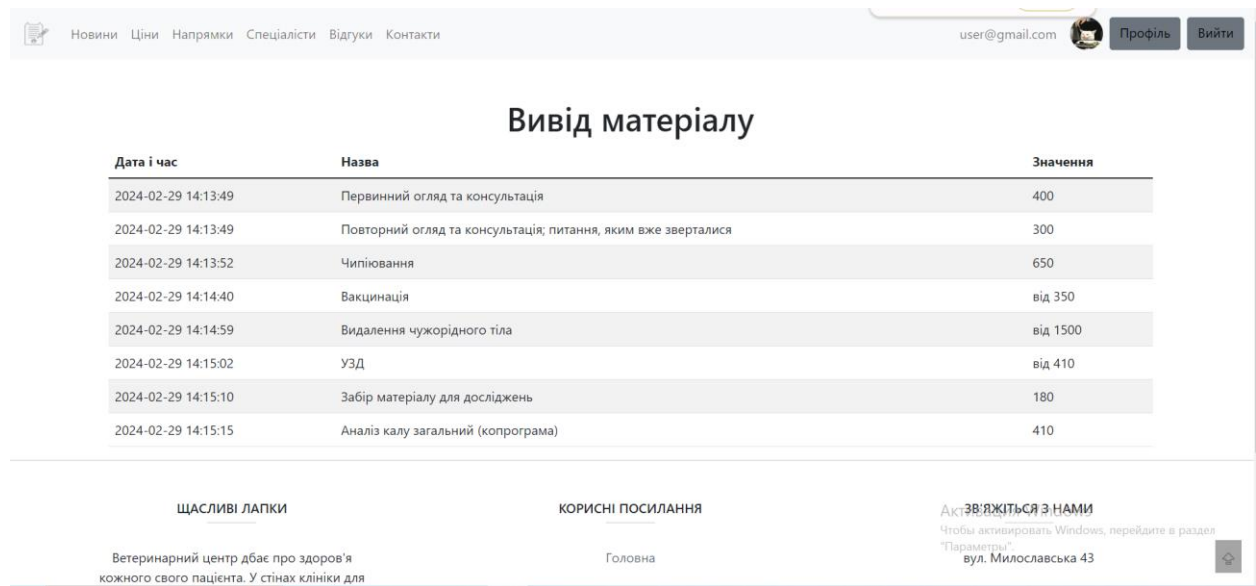


Рисунок 3.5. – Перегляд вкладки ціни

Користувач має змогу швидко та легко записатись на прийом чи консультацію за допомогою форми, вказавши особисті дані.

Новини Ціни Напрямки Спеціалісти Відгуки Контакти user@gmail.com Профіль Вийти

Запис на консультацію

Ім'я:

Номер телефону:

Послуга:

Відправити

ЩАСЛИВІ ЛАПКИ
Ветеринарний центр дбає про здоров'я кожного свого пацієнта. У стінах клініки для тварин працюють кваліфіковані фахівці, які люблять тварин. Це вузькоспеціалізовані ветлікари, які займаються лікуванням, діагностикою, реабілітацією тварин.

КОРИСНІ ПОСИЛАННЯ
Головна
Ціни
Наша команда

ЗВ'ЯЖІТЬСЯ З НАМИ
вул. Милославська 43
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".
з 9:00 до 20:00

Рисунок 3.6. – Форма запису на консультацію

На сайті наведено актуальну інформацію та основні пункти меню для зручного та легкого користування.

Новини Ціни Напрямки Спеціалісти Відгуки Контакти user@gmail.com Профіль Вийти

ЗВ'ЯЖІТЬСЯ З НАМИ

ул. Милославская, 43
Увеличить карту

Контакти клініки "Щасливі Лапки"
Електронна адреса
happy.paws@gmail.co
Номер телефону
976501856
Локація
вул. Милославська 43

ЩАСЛИВІ ЛАПКИ
Ветеринарний центр дбає про здоров'я кожного свого пацієнта. У стінах клініки для тварин працюють кваліфіковані фахівці, які люблять тварин. Це вузькоспеціалізовані ветлікари, які займаються лікуванням, діагностикою, реабілітацією тварин.

