

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему:

Розробка інформаційної системи для оренди житлових приміщень

Виконав: студент групи 2147 ст

_____ *Жевак О. О.*
(підпис)

Керівник роботи:

Доцент, к.т.н.
(посада, науковий ступінь вчене звання)

_____ *Ажицев В. Ф.*
(підпис)

м. Миколаїв – 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"
Гарант освітньої
програми

к.т.н, доц. Гайда А.Ю.
"_____"
2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "бакалавр"

Студенту Жеваку Олександр Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи:

Розробка інформаційної системи для оренди житлових приміщень

Керівник роботи

Ажищев В. Ф.

Затверджені наказом ректора № 243-уч від "18" березня 2024 року.

2. Термін подання роботи: 20 червня 2024 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації,
літературні джерела, технічна документація на існуючі аналоги систем,

4. Перелік питань, що належать до розробки (найменування розділів)

5. Перелік презентаційних матеріалів

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ажищев В.Ф		
2	Ажищев В.Ф		
3	Ажищев В.Ф		
4	Ажищев В.Ф		

7. Дата видачі завдання 05 лютого 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Анотації та вступ	10.02.2024	
2	Розділ 1	25.02.2024	
3	Розділ 2	17.03.2024	
4	Розділ 3	02.04.2024	
5	Розділ 4	25.04.2024	
6	Висновки	06.05.2024	
7	Підготовка презентаційних матеріалів	23.05.2024	
8	Підготовка до виступу	10.06.2024	
9			
10			

Студент

(підпис)

Жевак О. О.

(ПІБ)

Керівник роботи

(підпис)

Ажищев В.Ф.

(ПІБ)

Анотація

Кваліфікаційна робота: “Розробка інформаційної системи для оренди житлових приміщень” має об’єм 59 сторінок, містить розділи - 4, малюнки - 26, використані джерела – 23.

Об’єкт – організація що надає послуги з оренди.

Предмет – процес пошуку та здачі в оренду орендних приміщень.

Задача – розробка інформаційної системи.

Мета – підвищення ефективності процесу оренди житлових приміщень за рахунок впровадження інформаційної системи з оренди житлових приміщень.

Актуальність – попит на ринку систем для цих задач на сьогодні великий.

Новизна – створення сучасної та якісної системи, що забезпечує покращення ефективності процесу оренди за рахунок використання сучасних технологій.

Результатом проекту є створення інформаційної системи, яка автоматизує процеси оренди житлових приміщень, забезпечуючи зручний інтерфейс для користувачів та високу ефективність управління даними. Ця система дозволяє орендодавцям розміщувати оголошення про оренду, а орендарям – знаходити необхідне житло, бронювати його та керувати орендою онлайн.

Ключові слова: оренда, житло, інформаційна система, бронювання, веб-додаток, бази даних, алгоритм, блок-схема, технології.

Annotation

The qualification work titled "Development of an Information System for Residential Rental" is 59 pages long, includes 4 sections, 26 figures, and references 23 sources.

Object: An organization providing rental services.

Subject: The process of searching and renting residential properties.

Task: Development of the information system.

Goal: Improving the efficiency of the residential rental process and convenience for users.

Relevance: There is a high demand in the market for systems designed for these tasks today.

Novelty: Creating a modern and high-quality system that improves the efficiency of the rental process through the use of modern technologies.

The result of the project is the creation of an information system that automates the processes of residential rental, providing a convenient interface for users and high efficiency in data management. This system allows landlords to post rental listings and tenants to find the necessary housing, book it, and manage the rental online.

Keywords: rental, housing, information system, booking, web application, databases, algorithm, flowchart, technologies.

Зміст

Вступ	8
РОЗДІЛ 1 Розробка інформаційної системи для оренди житлових приміщень	10
1.1 Опис об'єкта дослідження і аналіз предметної області.....	10
1.2 Формування вимог до проекту.....	15
1.3 Розробка концепцій інформаційної системи	16
РОЗДІЛ 2 Розробка технічного проекту інформаційної системи	22
2.1 Рішення по інформаційному забезпеченню.....	22
2.1.1 Розробка концептуальної моделі даних	25
2.1.2 Розробка логічної моделі даних	26
2.1.3 Розробка фізичної моделі даних	28
2.2 Рішення по математичному забезпеченню	29
2.3 Рішення по програмному забезпеченню	33
2.3.1 База даних	33
2.3.2 Серверна частина	34
2.3.3 Клієнтська частина.....	35
2.4 Рішення по технічному забезпеченню	38
РОЗДІЛ 3 Реалізація проекту інформаційної системи.....	40
РОЗДІЛ 4 Охорона праці та техніка безпеки	45
4.1 Вимоги до умов праці в офісних приміщеннях.....	45
4.2 Пожежна безпека.....	46
4.3 Профілактика професійних захворювань	46
4.4 Робота за комп'ютером	47
ВИСНОВКИ.....	48
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	49
ДОДАТОК А.....	51
ДОДАТОК Б	55

Вступ

Кваліфікаційна робота виконана по темі “Розробка інформаційної системи для оренди житлових приміщень”.

Актуальність теми дипломної роботи полягає в тому, що існує попит на ринку програмних продуктів для оренди житлових приміщень.

Об’єктом проектування є організація, що надає послуги з оренди житлових приміщень. Предметом проектування є процеси пошуку, бронювання та розміщенням оренди житла.

Метою проектування є створення інформаційної системи, що підвищить ефективність процесу оренди житлових приміщень та запровадить зручність роботи для користувачів.

Задача кваліфікаційної роботи полягає в реалізації системи згідно заданої мети.

У першому розділі проведена розробка концепції інформаційної системи. Зроблений опис об’єкта дослідження і аналіз предметної області. Сформовані вимоги до інформаційної системи. На основі сформованих вимог розроблена концепція інформаційної системи.

У другому розділі розроблений технічний проект інформаційної системи. Наведені загальносистемні рішення для реалізації проекту. В рішеннях по інформаційному забезпеченню представлені концептуальна, логічна та фізична моделі бази даних. У математичному забезпеченні проведений математичний аналіз алгоритмів для різних типів користувачів. У програмному забезпеченні описані засоби та технології, за допомогою яких була проведена розробка проекту. У технічному забезпеченні описані характеристики пристроїв користувачів для експлуатації інформаційної системи.

У третьому розділі представлені інструктивно методичні матеріали та набір доступних операцій.

В четвертому розділі описана охорона праці та техніка безпеки.

Новизна роботи полягає в створенні сучасної та якісної системи, що забезпечує покращення ефективності процесу оренди за рахунок використання сучасних технологій..

РОЗДІЛ 1 Розробка інформаційної системи для оренди житлових приміщень

1.1 Опис об'єкта дослідження і аналіз предметної області

Предметом даного дослідження є процес оренди житлових приміщень, а також управління цим процесом в організації, що спеціалізується на наданні таких послуг.

Організація має таку структуру управління: на керівній позиції знаходиться директор, під його керівництвом працюють відділи з реклами, продажу, обслуговування клієнтів та бухгалтерії [1].

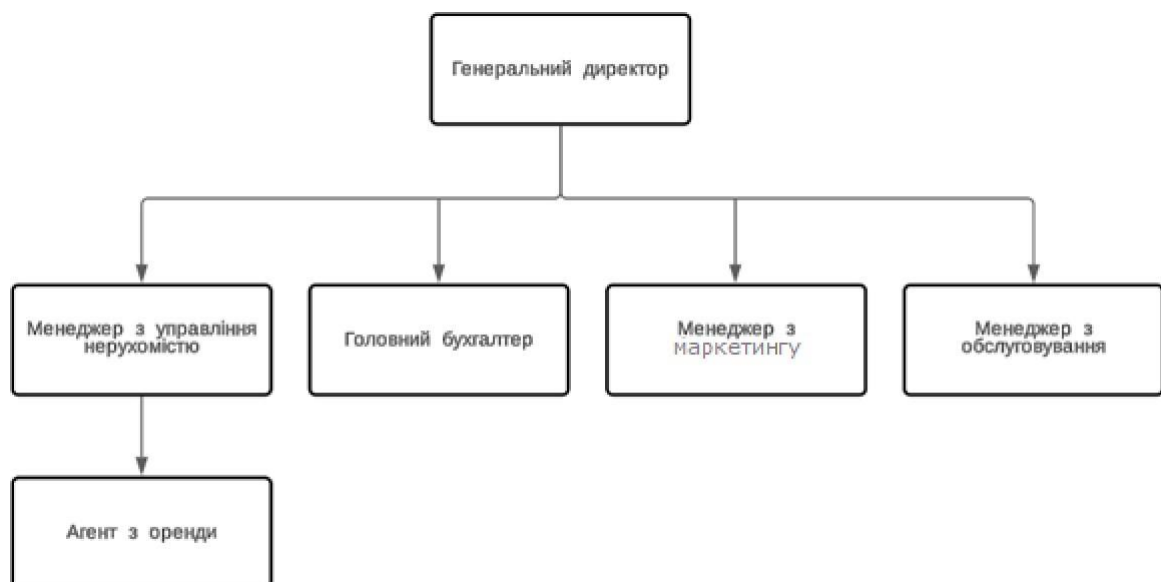


Рисунок 1.1 - Система управління організації

Організація спеціалізується на наданні послуг з пошуку оренди житлових приміщень. Її основними напрямками діяльності є здійснення пошуку клієнтів для оренди, укладання договорів оренди, обслуговування клієнтів та вирішення поточних питань, пов'язаних з орендою [2].

Оскільки існуюча система управління орендою не забезпечує ефективної комунікації та взаємодії між відділами, виникає потреба в розробці застосунку, який полегшить взаємодію з клієнтами та покращить ефективність управління процесом пошуку оренди.

Сьогодні лідерами ринку оренди квартир є OLX, Flatfy і booking. Всі три сервіси є гігантами цієї галузі і займають більшу частину ринку. Але навіть вони мають свої недоліки [3].

OLX-це веб-сайт або додаток для iOS та Android. OLX пропонує безліч категорій (рис. 1.2) Це універсальний сайт, де ви можете знайти майже все, що вам потрібно.



Рисунок 1.2 - Головні рубрики сайту OLX

OLX-це не нова система (раніше SLANDO), і сьогодні вона не має найкращої репутації. З нею пов'язані сотні випадків шахрайства. Отже, принаймні сайт має хороші характеристики (рис. 1.3) і широким вибором квартир (рис. 1.4), однак через його репутацію та той факт, що шахраї щороку винаходять нові способи обману, послуги сайту вважаються недуже надійними. Також існує обмеження на кількість безкоштовних оголошень на сайті.

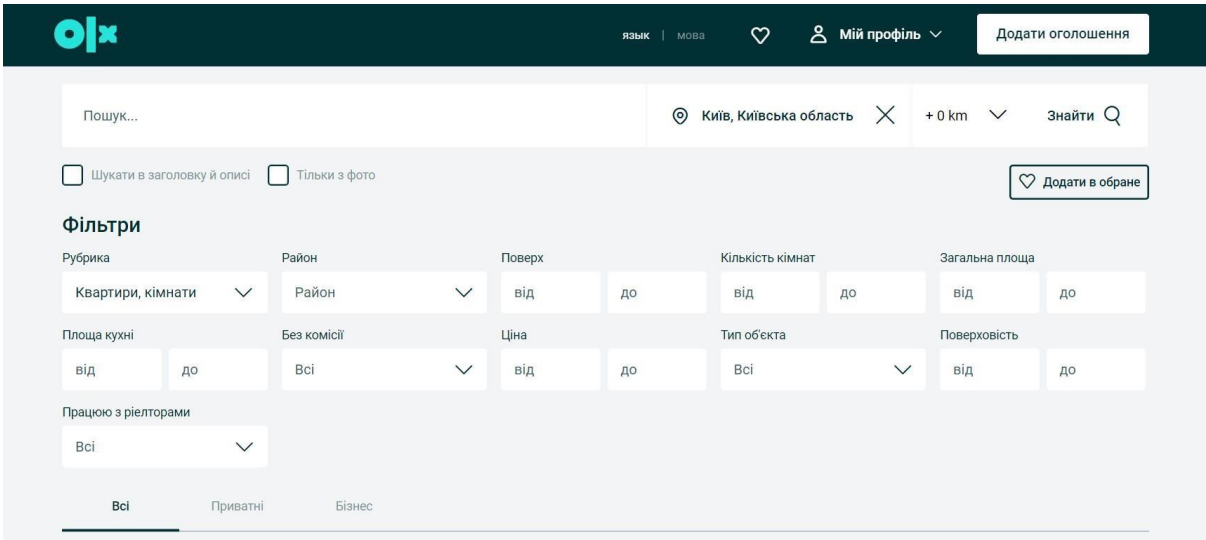


Рисунок 1.3 - Фільтри сайту OLX

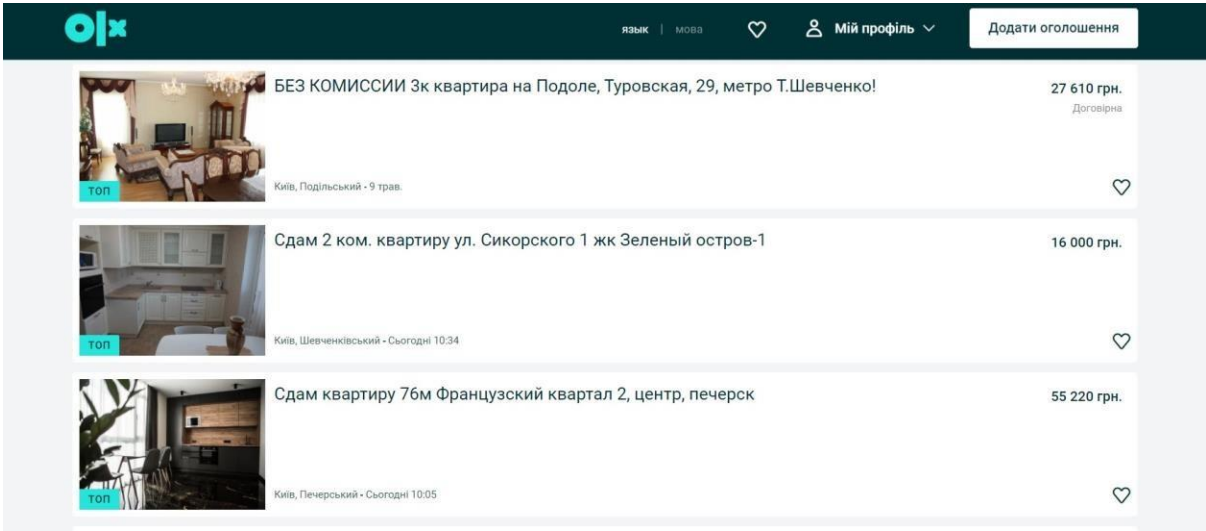


Рисунок 1.4 - Пошуку квартир на сайті OLX

Booking.com - ймовірно, одна з найпопулярніших і найбільших систем бронювання в світі, оскільки вона використовується практично в кожній країні. На сайті відмінна і зручна навігація, безліч категорій і пропозицій (рис. 1.5).

