

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### КУРСОВА РОБОТА

з «Моделювання та аналіз програмного забезпечення»  
(назва дисципліни)  
на тему: «Веб-додаток розміщення інформаційних статей»

Студента (ки) 4 курсу ІПЗ-19-2 групи  
Спеціальності 121 «Інженерія  
програмного забезпечення»  
Лимарь Д.А.  
(підпис) (прізвище та ініціали)  
Керівник зав. каф. КН, к.т.н.,  
доцент Сугоняк І.І.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)  
Національна шкала \_\_\_\_\_  
Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії	_____	<u>Сугоняк І.І.</u>
	(підпис)	(прізвище та ініціали)
	_____	<u>Власенко О.В.</u>
	(підпис)	(прізвище та ініціали)
	_____	<u>Кравченко С.М.</u>
	(підпис)	(прізвище та ініціали)

## ЗМІСТ

<b>ВСТУП.....</b>	<b>3</b>
<b>1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ .....</b>	<b>5</b>
1.1 Постановка завдання .....	5
1.2 Обґрунтування вибору засобів моделювання .....	5
1.3 Вимоги користувачів.....	8
<b>2 РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО КОМПЛЕКСУ НА ЛОГІЧНОМУ РІВНІ .....</b>	<b>12</b>
2.1 Алгоритм роботи та стани програмної системи.....	12
2.2 Взаємодія об'єктів системи .....	19
<b>3 ФІЗИЧНА МОДЕЛЬ ТА ПРОТОТИП ПРОГРАМНОГО КОМПЛЕКСУ 21</b>	
3.1 Взаємодія компонентів системи.....	21
3.2 Архітектура програмного комплексу та його розгортання.....	22
<b>ВИСНОВКИ .....</b>	<b>24</b>
<b>ЛІТЕРАТУРА.....</b>	<b>25</b>
<b>ДОДАТКИ.....</b>	<b>26</b>

					ІПЗ.420005.090-КР						
Змн.	Арк.	№ докум.	Підпис	Дата	Веб-додаток розміщення інформаційних статей			Лім.	Арк.	Акрушів	
Розроб.		Лимарь Д.А.									
Перевір.		Сугоняк І.І.								2	32
Реценз.		П.І.Б.						ФІКТ, гр. ІПЗ-19-2			
Н. Контр.		П.І.Б.									
Затверд.		П.І.Б.									

## ВСТУП

Під час роботи над проектом “ Веб-додаток розміщення інформаційних статей” були розглянуті основні методи і принципи моделювання та аналізу програмного забезпечення які були використані в подальшому для написання даної курсової роботи.

**Мета курсової роботи полягає в тому, щоб проаналізувати вимоги та проблеми, які можуть виникнути при розробці веб-орієнтованої системи розміщення замовлень. Даний програмний продукт повинен дати первинний досвід користувачу у сфері вищого навчального закладу з розпорядженням коштів та іншу необхідну інформацію.**

- Завданням на курсову роботу є:
- аналіз та опис вимог користування;
- аналіз теоретичних засад моделювання програмного забезпечення;
- методи модулювання функцій та поведінки системи;
- проектування об’єктної структури системи;
- фізичне моделювання програмних комплексів;
- кодогенерація із моделей

Для проектування програмного додатку була використана програма UML.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Основною причиною використання мови UML є спілкування розробників між собою. Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту. Діаграми підвищують супровід проекту і полегшують розробку документації. Предметом дослідження є можливості застосування CASE-засобів проектування

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

програмного забезпечення. Об'єктом дослідження є методи та засоби проектування програмного забезпечення та уніфікація процесу проектування.

					ІПЗ.420005.090-КР	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

# 1 АНАЛІЗ ВИМОГ КОРИСТУВАЧА ТА КОНЦЕПТУАЛЬНЕ МОДЕЛЮВАННЯ

## 1.1 Постановка завдання

Веб-додаток розміщення інформаційних статей

1. Проектування бази даних;
2. Реалізація структури бази даних:
  - Створення таблиці користувачів.
  - Створення таблиці постів.
  - Створення таблиці категорій.
  - Створення таблиці тегів.
3. Робота з серверною частиною:
  - Програмний модуль CRUD користувачів.
  - Програмний модуль CRUD постів.
  - Програмний модуль CRUD категорій.
  - Програмний модуль CRUD тегів.
4. Інтеграція серверної частини у шаблон.
5. Наповнити систему інформацією.

## 1.2 Обґрунтування вибору засобів моделювання

UML є мовою широкого профілю, це - відкритий стандарт , який використовує графічні позначення для створення абстрактної моделі системи , званої UML-моделлю . UML був створений для визначення, візуалізації, проектування та документування, в основному, програмних систем. UML не є мовою програмування, але на підставі UML-моделей можлива генерація коду .

Візуальний підхід до проектування з використанням раціонального уніфікованого підходу та уніфікованої мови моделювання (Unified Modeling Language,UML) дозволяє ефективно вести боротьбу із постійно зростаючою

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		5

складністю ПЗ, здійснювати їх аналіз, будувати стабільну архітектуру складних програмних систем різного призначення. Система є складною, якщо розробники для складання деякого цілісного уявлення про неї розглядають її не з однією, а з багатьох різних точок зору: з позиції об'єктів і відносин між ними, бізнес та інших процесів. Крім того, програмне забезпечення розглядається програмними інженерами з позицій глобальних і локальних змінних, однозначно ідентифікованих імен змінних, інкапсуляції частин програмних кодів і багатьох інших точок зору.

Різні точки зору призводять до створення різних систем. Світогляд розробника ПЗ залежить від обраної моделі. Розробники баз даних (БД) основну увагу приділяють UML-моделям «сутність – зв'язок», де поведінка інкапсульована в процедурах збереження та сховищах. Аналітик структурного підходу створив би модель, у центрі якої є алгоритми і передача даних від одного процесу до іншого тощо. Результатом роботи розробника об'єктно-орієнтованої парадигми проектування буде система, архітектура якої заснована на множині класів і зразках взаємодії, що визначають, кооперації цих класів у реалізації певних сценаріїв. Незважаючи на багатство засобів, UML є простою для розуміння й застосування.

UML як явна модель полегшує спілкування. Опис моделей мовою UML дозволяє розв'язати третю проблему: явна модель полегшує спілкування. Деякі речі краще моделювати в тексті, інші – графічно. Насправді у більшості систем існують структури, які неможливо виразити мовою програмування. UML – графічна мова, що дозволяє розв'язати другу з описаних вище проблем. UML – щось більше, ніж просто набір графічних символів, кожен з яких має чітку певну семантику. І це означає, що один розробник може описати модель мовою UML, а інший може її однозначно інтерпретувати її, що вирішує першу зі згаданих проблем (комунікації).

