

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра інженерії програмного забезпечення

КВАЛІФІКАЦІЙНА РОБОТА

«Довідково-інформаційна система агентства нерухомості»

Назва теми


Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

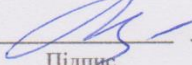
Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

Шифр КвРІПЗ.190131.01.07.ПЗ

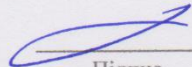
Виконала студентка IV курсу групи ІПЗ-19-1  А. Р. Доміна
Підпис Ініціали, прізвище

Керівник канд. техн. наук, доцент  О. М. Яшина
Науковий ступінь, звання Підпис Ініціали, прізвище

Нормоконтролер канд. техн. наук, доцент  Г. І. Радельчук
Підпис Ініціали, прізвище

До захисту допускаю:

Завідувач кафедри інженерії програмного забезпечення

 Л. П. Бедратюк
Підпис Ініціали, прізвище

6 червня 2023 р.

Хмельницький 2023

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет Інформаційних технологій

Кафедра Інженерії програмного забезпечення

Рівень вищої освіти Перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Л. П. Бедратюк

02 01 2023 р.

103

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Доміній Анастасії Романівні

Прізвище, ім'я, по батькові студента

1. Тема роботи Довідково-інформаційна система агентства нерухомості

Керівник роботи Яшина Оксана Миколаївна, канд. техн. наук, доцент

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2023 р. №5

2. Строк подання студентом роботи на кафедру 01.06.23 р.

3. Вихідні дані до роботи Методичні матеріали до кваліфікаційної роботи

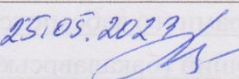
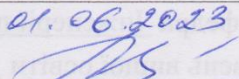
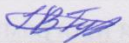

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити)

Дослідження предметної області та постановка задачі, проектування програмного забезпечення, програмна реалізація, тестування програмної системи

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень)

Три креслення (діаграма варіантів використання, модульна структура ДІС, діаграма сутність-зв'язок)

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Радельчук Г. І., доцент кафедри ІПЗ	25.05.2023 	01.06.2023 
Антиплагіат	Гурман І. В., доцент кафедри ІПЗ	5.06.2023 	5.06.2023 

7. Дата видачі завдання «02» січня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

Назва етапів (розділів) кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1 Ознайомлення з тематикою кваліфікаційної роботи (КвР), визначення та узгодження індивідуальних тем КвР	01.12 – 31.12.2022	
2 Збір матеріалу за темою КвР; дослідження предметної області, в якій планується використання програмного забезпечення (ПЗ), визначення задач та вимог, розробка технічного завдання	02.01 – 31.01.2023	
3 Проектування програмного забезпечення	01.02 – 28.02.2023	
4 Програмна реалізація з використанням відповідних засобів розробки	01.03 – 10.04.2023	
5 Тестування програмного забезпечення	11.04 – 30.04.2023	
6 Написання вступу, загальних висновків, оформлення переліку джерел посилання та додатків. Оформлення пояснювальної записки КвР згідно вимог	01.05 – 25.05.2023	
7 Попередній захист КвР	травень 2023 (згідно графіка)	
8 Перевірка КвР на плагіат, нормоконтроль, отримання відгуків, рецензій та інших супровідних документів. Брошування (зшиття) пояснювальної записки.	26.05 – 30.05.2023	
9 Здача КвР на кафедру; підготовка КвР для розміщення у репозиторії ХНУ; підготовка до захисту та захист КвР	з 01.06.2023	

Студент


 Підпис

 А. Р. Доміна
 Ініціали, прізвище

Керівник роботи


 Підпис

 О. М. Яшина
 Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Довідково-інформаційна система агентства нерухомості».

Автор роботи: Доміна Анастасія Романівна.

Керівник роботи: Яшина Оксана Миколаївна.

Пояснювальна записка: 88 с., 33 рис., 13 табл., 2 дод., 40 джерел.

Графічна частина: 3 креслення у форматі А3.

ЕЛЕКТРОННА КОМЕРЦІЯ, ВЕБ ЗАСТОСУНОК, НЕРУХОМІСТЬ, ASP.NET MVC, MYSQL, HTML, CSS, C#.

Мета кваліфікаційної роботи: спроектувати та розробити програмний продукт, який дозволить користувачам швидко знаходити, оглядати та домовлятися про оренду чи купівлю опублікованої на сайті нерухомості в будь-якій точці світу, не переживаючи за безпеку себе і своїх даних.

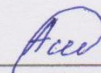
Під час виконання визначена специфіка роботи довідково-інформаційних систем агентств нерухомості, виконано проектування структури, логічної моделі бази даних та інтерфейсу користувача.

Систему реалізовано на мові програмування C#, з використанням шаблону проектування MVC, системи керування базами даних MySQL та HTML і CSS для розробки інтерфейсу користувача.

У результаті проектування здійснена програмна реалізація довідково-інформаційної системи агентства нерухомості з можливістю пошуку та огляду опублікованої на сайті нерухомості та замовлення дзвінку від орендодавця або продавця.

30.05.23

Дата



Підпис





ВІДОМІСТЬ ДОКУМЕНТІВ

№ рядка	Формат	Позначення документа	Найменування документа	К-сть аркушів	№ екз.	Примітка
			<u>Текстові документи</u>			
1	A4	КвРІПЗ.190131.01.07.ПЗ	Пояснювальна записка	88		
2	A4		Завдання на кваліфікаційну роботу	2		
3	A4		Анотація	1		
			<u>Графічні документи</u>			
4	A3	КвРІПЗ.190131.01.07.E8	Діаграма варіантів використання	1		
5	A3	КвРІПЗ.190131.01.07.E8	Модульна структура ДІС	1		
6	A3	КвРІПЗ.190131.01.07.E8	Діаграма сутність-зв'язок	1		

КвРІПЗ.190131.01.07.ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата
Виконала		Доміна А. Р.	<i>А.Р.Д.</i>	5.06
Керівник		Яшина О. М.	<i>О.М.Я.</i>	5.06
Н. контр.		Радельчук Г. І.	<i>Г.І.Р.</i>	01.06
Зав. каф.		Бедратюк Л.П.	<i>Л.П.Б.</i>	6.06
Довідково-інформаційна система агентства нерухомості				
Відомість документів				
Літ.	Арк.	Аркушів		
	1	1		
ХНУ, ІПЗ-19-1				

ЗМІСТ

Вступ	7
1 Дослідження предметної області та постановка задачі	9
1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей	9
1.2 Аналіз наявного програмно-технічного забезпечення предметної області	11
1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання.....	21
2 Проектування довідково-інформаційної системи	26
2.1 Проектування архітектури та структури системи	26
2.2 Проектування логічної моделі бази даних	30
2.3 Проектування інтерфейсу користувача	36
2.4 Аналіз та вибір технологій і методів реалізації системи	41
3 Програмна реалізація та тестування	44
3.1 Реалізація бази даних.....	44
3.2 Реалізація модулів системи.....	47
3.3 Інструкція користувача	56
3.4 Вимоги до апаратно-програмних засобів	62
3.5 Тестування системи	63
Висновки	67
Перелік джерел посилання.....	69
Додаток А Презентаційні матеріали	73
Додаток Б Технічне завдання	80

КвРІПЗ.190131.01.07.ПЗ								
Змн.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система агентства нерухомості Відомість документів			
		Виконала Доміна А. Р.		5.06		Літ.	Арк.	Аркушів
		Керівник Яшина О. М.		5.06			6	88
		Н. контр. Радельчук Г. І.		01.06		ХНУ, ІПЗ-19-1		
		Зав. каф. Бедратюк Л.П.		6.06				

ВСТУП

Щодня тисячі людей продають, купують, знімають і здають в оренду житло. Операції з нерухомістю – справа копітка, не проста та займає чимало часу, саме тому лише одиниці наважуються братися за неї самостійно.

Найкращий та найбільш надійний спосіб здійснювати операції – здійснювати їх через посередника. З професійною допомогою агентств нерухомості можливо зробити все максимально безпечно і вигідно для своєї кишені.

Переваги звернення в агентство досить численні:

- не потрібно витратити час на самостійний пошук варіантів;
- клієнт може сміливо довірити всі папери для оформлення ріелтору;
- угоди укладаються в найкоротші строки.

Професійний працівник агентства веде угоду від нульової стадії до фіналу – передачі квартири від покупця до продавця, при цьому, ретельно стежачи за юридичною чистотою операції, гарантуючи своїм клієнтам безпеку і спокій.

Клієнти не завжди мають змогу чи час «бігати» від одного об'єкта нерухомості до іншого, в пошуках найкращого, тому надають перевагу шукати нерухомість для покупки або оренди на просторах мережі Інтернет.

Саме це робить діджиталізацію бізнес-процесів найбільш підходящим варіантом для агентств нерухомості задля того, щоб залишатись «на плаву», розвивати бізнес та отримувати від нього прибуток.

Актуальність обраної теми кваліфікаційної роботи полягає у розробці сучасного та зручного рішення питань продажу, покупки та оренди об'єктів нерухомості та автоматизації робочого процесу агентства, шляхом переведення усієї одноманітної та шаблонної роботи в цифровий формат. Таким рішенням може послугувати довідково-інформаційна система, яка дозволить виконувати

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		7

всі операції просто не виходячи з дому та звільнить ріелтору декілька годин часу, які він зможе витратити на те, що справді приносить дохід компанії.

Мета кваліфікаційної роботи – спроектувати та розробити програмний продукт, який дозволить користувачам швидко знаходити, оглядати та домовлятися про оренду чи купівлю опублікованої на сайті нерухомості в будь-якій точці світу, не переживаючи за безпеку себе і своїх даних.

Завдання, які необхідно вирішити для досягнення мети:

- провести дослідження предметної області;
- детально проаналізувати існуючі рішення;
- визначити особливості та специфіку роботи подібних сайтів, їх переваги та недоліки;
- визначити функціональні задачі;
- розробити технічне завдання;
- побудувати архітектуру системи, обрати інструменти та технології, необхідні для розробки;
- розробити сучасний та функціональний інтерфейс;
- виконати програмну реалізацію продукту;
- провести тестування готового продукту.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		8

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Змістовний аналіз предметної області, її структурних та функціональних особливостей

Нерухомість або нерухоме майно – це таке майно, яке має безпосередній зв'язок з землею (земельні ділянки, будівлі, приміщення) та переміщення якого неможливе без нанесення шкоди його вартості або призначенню.

Виділяються такі типи нерухомості:

- житлова нерухомість;
- замиська нерухомість;
- комерційна нерухомість;
- ринок землі.

Об'єкти житлової нерухомості – це дачні та садові будинки, будівлі, записані, відповідно до законодавства, до житлового фонду.

До замиської нерухомості відносять земельні ділянки, приватні будинки, квартири та інші об'єкти, які розташовані за містом або біля міської межі.

Комерційна нерухомість не належить до житлового фонду та використовується у підприємницькій діяльності – для ведення бізнесу та заробляння грошей.

Ринок землі – це ринок, де продають і купують земельні ділянки, частки, здійснюють оренду, викуп.

Основними учасниками ринку нерухомості є:

- продавці – фізичні або юридичні особи, які виступають власниками нерухомості;
- покупці – фізичні або юридичні особи, які бажають придбати нерухомість;
- нотаріуси – особи, які відповідають за реєстрацію нерухомості та заповнення усіх документів, необхідних для проведення операцій з об'єктами

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		9

нерухомості;

– посередники (ріелтори, агентства нерухомості) – надають посередницькі послуги між продавцем і покупцем при проведенні операцій з нерухомістю.

Сучасні агентства нерухомості надають широкий спектр послуг у сфері нерухомості, зокрема:

- купівля-продаж, оренда нерухомості;
- підготовка, отримання і реєстрація повного пакету документів для всіх видів операцій з нерухомістю;
- юридичний супровід угоди;
- реєстрація або оформлення об'єкта в державних чи інших органах;
- консультації з питань укладення угод з нерухомістю;
- консультації з вибору страхової компанії та питань страхування нерухомості;
- оцінка всіх видів нерухомості, консультації з визначення вартості об'єктів.

Незамінними помічниками у організації та автоматизації більшості цих процесів стають сайти нерухомості. Вони допомагають скоротити витрати на маркетинг і просування агентства, підвищують ефективність роботи ріелторів та дають їм змогу фокусуватись на їх основному завданні – продажі.

Так як, за наявності пропозицій в Інтернеті, клієнт сам вибирає для себе найбільш відповідну, у працівника немає більше потреби витратити час на демонстрацію великої кількості квартир або будинків, багато з яких йому можуть взагалі не сподобатися, так як все уже попередньо зображено на фотографіях і розміщено на сторінці деталей про об'єкт.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
						10
Зм.	Арк	№ докум.	Підпис	Дата		

1.2 Аналіз наявного програмно-технічного забезпечення предметної області

Перед створенням довідково-інформаційної системи з нерухомості дуже важливо звернути увагу на особливості функціоналу вже наявних розробок.

Одним з найбільших сайтів з продажу нерухомості є Zillow.com (рисунок 1.1). Він був запущений у 2006 році і на сьогоднішній день є найнадійнішим сайтом для покупців, продавців, орендарів, орендодавців, агентів та фахівців з нерухомості.

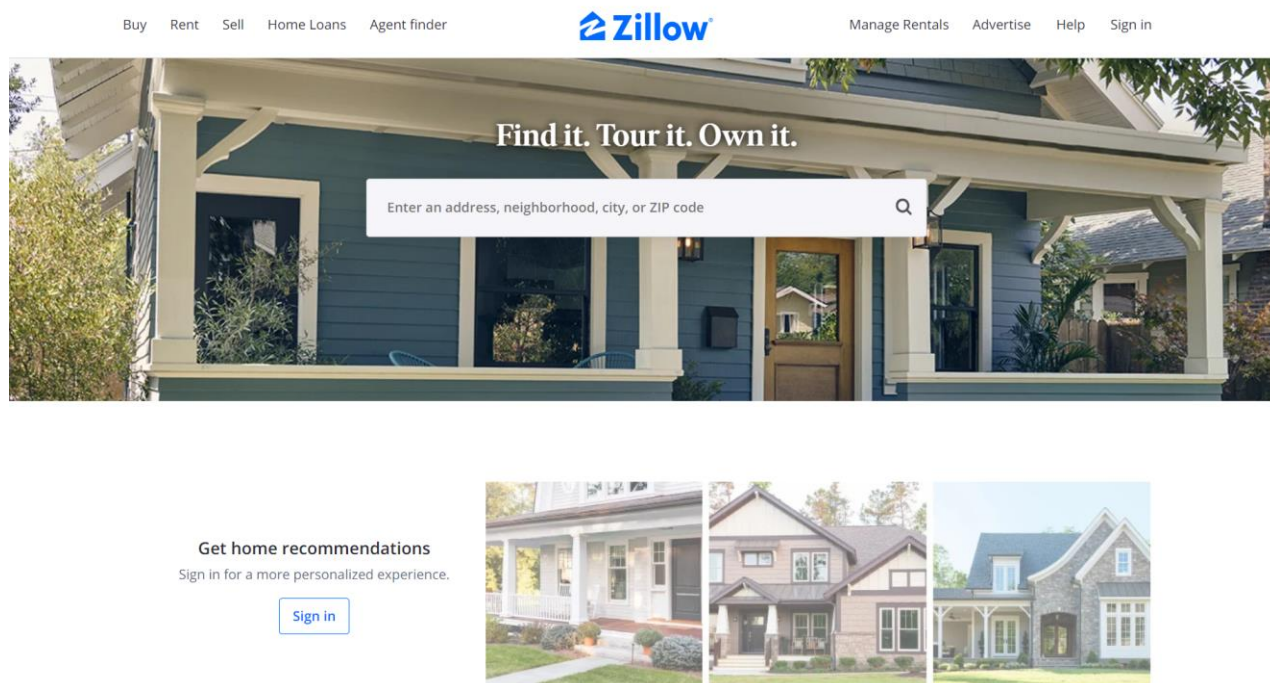


Рисунок 1.1 – Головна сторінка сайту Zillow.com

Zillow.com має продуману систему фільтрації (рисунок 1.2). Покупці можуть максимізувати результати пошуку та мінімізувати свій час, фільтруючи результати за ціною, місцем розташування, характеристиками та типом житла.

Zillow.com надає інформацію про унікальні характеристики житлових об'єктів, безпосередньо пов'язані з районом їх розташування, такі як час на дорогу, статистика району, розташування місцевих шкіл та парків.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		11

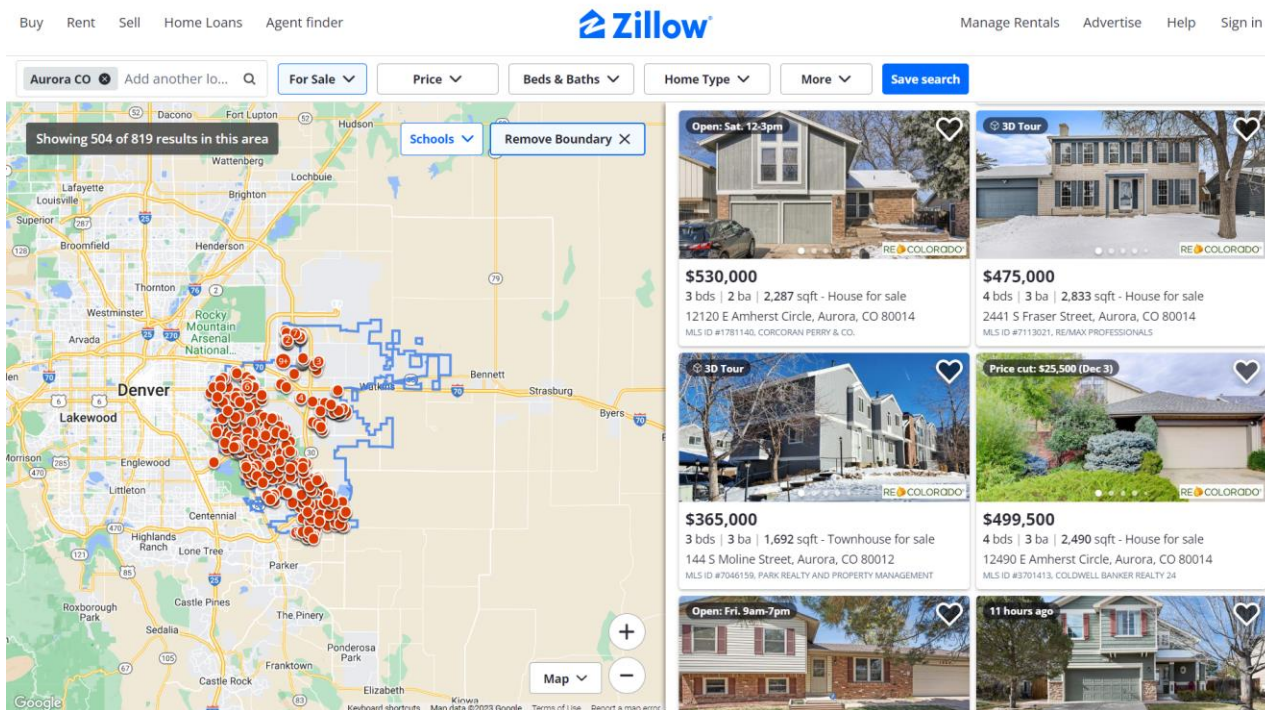


Рисунок 1.2 – Функція пошуку на сайті Zillow.com

Їхній фірмовий інструмент Zestimate показує ціну оголошення, оціночну ринкову вартість та приблизну орендну ставку.

Trulia.com (рисунок 1.3) є дочірньою компанією Zillow.com.

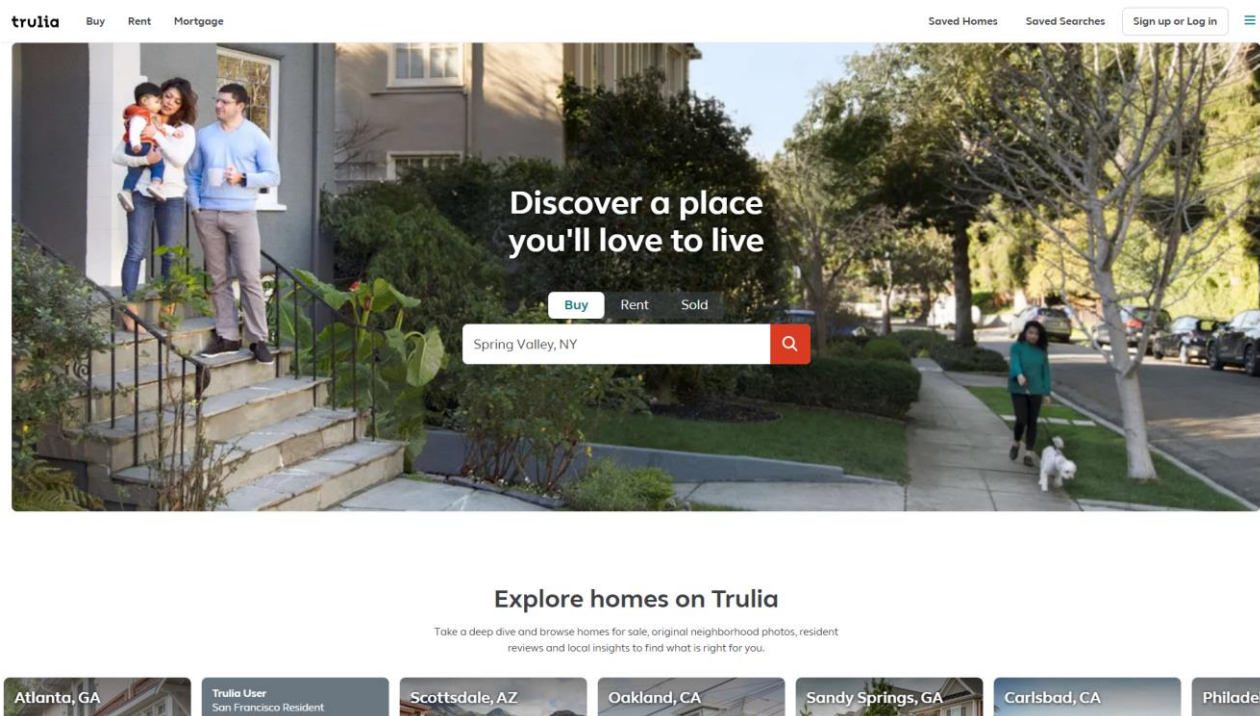


Рисунок 1.3 – Головна сторінка сайту Trulia.com

									Арк.
Зм.	Арк	№ докум.	Підпис	Дата				КВРІПЗ.190131.01.07.ПЗ	12

Як і Zillow.com, Trulia.com дає користувачу можливість досить швидко знайти житло у потрібному регіоні і легко порівняти об'єкти нерухомості.

Вона має розширені фільтри, включаючи звуження пошуку за такими характеристиками, як ціновий діапазон, стиль будинку, кількість спалень тощо.

Trulia.com також пропонує для користування інтерактивну карту з різними фільтрами (рисунок 1.4). Вони покажуть статистику району, розташування шкіл, вікові категорії населення, час на дорогу, ресторани і магазини по сусідству.