КОРИСНІ ПОСИЛАННЯ
Головна
Ціни

ЗВ'ЯЖІТЬСЯ З НАМИ
Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".
97 650-18-56

Рисунок 3.7. – Контакти

При вході на сайт під іменем адміністратора відкривається розширений функціонал. Він може переглядати та вносити зміни до результатів з форм запису на курс та консультацію. Також додатково з'являється пункт меню запис.

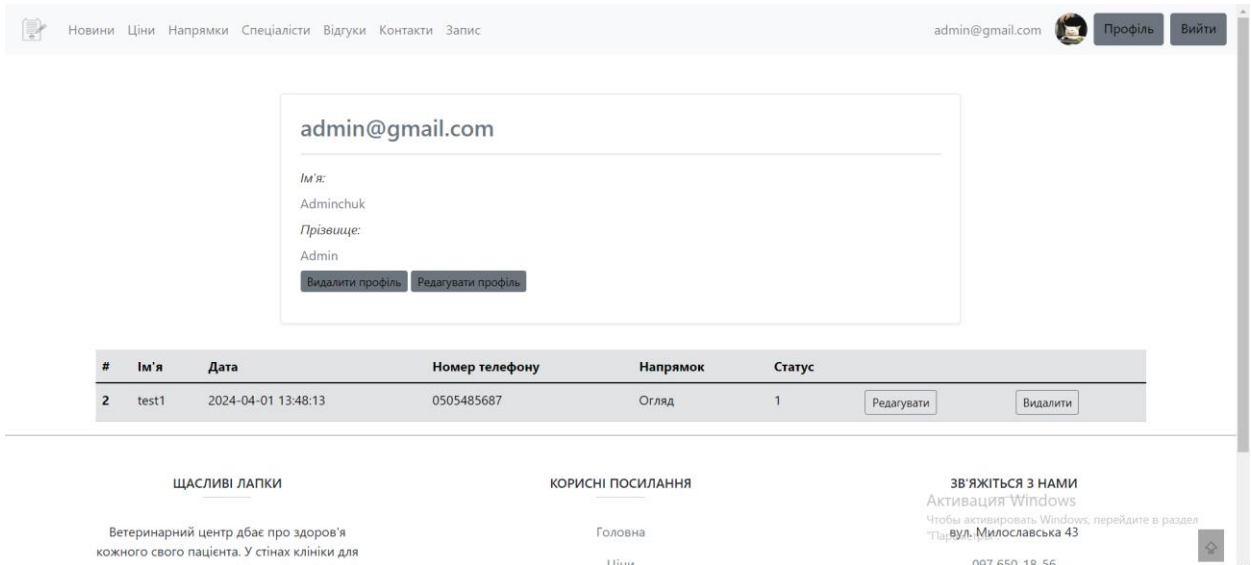


Рисунок 3.8. – Вхід під іменем адміністратора

Адміністратор може видаляти запити написані авторизованими користувачами, а також має змогу вносити нові дані та інформацію по актуальним розділам сайту.