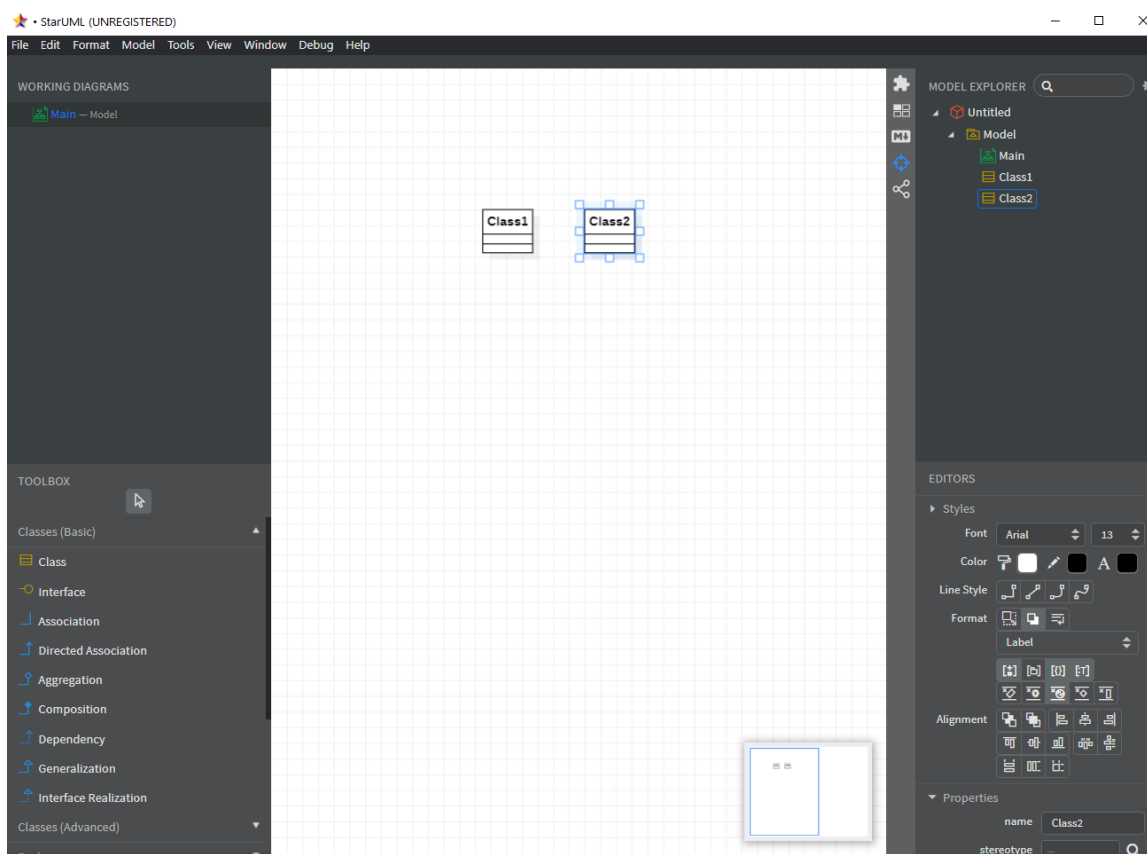
UML призначена для документування архітектури системи й усіх її деталей. Крім того, це мова для вираження вимог до системи й опису тестів.

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

Вона підходить для моделювання робіт на етапі проектування й керування версіями. Галузі використання UML як засобу моделювання й розроблення програмних систем у різних галузях (корпоративні інформаційні системи, телекомунікації, транспорт, авіація і космонавтика, наука, освіта, розподілені Web-сервіси тощо). Також Застосування UML не обмежене моделюванням ПЗ, а може застосовуватись і до не програмних систем.

Адаптується до користувача, StarUML надає максимальну ступінь адаптації середовища розробки користувача, пропонуючи настройку параметрів, які можуть впливати на методологію розробки програмного забезпечення, проектну платформу і мову.

Простота використання є найбільш важливою характеристикою при розробці додатків. Безкоштовна платформа StarUML вигідно відрізняється від своїх аналогів, в тому числі вона володіє великою кількістю особливостей такими як швидкий діалог, управління за допомогою клавіатури, огляд діаграм і багато іншого. Крім цього всього доволі зрозумілий інтерфейс для невідготовленого користувача.



					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

## Рисунок 1.1 – Інтерфейс програми StarUML

### 1.3 Вимоги користувачів

- **внутрішній користувач – адміністратор:**

1. Додавання користувачів, статей, тегів та категорій;
2. Контроль за діями користувачів додатку;
3. Контроль за стабільністю додатку;
4. Створення нових користувачів з ролями (Адмін / Користувач)

- **внутрішній користувач – користувач:**

1. Перегляд функціоналу сайта;
2. Взаємодія з особистим кабінетом;
3. Вподобання до постів;
4. Написання коментарів

Даний підрозділ (узагальнені і більш детальні ніж у підрозділі 1.1 першого розділу) описує високо рівневі вимоги, якими повинна володіти система (CRM оптової торгівлі фармацевтичними препаратами) управління взаємовідносинами з клієнтами, щоб мати можливість добре виконувати покладені на неї функції, тим самим відповідно задовольнивши всі зазначені стандарти, специфікації та інші формальні документи. Зазначені умови, яких повинен дотримуватись користувач для роботи з даною програмою та вказані основні переваги даної програми. Приведені можливості програми, обмеження та показники якості.

**Бізнес-вимоги:**

1. Основні цілі: додаток створений для можливості подання інформації авторів до читачів;
2. Можливості для авторів: розміщення статей ,обговорення із читачами;
3. Можливості для читачів: перегляд статей,збереження та обговорення;

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8



### **Функціональні вимоги:**

- В системі повинна бути представлена можливість авторизації користувачів, реєстрацію здійснює адмін з можливістю зміни ролі облікового запису ( адміністратор чи користувач);
- Валідація введених даних;
- Модифікація введених даних;
- Відображення результатів виконаної процедури;
- Можливість збереження інформації: Система повинна зберігати інформацію;

### **Не функціональні вимоги:**

#### **1. Сприйняття:**

- Час, потрібний для навчання роботи з програмою для звичайних користувачів — 1 години, для досвідчених — 20 хвилин;
- Час відповіді системи для звичайних запитів не повинен перевищувати 5 секунд, а для більш складних запитів — 9 секунд;
- Інтерфейс представлення програми повинен бути інтуїтивно зручним для користувача;

#### **2. Надійність:**

- Середній час безперервної роботи — 60 робочих днів;
- Максимальна норма помилок та дефектів в роботі програми - 1 помилка на 500 запитів користувача;

#### **3. Продуктивність:**

- Швидке отримання результатів при здійсненні функцій програми;
- Система повинна підтримувати мінімум 40 одночасно працюючих користувачів, пов'язаних з спільною базою даних.

#### **4. Можливість експлуатації:**

					ІПЗ.420005.090-КР	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

- Масштабування – система повинна мати можливість збільшувати потужності (продуктивність), зі збільшенням користувачів таким чином, щоб це аж ні як негативно не відобразилося на її роботі;
- Оновлення версій – Оновлення версій повинно здійснюватися в автоматичному режимі.

### Системні вимоги:

1. Вимоги до середовища виконання: Система повинна задовільнять вимогам на комп'ютері, що знаходиться в наступній мінімальній комплектації:
  - 2 гб оперативної пам'яті;
  - Процесор з тактовою частотою не нижче 1,7GHz;
  - Операційна система не нижче Windows 7;
  - Вимоги до операційного середовища (для сервера UNIX, до web-сервера Apache або Nginx);
  - PHP 5.6+ , MySQL.

Відповідно до детального аналізу вимог ми можемо побудувати діаграму варіантів використання (Use Case Diagram) «CRM оптової торгівлі фармацевтичними препаратами». Але для початку детальніше розглянемо поняття діаграми варіантів використання та її компоненти.

Діаграма варіантів використання - це UML діаграма за допомогою якої в графічному вигляді можна зобразити вимоги до розроблюваної системи, а також відношення між акторами та прецедентами в системі.

Діаграма має такі компоненти:

- Варіант використання (прецедент) – це послідовність дій, які повинні бути виконані системою при взаємодії з відповідним актором;
- Актор – це зовнішня до системи сутність, яка взаємодіє з нею для досягнення певних цілей.

Відношення:

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- Асоціація (association) – вказує конкретну роль, яку грає актор при взаємодії з варіантом використання;
- Розширення (extend) – визначає зв'язок екземплярів одного варіантів використання з більш загальним;
- Узагальнення (generalization) – вказує що варіант використання А може бути узагальненим до варіанту використання В; А – є спеціалізацією або нащадком, В- прашчур;
- Включення (include) – вказує що деяка поведінка варіанту використання являється складовою поведінки іншого варіанту використання.

Відповідно до усього вище згаданого була побудована діаграма варіантів використання зображена на рис.1.5.

Детальний опис варіантів використання наведено в додатку А.