Рисунок 1.5 - Головна форма Booking.com

Однак наявна система штрафів за скасування реєстрації користувачів. Також, через свою величезну популярність, сайт показує кількість місць, які залишаються при перегляді (рисунок 1.6). і (рисунок 1.7), це змушує користувача приймати необдумані і не найкращі рішення. Також в недоліки входить те, що сайт пропонує тільки короткострокову оренду (рис. 1.8).

Рисунок 1.6 - Сторінка бронювання

Представницький двомісний номер з видом на місто. Лише 3 номери залишилось на нашому сайті 1 широке двоспальне ліжко Двомісний номер з кондиціонером, зоною відпочинку та міні-баром. 22 кв. м Вид на місто Кондиціонер Власна ванна кімната Телевізор з плоским екраном Звукоізоляція Кавоварка Міні-бар Безкоштовний Wi-Fi Ще		базовий тариф	Сніданок коштує UAH 602 – оцінка Добре ✓ БЕЗКОШТОВНЕ скасування до 23:59, 15 червня 2021 ✓ БЕЗ ПЕРЕДОПЛАТИ – сплачуєте в помешканні	0	Жодних комісій за бронювання або використання кредитки!
		UAH 10 640 UAH 11 137 Включає податки та збори Обмежена пропозиція			
		базовий тариф			
		UAH 21 426 UAH 12 822 Включає податки та збори Обмежена пропозиція	Сніданок включено – оцінка добре ✓ БЕЗКОШТОВНЕ скасування до 23:59, 15 червня 2021 ✓ БЕЗ ПЕРЕДОПЛАТИ – сплачуєте в помешканні	0	

Рисунок 1.7 – Обмеження місць на сайті Booking.com

Знаходьте пропозиції готелів, приватних помешкань та багато іншого...

Від затишних заміських будиночків до цікавих апартаментів у місті

Прага × чт, 17 чер. – сб, 31 лип. 2 дорослих · 0 дітей · 1 номер

На жаль, бронювання можливе на максимум 30 ночей

Шукати

☐ Я подорожую у справах

Рисунок 1.8 - Обмеження строку бронювання на сайті Booking.com

Flatfy - це послуга, призначена лише для роботи з нерухомістю, тобто для оренди/продажу. Сервіс пропонується клієнтам багатьох країн. Має хорошу репутацію і гарантує якість реклами. Особливістю є надання геолокації квартири (рис. 1.9).

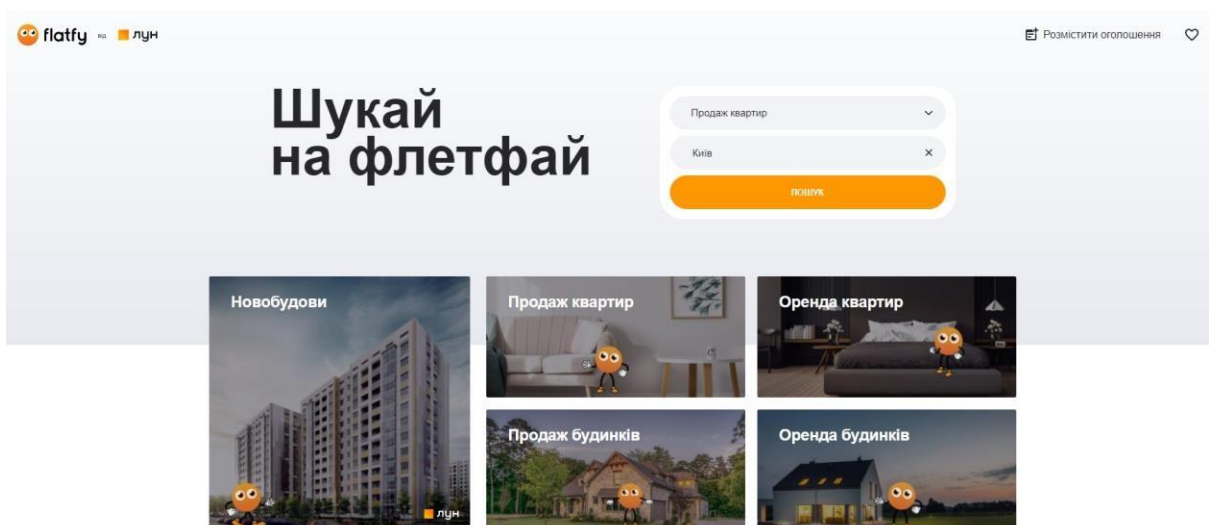


Рисунок 1.9 - Головна сторінка сайту Flatty

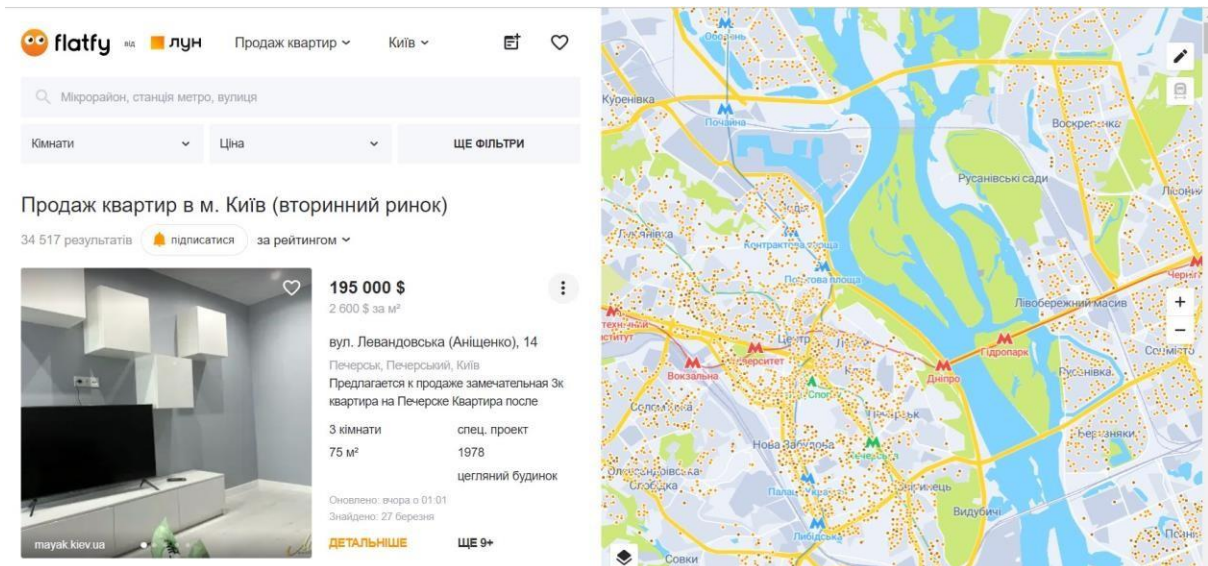


Рисунок 1.10 - Геолокація розташування квартир

Недоліком податку є те, що при виборі квартири ви перенаправляєтеся на сайт партнера, на якому розміщено це оголошення, а при пошуку квартири ви перенаправляєтеся на десятки різних сайтів.

Розглянувши систему, зроблено висновок, що вирішено створити єдину систему для організацій, що надають послуги оренди, і для користувачів, які хочуть орендувати або шукають житло. Це забезпечує взаємозв'язок між усіма етапами і не має недоліків конкурентів.

Враховуючи вище зазначене, до проекту був сформований перелік вимог.

1.2 Формування вимог до проекту

На основі вимог сформована система, що виконує такі функціональні задачі. Реєстрація користувачів, що включає клієнтів і працівників організації, надає можливість реєстрації як для організації, що надає послуги з оренди, так і для потенційних орендарів та надавачів нерухомості, забезпечуючи збір достовірних даних для подальшого використання в системі. Подача заявки на розміщення орендного оголошення включає функціонал для розміщення оголошень. Перевірка на достовірність заявок на орендні оголошення

забезпечує перевірку їх достовірності. Формування оголошення включає генерацію та публікацію оголошень після перевірки. Надання контактної інформації для зв'язку дозволяє відображати контактні дані орендодавців, забезпечуючи захист персональних даних. Формування нормативно-довідкової інформації про характеристики приміщення надає структуровану інформацію про нерухомість, включаючи площу, кількість кімнат та інші характеристики, з можливістю додавання фотографій і документів. Пошук орендних приміщень користувачем забезпечує можливість пошуку вільних приміщень для оренди за критеріями, що визначаються користувачем, такими як ціна, розташування та тип нерухомості.

Додаток включає в себе функції реєстрації користувачів, подачі заявок на оренду, перевірки заявок на достовірність та пошуку доступних приміщень. Користувачі можуть бути розділені на адміністраторів, агентів нерухомості та звичайних користувачів. Адміністратори відповідають за керування ролями та створення акаунтів агентів нерухомості. Агенти нерухомості перевіряють заявки на достовірність та підтверджують заявки на розміщення орендного оголошення, а звичайні користувачі використовують систему для розміщення своїх орендних приміщень або пошуку та перегляду оголошень про оренду [5].

Згідно з вимог сформульованих до проекту розроблена концепція інформаційної системи.

1.3 Розробка концепцій інформаційної системи

Для підвищення ефективності процесу оренди житлових приміщень розроблена концепція інформаційної системи, яка автоматизує основні задачі для трьох типів користувачів: агентів з нерухомості, орендодавців та орендарів. Основні функції системи включають реєстрацію користувачів,

подачу та перевірку заявок на оренду, формування та розміщення оголошень, а також пошук орендних приміщень.

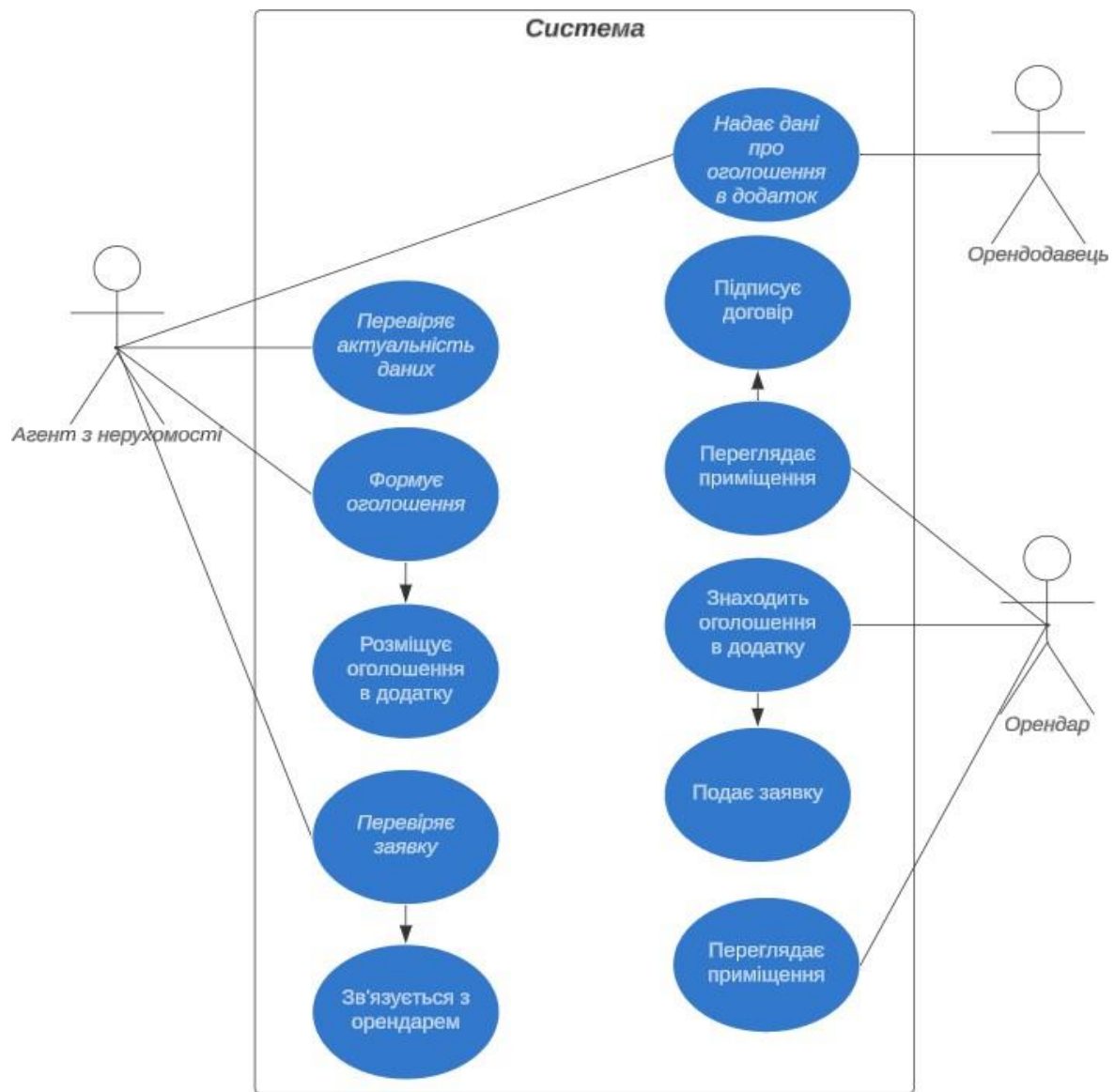


Рисунок 1.11 - Модель концепції інформаційної системи

На рисунку 1.11 наведена модель, яка ілюструє основні етапи процесу оренди житлових приміщень з використанням інформаційної системи. Ця схема демонструє взаємодію між агентом з нерухомості, орендодавцем та орендарем в межах системи.

Функціональні завдання інформаційної системи

Система виконує такі функціональні завдання. Реєстрація користувачів дозволяє реєструвати орендарів, орендодавців та працівників агентства, забезпечуючи створення профілів користувачів та підтвердження реєстрації. Подача заявки на розміщення орендного оголошення включає заповнення форми заявки, збір інформації про нерухомість та контактні дані орендодавця, а також надання статусу заявки та даних для формування оголошення. Перевірка заявок на достовірність здійснюється автоматично (перевірка на дублікати, контактна інформація) і вручну працівниками організації, результати перевірки повідомляються орендодавцям. Формування оголошення передбачає генерацію та публікацію оголошення після перевірки, з можливістю редагування та видалення оголошень орендодавцем. Надання контактної інформації забезпечує відображення контактів орендодавців для орендарів, при цьому захищаючи персональні дані та забезпечуючи конфіденційність. Формування нормативно-довідкової інформації про характеристики приміщення включає надання структурованої інформації про нерухомість (площа, кількість кімнат, розташування), можливість додавання фотографій та документів. Пошук орендних приміщень користувачем включає використання фільтрів пошуку (ціна, розташування, тип нерухомості) та відображення результатів пошуку із можливістю перегляду детальної інформації про об'єкти.

Вхідні та вихідні дані

При реєстрації користувачів вхідними даними є ім'я, прізвище, електронна пошта, номер телефону, пароль, тип користувача (орендодавець, орендар, працівник), а вихідними - профіль користувача та підтвердження реєстрації. При подачі заявки на розміщення орендного оголошення вхідними даними є дані про нерухомість (адреса, тип, площа, ціна, опис) та контактні дані орендодавця, а вихідними - повідомлення про статус заявки та дані для

формування оголошення. Для перевірки достовірності заявок вхідними даними є дані заявки на розміщення оголошення, а вихідними - повідомлення орендодавцю про результати перевірки. Формування оголошення використовує підтверджені дані заявки як вхідні дані, а вихідними є опубліковане оголошення, можливість перегляду, редагування та видалення оголошення. Надання контактної інформації використовує контактні дані орендодавця як вхідні дані, а вихідними є контактна інформація, доступна орендарям. Формування нормативно-довідкової інформації використовує характеристики нерухомості (площа, кількість кімнат, додаткові опції) як вхідні дані, а вихідними є опис об'єкту у зручному для користувача форматі. Пошук орендних приміщень використовує критерії пошуку (ціна, розташування, тип нерухомості) як вхідні дані, а вихідними є список результатів пошуку та детальна інформація про обрані об'єкти.

Під час розробки даної інформаційної системи прийняте рішення щодо використання конкретних технологій та пакетів прикладних програм з метою забезпечення ефективної розробки та функціональності системи. Прийняте рішення використовувати наступні технології:

Технологія ASP.NET Core обраний як основна для розробки веб-сервера через свою широку підтримку, високу продуктивність та масштабованість. Ця технологія дозволяє ефективно взаємодіяти з іншими компонентами системи та забезпечує високу безпеку [6].

Angular обраний для створення клієнтської частини системи з метою забезпечення багаторівневої архітектури та можливості створення динамічних та інтерактивних користувацьких інтерфейсів [7].

PostgreSQL обраний як система керування базами даних через свою надійність, розширюваність та дотримання стандартів. Вона забезпечує

ефективне зберігання та управління даними, що відповідає потребам проекту [8].

Git обраний як система контролю версій для забезпечення ефективного керування версіями програмного коду та спільної роботи над проектом. Він є стандартом у сучасній розробці програмного забезпечення та має широку підтримку та активну спільноту користувачів [9].

Обрані технології уважно обгрунтовані з урахуванням потреб проекту, їхньої функціональності та ефективності, що дозволить забезпечити успішну реалізацію та подальший розвиток інформаційної системи.

Для експлуатації системи обрані такі засоби. Процесор - багатоядерний (не менше 4 ядер), наприклад, Intel Xeon, що забезпечує високу обчислювальну потужність. Оперативна пам'ять становить не менше 16 ГБ RAM, бажано 32 ГБ або більше для забезпечення високої продуктивності. Жорсткий диск - SSD накопичувач об'ємом не менше 2 ТБ для швидкого доступу до даних. Мережева карта підтримує швидкість не менше 1 Гбіт/с для стабільного і швидкого інтернет-з'єднання. Операційна система - Linux, яка забезпечує стабільність та безпеку роботи системи.

Для функціонування додатку обране наступне програмне забезпечення:

IIS (Internet Information Services) для хостингу .NET-додатків. Має підтримку SSL/TLS для забезпечення захищеного з'єднання [10].

.NET 5+ для кросплатформених додатків та ASP.NET Core для створення веб-додатків і API [11].

Дотримання цих вимог до програмного забезпечення забезпечить надійне функціонування інформаційної системи, високу продуктивність, безпеку даних та зручність використання для користувачів. Це також сприятиме ефективному розвитку та підтримці системи в майбутньому.

Розробка інформаційної системи для оренди житлових приміщень на базі платформи .NET спрямована на підвищення ефективності роботи фірми. Застосування сучасних технологій та інструментів дозволяє автоматизувати ключові бізнес-процеси, забезпечуючи надійність, масштабованість та високу продуктивність системи.

Основні функціональні задачі, які реалізуються в системі, включають реєстрацію користувачів, подачу та перевірку орендних оголошень, формування та публікацію оголошень, а також пошук орендних приміщень користувачами. Ці функції забезпечують зручність та швидкість виконання завдань як для працівників організації, так і для клієнтів – орендодавців та орендарів.

Впровадження цієї системи дозволить знизити трудові витрати на виконання рутинних операцій, мінімізувати ризики помилок при ручному введенні даних, підвищити якість обслуговування клієнтів та забезпечити прозорість і контрольованість процесів. Використання інструментів безпеки, таких як SSL/TLS шифрування, а також систем аутентифікації та авторизації, забезпечить захист персональних даних та підвищить довіру користувачів до сервісу.

Розробка та впровадження інформаційної системи оренди житлових приміщень створює умови для значного підвищення ефективності роботи фірми, оптимізації бізнес-процесів та покращення якості наданих послуг.

РОЗДІЛ 2 Розробка технічного проекту інформаційної системи

2.1 Рішення по інформаційному забезпеченню

Система управління орендою нерухомості призначена для автоматизації процесів обробки заявок на оренду, управління інформацією про користувачів та об'єкти нерухомості, а також забезпечення збереження та верифікації документів. Вона складається з ключових компонентів:

- інтерфейс користувача, який дозволяє реєструватися, подавати заявки, завантажувати документи та переглядати статус заявок;
- адміністративний інтерфейс для управління користувачами, об'єктами нерухомості та обробки заявок;
- база даних, що зберігає інформацію про користувачів, об'єкти нерухомості, заявки та документи;
- модуль верифікації документів, який перевіряє справжність і відповідність наданих документів.

Перевірка достовірності даних при розміщенні заявки є критично важливою для забезпечення безпеки та надійності системи оренди житлових приміщень. Під час подання заявки орендодавцем необхідно підтвердити особу орендатора та наявність прав власності на майно. Це дозволяє уникнути шахрайства, захищає інтереси всіх сторін та забезпечує правову чистоту операцій.

Процес перевірки включає подання документів, що підтверджують особу орендатора (наприклад, паспорт або ID-картка), та документів, що підтверджують право власності на майно (наприклад, свідоцтво про право власності). Ці документи завантажуються у систему та проходять перевірку на відповідність і автентичність. У разі виявлення проблем або невідповідностей, заявка не приймається до розгляду, що забезпечує додатковий рівень захисту від неправомірних дій.

Така практика підвищує довіру до системи з боку користувачів, забезпечує прозорість процесу оренди та мінімізує ризики, пов'язані з орендними відносинами.

Для забезпечення роботи інформаційної системи обрана СУБД PostgreSQL.

На основі сформованих функціональних компонентів системи розроблено такі моделі предметної області що відображають ключові компоненти системи та їх взаємозв'язки.

Модель користувачів:

- id: Унікальний ідентифікатор;
- ім'я, прізвище, електронна пошта, телефон: Контактна інформація;
- тип_користувача: Орендар або власник;
- номер_документу, тип_документу: Інформація про документ, що підтверджує особу;

Модель об'єктів нерухомості:

- id: Унікальний ідентифікатор;
- адреса, тип, площа, кімнати, вартість, стан: Характеристики об'єкта;
- номер_документу, тип_документу: Інформація про право володіння;

Модель заявок:

id: Унікальний ідентифікатор;

id_користувача: Посилання на користувача;

id_об'єкта: Посилання на об'єкт;

статус: Статус заявки;

дата_подання: Дата подання заявки;

файл_документу: Шлях до файлу, що підтверджує право володіння;

На основі моделей предметної області створені таблиці інформаційної системи.

Таблиця "користувачі"

Поле	Тип даних	Опис
id	INT	Первинний ключ, унікальний ідентифікатор
ім'я	VARCHAR(50)	Ім'я користувача
прізвище	VARCHAR(50)	Прізвище користувача
електронна пошта	VARCHAR(100)	Електронна пошта користувача
телефон	VARCHAR(15)	Номер телефону користувача
тип_користувача	INT	Тип користувача (орендар, власник тощо)
дата_реєстрації	DATE	Дата реєстрації користувача
номер_документу	VARCHAR(20)	Номер документа, що підтверджує особу
тип_документу	VARCHAR(20)	Тип документа, що підтверджує особу

Таблиця "об'єкти"

Поле	Тип даних	Опис
id	INT	Первинний ключ, унікальний ідентифікатор
адреса	VARCHAR(255)	Адреса об'єкта
тип	INT	Тип об'єкта (квартира, будинок тощо)
площа	FLOAT	Площа об'єкта
кімнати	INT	Кількість кімнат
вартість	DECIMAL(10,2)	Вартість об'єкта
стан	INT	Стан об'єкта (новий, вторинний тощо)
опис	TEXT	Опис об'єкта
номер_документу_володіння	VARCHAR(20)	Номер документа, що підтверджує право власності
тип_документу_володіння	VARCHAR(20)	Тип документа, що підтверджує право власності

Таблиця "заявки"

Поле	Тип даних	Опис
id	INT	Первинний ключ, унікальний ідентифікатор
id_користувача	INT	Зовнішній ключ, посилання на користувача
id_об'єкта	INT	Зовнішній ключ, посилання на об'єкт
статус	INT	Статус заявки
дата_подання	DATE	Дата подання заявки
файл_документу	VARCHAR(255)	Шлях до файлу, що підтверджує право володіння майном

На основі наведених таблиць сконструйовані концептуальна, логічна та фізична моделі даних.

2.1.1 Розробка концептуальної моделі даних

Концептуальна модель бази даних – це абстрактне уявлення про структуру даних, яке відображає основні сутності та їх взаємозв'язки без конкретизації фізичної реалізації. Вона визначає логічні аспекти даних, включаючи сутності, атрибути та зв'язки між ними. Така модель створюється на ранніх етапах проектування системи і служить основою для подальшого створення фізичної моделі бази даних [12].

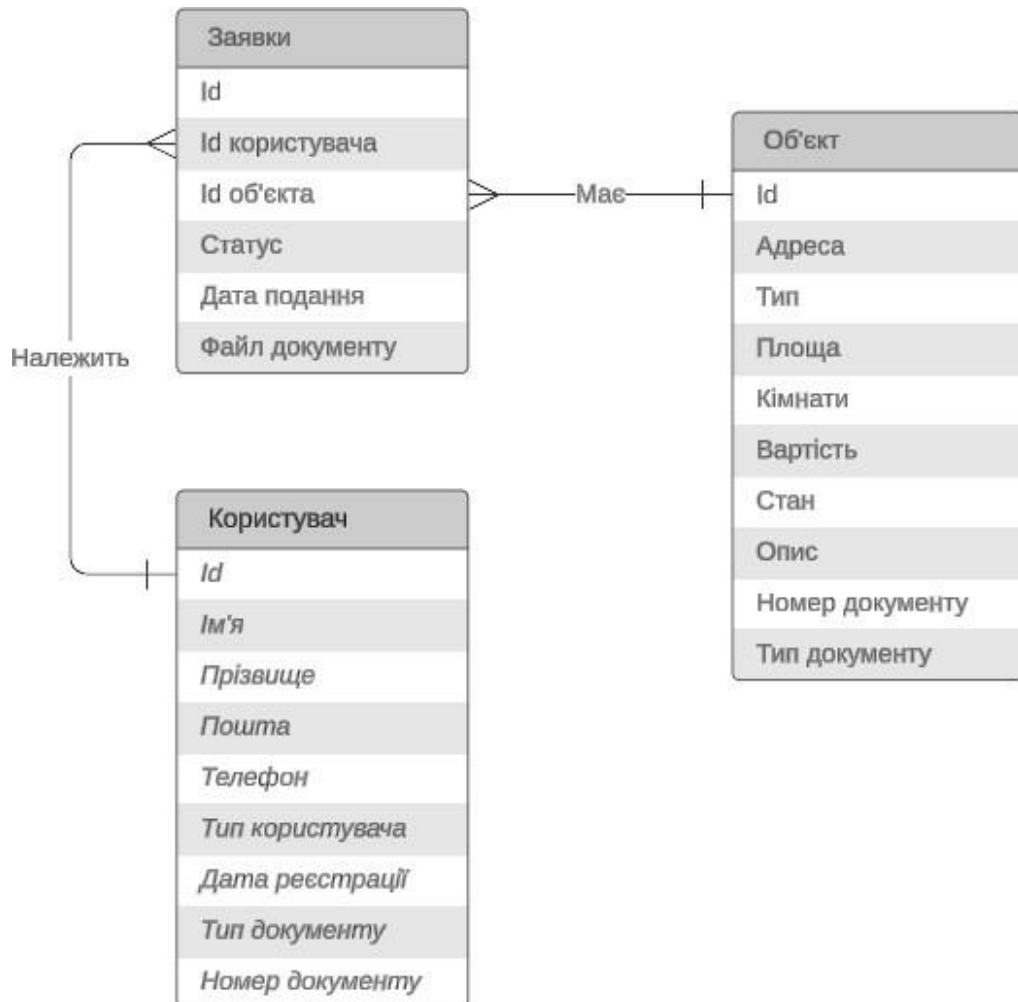


Рисунок 2.1 – Концептуальна модель бази даних

Концептуальна модель бази даних що наведена на рисунку 2.1 необхідна для чіткого визначення структури даних, що використовуються в системі управління орендою нерухомості. Модель включає сутності, такі як "Користувач", "Об'єкт" та "Заявки", а також їх атрибути та взаємозв'язки. Вона допомагає забезпечити правильну організацію даних, що дозволяє ефективно зберігати та обробляти інформацію про користувачів, об'єкти нерухомості та заявки на оренду. Концептуальна модель слугує базою для створення фізичної моделі бази даних, яка буде реалізована у відповідній СУБД, і забезпечує послідовність та узгодженість даних у системі. Завдяки цій моделі можна гарантувати, що всі важливі аспекти системи враховані, а структура бази даних відповідає потребам користувачів та бізнес-процесам.

2.1.2 Розробка логічної моделі даних

Логічна модель даних — це більш детальне уявлення структури бази даних, яке конкретизує концептуальну модель шляхом визначення типів даних, обмежень та зв'язків між таблицями. Вона фокусується на логічних аспектах зберігання та операцій над даними, але все ще незалежна від конкретних фізичних засобів їх реалізації. Логічна модель включає детальний опис таблиць, їх атрибутів, первинних та зовнішніх ключів, а також нормалізацію даних для усунення дублювання та забезпечення цілісності інформації. Вона допомагає визначити, як дані будуть організовані і якими будуть взаємозв'язки між різними сутностями в системі [13].

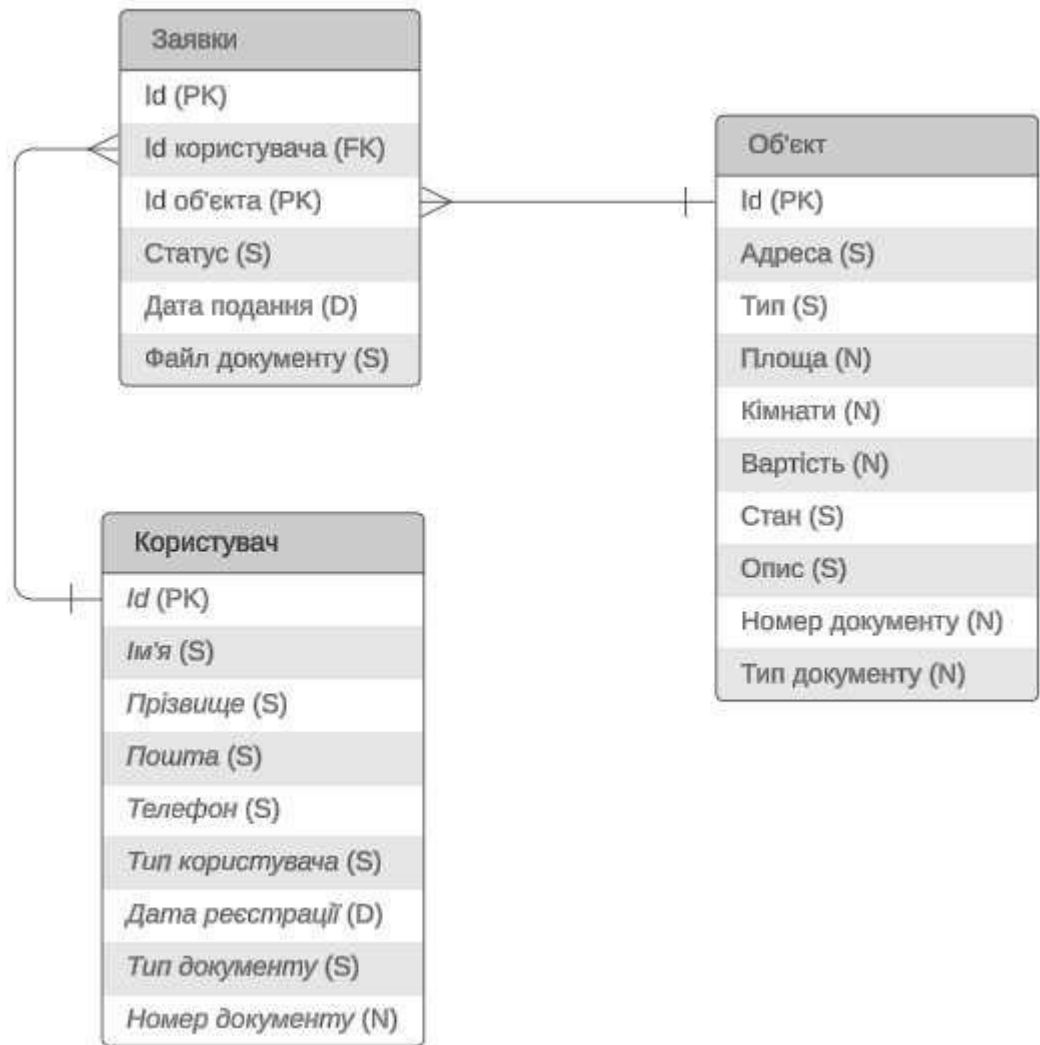


Рисунок 2.2 – Логічна модель бази даних

На рисунку 2.2 зображена логічна модель даних інформаційної системи, вона є критично важливою для точного визначення того, як будуть зберігатися та взаємодіяти дані в системі управління орендою нерухомості та перетворює концептуальну модель на більш деталізовану схему, яка включає типи даних для кожного атрибута, визначає зв'язки між таблицями, та забезпечує підтримку цілісності даних через механізми первинних і зовнішніх ключів. Логічна модель дозволяє чітко зрозуміти, як дані будуть організовані в базі даних, забезпечуючи ефективну роботу системи та оптимізацію запитів. Вона також допомагає ідентифікувати потенційні проблеми з даними на ранніх

етапах розробки, що дозволяє заздалегідь вирішити їх, забезпечуючи стабільність та надійність системи при впровадженні в експлуатацію.

2.1.3 Розробка фізичної моделі даних

Фізична модель бази даних - це модель даних, яка визначає, яким чином надаються дані, і містить всі деталі, необхідні СУБД для створення бази даних. Фізична модель бази даних інформаційної системи зазначена на рисунку 2.3 [14].

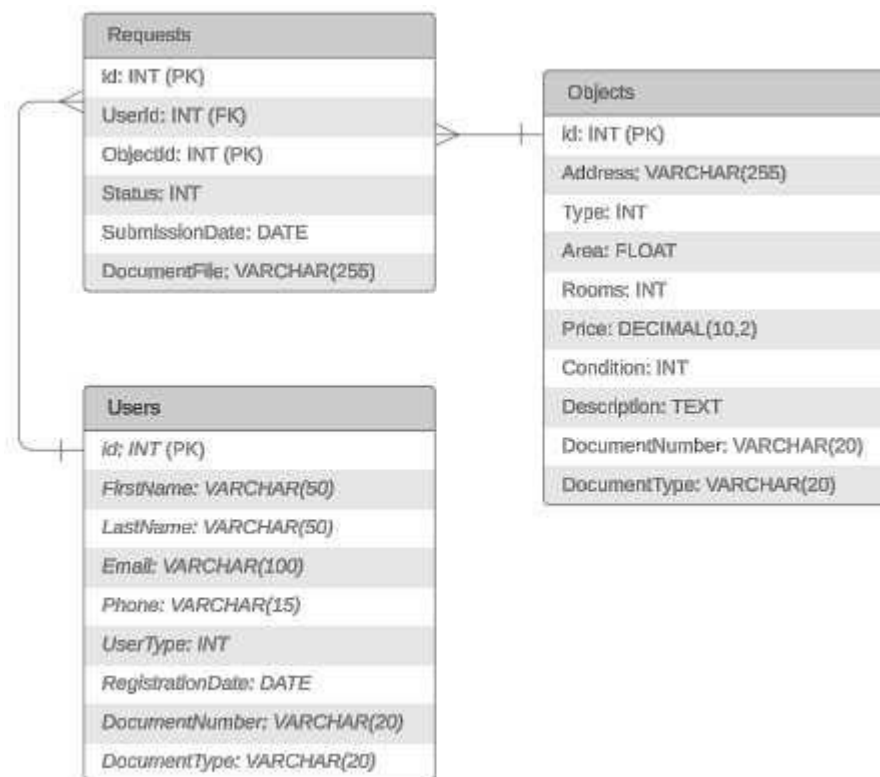


Рисунок 2.3 – Фізична модель бази даних

Для заповнення всіх модулів бази даних використовуються форми по створенню об'єктів, заявок на розміщення та форма реєстрації в системі.

На основі сформованих моделей розробляється програмне забезпечення для бази даних інформаційної системи.

2.2 Рішення по математичному забезпеченню

Рішення по математичному забезпеченню прийнято на основі функціональних алгоритмів інформаційної системи. Функціонал системи описано з урахуванням трьох основних типів користувачів: орендодавці, орендарі та адміністратори. Кожен тип користувача має свій набір функцій та обов'язків у системі [15].

Функціонал алгоритму орендодавця зазначений на рисунку 2.4. Орендодавці володіють нерухомістю та здають її в оренду. Вони реєструються в системі, заповнюючи свої особисті дані та завантажуючи документи, що підтверджують особу та право власності на нерухомість. Після реєстрації орендодавці можуть додавати об'єкти нерухомості до системи, вводячи деталі про об'єкт, такі як адреса, тип, площа, кількість кімнат, вартість та стан, а також завантажуючи документи, що підтверджують право власності. Орендодавці можуть переглядати список своїх об'єктів нерухомості та редагувати інформацію про них при необхідності. Вони також отримують повідомлення про нові заявки на оренду, переглядають ці заявки та вирішують, чи приймати їх.

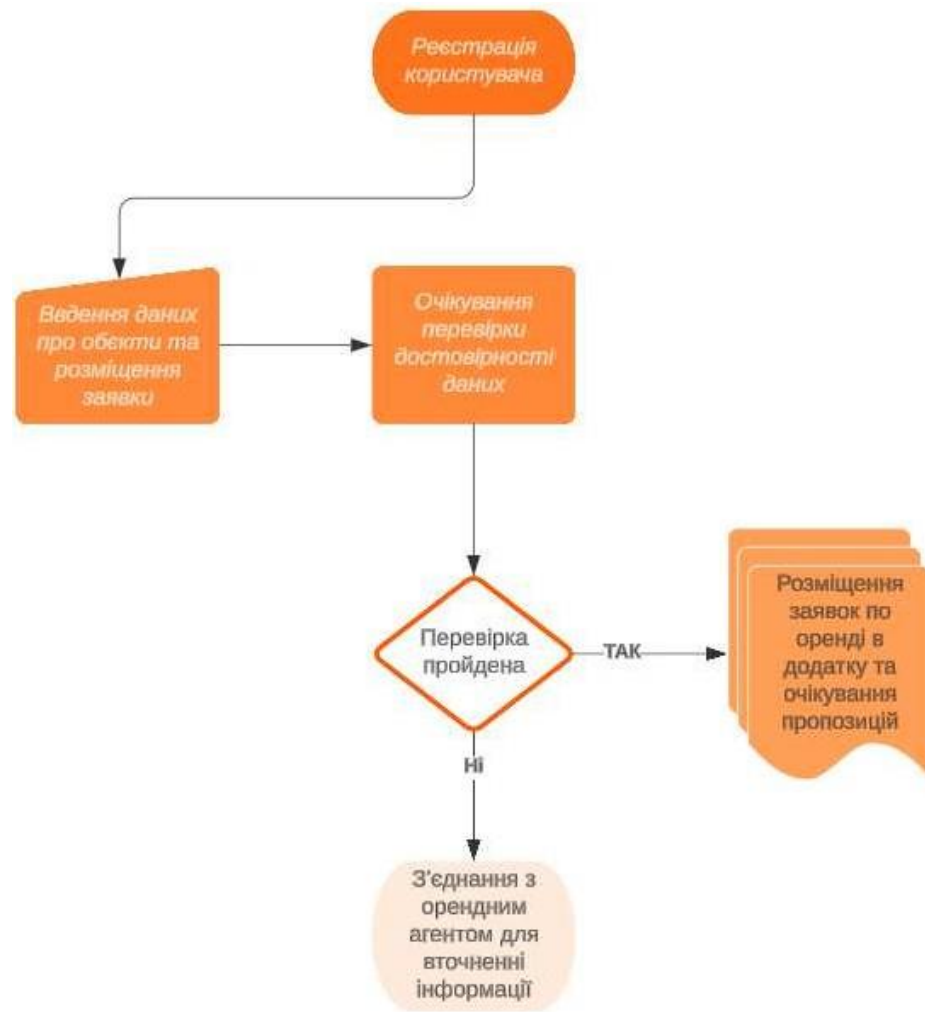


Рисунок 2.4 – Функціонал алгоритму орендодавця

Орендарі – шукають нерухомість для оренди. Вони реєструються в системі, заповнюючи свої особисті дані та завантажуючи документи, що підтверджують особу. Після реєстрації орендарі можуть переглядати доступні об'єкти нерухомості, використовуючи фільтри для пошуку об'єктів за адресою, типом, площею, вартістю та іншими параметрами. Орендарі вибирають об'єкт та подають заявку на оренду, вводячи необхідну інформацію та завантажуючи документи, що підтверджують особу. Алгоритм зображений на рисунку 2.5.

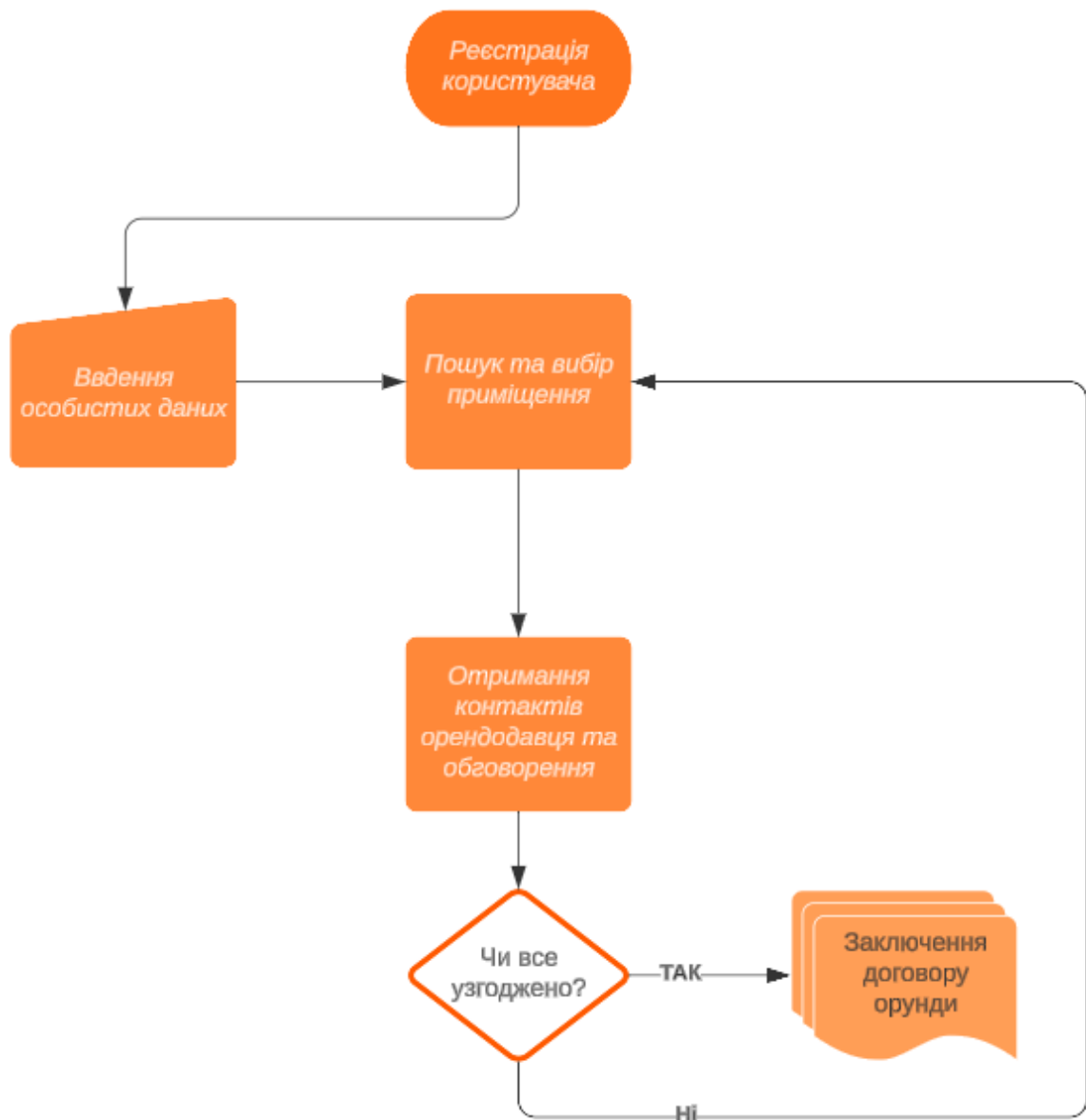


Рисунок 2.5 - Функціонал алгоритму орендаря

Агенти з нерухомості – забезпечують функціонування системи, обробляють заявки та перевіряють документи. Вони перевіряють документи орендодавців та орендарів, що підтверджують особу та право власності, підтверджуючи або відхиляючи реєстрацію користувачів на основі перевірки документів. Агенти також перевіряють інформацію про об'єкти нерухомості, завантажену орендодавцями, підтверджуючи або відхиляючи додавання нових об'єктів до системи. Вони обробляють заявки на оренду, подані орендарями,

перевіряючи документи та інформацію, надану в заявках. Алгоритм представлений на рисунку 2.6.

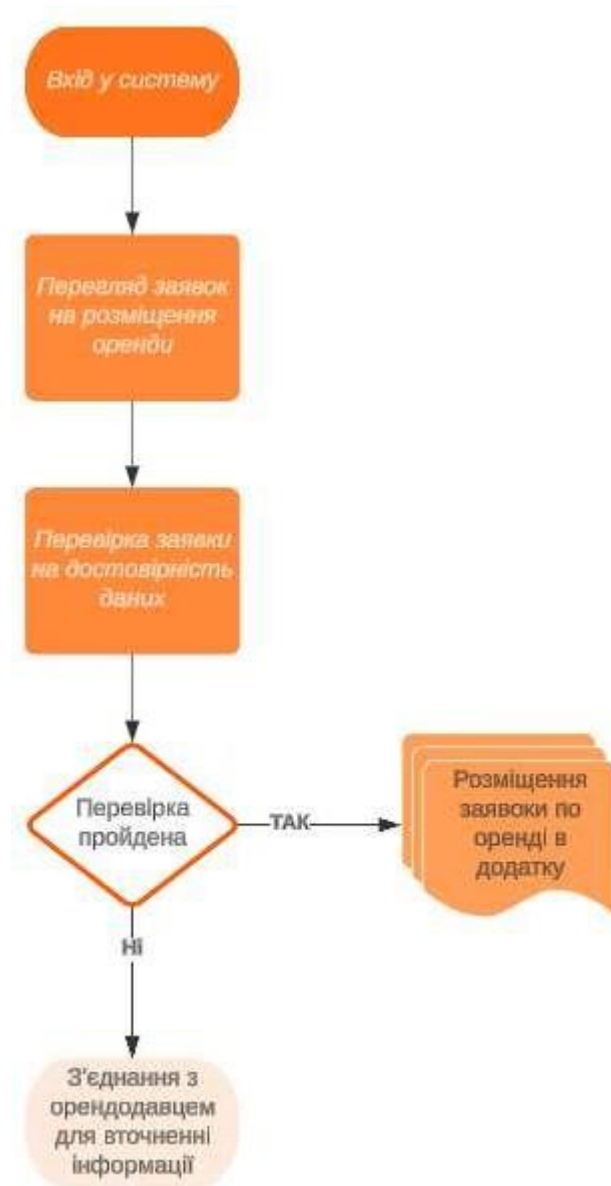


Рисунок 2.6 – Функціонал алгоритму агента з нерухомості

Створене рішення по математичному забезпеченню включає алгоритми для обробки заявок, перевірки документів та пошуку орендних приміщень, а також методи моделювання для прогнозування попиту та оптимізації ресурсів, що забезпечує ефективне функціонування системи автоматизованого управління орендою нерухомості.

Алгоритми для обробки заявок дозволяють автоматизувати процеси реєстрації орендодавців та орендарів, що значно зменшує час на обробку документів та підвищує точність даних. Зокрема, орендодавці можуть швидко додавати свої об'єкти нерухомості до системи, а орендарі — знаходити відповідні варіанти для оренди, використовуючи різні фільтри пошуку. Це забезпечує зручність та ефективність для обох сторін процесу оренди. Алгоритми перевірки документів, в свою чергу, гарантують надійність та безпеку угод, оскільки всі документи ретельно перевіряються на автентичність та відповідність вимогам. На основі них розроблено рішення по програмному забезпеченню інформаційної системи.

2.3 Рішення по програмному забезпеченню

Для забезпечення функціонування інформаційної системи з оренди житлових приміщень розроблено програмне забезпечення. Воно включає серверне та клієнтське програмне забезпечення, бази даних, інструменти для розробки.

2.3.1 База даних

В якості СУБД обрано PostgreSQL, це потужна реляційна система управління базами даних з відкритим вихідним кодом. Вона ідеально підходить для зберігання та обробки великих обсягів даних, а також складних запитів. Враховуючи, що система оренди житлових приміщень вимагає надійного зберігання даних про орендодавців, орендарів, об'єкти нерухомості та договори, PostgreSQL обрана через її надійність, масштабованість та розширені можливості для роботи з транзакціями [8].

Основний функціонал бази даних включає зберігання інформації про користувачів системи, управління даними про об'єкти нерухомості, обробку договорів оренди, а також ведення історії транзакцій та дій користувачів для забезпечення безпеки та аудиту. Архітектура бази даних включає детально

розроблені таблиці з атрибутами, первинними та зовнішніми ключами, а також нормалізовані дані для усунення дублювання та забезпечення цілісності інформації. Для роботи з базою даних використовуватимуться інструменти, такі як pgAdmin та DBeaver, що забезпечують зручне налаштування та відлагодження. По базі даних були розроблені модулі для програмного забезпечення серверної частини.

2.3.2 Серверна частина

Серверна частина системи розроблена на основі .NET Core, кросплатформеного фреймворку від Microsoft, що дозволяє створювати веб-додатки та API. Це забезпечує гнучкість у виборі серверного середовища та знижує витрати на інфраструктуру. Мова програмування - C#, яка забезпечує високу продуктивність та безпеку додатків [6].

Функціонал серверної частини включає обробку запитів від клієнтської частини, управління сесіями користувачів та аутентифікацію/авторизацію, інтеграцію з базою даних PostgreSQL для зберігання та отримання даних, а також впровадження бізнес-логіки для оренди житлових приміщень. Основні модулі API включають модуль управління користувачами, модуль управління об'єктами нерухомості.

Архітектура серверної частини реалізована у вигляді мікросервісів, що забезпечує легке масштабування системи та високу доступність. Мікросервіси дозволяють розподіляти навантаження та забезпечують можливість незалежного оновлення окремих компонентів системи. Кожен мікросервіс відповідає за окрему частину функціональності системи, наприклад, обробку запитів до бази даних або управління аутентифікацією. Цей підхід також полегшує тестування та розгортання нових версій додатка.

Кожен мікросервіс реалізовує один з наявних модулів.

Мікросервіс реєстрації та входу користувачів відповідає за обробку процесу реєстрації нових користувачів і авторизації існуючих. Він включає запити на створення або перевірки облікових записів.

Сервіс управління об'єктами нерухомості відповідає за обробку запитів на управління об'єктами нерухомості.

Для забезпечення безперебійної роботи системи використовуються такі технології, як Docker для контейнеризації додатків і Kubernetes для оркестрації контейнерів. Це дозволяє легко масштабувати систему відповідно до навантаження та забезпечувати високу доступність навіть під час пікових навантажень. Використання контейнеризації також забезпечує ізоляцію середовища виконання додатків, що знижує ризик конфліктів між різними компонентами системи та полегшує управління залежностями [16].

Для забезпечення безпеки та управління доступом використовується ASP.NET Identity, яка забезпечує керування користувачами, аутентифікацію та авторизацію на основі ролей. Для підвищення безпеки системи впроваджено багатофакторну аутентифікацію (MFA), що додатково захищає облікові записи користувачів від несанкціонованого доступу. Для захисту даних під час передачі використовуються SSL/TLS сертифікати, що забезпечують шифрування з'єднань між клієнтською та серверною частинами [17].

Всі модулі системи пройшли налаштування та відлагодження за допомогою Visual Studio та готові до експлуатації.

2.3.3 Клієнтська частина

Клієнтська частина системи розроблена на основі Angular, потужного фреймворку для створення динамічних веб-додатків від Google. Angular дозволяє створювати односторінкові додатки (SPA), що забезпечують швидкий і зручний інтерфейс користувача для орендодавців, орендарів та

адміністраторів. Мова програмування - TypeScript, яка забезпечує надійну типізацію та підвищує якість коду [7].

Функціонал клієнтської частини включає інтерфейси для реєстрації та входу користувачів, панель управління для орендодавців, де вони можуть додавати, редагувати та видаляти об'єкти нерухомості, інтерфейс для орендарів, що дозволяє переглядати доступні об'єкти, а також адміністративну панель для адміністраторів системи. Важливі форми, що будуть реалізовані, включають форму додавання нових об'єктів нерухомості, форму подачі заявки на оренду.

Архітектура клієнтської частини базується на компонентному підході Angular, що забезпечує високу модульність та повторне використання коду. Кожен компонент відповідає за окрему частину інтерфейсу, що полегшує розробку та тестування. Компоненти включають модулі для реєстрації та входу користувачів, панель управління для орендодавців з можливістю управління об'єктами нерухомості, інтерфейси для перегляду та подачі заявок на оренду для орендарів, а також адміністраторську панель для моніторингу та управління системою.

Компонент реєстрації та входу користувачів відповідає за обробку процесу реєстрації нових користувачів і авторизації існуючих. Він включає форми для введення особистих даних, валідацію введеної інформації та обробку запитів до сервера для створення або перевірки облікових записів.

Панель управління для орендодавців забезпечує орендодавцям інструменти для управління своїми об'єктами нерухомості. Вони можуть додавати нові оголошення, редагувати існуючі та видаляти непотрібні. Інтерфейс зручний і інтуїтивно зрозумілий, з можливістю завантаження фотографій та документів.

Інтерфейс для орендарів дозволяє орендарям переглядати доступні для оренди об'єкти нерухомості. Він включає розширені фільтри для пошуку за критеріями, такими як ціна, розташування, тип нерухомості та інші характеристики. Орендарі можуть переглядати детальні описи, фотографії та зв'язуватися з орендодавцями.

Адміністративна панель надає адміністраторам системи інструменти для моніторингу та управління користувачами та оголошеннями. Адміністратори можуть перевіряти достовірність поданих заявок, редагувати або видаляти оголошення, а також здійснювати загальне управління системою для забезпечення її безперебійної роботи та відповідності політикам конфіденційності та безпеки.

Для підвищення продуктивності та зручності користувачів використовуються сучасні фронтенд-технології, такі як RxJS для роботи з асинхронними запитами, NgRx для управління станом додатка та Angular Material для створення сучасного та інтуїтивно зрозумілого інтерфейсу користувача. Це забезпечує швидку реакцію додатка на дії користувачів та зменшує час завантаження сторінок [18] [19].

Для налаштування та відлагодження використовувалися Angular CLI та Chrome DevTools.

Для системи контролю версій обрано Git, розподілену систему управління версіями, яка забезпечує ефективне керування змінами в коді та співпрацю з іншими розробниками. Git дозволяє відстежувати всі зміни, легко повертатися до попередніх версій та ефективно співпрацювати в команді [9].

Основний функціонал системи версіонування включає ведення історії змін коду для кожного розробника та всієї команди, можливість створення гілок для розробки нових функцій без впливу на основну версію коду, інтеграцію з сервісами GitHub для централізованого зберігання репозиторіїв

та управління проектами, а також автоматизоване злиття та вирішення конфліктів коду.

Всі розроблені модулі системи відлагоджені та відповідають сформованим до проекту вимогам. Фрагменти коду програмних модулів зазначені в додатку А.

Вибір вказаних технологій та інструментів дозволив створити надійну, масштабовану та зручну у використанні систему оренди житлових приміщень, що відповідає сучасним вимогам безпеки та продуктивності. Цей підхід гарантує стабільне функціонування системи, зручність для користувачів та можливість подальшого розвитку та масштабування. На основі рішень по програмному забезпеченню сформовані рішення до системного забезпечення.

2.4 Рішення по технічному забезпеченню

Для забезпечення надійної роботи інформаційної системи з оренди житлових приміщень необхідно підібрано технічне забезпечення. Воно включає серверне обладнання, мережеве обладнання та клієнтські пристрої, які разом забезпечують стабільність, продуктивність і безпеку системи.

Серверне обладнання відіграє ключову роль у забезпеченні стабільної роботи системи. Обрано сервери з багатоядерними процесорами та великим обсягом оперативної пам'яті. Це гарантує швидку обробку запитів та високу продуктивність додатків. Сервери оснащені швидкими SSD-дисками, які забезпечують оперативний доступ до даних та їх швидке зчитування і запис. Для хостингу .NET-додатків використовується веб-сервер IIS, який відомий своєю надійністю та масштабованістю. Використання SSL/TLS забезпечує захищене з'єднання, що є критично важливим для безпеки даних користувачів.

Для експлуатації системи обрані наступні технології: багатоядерний процесор (не менше 4 ядер), наприклад, Intel Xeon або AMD EPYC, оперативна пам'ять об'ємом не менше 16 ГБ RAM, бажано 32 ГБ або більше для високої

продуктивності. Жорсткий диск - SSD накопичувач об'ємом не менше 2 ТБ для швидкого доступу до даних. Мережева карта повинна підтримувати швидкість не менше 1 Гбіт/с. Операційна система - Linux.

Для зберігання та обробки даних обрано PostgreSQL як сервер баз даних. PostgreSQL забезпечує високу продуктивність та надійність при роботі з великими обсягами даних, що особливо важливо для системи оренди житлових приміщень. Використання резервного копіювання та відновлення даних забезпечує безперервність роботи системи навіть у разі виникнення збоїв.

Клієнтське обладнання включає сучасні комп'ютери та мобільні пристрої, які дозволяють користувачам зручно взаємодіяти з системою. Робочі станції працівників організації працюють на операційних системах Windows 10, що забезпечує сумісність з необхідним програмним забезпеченням.

Мережеве обладнання включає маршрутизатори та комутатори, які забезпечують швидке та стабільне з'єднання між клієнтами та сервером.

Ретельно підібране технічне забезпечення гарантує ефективну роботу інформаційної системи оренди житлових приміщень, забезпечуючи високу продуктивність, безпеку даних та зручність для користувачів.

РОЗДІЛ 3 Реалізація проекту інформаційної системи

Для забезпечення ефективної та правильної експлуатації системи оренди житлових приміщень, було розроблено детальні інструктивно-методичні матеріали. Ці матеріали містять покрокові інструкції, рекомендації та кращі практики використання функціоналу системи, включаючи процеси реєстрації, подачі заявок, перевірки оголошень, пошуку приміщень та управління оголошеннями. Інструктивні матеріали розроблені з урахуванням різних типів користувачів. Матеріали продемонстровані в додатку Б.

Система оренди житлових приміщень призначена для автоматизації процесів оренди, включаючи реєстрацію користувачів, подачу заявок на розміщення оголошень, перевірку та публікацію оголошень, а також пошук орендних приміщень. Вона забезпечує зручний інтерфейс для орендодавців, орендарів та адміністраторів, дозволяючи ефективно керувати процесами оренди.

Ця система може бути використана агентствами нерухомості та приватними орендодавцями для автоматизації та спрощення процесу оренди житлових приміщень. Вона забезпечує можливість створення, редагування та публікації оголошень, а також пошуку орендних приміщень потенційними орендарями.

Система дозволяє користувачам:

- реєструватися як орендарі, орендодавці або працівники агентства.
- подавати заявки на розміщення орендних оголошень.
- автоматично та вручну перевіряти заявки на достовірність.
- формувати та публікувати оголошення на платформі.
- шукати орендні приміщення за різними критеріями.

Система розрахована на користувачів з базовими навичками роботи з комп'ютером та інтернетом. Для адміністраторів потрібні додаткові знання у сфері управління нерухомістю.

Присутні наступні операції в системі як

1. Реєстрація користувачів

Система надає функціонал для реєстрації різних типів користувачів:

Орендарі: можуть створювати обліковий запис, заповнюючи форму реєстрації з особистими даними та контактною інформацією.

Орендодавці: створюють обліковий запис, заповнюючи форму з особистими даними, контактною інформацією та інформацією про власність.

Працівники агентства (адміністратори): реєструються через спеціальну форму з додатковими полями для службової інформації.

Процес реєстрації користувача в системі позначений на рисунку 3.1.

Flatty About US

EN UA BECAME OWNER BECAME RENTER SIGN IN

Registration

Enter your name Enter your surname Enter your father name

05/15/2021 ☐ Woman ☐ Man

Enter your email

Enter your phone

Enter your password

Confirmed your password

Have account ? [Login](#) REGISTER

Рисунок 3.1 – Реєстрація в системі

2. Подача заявки на розміщення орендного оголошення

Орендодавці можуть подавати заявки на розміщення оголошень через спеціальну форму, де вказується інформація про нерухомість, контактні дані та бажані умови оренди. Заявка включає:

- опис нерухомості;
- фотографії приміщення;
- контактну інформацію орендодавця;
- перевірка на достовірність заявок;

Процес подачі заявки на розміщення оголошення зображений на рисунку 3.2.

Рисунок 3.2 – Заявка на розміщення оголошення

Заявки на розміщення оголошень проходять перевірку для підтвердження достовірності даних. У разі виявлення помилок або невідповідностей, заявка може бути відхилена або відправлена на доопрацювання.

3. Формування оголошення

Після успішної перевірки заявки оголошення автоматично формується та публікується на платформі. Орендодавець має можливість редагувати або видаляти свої оголошення через особистий кабінет. Кожне оголошення містить:

- опис нерухомості;
- фотографії;

- контактну інформацію орендодавця;
- надання контактної інформації для зв'язку;

Контактна інформація орендодавців відображається для зареєстрованих орендарів. Система забезпечує захист персональних даних та конфіденційність, запобігаючи несанкціонованому доступу до особистої інформації.

4. Формування нормативно-довідкової інформації про характеристики приміщення

Система надає можливість структурованого додавання інформації про характеристики нерухомості, включаючи:

- площа;
- кількість кімнат;
- розташування;
- наявність додаткових зручностей;

Користувачі можуть додавати фотографії та супровідні документи, що підвищує привабливість оголошення.

5. Пошук орендних приміщень користувачем

Орендарі можуть використовувати розширені фільтри для пошуку житлових приміщень за різними критеріями, такими як ціна, розташування, тип нерухомості тощо. Результати пошуку відображаються у вигляді списку з можливістю перегляду детальної інформації про кожен об'єкт.

Процес пошуку приміщень зображений на рисунку 3.2.

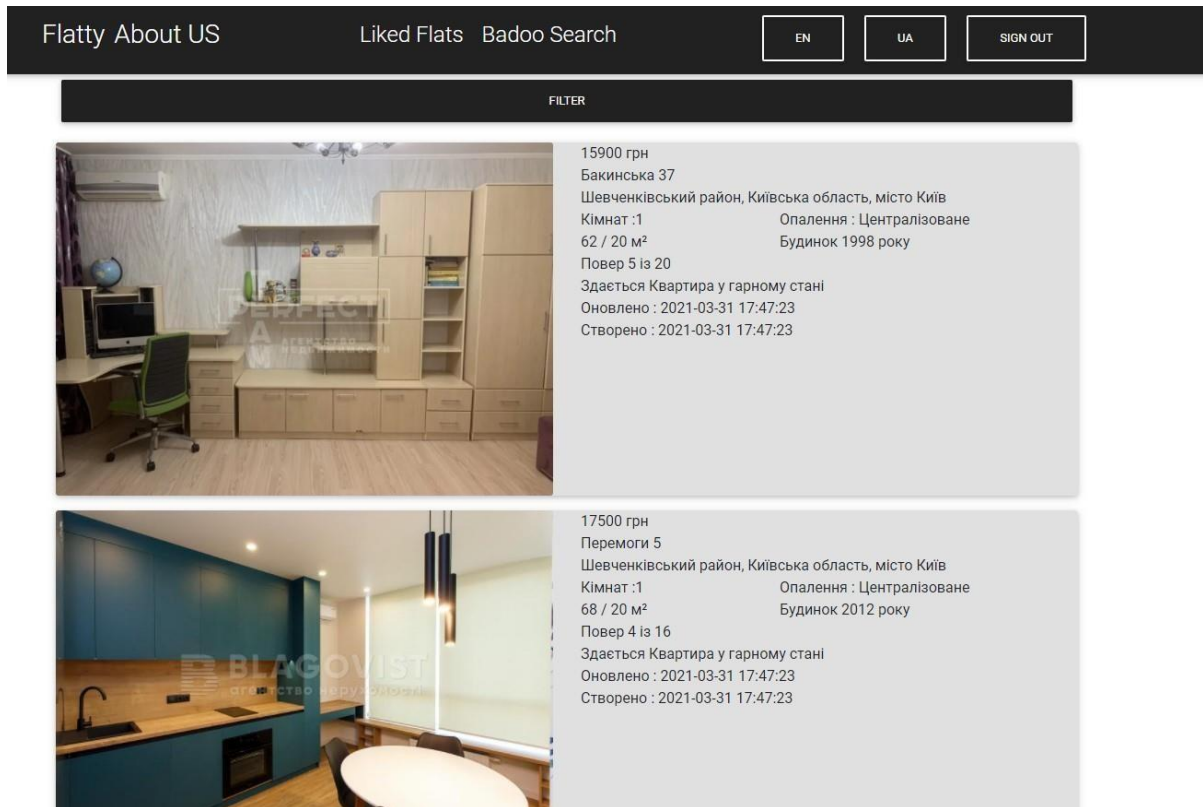


Рисунок 3.3 – Сторінка пошуку приміщень

Після розгортання та початку експлуатації системи проводився моніторинг її роботи для оцінки продуктивності та відповідності вимогам замовника. Зібрано відгуки користувачів, що дозволило виявити недоліки та провести необхідні коригування. Результати аналізу показали, що система відповідає функціональним вимогам та забезпечує зручність використання для всіх типів користувачів.

РОЗДІЛ 4 Охорона праці та техніка безпеки

Забезпечення охорони праці на підприємствах, що працюють у сфері інформаційних технологій, є важливою складовою успішного функціонування компанії та збереження здоров'я працівників. Охорона праці включає комплекс заходів, спрямованих на запобігання нещасним випадкам, професійним захворюванням і аваріям, а також створення безпечних і здорових умов праці .

Основні нормативно-правові акти, що регулюють питання охорони праці в Україні, включають Кодекс законів про працю України, Закон України "Про охорону праці" та інші нормативно-правові акти, що регулюють трудові відносини та охорону праці .

4.1 Вимоги до умов праці в офісних приміщеннях

Для забезпечення безпечних умов праці в офісних приміщеннях, де переважно працюють фахівці з комп'ютерних наук, необхідно дотримуватися певних вимог. Робочі місця повинні бути обладнані зручними меблями, що відповідають ергономічним вимогам. Це включає робочі столи і стільці, які мають бути регульованими по висоті та нахилу.

Освітлення в офісних приміщеннях повинно бути достатнім для виконання візуальних робіт без напруження зору. Природне і штучне освітлення повинні забезпечувати рівень освітленості, достатній для комфортної роботи. Екран монітора не повинен відблискувати, що забезпечується правильним розташуванням робочого місця відносно джерел світла.

Температура повітря в приміщенні має бути в межах 18-24°C, а відносна вологість повітря – 40-60%. Регулярне провітрювання приміщень забезпечує підтримку оптимального мікроклімату. Рівень шуму в офісі не повинен

перевищувати допустимих норм, встановлених ДС. Для зниження рівня шуму рекомендується використовувати шумопоглинаючі матеріали [20].

4.2 Пожежна безпека

Пожежна безпека в офісних приміщеннях забезпечується шляхом дотримання певних вимог. Евакуаційні виходи повинні бути вільними та доступними, їх розташування має бути чітко позначене. Ширина евакуаційних шляхів не повинна бути меншою за 1 метр. Приміщення повинні бути обладнані первинними засобами пожежогасіння, такими як вогнегасники. Вогнегасники повинні розташовуватися у легкодоступних місцях і регулярно перевірятися на працездатність [21].

В офісних приміщеннях повинні бути встановлені системи пожежної сигналізації та оповіщення про пожежу. Ці системи мають регулярно перевірятися на працездатність. На кожному поверсі приміщення повинен бути розміщений план евакуації на випадок пожежі. Персонал повинен бути ознайомлений з планом та пройти відповідне навчання.

4.3 Профілактика професійних захворювань

Для запобігання професійним захворюванням серед працівників у сфері комп'ютерних наук необхідно проводити ряд заходів. Працівники повинні проходити періодичні медичні огляди для виявлення ранніх ознак професійних захворювань та запобігання їх розвитку. Необхідно дотримуватися режиму праці та відпочинку, включаючи регулярні перерви під час роботи за комп'ютером. Рекомендується проводити перерви через кожну годину роботи, виконуючи прості фізичні вправи для зняття напруження.

Робочі місця повинні бути організовані відповідно до ергономічних вимог. Особлива увага повинна приділятися правильній посаді, положенню рук і ніг, а також налаштуванню монітора на комфортну висоту і відстань. Створення сприятливого психологічного клімату в колективі, проведення

тренінгів з управління стресом і профілактики професійного вигорання є важливими складовими профілактики професійних захворювань [22].

4.4 Робота за комп'ютером

Робота за комп'ютером має свої специфічні вимоги щодо охорони праці. Важливо забезпечити правильну організацію робочого місця для зниження навантаження на зір і опорно-рухову систему. Монітор повинен розташовуватися на відстані 50-70 см від очей користувача, а верхній край екрану має бути на рівні або трохи нижче рівня очей [23].

Клавіатура і мишка повинні знаходитися на зручній висоті для рук, щоб уникнути зайвого напруження м'язів. Регулярні перерви в роботі за комп'ютером дозволяють зменшити втому і напруження очей. Під час перерв рекомендується виконувати прості вправи для очей, шиї та рук. Використання спеціальних програм, що нагадують про необхідність зробити перерву, може бути корисним для підтримки здоров'я.

Забезпечення охорони праці на підприємстві є комплексним процесом, що включає виконання багатьох заходів і дотримання встановлених норм та правил. Впровадження сучасних систем і технологій, а також постійний контроль за дотриманням вимог охорони праці, дозволяє значно знизити ризики виробничих травм і захворювань, підвищити ефективність роботи та створити комфортні умови для працівників.

ВИСНОВКИ

У кваліфікаційній роботі виконано розробку інформаційної системи для оренди житлових приміщень.

Зроблений опис об'єкту дослідження і аналіз предметної області, сформовані вимоги, на основі яких розроблена концепція інформаційної системи.

Розроблений технічний проект системи. Представлені загальносистемні рішення для вибору типових програмних засобів. Спроектовано базу даних.

Описані алгоритми роботи проекту та визначені програмні і технічні засоби реалізації і експлуатації проекту.

Розроблені інструктивно-методичні матеріали та проведені тестування, що перевірили коректність роботи усіх програмних модулів інформаційної системи.

Представлені правила охорони праці під час роботи за комп'ютером.

Актуальність та новизна проекту полягає в створенні сучасної та якісної системи, що забезпечує покращення ефективності процесу оренди за рахунок використання сучасних технологій, що має попит на ринку систем для задач здійснення оренди.

В результаті розроблена та готова до експлуатації система що відповідає опису та всім вимогам для підвищенню ефективності процесів оренди.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кравченко, І. В. (2020). Управління нерухомістю: теорія та практика. Київ: Видавництво КНЕУ.
2. Петров, О. М. (2019). Системи управління орендою нерухомості. Харків: Видавництво ХНУРЕ.
3. Іваненко, Л. О. (2021). Цифрові платформи на ринку нерухомості. Львів: ЛНУ імені Івана Франка. Петров, О. М. (2019). Системи управління орендою нерухомості. Харків: Видавництво ХНУРЕ.
4. Гринько, Т. М. (2017). Системи верифікації для управління нерухомістю. Дніпро: ДНУ імені Олеся Гончара.
5. Сидоренко, Н. В. (2018). Розробка додатків для оренди нерухомості. Одеса: Видавництво ОНУ.
6. Microsoft. (2024). ASP.NET Core Overview. URL: <https://learn.microsoft.com/en-us/aspnet/core/>
7. Angular Team. (2024). Angular Documentation. URL: <https://angular.io/docs>
8. PostgreSQL Global Development Group. (2024). PostgreSQL Overview. URL: <https://www.postgresql.org/>
9. Git Documentation Team. (2024). Git Documentation. URL: <https://git-scm.com/doc>
10. Microsoft. (2024). IIS (Internet Information Services) Overview. URL: <https://learn.microsoft.com/en-us/iis/>
11. Microsoft. (2024). .NET Overview. URL: <https://learn.microsoft.com/en-us/dotnet/core/>
12. Кудрявцев, В. М. "Бази даних: проектування, реалізація та обслуговування".
13. Ляшенко, В. І. "Системи управління базами даних".
14. Дорошенко, О. В. "Проектування баз даних".

15. Шаров Ю. П., Бакуменко О. О., Ткаченко І. І. "Математичне моделювання інформаційних систем". Київ: Видавництво НАУ, 2011. – 244 с.
16. Офіційна документація Docker. URL: <https://docs.docker.com/>
17. Офіційна документація Microsoft. URL: <https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-5.0>
18. ReactiveX. (2024). RxJS Documentation. URL: <https://rxjs.dev/guide/overview>
19. Angular Team. (2024). Angular Material Documentation. URL: <https://material.angular.io/>
20. ДСН 3.3.6.037-99. Державні санітарні норми виробничого шуму, ультразвуку та інфразвуку.
21. Правила пожежної безпеки в Україні, затверджені Наказом МНС України від 30.12.2014 № 1417.
22. Методичні рекомендації щодо організації праці та відпочинку при роботі з персональними комп'ютерами
23. ДСанПіН 3.3.2.007-98 "Державні санітарні правила і норми роботи з комп'ютерами"

ДОДАТОК А

```

public class Apartment
{
    public int Id { get; set; }
    public string Title { get; set; }
    public string Description { get; set; }
    public string Address { get; set; }
    public decimal Price { get; set; }
    public List<string> Photos { get; set; }
    public string ContactInfo { get; set; }
}

public class ApplicationDbContext : DbContext
{
    public DbSet<Apartment> Apartments { get; set; }

    public ApplicationDbContext(DbContextOptions<ApplicationDbContext> options) :
base(options) { }
}

[ApiController]
[Route("api/[controller]")]
public class ApartmentsController : ControllerBase
{
    private readonly ApplicationDbContext _context;

    public ApartmentsController(ApplicationDbContext context)
    {
        _context = context;
    }

    [HttpGet]
    public async Task<ActionResult<IEnumerable<Apartment>>> GetApartments()
    {
        return await _context.Apartments.ToListAsync();
    }

    [HttpGet("{id}")]
    public async Task<ActionResult<Apartment>> GetApartment(int id)
    {
        var apartment = await _context.Apartments.FindAsync(id);

        if (apartment == null)
        {
            return NotFound();
        }

        return apartment;
    }

    [HttpPost]
    public async Task<ActionResult<Apartment>> PostApartment(Apartment apartment)

```

```

{
    _contextApartments.Add(apartment);
    await _context.SaveChangesAsync();

    return CreatedAtAction(nameof(GetApartment), new { id = apartment.Id }, apartment);
}

[HttpPut("{id}")]
public async Task<IActionResult> PutApartment(int id, Apartment apartment)
{
    if (id != apartment.Id)
    {
        return BadRequest();
    }

    _context.Entry(apartment).State = EntityState.Modified;

    try
    {
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!ApartmentExists(id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }

    return NoContent();
}

[HttpDelete("{id}")]
public async Task<IActionResult> DeleteApartment(int id)
{
    var apartment = await _contextApartments.FindAsync(id);
    if (apartment == null)
    {
        return NotFound();
    }

    _contextApartments.Remove(apartment);
    await _context.SaveChangesAsync();

    return NoContent();
}

private bool ApartmentExists(int id)

```

```

    {
        return _contextApartments.Any(e => e.Id == id);
    }
}

export interface Apartment {
    id: number;
    title: string;
    description: string;
    address: string;
    price: number;
    photos: string[];
    contactInfo: string;
}

import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';
import { Apartment } from './apartment.model';

@Injectable({
    providedIn: 'root'
})
export class ApartmentService {
    private apiUrl = 'https://localhost:5001/api/apartments';

    constructor(private http: HttpClient) { }

    getApartments(): Observable<Apartment[]> {
        return this.http.get<Apartment[]>(this.apiUrl);
    }

    getApartment(id: number): Observable<Apartment> {
        return this.http.get<Apartment>(`${this.apiUrl}/${id}`);
    }

    addApartment(apartment: Apartment): Observable<Apartment> {
        return this.http.post<Apartment>(this.apiUrl, apartment);
    }

    updateApartment(id: number, apartment: Apartment): Observable<void> {
        return this.http.put<void>(`${this.apiUrl}/${id}`, apartment);
    }

    deleteApartment(id: number): Observable<void> {
        return this.http.delete<void>(`${this.apiUrl}/${id}`);
    }
}

import { Component, OnInit } from '@angular/core';
import { ApartmentService } from './apartment.service';
import { Apartment } from './apartment.model';

```

```

@Component({
  selector: 'app-apartment-list',
  templateUrl: './apartment-list.component.html',
  styleUrls: ['./apartment-list.component.css']
})
export class ApartmentListComponent implements OnInit {
  apartments: Apartment[] = [];

  constructor(private apartmentService: ApartmentService) { }

  ngOnInit(): void {
    this.apartmentService.getApartments().subscribe(data => {
      this.apartments = data;
    });
  }
}

<div *ngIf="apartments.length > 0">
  <div *ngFor="let apartment of apartments" class="apartment-card">
    <h2>{{ apartment.title }}</h2>
    <p>{{ apartment.description }}</p>
    <p>{{ apartment.address }}</p>
    <p>{{ apartment.price | currency }}</p>
    <img *ngFor="let photo of apartment.photos" [src]="photo" alt="Apartment photo">
    <p>Contact: {{ apartment.contactInfo }}</p>
  </div>
</div>
<div *ngIf="apartments.length === 0">
  <p>No apartments available.</p>
</div>

```

ДОДАТОК Б

Керівництво користувача

Для початку роботи з системою оренди житлових приміщень, користувачу потрібно відкрити браузер та ввести адресу сайту у адресний рядок. Після завантаження сторінки, користувач побачить головне вікно веб-застосунку.

Головна сторінка сайту відкривається при вході на сайт або при введенні некоректної адреси. На цій сторінці знаходиться меню, де можна увійти в систему як звичайний користувач чи як власник квартири. Функціонал зображено на рисунках 1 та 2. Залежно від ролі користувача, доступні різні інструменти:

Для власника квартири: “Створити квартиру”, “Перегляд ваших квартир”.

Для орендаря: “Вподобані квартири”. У центральній частині сторінки розташована кнопка “Перейти”, яка перенаправляє на сторінку всіх квартир.

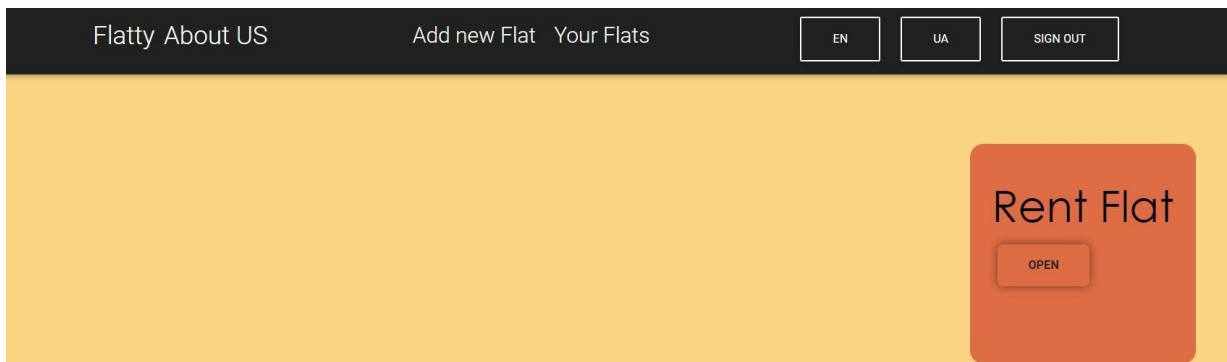


Рисунок 1 – Головна сторінка для власника

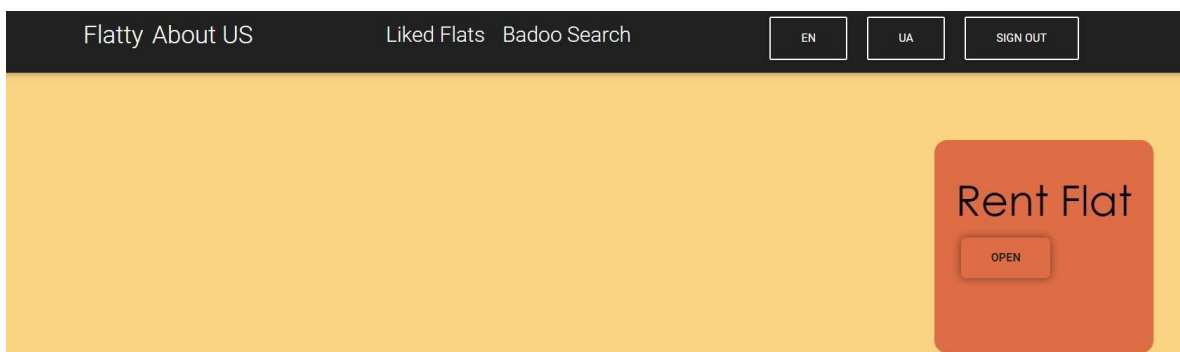


Рисунок 2 – головна сторінка для орендаря

Сторінка для реєстрації нового користувача включає різні ролі. Якщо у вас вже є обліковий запис, ви можете перейти на сторінку “Авторизація”. Реєстрація можлива у двох видах: реєстрація нового власника квартири та нового орендаря.

Реєстрація власника квартири

Натисніть кнопку “Стати власником” на головній сторінці.

Вас перенаправить на сторінку реєстрації, де потрібно заповнити всі поля: ім’я, прізвище, по-батькові, дата народження, стать, пошта, номер телефону, пароль.

Після заповнення всіх полів стає доступною кнопка “Зареєструватися”.

Реєстрація орендаря

Натисніть кнопку “Зареєструватися” на головній сторінці.

Заповніть ідентичну форму, що і при реєстрації власника квартири.

Сторінка реєстрації зображена на рисунку 3.

Flatty About US

EN UA BECAME OWNER BECAME RENTER SIGN IN

Registration

Enter your name Enter your surname Enter your father name

05/15/2021 ☐ Woman ☐ Man

Enter your email

Enter your phone

Enter your password

Confirmed your password

Have account ? [Login](#)

REGISTER

Рисунок 3 – Сторінка реєстрації

Авторизація

Сторінка для авторизації користувача за поштою та паролем має можливість перейти на сторінку реєстрації, якщо ви ще не зареєстровані. При натисканні кнопки “Увійти” на головній сторінці ви потрапляєте на сторінку авторизації. Введіть пошту та пароль. Якщо дані некоректні, кнопка “ВХІД” буде недоступною, і біля некоректних полів з’явиться повідомлення з помилкою. Якщо всі поля коректні, натисніть кнопку “ВХІД”.

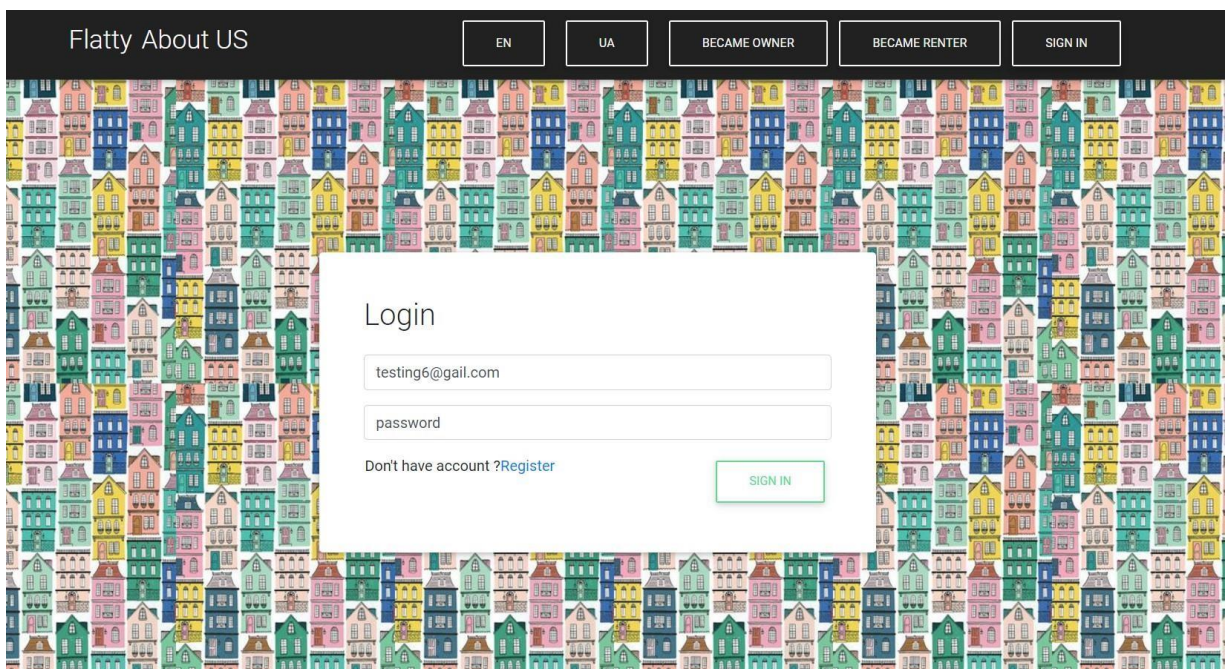


Рисунок 4 – Сторінка авторизації

Перегляд квартир

Сторінка для перегляду всіх доступних квартир має фільтр, який можна відобразити чи сховати сторінку. При натисканні на обрану квартиру ви можете перейти на її сторінку.

Пошук квартир

Для пошуку квартир скористайтесь випадającym фільтром. Натисніть кнопку “Фільтр”, щоб відкрити фільтр, введіть необхідні дані та натисніть кнопку “Пошук” для здійснення пошуку за вказаними характеристиками.

Сторінка квартири

Ця сторінка містить всю необхідну інформацію про квартиру, дозволяє переглянути всі фотографії та отримати контакти власника. Для адміністратора та власника квартири є кнопка видалення оголошення.

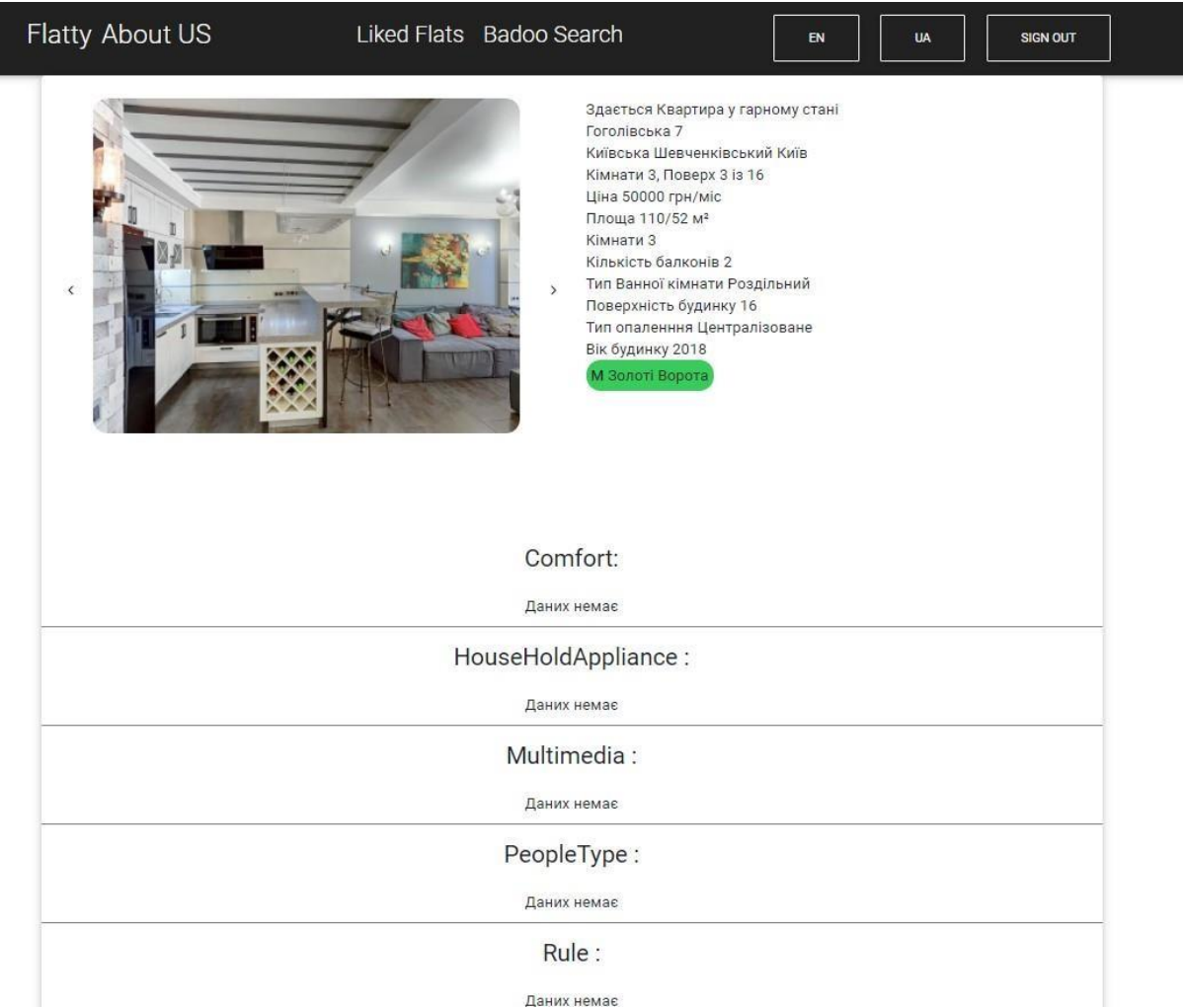


Рисунок 5 – Сторінка квартири

Перегляд квартир власника

Сторінка зображена на рисунку 6, вона показує всі квартири певного власника. Натиснувши на квартиру, ви переходите на її сторінку.

[Flatty About US](#)
[Add new Flat](#)
[Your Flats](#)

[EN](#)
[UA](#)
[SIGN OUT](#)



15900 грн
Бакинська 37
Шевченківський район, Київська область, місто Київ
Кімнат :1 Опалення : Централізоване
62 / 20 м² Будинок 1998 року
Повер 5 із

Здається Квартира у гарному стані
Оновлено : 2021-03-31 17:47:23
Створено : 2021-03-31 17:47:23



17500 грн
Перемоги 5
Шевченківський район, Київська область, місто Київ
Кімнат :1 Опалення : Централізоване
68 / 20 м² Будинок 2012 року
Повер 4 із

Здається Квартира у гарному стані
Оновлено : 2021-03-31 17:47:23
Створено : 2021-03-31 17:47:23

Рисунок 6 – Сторінка всіх квартир власника