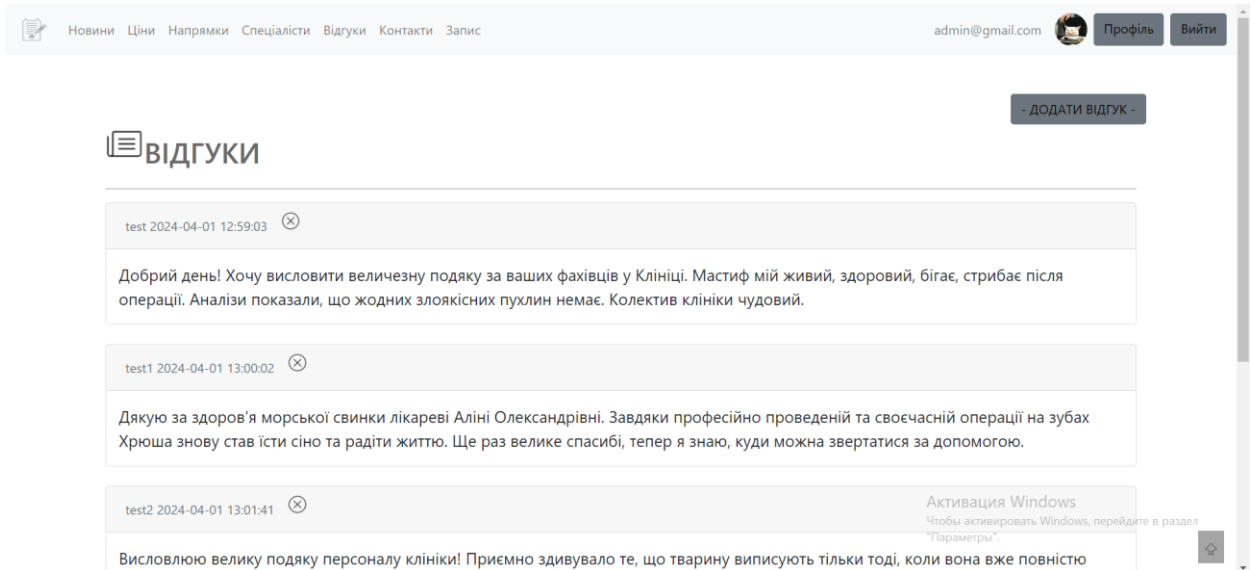


Рисунок 3.9. – Видалення коментарів адміністратором

3.3 Тестування роботи сайту ветеринарної клініки

Тестування роботи сайту ветеринарної клініки може включати декілька етапів, щоб перевірити різні аспекти його функціональності та продуктивності. Важливі аспекти, які потрібно врахувати під час тестування:

1. Функціональність. Всі функції та можливості сайту працюють належним чином. Тестування реєстрації, входу у систему, додавання, редагування та видалення контенту, форми зв'язку та будь-які інші інтерактивні елементи.
2. Сумісність. Перевірка, як сайт відображається на різних вебпереглядачах (Chrome, Firefox, Safari, Edge тощо) та різних пристроях (смартфони, планшети, настільні комп'ютери). Переконайтеся, що він коректно працює на всіх платформах та екранних розмірах.
3. Мобільна адаптивність. Сайт належним чином адаптується до різних мобільних пристроїв та розмірів екранів, забезпечуючи зручний користувацький досвід на смартфонах та планшетах.
4. Тестування форм і валідація даних. Перевірка всіх форм на сайті, включаючи форми зв'язку, реєстрації. Переконайтеся, що дані вводяться коректно, а також проведення тестування валідації даних на стороні клієнта та сервера.
5. Тестування зворотного зв'язку. Забезпечення можливості користувачам надсилати відгуки та повідомлення про помилки.

Перш за все варто звернути увагу, що для авторизованого користувача відкриваються додаткові пункти меню та активні кнопки на сайті.

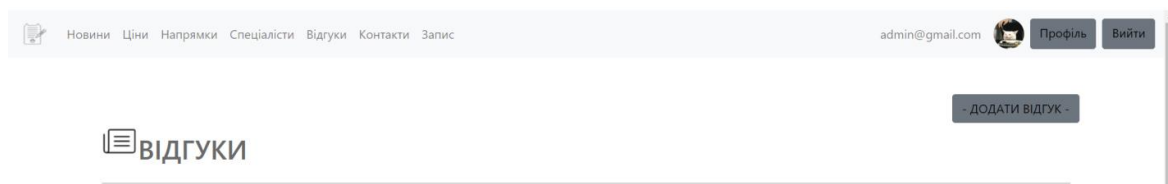


Рисунок 3.10. – Додаткові пункти для авторизованого користувача

Для неавторизованих користувачів спрацьовує відповідна сторінка з попередженням.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						46
Змн.	Арк.	№ докум.	Підпис	Дата		