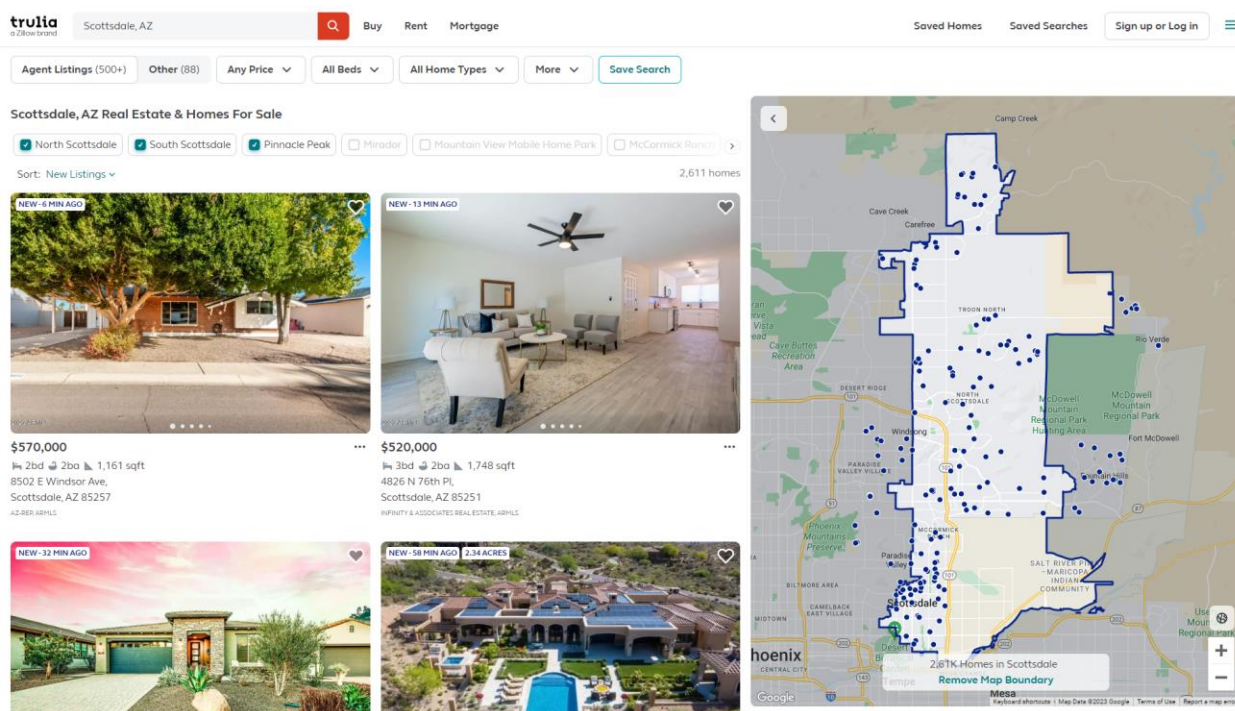


Рисунок 1.4 – Функція пошуку на сайті Trulia.com

Realtor.com (рисунок 1.5) пов'язаний з Національною асоціацією ріелторів. Він має 800 регіональних служб багаторазового розміщення оголошень, що робить його на сьогоднішній день найточнішим сайтом про житло.

Окрім фантастичної точності, Realtor.com має багато статистики та даних, доступних для покупців житла.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		13

Тут є інформація про вартість житла, ринкові прогнози, детальна інформація про школи та підприємства в певному районі, а також час на дорогу. На сайті навіть можна знайти інформацію про рівень злочинності.



New Listings in Spring Valley, NY

[View All 7 New Listings](#)

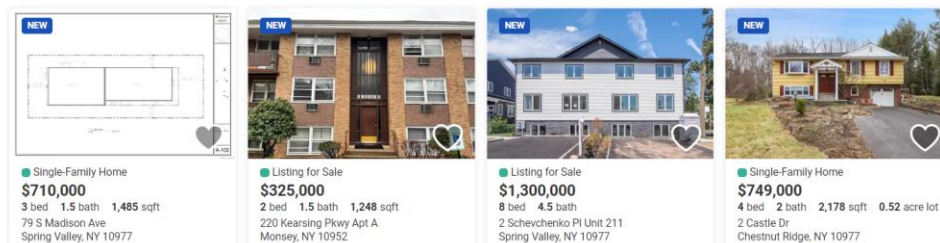


Рисунок 1.5 – Головна сторінка сайту Realtor.com

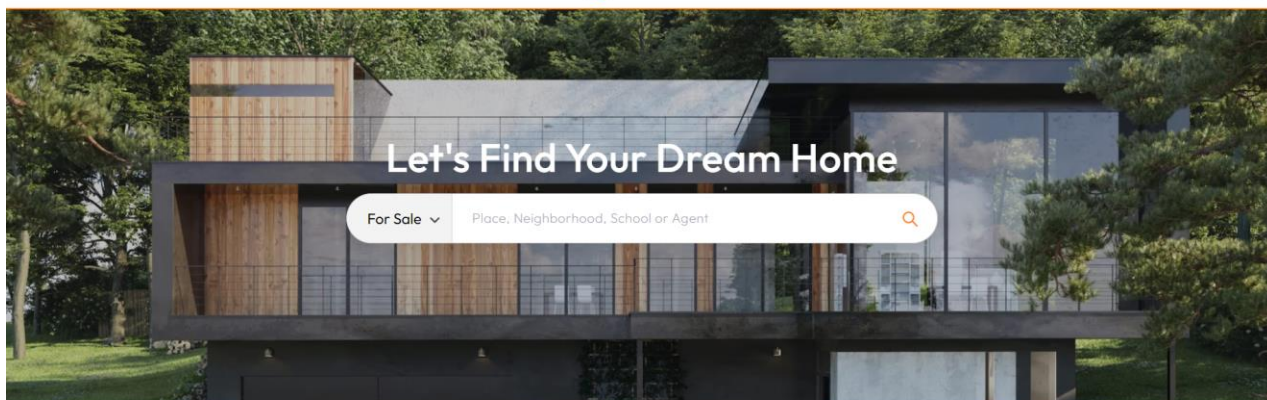
Всі списки даних на сторінках оновлюються в режимі реального часу, тому на сайті завжди відображається найактуальніша інформація.

Також, на Realtor.com немає об'єктів виставлених на продаж напряму від власників, що забезпечує покупців від можливості бути обманутим шахраями.

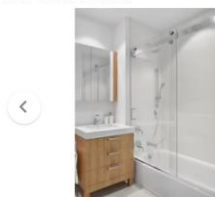
Унікальною особливістю Homes.com (рисунок 1.6) є те, що користувачі можуть бачити, як змінювалася вартість нерухомості. Розуміння вартості нерухомості може дати їм уявлення про те, чи зростає цінність району, чи стоїть на місці, чи знижується.

Також, на Homes.com є функція Snap & Search. Завдяки цьому інструменту користувач може зробити знімок будинку, який йому подобається, та знайти подібні варіанти у вказаній місцевості.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		14



New Construction



Condos



Spacious Homes



Рисунок 1.6 – Головна сторінка сайту Homes.com

Redfin.com (рисунок 1.7) забезпечує покупця можливістю віртуального огляду будинку. Така функція значно економить його час та кошти.

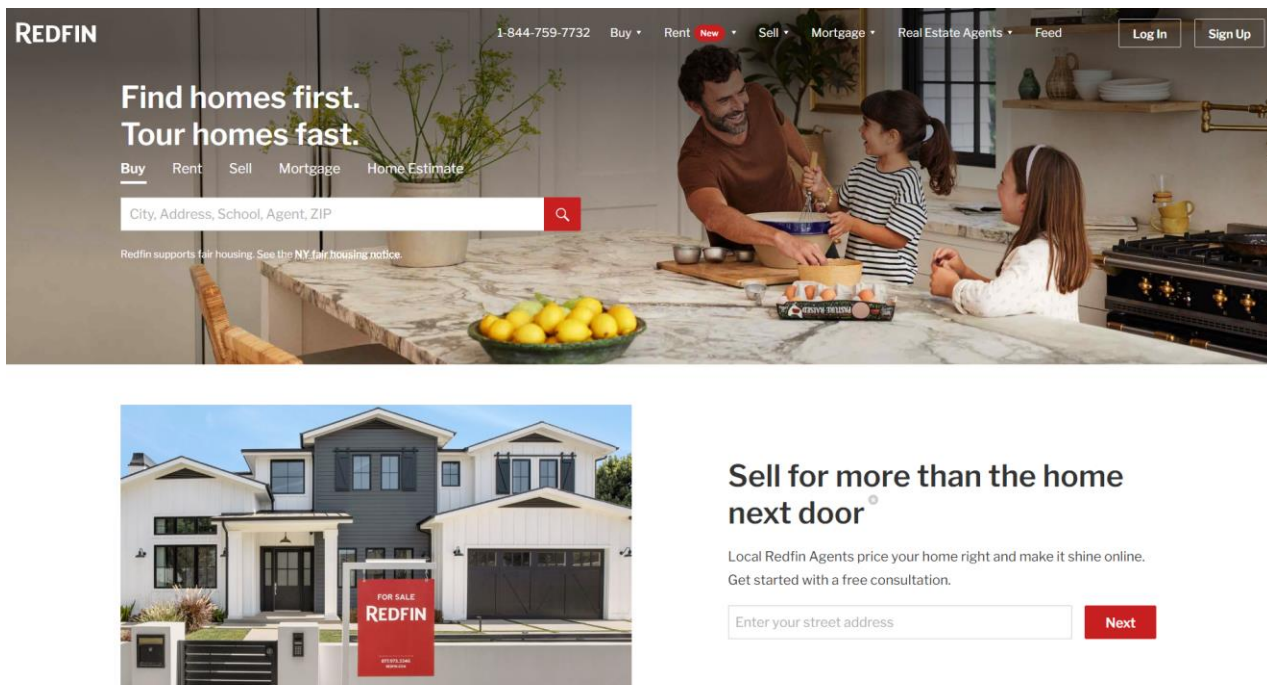


Рисунок 1.7 – Головна сторінка сайту Redfin.com

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		15

Калькулятор Redfin.com (рисунок 1.8) дає користувачу уявлення про те, яку суму він може собі дозволити витратити на житло. Алгоритм базується на оцінці району його проживання, доходу, першого внеску та загального боргу.

REDFIN City, Address, School, Agent, ZIP 1-844-759-7732 Buy • Rent **New** • Sell • Mortgage • Real Estate Agents • Feed Log In Sign Up

Mortgage Calculator

Estimate your mortgage payment, including the principal and interest, taxes, insurance, HOA, and PMI. Add your location for more accurate estimates.

Home price
\$250,000
The amount you plan to offer for a home.

Down payment
\$50,000 20%
Cash you can pay when you close.

Where are you buying?
City, neighborhood, or zip
To calculate local taxes and costs.

Loan type
30 Year Fixed
Affects interest rates. 30- or 15-year loans are standard.

Advanced options

\$1,315 per month
30 Year Fixed, 4.000% Interest ⓘ

- Principal and Interest \$955 (73%)
- Property Taxes \$248 (19%)
- Homeowners' Insurance \$113 (9%)

Add a location to see mortgage rates and providers

Рисунок 1.8 – Калькулятор Redfin.com

HomeFinder.com (рисунок 1.9) схожий на Zillow.com і Trulia.com, він також має широку пропозицію будинків, доступних як для оренди, так і для купівлі. Користувачі можуть застосовувати пошук для фільтрування об'єктів нерухомості за ціною, розміром і типом. HomeFinder.com також дозволяє переглядати нерухомість, яка більше не продається на ринку, що дає користувачам можливість побачити поточні ринкові тенденції в певному регіоні.

Тепер можна провести порівняльний аналіз усіх розглянутих веб-сайтів за різними характеристиками. Перше, на що звертає увагу користувач при відвіданні сторінки сайту – це її загальний вигляд: дизайн інтерфейсу, читабельність тексту, легкість знаходження потрібної інформації.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		16

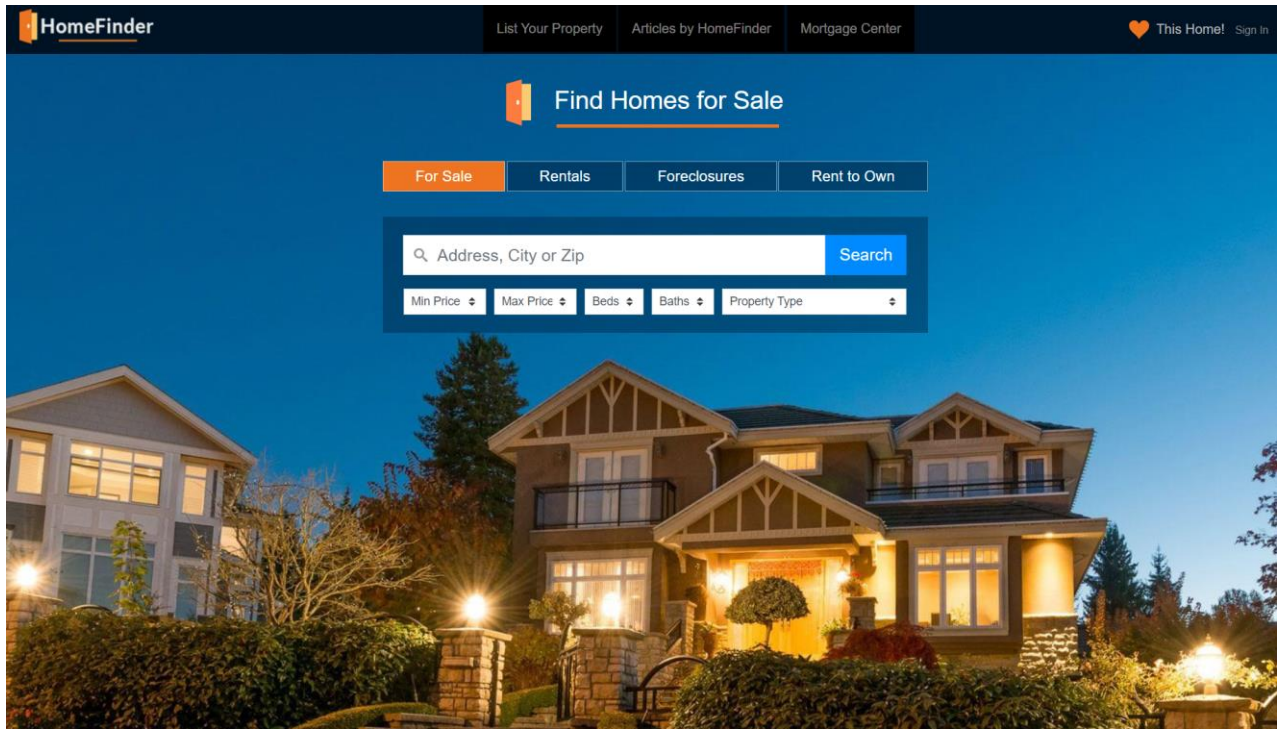


Рисунок 1.9 – Головна сторінка сайту HomeFinder.com

Саме за останній пункт відповідає рядок пошуку. Важливо, щоб у користувача була можливість вказати місцевість, в межах якої він планує здійснювати пошук, а також такі параметри, як кількість спальних місць та ванних кімнат, площа приміщення та точна адреса знаходження об'єкта.

Ці дані обов'язково повинні вказуватись в описі результатів пошуку, щоб користувач міг розуміти, що йому пропонується. Також, результати пошуку повинні містити зображення об'єктів нерухомості в хорошій якості, адже це грає чималу роль у фінальному виборі. Візуально привабливо картинка викликає більше інтересу до себе, ніж нечітка та незрозуміла «пляма».

І останнім параметром є те, без чого користувач не зможе повідомити продавця про свою зацікавленість у пропозиції, – власний аккаунт. Для цього на сайті повинна бути функція авторизації/реєстрації з одним або декількома способами входу, можливістю редагування, поповнення інформації про себе та видалення аккаунту, у випадку якщо користувач більше не захоче, щоб його дані зберігались та могли бути використані модераторами або іншими людьми

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		17

на сайті. У таблиці 1.1 виконано порівняння застосунків за вище описаними критеріями.

Таблиця 1.1 – Порівняльна характеристика існуючих рішень

Сайт	Дизайн користувачького інтерфейсу	Пошук	Результати пошуку	Реєстрація та функціонал особистого кабінету
1	2	3	4	5
Zillow.com	Мінімалістичний, інтуїтивно зрозумілий. Місцями виконаний не зовсім охайно. Заголовки в шапці не відцентровані. Якість основного (і найбільшого) зображення на головній сторінці погана. Через те, що воно розтягнуте по ширині на 100%, здається розмитим. Футгер перенасичений інформацією, мілкий шрифт робить текст важким для читання.	Звичайний, який притаманний більшості профільних сайтів. Потребує введення адреси, сусідства, міста чи коду місцевості. Після введення одного з вище названих параметрів, в правій частині екрану відображаються результати пошуку, а в лівій дублюються у вигляді точок на мапі. Додаються розширені фільтри по ціні, кількості спальних місць, ванних кімнат і т.д.	Містять інформацію про кількість спальних місць, ванних кімнат, «квадратуру» помешкання, ціну, адресу розташування та тип операції, задіяний до даного об'єкту нерухомості (купівля, оренда). Дизайн блоку з інформацією виконаний не дуже добре, блоки знаходяться близько один до одного, через це текст «зливається» воедино.	Реєстрація через електронну пошту, Apple, Facebook та Google. При спробі реєстрації за допомогою Google, сторінка сайту оновлюється, вхід в створений аккаунт не здійснюється. Введення вручну пошти та пароля дозволило створити особистий кабінет, є редагування даних та можливість деактивації аккаунту.
Trulia.com	Trulia.com та Zillow.com схожі у дизайні. Тому, вище перераховані переваги та недоліки можна	Звичайний. Результати пошуку відображаються в лівій частині сторінки, мапа – в правій.	Містять інформацію про ціну, кількість ванних кімнат та спальних місць,	Реєстрація через електронну пошту, Facebook та Google. Є можливість редагування даних

Продовження таблиці 1.1

1	2	3	4	5
	віднести і до даного сайту. Гарно виконана рубрика Explore homes on Trulia.	Колір точок на мапі збігається з кольором меж вибраного регіону, що робить їх нечитабельними. Є розширені фільтри.	розташування об'єкта нерухомості. Дизайн блоку з інформацією виконаний добре, текст читабельний, присутні малюнки.	в особистому кабінеті, немає кнопки видалення аккаунту.
Realtor.com	Сайт має багато мілкого тексту, що робить його важким для сприйняття. Футтер перенасичений інформацією. Багато розміщених фотографій у поганій якості.	Звичайний. Є два режими відображення результатів пошуку: на мапі та у вигляді списку. Другий увімкнений по замовчуванню. Перемикання відбувається по натисканню на відповідну кнопку. Є розширені фільтри та режим малювання на мапі.	Розміщені по 3 в ряд, тому виглядають завеликими, у порівнянні з текстом та іншими частинами сайту. Кольором позначені види операцій, пов'язані з певним об'єктом нерухомості. Одразу є кнопка зв'язку з агентом.	Реєстрація через електронну пошту, Apple, Facebook, Google та посилання на електронну пошту. Є можливість редагування даних та видалення аккаунту.
Homes.com	Шапка сайту не містить лишньої інформації. Все розміщено в прихованій боковій панелі. Нижня частина сторінки перевантажена великою кількістю статей, фотографій і тексту.	Звичайний. Результати пошуку відображаються в лівій частині сторінки, мапа – в правій. Колір точок на мапі збігається з кольором меж вибраного регіону, що робить їх	Відгуки про кожен з об'єктів сильно відволікають увагу від характеристик самого помешкання.	Реєстрація через електронну пошту, прізвище та ім'я користувача, Facebook, Google та Apple. Є можливість редагування даних в особистому кабінеті, немає кнопки

Продовження таблиці 1.1

1	2	3	4	5
		нечитабельними. Є розширені фільтри та режим малювання на мапі.		видалення аккаунту. Перед створенням аккаунту потрібно попередньо підтвердити дію в листі, який приходить на електронну пошту.
Redfin.com	Шапка сайту обтяжена великою кількістю мілких надписів. Текст в деяких частинах сторінки нечитабельний через те, що зливається з фоном. Футгер перенасичений інформацією.	Звичайний. В правій частині екрану відображаються результати пошуку, а в лівій дублюються у вигляді точок на мапі. Додаються розширені фільтри по ціні, кількості спальних місць, ванних кімнат і т.д. Є режим малювання на мапі.	Містять інформацію про ціну, кількість ванних кімнат та спальних місць, розташування об'єкта нерухомості. Дизайн блоку з інформацією виконаний добре, текст читабельний, присутні малюнки.	Реєстрація через електронну пошту, Apple, Facebook, Google та посилання на електронну пошту. Є можливість редагування даних та видалення аккаунту.
Homefinder.com	Простий. Блоки з текстом в центрі сторінки по кольору зливаються з кольором шапки сайту. Деякі частини тексту не відповідають по розмірам загальній темі. Центральна	Одразу розширений. Містить поля для введення бажаної мінімальної та максимальної ціни об'єкту нерухомості, кількості спальних місць та ванних кімнат, типу нерухомості.	Розміщені в дві колонки. Зображення великі, адреси відображаються поверх них, додаткова інформація в списку знизу. Між результатами з'являється	Реєстрація через електронну пошту, Facebook та Google. Є можливість редагування даних в особистому кабінеті, немає кнопки видалення аккаунту.

Кінець таблиці 1.1

1	2	3	4	5
	частина головної сторінки заповнена лише текстом, відсутні будь-які малюнки.	Є мапа з результатами, дизайн простий.	реклама.	

Провівши аналіз існуючих рішень, можна виокремити список характеристик, якими має бути наділена довідково-інформаційна система агентства нерухомості, задля того, щоб досягти успіху та пробитись на верхівку ринку:

- можливість публікації, редагування та видалення пропозицій з продажу та оренди нерухомості;
- наявність зручного пошуку об'єктів нерухомості;
- сучасний та інтуїтивно зрозумілий дизайн інтерфейсу;
- якісний контент та відсутність надлишкової інформації;
- зручна форма авторизації та реєстрації;
- налаштування особистого кабінету;
- можливість видалення профілю;
- наявність панелі адміністратора.

1.3 Визначення вимог до програмного забезпечення та розробка технічного завдання

Проаналізувавши предметну область та уже існуючі рішення, необхідно визначити та описати вимоги до розроблюваного ПЗ.

Для цього використовується мова UML, основною функцією якої є надання можливості візуалізації, побудови та документування складових програм. UML дозволяє показати продукт в роботі із різних ракурсів за допомогою діаграм. При цьому, опис об'єкту відбувається у єдиному

синтаксисі, тому, хоч би де розробник не намалював діаграму, її правила будуть зрозумілими для всіх, хто знайомий із цією графічною мовою.

В UML визначено 15 видів діаграм, які поділяються на два типи: діаграми, які описують структуру, та діаграми, які описують поведінку системи.

Найпростіша з поведінкових діаграм UML – діаграма варіантів використання. Вона є найкращою для вираження функціональних особливостей програми та для людей, які не мають глибоких знань в галузі ІТ.

Ця діаграма застосовується для опису функціональних вимог до програми, її підсистем, або до іншого об'єкта. Вона представляє загальну картину того, як програма буде використовуватись.

У таблицях 1.2 та 1.3 приведені складені описи користувачів системи (акторів) та необхідних їм функцій програмного продукту (варіантів використання).

Таблиця 1.2 – Опис користувачів системи (акторів)

Актор	Короткий опис
1	2
Незареєстрований користувач (гість)	Має можливість переглядати та шукати пропозиції з продажу та оренди нерухомості, статі.
Зареєстрований користувач	Окрім можливостей незареєстрованого користувача, має можливість зберігати пропозиції та пошуки, ставити оцінки, керувати своїм обліковим записом.
Адміністратор, модератор	Окрім дій, які можуть виконувати звичайні користувачі, може розміщувати, змінювати, видаляти публікації на сайті, блокувати та видаляти користувацькі облікові записи, додавати статі.

Як видно з таблиці 1.2, у системі є три актори. До них відносяться зареєстрований, незареєстрований користувачі, адміністратор і модератор. У кожного з них є визначений ряд прав та функцій.

Таблиця 1.3 – Опис функцій (варіантів використання)

Актор	Найменування ВВ	Опис ВВ
1	2	3
Незареєстрований користувач	Реєстрація	Незареєстрований користувач може створити аккаунт, вказавши свою електронну пошту та пароль або дані облікового запису соціальної мережі.
	Пошук пропозицій	Здійснення пошуку нерухомості за вказаними критеріями.
	Перегляд пропозицій	Перегляд результатів пошуку за заданими критеріями.
Зареєстрований користувач	Вхід в аккаунт	Вхід в зареєстрований аккаунт за допомогою одного з методів, перелічених у ВВ Реєстрація.
	Редагування інформації про себе	Додавання фотографії, імені, прізвища, номера телефону і т.д. до свого профілю, а також його видалення.
	Пошук пропозицій	Здійснення пошуку нерухомості за вказаними критеріями.
	Перегляд пропозицій	Перегляд результатів пошуку за заданими критеріями.
	Збереження результатів пошуку	Додавання пропозицій в список збережених, шляхом виставлення позначок вподобання, для того, щоб легше знайти та повернутись до них у майбутньому.
Адміністратор, модератор	Додавання пропозицій та іншої інформації	Створення нових публікацій (додавання інформації про об'єкти нерухомості, їхні фотографії, ціни) та розміщення їх на сторінках сайту.
	Видалення пропозицій	Видалення неактуальних, куплених/орендованих

Кінець таблиці 1.3

1	2	3
		об'єктів з сторінок сайту.
	Редагування пропозицій	Оновлення даних про об'єкти.
	Блокування та видалення користувацьких записів	Блокування та видалення аккаунтів, що порушують правила користування платформою, або є неактивними певний період часу.