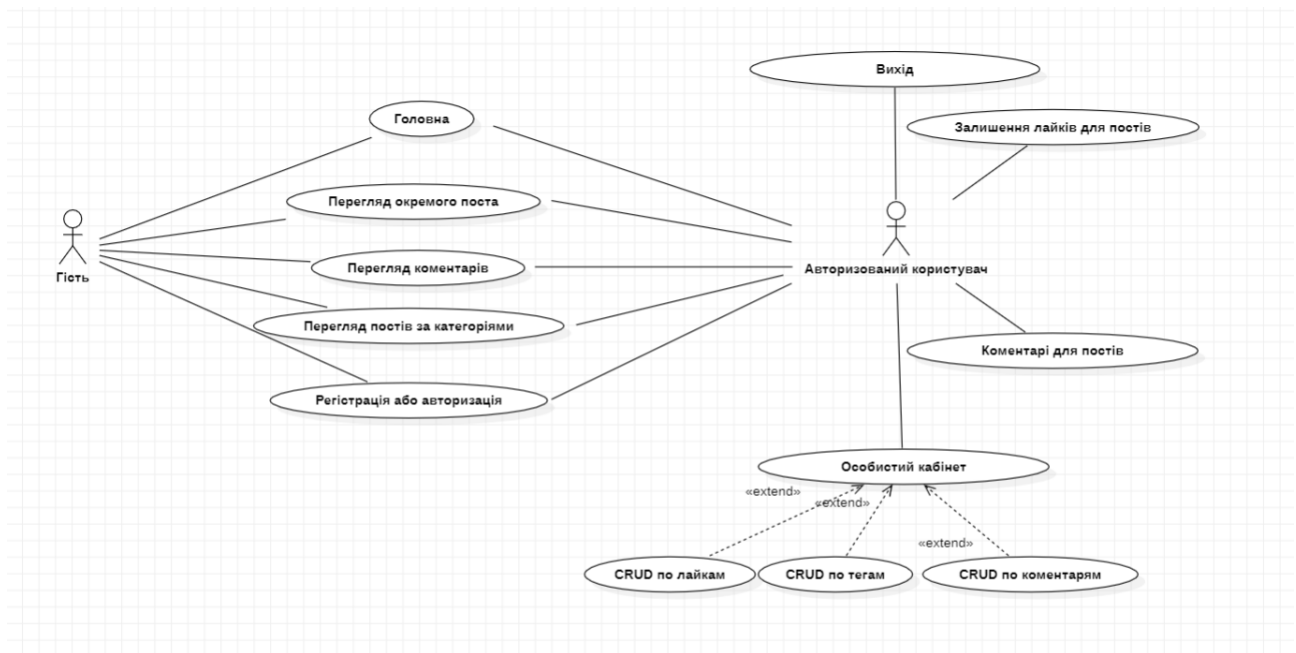


Рисунок 1.5 – Діаграма варіантів використання для гостя та користувача

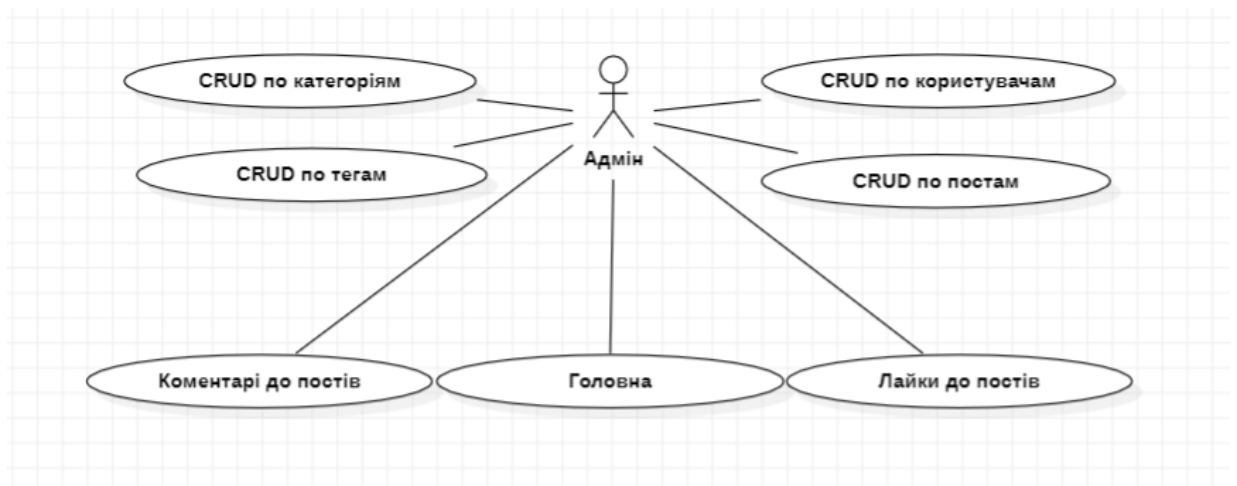


Рисунок 1.6 – Діаграма варіантів використання для адміністратора

## 2 РОЗРОБКА МОДЕЛІ ПРОГРАМНОГО КОМПЛЕКСУ НА ЛОГІЧНОМУ РІВНІ

### 2.1 Алгоритм роботи та стани програмної системи

Для моделювання процесу виконання операцій в мові UML використовуються так звані діаграми діяльності – це діаграма на якій показано розкладання деякої діяльності на її складові частини. Під діяльністю (activity) розуміють окремий крок у процесі. Одній діяльності відповідає окремий стан у системі з внутрішньою діяльністю і, принаймні, одна вихідна транзакція. Діаграми діяльності завжди пов'язано з класом, операцією або випадком використання.

Діаграми діяльності будуються з обмеженої кількості фігур, з'єднаних стрілочками. Найважливіші типи фігур:

- округлені прямокутники позначають дії;
- ромби позначають рішення;
- риси позначають початок (розподіл) чи кінець (об'єднання) паралельних активностей;
- чорний кружок позначає старт (початковий стан) процесу;

- чорний кружок в колі позначає кінець (кінцевий стан).

Переглянути створену діаграму діяльності можна на рис.2.1. Також представлені діаграми діяльності окремо для акторів адміністратора та консультанта, рис.2.2 та рис 2.3 відповідно.

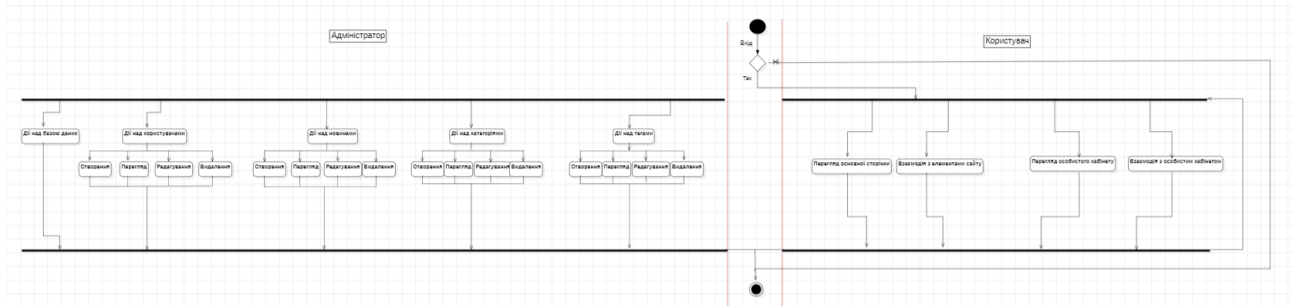


Рисунок 2.1 – Діаграма активності та відповідність станів та дій

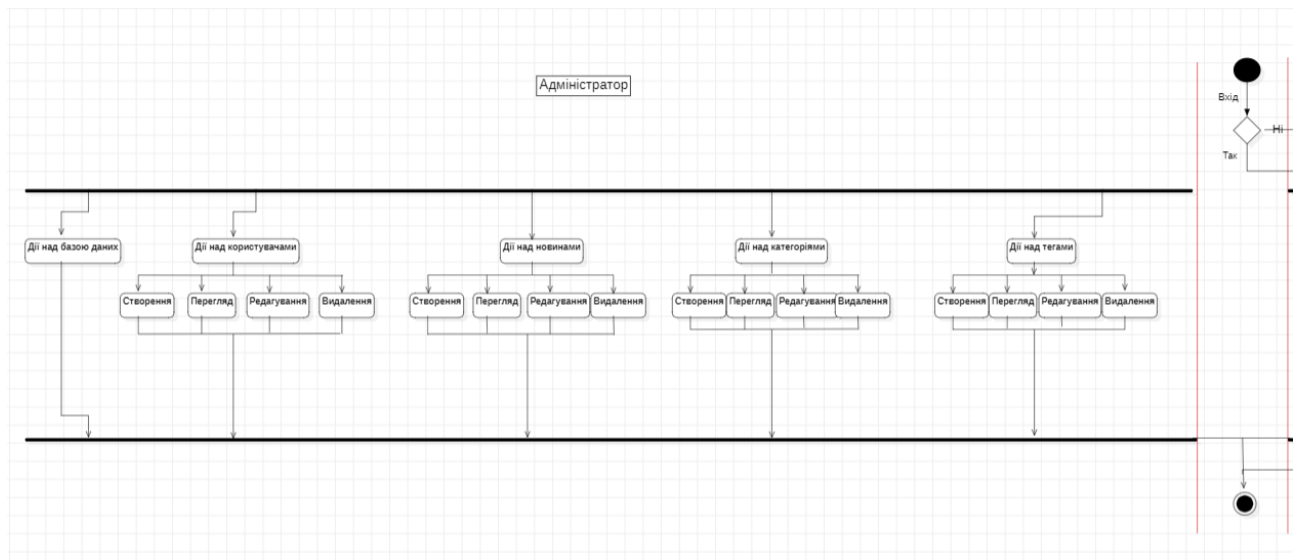


Рисунок 2.2 – Діаграма активності та відповідність станів та дій (адміністратор)

Опис діаграма активності (адміністратор):

- Вхід (так – перехід до компонентів системи, ні – завершення або перехід до початку);
- Дії над обліковими записами користувачів:
  - Додавання;
  - Видалення;
  - Редагування;
  - Перегляд.
- Дії над постами:

- Додавання;
- Видалення;
- Редагування;
- Перегляд.
- Дії над категоріями:
  - Додавання;
  - Видалення;
  - Редагування;
  - Перегляд.
- Дії над тегами:
  - Додавання;
  - Видалення;
  - Редагування;
  - Перегляд.
- Дії над базою даних;
- Завершення роботи.

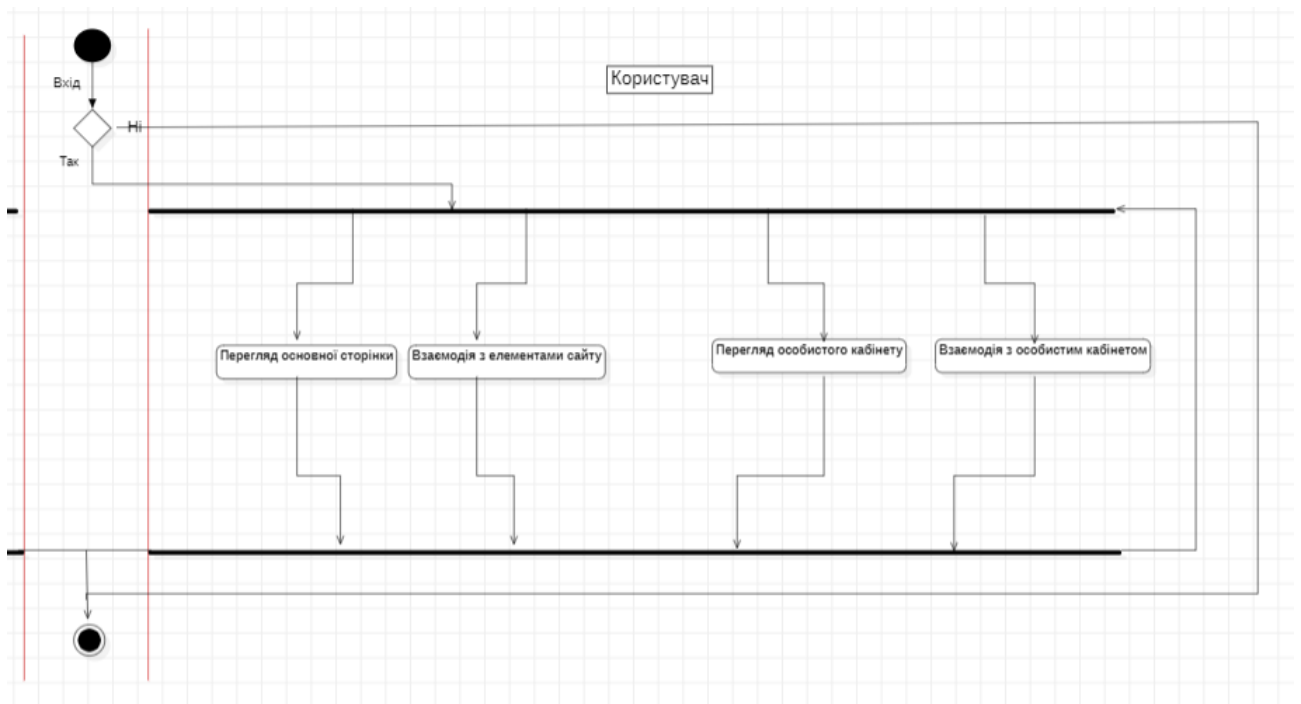


Рисунок 2.3 – Діаграма активності та відповідність станів та дій (консультант)

Опис діаграма активності (користувач):

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

- Вхід (так – перехід до компонентів системи, ні – завершення або перехід до початку);
- Дії на головній сторінці:
  - Авторизація;
  - Перегляд новини;
  - Вподобання;
  - Коментарі.
- Дії в особистому кабінеті:
  - Перегляд новин ,яким були залишенні вподобання;
  - Перегляд коментарів які були залишені;
- Завершення роботи.

**Діаграма класів** — статичне представлення структури моделі. Відображає статичні елементи, такі як: класи, типи даних, їх зміст та відношення. Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів.

**Клас** – це множина об’єктів, які мають однакову структуру, поведінку та відношення з об’єктами з інших класів. Обов’язковим являється ім’я класу, воно повинно бути унікальним, записується з великої літери та напівжирним шрифтом.

Клас називається абстрактним якщо не містить об’єктів; тоді його ім’я записується курсивом.

- Символ + - загальнодоступний (Public) – атрибут доступний з будь-якого класу.

- Символ # - захищений (Protected) – атрибут доступний лише підкласам даного класу.

- Символ — - закритий (Private) – атрибут недоступний жодному іншому класу.

**Ім’я атрибута** – являється обов’язковим.

					ІПЗ.420005.090-КР	Арк.
						15
Змн.	Арк.	№ докум.	Підпис	Дата		

**Кратність** – загальна кількість атрибутів даного типу що входять у склад даного класу. Якщо рядок атрибута підкреслений – це означає що атрибут приймає лише деяку множину значень, які вказані у рядку властивості.

**Операція** – це деякий сервіс, який надає екземпляр класу за певною вимогою.

**Рядок** - властивість потрібна для визначення значень властивостей, які можуть бути застосовані до даного елемента. Деякі операції можуть виконуватися одночасно, а деякі лише послідовно

**Ім'я приймає одне з 3 значень:**

1. Послідовне
2. Паралельне
3. Захищене – всі звершення до даної операції повинні бути впорядковані у часі.

**Агрегації** – клас включає в себе як складові інші класи.

**Композиції** – являється частинами випадку відношенням агрегації при якій складові знаходяться в середині цілого.

**Шаблонний метод** - це поведінковий патерн проектування, який визначає скелет алгоритму, перекладаючи відповідальність за деякі його кроки на підкласи. Патерн дозволяє підкласам перевизначати кроки алгоритму, не змінюючи його загальної структури

Виходячи з даних відомостей, було побудовано діаграму класів, зображену на рис.2.5.



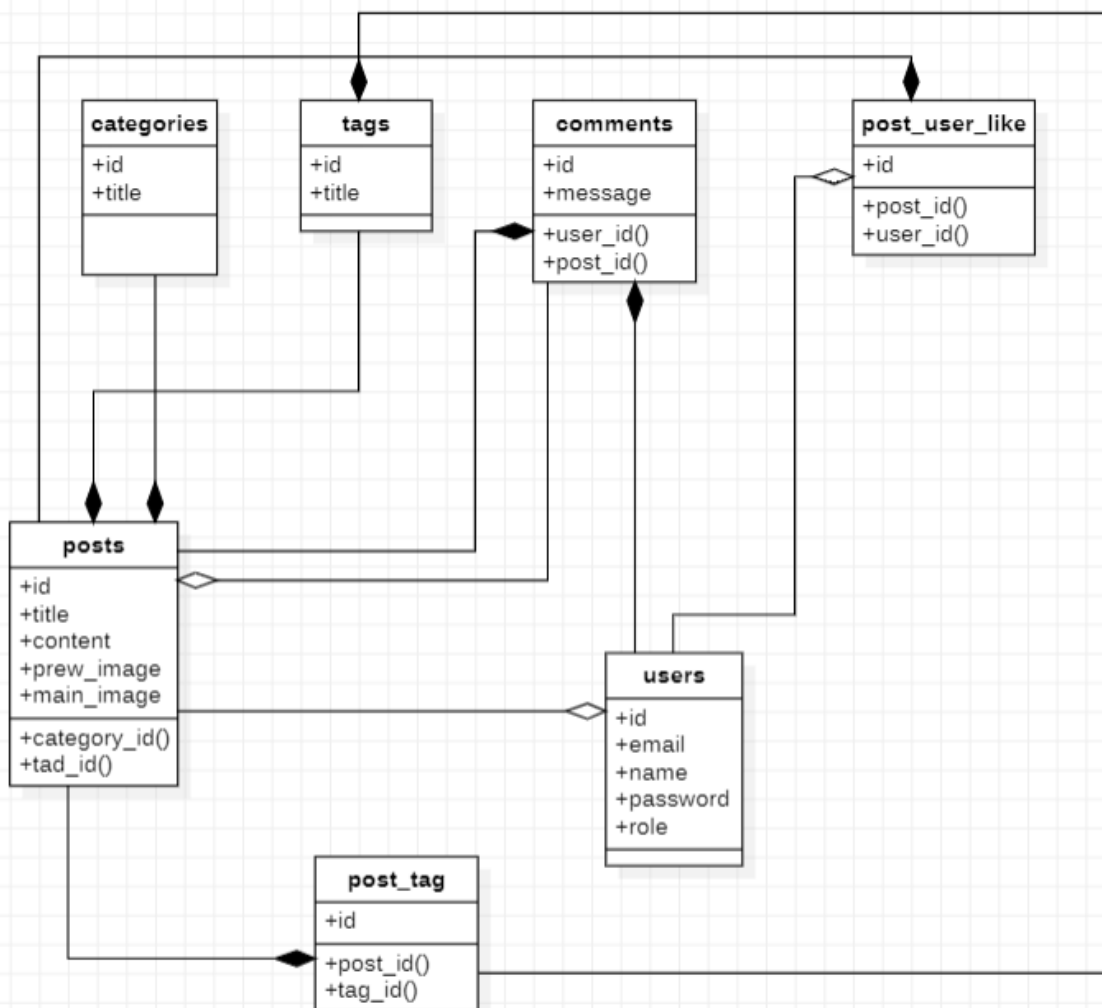


Рисунок 2.5 - Діаграма класів з патернами проектування

Опис класів, а також їх полів (у класах використовується англійська назва полів та методів з використанням так званого верблюжого регістру - camelCase):

- USER, клас, що описує користувачів у системі. Має такі властивості як: **email, password** – необхідні для авторизації у систему; персональні дані – **name**, а властивість **role** визначає тип користувача – адміністратор або користувач.
- POSTS, клас, що описує пости у системі. Має властивості опису послуги(**title, content, prewiev\_image, main\_image, category\_id**).
- Клас CATEGORIES містить поля з списком категорій (**title**);
- Клас TAGS містить поля з списком тегів (**title**);

- Клас POST\_TAG являє собою таблицю для поєднання постів та тегів (**title**);
- Клас COMMENTS містить поля з списком залишених коментарів конкретним користувачів до відповідного поста (**message**);

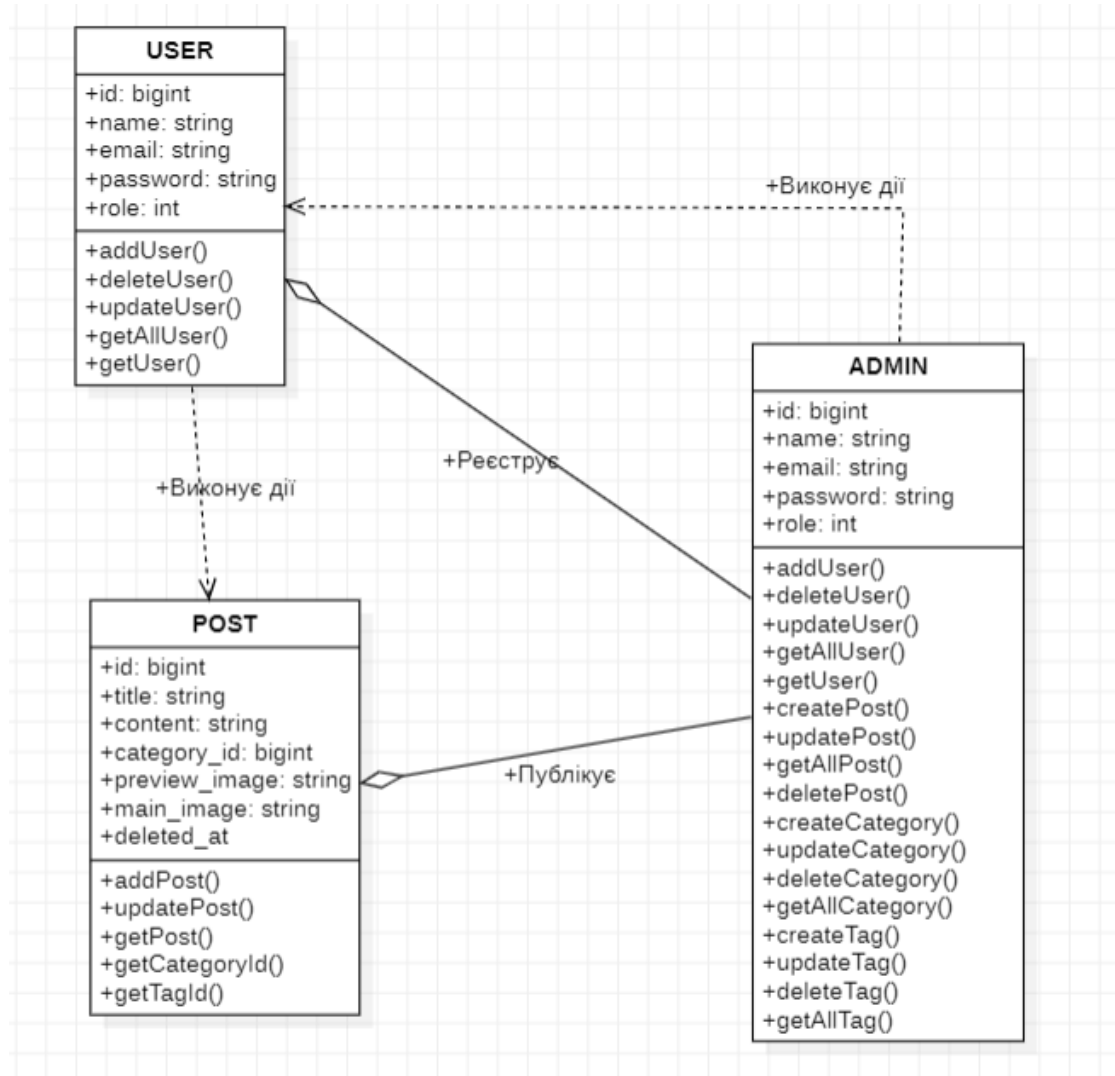


Рисунок 2.6 - Діаграма класів

Опис методів класів:

- USER, клас, що описує користувачів у системі. Має такі методи як: addPost, updatePost, getPost, getPost, getCategoryId, getTagId.
- ADMIN, клас, що керує всіма сутностями системи. Має такі методи: addPost, updatePost, getPost, getPost, getCategoryId, getTagId, getUser, getAllPost, createCategory, deleteCategory, getAllTag т.д..

Як було згадано раніше у проектуванні були використані патерни. Патерн

проектування — це типовий спосіб вирішення певної проблеми, що часто зустрічається при проектуванні архітектури програм.

На відміну від готових функцій чи бібліотек, патерн не можна просто взяти й скопіювати в програму. Патерн являє собою не якийсь конкретний код, а загальний принцип вирішення певної проблеми, який майже завжди треба підлаштовувати для потреб тієї чи іншої програми.

Ви можете цілком успішно працювати, не знаючи жодного патерну. Більше того, ви могли вже не раз реалізувати який-небудь з патернів, навіть не підозрюючи про це. Отже, навіщо ж знати патерни?

- Перевірені рішення;
- Стандартизація коду;
- Загальний словник програмістів.

Патерни відрізняються за рівнем складності, деталізації та охоплення проектованої системи. Розглянемо основні групи патернів:

- Породжуючі патерни піклуються про гнучке створення об'єктів без внесення в програму зайвих залежностей;
- Структурні патерни показують різні способи побудови зв'язків між об'єктами;
- Поведінкові патерни піклуються про ефективну комунікацію між об'єктами.

Ми не будемо розглядати усі існуючі патерни, а зупинимось на тих які були використані:

## 2.2 Взаємодія об'єктів системи

Діаграма послідовності - різновид діаграми в UML. Діаграма послідовності відображає взаємодії об'єктів впорядкованих за часом. Зокрема, такі діаграми відображають задіяні об'єкти та послідовність відправлених повідомлень. На діаграмі послідовностей показано у вигляді вертикальних ліній різні процеси або об'єкти, що існують водночас. Надіслані повідомлення

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

зображуються у вигляді горизонтальних ліній, в порядку відправлення. Визначені стандартом UML 2.0 діаграми послідовностей мають ті ж можливості що і визначені стандартом UML 1.x, та підтримують додаткові можливості зміни стандартного порядку повідомлень.

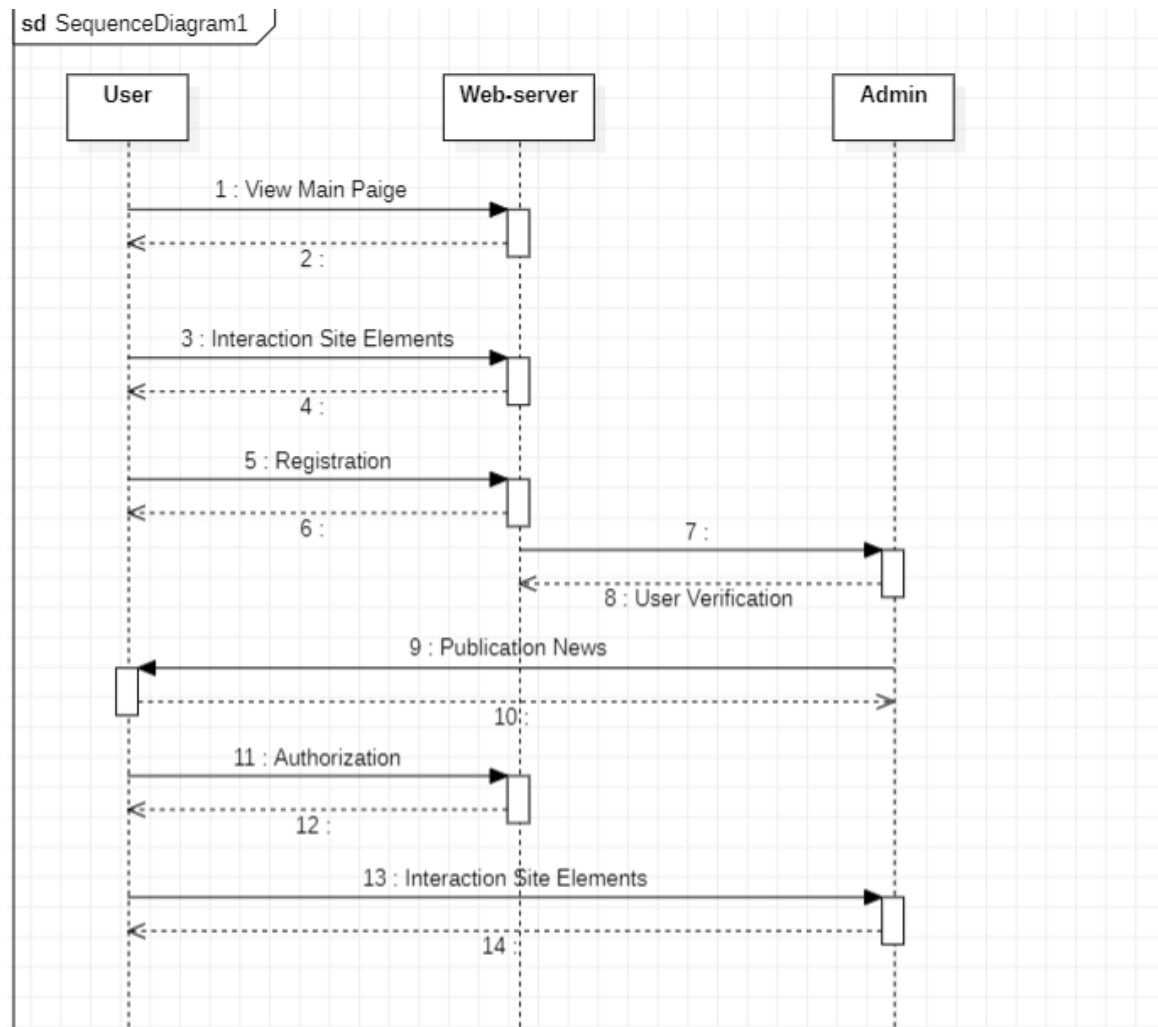


Рисунок 2.9 – Діаграма послідовності

На рис.2.10 показано як відбувається процес додавання, редагування, видалення замовлень та виконання відповідних дій з клієнтами, а також перегляд наявних товарів та робота зі звітами, ці дії виконує консультант.

### 3 ФІЗИЧНА МОДЕЛЬ ТА ПРОТОТИП ПРОГРАМНОГО КОМПЛЕКСУ

#### 3.1 Взаємодія компонентів системи

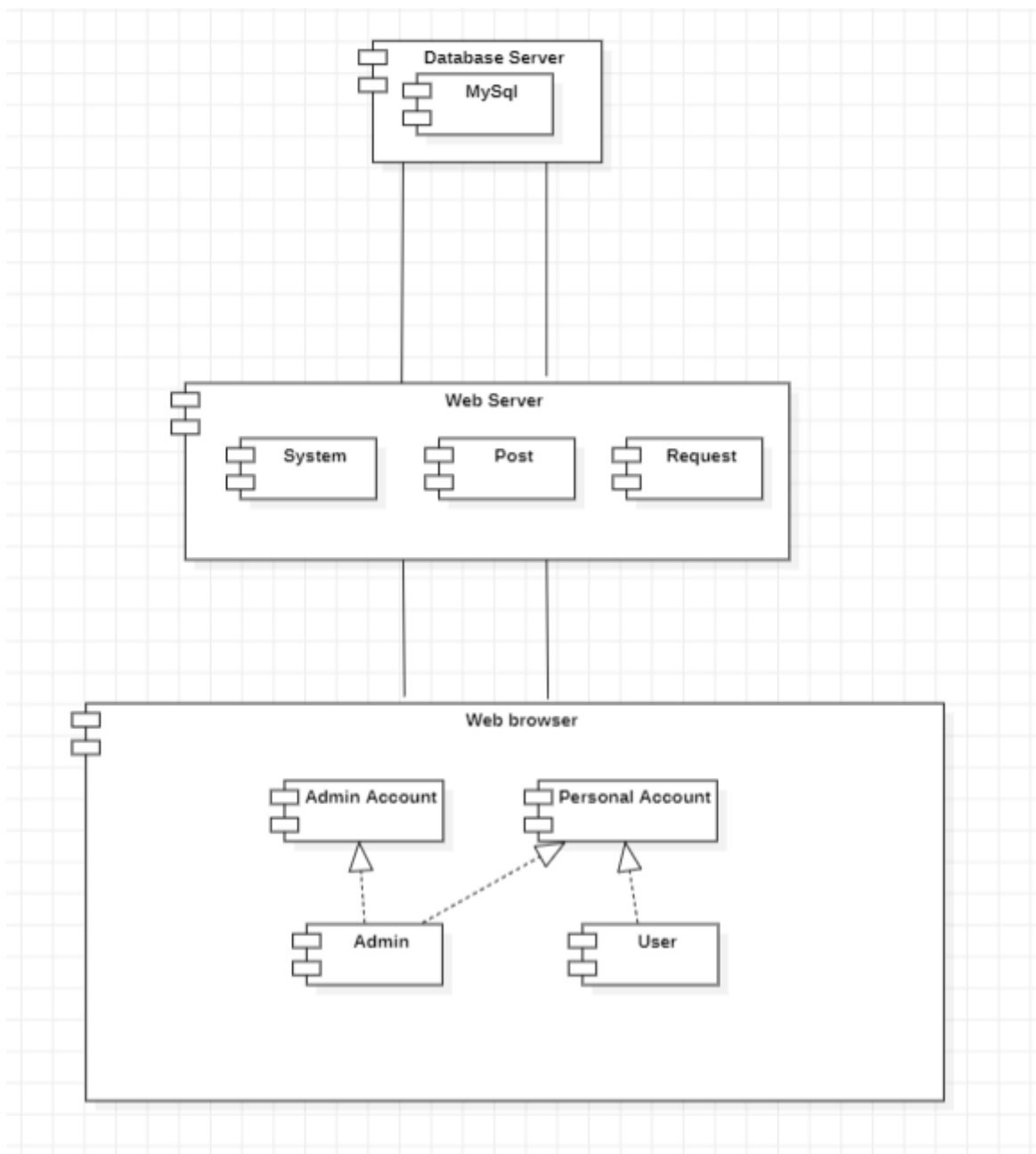


Рис.3.1. Діаграма компонентів

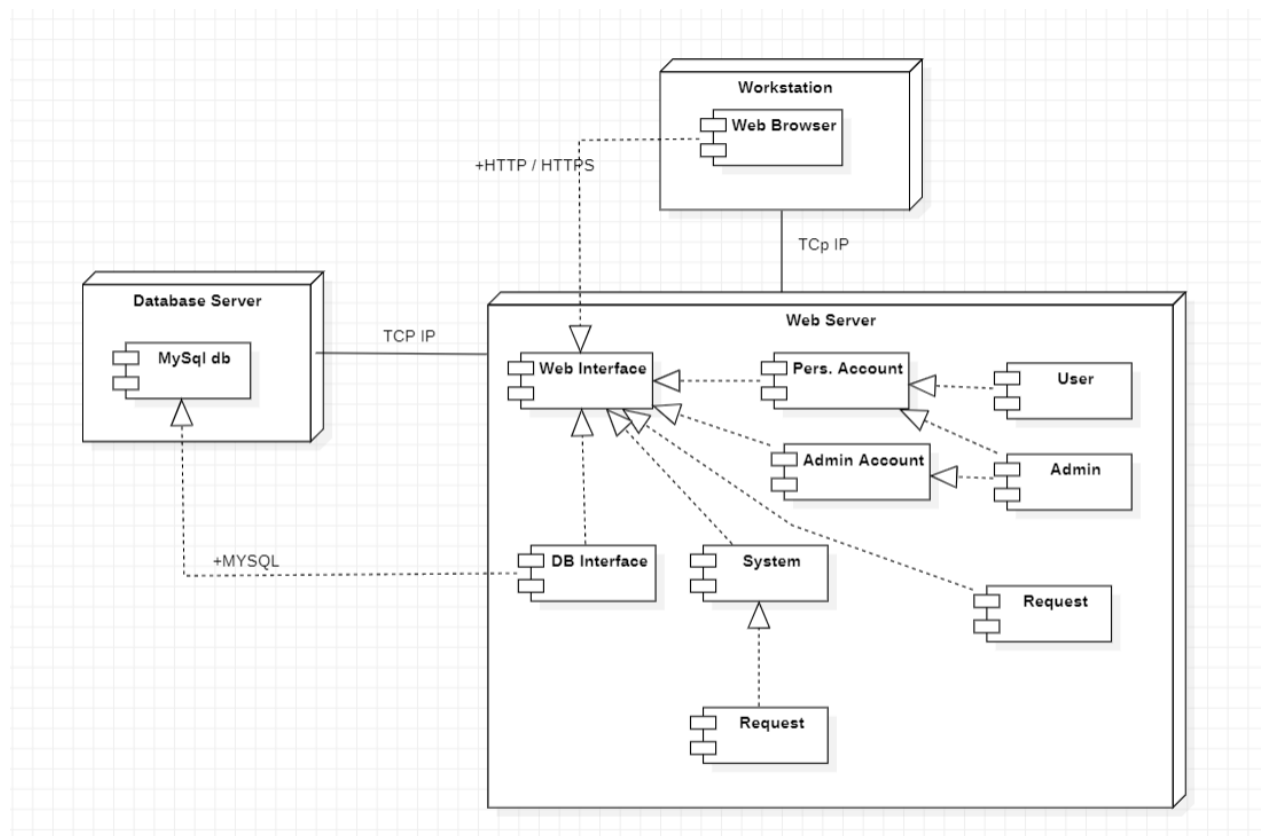
Діаграма компонентів відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись. Модуль програмного забезпечення може бути представлено як компоненту. Деякі компоненти існують під час компіляції, деякі - під час компонування, а деякі під час роботи програми.

На діаграмі показані такі елементи: – Компоненти – фізично існуюча частина системи, яка забезпечує реалізацію класів і відносин, а також функціональної поведінки модельованої програмної системи.

На даній діаграмі це:

- MySQL – база даних;
- System – компонент, що відповідає за системні операції;
- Post – створені новини;
- Requests – створення та збереження заявок ;
- Admin Account – кабінет адміністратора;
- Personal Account – персональний кабінет;
- User – тип користувача у системі(користувач);
- Admin – тип користувача у системі(адмін);
- Залежності:
- Відношення залежності

### 3.2 Архітектура програмного комплексу та його розгортання



### Рис.3.2. Діаграма розгортання

Діаграма розгортання – діаграма, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Діаграма має наступну структуру:

- Web Interface – веб інтерфейс, у нашому проєкті це шаблони сайту;
- DataBase Interface – інтерфейс для взаємодії з базою даних;
- Web Browser – програмний продукт встановлений на пристрої необхідний користувачам системи для взаємодії з системою.

					ІПЗ.420005.090-КР	Арк.
						23
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВКИ

Курсова робота призначена моделюванню додаток розміщення інформаційних статей з метою покращення зручності подання інформації. В процесі роботи були створені діаграми UML за допомогою редактора StarUML, а саме:

- Діаграма варіантів використання на якій відображаються відносини між акторами та прецендентами які вони можуть виконувати. На діаграмі присутні прецеденти як загальні для усіх акторів так і індивідуальні для кожного.
- Діаграма діяльності сприяє кращому аналізу роботи системи перегляду, скачування користувачем та керування веб-сайтом адміністратором.
- Діаграма класів на якій показано структуру класів для працюванню додатка.
- Діаграма послідовності на якій зручно стежити за класами які задіюються для певних дій у системі та методи які в них використовуються.
- Діаграма компонентів.
- Діаграма розгортання.

На основі створених діаграм було створено код на мові PHP, за допомогою функції кодогенерації плагіну на StarUML

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24



## ЛІТЕРАТУРА

1. Леоненков А. В. Самоучитель UML / А. В. Леоненков. – СПб. : БХВПетербург, 2004. – 432 с.
2. Г. Буч, Дж. Рамбо , А. Джекобсон Язык UML. Руководство пользователя.:
3. Пер. с англ. – М.: ДМК, 2000. – 432с.
4. Кватрани Т. Визуальное моделирование с помощью Rational Rose 2002 и
5. К. Ларман, Применение UML 2.0 и шаблонов проектирования (3-е издание)
6. Паттерны проектирования. СПб. Питер, 2006. - 366 с.
7. Бабич Введение в UML. – Электронный ресурс. – <http://www.intuit.ru/>
8. Леоненков А.В.Нотация и семантика языка UML. –
9. Электронный ресурс. – <http://www.intuit.ru/departement/pl/umlbasics/>
10. 12. Планирование сайта с помощью UML. – Электронный ресурс. –
11. <https://www.webmascon.com/topics/planning/15a.asp>
12. Теория и практика UML. Диаграмма последовательности. –
13. Электронный
14. ресурс. – [http://it-gost.ru/articles/view\\_articles/94](http://it-gost.ru/articles/view_articles/94)
15. 14. Отношение классов от UML к коду. – Электронный ресурс. –
16. <https://habr.com/post/150041/>

					<i>ІПЗ.420005.090-КР</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

## ДОДАТКИ

### Додаток А

Таблиця А.1 – Специфікація прецедентів

Ім'я варіанту використання: Перегляд діючих та усіх надаваних послуг		Рівень важливості: високий
Основний актор: Гість		
Короткий опис:		Перегляд основної інф. на сайті, взаємодія з деякими елементами сайту
Передумови:		Не авторизований в системі
Пост-умови:		
Відносини:		Не відчутні
	Асоціація:	
	Включення:	Регістрація/Авторизація
	Розширення:	
	Узагальнення:	
Основний потік:		Гостю для взаємодії зі всіма елементами потрібно авторизуватися чи зареєструватися
Ім'я варіанту використання: Перегляд діючих та усіх надаваних послуг		Рівень важливості: високий

Ім'я варіанту використання: Використання веб-додатку		Рівень важливості: високий
Основний актор: Авторизований користувач		
Короткий опис:		Взаємодія зі всіма елементами сайту, та робота із особистим кабінетом
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		Не відчутні
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	

Узагальнення:

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Основний потік:	Користувачі	
Альтернативні/виключні потоки:	Не виведена жодна інформація	

Ім'я варіанту використання: Робота веб-додатку		Рівень важливості: високий
Основний актор: Адміністратор		
Короткий опис:		Контроль за роботою сайту
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		Не відчутні
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Веб-додаток
Альтернативні/виключні потоки:		Не виведена жодна інформація

Ім'я варіанту використання: Додавання користувачів та ролі		Рівень важливості: високий
Основний актор: Менеджер		
Короткий опис:		Додавання адміністратором користувачів та надавання їм ролі адміністратора або читача
Передумови:		Ініціалізація в системі
Пост-умови:		
Відносини:		Не відчутні
	Асоціація:	
	Включення:	Ініціалізація
Розширення:		
	Узагальнення:	
Основний потік:	Користувачі та ролі	

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Альтернативні/виключні потоки:	Не виведена жодна інформація
Ім'я варіанту використання: Додавання користувачів та ролі	Рівень важливості: високий

Ім'я варіанту використання: Перегляд бази користувачів	Рівень важливості: високий
Основний актор: Менеджер	
Короткий опис:	Перегляд в базі користувачі
Передумови:	Ініціалізація в системі
Пост-умови:	
Відносини:	Не відчутні
	Асоціація:
	Включення:
	Ініціалізація
	Розширення:
	Узагальнення:
Основний потік:	Веб-додаток
Альтернативні/виключні потоки:	Не виведена жодна інформація

Ім'я варіанту використання: Основна інформація сайту	Рівень важливості: високий
Основний актор: Адміністратор	
Короткий опис:	Робота адміністратора по додаванню основних сутностей та їх подальше редагування
Передумови:	Ініціалізація в системі
Пост-умови:	
Відносини:	Не відчутні
	Асоціація:
	Включення:
	Ініціалізація

Розширення:

	Узагальнення:	
Основний потік:	База даних	
Альтернативні/виключні потоки:	Не виведена жодна інформація	
Ім'я варіанту використання: Основна інформація сайту	Рівень важливості: високий	

Ім'я варіанту використання: Дії (додавання, редагування, видалення) над товарами		Рівень важливості: високий
Основний актор: адміністратор		
Короткий опис:		Додавання, редагування, видалення товарів в систему
Передумови:		Ініціалізація адміністратора в системі
Пост-умови:		Зберігання, видалення, оновлення даних
Відносини:		Не відчутні
	Асоціація:	
	Включення:	Ініціалізація
	Розширення:	
	Узагальнення:	
Основний потік:		Додавання, редагування, видалення даних
Альтернативні/виключні потоки:		Не виведена жодна інформація

Ім'я варіанту використання: Контроль за правилами веб-додатку		Рівень важливості: високий
Основний актор: Менеджер		
Короткий опис:		Контроль за небажаним контентом
Передумови:		Ініціалізація
Пост-умови:		
Відносини:		Не відчутні
	Асоціація:	
	Включення:	Ініціалізація
Розширення:		
	Узагальнення:	
Основний потік:	Коментарі	
Альтернативні/виключні потоки:		Не виведена жодна інформація
Ім'я варіанту використання: Контроль за правилами веб-додатку		Рівень важливості: високий



## Додаток Б

### Клас User

```
public function create()
{
}

public function delete(User $user)
{
}

public function edit(User $user)
{
}

public function invoke()
{
}

public function show(User $user)
{
}

public function store(StoreRequest $request)
{
}

public function update(UpdateRequest $request, User $user)
{
}

public static function getRoles()
{
    return [
        self::ROLE_ADMIN => 'Admin',
        self::ROLE_READER => 'Читатель',
    ];
}
```

### Клас POST

```
public function store($data): void
{
}

public function update($data, $post)
{
}

public function create()
{
}
```

					ІПЗ.420005.090-КР	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

```

}
public function delete(Post $post)
{
}
public function edit(Post $post)
{
}
public function __invoke()
{
}
public function show(Post $post)
{
}
public function store(StoreRequest $request)
{
}
public function update(UpdateRequest $request, Post $post)
{
}

```

					ІПЗ.420005.090-КР	Арк.
						32
Змн.	Арк.	№ докум.	Підпис	Дата		