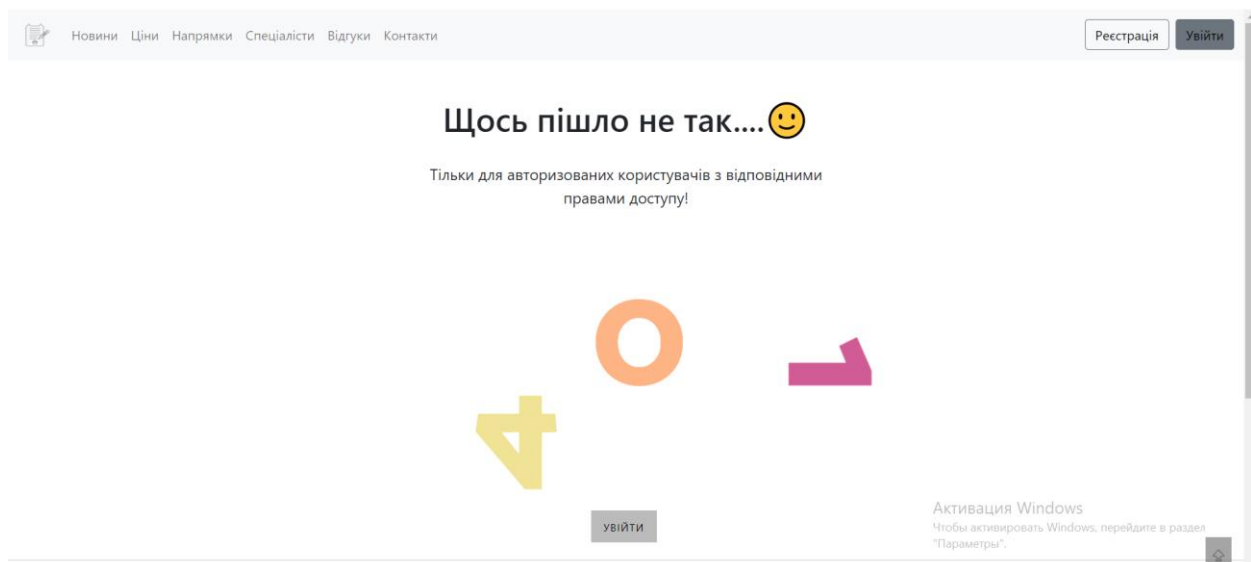


Рисунок 3.11. – Сторінка з попередженням для неавторизованих користувачів

Щоб запобігти переходу користувачу на відсутні сторінки та утримати його на сайті було розроблено власну сторінку помилки з кнопкою переходу до ГОЛОВНОГО МЕНЮ.