Відповідно до описаних в таблицях акторів та варіантів використання побудовано діаграму, яка зображена на рисунку 1.10.

Технічне завдання на розробку довідково-інформаційної системи подано у додатку А.

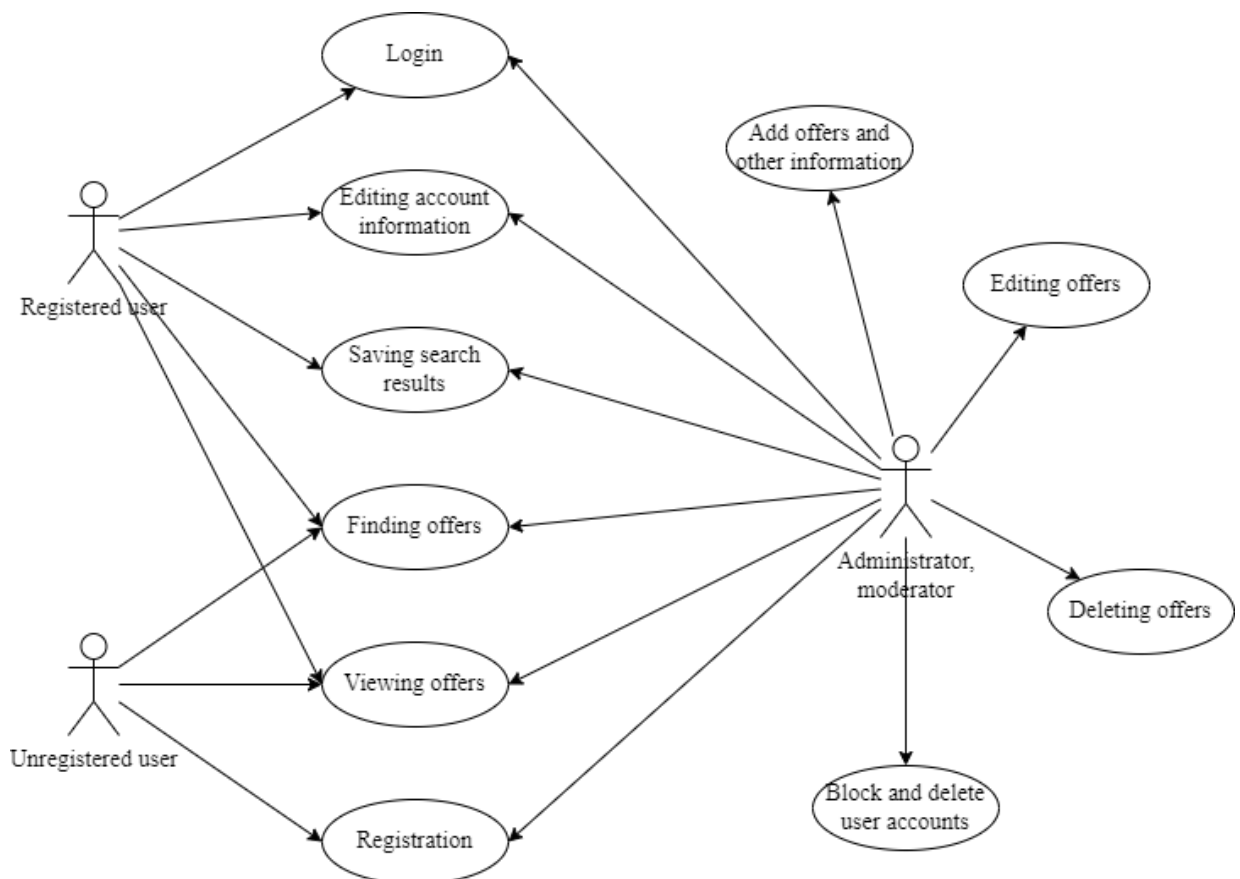


Рисунок 1.10 – Діаграма варіантів використання

Отже, у даному розділі було здійснено дослідження предметної області, її структурних та функціональних особливостей, проаналізовано існуючі рішення у профільній галузі, проведено і занесено у таблиці їх порівняльну характеристику та визначено вимоги до розроблюваного програмного забезпечення, які розгорнуто описано у технічному завданні.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		25

2 ПРОЕКТУВАННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Проектування архітектури та структури системи

Клієнт-серверна архітектура – архітектура, яка лежить в основі більшості соціальних мереж, сайтів електронної комерції, мобільних застосунків, пристроїв Інтернету речей і т.п. Лише з назви цієї архітектури можливо визначити два основних компоненти з яких вона складається – клієнт та сервер.

Клієнтом виступає особа або група осіб, організація або комп'ютер. Усі вони користуються певною послугою або набором послуг, які надає сервер, а також приймають вхідну інформацію.

Під поняттям сервера розуміється віддалений пристрій, який розміщує, надає та управляє більшістю ресурсів і послуг, що споживаються клієнтом. Зазвичай сервери є фізичними, наприклад, стійкові сервери, та віртуальними. І ті, і ті часто застосовуються у сьогоденні.

Мережа – ще один не менш важливий елемент клієнт-серверної архітектури, який виступає зв'язком між клієнтом та сервером і забезпечує процес обміну інформацією.

Загальний принцип роботи клієнт-серверної архітектури (рисунк 2.1) полягає у тому, що клієнт надсилає серверу запит на отримання даних. Сервер приймає і обробляє запит та надсилає пакети даних назад користувачеві, який їх потребує.

Архітектура клієнт-сервер може змінюватись, залежно від кількості рівнів, які вона містить.

Дворівнева архітектура використовується для опису клієнт-серверних систем, де клієнт робить запит, а сервер відповідає на нього, використовуючи безпосередньо власні ресурси. Це означає, що сервер не звертається до інших застосунків для надання частини послуг. Дворівнева архітектура підходить для простих систем, що працюють з малими об'ємами інформації.

Перевантаження серверу може призвести до її втрати.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		26

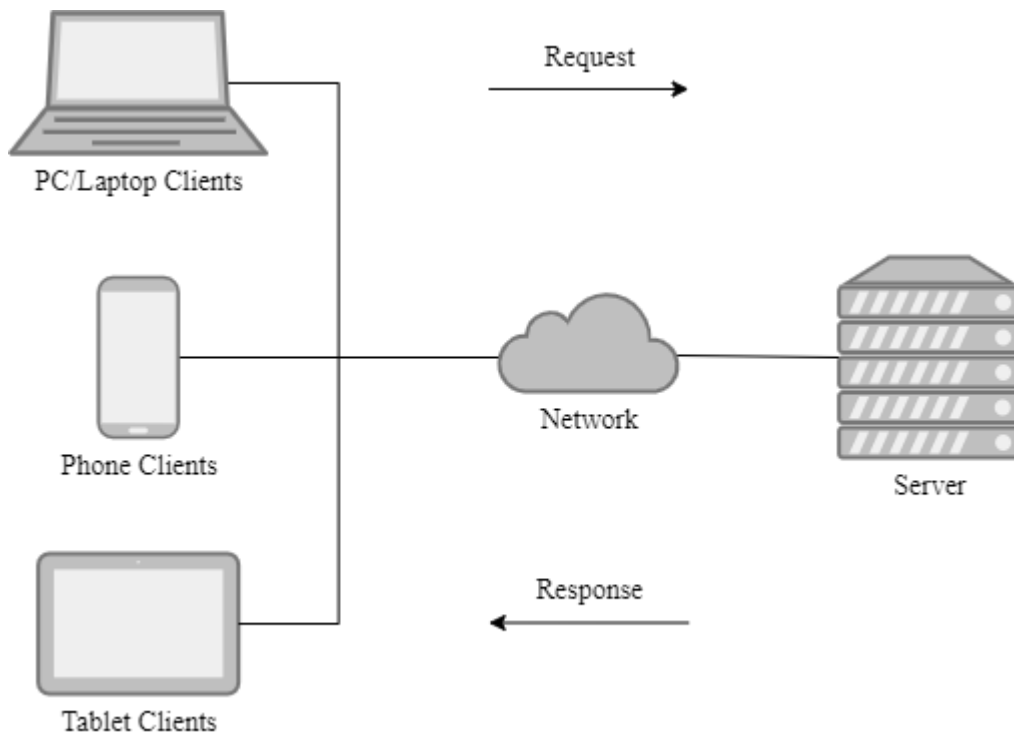


Рисунок 2.1 – Принцип роботи клієнт-серверної архітектури

Задля уникнення випадку втрати даних, надійним варіантом є зберігання інформації в базах даних, що робить дворівневу архітектуру трирівневою.

Трирівнева архітектура поділяє систему на три рівні:

- рівень представлення, або інтерфейс користувача;
- рівень застосунку, де обробляються дані;
- рівень даних, де зберігаються та управляються дані, пов'язані з застосунком.

Головна перевага трирівневої архітектури полягає в тому, що, оскільки кожен рівень працює на власній інфраструктурі, він може розроблятися одночасно окремою командою розробників і може бути оновлений або масштабований за потреби без впливу на інші рівні.

Протягом десятиліть саме трирівнева архітектура була переважальною архітектурою для клієнт-серверних застосунків. Тому, для реалізації довідково-інформаційної системи агентства нерухомості вибрано трирівневу клієнт-серверну архітектуру.

MVC (Model-View-Controller) – це шаблон для структурування застосунку на стороні сервера та означає логічне розділення його на 3 рівні (рисунок 2.2): модель, яка відповідає за зберігання даних, що мають відношення до застосунку, представлення, яке відповідає за представлення цих даних, і контролер, який є обробляє дані та забезпечує функціонал.

Перевагами шаблону є:

- зменшення залежностей між частинами застосунку та, водночас, збільшення його гнучкості;
- розподілення відповідальностей при розробці, що сприяє полегшенню роботи над програмним продуктом;
- досить легке тестування, так як кожен компонент тестується незалежно один від одного.

Такий шаблон буде використано при організації коду у процесі написання довідково-інформаційної системи.

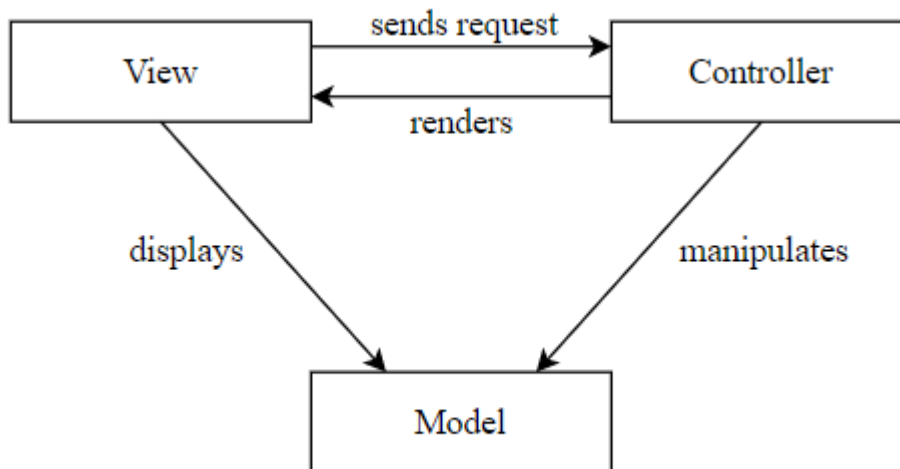


Рисунок 2.2 – Складові шаблону MVC

Далі, необхідно провести розбиття системи на модулі, відповідно до тих задач, які вона має виконувати.

Модуль – незалежна, завершена та функціонуюча частина програми, оформлена у вигляді файлу з кодом та призначена для використання в

програмах. Модулі формують структуру програмного продукту, дають змогу уникнути ділянок коду, що повторюються, зменшують розмір застосунку та пришвидшують його роботу.

Добре спроектований модуль повинен володіти наступними властивостями:

– незалежність результату роботи модуля від роботи інших модулів, але залежність від вхідних даних;

– слабкі інформаційні зв'язки та, по можливості, мінімізований обмін інформацією з іншими модулями;

– отримання на вході певного набору даних, їх опрацювання і повернення одного набору результатних даних, тобто реалізація стандартного принципу IPO – вхід-процес-вихід;

– повноцінна реалізація модулем однієї функції, самостійне (без допомоги додаткових засобів) виконання повного набору операцій для реалізації своєї функції.

Розподіл на модулі найкраще проводити виходячи з тих завдань, які вирішує система. Основне завдання розбивається на складові підзавдання, які можуть виконуватися автономно, незалежно одне від одного.

Якщо декомпозицію на модулі виконано правильно, то взаємодія з іншими модулями (що відповідають за інші функції) буде мінімальною. Вона може бути, але її відсутність не повинна бути критично важливою.

Кожен модуль має відповідати за розв'язання якогось підзавдання та виконувати відповідну йому функцію, а також характеризуватись набором даних, необхідних йому для виконання його функції.

Таким чином, грамотна декомпозиція ґрунтується насамперед на аналізі функцій системи і необхідних для виконання цих функцій даних.

У додатку Б «Технічне завдання», пункті «Вимоги до функціональних характеристик» було описано, які функції повинна виконувати довідково-інформаційна система. Виходячи з цього можна виділити наступні модулі системи:

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		29

- модуль відображення та сортування інформації, який відповідальний за виведення сторінок і представлення інформації користувачам;
- модуль для роботи з контентом (додавання, редагування та видалення пропозицій з продажу та оренди нерухомості);
- модуль пошуку об'єктів нерухомості;
- модуль авторизації та реєстрації користувача;
- модуль керування аккаунтами;
- модуль замовлення дзвінка.

На рисунку 2.3 зображено всі модулі системи та зв'язок між ними:

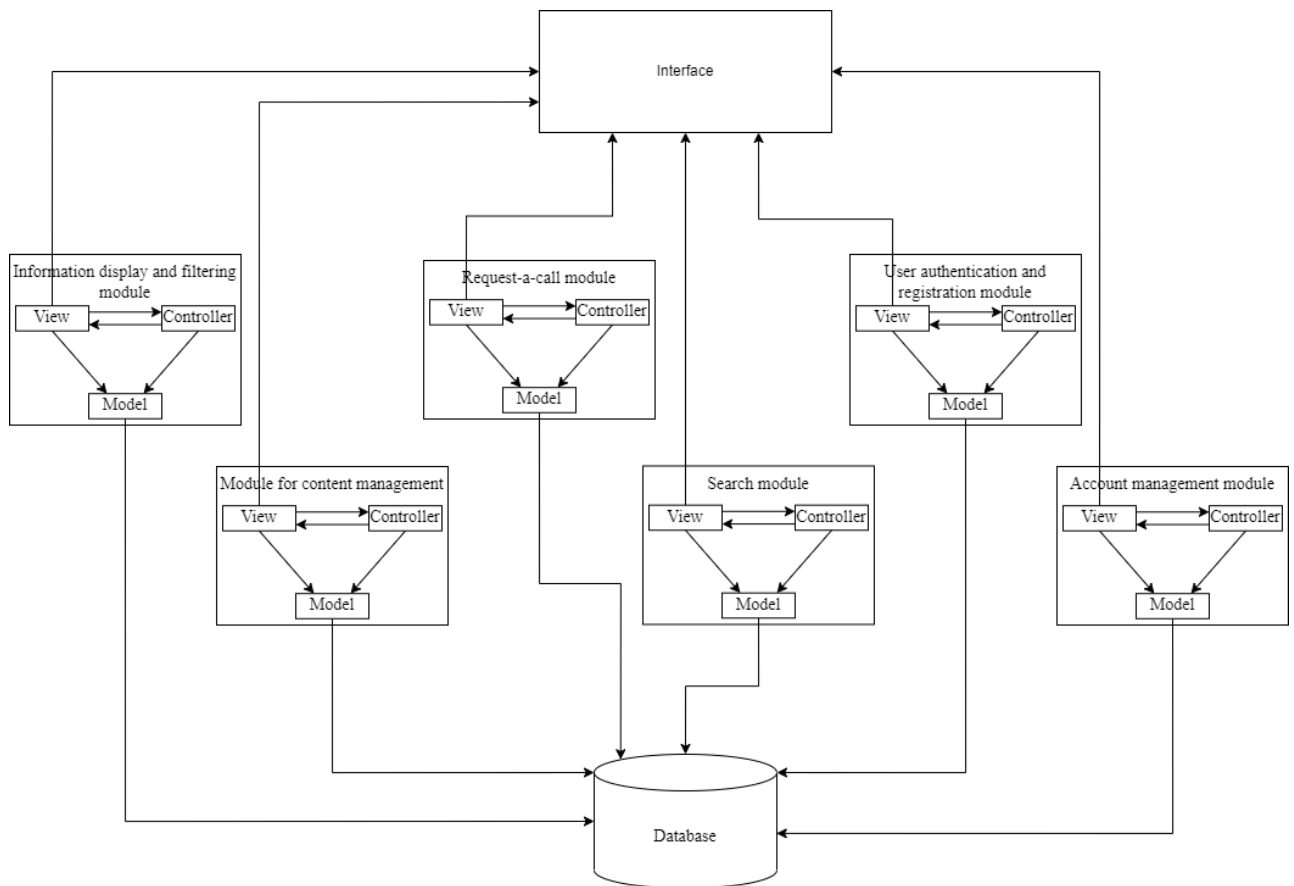


Рисунок 2.3 – Модульна структура ДІС агентства нерухомості

2.2 Проектування логічної моделі бази даних

Перш ніж створювати логічну модель БД необхідно уніфікувати та

стандартизувати її складові. Для цього потрібно:

- визначити перелік сутностей, інформація про які зберігатиметься у БД, навести їх опис;
- визначити перелік атрибутів сутностей, навести їх опис;
- визначити зв'язки між сутностями БД, навести їх опис;
- визначити типи зв'язків та необхідні обмеження;
- визначити первинні ключі для кожного об'єкта БД;
- виконати нормалізацію даних (до третьої нормальної форми).

Сутність в базі даних – це будь-який об'єкт, який можна виділити виходячи з суті предметної області для якої розробляється база даних. Сутність може бути особою, місцем, подією та/або об'єктом, що має відношення до даної системи. У фізичній моделі бази даних сутність називається таблицею.

На основі проведеного аналізу для довідково-інформаційної системи агентства нерухомості можна виділити наступні сутності (таблиця 2.1):

Таблиця 2.1 – Перелік сутностей

Сутність	Опис
1	2
Property	Нерухомість, розташована за вказаною адресою, має певну площу, визначену кількість ванних кімнат та спальних місць, ціну, номер телефону власника або орендодавця.
Category	Категорія, до якої відноситься нерухомість: квартира/будинок/ділянка.
Address	Місто, країна, штат, вулиця, будинок, номер квартири – точна адреса, за якою розташовується об'єкт.
Operation	Призначення об'єкту нерухомості: для оренди/для продажу.

Атрибут – властивість, риса або характеристика, що описує сутність. Сутність може мати стільки атрибутів, скільки необхідно. Для кожного атрибуту визначається тип даних, їхній розмір, допустимі значення та обмеження цілісності. У фізичній моделі БД атрибути називають стовпцями.

Обмеження цілісності використовуються для зазначення правил для даних у таблиці, а саме для обмеження типів даних, які можуть бути поміщені в неї. Це забезпечує точність і достовірність даних і, у випадку, якщо існує будь-яке порушення між обмеженням і дією даних, дія переривається.

Обмеження можуть застосовуватися на рівні стовпців та на рівні таблиці. Обмеження на рівні стовпця застосовуються тільки до одного стовпця, тоді як обмеження на рівні таблиці застосовуються до всієї таблиці.

Найбільш часто до даних застосовуються наступні обмеження:

- NOT NULL – гарантує, що стовпець не матиме значення NULL;
- DEFAULT – задає значення за замовчуванням для стовпця, у випадку, якщо воно не вказане;
- UNIQUE – уніфікує усі значення в стовпці, запобігає появі повторень;
- PRIMARY KEY – унікально ідентифікує рядок/запис в таблиці;
- FOREIGN KEY – унікально ідентифікує рядок/запис у будь-якій іншій таблиці.

Первинний ключ (або PRIMARY KEY, PK) має певні особливості, а саме:

- в одній таблиці бази даних може бути тільки один первинний ключ;
- рядки, які визначені первинним ключем, не можуть бути пустими або невизначеними;
- значення, в рядку, визначеному, як первинний ключ, не повинно змінюватись.

Зовнішній ключ (або FOREIGN KEY, FK) використовується для зв'язування двох таблиць та посилається на первинний ключ іншої сутності. Причому, поля, що назначені ключами, у обох таблицях повинні мати один і той самий тип даних.

У таблицях 2.2 – 2.5 описано атрибути кожної сутності:

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		32

Таблиця 2.2 – Опис атрибутів сутності «Property»

Атрибут	Опис
1	2
PropertyId	Унікальний ідентифікатор об'єкту нерухомості.
AddressId	Ідентифікатор адреси, за якою розташований об'єкт нерухомості.
Area	Площа приміщення.
BedNumber	Кількість спальних місць в приміщенні.
BathNumber	Кількість ванних кімнат в приміщенні.
Description	Опис об'єкту нерухомості.
Image	Зображення об'єкту нерухомості.
Price	Ціна на об'єкт нерухомості.
PhoneNumber	Номер телефону власника або орендодавця.
CategoryId	Ідентифікатор категорії, до якої належить об'єкт нерухомості.
OperationId	Ідентифікатор операції, яка проводиться відносно об'єкту нерухомості.

Таблиця 2.3 – Опис атрибутів сутності «Address»

Атрибут	Опис
1	2
AddressId	Унікальний ідентифікатор адреси, за якою розташований об'єкт нерухомості.
State	Штат, в якому розташований об'єкт нерухомості.
City	Місто, в якому розташований об'єкт нерухомості.
Country	Країна, в якій розташований об'єкт нерухомості.
Street	Вулиця, на якій розташований об'єкт нерухомості.
Unit	Номер об'єкту нерухомості.

Таблиця 2.4 – Опис атрибутів сутності «Category»

Атрибут	Опис
1	2
CategoryId	Унікальний ідентифікатор категорії, до якої відноситься об'єкт нерухомості.
PropertyCategory	Назва категорії.

Таблиця 2.5 – Опис атрибутів сутності «Operation»

Атрибут	Опис
1	2
OperationId	Унікальний ідентифікатор операції, яка здійснюється по відношенню до об'єкту нерухомості.
OperationType	Тип операції.

Між двома сутностями може бути встановлений зв'язок. Це дає змогу за однією сутністю знаходити інші сутності, пов'язані з нею.

Розрізняють наступні 3 типи зв'язків:

– «один-до-одного». У цьому зв'язку один екземпляр першої сутності може бути пов'язаний лише з одним екземпляром другої сутності і навпаки;

– «один-до-багатьох» (найпоширеніший тип зв'язку в базі даних). Такий зв'язок означає, що кожен екземпляр першої сутності може бути пов'язаний з декількома екземплярами другої сутності. Це також означає, що кожен екземпляр другої сутності може бути пов'язаний лише з одним екземпляром першої сутності;

– «багато-до-багато». Означає, що кожен екземпляр першої сутності може бути пов'язаний з більш ніж одним екземпляром другої сутності. Це також означає, що кожен екземпляр другої сутності може бути пов'язаний з декількома екземплярами першої сутності.

Нормалізація – це процес організації даних у БД, який включає створення таблиць і встановлення відносин між ними відповідно до правил, що забезпечують захист даних і роблять базу даних більш гнучкою, усуваючи надмірність і неузгоджені залежності.

Існує кілька правил нормалізації баз даних:

- ненормалізована форма або нульова нормальна форма;
- перша нормальна форма;
- друга нормальна форма;
- третя нормальна форма;
- нормальна форма Бойса-Кодда;

- четверта нормальна форма;
- п'ята нормальна форма;
- доменно-ключова нормальна форма;
- шоста нормальна форма.

База даних вважається нормалізованою, якщо вона перебуває як мінімум у третій нормальній формі. У реальному світі нормалізація до третьої нормальної форми є звичайною, стандартною практикою, тому що третя нормальна форма усуває достатню кількість аномалій, водночас продуктивність бази даних, а також зручність її використання не знижується, що не можна сказати про всі наступні форми.

Вимога першої нормальної форми полягає в тому, щоб:

- у таблиці не було рядків, що повторюються;
- у стовпці зберігалися дані одного типу;
- були відсутні масиви та списки в будь-якому вигляді.

Щоб база даних перебувала в другій нормальній формі, необхідно, щоб:

- її таблиці вже перебували в першій нормальній формі;
- усі таблиці мали свій ключ;
- усі неключові стовпці таблиці залежали від повного ключа (у разі якщо він складний).

Якщо ключ складний, тобто складається з декількох стовпців, то всі інші неключові стовпці мають залежати від усього ключа, тобто від усіх стовпців у цьому ключі. Якщо якийсь атрибут (стовпець) залежить тільки від одного стовпця в ключі, значить, база даних не перебуває в другій нормальній формі.

Щоб нормалізувати базу даних до третьої нормальної форми, необхідно зробити так, щоб у таблицях були відсутні неключові стовпці, які залежать від інших неключових стовпців.

Провівши нормалізацію бази даних до третьої нормальної форми, було створено логічну модель (рисунок 2.4):

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		35

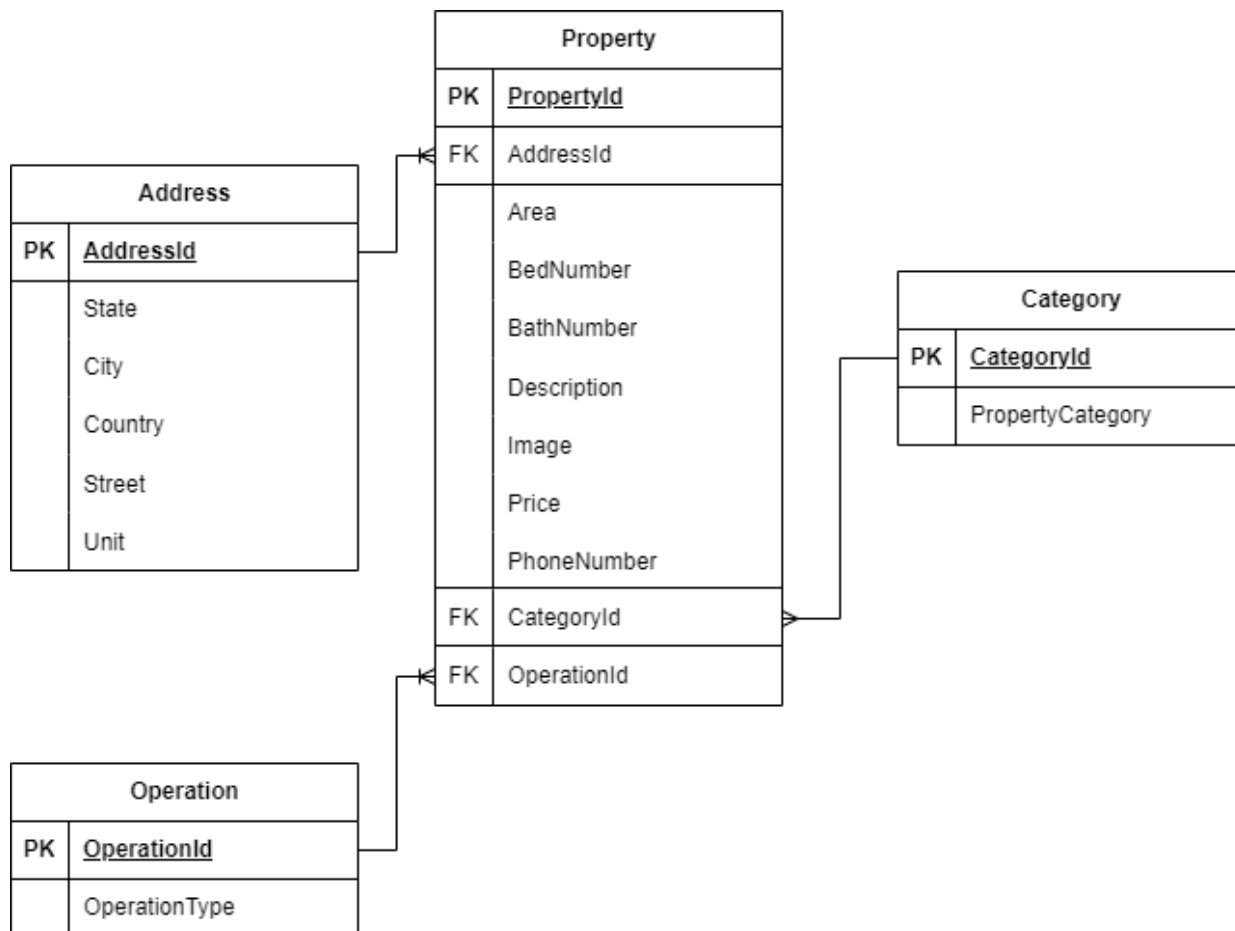


Рисунок 2.4 – Логічна модель бази даних агентства нерухомості

2.3 Проектування інтерфейсу користувача

Користувацький інтерфейс часто розуміють тільки як зовнішній вигляд програми. Однак через взаємодію з інтерфейсом у людини складається враження про продукт загалом. Розробка користувацького інтерфейсу об'єднує в собі безліч елементів і компонентів, які здатні впливати на реалізацію програмного продукту. А ще інтерфейс впливає на те, чи буде продукт приносити гроші та привертати інтерес аудиторії.

Процес проектування користувацького інтерфейсу – це робота з певною послідовністю. Завдання розробки інтерфейсу: зробити його корисним, інтуїтивно зрозумілим і максимально простим.

Розробка користувацького інтерфейсу поділяється на кілька етапів:

– збір інформації. Розробка ТЗ, визначення термінів, цільової аудиторії, особливостей продукту і т.д.;

– дослідження. Вивчення ринку, конкурентів і вподобань цільової аудиторії, збір референсів – прикладів вдалих інтерфейсів продуктів із тієї самої предметної області або для тієї самої цільової аудиторії. Це допоможе зрозуміти, від чого відштовхуватися і в якому напрямку рухатися;

– чорновий макет. Використання матеріалів, знайдених під час дослідження, і перевірка їх в роботі на практиці;

– готовий макет або прототип – фінальна версія дизайну інтерфейсу, затверджена замовником;

– розробка, тестування і запуск продукту веб-розробниками, тестувальниками та авторами контенту. На цьому етапі розробник інтерфейсу підключається, тільки тоді, коли потрібно детально пояснити, як саме має працювати той чи інший елемент.

Збір інформації та дослідження уже були проведені, їх результати описані у пунктах раніше. На даному етапі потрібно розробити чорновий макет інтерфейсу довідково-інформаційної системи.

На рисунку 2.5 зображений спрощений вигляд сторінок та розміщення на них основних елементів керування. У лівій частині продемонстровано варіант для звичайного користувача, якому надані права user'a, у правій – для користувача з правами адміністратора (admin).

Такі елементи, як блоки перегляду об'єктів нерухомості та категорій, форми замовлення виклику від власника, форми реєстрації і авторизації користувача матимуть однаковий дизайн та оформлення для відвідувачів сайту, незалежно він наданих їм привілеїв. Для адміністратора ж буде додано сторінки управління ролями та редагування розміщеного контенту.

Відносно стилістики сторінок, при розробці дизайну інтерфейсу вирішено використовувати наступну.

Колір. Так як, в основному, вся увага користувача буде зосереджена на сторінках з доступними пропозиціями, колір яких буде дуже різноманітним та

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		37

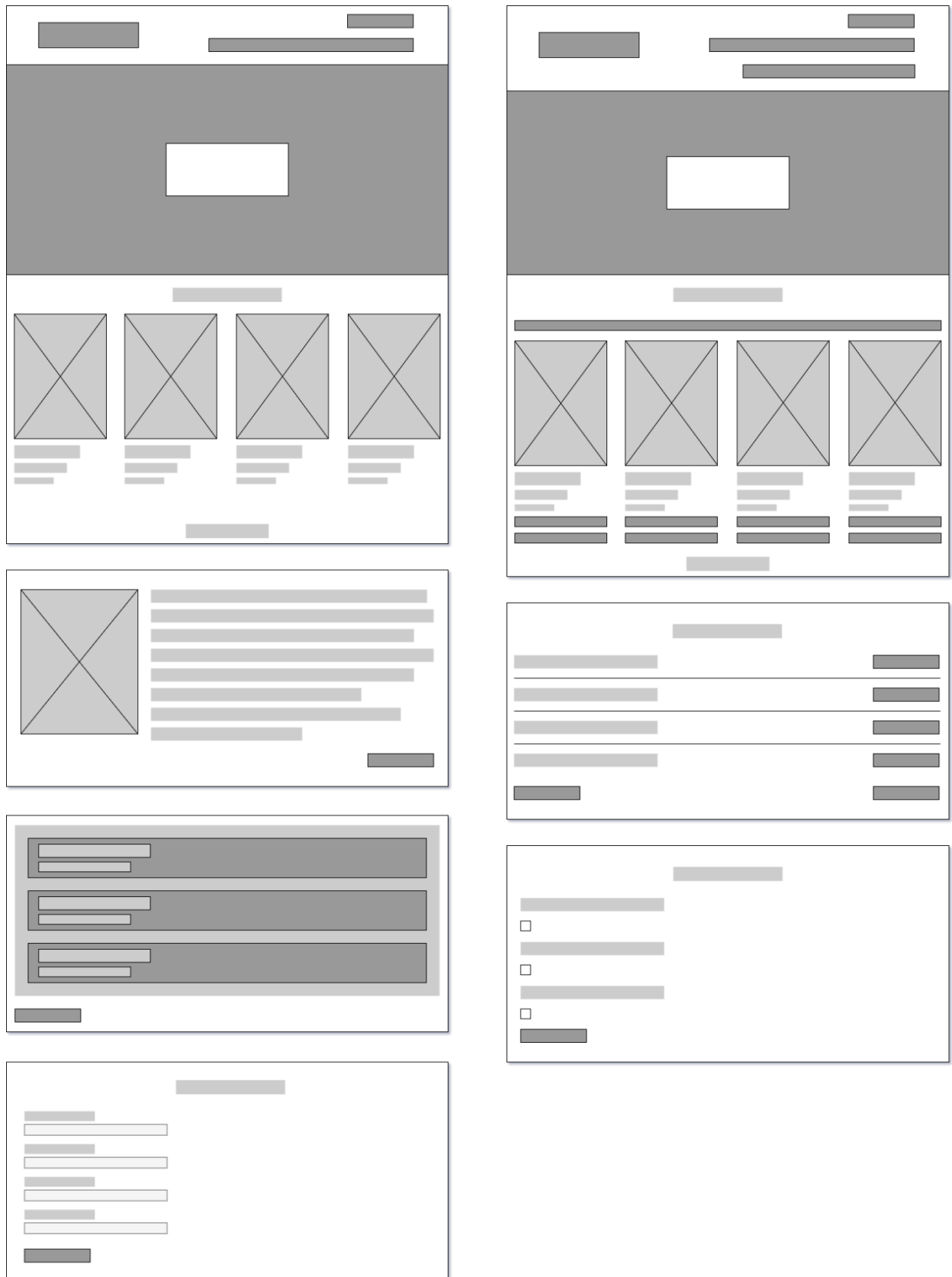


Рисунок 2.5 – Чорновий макет сторінок довідково-інформаційної системи

залежати від кольору інтер'єру приміщення, яке здається в оренду або виставлене на продаж, щоб не перевантажувати сторінки, в якості фону взято

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		38

білий. У шапці сторінки буде розміщений банер з деякою інформацією, акцентні кольори будуть підбиратись таким чином, щоб поєднуватись його з кольоровою гаммою. Блок навігації буде лежати поверх банеру та мати чорний колір з деяким відсотком прозорості, щоб сильно не виділялись на фоні інших об'єктів та гармонійно вписуватись у загальний вигляд.

Шрифти. 'Source Serif Pro', serif використаний на всіх сторінках сайту, включно з полями введення інформації. Там, де потрібно підкреслити певну інформацію додано більшу жирність.

Форми. Фотокарткам, кнопкам та полям введення надана прямокутна форма із заокругленими кутами, щоб вони візуально виглядали м'якше та приємніше для ока користувача.

Навігаційне меню, доступ до якого є з кожної сторінки, складається з контактної інформації агентства нерухомості: сторінок в соціальних мережах та номеру телефону, та чотирьох пунктів меню: пропозиції, категорії, логін та реєстрація.

Скорочена інформація про нерухомість на головній сторінці складається з вказаних місця розташування об'єкту, кількості ванних та спальних кімнат та ціни, повна – доповняється додатковим описом нерухомості, району, в якому вона розташована і т.д.

Форми логіну та реєстрації потребують вказання електронної адреси і/або власного імені та паролю користувача.

Сторінки управління ролями дозволяють додавати, видаляти та змінювати ролі для зареєстрованих користувачів. Аналогічно зі сторінками управління пропозиціями. Редагуванню та додаванню підлягають усі дані про об'єкт, що описують сутність «Property», окрім унікального ідентифікатора. Його зміна чи видалення може порушити роботу та цілісність бази даних.

Правильно створена структура – фундамент сайту. Нею є сукупність сторінок, розташованих з урахуванням ієрархії. Структура має давати можливість користувачу здійснювати зручний перехід від загальних тем до конкретної інформації, за якою він прийшов. Від того, наскільки правильно і

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		39

якісно розроблена структура, залежить успіх взаємодії користувачів і пошукових роботів із сайтом. Існують наступні види структур:

– лінійна (розташування сторінок у вигляді ланцюжка). У цій структурі немає ієрархії і всі сторінки рівноправні. Вони проглядаються одна за одною, починаючи з першої і закінчуючи останньою. Використовуючи таку структуру, важливо, щоб на сторінках стояли посилання одна на одну. Лінійна структура часто використовується для сайтів-візиток, сайтів-презентацій, лендінгів та інших;

– лінійна структура з розгалуженнями. Працює за тим самим принципом, що і лінійна, проте має відгалуження в межах ланцюжка. Таку структура, як правило, мають невеликі корпоративні сайти, авторські блоги, онлайн-книжки, лендінги, сайти-візитки та інші;

– блокова (решітчаста) структура сайту. У цій структурі всі сторінки рівноправні, за винятком головної. Усі блоки розташовані поруч один з одним і посилаються на кілька інших. Блокову структуру зазвичай використовують для конкретної послуги або продукту, де кожна сторінка описує характеристики або вид послуги;

– деревоподібна (ієрархічна) структура сайту. Найпопулярніший і оптимальний варіант для більшості сайтів, коли кожна наступна сторінка – частина попередньої. Універсальність деревоподібної структури дає змогу застосовувати її для сайтів різних типів. Вона використовується для інтернет-магазинів, інформаційних ресурсів, комерційних сайтів, порталів тощо.

Остання і буде використана у процесі розробки довідково-інформаційної системи, так як, є найбільш підходящою, адже доступ до всіх сторінок сайту здійснюється, в основному, за допомогою посилань в шапці. Нерідко, з відкритих сторінок відбувається перехід на дочірні сторінки, які і створюють собою ієрархію, що лежить в основі даного виду структури.

На рисунку 2.6 зображено логічну структуру веб-сайту довідково-інформаційної системи агентства нерухомості:

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		40

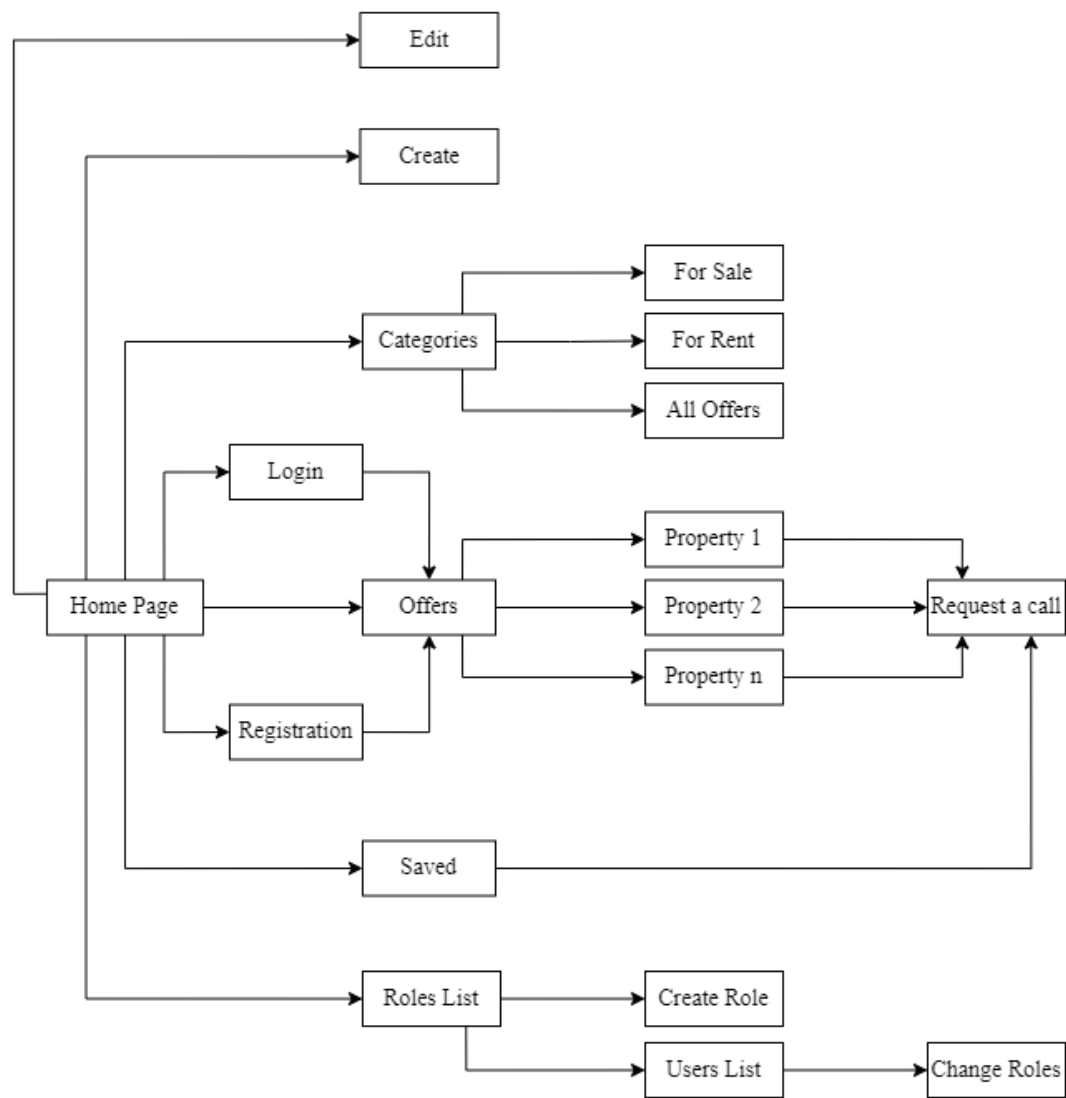


Рисунок 2.6 – Логічна структура ДІС

2.4 Аналіз та вибір технологій і методів реалізації системи

В основі довідково-інформаційної системи лежатиме шаблон проектування MVC (Model-View-Controller). За допомогою нього застосунок буде розділено на три основні логічні компоненти: модель, вид і контролер. Контролер отримуватиме усі запити до застосунку і даватиме вказівку моделі підготувати будь-яку інформацію, необхідну для представлення. Представлення використовуватиме дані, підготовлені контролером, для відображення кінцевого результату.

Для створення моделі (інтерфейсу) довідково-інформаційної системи використовуватимуться:

- мова розмітки HTML (Hyper Text Markup Language). За її допомогою буде розроблятися структура веб-сторінок;

- механізм додавання стилів CSS, який дозволить описати оформлення веб-сторінок, включно з кольорами, розміщенням елементів і шрифтами та адаптувати сторінку до різних типів пристроїв, а саме великих і малих екранів;

- мова програмування JavaScript, для надання інтерактивності веб-сторінкам та створення більш динамічної взаємодії з елементами.

Об'єктно-орієнтована мова програмування C# буде задіяна при написанні контролерів та моделей, для отримання доступу до даних, надання логіки поведінки сторінкам та забезпечення зручної взаємодії користувача з веб-сайтом через визначений раніше (у ході кваліфікаційної роботи) функціонал.

При розробці веб-застосунків C# використовуватиметься суміжно з платформою .NET, а саме з ASP.NET Core.

ASP.NET Core Identity забезпечить управління користувачами (паролями, даними профілю, ролями, вимогами, токенами, автентифікацією) та зберігання облікових записів. Користувачі зможуть створити обліковий запис з інформацією для входу, що зберігатиметься в Identity, або скористатися зовнішнім провайдером логінів. Підтримувані зовнішні провайдери включають Facebook, Google, обліковий запис Microsoft і Twitter.

Identity налаштовуватиметься з використанням бази даних SQL Server для зберігання імен користувачів, паролів і даних профілю.

За допомогою технології Entity Framework Core буде отримано доступ та здійснено керування реляційними даними та перетворення їх у об'єктно-орієнтовану структуру. Для доступу до даних буде використано модель, яка дозволить користувачам запитувати і зберігати дані.

Екземпляри класів сутностей отримуватимуться з бази даних за допомогою LINQ (Language Integrated Query).

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		42

Екземпляри класів-сутностей можна буде використовувати для створення, видалення та модифікації даних у базі даних.

Отже, у даному розділі було визначено тип архітектури та шаблон проектування, які будуть використовуватись для розробки довідково-інформаційної системи, виділено ряд сутностей і атрибутів, нормалізовано дані та на основі них спроектовано логічну модель БД. Також, спроектовано інтерфейс користувача, розроблено макети сторінок, визначено тип структури веб-сайту та зображено впорядковані сторінки на рисунку логічної структури ДІС (рисунок 2.6). У пункті 2.4 вибрано технології для розробки за допомогою яких буде відбуватися подальша реалізація системи.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
						43
Зм.	Арк	№ докум.	Підпис	Дата		

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ

3.1 Реалізація бази даних

Побудувати фізичну модель бази даних можна на основі спроектованої в пункті 2.2 «Проектування логічної моделі бази даних» логічної. Для цього потрібно зобразити сутності за допомогою таблиць, а в якості полів використати атрибути.

У кожній таблиці необхідно створити первинні та зовнішні ключі, завдяки чому зв'язати пов'язані таблиці між собою. Для кожного поля слід задати тип та розмір даних, за необхідності встановити обмеження.

Готові таблиці майбутньої бази даних наведено нижче (таблиця 3.1 – 3.4):

Таблиця 3.1 – Таблиця «Property» з урахуванням обмежень, типу та розміру даних

Поле	Тип та розмір даних	Обмеження
1	2	3
PropertyId	INT	PRIMARY KEY, UNIQUE KEY, NOT NULL
AddressId	INT	FOREIGN KEY, NOT NULL
Area	NVARCHAR(MAX)	NOT NULL
BedNumber	INT	NOT NULL
BathNumber	INT	NOT NULL
Description	NVARCHAR(MAX)	NOT NULL
Image	NVARCHAR(MAX)	NOT NULL
Price	DECIMAL(15,2)	NOT NULL
PhoneNumber	NVARCHAR(MAX)	NOT NULL
CategoryId	INT	FOREIGN KEY, NOT NULL
OperationId	INT	FOREIGN KEY, NOT NULL

Таблиця 3.2 – Таблиця «Address» з урахуванням обмежень, типу та розміру даних

Поле	Тип та розмір даних	Обмеження
1	2	3
AddressId	INT	PRIMARY KEY, UNIQUE KEY, NOT NULL
State	NVARCHAR(MAX)	NOT NULL

Кінець таблиці 3.2

1	2	3
City	NVARCHAR(MAX)	NOT NULL
Country	NVARCHAR(MAX)	NOT NULL
Street	NVARCHAR(MAX)	NOT NULL
Unit	NVARCHAR(MAX)	NOT NULL

Таблиця 3.3 – Таблиця «Category» з урахуванням обмежень, типу та розміру даних

Поле	Тип та розмір даних	Обмеження
1	2	3
CategoryId	INT	PRIMARY KEY, UNIQUE KEY, NOT NULL
PropertyCategory	NVARCHAR(MAX)	NOT NULL

Таблиця 3.4 – Таблиця «Operation» з урахуванням обмежень, типу та розміру даних

Поле	Тип та розмір даних	Обмеження
1	2	3
OperationId	INT	PRIMARY KEY, UNIQUE KEY, NOT NULL
OperationType	NVARCHAR(MAX)	NOT NULL

Створення бази даних відбуватиметься у середовищі розробки Microsoft Visual Studio. Для цього, до уже існуючого проекту потрібно додати новий об'єкт, натиснувши «Назва проекту» → «Додати» → «Новий об'єкт» → «База даних». Далі ввести назву БД («Properties.mdf») та підтвердити створення.

Для додавання та налаштування таблиць використовується вбудований графічний редактор або SQL запити. На прикладі таблиці «Address» розглянуто обидва варіанти:

```
CREATE TABLE [dbo].[Address] (
    [AddressId] INT IDENTITY (1, 1) NOT NULL,
    [State] NVARCHAR (MAX) NOT NULL,
    [City] NVARCHAR (MAX) NOT NULL,
    [Country] NVARCHAR (MAX) NOT NULL,
    [Street] NVARCHAR (MAX) NOT NULL,
    [Unit] NVARCHAR (MAX) NOT NULL,
    PRIMARY KEY CLUSTERED ([AddressId] ASC)
```

) ;

	Name	Data Type	Allow Nulls	Default
PK	AddressId	int	<input type="checkbox"/>	
	State	nvarchar(MAX)	<input type="checkbox"/>	
	City	nvarchar(MAX)	<input type="checkbox"/>	
	Country	nvarchar(MAX)	<input type="checkbox"/>	
	Street	nvarchar(MAX)	<input type="checkbox"/>	
	Unit	nvarchar(MAX)	<input type="checkbox"/>	

Рисунок 3.1 – Створення таблиці «Address» у графічному редакторі Visual Studio

Те ж саме потрібно повторити для усіх інших таблиць та, за допомогою ключів, налаштувати зв'язки між ними. Готова база даних матиме наступний вигляд (рисунок 3.2 – рисунок 3.3):

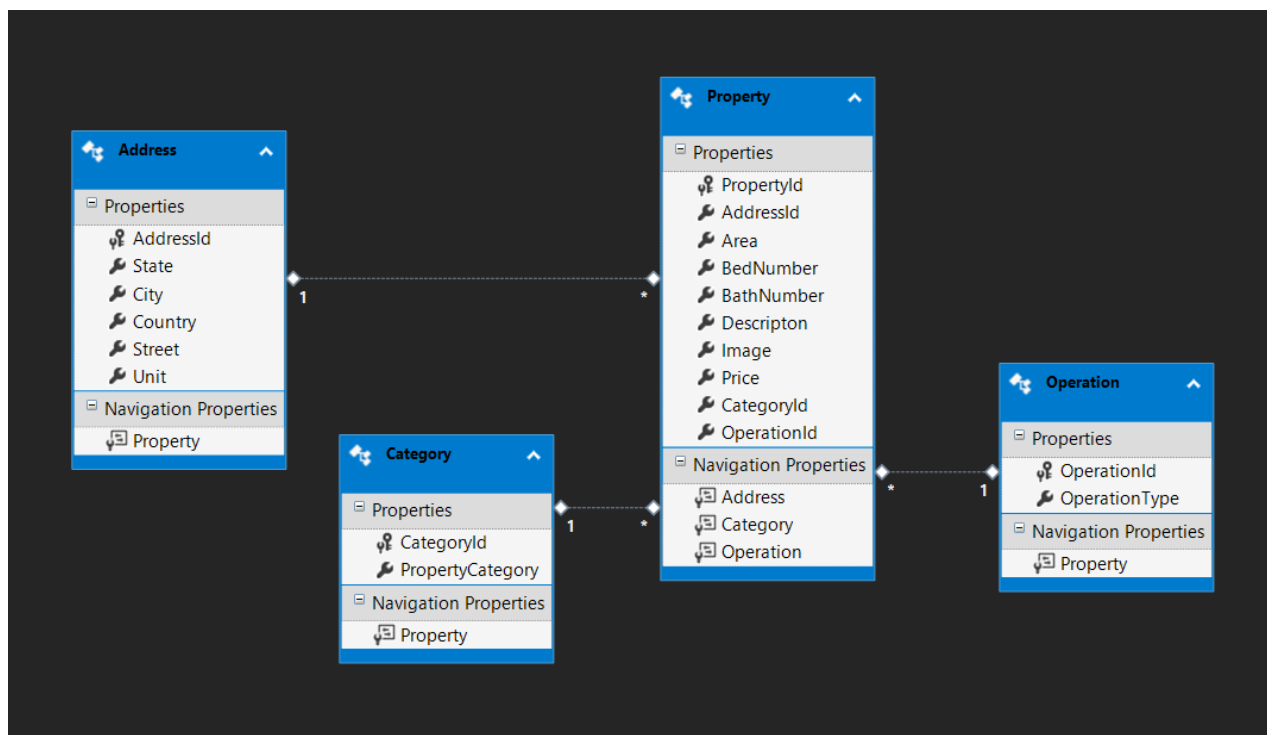


Рисунок 3.2 – Фізична модель бази даних ДІС агентства нерухомості

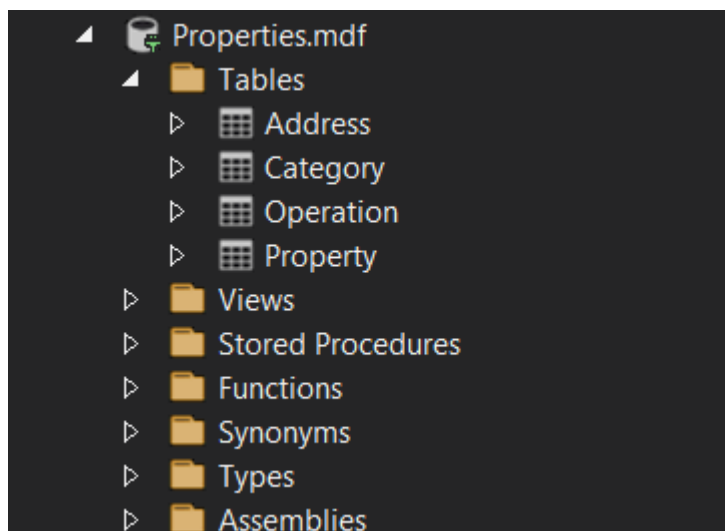


Рисунок 3.3 – Структура БД у вікні Server Explorer

Для можливості реєстрації та авторизації користувачів у системі до проекту також необхідно підключити пакети Identity, з якими, до уже створеної бази даних, додаються ще декілька таблиць (рисунок 3.4):

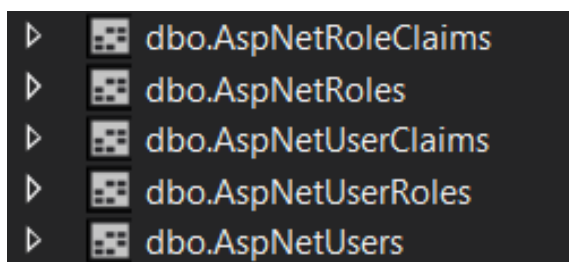


Рисунок 3.4 – Таблиці ASP.NET Identity у вікні Server Explorer

В цих таблицях будуть зберігатись дані про користувачів, такі як ім'я, електронна пошта, паролі, ролі та відповідно призначені права.

3.2 Реалізація модулів системи

Реалізація довідково-інформаційної системи починається з підключення до проекту усіх необхідних пакетів, перелічених вище, та реалізації CRUD функціоналу. CRUD означає CREATE, READ, UPDATE і DELETE, що описує

чотири основні операції для створення та управління постійними елементами даних, переважно в реляційних та NoSQL базах даних.

Ці функції описуються в основному контролері, який відповідає за відображення та роботу елементів стартової сторінки.

Отож, наступним чином функція View виводить на екран попередньо внесену в базу даних інформацію про пропозиції:

```
public ActionResult View(int id)
{
    var prop = propRepository.Props.FirstOrDefault(i =>
i.PropertyId == id);
    if (prop != null)
    {
        return View(prop);
    }
    return RedirectToAction("Index");
}
```

Функція Edit забезпечує можливість редагування існуючих даних про об'єкт нерухомості та їх миттєве оновлення:

```
[HttpGet]
public ActionResult Edit(int id)
{
    var prop = propRepository.Props.FirstOrDefault(i =>
i.PropertyId == id);
    SelectList cat = new SelectList(db.Category,
"CategoryId", "PropertyCategory", prop.CategoryId);
    ViewBag.Cat = cat;
    if (prop != null)
    {
        return View(prop);
    }
    return RedirectToAction("Index");
}

[HttpPost]
public ActionResult Edit(Property prop)
{
    db.Entry(prop).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		48

Функція Create дозволяє створювати нові об'єкти в базі даних, відповідно й на сторінці сайту, також одразу оновлювати сторінку з пропозиціями:

```
[HttpGet]
public ActionResult Create()
{
    SelectList cat = new SelectList(db.Category,
"CategoryId", "PropertyCategory");
    ViewBag.Cat = cat;
    return View();
}
```

```
[HttpPost]
public ActionResult Create(Property prop)
{
    db.Entry(prop).State = EntityState.Added;
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

Delete очищує дані про об'єкт, видаляє їх як з бази даних, так і зі сторінки сайту:

```
[HttpPost]
public ActionResult Delete(int? id)
{
    var prop = propRepository.Props.FirstOrDefault(i =>
i.PropertyId == id);
    db.Property.Remove(prop);
    db.SaveChanges();
    return RedirectToAction("Index");
}
```

Розділення об'єктів по категоріям відбувається наступним чином:

```
public ViewResult Category()
{
    var categories = new CategoryViewModel
    {
        categories = categoryRepository.AllCategories
    };
    return View(categories);
}
```

```
[Route("Home/Category/{category}")]
public ActionResult CategoryItems(string category)
```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
						49
Зм.	Арк	№ докум.	Підпис	Дата		

```

    {
        string _category = category;
        IEnumerable<Property> props = null;
        string currCategory = "";
        if (string.Equals("All Offers", category,
StringComparison.OrdinalIgnoreCase))
        {
            return RedirectToAction("Index");
        }
        else
        {
            if (string.Equals("Rent", category,
StringComparison.OrdinalIgnoreCase))
            {
                props = propRepository.Props.Where(i =>
i.Category.Name.Equals("Rent")).OrderBy(i => i.PropertyId);
            }
            else
            {
                if (string.Equals("Sale", category,
StringComparison.OrdinalIgnoreCase))
                {
                    props = propRepository.Props.Where(i =>
i.Category.Name.Equals("Sale")).OrderBy(i => i.PropertyId);
                }
            }

            currCategory = _category;
        }

        var propObj = new CategoryViewModel
        {
            allProps = props,
            currentCategory = currCategory
        };

        ViewBag.Title = "Unreal Estate";
        return View(propObj);
    }

```

За допомогою функції Equals здійснюється порівняння назви категорій, в рядку переходу на сторінку, з тією, що вказана в інформації про об'єкт нерухомості в базі даних. При виникненні співпадіння, об'єкти фільтруються та відображаються лише ті, які відповідають поставленій умові.

Реєстрація та логін реалізовані за допомогою відповідних методів:

```

[HttpGet]
public IActionResult Register()

```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		50

```

        {
            return View();
        }

[HttpPost]
public async Task<IActionResult> Register(RegisterViewModel
model)
    {
        if (ModelState.IsValid)
        {
            User user = new User { UserName = model.Email,
Email = model.Name, PasswordHash = model.PasswordConfirm };
            var result = await _userManager.CreateAsync(user,
model.Password);
            if (result.Succeeded)
            {
                await _signInManager.SignInAsync(user,
false);
                return RedirectToAction("Index", "Home");
            }
            else
            {
                foreach (var error in result.Errors)
                {
                    ModelState.AddModelError(string.Empty,
error.Description);
                }
            }
        }
        return View(model);
    }

```

На основі коректно введених даних створюється новий об'єкт типу User, після чого відбувається перехід на головну сторінку сайту.

```

[HttpGet]
public IActionResult Login(string returnUrl = null)
    {
        return View(new LoginViewModel { ReturnUrl =
returnUrl });
    }

[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Login(LoginViewModel model)
    {
        if (ModelState.IsValid)
        {
            var result = await
_signInManager.PasswordSignInAsync(model.Email, model.Password,
model.RememberMe = false, false);

```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		51

```

        if (result.Succeeded)
        {
            if (!string.IsNullOrEmpty(model.ReturnUrl) &&
                Url.IsLocalUrl(model.ReturnUrl))
            {
                return Redirect(model.ReturnUrl);
            }
            else
            {
                return RedirectToAction("Index", "Home");
            }
        }
        else
        {
            ModelState.AddModelError("", "Invalid login
and (or) password");
        }
    }
    return View(model);
}

```

Функція Login звіряє введені дані з даними БД і якщо такі зареєстровані, також перенаправляє користувача на головну.

Налаштування ролей відбувається лише зі сторінки адміністратора. Створення, видалення та присвоєння описані в функціях нижче:

```

[HttpPost]
public async Task<IActionResult> Create(string name)
{
    if (!string.IsNullOrEmpty(name))
    {
        IdentityResult result = await
        _roleManager.CreateAsync(new IdentityRole(name));
        if (result.Succeeded)
        {
            return RedirectToAction("Index");
        }
        else
        {
            foreach (var error in result.Errors)
            {
                ModelState.AddModelError(string.Empty,
error.Description);
            }
        }
    }
    return View(name);
}

```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		52

```

[HttpPost]
public async Task<IActionResult> Delete(string id)
{
    IdentityRole role = await
_roleManager.FindByIdAsync(id);
    if (role != null)
    {
        IdentityResult result = await
_roleManager.DeleteAsync(role);
    }
    return RedirectToAction("Index");
}

public async Task<IActionResult> Edit(string userId)
{
    User user = await _userManager.FindByIdAsync(userId);
    if (user != null)
    {
        var userRoles = await
_userManager.GetRolesAsync(user);
        var allRoles = _roleManager.Roles.ToList();
        ChangeRoleViewModel model = new
ChangeRoleViewModel
        {
            UserId = user.Id,
            userEmail = user.Email,
            UserRoles = userRoles,
            AllRoles = allRoles
        };
        return View(model);
    }

    return NotFound();
}

[HttpPost]
public async Task<IActionResult> Edit(string userId,
List<string> roles)
{
    User user = await _userManager.FindByIdAsync(userId);
    if (user != null)
    {
        var userRoles = await
_userManager.GetRolesAsync(user);
        var allRoles = _roleManager.Roles.ToList();
        var addedRoles = roles.Except(userRoles);
        var removedRoles = userRoles.Except(roles);
        await _userManager.AddToRolesAsync(user,
addedRoles);
        await _userManager.RemoveFromRolesAsync(user,
removedRoles);
        return RedirectToAction("UserList");
    }
}

```

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		53

```
return NotFound();
}
```

Спільна для всіх сторінок розкладка інтерфейсу наведена у фрагменті коду нижче:

```
<body>
  <header>
    <div class="header-content">
      <div class="title">Unreal Estate</div>
      <div class="navigation">
        <ul class="header-contacts">
          <li class="header-contacts-item">
            <a href="tel:123-456-7890" class="header-
phone-number">123-456-7890</a>
          </li>
          <li class="header-contacts-item">
            <a href="https://www.instagram.com/"
class="header-contacts-links"></a>
          </li>
          <li class="header-contacts-item">
            <a href="https://www.facebook.com/"
class="header-contacts-links"></a>
          </li>
        </ul>
        <ul class="header-menu">
          <li class="header-menu-item">
            <a href="~/Home/#available"
class="header-menu-links">Offers</a>
          </li>
          <li class="header-menu-item">
            <a href="~/Home/Category/#categories"
class="header-menu-links">Categories</a>
          </li>
          <li class="header-menu-item">
            <@if (User.Identity.IsAuthenticated)
            {
              <a href="~/Request/#request"
class="header-menu-links">Requests</a>
            }
          </li>
          <li class="header-menu-item">
            <@if (User.Identity.IsAuthenticated)
            {
              <@if (User.IsInRole("Admin"))
              {
                <a href="~/Admin/#admin"
class="header-menu-links">Admin</a>
              }
            }
          </li>
        </ul>
      </div>
    </div>
  </header>
</body>
```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		54

```

                                <a href="~/Role/#roles"
class="header-menu-links">Roles</a>
                                </li>
                                }
                                }
                                else
                                {
                                    <li class="header-menu-item">
                                        <a href="~/Account/Login/#login"
class="header-menu-links">Login</a>
                                        </li>
                                        <li class="header-menu-item">
                                            <a
href="~/Account/Register/#register" class="header-menu-
links">Register</a>
                                        </li>
                                    }
                                </ul>
                                @if (User.Identity.IsAuthenticated)
                                {
                                    <ul class="header-menu" style="margin-bottom:
0px;">
                                        <li class="header-menu-item">
                                            <a class="header-menu-links">Welcome,
@User.Identity.Name!</a>
                                        </li>
                                        <li class="header-menu-item">
                                            <a class="header-menu-links" aria-
current="page">
                                                <form method="post" asp-
controller="Account" asp-action="Logout">
                                                    <input class="btn btn-
warning" type="submit" value="Logout" />
                                                </form>
                                            </a>
                                        </li>
                                    </ul>
                                }
                                </div>
                            </div>
</header>
<div class="back-image">
    <div class="slogan">
        <div class="border">
            <span>Dream Big.</span>
            <br>
            <span>Live Well</span>
        </div>
    </div>
</div>
<div class="container mt-3">
    @RenderBody()
</div>

```

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		55

```
<footer>
  <div>© 2023 - Unreal Estate.com</div>
</footer>
<script src="~/js/bootstrap.min.js"></script>
</body>
```

3.3 Інструкція користувача

Інструкція користувача призначена для того, щоб допомогти користувачам у використанні певного продукту, послуги або програми. Більшість посібників користувача містять як письмову інструкцію, так і пов'язані з нею зображення. У випадку комп'ютерних програм зазвичай додають скріншоти людино-машинного інтерфейсу, а посібники для обладнання часто містять чіткі, спрощені схеми. Мова, що використовується, адаптована до цільової аудиторії, а жаргонізми зведені до мінімуму або ретельно пояснені.

Опис роботи ДІС повинен містити наступні дані:

- загальні відомості про розроблену систему;
- вхідні та вихідні дані;
- її функціональне призначення;
- послідовність дій користувача, які забезпечують завантаження, виконання та завершення програми;
- копії вікон інтерфейсу, результатів роботи програми тощо.

Довідково-інформаційна система агентства нерухомості розроблена у вигляді веб-сайту за допомогою мови програмування С# у середовищі розробки Microsoft Visual Studio 2019. Для роботи із системою користувач має бути досвідченим користувачем комп'ютера, а також мати пристрій із встановленим браузером та стабільним підключенням до мережі Інтернет.

Довідково-інформаційна система агентства нерухомості призначена для забезпечення користувачів можливістю швидкого пошуку, огляду та укладення

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		56

договору про оренду чи купівлю опублікованої на сайті нерухомості в будь-якій точці світу та з будь-якої точки світу, не переживаючи за безпеку себе і своїх даних. У системі реалізовані наступні функціональні можливості:

- публікації, редагування та видалення пропозицій з продажу та оренди нерухомості;
- пошуку об'єктів нерухомості;
- авторизації та реєстрації користувача;
- керування системою від імені адміністратора: блокування та видалення аккаунтів, створення, редагування та приховування старих публікацій.

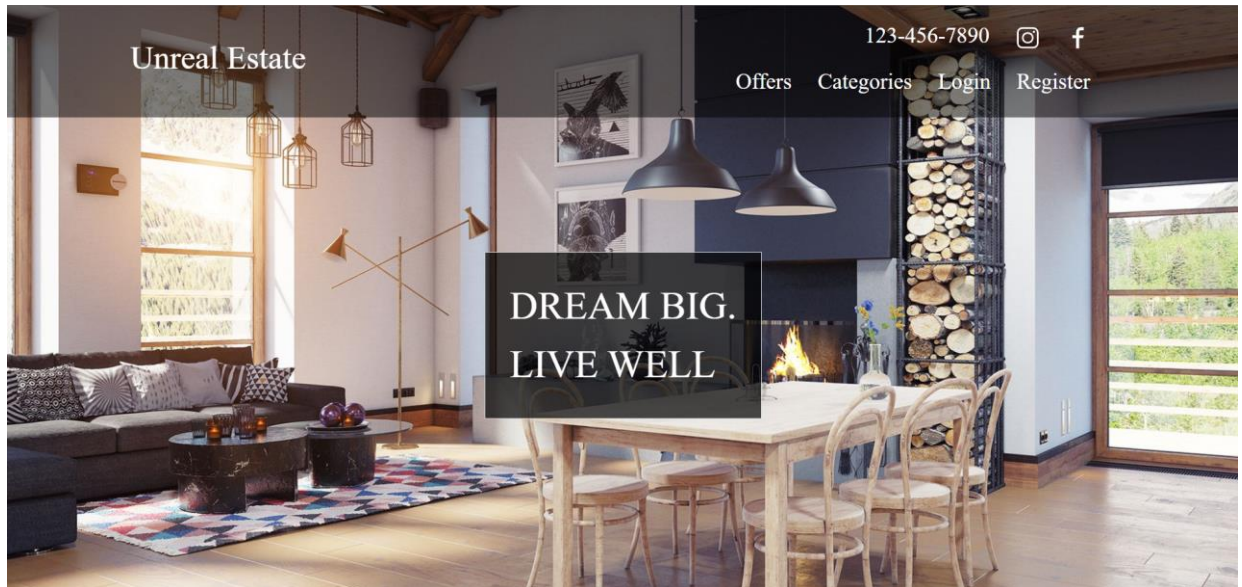
Нижче наведені алгоритми дій за виконання яких користувач зможе отримати бажані результати.

Перш за все, потрібно розглянути стандартний випадок, коли звичайний користувач заходить на сайт, шукаючи помешкання для оренди або покупки.

Для цього йому потрібно проглянути список доступних пропозицій (рисунк 3.5), обрати ту, що є найбільш підходящою, і перейти на сторінку детальної інформації про неї, просто натиснувши на зображення або підпис під ним (рисунк 3.6). Щоб замовити дзвінок від власника або орендодавця, потрібно натиснути на кнопку «Request a call» в правому нижньому кутку сторінки.

Якщо користувач ще не увійшов в акаунт, перед ним відкриється сторінка авторизації (рисунк 3.7), де його попросить ввести свою електронну пошту та пароль. Якщо ж він не зареєстрований, потрібно буде перейти на форму реєстрації по натисканню на відповідну кнопку в шапці сторінки.

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		57



We Offer



Woodgrove at Sterlington
950 Woodmont Ln, Slootsburg, NY 10974
3 249,00 €



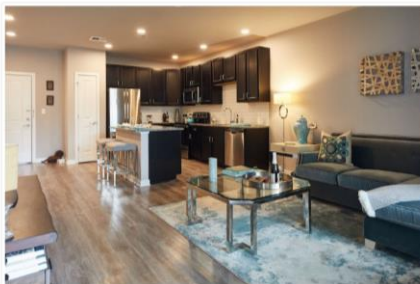
Fairview Apartments
100 Fairview Sq, Ithaca, NY 14850
2 945,00 €



The Brix
1000 Brix Pl, Uniondale, NY 11553
2 850,00 €



Рисунок 3.5 – Головна сторінка, список доступних пропозицій



Fairview Apartments

100 Fairview Sq, Ithaca, NY 14850

For the ultimate high-rise or garden-style living experience, make your new home at Fairview Apartments. Fairview Apartments offers fully furnished studio, 1, and 2 bedroom apartments and 2 and 3 bedroom townhomes for rent in East Ithaca. Located just steps from Cornell University with stunning views, these homes offer convenient living in the heart of it all. With all utilities, cable and high-speed internet included in your rent, you can enjoy more & worry less when you choose to live at Fairview Apartments.

Amenities: 3 Beds , 2 baths, 1100 sqft

Price: 2 945,00 €

[Request a call](#)

© 2023 - Unreal Estate.com

Рисунок 3.6 – Детальна інформація про пропозицію

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		58

Login

Email

Password

Login

© 2023 - Unreal Estate.com

Рисунок 3.7 – Авторизація користувача

Після натискання на кнопку «Request a call» користувачу потрібно буде ввести персональні дані для зворотного зв'язку з ним та підтвердити відправку інформації натиском кнопки «Submit» (рисунок 3.8):

Fill in your personal info

Name

Surname

Address

Phone Number

Email

Submit

© 2023 - Unreal Estate.com

Рисунок 3.8 – Заповнення форми зворотного зв'язку

Тепер, можна перейти до випадку, коли вхід в систему здійснює не звичайний користувач, а користувач з встановленими правами адміністратора. Для цієї ролі головна сторінка набуває наступного вигляду (рисунок 3.9). На ній з'являються кнопки видалення, додавання (рисунок 3.10) та редагування (рисунок 3.11) записів.

Для додавання адміністратору потрібно заповнити пусті поля на формі та натиснути кнопку «Create», після чого створиться новий об'єкт, який буде додано в кінець списку на головній сторінці.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		59

We Offer

Create







		
Woodgrove at Sterlington 950 Woodmont Ln, Sloatsburg, NY 10974 3 249,00 ₺	Fairview Apartments 100 Fairview Sq, Ithaca, NY 14850 2 945,00 ₺	The Brix 1000 Brix Pl, Uniondale, NY 11553 2 850,00 ₺
Edit	Edit	Edit
Delete	Delete	Delete
		

Рисунок 3.9 – Головна сторінка для адміністратора

New Property

Name

Type

Description

Care

Image

Price

IsFavourite

Available

Category

Рисунок 3.10 – Сторінка додавання нового об'єкту

Для редагування адміністратор повинен змінити інформацію про об'єкт в одному чи декількох полях та натиснути кнопку «Edit». Змінений об'єкт буде оновлено та розміщено головній сторінці.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
						60
Зм.	Арк	№ докум.	Підпис	Дата		

Fairview Apartments

Name
Fairview Apartments

Type
100 Fairview Sq, Ithaca, NY 14850

Description
For the ultimate high-rise or garden-style living experience, make your new home

Care
3 Beds, 2 baths, 1100 sqft

Image
Choose image...

Price
2945

IsFavourite

Available

Category
Sale

Edit

Рисунок 3.11 – Сторінка редагування існуючого об'єкту

При натисненні на кнопку «Roles» в шапці сайту, перед адміністратором відкривається сторінка ролей (рисунок 3.12):

Roles List

User	Delete
Admin	Delete
Users List	Add Role

© 2023 - Unreal Estate.com

Рисунок 3.12 – Список ролей

Для присвоєння ролі користувачу потрібно натиснути на кнопку «Users List», після якої відкриється сторінка користувачів, зареєстрованих на сайті (рисунок 3.13):

Users List

Anastasia	Permissions
First User	Permissions

© 2023 - Unreal Estate.com

Рисунок 3.13 – Список користувачів

Напроти кожного користувача можна побачити кнопку «Permissions», натиснення якої і відкриває сторінку, де можна вибрати роль, яка буде йому надана (рисунок 3.14):

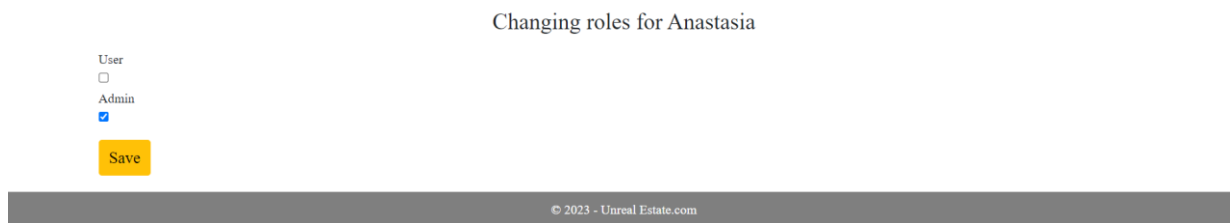


Рисунок 3.14 – Зміна ролей для користувача

При роботі з довідково-інформаційною системою можна застосовувати будь-які технічні засоби, що мають доступ до Інтернету та встановлений на них браузер. Такими можуть стати ноутбуки, персональні комп'ютери, мобільні пристрої або планшети.

Для відкриття довідково-інформаційної системи потрібно в пошуковому рядку браузера ввести адресу і перейти на сайт. Для безпроблемного використання системи – скористатися інструкцією, описаною вище.

Вхідні дані системи можна поділити на два типи: ті, що вводить користувач (особисті дані для облікового запису, дані для отримання зворотного зв'язку) і ті, що вводить адміністратор (дані таблиць бази даних). До вихідних даних належить інформація про об'єкти нерухомості, розташування, ціни, виведені на сторінках сайту, і контактна інформація агентства нерухомості.

3.4 Вимоги до апаратно-програмних засобів

Для нормальної роботи довідково-інформаційної системи вимоги для функціонування відповідають мінімальним системним вимогам будь-якого браузера.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		62

Google Chrome працюватиме на комп'ютерах з процесором Pentium 4 або новішої версії, що охоплює більшість комп'ютерів, випущених з 2001 року. Комп'ютер повинен мати приблизно 100 МБ вільного місця на жорсткому диску та 128 МБ оперативної пам'яті. Найстарішою версією Windows, яку підтримує Chrome, є Windows XP зі встановленим Service Pack 2. Chrome також працює на комп'ютерах зі встановленою Windows Vista або Windows 7.

Щоб використовувати браузер Chrome на Android необхідно мати версію операційної системи Android 7.0 Nougat.

3.5 Тестування системи

Перед надсиланням даних на сервер важливо переконатися, що всі обов'язкові поля форми заповнені даними в коректному форматі. Це називається валідацією на стороні клієнта і допомагає переконатися, що дані, введені в кожен елемент форми, відповідають встановленим вимогам.

Зайшовши на будь-який популярний сайт, що має форму реєстрації, можна помітити, що при введенні даних у неправильному форматі, користувача одразу повідомляють про наявність проблеми.

Часто зустрічаються повідомлення:

- «Обов'язкове поле»;
- «Будь ласка, введіть коректну адресу поштової скриньки»;
- «Довжина пароля має бути від 8 до 30 символів і містити одну велику літеру, один символ і одну цифру».

Чому валідація даних важлива? Є три основні причини:

- застосунок не працюватиме належним чином, якщо дані від користувачів зберігатимуться в неправильному форматі, будуть некоректними або пропущеними;
- введення надійних паролів полегшить захист акаунтів користувачів;
- більш захищена форма допоможе уникнути взлому.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		63

В розроблюваній системі валідація буде здійснюватися за допомогою атрибутів валідації моделі. Завдяки ним можна керувати валідацією і вкладати логіку перевірки значень властивостей в атрибути цих властивостей, не вдаючись до коду.

ASP.NET Core MVC пропонує до використання атрибути, що встановлюють обмеження по довжині рядка, його важливості, типу даних, виведеному повідомленню помилки і т.д. Вони розміщуються в квадратних дужках над кожним полем моделі.

До прикладу можна взяти модель замовлення:

```
[StringLength(20)]
[Required(ErrorMessage = "The Name field is required")]
public string Name { get; set; }

[StringLength(20)]
[Required(ErrorMessage = "The Surname field is required")]
public string Surname { get; set; }

[StringLength(30)]
[Required(ErrorMessage = "The Address field is required")]
public string Address { get; set; }

[Display(Name = "Phone Number")]
[StringLength(15)]
[DataType(DataType.PhoneNumber)]
[Required(ErrorMessage = "The Phone Number field is required")]
public string PhoneNumber { get; set; }

[StringLength(20)]
[DataType(DataType.EmailAddress)]
[Required(ErrorMessage = "The Email field is required")]
public string Email { get; set; }
```

Підтвердження відправки форми замовлення відбувається лише при умові правильності усіх введених даних.

```
if (ModelState.IsValid)
{
    orders.createOrder(order);
    return RedirectToAction("Complete");
}
```

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		64

На сайті процес валідації має наступний вигляд (рисунок 3.15):

Name

The Name field is required

Surname

The Surname field is required

Address

The Address field is required

Phone Number

The Phone Number field is required

Email

The Email field is required

Submit

Рисунок 3.15 – Не пройдена валідація даних, поля не заповнені

Аналогічним чином виглядає і валідація при заповненні полів даними на сторінці логіну та реєстрації (рисунок 3.16 – рисунок 3.17):

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		65

Email

The Email field is required.

Name

The Name field is required.

Password

The Password field is required.

Confirm password

The Email field is required.

Register

Рисунок 3.16 – Валідація даних на сторінці реєстрації

Email

The Email field is required.

Password

The Password field is required.

Login

Рисунок 3.17 – Валідація даних на сторінці авторизації

Отже, у цьому розділі було виконано реалізацію бази даних та основних модулів системи. Відповідно до них було розроблено інструкцію користувача. Також були визначені вимоги до апаратно-програмних засобів, що забезпечать нормальну роботу системи, і проведено валідацію форм введення даних.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		66

ВИСНОВКИ

Довідково-інформаційні системи значно спрощують пошук інформації користувачам у певній предметній області. Системи агентств нерухомості стають сучасним та зручним рішенням питань продажу, покупки та оренди об'єктів нерухомості та автоматизації робочих процесів, шляхом переведення усієї одноманітної та шаблонної роботи ріелторів в цифровий формат.

Завдання, які підлягали вирішенню в ході виконання даної кваліфікаційної роботи:

- проведення дослідження предметної області;
- детальний аналіз існуючих рішень;
- визначення особливостей та специфіки роботи подібних сайтів, їх переваг та недоліків;
- визначення функціональних задач;
- розробка технічного завдання;
- побудова архітектури системи, вибір інструментів та технологій, необхідних для розробки;
- розробка сучасного та функціонального інтерфейсу;
- виконання програмної реалізації продукту;
- проведення тестування готового продукту.

То ж, у першому розділі було здійснено дослідження предметної області, її структурних та функціональних особливостей, проаналізовано існуючі рішення у профільній галузі, проведено і занесено у таблиці їх порівняльну характеристику та визначено вимоги до розроблюваного програмного забезпечення, які розгорнуто описано у технічному завданні.

У другому розділі було визначено тип архітектури та шаблон проектування, які використовувались для розробки довідково-інформаційної системи, виділено ряд сутностей і атрибутів, нормалізовано дані та на основі них спроектовано логічну модель БД. Також, спроектовано інтерфейс

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		67

користувача, розроблено макети сторінок, визначено тип структури веб-сайту та зображено впорядковані сторінки на рисунку логічної структури ДІС (рисунок 2.6). У пункті 2.4 вибрано технології для розробки за допомогою яких відбувалась реалізація системи.

У третьому розділі було виконано реалізацію бази даних та основних модулів системи. Відповідно до них було розроблено інструкцію користувача. Також були визначені вимоги до апаратно-програмних засобів, що забезпечать нормальну роботу системи, і проведено валідацію форм введення даних.

У результаті виконання кваліфікаційної роботи було реалізовано довідково-інформаційну систему агентства нерухомості, що надає користувачеві можливість швидкого пошуку, огляду та укладення договору про оренду чи купівлю опублікованої на сайті нерухомості в будь-якій точці світу та з будь-якої точки світу, не переживаючи за безпеку себе і своїх даних.

					КвРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		68

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Автоматизація для нерухомості. OneBox OS. Операційна система для вашого бізнесу. URL: <https://1b.app/ua/automation-for/real-estates/> (дата звернення: 02.01.2023).

2. Діджиталізація агентства нерухомості. Збільшення продажів на 40%. Webcase. URL: <https://webcase.com.ua/uk/blog/how-it-can-increase-real-estate-agency-sales-300/> (дата звернення: 02.01.2023).

3. CRM для агентств нерухомості: застосунок, веб-сайт – вигода розробки, основні функції, приклади. IT-компанія повного цикла розробки програмних продуктів WEZOM – Київ, Україна. URL: <https://wezom.com.ua/ua/blog/digital-reshenie-dlya-segmenta-nedvizhimosti> (дата звернення: 02.01.2023).

4. Які суб'єкти є на ринку нерухомості. Riel.ua. URL: <https://riel.ua/blogs/rozmayittya-ta-bagatolikist-rinku-nerukhomosti> (дата звернення: 10.01.2023).

5. Інвестору на замітку: головні учасники ринку нерухомості України. 24 Канал. URL: https://realestate.24tv.ua/investoru-zamitku-golovni-uchasniki-rinku-nerukhomosti-ukrayini_n1843355 (дата звернення: 10.01.2023).

6. Zillow. URL: <https://www.zillow.com/> (дата звернення: 17.01.2023).

7. Trulia. URL: <https://www.trulia.com/rent/> (дата звернення: 17.01.2023).

8. Realtor.com® | Homes for Sale, Apartments & Houses for Rent. URL: <https://www.realtor.com/> (дата звернення: 17.01.2023).

9. Redfin. URL: <https://www.redfin.com/> (дата звернення: 17.01.2023).

10. Homes.com. URL: <https://www.homes.com/> (дата звернення: 17.01.2023).

11. HomeFinder | Real Estate & Homes for Sale | Property Rentals & Foreclosures. HomeFinder. URL: <https://homefinder.com/> (дата звернення: 17.01.2023).

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		69

12. Carmody B. The 7 Best Real Estate Websites of 2023. Investopedia. URL: <https://www.investopedia.com/best-real-estate-websites-5069964> (дата звернення: 19.01.2023).

13. The 12 Best Real Estate Websites in America (2023 Updates). Houzeo Blog. URL: <https://www.houzeo.com/blog/best-real-estate-websites/> (дата звернення: 19.01.2023).

14. Top 10 Real Estate Websites in the U.S. (2023). RubyHome.com. URL: <https://www.rubyhome.com/blog/most-visited-real-estate-websites/> (дата звернення: 19.01.2023).

15. Вадим С. Использование диаграммы вариантов использования UML при проектировании программного обеспечения. Хабр. URL: <https://habr.com/ru/articles/566218/> (дата звернення: 25.01.2023).

16. Диаграмма вариантов использования (UseCase diagram). Flexberry PLATFORM Documentation. URL: https://flexberry.github.io/ru/fd_use-case-diagram.html (дата звернення: 25.01.2023).

17. UML Use Case Diagram Tutorial. Lucidchart. URL: <https://www.lucidchart.com/pages/uml-use-case-diagram> (дата звернення: 25.01.2023).

18. The Four Types of Relationship in Use Case Diagram - Visual Paradigm Blog. Visual Paradigm Blog. URL: <https://blog.visual-paradigm.com/the-four-types-of-relationship-in-use-case-diagram/> (дата звернення: 25.01.2023).

19. Use Case Diagram Relationships Explained with Examples - Creately Blog. Creately Blog. URL: <https://creately.com/blog/diagrams/use-case-diagram-relationships/> (дата звернення: 25.01.2023).

20. Клиент-серверная архитектура / Введение в интернет. Хекслет. URL: https://ru.hexlet.io/courses/internet-fundamentals/lessons/client-server/theory_unit (дата звернення: 01.02.23).

21. ITELON. Архитектура «Клиент-Сервер». Продажа серверного оборудования от компании Itelon. URL: <https://itelon.ru/blog/arkhitektura-klient-server/> (дата звернення: 01.02.23).

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		70

22. ASP.NET MVC Pattern | .NET. Microsoft. URL: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc> (дата звернення: 05.02.23).
23. MVC Framework Introduction - GeeksforGeeks. GeeksforGeeks. URL: <https://www.geeksforgeeks.org/mvc-framework-introduction/> (дата звернення: 05.02.23).
24. MVC Framework - Introduction. Online Courses and eBooks Library. URL: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm (дата звернення: 05.02.23).
25. Module. Techopedia. URL: <https://www.techopedia.com/definition/3843/module> (дата звернення: 10.02.23).
26. Wright G. What is a module in software, hardware and programming?. WhatIs.com. URL: <https://www.techtarget.com/whatis/definition/module#:~:text=What%20are%20software%20modules?,that%20software%20can%20be%20modular> (дата звернення: 10.02.23).
27. Логическая структура базы данных. Studbooks. URL: https://studbooks.net/2271972/informatika/logicheskaya_struktura_bazy_dannyh (дата звернення: 12.02.23).
28. Полонец В. Что такое логическая структура сайта и зачем она нужна? Netpeak Journal - журнал про интернет-маркетинг и онлайн-бизнес в деталях. URL: <https://netpeak.net/ru/blog/что-такое-логическая-структура-сайта-и-зачем-она-нужна/> (дата звернення: 12.02.23).
29. Elearning SumDU. URL: https://elearning.sumdu.edu.ua/free_content/lectured:89b3d175c06a6b137e410cb14821d0e94549ad5a/20151030211833/44700/index.html (дата звернення: 15.02.23).
30. Бази даних. Нормалізація. Поняття та необхідність застосування. Аномалії модифікації. Приклади | BestProg. BestProg | Програмування: теорія та практика. URL: <https://www.bestprog.net/uk/2020/10/17/databases-normalization-concept-and-necessity-of-application-modification-anomalies-examples-ua/> (дата звернення: 15.02.23).

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		71

31. Wikiwand - Нормалізація баз даних. Wikiwand. URL: https://wikiwand.com/uk/Нормалізація_баз_даних (дата звернення: 15.02.23).

32. What is an Entity Relationship Diagram (ERD)? Lucidchart. URL: <https://www.lucidchart.com/pages/er-diagrams> (дата звернення: 18.02.23).

33. Что такое пользовательский интерфейс: ценность, виды UI и элементы. Продуктовый дизайн (UX/UI), брендинг и аналитика. URL: <https://ux-journal.ru/chto-takoe-polzovatelskij-interfejs.html> (дата звернення: 23.02.23).

34. Пользовательский интерфейс: виды и правила создания. GeekBrains. URL: <https://gb.ru/blog/polzovatelskij-interfejs/> (дата звернення: 23.02.23).

35. Правильная структура сайта: классические составляющие и фишки. Envybox блог. URL: <https://envybox.io/blog/struktura-sajta/> (дата звернення: 24.02.23).

36. HTML Tutorial. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com/html/> (дата звернення: 26.02.23).

37. CSS Tutorial. W3Schools Online Web Tutorials. URL: <https://www.w3schools.com/css/> (дата звернення: 26.02.23).

38. Introduction to Identity on ASP.NET Core. Microsoft Learn: Build skills that open doors in your career. URL: <https://learn.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-7.0&tabs=visual-studio> (дата звернення: 26.02.23).

39. Web Form Validation: Best Practices and Tutorials - Smashing Magazine. Smashing Magazine. URL: <https://www.smashingmagazine.com/2009/07/web-form-validation-best-practices-and-tutorials/> (дата звернення: 20.04.23).

40. Ткачева А. 3 способа валидации форм. HTML Academy. URL: <https://htmlacademy.ru/blog/html/valid-forms> (дата звернення: 20.04.23).

					КВРІПЗ.190131.01.07.ПЗ	Арк.
Зм.	Арк	№ докум.	Підпис	Дата		72

Додаток А
(обов'язковий)

ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

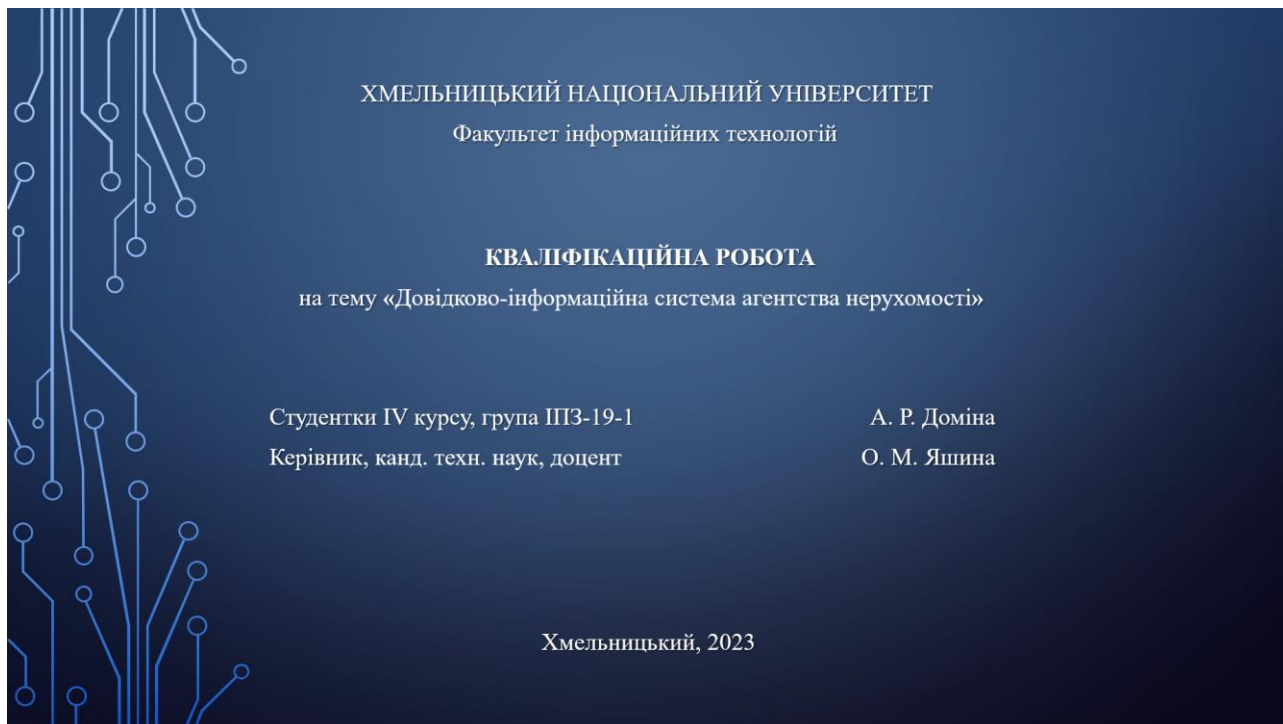


Рисунок А.1 – Титульний слайд, представлення

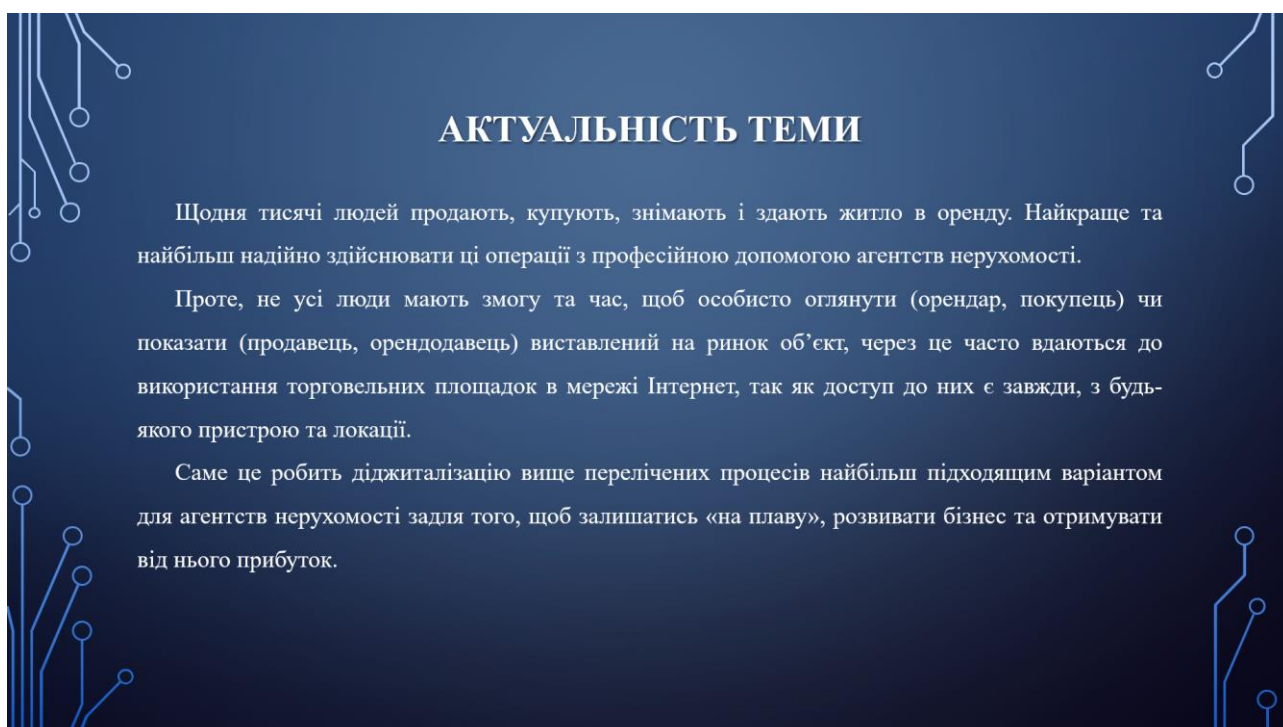


Рисунок А.2 – Актуальність теми

МЕТА І ЗАВДАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Мета роботи:

- спроектувати та розробити програмний продукт, який дозволить користувачам швидко знаходити, оглядати та домовлятися про оренду чи купівлю опублікованої в системі нерухомості в будь-якій точці світу, не переживаючи за безпеку себе і своїх даних.

Завдання роботи:

- провести дослідження предметної області та проаналізувати існуючі рішення;
- визначити особливості та специфіку роботи подібних систем, їх переваги, недоліки, функціональні задачі;
- розробити технічне завдання;
- побудувати архітектуру системи, обрати інструменти та технології, необхідні для розробки;
- розробити сучасний та функціональний інтерфейс;
- виконати програмну реалізацію продукту;
- провести тестування готового продукту.

Рисунок А.3 – Мета і завдання кваліфікаційної роботи

ВИЗНАЧЕННЯ ВИМОГ ДО РОЗРОБЛЮВАНОВОГО ПЗ

Провівши аналіз існуючих рішень, виокремлено список основних характеристик, якими має бути наділена довідково-інформаційна система агентства нерухомості, задля того, щоб досягти успіху та пробитись на верхівку ринку:

- можливість публікації, редагування та видалення пропозицій з продажу та оренди нерухомості;
- сучасний та інтуїтивно зрозумілий дизайн інтерфейсу;
- якісний контент та відсутність надлишкової інформації;
- зручна форма авторизації та реєстрації;
- налаштування особистого кабінету;
- наявність панелі адміністратора.

Рисунок А.4 – Визначення вимог до розроблюваного ПЗ

ДІАГРАМА ВАРІАНТІВ ВИКОРИСТАННЯ

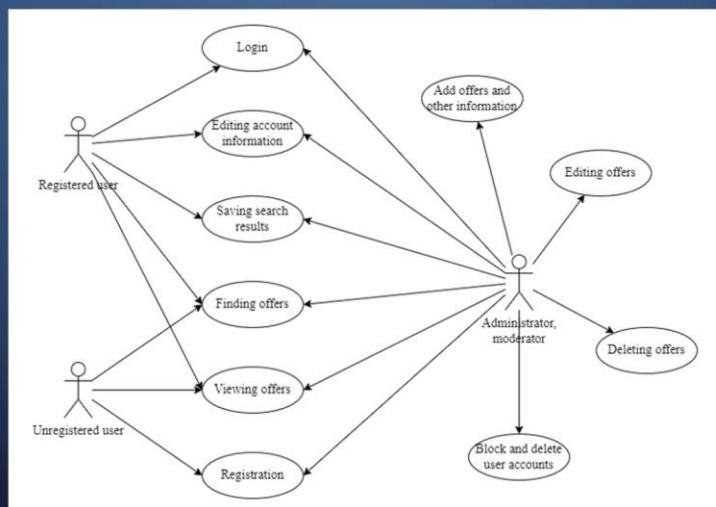
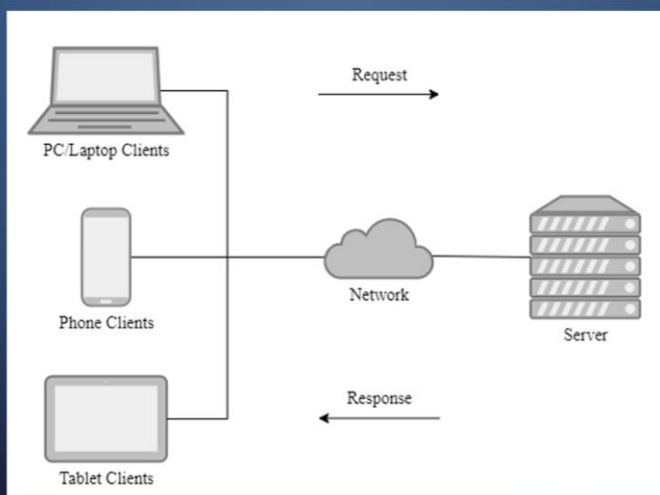


Рисунок А.5 – Діаграма варіантів використання

ВИБІР ТИПУ АРХІТЕКТУРИ



Клієнт-серверна архітектура

Рисунок А.6 – Вибір типу архітектури

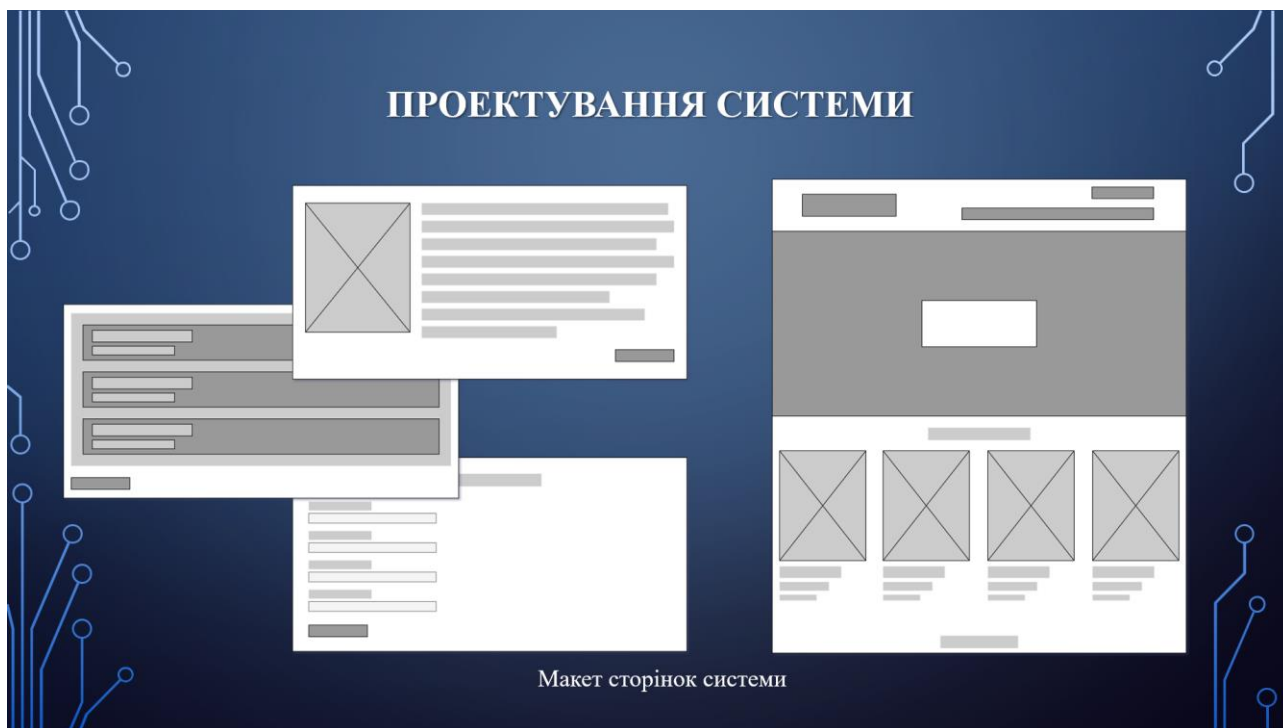


Рисунок А.7 – Проектування системи

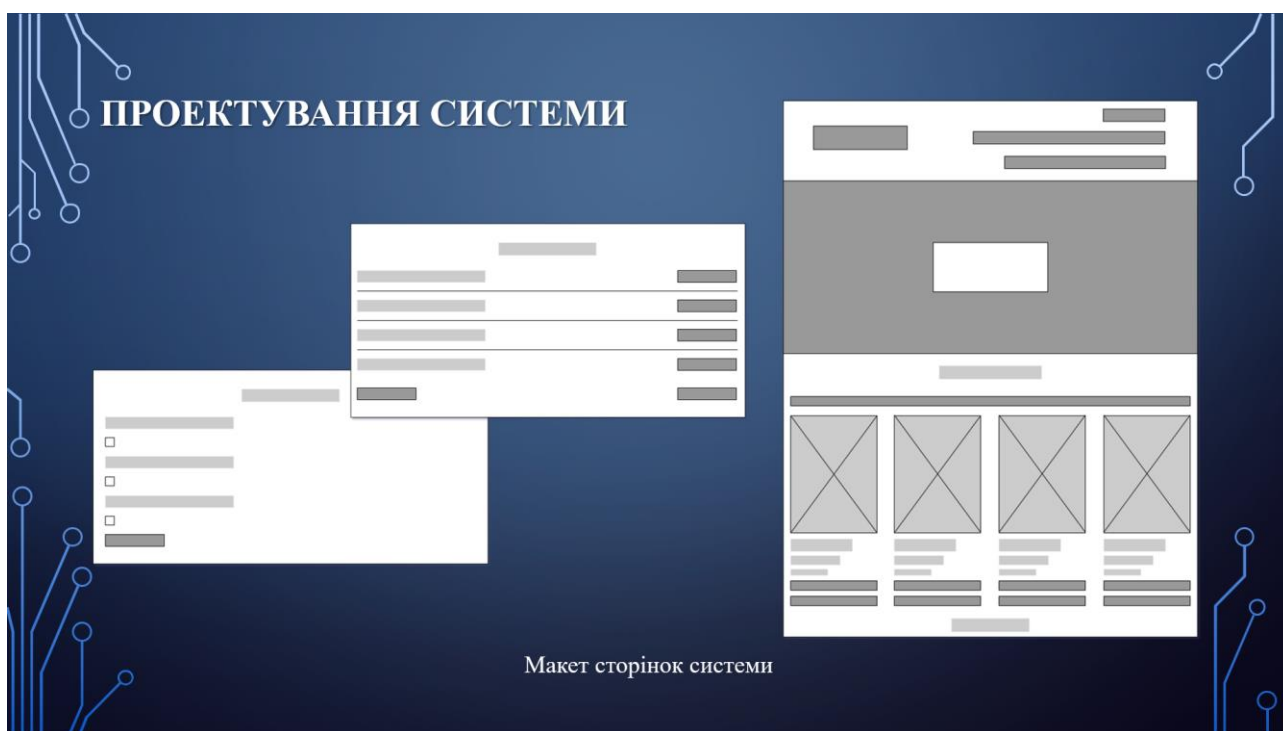
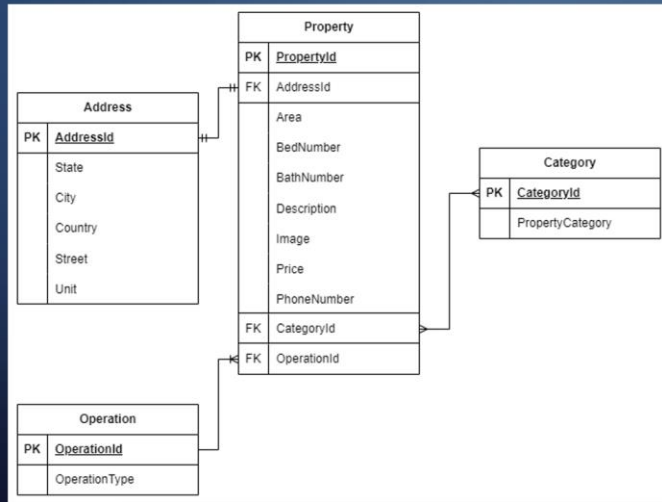


Рисунок А.8 – Проектування системи (слайд 2)

ПРОЕКТУВАННЯ БАЗИ ДАНИХ



ER-діаграма

- ▷ dbo.AspNetRoleClaims
- ▷ dbo.AspNetRoles
- ▷ dbo.AspNetUserClaims
- ▷ dbo.AspNetUserRoles
- ▷ dbo.AspNetUsers

ASP.NET Identity

Рисунок А.9 – Проектування бази даних

Unreal Estate | 123-456-7890 | Offers | Categories | Login | Register

DREAM BIG. LIVE WELL.

We Offer

- Woodgrove at Sterlington | 950 Woodmont Ln, Sloatsburg, NY 10974 | 3 249,00 \$
- Fairview Apartments | 100 Fairview Sq, Hiram, NY 14850 | 2 560,00 \$

РЕАЛІЗАЦІЯ СИСТЕМИ

Сторінка перегляду даних про товар

Woodgrove at Sterlington
950 Woodmont Ln, Sloatsburg, NY 10974
This apartment is located at 950 Woodmont Ln, Sloatsburg, NY. 950 Woodmont Ln is in Sloatsburg, NY and in ZIP code 10974.
Amenities: 2 beds, 2 baths, 1160 sqft
Price: 3 249,00 \$

[Request a call](#)

Головна сторінка звичайного користувача

Registration

Email:

Name:

First User:

Password:

Confirm password:

[Register](#)

Рєєстрація

Рисунок А.10 – Реалізація системи

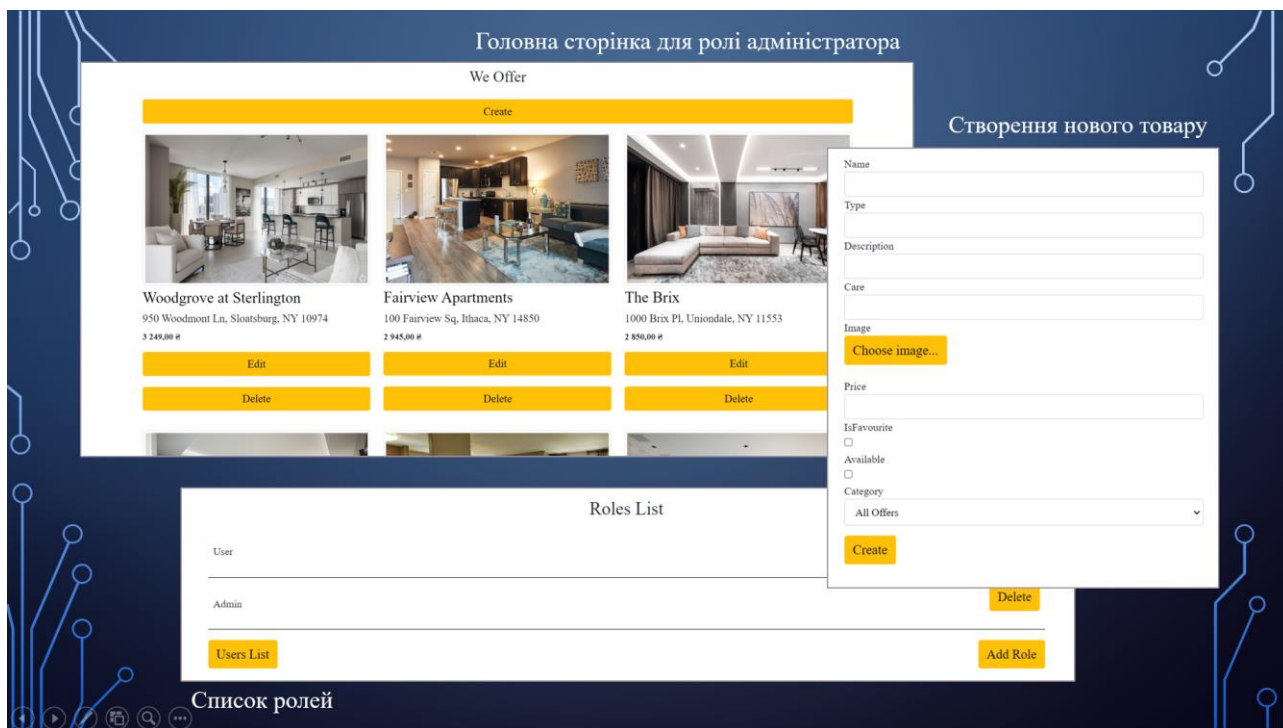


Рисунок А.11 – Реалізація системи (слайд 2)

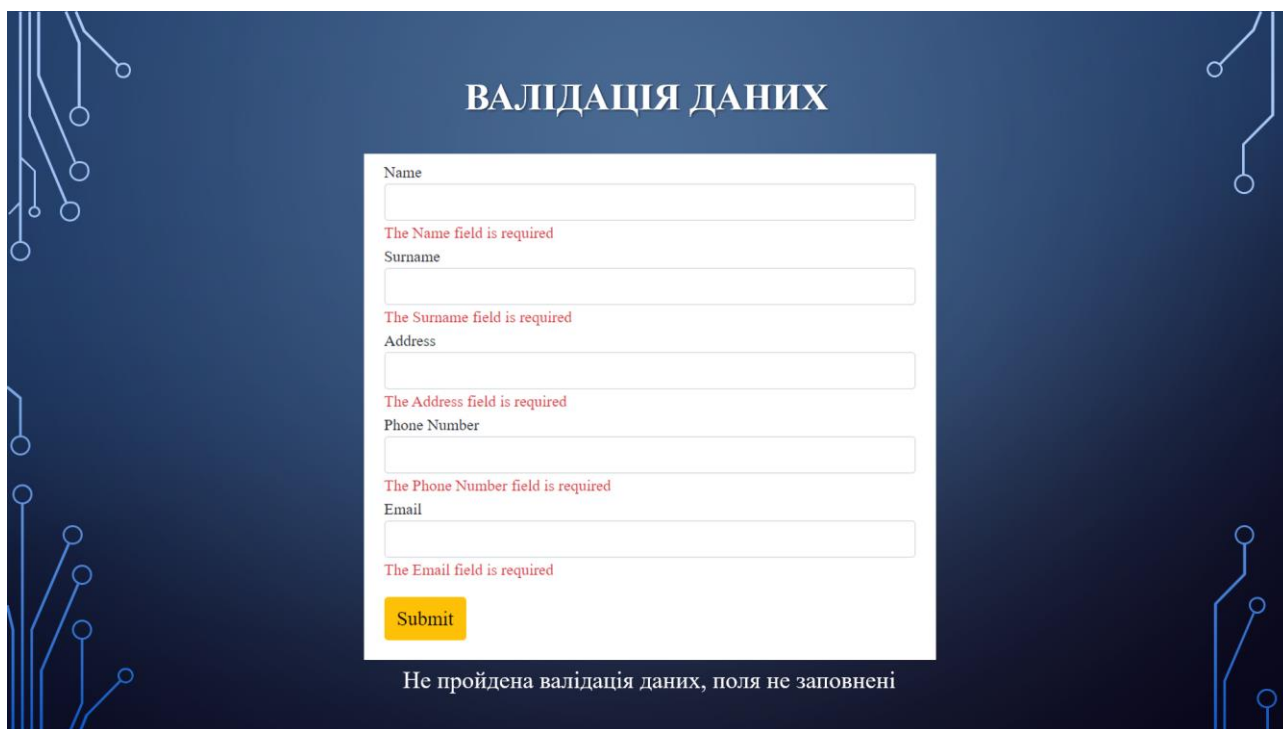


Рисунок А.12 – Валідація системи



Рисунок А.13 – Дякую за увагу

Додаток Б
(обов'язковий)

ТЕХНІЧНЕ ЗАВДАННЯ

Введення

Робота виконується в рамках проекту розробки довідково-інформаційної системи агентства нерухомості.

1 Підстава для розробки

Підставою для розробки є «Завдання на кваліфікаційну роботу», затверджене завідувачем кафедри інженерії програмного забезпечення.

Найменування розробки: Довідково-інформаційна система агентства нерухомості.

2 Призначення розробки

Довідково-інформаційна система агентства нерухомості призначена для забезпечення користувачів можливістю швидкого пошуку, огляду та укладення договору про оренду чи купівлю опублікованої на сайті нерухомості в будь-якій точці світу та з будь-якої точки світу, не переживаючи за безпеку себе і своїх даних.

Користувачами системи є працівники агентства нерухомості та люди, які знаходяться в активному пошуку житла для оренди або купівлі.

Пошук нерухомості відбувається шляхом вказання таких параметрів, як приблизний регіон, в межах якого повинно знаходитись потенційне житло. В якості додаткових параметрів для підбору можуть бути вказані ціновий діапазон, кількість спальних місць та ванних кімнат.

Окрім цього, функціональне призначення передбачає: оцінювання існуючих на сайті пропозицій та занесення їх в список збережених для легшого повторного знаходження, реєстрація та авторизація користувачів, редагування даних облікового запису та його деактивація.

Додавання нових публікацій, редагування або видалення існуючих, перевірка контенту на доцільність, керування аккаунтами користувачів, їх видалення та блокування відбувається адміністраторами і модераторами.

Щодо експлуатаційного призначення, система може використовуватися на будь-якому персональному комп'ютері, у будь-якому встановленому браузері, ніяких додаткових налаштувань не потребує.

3 Вимоги до програми

3.1 Вимоги до функціональних характеристик

Довідково-інформаційна система агентства нерухомості повинна забезпечувати можливість:

- публікації, редагування та видалення пропозицій з продажу та оренди нерухомості;
- зручного пошуку об'єктів нерухомості;
- авторизації та реєстрації користувача з можливістю входу через соціальні мережі;
- налаштування особистого кабінету та деактивації профілю;
- керування системою від імені адміністратора: блокування та видалення аккаунтів, створення, редагування та приховування старих публікацій.

Вимоги до інтерфейсу:

- сучасний та інтуїтивно зрозумілий дизайн інтерфейсу;
- якісний контент та відсутність надлишкової інформації.

3.2 Вимоги до надійності

Розроблюване ПЗ повинно мати:

- можливість самовідновлення після збоїв, таких як відключення електроживлення;
- обмеження несанкціонованого доступу до даних;
- розмежування прав користувачів у системі;
- контроль інформації, що вводиться, та блокування некоректних дій користувача при роботі з системою.

3.3 Вимоги до складу та параметрів технічних засобів

Мінімальні вимоги для функціонування системи повинні відповідати вимогам будь-якого браузера. Нижче наведено приклад мінімальних вимог для браузера Google Chrome:

- стабільне з'єднання з Інтернетом;
- операційна система: Windows 10, 8, 8.1, 7;
- розрядність: x86 (32-bit) або x64 (64-bit);
- процесор: Pentium 4 з SSE2;
- вінчестер: 350 Mb;
- відеоадаптер: 3D адаптер nVidia, Intel, AMD / ATI;
- оперативна пам'ять: 512 Mb;
- будь-яка аудіокарта;
- роздільна здатність екрану: SVGA 800x600;
- контролери: клавіатура, миша.

3.4 Вимоги до інформаційної та програмної сумісності

Для створення серверної частини будуть використовуватися такі технології:

- об'єктно-орієнтована мова програмування C#;
- фреймворк ASP.NET Core;
- патерн проектування Model-View-Controller;
- технологія доступу та керування реляційними даними, та перетворення їх у об'єктно-орієнтовану структуру Entity Framework Core.

Для створення клієнтської частини використовуватимуться такі технології:

- мова розмітки HTML;
- таблиці стилів CSS;
- мова програмування JavaScript.

Для роботи з базою даних буде використовуватися мова SQL та система керування базами даних SQL Server.

3.5 Вимоги до транспортування та зберігання

Програма, її документація поставляються у цифровому вигляді. Умови експлуатації програмного забезпечення збігаються з умовами експлуатації серверу, на якому буде розміщене ПЗ.

3.6 Спеціальні вимоги

Програма повинна мати дружній інтерфейс, розрахований на користувача середньої кваліфікації (з точки зору комп'ютерної грамотності). Мова програмування визначається вибором виконавця.

4 Вимоги до програмної документації

В ході розробки програми повинні бути підготовлені:

- технічне завдання;
- опис програми;
- текст програми з коментарями та поясненнями;
- методика випробувань;
- відомості про функціонування програми;
- керівництво користувача;
- керівництво програміста.

5 Стадії та етапи розробки

Розробка програмного продукту проходить декілька стадій і етапів, які представлені в таблиці Б.1.

Таблиця Б.1 – Стадії та етапи розробки програмного продукту

Стадія розробки	Етап робіт	Зміст робіт
1	2	3
Технічне завдання	Обґрунтування необхідності розробки програми	Коротка характеристика програмного забезпечення; підстава і призначення розробки; вимоги до програмної системи і документація; стадії і етапи розробки програми; порядок контролю і приймання.
Ескізний проект	Розробка ескізного проекту	Попередня розробка структури вхідних і вихідних даних; уточнення середовища програмування; розробка і опис загальної алгоритмічної структури системи, що буде розроблюватися.

Кінець таблиці Б.1

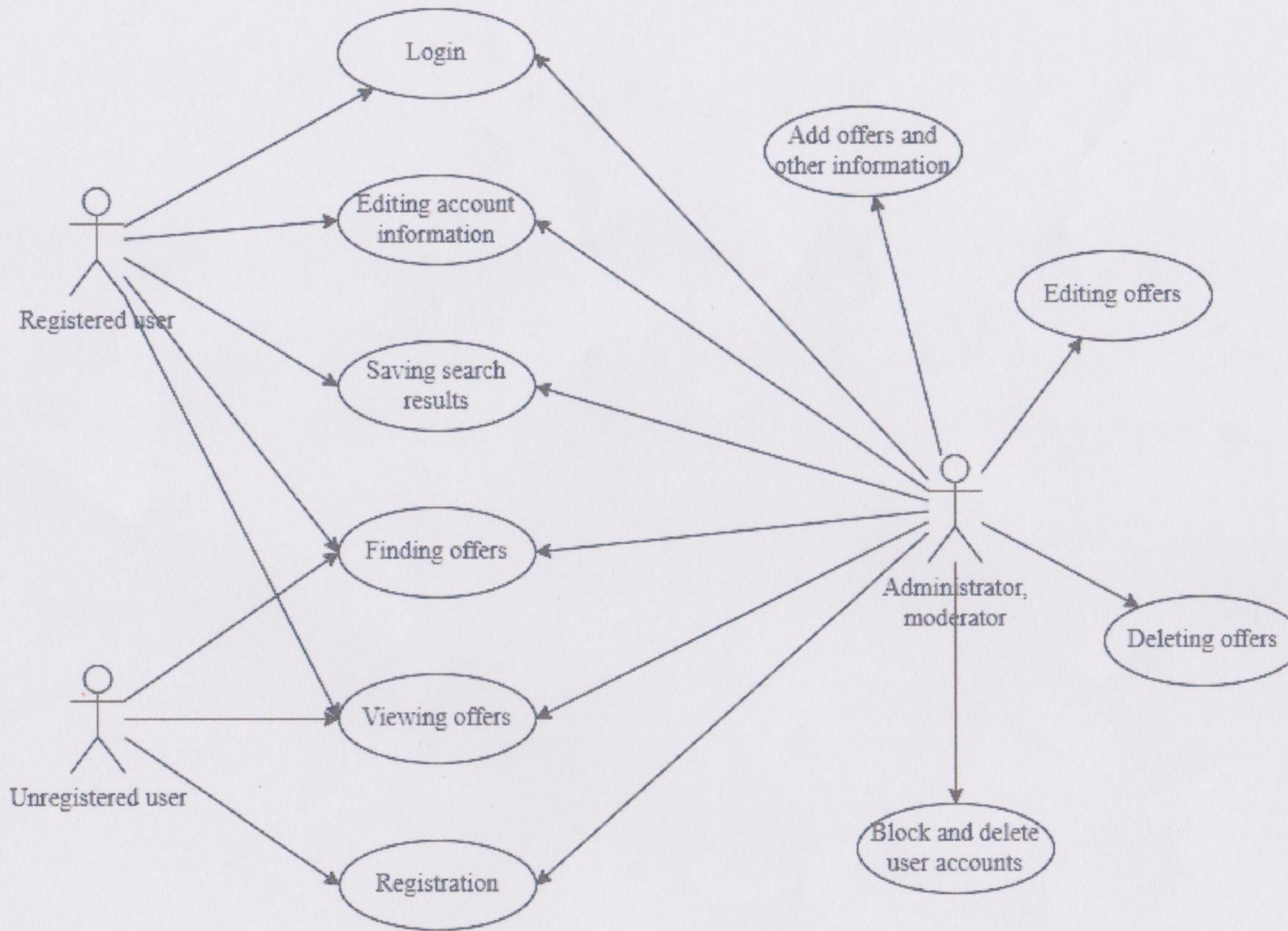
1	2	3
Технічний проект	Розробка технічного проекту	Уточнення структури вхідних і вихідних даних; розробка докладного алгоритму; розробка структури програми; остаточне визначення конфігурації технічних засобів.
Робочий проект	Розробка програмного забезпечення	Реалізація програмного забезпечення; відладка; проведення попереднього тестування.
Розробка програмної документації	Розробка документації до програмного забезпечення	Розробка необхідної документації, передбаченої технічним завданням.
Тестування системи	Проведення тестування програмного забезпечення	Розробка методики тестування; проведення основних тестів; коректування програмного забезпечення.
Впровадження	Підготовка і передача програми	Підготовка і передача програмного забезпечення; навчання персоналу використуванню програмного забезпечення; внесення коректувань в програмне забезпечення і документацію.

6 Порядок контролю та приймання

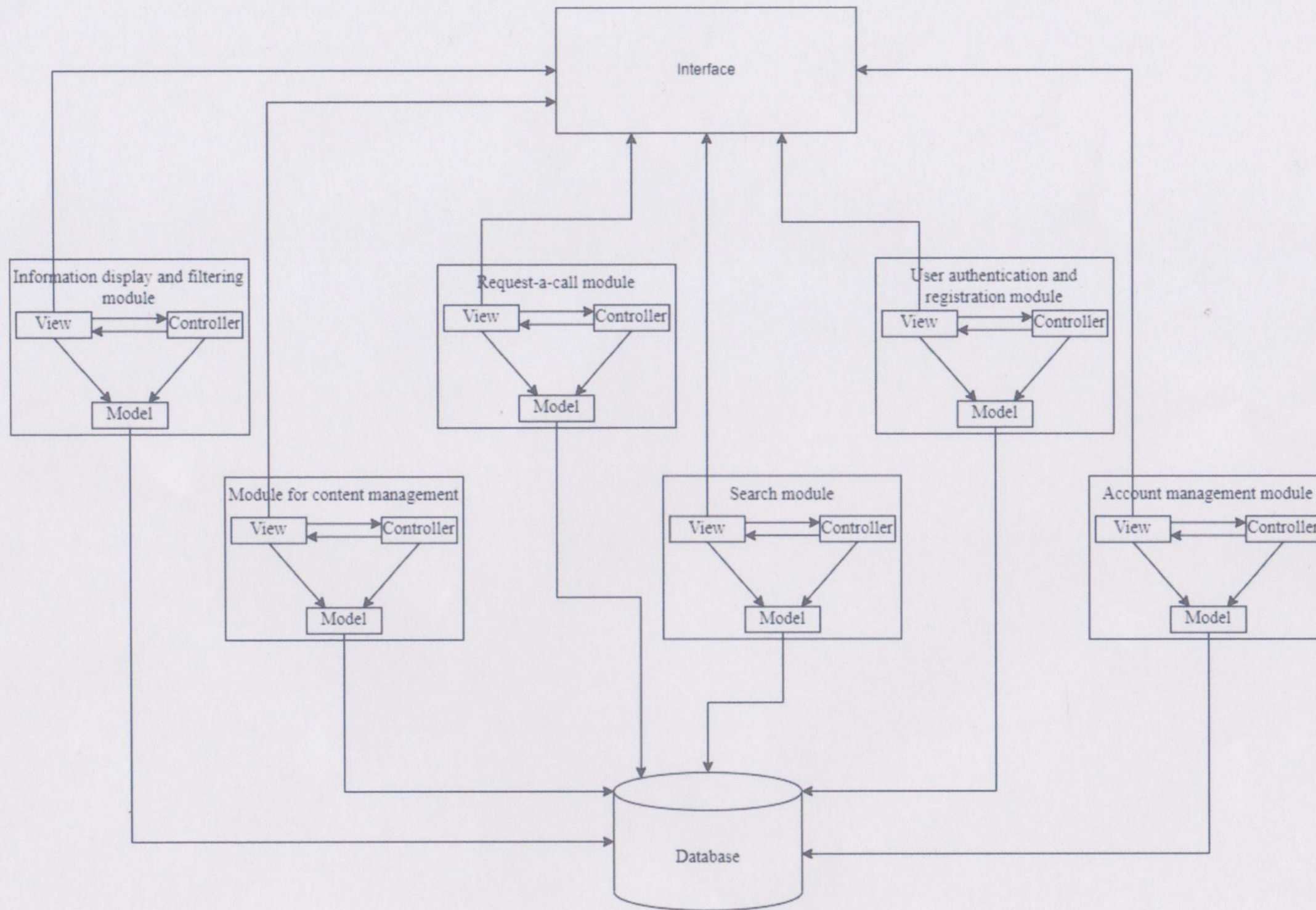
Контроль здійснюється кінцевими користувачами системи, підключеними на етапі тестування. Прийом системи здійснюється після її повного розгортання та встановлення на хостинг і налаштування для нормального функціонування.

Після закінчення розробки системи повинні бути проведені тестування на захист від некоректного введення.

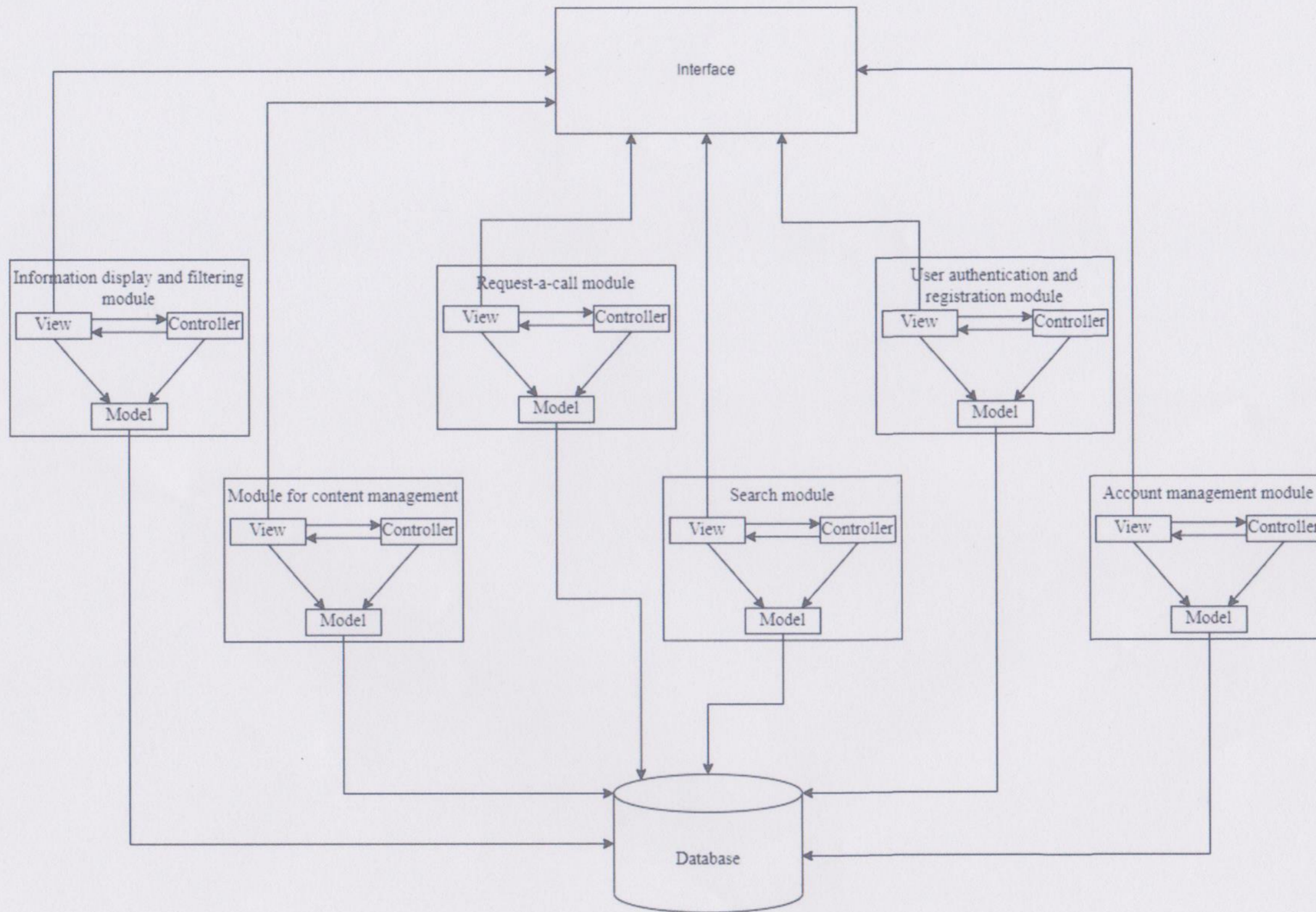
ГРАФІЧНА ЧАСТИНА



					КвРІПЗ.190131.01.07.Е8			
Зм.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система агентства нерухомості	Літера	Маса	Масштаб
Розробив		Доміна А. Р.	<i>[Signature]</i>	5.06				
Керівник		Яшина О. М.	<i>[Signature]</i>	5.06				
Консульта.					Діаграма варіантів використання	Аркуш 1	Аркуші 3	
Н. Контр.		Радельчук Г. І.	<i>[Signature]</i>	0.06		ХНУ, ІПЗ-19-1		
Зав. каф.		Бедратюк Л. П.	<i>[Signature]</i>	6.06				



				КвРІПЗ.190131.01.07.Е8				
Зм.	Арк.	№ докум.	Підпис	Дата	Довідково-інформаційна система	Літера	Маса	Масштаб
Розробив		Доміна А. Р.	<i>AD</i>	01.06	агентства нерухомості			
Керівник		Яшина О. М.	<i>OM</i>	15.06	Модульна структура ДІС	Аркуш 2	Аркуші 3	
Консульт.								
Н. Контр.		Радельчук Г. І.	<i>GR</i>	01.06		ХНУ, ІПЗ-19-1		
Зав. каф.		Бедратюк Л. П.	<i>LP</i>	01.06				



					КвРІПЗ.190131.01.07.Е8			
					Довідково-інформаційна система	Літера	Маса	Масштаб
Зм. Арк.	№ докум.	Підпис	Дата		агентства нерухомості			
Розробив	Доміна А. Р.	<i>[Signature]</i>	01.06		Модульна структура ДІС	Аркуш 2	Аркуші 3	
Керівник	Яшина О. М.	<i>[Signature]</i>	15.06					
Консульт.								
Н. Контр.	Радельчук Г. І.	<i>[Signature]</i>	01.06			ХНУ, ІПЗ-19-1		
Зав. каф.	Бедратюк Л. П.	<i>[Signature]</i>	06					

СУПРОВІДНІ ДОКУМЕНТИ

Завідувачу кафедри інженерії програмного
забезпечення проф. Бедратюку Л. П.

здобувача вищої освіти

Доміної А. Р.

Прізвище, ініціали

факультет ІТ, 4 курс, група ІІЗ-19-1

ЗАЯВА

З правилами чинного Положення «Про систему забезпечення академічної доброчесності в Хмельницькому національному університеті» від 01.07.2022, згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів дисциплінарної та академічної відповідальності, ознайомена. Про використання програмно-технічних засобів для перевірки кваліфікаційних робіт здобувачів вищої освіти на плагіат оповіщена та надаю свою згоду на обробку й збереження університетом моєї роботи в інституційному репозитарії університету.

Також надаю університету право на передачу моєї роботи для обробки та збереження в базах даних програмно-технічних засобів (Unicheck та/або Anti-Plagiarism) і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються програмно-технічними засобами та користувачами, що мають доступ до цих програмно-технічних засобів, виключно в обмежених цілях для виявлення плагіату в текстах робіт.

Робота для перевірки університетом надається в друкованому та електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

05.02.23

дата



підпис

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 18.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилоч в документах: 13%

ID: 114846 Назва: БКР Довідково-інформаційна система агентства нерухомості Додано в БД: 2023-06-05 Автора: Доміна А.Р. Керівники: Яшина О.М. к.т.н. доц. Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	63612	644	14802 (23%)	172 (27%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми
111878	Назва: Звіт з переддипломної практики бакалавр на тему: Довідково-інформаційна система агентства нерухомості Додано в БД: 2023-03-03 Автора: Доміна А.Р. Керівники: Яшина О.М. Консультанти: Опоненти:	11521 (18.0%)	132 (20.0%)

Ім'я користувача:
Кафедра ІПЗ

ID перевірки:
1015441074

Дата перевірки:
05.06.2023 19:29:22 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
05.06.2023 19:44:46 EEST

ID користувача:
100005589

Назва документа: КвР для перевірки на плагіат

Кількість сторінок: 69 Кількість слів: 10149 Кількість символів: 80747 Розмір файлу: 23.48 MB ID файлу: 1015101710

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

16.3%
Схожість

Найбільша схожість: 3.51% з джерелом з Бібліотеки (ID файлу: 1008215933)

14.6% Джерела з Інтернету

770

Сторінка 71

5.35% Джерела з Бібліотеки

119

Сторінка 76

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

1

Підозріле форматування

19
сторінок

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ
освітнього ступеня «Бакалавр»

Дипломник Доміна Анастасія Романівна

Тема Довідково-інформаційна система агентства нерухомості

Спеціальність 121 – Інженерія програмного забезпечення

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3; кількість сторінок записки 88

1. Короткий зміст пояснювальної записки та прийнятих рішень У кваліфікаційній роботі досліджено і проаналізовано предметну область, визначено усі функціональні та нефункціональні вимоги. Проведений аналіз існуючих рішень на ринку, розглянуто їх переваги і недоліки та доведено актуальність розробки нового програмного забезпечення. Розглянуто інструменти для реалізації спроектованих рішень, в результаті чого було обрано інструментарій для створення програмного забезпечення. Реалізація програмного забезпечення було здійснено на мові програмування С#, з використанням шаблону проектування MVC, системи керування базами даних MySQL та HTML і CSS для розробки інтерфейсу користувача Також проведено тестування програми, за результатами якого доведено, що розроблене програмне забезпечення працює коректно та готове до експлуатації.

2. Висновок про відповідність роботи поставленому завданню Кваліфікаційна робота виконана відповідно до поставленого завдання та з дотриманням всіх вимог.

3. Характеристика виконання кожного розділу роботи, ступінь використання останніх досягнень науки і техніки та передових методів роботи У вступі доведено актуальність теми, визначено мету та завдання дипломного проектування. У першому розділі проведено аналіз предметної області, розглянуто існуючі рішення та визначені функціональні і нефункціональні вимоги до розроблюваного програмного забезпечення. У другому розділі визначено тип архітектури та шаблон проектування, які використовувались для розробки довідково-інформаційної системи, виділено ряд сутностей і атрибутів, нормалізовано дані та на основі них спроектовано логічну модель БД. Також, спроектовано інтерфейс користувача, розроблено макети сторінок, визначено тип структури веб-сайту та розроблено логічну структуру ДІС. Також, вибрано технології для розробки, за допомогою яких відбувалась реалізація системи. У третьому розділі виконано реалізацію бази даних та основних модулів системи. Відповідно до них розроблено інструкцію користувача. Також визначені вимоги до апаратно-програмних засобів, що забезпечать коректну роботу системи, і проведено валідацію форм введення даних.

4. Позитивні сторони роботи Тематика кваліфікаційної роботи є завжди актуальною. У роботі застосовані новітні технології для побудови програмного продукту та актуальне архітектурне рішення.

5. Негативні сторони роботи До системи було б доцільно додати розширений пошук і забезпечити надання більш повної інформації та можливість завантаження фотографій на сторінку деталей про об'єкти нерухомості, а також створити сторінку з особистими даними користувача.

6. Оцінка графічного оформлення та пояснювальної записки Графічне оформлення виконане відповідно до теми кваліфікаційної роботи та подане у вигляді діаграм і рисунків. Пояснювальна записка оформлена згідно вимог чинних стандартів.

7. Відгук про кваліфікаційну роботу в цілому Кваліфікаційна робота заслуговує позитивної оцінки. Матеріал пояснювальної записки структурований, послідовний, чіткий та простий, що дозволяє чітко зрозуміти викладений матеріал у межах тематики проектування. Графічний матеріал надає можливість наочно побачити деталі проектування системи.

8. Інші зауваження _____

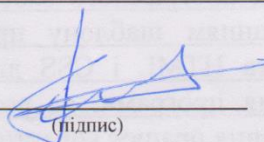
9. Оцінка кваліфікаційної роботи Кваліфікаційна робота виконана у повному обсязі, відповідає поставленій задачі та заслуговує на оцінку «добре».

РЕЦЕНЗЕНТ Бобровнікова Кіра Юліївна, кандидат технічних наук, доцент кафедри комп'ютерної інженерії та інформаційних систем (ККІС) ХНУ .

“ 06”

06

2023 р.



(підпис)

**РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ**

Підтверджуємо ознайомлення з результатом звіту подібності щодо роботи, генерованого системою виявлення текстових збігів/ідентичності/схожості:

Назва: «Довідково-інформаційна система агентства нерухомості»

Автор: Доміна Анастасія Романівна

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Освітньо-професійна програма «Інженерія програмного забезпечення»

Науковий керівник: Яшина Оксана Миколаївна, кандидат технічних наук, доцент

Після аналізу звіту подібності зроблено такий висновок:

№	Висновок	Позначка про відповідність
1	Запозичення, виявлені в роботі, є законними і не є плагіатом. Робота приймається до захисту.	відповідає
2	Виявлені запозичення не є плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота приймається до захисту, але має бути відкоригована. Відкоригований варіант має бути поданий на кафедру за 2 дні до захисту, разом із заявою щодо самостійності виконання письмової роботи та ідентичності друкованої й електронної версії роботи	
3	Виявлені запозичення не є плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. В зв'язку з цим мета роботи та поставлені завдання не були досягнені. Робота може бути допущена до захисту (наступного року) після того, як буде відкоригована та допрацьована і успішно пройде повторну перевірку на академічний плагіат.	
4	Робота містить навмисні текстові спотворення, передбачувані спроби укриття запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
5	Інше:	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

1) у тексті кваліфікаційної роботи системами перевірки на плагіат виявлено схожість з деякими документами в частині загальноживаних обов'язкових словосполучень у стандартних бланках (титулка, відомість документів), у структурі змісту, назвах розділів/підрозділів тощо, у назвах публікацій у переліку джерел посилання;

2) в якості запозичень системою було зафіксовано деякі послідовності вихідного коду і посилання на бібліотеки, які є стандартними мовними конструкціями програмування та не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;

3) усі запозичення є фрагментарними або мають належним чином оформленні посилання;

4) виявлені модифікації тексту не впливають на відсоток схожості.

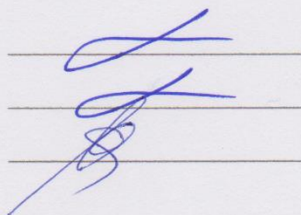
Сумарний обсяг всіх запозичень, визначений системою виявлення збігів ідентичності/ схожості, складає 18% і адресується до 770 джерел, що, з урахуванням наведених обґрунтувань, відповідає характеру теми і свідчить на користь кваліфікаційної роботи.

Дата 6.06.23

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи



Леонід БЕДРАТЮК

Леонід БЕДРАТЮК

Оксана ЯШИНА