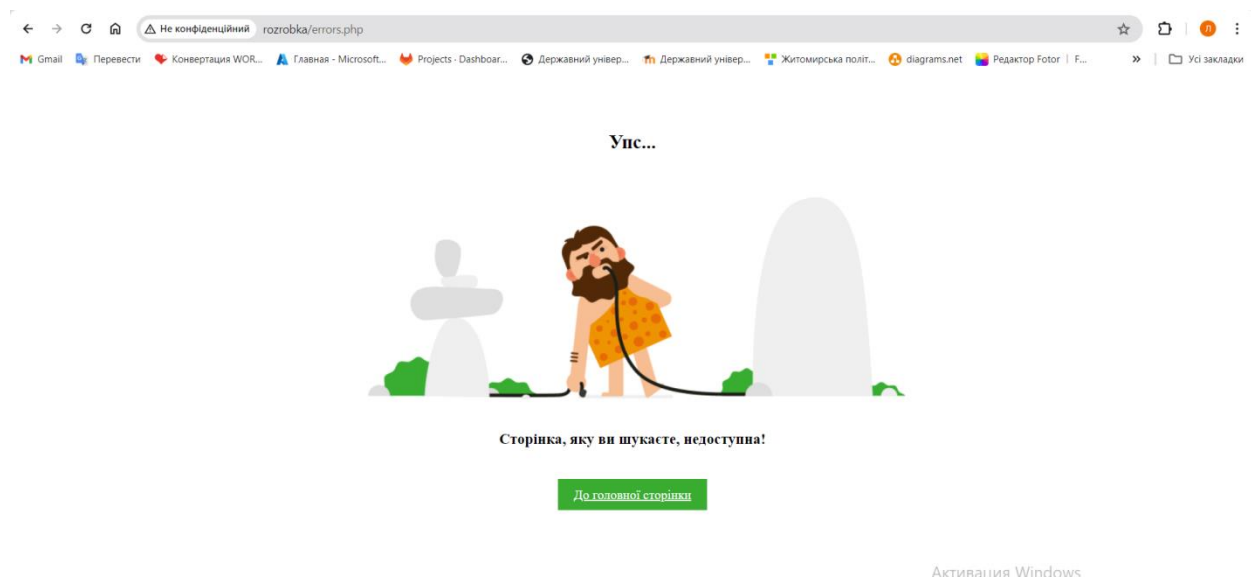


Рисунок 3.12. – Сторінка помилки

При відправленні пустих або не до кінця заповнених форм користувач має повідомлення з вказанням помилки та допоміжних даних для коректного заповнення.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						47
Змн.	Арк.	№ докум.	Підпис	Дата		

Рисунок 3.13. – Відправлення некоректно заповнених або пустих форм

Даний сайт, розроблений таким чином, щоб він оптимально відображався на різних пристроях, таких як смартфони, планшети й комп'ютери. Це означає, що дизайн та вміст сайту автоматично адаптується до розміру та орієнтації екрана пристрою, щоб забезпечити зручний й ефективний користувацький досвід.

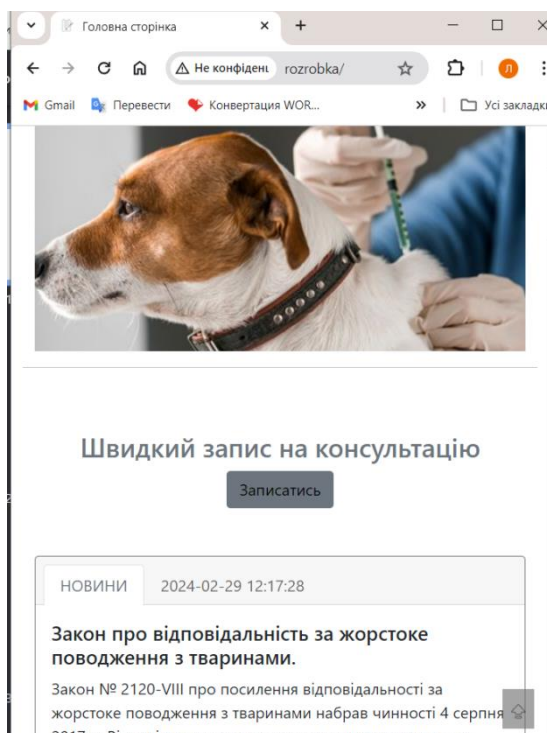


Рисунок 3.12. – Реакція сайту на зміну розмірів

Під час тестування програмного забезпечення ми маємо змогу перевірити, що розроблюваний продукт відповідає необхідним функціональним можливостям та оптимально працює в різних умовах. Ці тести допомагають

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						48
Змн.	Арк.	№ докум.	Підпис	Дата		

виявити та усунути будь-які потенційні проблеми або помилки, що забезпечує стабільну роботу сайту.

Висновки з третього розділу

У ході виконання третього розділу, що був присвячений інтерфейсу та порядку роботи з сайтом ветеринарної клініки, було охоплено три основні напрямки: встановлення та налаштування параметрів додатку, структура інтерфейсу та тестування роботи програмного комплексу. Було описано та показано основні ключові моменти над роботою сайту. Друга частина розділу була присвячена структурі інтерфейсу. У ній було представлено огляд різних компонентів і модулів, що складають інтерфейс програмного продукту. Також було розглянуто такий важливий аспект, як тестування роботи програмного комплексу.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час виконання кваліфікаційної роботи бакалавра було спроектовано та реалізовано сайт ветеринарного центру. Проведено аналіз предметної області та виділено ключові моменти над роботою у даному проєкті. На основі цього аналізу були визначені основні вимоги до сайту, розроблено архітектуру проєкту.

Проектування сайту охоплювало в собі аналіз та розробку основних варіантів використання з врахуванням моделювання структури проєкту, бази даних та його основних складових.

Описано алгоритм роботи з додатком та послідовні пункти варіантів його використання. Розглянуто основний функціонал, включаючи всі можливі аспекти сайту.

Також була надана детальна схема розгортання елементів сайту, описано взаємодію структурних елементів системи та наведено опис інтерфейсу розробленого програмного продукту для різних типів користувачів.

Додаток має простий та інтуїтивний інтерфейс, який дозволяє користувачам легко та швидко взаємодіяти з сайтом для задоволення власних потреб.

У розробці цього проєкту покладено особливий акцент на тестування роботи програмного забезпечення, щоб забезпечити його функціональність, надійність та продуктивність. Результатом є програмний комплекс, який має весь необхідний функціонал для швидкого та легкого користування сайтом. Завдяки тому, сайт здатний вирішувати найрізноманітніші завдання, тому програмний комплекс має великі можливості для розширення в майбутньому.

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Патерни проектування.[Електронний ресурс] – Режим доступу до ресурсу: <https://refactoring.guru/uk/design-patterns>
2. Елементи UML. [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-elements.html>
3. Class Diagram Tutorial. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/>
4. Використання UML діаграм. [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/ofzod>
5. Уніфікована мова UML. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.maxzosim.com/unifikovana-mova-modeluvannia/>
6. UML Activity Diagram Tutorial. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lucidchart.com/pages/uml-activity-diagram>
7. Як будувати UML діаграми. [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/forums/topic/40575/>
8. Створення схеми розгортання UML. [Електронний ресурс] – Режим доступу до ресурсу: <http://surl.li/omrxi>
9. Documentation phpMyAdmin. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.phpmyadmin.net/docs/>
10. Документація OpenServer. [Електронний ресурс] – Режим доступу до ресурсу: <https://ospanel.io/docs/>
11. Керівництво MySQL. [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sql/mysql/>
12. Ветеринарна клініка Авет. [Електронний ресурс] – Режим доступу до ресурсу: <https://avet.kiev.ua/>
13. Ветеринарна клініка vet.ua. [Електронний ресурс] – Режим доступу до ресурсу: <http://vet-ua.com/>
14. А.Зандстра, Ф.Матт. PHP Objects, Patterns, and Practice, 2013. Режим доступу до ресурсу: <https://www.apress.com/la/book/9781430260325>

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						51
Змн.	Арк.	№ докум.	Підпис	Дата		

15. Ветеринарна клініка АртВет. [Електронний ресурс] – Режим доступу до ресурсу: <https://artvet.kiev.ua/>
16. Довідник PHP. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/index.php>
17. Основні функціональні можливості та вхід в phpMyAdmin. [Електронний ресурс] – Режим доступу до ресурсу: <https://thehost.ua/ua/wiki/ispmanager/database/phpmyadmin>
18. PhpMyAdmin. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.phpmyadmin.net/>
19. Встановлення та налаштування. [Електронний ресурс] – Режим доступу до ресурсу: <https://timeweb.com/ru/community/articles/ustanovka-i-nastroyka-openserver>
20. Керівництво PHP. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.php.net/manual/en/index.php>
21. WebStorm. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.jetbrains.com/webstorm/learn/>
22. Діаграма розгортання. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.guru99.com/uk/deployment-diagram-uml-example.html>
23. Діаграма кооперації. [Електронний ресурс] – Режим доступу до ресурсу: <https://studfile.net/preview/5010027/page:4/>
24. Діаграма активності. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mindonmap.com/uk/blog/uml-activity-diagram/>
25. Діаграма послідовності. [Електронний ресурс] – Режим доступу до ресурсу: <https://www.maxzosim.com/sequence-diagrams/>

ДОДАТКИ

					ІПЗ.КР.Б – 121 – 24 – ПЗ	Арк.
						53
Змн.	Арк.	№ докум.	Підпис	Дата		

Програмний код

Class Core:

```

<?php
namespace Core;
use http\Client\Request;
class core{
    private static $instance;
    private static $mainTemplate;
    private static $db;
    private function __construct(){
        global $Config;
        spl_autoload_register('\Core\core::__autoload');
        self::$db = new \Core\DB($Config['Database']['Server'],
            $Config['Database']['Username'],
            $Config['Database']['Password'],
            $Config['Database']['Database']);
    }
    public static function getInstance(){
        if (empty(self::$instance)) {
            self::$instance = new core();
            return self::getInstance();
        } else
            return self::$instance;
    }
    public function getDB(){
        return self::$db;
    }
    public function init(){
        session_start();
        self::$mainTemplate = new Template();
    }
    public function run(){
        $path = $_GET['path'];
        $pathParts = explode('/', $path);
        $className = ucfirst($pathParts[0]);
        if (empty($className))
            $fullClassName = 'Controllers\Site';
        else $fullClassName = 'Controllers\' . $className;
        $methodName = ucfirst($pathParts[1]);
        if (empty($methodName))
            $fullMethodName = 'actionIndex';
        else $fullMethodName = 'action' . $methodName;
        if (class_exists($fullClassName)) {
            $controller = new $fullClassName();
            if (method_exists($controller, $fullMethodName)) {
                $method = new \ReflectionMethod($fullClassName, $fullMethodName);
                $paramArray = [];
                foreach ($method->getParameters() as $parameter) {
                    array_push($paramArray, isset($_GET[$parameter->name]) ?
                        $_GET[$parameter->name] : null);
                }
                $rez = $method->invokeArgs($controller, $paramArray);
                if (is_array($rez)) {
                    self::$mainTemplate->setParams($rez);
                } else
                    header('Location: http://course-project/errors.php');
            } else
                header('Location:http://course-project/errors.php');
        }
        public function done() {
            self::$mainTemplate->display('Views/layout/index.php');
        }
        public static function __autoload($className){
            $fileName = $className . '.php';
            if (is_file($fileName))
                include($fileName);
        }
    }
}

```

Class DB:

```

<?php

namespace Core;

```

					ІПЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						54
Змн.	Арк.	№ докум.	Підпис	Дата		

```

class DB
{
    protected $pdo;

    public function __construct($server, $login, $password, $database)
    {
        $this->pdo = new \PDO("mysql:host={$server};dbname={$database};charset=UTF8",
        $login, $password);
        $this->pdo->setAttribute(\PDO::ATTR_ERRMODE, \PDO::ERRMODE_EXCEPTION);
    }

    public function select($table, $fields = '*', $where = null, $orderBy = null, $limit
= null, $offset = null)
    {
        $fieldsStr = "*";
        if (is_string($fields))
            $fieldsStr = $fields;
        if (is_array($fieldsStr))
            $fieldsStr = implode(", ", $fields);
        $sql = "SELECT {$fieldsStr} FROM {$table}";
        if (is_array($where) && count($where) > 0) {
            $whereParts = [];
            foreach ($where as $key => $value)
                $whereParts[] = "{$key} = ?";
            $whereStr = implode(" AND ", $whereParts);
            $sql .= ' WHERE ' . $whereStr;
        }
        if (is_string($where))
            $sql .= ' WHERE ' . $where;
        if (is_array($orderBy)) {
            $orderByParts = [];
            foreach ($orderBy as $key => $value) {
                $orderByParts[] = "{$key} {$value}";
            }
            $sql .= ' ORDER BY ' . implode(' ', $orderByParts);
        }
        if (!empty($limit)) {
            if (!empty($offset))
                $sql .= " LIMIT {$offset}, {$limit}";
            else
                $sql .= " LIMIT {$limit}";
        }
        $sth = $this->pdo->prepare($sql);

        if (is_array($where) && count($where) > 0)
            $sth->execute(array_values($where));
        else
            $sth->execute();
        return $sth->fetchAll();
    }

    public function insert($table, $row)
    {
        $fieldsStr = implode(', ', array_keys($row));
        $valueParts = [];
        foreach ($row as $key => $value) {
            $valueParts[] = '?';
        }
        $valueStr = implode(', ', $valueParts);
        $sql = "INSERT INTO {$table} ({$fieldsStr}) VALUES ({$valueStr})";
        $sth = $this->pdo->prepare($sql);
        $sth->execute(array_values($row));
        return $this->pdo->lastInsertId();
    }

    public function delete($table, $where)
    {
        $sql = "DELETE FROM {$table}";
    }
}

```

					ІІЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						55
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        if (is_array($where) && count($where) > 0) {
            $whereParts = [];
            foreach ($where as $key => $value)
                $whereParts[] = "{$key} = ?";
            $whereStr = implode(" AND ", $whereParts);
            $sql .= ' WHERE ' . $whereStr;
        }
        if (is_string($where))
            $sql .= ' WHERE ' . $where;

        $sth = $this->pdo->prepare($sql);
        if (is_array($where) && count($where) > 0)
            $sth->execute(array_values($where));
        else
            $sth->execute();
    }

    public function update($table, $newPow, $where)
    {
        $sql = "UPDATE {$table} SET ";
        $setParts = [];
        $paramsArr = [];
        foreach ($newPow as $key => $value) {
            $setParts[] = "{$key} = ?";
            $paramsArr[] = $value;
        }
        $sql .= implode(', ', $setParts);
        if (is_array($where) && count($where) > 0) {
            $whereParts = [];
            foreach ($where as $key => $value) {
                $whereParts[] = "{$key} = ?";
                $paramsArr[] = $value;
            }
            $whereStr = implode(" AND ", $whereParts);
            $sql .= ' WHERE ' . $whereStr;
        }
        if (is_string($where))
            $sql .= ' WHERE ' . $where;
        $sth = $this->pdo->prepare($sql);
        $sth->execute($paramsArr);
    }
}

```

Class Template:

```

<?php

namespace Core;

/**
 * клас шаблонізатора
 */
class Template
{
    protected $parameters;

    public function __construct()
    {
        $this->parameters = [];
    }

    public function setParam($name, $value)
    {
        $this->parameters[$name] = $value;
    }

    public function getParam($name)
    {

```

					ІПЗ.КР.Б – 121 – 24 – ІІЗ	Арк.
						56
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        return $this->parameters[$name];
    }

    public function setParams($array)
    {
        foreach ($array as $key => $value)
            $this->parameters[$key] = $value;
    }

    public function render($path)
    {
        extract($this->parameters);
        ob_start();
        include($path);
        $html = ob_get_contents();
        ob_end_clean();
        return $html;
    }

    public function display($path)
    {
        echo $this->render($path);
    }
}

```

Class Controller:

<?php

namespace Core;

/**
 * базовий клас для всіх контролерів
 */

class Controller

```

{
    public function isPost()
    {
        if ($_SERVER['REQUEST_METHOD'] == 'POST')
            return true;
        else
            return false;
    }

    public function isGet()
    {
        if ($_SERVER['REQUEST_METHOD'] == 'GET')
            return true;
        else
            return false;
    }

    public function postFilter($fields)
    {
        return Utils::ArrFilter($_POST, $fields);
    }

    public function render($viewName, $localParams = null, $globalParams = null)
    {
        $tpl = new Template();
        if (is_array($localParams))
            $tpl->setParams($localParams);
        if (!is_array($globalParams))
            $globalParams = [];
        $moduleName = strtolower((new \ReflectionClass($this))->getShortName());

        $globalParams['PageContent'] = $tpl-
>render("Views/{ $moduleName}/{ $viewName}.php");
        return $globalParams;
    }
}

```

| | | | | | | |
|------|------|----------|--------|------|---------------------------|------|
| | | | | | ІПЗ.КР.Б – 121 – 24 – ІІЗ | Арк. |
| | | | | | | 57 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

    }

    public function renderMessage($type, $message, $localParams = null, $globalParams =
null)
    {
        $tpl = new Template();
        if (is_array($localParams))
            $tpl->setParams($localParams);
        $tpl->setParam('MessageText', $message);
        switch ($type) {
            case 'success' :
                $tpl->setParam('MessageClass', 'success');
                break;
            case 'error' :
                $tpl->setParam('MessageClass', 'danger');
                break;
            case 'info' :
                $tpl->setParam('MessageClass', 'info');
                break;
        }
        if (!is_array($globalParams))
            $globalParams = [];
        $globalParams['PageContent'] = $tpl->render("Views/layout/message.php");
        return $globalParams;
    }
}

```

| | | | | | | |
|------|------|----------|--------|------|--------------------------|------|
| | | | | | ІПЗ.КР.Б – 121 – 24 – ІЗ | Арк. |
| | | | | | | 58 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |