

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіоелектроніки,
Факультет комп'ютерних наук і технологій
(повне найменування інституту, назва факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавра

(ступінь вищої освіти)

на тему ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОРОТКОСТРОКОВОЇ
ОРЕНДИ ЖИТЛА
SOFTWARE FOR SHORT-TERM ACCOMMODATION RENTAL

Виконав: студент(ка) 4 курсу, групи КНТ-218сп
Спеціальності 122 Комп'ютерні науки
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Комп'ютерні науки

Дерев'янка Є.В.

(прізвище та ініціали)

Керівник Льовкін В.М.

(прізвище та ініціали)

Рецензент Голуб Т.В.

(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
(повне найменування закладу вищої освіти)

Інститут, факультет ІРЕ, ФКНТ

Кафедра програмних засобів

Ступінь вищої освіти бакалавр

Спеціальність 122 Комп'ютерні науки

(код і найменування)

Освітня програма (спеціалізація) Комп'ютерні науки

(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.

С.О. Субботін

“ ” 2021 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

Дерев'янка Євгена Васильовича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Програмне забезпечення короткострокової оренди житла

Software for Short-Term Accommodation Rental

керівник проєкту (роботи) Льовкін Валерій Миколайович, к.т.н., доцент,

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “30” березня 2021 року № 103

2. Строк подання студентом проєкту (роботи) травень 2021 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Проєктування програмного забезпечення. 3. Основні рішення щодо реалізації компонентів системи.

4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	Льовкін В.М., доцент		
Нормоконтролер	Камінська Ж.К., асистент		

7. Дата видачі завдання “10” березня 2021 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області	2-3 тижні	Розділ 1
3	Розробка архітектури програми	4 тиждень	Розділ 2
4	Розробка програми	5-6 тижні	Розділ 3
5	Тестування та експериментальне дослідження програми	7 тиждень	Розділ 4
6	Оформлення пояснювальної записки та документів до неї. Нормоконтроль та рецензування	8 тиждень	Додатки
7	Захист роботи	9 тиждень	

Студент(ка)

_____ Дерев'янка Є.В.
(підпис) (прізвище та ініціали)

Керівник проєкту (роботи)

_____ Льовкін В.М.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 85 с., 27 рис., 1 табл., 3 дод., 15 джерел.

ОРЕНДА ЖИТЛА, КОРОТКОСТРОКОВА ОРЕНДА, MODEL-VIEW-VIEWMODEL, MICROSOFT SQL SERVER, WPF.

Об'єкт роботи – процес оренди житла. Предмет роботи – програмне забезпечення короткострокової оренди житла.

Мета роботи – розроблення програмного забезпечення короткострокової оренди житла для забезпечення взаємодії власників житла з орендарями.

Матеріали, методи та технічні засоби: мова програмування C#, патерн проєктування Model-View-ViewModel, система Microsoft SQL Server.

Результати. Розроблене програмне забезпечення дозволяє створювати заявки на оренду житла, власникам підтверджувати їх або скасовувати, сплачувати за бронювання за допомогою платіжної системи PayPal, підтверджувати орендарю заселення, а власнику – виселення, створювати відгуки про власний досвід, виконувати пошук нових варіант бронювання, переглядати власні бронювання та бронювання, оформлені на пропозиції оренди житла власника.

Висновки. Роботу акцентовано на реалізації короткострокової оренди житла на основі створення відповідної програмної платформи, де власники житла можуть взаємодіяти з орендарем без участі посередників. Для створення довіри між учасниками процесу передбачено систему відгуків на основі оцінок та текстових рецензій, які створюються як для орендарів, так і для самого житла.

Галузь використання. Платформи підтримки короткострокової оренди житла, включаючи квартири, котеджі.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 85 pages, 27 figures, 1 table, 3 appendixes, 15 sources.

ACCOMODATION RENTAL, SHORT-TERM RENTAL, MODEL-VIEW-VIEWMODEL, MICROSOFT SQL SERVER, WPF.

The object of the work is the process of accommodation rental. Subject of the work is software for short-term accommodation rental.

The purpose of the work is to develop software for short-term accommodation rental to guarantee interaction of homeowners with tenants.

Materials, methods and technical means: C# programming language, Model-View-ViewModel design pattern, Microsoft SQL Server system.

Results. The developed software allows users to create rental applications, allows owners to confirm or cancel applications, to pay for reservations using the PayPal payment system, to confirm the occupancy by a tenant and eviction by an owner, to create reviews of own experience, to search for new reservations, to view own reservations and bookings made on propositions of rental of owner's habitation.

Conclusions. The work focuses on the implementation of short-term accommodation rental based on the creation of an appropriate software platform where homeowners can interact with tenants without the participation of intermediaries. For the creation of trust between participants of the rental process, there is a system of feedback based on ratings and text reviews, which are created both for tenants and for the accomodation itself.

Field of usage. Platforms for support of short-term accommodation rental, including apartments, cottages.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	7
Вступ.....	8
1 Аналіз предметної області.....	10
1.1 Проблема короткострокової оренди житла.....	10
1.2 Огляд програмних аналогів.....	14
1.3 Висновки за розділом 1	18
2 Проєктування програмного забезпечення	20
2.1 Вибір мови програмування	20
2.2 Вимоги до програмного забезпечення	22
2.3 Проєктування бази даних	25
2.4 Визначення архітектури програмного забезпечення.....	30
2.5 Висновки за розділом 2	31
3 Основні рішення щодо реалізації компонентів системи.....	32
3.1 Структура розробленого програмного забезпечення	32
3.2 Опис логіки реалізації.....	35
3.3 Висновки за розділом 3	37
4 Експлуатація, тестування та експериментальне дослідження програми	38
4.1 Призначення програми	38
4.2 Умови виконання програми	38
4.3 Тестування сценаріїв роботи з програмою.....	39
4.4 Висновки за розділом 4	50
Висновки	51
Перелік джерел посилання	52
Додаток А Технічне завдання	54
Додаток Б Текст програми	58
Додаток В Слайди презентації.....	79

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

MVVM	– Model-View-ViewModel;
WPF	– Windows Presentation Foundation;
БД	– база даних.

ВСТУП

З активним розвитком бізнесу поступово розвивається і короткострокова оренда житла, адже вона підживлюється туризмом, робочими процесами, пов'язаними з відрядженнями, розважальною галуззю. Усі ці варіанти оренди передбачають оренду саме на строк до місяця, а частіше – на декілька днів або тиждень. Раніше все це було пов'язано з необхідністю взаємодії через певних агентів, відповідно заважало розвитку адже збільшення вартості для орендарів призводило до зменшення конкурентоспроможності порівняно з готелями. Тепер, з розвитком інформаційних технологій, ситуація змінилася.

Одночасно варто зазначити, що тільки програмні засоби дозволяють створити функціональність у цій сфері, що напряму впливає на її розвиток. Це інструменти зворотного зв'язку, за допомогою яких власники можуть дізнатися про орендарів, адже від них залежить стан їх квартири після оренди, при чому не з загальних джерел, а від таких самих господарів, у яких вони проживали раніше. У свою чергу орендарі можуть дізнатися про актуальний стан житла, яке вони збираються винаймати. В Україні з цим є певні проблеми, адже вебсайти, на яких відбувається рекламування житла (саме цей напрямок даної проблеми розвивався більш-менш активно на відміну від самої оренди в цифровій формі), часто мають неактуальну інформацію і фактичний досвід проживання відрізняється від того, який був описаний господарем декілька років тому одразу після виконання ремонту.

На даний момент в Україні тим не менш відчувається певний брак таких програмних засобів. У даній сфері окрім загальнонаціональних корпорацій у багатьох країнах існують власні платформи, що доповнюють і допомагають розвиватися галузі. Такі програмні засоби можуть зокрема підтримувати активно роботу в національній валюті, надають більше національних пропозицій житла. В Україні з цим проблеми. Тому створення

програмного забезпечення короткострокової оренди житла можна вважати актуальним.

Метою роботи в результаті було визначено розроблення програмного забезпечення короткострокової оренди житла для забезпечення взаємодії власників житла з орендарями.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Проблема короткострокової оренди житла

Короткотермінова оренда (зазвичай її називають орендою на відпустку) – це здача в оренду мебльованого нерухомого майна на короткостроковій основі, залежно від власника, місця оренди та багатьох інших факторів, ці об'єкти здаються в оренду на тиждень або на ніч, що відзначаються наступним:

- багато власників на короткий термін орендують своє майно протягом більшої частини року, коли вони не користуються ним самостійно;

- короткострокова оренда стала надзвичайно популярною альтернативою готелям протягом останніх 20-ти років, особливо, завдяки сплеску вебсайтів, таких як HomeAway, Airbnb та VRBO;

- на відміну від короткострокової оренди, довгострокова оренда житла, як правило, відноситься до випадків, коли здається житло в оренду на період від одного місяця або довше, незалежно від того, чи це цілий будинок, чи лише одна кімната, яку здають в оренду, загалом орендарі платять власнику житла щомісяця і зазвичай беруть на себе інші витрати, такі як комунальні платежі, на відміну від короткострокової оренди житла [1].

Переваги короткострокової оренди житла:

- вищий потенційний дохід від орендної плати: можна не тільки встановити коливальні ставки залежно від високого та низького сезону в районі, але також можна встановити мінімальну тривалість перебування для найпопулярніших пір року, щоб забезпечити максимальний прибуток;

- більша гнучкість: для власників, які надають в оренду свою нерухомість на короткий термін, передбачено набагато більше можливостей, можна спеціально заблокувати календарні періоди, які потрібно залишити безкоштовними для особистого користування, не завдаючи нікому незручностей;

– менший знос: загалом тривалість оренди становить від декількох днів до максимум декількох тижнів, приймаючи гостей короткими сплесками, так можна гарантувати що гості просто відвідають власність і не будуть думати про ремонт чи перестановку меблів;

– податкові пільги та відрахування: багато власників орендованих будинків мають право на певні податкові пільги або навіть на відшкодування майнових витрат, оскільки майно не дається на тривалий термін;

– сприяння економії спільного використання: передаючи відпочиваючим своє майно на короткий термін, власник показує, що він новатор, який допомагає внести свій внесок у цей сучасний спосіб життя та мислення [1].

Ефективне керування майном вимагає слідування основним принципам:

– присутність: першим кроком є розуміння важливості максимізації присутності для списків житла, якими керують, найкраща послуга з управління майном нічого не означає, якщо немає властивостей, якими можна насправді керувати, і найпростіший спосіб забезпечити постійний приплив замовлень – це максимізувати шанси на те, що потрібні об'єкти знайдуться в першу чергу, зростання та велика кількість сайтів бронювання та Інтернет-агентств подорожей представляє як виклик, так і можливість (якщо правильно керувати оголошеннями, розміщення на декількох сайтах відкриє менеджеру нерухомості ще до тисяч мандрівників, бажаючих забронювати об'єкти, якими менеджер керує), для досягнення останнього потрібна допомога інструментів та технологій, що перетворюють складність розширеного портфолію в простоту централізованої платформи управління, найкращим інструментом для цього є менеджер каналів, який дозволяє просто і легко керувати декількома обліковими записами на різних платформах;

– ефективність: керування орендою нерухомості на короткий термін не залишає можливості виконувати будь-яке завдання менш ефективним

способом, досягнення цієї мети також залежить від інструментів та технологій, які вирішено впровадити у бізнес, менеджери нерухомості повинні використати нові та інноваційні засоби автоматизації, які дозволять ставити багато щоденних завдань на автоматичне виконання, не погіршуючи при цьому якість результатів або рівень послуг, зокрема спілкування з гостями, щоденне встановлення цін або залишення відгуків відвідувачів – багато з цих повторюваних завдань можуть бути замінені використанням технологій;

– рефлексорність: у пошуках автоматизації та спрощення легко забути про необхідність надання стандартів обслуговування гостям, утримання щасливих гостей є критичним фактором для стійкого управління портфелем об'єктів нерухомості, якщо якість обслуговування зникне, портфель швидко розсиплеться, найпростіший спосіб продемонструвати увагу та відданість, яку очікують гості, – залишатися чуйним на всі їхні бажання та потреби, гості можуть бути різними, але в кінцевому рахунку вони мають однакові сподівання щодо надання швидких відповідей на їхні запитання, саме це призводить до отримання рейтингів у 5 зірок, потрібно переконатися, що спілкування відбувається за найкращими каналами зв'язку для гостей (наприклад, деякі гості віддають перевагу електронній пошті, інші можуть надавати перевагу SMS-повідомленням, треті – месенджерам, потрібно знайти рішення, яке враховує різні комунікативні уподобання різних типів гостей, не обтяжуючи себе складністю необхідності орієнтуватися на декількох платформах зв'язку);

– динамізм: ефективне управління майном – це настільки ж мислення, як і діяльність, хороше короткострокове управління майном – це цілодобова робота цілий рік, технічний прогрес звільнив розпорядників майна від обмежень фізичного офісу чи визначеного місця, але при цьому хоча вдома, хоч у автобусі або в супермаркеті – менеджери мають взяти на себе 99% своїх основних обов'язків щодо розміщення, де б вони не опинились (спілкування з гостями, оновлення їх календарів чи спілкування з

прибиральниками), так високоефективні менеджери з нерухомості працюють динамічно та в дорозі, не обмежуючись своїм фізичним розташуванням;

– постійне покращення: успішні менеджери з нерухомості живуть за принципами постійного вдосконалення, вони використовують свій багаторічний досвід розміщення, щоб створити чудовий досвід для своїх гостей, а також вигідні результати для своїх клієнтів, щоб постійно отримувати ці результати, вони застосовують підхід до прийняття рішень, що базується на даних (використовується принцип, що не можна покращити те, що не можна виміряти), вони невблаганно підтверджують свою ефективність, використовуючи статистичні дані, які висвітлюють, що виконується добре, а також те, де щось можна зробити краще, тобто вони ставлять себе на шлях постійного вдосконалення та створюють прозорість для клієнтів, яких вони представляють [2].

У даній сфері розрізняють системи, які направлені на організацію сервісу короткострокової оренди житла і відповідно створюють інтерфейс окрема і для орендарів у першу чергу, і системи, які направлені на роботу менеджера нерухомості.

Загалом другі системи короткострокової оренди житла направлені не на саме забезпечення бронювання житла, а швидше на надання інструментів для агрегації різних таких засобів у одній точці і включають такі особливості:

– календар: один календар, де можна побачити всі бронювання з усіх каналів для всіх переліків житла;

– управління бронюванням: модуль системи бронювання дає можливість змінювати дати, ціни та інші деталі для існуючих бронювань;

– управління каналами: синхронізація швидкості, доступності та інколи вмісту на всіх сайтах, таких як AirBnB, VrBO, Homeaway та Flipkey, за допомогою менеджера каналів можна змінити доступність у системі управління власністю і дозволити надсилати зміни по всіх каналах, що входять до переліку;

- механізм бронювання: програмне забезпечення для бронювання надає можливість вводити прямі бронювання, заброньовані в автономному режимі (по телефону або особисто) або в Інтернеті (через власний вебсайт або віджет бронювання), а також збільшувати дохід від прямих каналів;

- інтеграції: система управління нерухомістю може пропонувати бібліотеку інтеграцій, яка варіюється від платіжних процесорів до динамічних систем ціноутворення та рішень для безключового доступу;

- мобільний застосунок: багато власників орендованих будинків управляють своїми об'єктами віддалено, тому у складі системи зручним буде наявність також додатково мобільного застосунку, щоб вести свій бізнес у дорозі де завгодно та коли завгодно [3].

У даній роботі акцент буде зроблено на розробці програмного забезпечення, яке підтримує в першу чергу замовлення орендарів на короткострокову оренду житла та надає необхідні для цього засоби для орендодавців.

1.2 Огляд програмних аналогів

Airbnb [4] – історично іноваційний застосунок для оренди житла в подорожах з тисячами різноманітних переліків нерухомості – від одиночних квартир до якірних яхт та замків, ця програма дозволяє встановити комунікацію з власниками з усього світу, загалом надається більше ніж 2 мільйони варіантів житла у понад 34 000 різних містах, програма відзначається наступним:

- програмне забезпечення працює в рамках хостингової системи (списки формуються безкоштовно та підбираються різними власниками, які сплачують певну суму грошей на тиждень, ніч або місяць);

- Airbnb надає своїм користувачам надійну, широку спільноту та безпечну платіжну систему;

– доступні настроювані параметри пошуку, при цьому можна будь-коли здійснювати пошук у базі даних (БД) як гість, але для реєстрації будь-якого бронювання потрібен профіль: можна переглянути апартаменти, квартири, особняки, будинки для відпочинку за ціною, розміром, мовою господаря та типом кімнати, можна також ввести конкретні ключові слова у пошукову систему та переглянути як стандартні місця розташування, так і нетрадиційні варіанти (такі як замки, яхти та печери), крім того можна керувати своїми списками безпосередньо зі свого смартфона, додавати улюблені розташування в закладки та ділитися потрібними місцями з друзями та родиною через соціальні мережі;

– після того, як було знайдено потрібне житло, необхідно натиснути на кнопку “Забронювати” та ввести свої платіжні реквізити, також можна отримати вказівки для досягнення обраного місця;

– господар може миттєво відповідати на повідомлення, приймати або відхиляти запити, надсилати оновлення та знижки для своїх гостей, а також попередньо затверджувати певні запити, після підтвердження бронювання мандрівником Airbnb утримує кошти для оплати протягом 24 годин, перш ніж віддати їх господареві, застосунок також бере плату в розмірі 3% за бронювання від власника, щоб компенсувати витрати на обробку транзакції;

– важливу роль у роботі платформи Airbnb відіграє функція відгуків: відгуки допомагають зміцнити довіру між господарем та орендарем, відповідно господарі покладаються на відгуки гостей, щоб уникнути поганих гостей, більшість господарів завжди переглядають відгуки гостей Airbnb перш ніж підтвердити бронювання, тому для заохочення Airbnb надсилає електронне повідомлення та повідомлення в застосунок після кожної поїздки з проханням переглянути відгук гостя, окрім того і заохочує гостей залишити відгук, надіславши повідомлення, таким чином і гість, і власник збільшать кількість своїх відгуків та отримають більше підтверджених запитів і замовлень відповідно (рис. 1.1) [5]-[6].

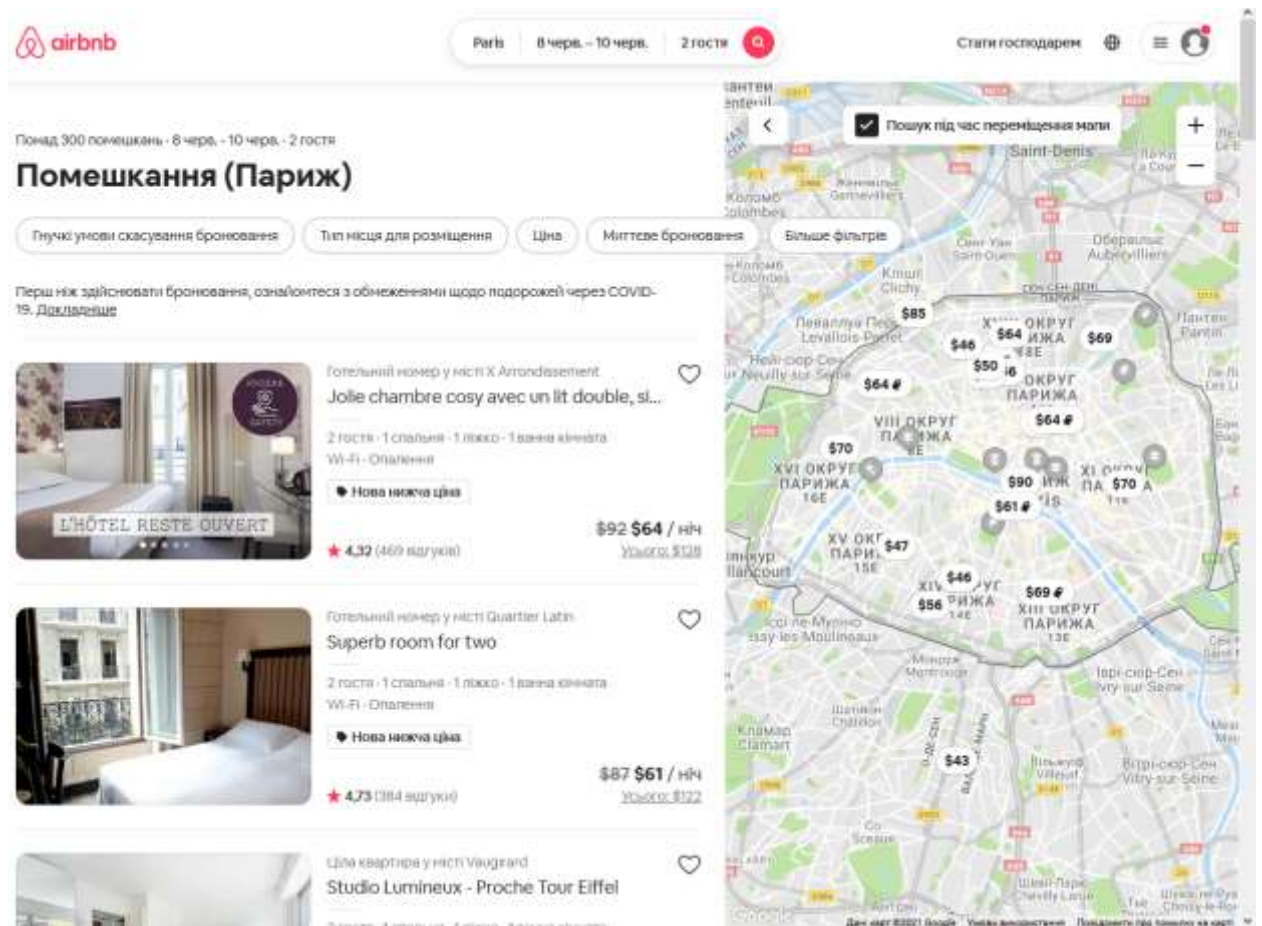


Рисунок 1.1 – Інтерфейс Airbnb

Ще одним програмним аналогом розроблюваного програмного забезпечення є VrBO [7] (рис. 1.2), який пропонує цілий ряд видів оренди, включаючи будинки, квартири, замки, будиночки на деревах тощо, однак, на відміну від Airbnb, можна орендувати лише цілі будинки, тому варіантів оренди тут менше ніж у Airbnb (2 мільйони варіантів оренди по всьому світу порівняно з 4 мільйонами Airbnb), окрім того щодо особливостей можна відзначити наступне:

- існує розділ огляду, який дозволяє користувачам оцінювати своє перебування у певному місці, окрім того сайт дозволяє власникам розміщувати відповідь рецензенту, яка відображається під даним оглядом, це означає, що користувачі VrBO можуть не тільки дізнатись більше інформації про досвід перебування в помешканні, вони також можуть отримати більше

розуміння від власників будинків, перш ніж розпочати процес бронювання проживання;

– на відміну від Airbnb програма має величезний набір фільтрів: користувачі можуть шукати за типом місцезнаходження (пляж, вид на озеро, гори), за більш детальним типом нерухомості, фільтри VrBO є більш повними та надійними [8]-[9].

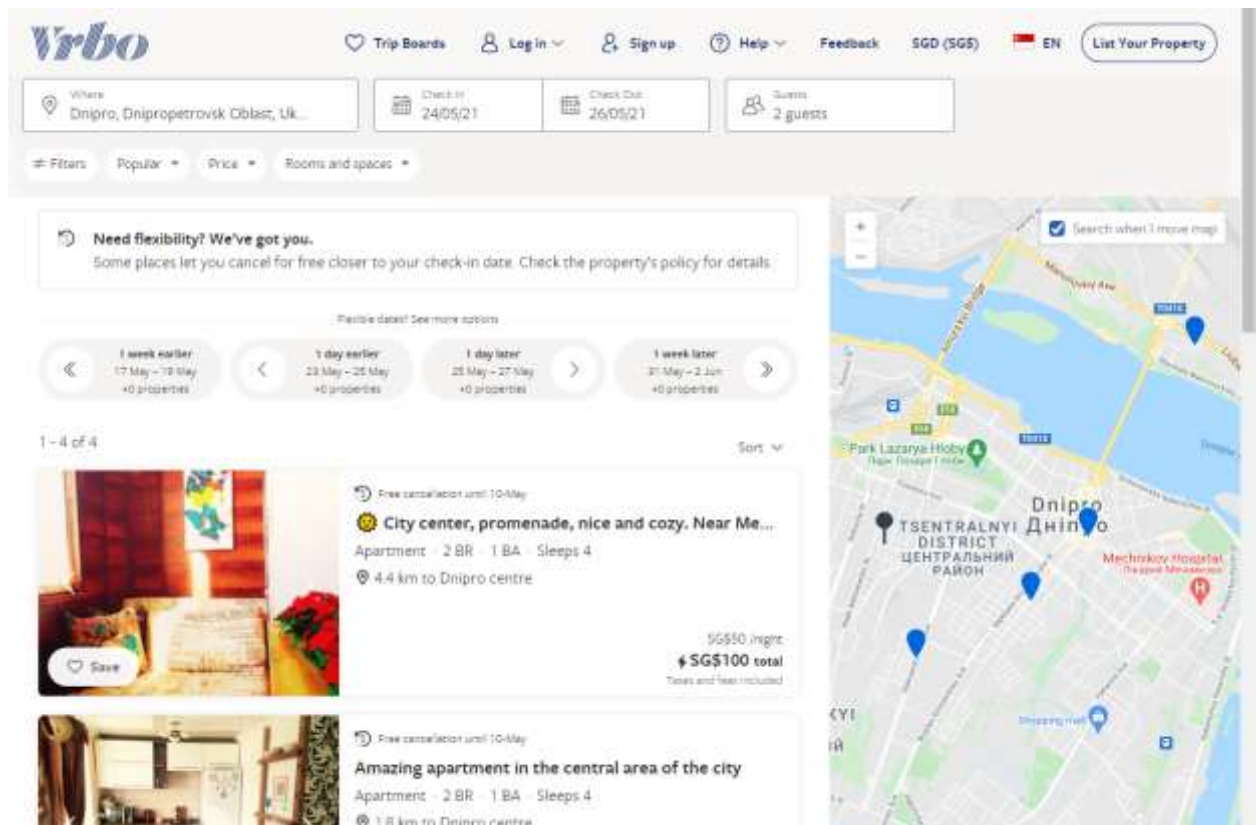


Рисунок 1.2 – Інтерфейс VrBO

Flipkey [10] (рис. 1.3) – це онлайн-ринок для оренди житла на відпустку, який існує з 2007 року, а у 2008 році він став частиною TripAdvisor Rentals, зараз надається понад 830 000 варіантів житла майже в усіх країнах світу: на сторінці результатів пошуку з переліком доступних варіантів житла визначається вартість житла за день без податків, а загальна сума відображається на сторінці самого житла, яке визначається наступними параметрами:

– ціна за ніч;

- загальна вартість;
- кількість спалень та ванних кімнат;
- кількість гостей, що дозволяється;
- відгуки на Tripadvisor;
- час відповіді власника;
- будь-які отримані нагороди [11].

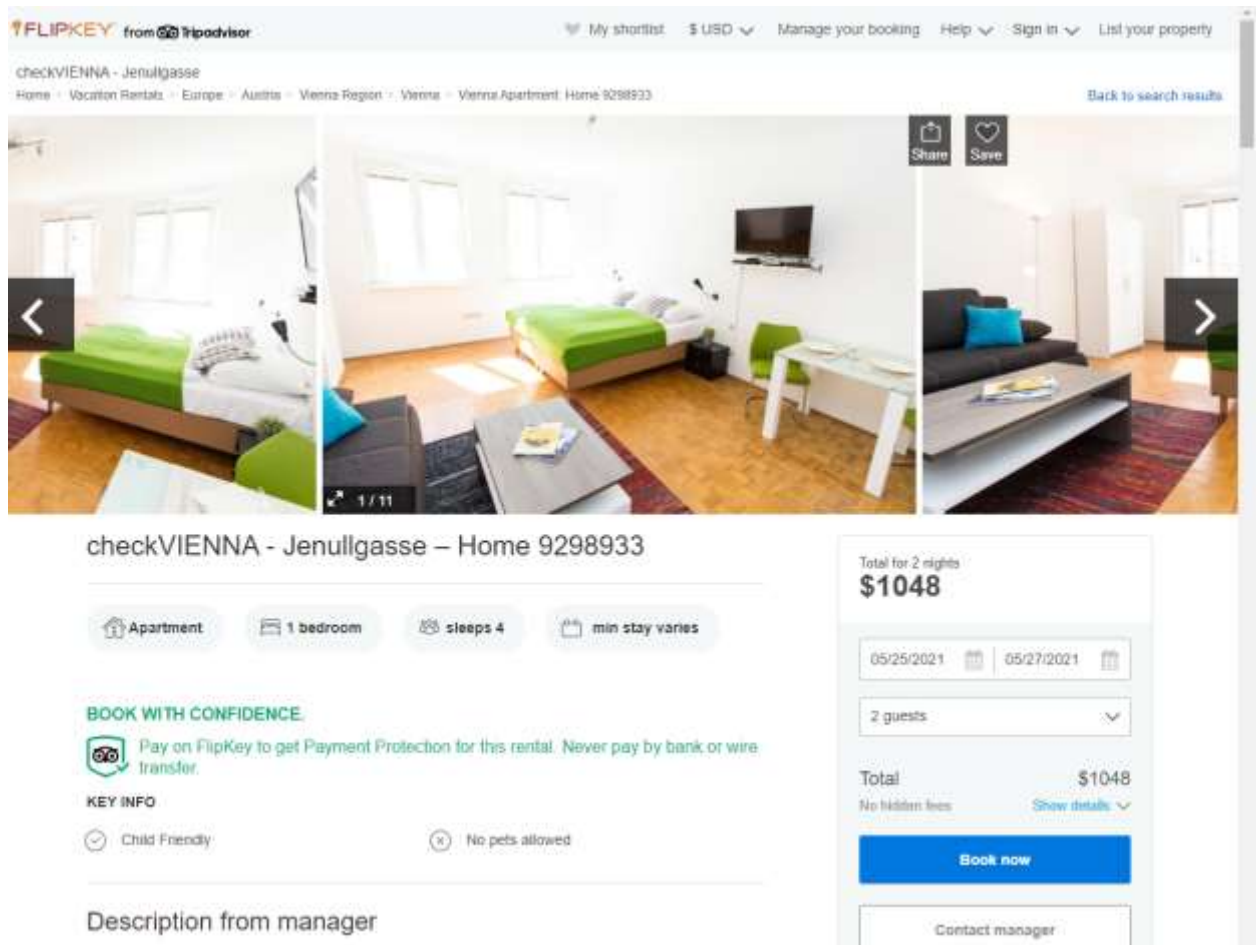


Рисунок 1.3 – Інтерфейс Flipkey

1.3 Висновки за розділом 1

У першому розділі виконано аналіз проблеми короткострокової оренди житла: розглянуто особливості короткострокової оренди порівняно з довгостроковою, особливості організації програмного забезпечення для підтримки даного процесу, виконано аналіз програмних аналогів.

У даній дипломній кваліфікаційній роботі має бути розв'язано наступні завдання:

- визначення вимог до програмного забезпечення короткострокової оренди житла;
- вибір мови і засобів проєктування і розробки;
- проєктування програмного забезпечення;
- реалізація і налагодження програмного забезпечення.

2 ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Вибір мови програмування

Створення застосунку для роботи під керуванням операційної системи Windows з потенційним розширенням на інші настільні системи є завданням даної роботи. У якості альтернатив було розглянуто мови програмування C# [12]-[13] та Java [14] (табл. 2.1).

Мова програмування C# дозволяє створювати графічний інтерфейс користувача на основі системи Windows Presentation Foundation (WPF).

Таблиця 2.1 – Вибір мови програмування

Критерій	Java	C#
Об'єктно-орієнтована парадигма програмування	Так	Так
Корпоративний рівень взаємодії з БД	Так	Так
Створення кросплатформних рішень	Так	Так, за рахунок платформи .NET Core
Підтримка фреймворків розробки	Так	Так
Виконання програм	Середня швидкість	Висока швидкість
Проектування і розробка графічного інтерфейсу	Так	Зручна у WPF-рішеннях за рахунок відділення етапу

У результаті перевагу віддано мові програмування C# [12]-[13], хоча альтернатива у вигляді мови Java достатньо конкурентна і мало відрізняється, хоча дослідження [15] щодо швидкості виконання програм вказують, що програми мовою C# швидші за програми мовою Java.

Мова C# відзначається наступними особливостями:

- дозволяє створювати код на основі об'єктно-орієнтованої парадигми в основі, але при цьому надає також інші парадигми, які можна використати для реалізації окремих особливостей, окрім того на рівні створення коду мова своїми засобами спрощує цей процес, в результаті доводиться писати коду менше;

- початково була призначена для роботи з Windows-застосунками, але з часом за рахунок створення платформи .NET Core можливо стало запускати створені в результаті застосунки на інших платформах також, що зробило рішення кросплатформними і відповідно нівелювало, наприклад, переваги застосунків, створених мовою Java;

- надається великий набір готових бібліотек, шаблонів створення програмного забезпечення, що спрощує процес розробки, особливо під час написання Windows-застосунків з графічним інтерфейсом;

- засоби розробки достатньо зручні у використанні, не виникає потреби використовувати інші додаткові рішення, відповідно не витрачається додатковий час;

- використання мови програмування C# передбачає використання базово цілої екосистеми засобів, до яких входить не тільки середовище розробки, але і, наприклад, система керування базами даних, що є достатньо надійною і ефективною;

- суворі типізація, що призводить до уникнення появи додаткових помилок;

- надає багато засобів створення застосунків з графічним інтерфейсом, що охоплюють зокрема Windows Forms, Windows Presentation Foundation.

2.2 Вимоги до програмного забезпечення

Передбачається, що виставляти житло на оренду будуть самі господарі, тобто система не передбачає роботу посередників. Виставлятися може тільки цілісне житло, тобто оренда кімнат в квартирі не передбачається. Окрім того для оплачування має використовуватися єдина валюта.

З програмою можуть працювати неавторизований користувач і авторизований. Неавторизований користувач має обмежену функціональність і для виконання основних дій має авторизуватися.

Авторизований користувач у будь-яку мить може виступати і як орендар, і як власник житла. Тобто функціональність для виконання і тих, і тих дій доступна кожному користувачу. Для того щоб стати власником, потрібно лише створити власне оголошення. Після цього стає доступною вся інша функціональність роботи з програмою в якості власника, що передбачає керування власним житлом.

Неавторизований користувач може використовувати наступні функції програми (рис. 2.1):

- пошук житла для бронювання;
- перегляд оголошення про оренду житла;
- розрахунок вартості проживання за заданими параметрами;
- перегляд даних про власника житла;
- перегляд відгуків про житло;
- реєстрація і авторизація.

Для неавторизованого користувача варіант використання «реєстрація» є альтернативним потоком, адже використовується один раз, а у всі інші – авторизація.

Для авторизованого користувача альтернативними потоками є відправлення повідомлення і скасування власного бронювання, адже

виконуються вони тільки за потреби, а в стандартний потік взаємодії не ВХОДЯТЬ.

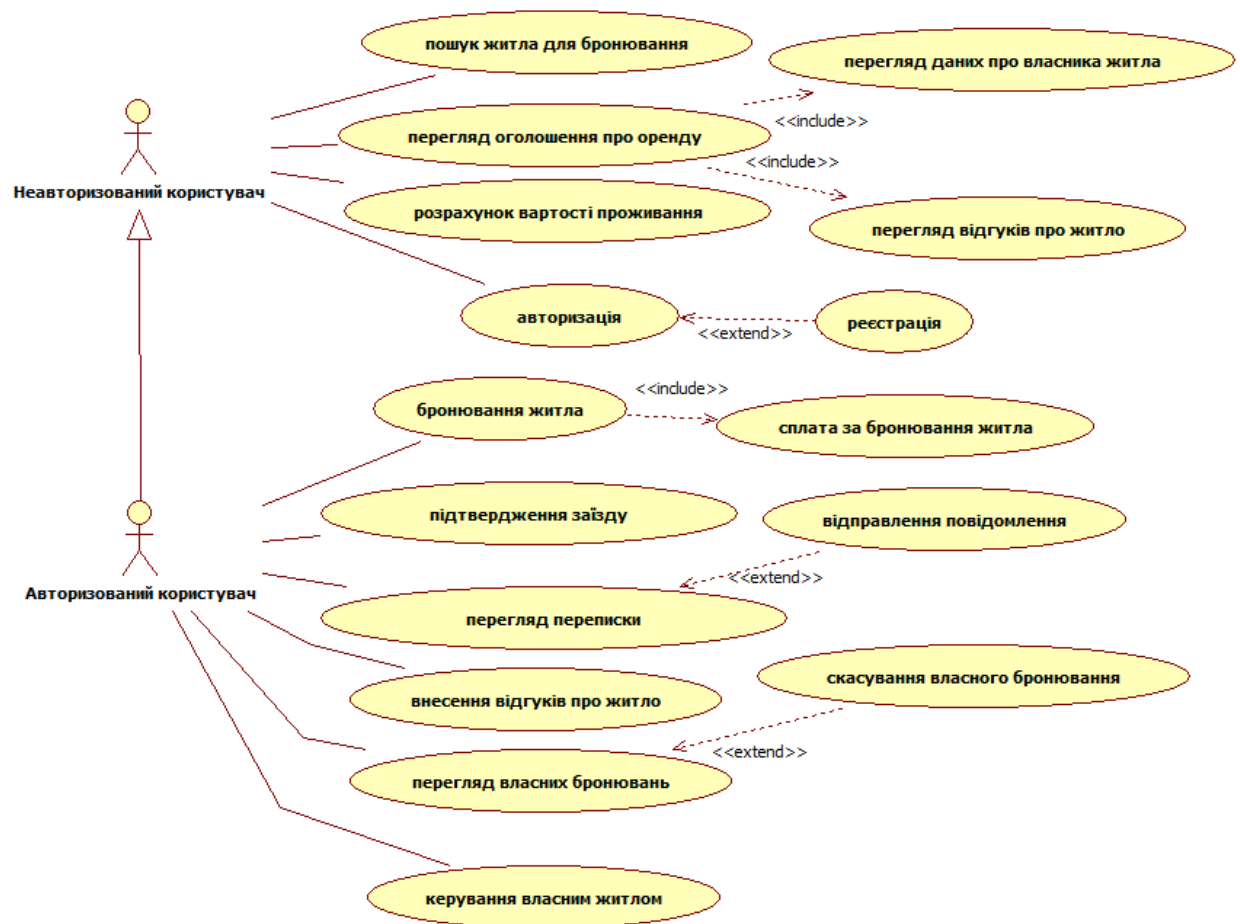


Рисунок 2.1 – Діаграма прецедентів роботи з програмою

Авторизований користувач може використовувати наступні функції:

- пошук житла для бронювання;
- перегляд оголошення про оренду житла;
- розрахунок вартості проживання за заданими параметрами;
- перегляд даних про власника житла;
- перегляд відгуків про житло;
- виставлення власного оголошення про бронювання житла;
- перегляд списку власних оголошень;
- визначення вартості бронювання житла на проміжок часу;
- визначення доступності житла на проміжок часу;

- бронювання житла;
- сплата за бронювання житла;
- відправлення повідомлень власнику житла;
- перегляд переписки з власником;
- створення відповідей на повідомлення орендаря;
- внесення відгуків про житло після завершення бронювання;
- підтвердження проживання власником;
- підтвердження заїзду орендарем;
- перегляд власних бронювань;
- підтвердження бронювання власником (рис. 2.2);
- скасування бронювання власником;
- скасування власного бронювання;
- перегляд бронювань власного житла;
- авторизація.



Рисунок 2.2 – Діаграма прецедента «Керування власним житлом»

2.3 Проектування бази даних

Питання організації роботи з даними було вирішено шляхом створення БД на основі системи Microsoft SQL Server.

Таблиця Accomodation визначає саме житло, оренда якого реалізується, і характеризується наступними полями:

- ID – номер житла;
- Title – назва;
- Description – опис житла;
- Region – район, де розташовано житло;
- Rooms – кількість кімнат;
- Floors – кількість поверхів;
- Bedrooms – кількість спальних кімнат;
- Beds – кількість ліжок;
- WiFi – наявність безпроводного інтернету;
- Parking – наявність безкоштовного паркувального місця;
- Conditioner – наявність кондиціонера;
- TV – наявність телебачення;
- Acc_Type – тип житла (квартира, таунхаус, котедж);
- Heating – наявність обігріву;
- Kitchen – наявність кухні;
- Bathroom – наявність ванної кімнати;
- MaxStay – максимальна кількість днів, протягом якої можна виконати оренду;
- Washer – наявність пральної машини;
- Elevator – наявність ліфта;
- Refund – спосіб повернення грошей при скасуванні бронювання;
- Guests – максимальна кількість гостей;
- Pets – можливість проживання з домашніми тваринами;
- Address – повна адреса в місті;

- City_ID – місто;
- Points – сумарна кількість балів, набрана за відгуками;
- Views – кількість відгуків.

Таблиця Booking визначає бронювання і характеризується наступними полями:

- ID – номер бронювання;
- User_ID – користувач, який забронював житло;
- Accomodation_ID – житло;
- From – початок бронювання;
- To – завершення бронювання;
- Fixed – час, коли було виконано бронювання;
- Status – стан бронювання (активний, скасований, виконаний);
- Price – сплачена сума;
- Guests – кількість гостей;
- Infants – кількість дітей.

Таблиця AccomodationPrice визначає вартість проживання в житлі і характеризується наступними полями:

- ID – номер бронювання;
- Accomodation_ID – номер житла;
- From – дата, з якої активна дана вартість;
- To – дата, до якої активна дана вартість;
- Price – вартість за добу;
- Guests – кількість гостей.

Таблиця Photo визначає зображення житла і характеризується наступними полями:

- ID – номер бронювання;
- Accomodation_ID – номер житла;
- Route – шлях до файла з зображенням.

Таблиця AccomodationReview визначає відгук на проживання і характеризується наступними полями:

- ID – номер відгуку;
- Accomodation_ID – номер житла;
- User_ID – користувач, який забронював житло і залишив відгук;
- Message – текст відгуку;
- Points – оцінка у відгуку;
- Fixed – дата створення відгуку.

Таблиця User визначає користувача програми і характеризується наступними полями:

- ID – номер користувача;
- Surname – прізвище користувача;
- Name – ім'я користувача;
- City – місто проживання користувача;
- Points – сумарна кількість балів, набрана за відгуками;
- Views – кількість відгуків;
- Passport – паспорт;
- About – інформація про користувача (власний опис);
- Mail – електронна пошта користувача;
- Phone – номер телефону користувача;
- Status – стан (необхідний для можливості блокування: заблокований або активний);
- User_Type – тип облікового запису користувача (власник або орендар).

Таблиця City визначає місто і характеризується наступними полями:

- ID – номер міста;
- Title – назва міста;
- Country_ID – номер країни.

Таблиця Country визначає країну і характеризується наступними полями:

- ID – номер країни;
- Title – назва країни.

Таблиця Communication визначає персональне повідомлення і характеризується наступними полями:

- ID – номер повідомлення;
- FromUser_ID – автор повідомлення;
- ToUser_ID – адресат повідомлення;
- Fixed – дата створення повідомлення;
- Message – текст повідомлення.

Таблиця Language визначає мову і характеризується наступними полями:

- ID – номер мови;
- Title – назва мови.

Таблиця UserLanguage визначає мову, якою спілкується користувач, необхідна для реалізації зв'язку «багато-до-багатьох» між таблицями мови і користувача, оскільки один користувач може знати декілька мов, а кожну мову відповідно може знати багато користувачів.

Таблиця UserLanguage характеризується наступними полями:

- ID – номер;
- User_ID – номер користувача;
- Language_ID – номер мови, якою спілкується користувач.

Таблиця UserReview визначає відгук на користувача від власника житла і характеризується наступними полями:

- ID – номер відгука;
- User_ID – користувач, щодо якого оформлено відгук;
- RevUser_ID – користувач, який створив відгук;
- Message – текст відгуку;
- Points – кількість балів;
- Fixed – дата внесення відгука.

На рис. 2.3 представлена схема БД.

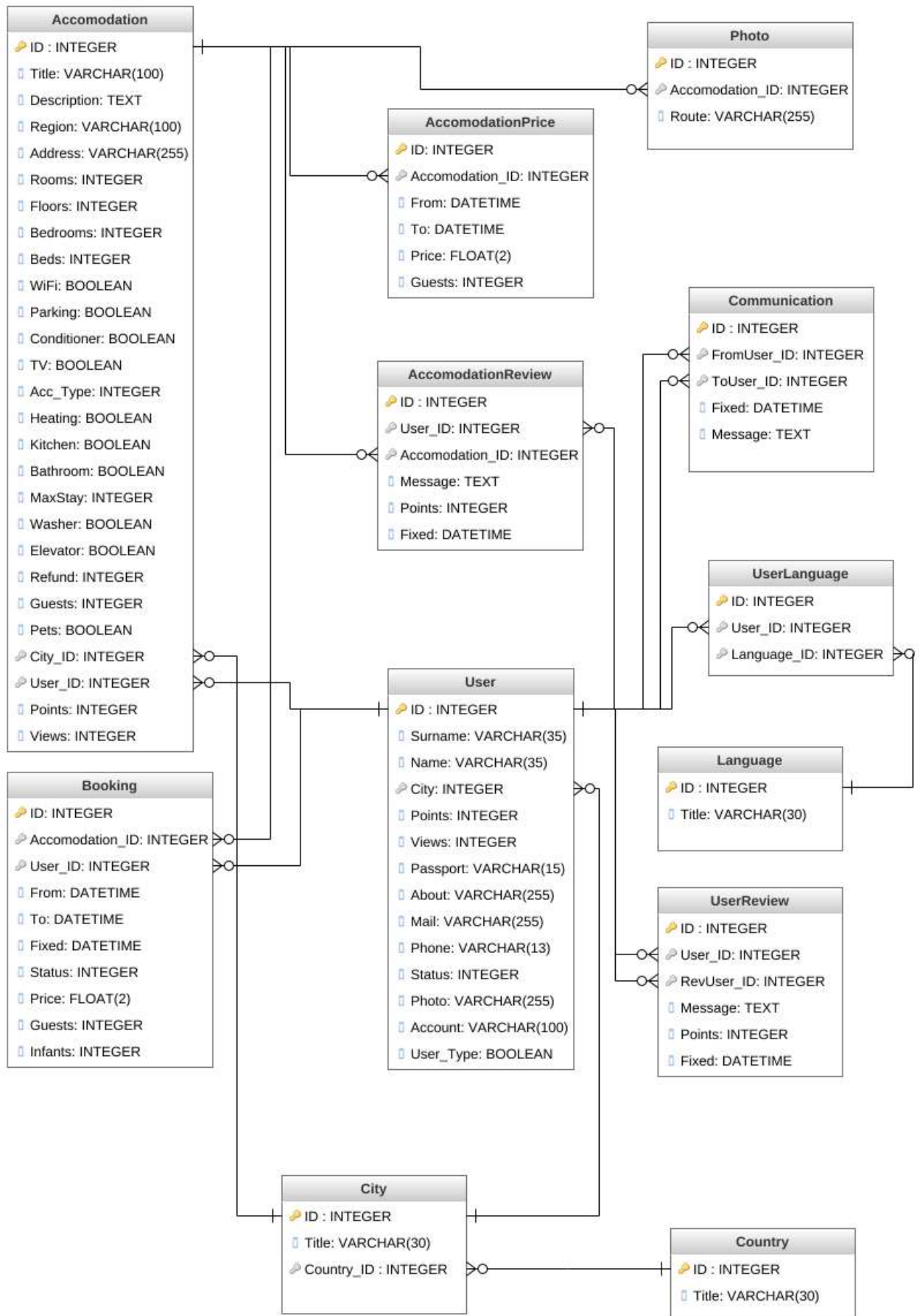


Рисунок 2.3 – Схема БД

2.4 Визначення архітектури програмного забезпечення

Для побудови програмного забезпечення було використано архітектурний шаблон Model-View-ViewModel (MVVM). Таке рішення дозволяє зокрема застосовувати реалізацію предметної області у вигляді моделі для систем з подібною логікою роботою з даними, але іншим графічним інтерфейсом зокрема. Окрім того використання MVVM покращує процеси тестування і подальшої підтримки розробленого застосунку. Все це стало аргументами на користь використання даного підходу.

Архітектурний шаблон MVVM (рис. 2.4) складається з наступних складових:

- моделі, які структурують дані в предметній області, представляючи окрім безпосередньо самої структури даних також бізнес-логіку;
- представлення, які визначають, у якому вигляді дані будуть візуалізовані користувачу;
- моделі представлень, які призначені для формування прив'язування даних, формування даних у тому вигляді, в якому вони будуть відображені в представленнях.

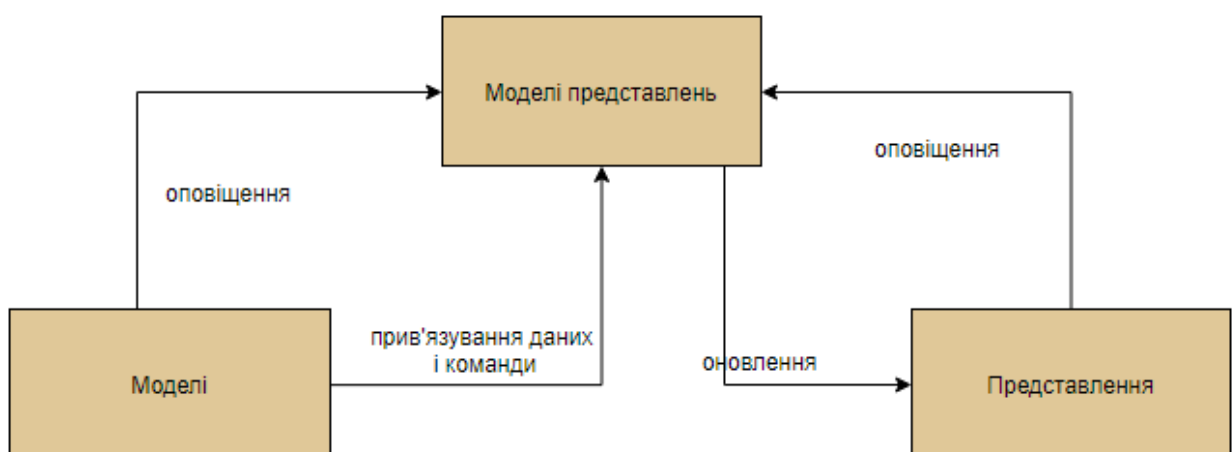


Рисунок 2.4 – Архітектурний шаблон MVVM

2.5 Висновки за розділом 2

Реалізацію програми було вирішено виконувати за допомогою мови програмування C#, для чого було виконано порівняння альтернативних варіантів та створено таблицю порівняння. Виконано моделювання сценаріїв роботи авторизованого та неавторизованого користувачів з програмою, в результаті чого побудовано відповідні діаграми прецедентів. Виконано проєктування БД для забезпечення роботи з даними. Обрано архітектурний шаблон MVVM для створення програмного забезпечення короткострокової оренди житла.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Структура розробленого програмного забезпечення

Розроблене програмне забезпечення побудоване на основі використання шаблону MVVM.

Розроблені представлення застосунку містяться в папці View і включають:

- SearchView.xaml – файл, в якому міститься представлення пошуку варіантів оренди для неавторизованого користувача;
- AdvancedSearchView.xaml – файл, в якому міститься представлення розширеного пошуку варіантів оренди для неавторизованого користувача;
- AuthView.xaml – файл, в якому міститься представлення авторизації користувача;
- RegView.xaml – файл, в якому міститься представлення реєстрації користувача;
- AuthSearchView.xaml – файл, в якому міститься представлення пошуку варіантів оренди для авторизованого користувача;
- AddRentalView.xaml – файл, в якому міститься представлення створення нового оголошення;
- AccomodationView.xaml – файл, в якому міститься представлення перегляду житла;
- PriceView.xaml – файл, в якому міститься представлення визначення вартості проживання в апартаментах;
- AccessView.xaml – файл, в якому міститься представлення визначення доступності апартаментів на задані дати;
- UserView.xaml – файл, в якому міститься представлення перегляду інформації про орендаря;
- EditAccomodationView.xaml – файл, в якому міститься представлення редагування даних про апартаменти;

– BookingView.xaml – файл, в якому міститься представлення оформлення бронювання житла;

– AuthAdvSearchView.xaml – файл, в якому міститься представлення розширеного пошуку варіантів оренди для авторизованого користувача;

– ProfileView.xaml – файл, в якому міститься представлення редагування даних профілю користувача;

– ReviewsView.xaml – файл, в якому міститься представлення перегляду відгуків на користувача;

– MessagesView.xaml – файл, в якому міститься представлення комунікації між користувачами;

– AccomodationsView.xaml – файл, в якому міститься представлення керування власним житлом;

– BookingsView.xaml – файл, в якому міститься представлення перегляду власних бронювань;

– AccomBookingsView.xaml – файл, в якому міститься представлення керування бронюваннями житла;

– AddUserReviewView.xaml – файл, в якому міститься представлення створення відгуку власника житла на орендаря;

– AddAccomReviewView.xaml – файл, в якому міститься представлення створення відгуку орендаря на проживання;

– AccomReviewsView.xaml – файл, в якому міститься представлення перегляду відгуків про житло.

Для кожного представлення розроблено аналогічну модель представлення для підготовки необхідних даних. Усі моделі представлень містяться в папці ViewModel.

У папці Resources містяться зображення, шляхи до яких збережено в БД.

Результуюча структура програмного забезпечення представлена на рис. 3.1.

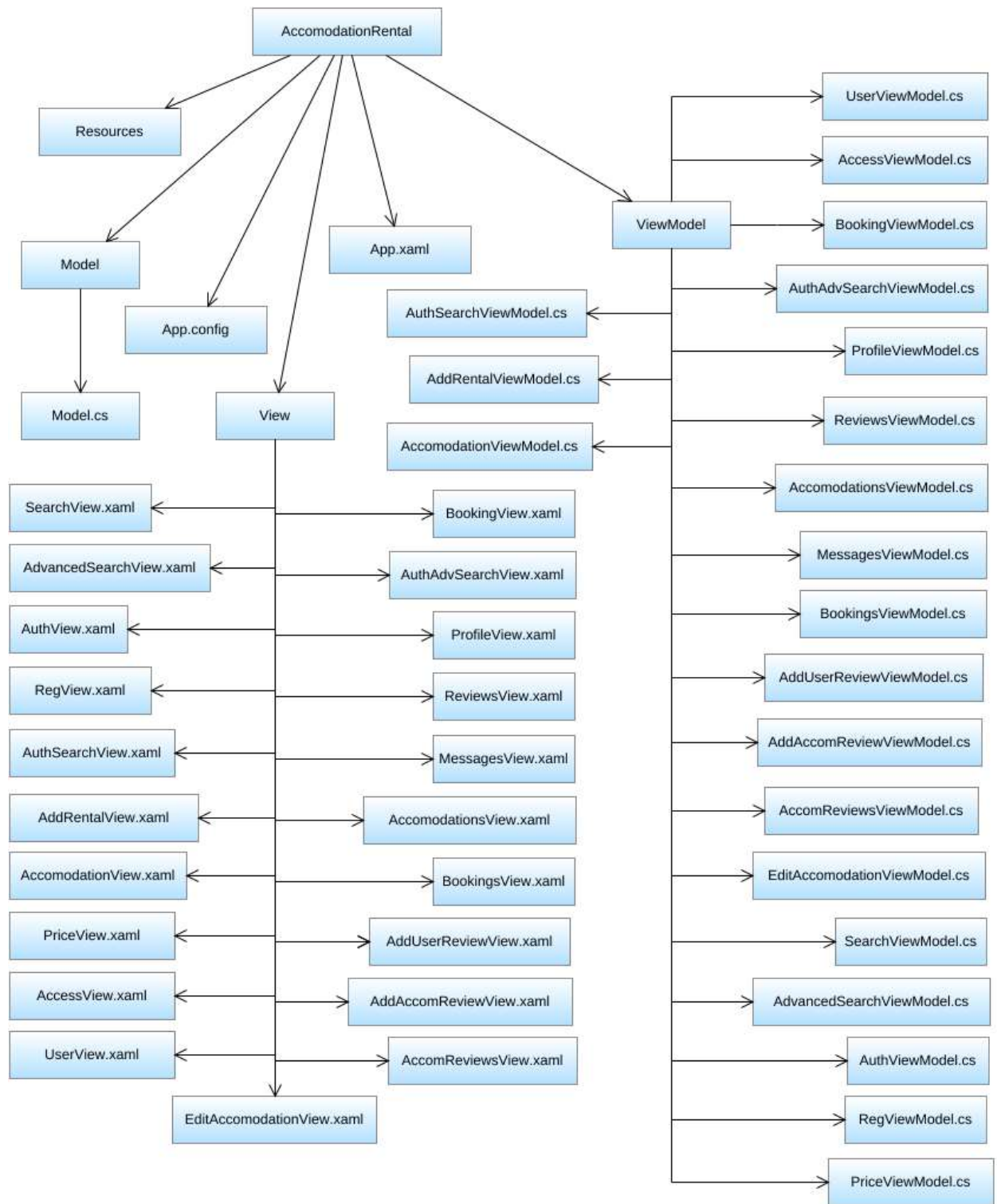


Рисунок 3.1 – Структура розробленого програмного забезпечення

У файлі App.xaml визначено загальну структуру проєкту і представлення, з якого починається робота програми.

У файлі App.config визначено налаштування проєкту, зокрема версію платформи .NET.

3.2 Опис логіки реалізації

Для реалізації логіки взаємодії з користувачем було створено набір представлень. Кожне представлення визначає, по-перше, розмітку графічного інтерфейсу користувача. В основі розмітки елемент сітки Grid, в чарунках якого розташовуються відповідні графічні елементи. Увесь інтерфейс кожного вікна застосунку побудовано таким чином.

Так для побудови меню вікна використано набір кнопок, які було упаковано в елемент StackPanel, для якого задано номер рядка, стовпця, а також протяжність в рядках, тобто кількість рядків, на яку розтягується даний елемент:

```
<StackPanel Grid.Row="0" Grid.Column="0" Grid.RowSpan="9">
  <Button      Margin="2,2,2,2"      Width="186"      Content="Пошук"
  Command="{Binding      AccomSearchCommand}"      Style="{StaticResource
  DerevButtonStyle}"      HorizontalAlignment="Center"      Height="50"
  FontFamily="Segoe UI"      FontSize="20"      VerticalAlignment="Center"
  Foreground="#FF4F4A4A"/>
```

Кнопки створено на основі створеного в програмі стилю DerevButtonStyle, який визначає необхідність скруглення кутів кнопок і встановлює товщину ліній.

При побудові представлень використовуються дані, які сформовані для представлень у моделях представлень. Для використання застосовується механізм прив'язування даних. Наприклад, формування рядка з вартістю бронювання відбувається наступним чином:

```
Content="{Binding      StringFormat=Вартість:      {0}      грн.,
Path=BookingPrice}"
```

Визначення елемента з прапорця для наявності або відсутності певної властивості виконується також на основі прив'язування до моделі представлення на основі відповідної властивості та з деактивацією можливості змінювання характеристики:

```
<CheckBox Margin="5,0" IsHitTestVisible="False" Focusable="False"
Grid.Row="8" IsChecked="{Binding WiFi}" Grid.Column="2"
TextBlock.FontFamily="Segoe UI" VerticalAlignment="Center"
TextBlock.FontSize="16" Content="Wi-Fi"></CheckBox>
```

Модель представлення побудована за наступною логікою.

Кожен клас має конструктор, в якому відбувається з'єднання з БД, витягання необхідних даних та початкове заповнення цими даними відповідних полів класу:

```
connec = new DataBaseConnector ();
connec.Cities.Load();
ObservableCollection<City> Cities = connec.Cities.Local.ToBindingList();
```

Наприклад, дані про всі міста зберігаються весь час в полі класу, а виділення міст, що належать до обраної країни реалізується під час виклику методу `get` і фільтрування цих даних на основі відповідності країні, яку обрав користувач:

```
get { return privCities.Where((country) =>
country.Title.Equals(AccomCountry.Title));
```

Для кожного поля, яке має бути відображено в представленні (як з можливістю редагування, так і без можливості редагування), створено

відповідну властивість, до якої, як було продемонстровано вище, виконується прив'язування в представленні:

```
public string Address
{
    get { return privaddress; }
    set
    {
        privaddress = value;
        OnPropertyChanged("Address");
    }
}
```

Усі моделі розроблені на основі структури даних, яка була розглянута під час створення БД. Відповідно під кожну сутність виділено відповідну модель, створену на основі відповідного класу. Також створено клас для комунікації з БД. Це клас `DataBaseConnector`, він відповідальний за встановлення з'єднання з БД та за створення і виконання відповідних запитів для БД.

3.3 Висновки за розділом 3

Розроблено набір представлень, моделей представлень та моделей, які потрібні для роботи програмного забезпечення короткострокової оренди житла. Для опису отриманих результатів під час реалізації цих складових було приведено структуру програмного забезпечення, описано логіку рішень під час їх реалізації.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення програми

Розроблений застосунок призначений для організації підтримки процесу оренди житла на короткий строк. Орендованим житлом можуть бути квартири, котеджі, таунхаузи. Винаймання декількох варіантів за однією одиницею не передбачено, тобто не можна здавати окремі кімнати, коли є можливість орендувати і кімнату, і квартиру цілком. Програма призначена для роботи з власниками житла.

Програма забезпечує роботу як власників житла, надаючи їм засоби виставлення свого нерухомого майна для оренди, керування процесом оренди, так і для орендарів, які можуть знаходити потрібні їм оголошення, оформлювати бронювання.

4.2 Умови виконання програми

Клієнтська частина програмного забезпечення має відповідати наступним вимогам:

- стаціонарний або мобільний (ноутбук) комп'ютер під керуванням операційної системи Windows, починаючи з версії 7;
- процесор не нижче Intel Celeron з тактовою частотою не менше 300 МГц;
- не менше 1 Гб оперативної пам'яті;
- 100 Мб вільного простору на жорсткому диску;
- монітор і відеокарта, які мають забезпечувати роботу з роздільною здатністю не менше 800x600.

4.3 Тестування сценаріїв роботи з програмою

Перед запуском програмного застосунку потрібно впевнитися, що доступ до сервера БД відкрито. Якщо ні, то буде виведено повідомлення про неможливість встановлення з'єднання з сервером БД, у такому випадку потрібно звернутися до адміністратора системи.

Якщо запуск відбувся успішно, то користувач побачить перед собою вікно, вигляд якого буде подібний до вигляду, представленого на рис. 4.1.

Робота неавторизованого користувача реалізується через вікно пошуку житла. У цьому вікні користувач має задати місто або місце, де він бажає виконати бронювання житла, потім визначити дати поїздки через поля вибору дати з випадаючим календарем, та задати кількість гостей, а далі – натиснути на кнопку «Пошук».

Після цього нижче буде оновлено перелік оголошень такими, що відповідають заданим критеріям пошуку. Якщо критерії не задано (як спочатку), то буде виведено повний перелік всіх доступних варіантів оренди у БД.

Кожне оголошення представляє собою рядок, що включає: зображення житла, стовпчик основних характеристик (назва житла, тип проживання та район), стовпчик додаткових характеристик (кількість кімнат, максимальну кількість гостей, які одночасно можуть проживати, кількість спалень, а також наявність кондиціонера та безпроводного інтернету, які за відсутності просто не відображаються, і інформація про вартість проживання). Вартість проживання вказується одразу в тій величині, що відповідає кінцевій сумі під час бронювання, що зокрема відрізняє створену програму від існуючих. Це дозволяє зручним чином переглядати швидко потрібні оголошення.

Для кожного оголошення доступні кнопки для виконання переходу до повного оголошення та бронювання житла. Кнопка бронювання у даному

випадку буде призводити до переходу до вікна авторизації, адже без цього виконати бронювання не вдасться.

Пошук житла

Місто

Львів

Прибуття

23.05.2021

Відбуття

25.05.2021

Гостей

2

Пошук

	Затишна квартира на площі Ринок квартира Галицький район	Кімнат: 3 Гостей: 4 Спальнь: 2 Wi-Fi Кондиціонер Вартість: 1600 грн.	Перейти Забронювати
	Зелені апартаменти квартира Шевченківський район	Кімнат: 3 Гостей: 4 Спальнь: 2 Wi-Fi Кондиціонер Вартість: 1300 грн.	Перейти Забронювати
	Будинок на Підзамче таунгауз Галицький район	Кімнат: 3 Гостей: 4 Спальнь: 2 Wi-Fi Кондиціонер Вартість: 1400 грн.	Перейти Забронювати
	Місце для всіх таунгауз Франківський район	Кімнат: 4 Гостей: 8 Спальнь: 3 Wi-Fi Кондиціонер Вартість: 1600 грн.	Перейти Забронювати
	На Вірменській квартира Галицький район	Кімнат: 2 Гостей: 3 Спальнь: 1 Wi-Fi Вартість: 1300 грн.	Перейти Забронювати

Розширений пошук

Авторизація

Рисунок 4.1 – Пошук житла неавторизованим користувачем

Окрім того внизу вікна доступні кнопки для виконання розширеного пошуку та для авторизації.

Варіант даного вікна для авторизованого користувача представлено на рис. 4.2.

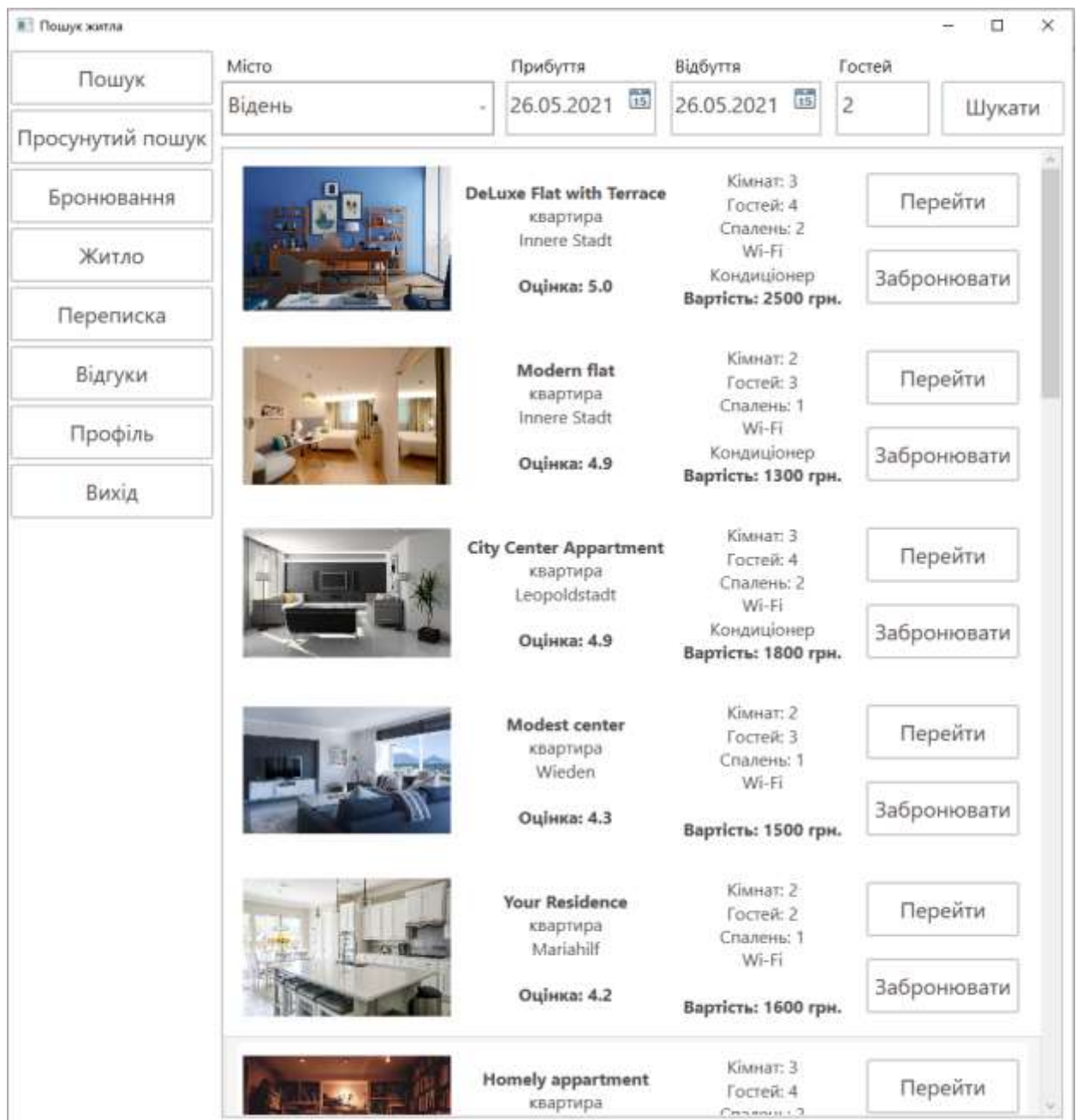


Рисунок 4.2 – Пошук житла авторизованим користувачем

Для виконання пошуку потрібно натиснути на кнопку «Шукати». Авторизований користувач може з цього вікна одразу перейти до вікна

бронювання, де потрібно виконати сплату.

Для авторизованого користувача окрім функціональності пошуку житла доступний цілий набір пунктів меню, представленими відповідними кнопками:

- «Пошук», що дозволяє виконати звичайний швидкий пошук житла;
- «Просунутий пошук», що дозволяє виконати розширений варіант пошуку житла для оренди з визначенням ширшої кількості параметрів (рис. 4.3);
- «Бронювання», що дозволяє переглянути перелік власних виконаних бронювань;
- «Житло», що дозволяє керувати власними пропозиціями житла для оренди, а також запитами на бронювання і бронюваннями;
- «Переписка», що дозволяє переглядати переписку між власниками та орендарями;
- «Відгуки», що дозволяє переглядати відгуки, які були залишені на користувача;
- «Профіль», що дозволяє керувати даними власного облікового запису;
- «Вихід», що дозволяє вийти з запису, під яким відбувається робота і перейти до неавторизованого режиму.

У розширеному пошуку можна окрім міста, інтервалу подорожі та кількості гостей вказати також:

- інтервал кількості кімнат житла;
- інтервал кількості поверхів;
- інтервал кількості спалень;
- інтервал кількості ліжок, якої потребує користувач;
- інтервал вартості за дану кількість днів (сумарної);
- інтервал оцінки від орендарів.

Продовжений пошук житла

Пошук

Продовжений пошук

Бронювання

Житло

Переписка

Відгуки

Профіль

Вихід

Місто

Варшава

Прибуття

26.05.2021

Відбуття

27.05.2021

Гостей

2

Шукати

Кімнат від

1

до

3

Спальень від

1

до

1

Вартість від

100

до

1700

☐ Кондиціонер

☐ Паркінг

☐ Пральна машина

☐ Підйомник

☐ Можна з тваринами

☐ Обігрів

Сортування

Оцінка

Поверхів від

1

до

1

Ліжок від

до

Оцінка від

4.1

до

5

☐ Телебачення

☒ Wi-Fi

☒ Ванна

☒ Кухня



A brand new flat
 квартира
 Srodmiescie
 Оцінка: 4.9

Кімнат: 2
 Гостей: 2
 Спальень: 1
 Wi-Fi
 Кондиціонер
Вартість: 1500 грн.

Перейти

Забронювати



Near Old Town
 квартира
 Mokotow
 Оцінка: 4.9

Кімнат: 3
 Гостей: 3
 Спальень: 1
 Wi-Fi

Вартість: 900 грн.

Перейти

Забронювати



Cozy Apartment
 квартира
 Srodmiescie
 Оцінка: 4.8

Кімнат: 1
 Гостей: 2
 Спальень: 1
 Wi-Fi

Вартість: 1300 грн.

Перейти

Забронювати

Рисунок 4.3 – Розширений пошук житла

Під час пошуку можна також вказати характеристики, які є обов'язковими в оголошенні і можуть включати наявність кондиціонеру, паркінгу, пральної машини, підйомника (ліфта), обігріву, доступу з тваринами, телебачення, безпроводного інтернету, кухні та ванної кімнати.

Результати пошуку будуть відсортовані за обраним варіантом: середньою оцінкою оголошення, загальною вартістю оренди або за алфавітом.

Якщо натиснути на кнопку «Перейти» для будь-якого варіанту житла, то відкриється окреме вікно перегляду оголошення (рис. 4.4).

Пропозиція для оренди

Пошук

Просунутий пошук

Бронювання

Житло

Переписка

Відгуки

Профіль

Вихід

Lake Paradise, 5.0



Місто: Монтре, Швейцарія

Женевське озеро

Котедж

Кімнат: 7

Гостей максимально: 17

Найдовша оренда: 7 діб

Поверхів: 1

Спальнь: 4

Ліжок: 4

- ☒ Кондиціонер
- ☒ Паркінг
- ☒ Телебачення
- ☒ Wi-Fi
- ☒ Пральна машина
- ☐ Підйомник
- ☒ Ванна
- ☒ Кухня
- ☒ Можна з тваринами
- ☒ Обігрів

Ми подбали про те, щоб рай на землі справді існував. Котедж на березі Женевського озера в Монтре - це місце для справжнього відпочинку. Вличезна територія, незабутній краєвид та найкращі сходи та заходи сонця. В котеджі є все, що тільки може знадобитись.



Патчет Джо
08.05.2021
Оцінка: 5.0

Написати

Це неймовірне місце, все продумано до дрібниць! Я вражений! Басейн з видом на озеро і це озеро, це щось!



Аркадіо Путц

Переписка

Прибуття

11.06.2021 15

Відбуття

12.06.2021 15

Гостей

2

Обчислити

Вартість: 10700 грн.

Забронювати

Рисунок 4.4 – Перегляд оголошення

У даному вікні міститься повна інформація про обраний варіант житла. Тільки у випадку підтвердженого бронювання в цьому вікні також відображається точна адреса проживання. У випадку попереднього огляду відображається тільки район.

Для місця проживання відображається набір фотографій у правому верхньому куті, який можна гортати, обирати потрібну фотографію, і вона буде відображатися у великому форматі посередині.

У верхньому рядку міститься назва апартаментів та середня виставлена порендарями оцінка.

Під зображенням вказано місто, в якому знаходиться житло, а ще нижче – район або повна адреса. Після цього в лівому стовпчику інформація доповнена типом приміщення, кількістю кімнат, інформацією про максимальну кількість гостей та найтриваліше можливу оренду, кількість поверхів, спалень і ліжок. У правому стовпчику міститься інформація про наявність кондиціонера, паркінгу, телебачення, безпроводного інтернету, пральної машини, підйомника, ванної кімнати, кухні, можливості проживання з тваринами і наявності обігріву. Ще нижче виокремлено опис від господаря.

У окремому переліку в самому низу вікна розташовано відгуки орендарів. Окрім можливості прочитати відгук можна натиснути на кнопку «Написати» і зв'язатися з користувачем, щоб отримати додаткову інформацію, якщо він забажає відповісти. Це є особливістю застосунку, функцією, яка відсутня в інших подібних аналогах.

У правій частині вікна містяться параметри бронювання: можна змінити дати прибуття і відбуття, кількість гостей і обчислити вартість проживання.

За допомогою кнопки «Забронювати» можна перейти до оформлення бронювання і сплати через PayPal, там же потрібно буде вказати кількість дітей.

Нижче також інформацію про господаря і є можливість написати йому повідомлення.

Вікно «Бронювання» (рис. 4.5) дозволяє користувачу переглянути бронювання, які він вже виконав. Усі бронювання відсортовані за датою реалізації. Бронювання, за якими ще не було виконано проживання, дозволяють зафіксувати користувачу факт заїзду в житло за допомогою кнопки «Проживання». Тоді після цього власник зможе зафіксувати факт виїзду орендаря з житла. Успішне виконання цих двох операцій призведе до того, що сума коштів буде переведена на рахунок власника житла.

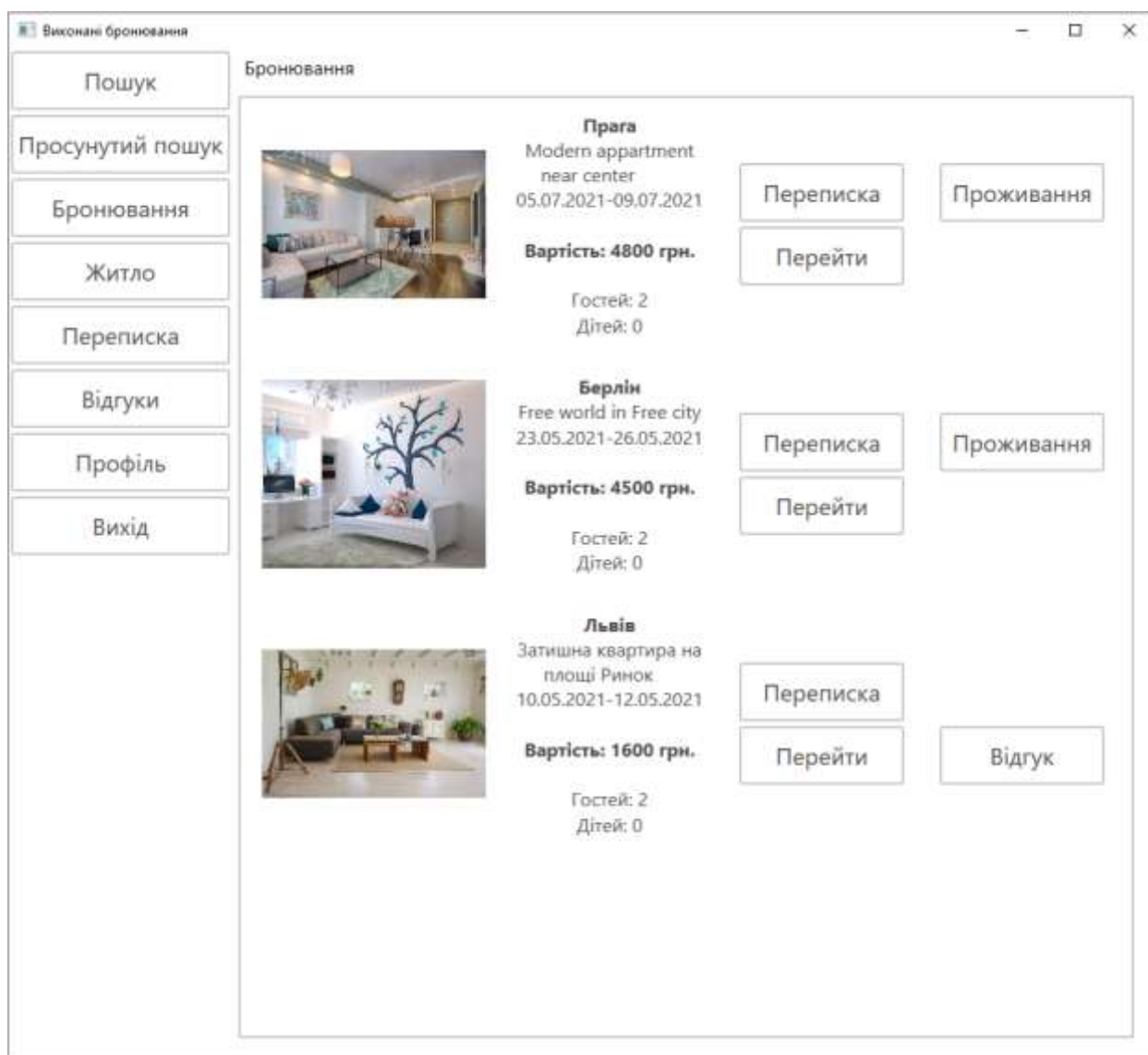


Рисунок 4.5 – Робота з виконаними користувачем бронюваннями

Для бронювань, які вже було завершено, можна натиснути на кнопку «Відгук» і залишити відгук, обравши оцінку від 1 до 5, а також додавши текстову рецензію з описом деталей.

Для кожного бронювання можна переглянути деталі про саме житло за допомогою кнопки «Перейти», за натискання відкриється вікно перегляду деталей житла. Також доступна кнопка «Переписка», яка дозволяє переглянути переписку з власником і відправити за необхідності йому нове повідомлення.

Вікно «Житло» надає доступ до списку власних пропозицій житла (рис. 4.6).

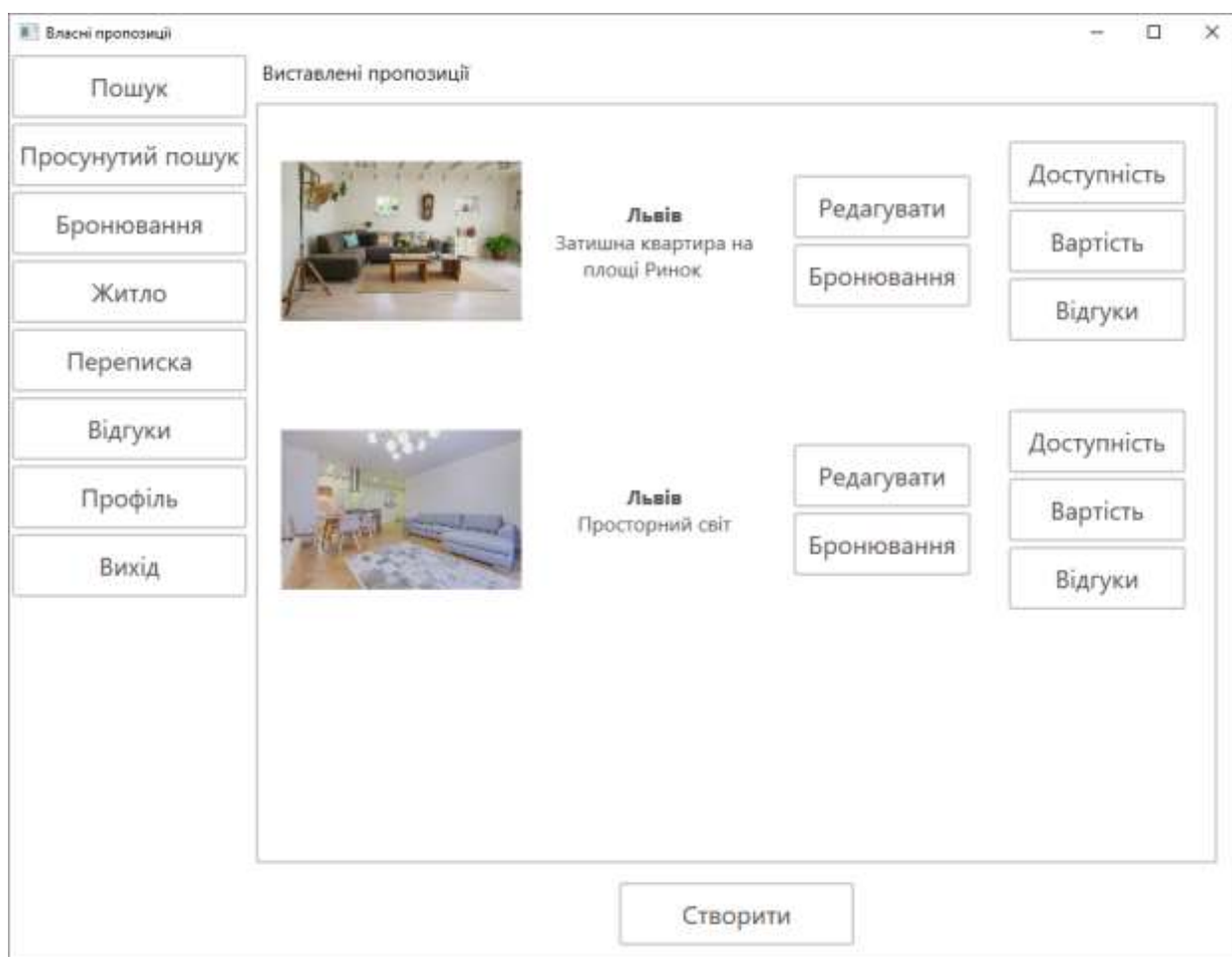


Рисунок 4.6 – Керування пропозиціями житла

Кожен варіант проживання характеризується фотографією, назвою міста, де воно розташовано, назвою апартаментів, а також засобами

керування.

Засоби керування включають кнопки:

– «Редагувати» для внесення змін до оголошення, яке бачать потенційні орендарі;

– «Бронювання» для перегляду всіх запитів на бронювання та виконаних бронювань за даною пропозицією, тобто звідси можна переглядати, хто і коли буде проживати у житлі власника;

– «Доступність» для внесення змін про доступність апартаментів за заданими датами;

– «Вартість» для внесення змін у вартість проживання у апартаментах заданої кількості людей на заданий проміжок дат;

– «Відгуки» для перегляду всіх відгуків за даним житлом.

При натискання на кнопку «Бронювання» відкривається окреме вікно, де виведено перелік бронювань за обраним житлом. Усі вони поділені на три частини (рис. 4.7).

Перша частина – запити на бронювання. Даний розділ містить запити, які ще не були підтверджені власником. Усі вони відсортовані за збільшенням дати заїзду. Для кожного запиту міститься фотографія орендаря, його ім'я, місце проживання, кількість гостей та дітей (через знак плюс вказується додатково кількість дітей), яка планує відвідати апартаменти, та середня оцінка орендаря за відгуками господарів. Якщо натиснути на кнопку «Користувач», то можна відкрити окреме вікно з переглядом даних про орендаря, що включають перелік оцінок та рецензій на його проживання. Для кожного запиту можна натиснути на кнопку «Підтвердити» або «Скасувати», дозволивши проживання орендаря або скасувавши його.

Після підтвердження бронювання воно переміщується в другий розділ вікна – очікувані бронювання. Сюди належать бронювання оренди, які ще не виконані, але вже підтверджені.

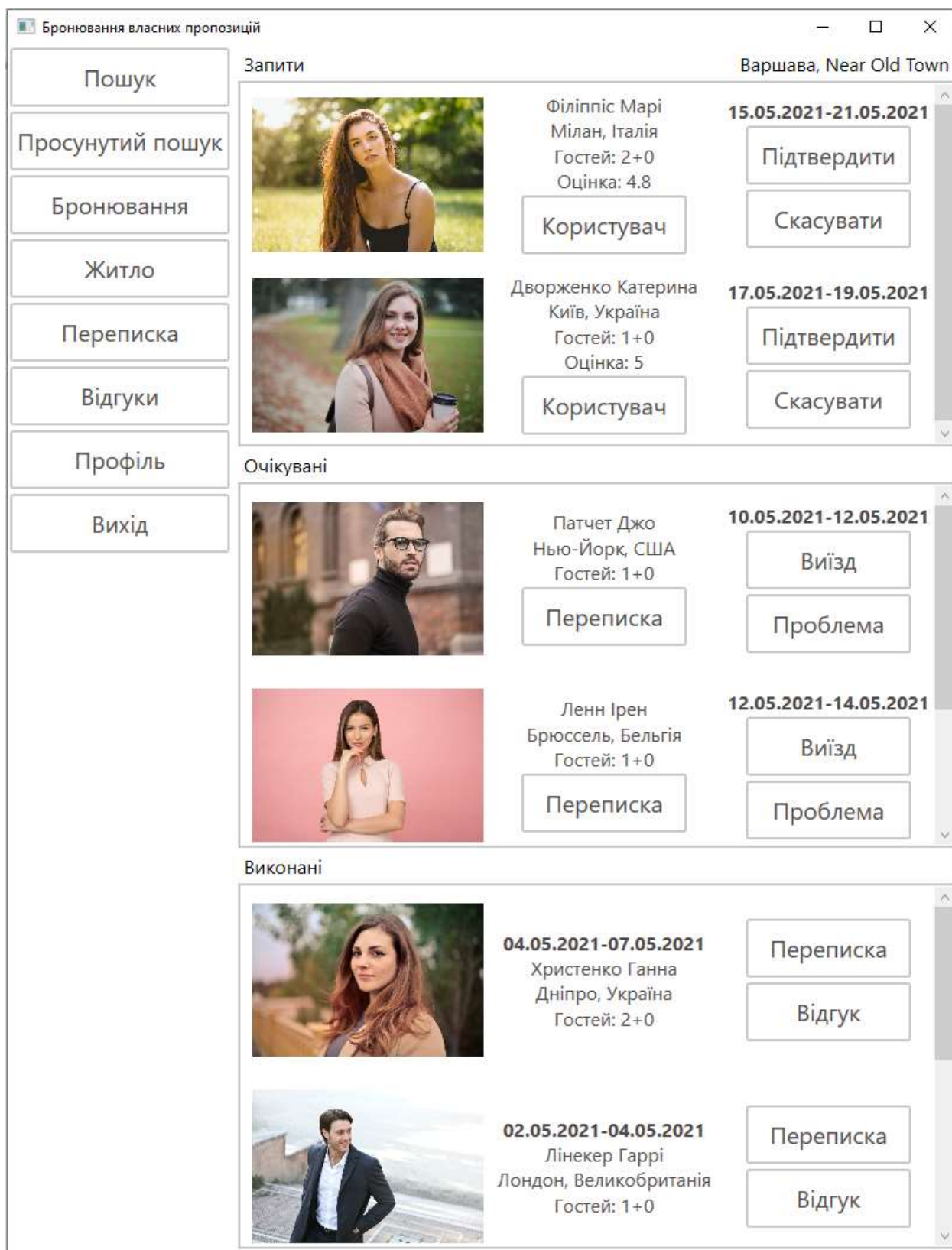


Рисунок 4.7 – Керування бронюваннями

Для кожного такого бронювання можна перейти до переписки з орендарем шляхом натискання на кнопку «Переписка». Тут же можна

підтвердити виїзд орендаря шляхом натискання на кнопку «Виїзд», а також повідомити про проблему шляхом натискання на однойменну кнопку, якщо орендар вчасно не виїхав або трапились будь-які серйозні проблеми, пов'язані з участю даного орендаря.

Третя частина – виконані бронювання. Це бронювання, проживання за якими закінчено. Сюди вони потрапляють після підтвердження виїзду власником. Для кожного такого бронювання доступні кнопки «Переписка», що дозволяють за необхідності направити людині повідомлення, та «Відгук» для того щоб залишити відгук про орендаря, що і є основним призначенням даної частини вікна.

4.4 Висновки за розділом 4

У даному розділі розглянуто основні сценарії роботи неавторизованого користувача з програмою, власника житла та орендаря. Розглянуто основні засоби доступні для їх роботи.

ВИСНОВКИ

Дипломну кваліфікаційну роботу бакалавра присвячено проблемі розробки програмного забезпечення короткострокової оренди житла. Для цього розглянуто особливості короткострокової оренди порівняно з довгостроковою, особливості організації програмного забезпечення для підтримки даного процесу, виконано аналіз програмних аналогів, що включають Airbnb, VrBO, Flipkey.

Реалізацію програми було вирішено виконувати за допомогою мови програмування C#, для чого було виконано порівняння альтернативних варіантів та створено таблицю порівняння. Виконано моделювання сценаріїв роботи авторизованого та неавторизованого користувачів з програмою, в результаті чого побудовано відповідні діаграми прецедентів. Виконано проєктування БД для забезпечення роботи з даними. Обрано архітектурний шаблон MVVM для створення програмного забезпечення короткострокової оренди житла.

Розроблено програму і описано структуру програмного забезпечення з представленням структурної схеми. Створений програмний застосунок дозволив реалізувати програмну платформу, де власники житла можуть взаємодіяти з орендарем без участі посередників. Для створення довіри між учасниками процесу передбачено систему відгуків на основі оцінок та текстових рецензій, які створюються як для орендарів, так і для самого житла. Розроблене програмне забезпечення дозволяє створювати заявки на оренду житла, власникам підтверджувати їх або скасовувати, підтверджувати орендарю заселення, а власнику – виселення, створювати відгуки про власний досвід, виконувати пошук нових варіант бронювання, переглядати власні бронювання та бронювання, оформлені на пропозиції оренди житла власника.

Усі завдання, які було визначено у дипломній кваліфікаційній роботі бакалавра, виконано у повному обсязі.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ashworth, J. Long-Term vs. Short-Term Rentals [Electronic resource]: What You Need to Know / J. Ashworth. – Access mode : <https://www.lodgify.com/blog/long-short-term-vacation-rentals/>
2. Barnes, M. 5 Principles Every Short-Term Property Manager Should Live By [Electronic resource] / M. Barnes. – Access mode : <https://www.guesty.com/blog/5-principles-every-short-term-property-manager-should-live-by/>
3. Fors, A. Top 30 Vacation Rental Software to Manage Properties in 2021 [Electronic resource] / A. Fors. – Access mode : <https://hoteltechreport.com/news/vacation-rental-software-guide>
4. Оренда для відпочинку, помешкання, готелі, враження та інше – Airbnb [Електрон. ресурс]. – Режим доступу : <https://www.airbnb.com.ua/>
5. Review Of AirBnB Mobile App [Electronic resource]. – Access mode : <https://robusttechhouse.com/review-of-airbnb-mobile-app/>
6. Airbnb Host Reviews & How To Automate Them | Smartbnb [Electronic resource]. – Access mode : <https://www.smartbnb.io/airbnb-host-reviews-how-to-automate-them/>
7. Vrbo | Book your vacation rentals: beach houses, cabins, condos & more [Electronic resource]. – Access mode : <https://www.vrbo.com/>
8. The difference between Airbnb vs. Vrbo, explained [Electronic resource]. – Access mode : <https://mashable.com/article/airbnb-vs-vrbo-trip-booking/>
9. Full Review of VRBO 2021 - Is it legit? Big Comparison Step-by-Step [Electronic resource]. – Access mode : <https://one-week-in.com/vrbo-reviews-vacation-rental-alternatives-airbnb/>
10. Vacation Rentals – Beach Houses, Cabins, Condos, Cottages, Vacation Homes & Villas | FlipKey [Electronic resource]. – Access mode : <https://www.flipkey.com/>

11. Seemann, K. The Ultimate Guide to Flipkey Vacation Rentals in 2020 [Electronic resource] / K. Seemann. – Access mode : <https://upgradedpoints.com/flipkey-vacation-rentals-guide>
12. Фленов, М. Библия C# [Текст] / М. Фленов. – СПб. : БХВ-Петербург, 2020. – 512 с.
13. Price, M. J. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development [Text] : Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and ML.NET using Visual Studio Code / M. J. Price. – Birmingham : Packt Publishing, 2019. – 818 p.
14. Gaddis, T. Starting Out with Java: From Control Structures through Objects (What's New in Computer Science) [Text] / T. Gaddis. – London : Pearson, 2018. – 1168 p.
15. C# .NET vs Java – Which programs are fastest? | Computer Language Benchmarks Game [Electronic resource]. – Access mode : <https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/csharp.html>

ДОДАТОК А
Технічне завдання

Вступ

Програма повинна забезпечувати роботу як власників житла, надаючи їм засоби виставлення свого нерухомого майна для оренди, керування процесом оренди, так і орендарів, які можуть знаходити потрібні їм оголошення, оформлювати бронювання.

A.1 Підстави для розробки

Розробка має бути виконана в межах завдання на дипломну кваліфікаційну роботу, яке було поставлено за темою «Програмне забезпечення короткострокової оренди житла» у відповідності з наказом № 103 від 30 березня 2021 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

Програмний застосунок, що розробляється, призначений для організації підтримки процесу оренди житла на короткий строк. Орендованим житлом можуть бути квартири, котеджі, таунхаузи. Винаймання декількох варіантів за однією одиницею не має бути передбачено, тобто не можна здавати окремі кімнати, коли є можливість орендувати і кімнату, і квартиру цілком. Програма має бути призначена для роботи безпосередньо з власниками житла.

A.3 Основні вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Для реалізації визначеного призначення програма має надавати такі функціональні особливості:

- пошук житла для бронювання;
- перегляд оголошення про оренду житла;
- розрахунок вартості проживання за заданими параметрами;
- перегляд даних про власника житла;
- перегляд відгуків про житло;
- виставлення власного оголошення про бронювання житла;
- перегляд списку власних оголошень;
- визначення вартості бронювання житла на проміжок часу;
- визначення доступності житла на проміжок часу;
- бронювання житла;
- сплата за бронювання житла;
- відправлення повідомлень власнику житла;
- перегляд переписки з власником;
- створення відповідей на повідомлення орендаря;
- внесення відгуків про житло після завершення бронювання;
- підтвердження проживання власником;
- підтвердження виїзду орендарем;
- перегляд власних бронювань;
- підтвердження бронювання власником;
- скасування бронювання власником;
- скасування власного бронювання;
- перегляд бронювань власного житла;
- авторизація.

A.3.2 Вимоги до надійності

За результатами введених користувачем даних має здійснюватися перевірка того, чи було введено коректні дані для кожного поля (зокрема для кожного рядка має перевірятися, чи введене значення не є більшим за довжину вказану для відповідного поля в таблиці БД). Окрім того має

виконуватися перевірка підключення до сервера БД, перевірка успішності виконання операцій, пов'язаних з внесенням змін в БД.

А.3.3 Умови експлуатації

Для експлуатації програми особливих вмінь від користувачів окрім вміння користуватися комп'ютером не висувається.

Для забезпечення роботи БД на сервері необхідні вміння системного адміністратора для підтримки роботи сервера системи керування БД Microsoft SQL Server.

А.3.4 Вимоги до складу та параметрів технічних засобів

Вимоги до технічних засобів клієнтської сторони:

- стаціонарний або мобільний (ноутбук) комп'ютер під керуванням операційної системи Windows, починаючи з версії 7;
- процесор не нижче Intel Celeron з тактовою частотою не менше 300 МГц;
- не менше 1 Гб оперативної пам'яті;
- 100 Мб вільного простору на жорсткому диску;
- монітор і відеокарта, які мають забезпечувати роботу з роздільною здатністю не менше 800х600.

А.4 Порядок контролю та приймання

Принципи контролю виконання роботи визначаються календарним планом, розташованим у завданні на дипломну кваліфікаційну роботу, де вказано назви етапів, строки їх виконання та результат.

ДОДАТОК Б
Текст програми

Б.1 Текст файла AccomodationView.cs

```

<Window x:Class="AccomodationRental.View.AccomodationView "
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
      xmlns:local="clr-namespace:AccomodationRental.View"
      mc:Ignorable="d"
      Title="Пропозиція для оренди" Height="1110" Width="820">

    <Window.Resources>
        <Style x:Key="DerevButtonStyle" TargetType="Button" >
            <Setter Property="TextBlock.TextAlignment"
Value="Center" />
            <Setter Property="Template">
                <Setter.Value>
                    <ControlTemplate TargetType="Button" >
                        <Border CornerRadius="2,2,2,2"
BorderBrush="{DynamicResource {x:Static
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
BorderThickness="2,2,2,2" >
                            <ContentPresenter
x:Name="DerevButtonPresenter" ContentTemplate="{TemplateBinding
ContentTemplate}" Content="{TemplateBinding Content}"
HorizontalAlignment="{TemplateBinding HorizontalContentAlignment}"
Margin="{TemplateBinding Padding}" VerticalAlignment="{TemplateBinding
VerticalContentAlignment}"/>
                        </Border>
                    </ControlTemplate>
                </Setter.Value>
            </Setter>
        </Style>

        <SolidColorBrush x:Key="DerevElem" Color="White"/>
        <Style x:Key="lbDerev" TargetType="TextBlock">
            <Setter Property="Control.FontSize" Value="14"/>
        </Style>

```

```

</Window.Resources>
<Grid>

    <Grid.RowDefinitions>
        <RowDefinition Height="0"/>
        <RowDefinition Height="40"/>
        <RowDefinition Height="225"/>
        <RowDefinition Height="40"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="70"/>
        <RowDefinition Height="130"/>
        <RowDefinition Height="120"/>
        <RowDefinition Height="30"/>
        <RowDefinition Height="60"/>
        <RowDefinition Height="3"/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition Width="190"/>
        <ColumnDefinition Width="210"/>
        <ColumnDefinition Width="210"/>
        <ColumnDefinition Width="170"/>
    </Grid.ColumnDefinitions>

    <StackPanel Grid.Row="0" Grid.Column="0"
Grid.RowSpan="9">
        <Button Margin="2,2,2,2" Width="186" Content="Пошук"
Command="{Binding AccomSearchCommand}" Style="{StaticResource
DerevButtonStyle}" HorizontalAlignment="Center" Height="50"
FontFamily="Segoe UI" FontSize="20" VerticalAlignment="Center"
Foreground="#FF4F4A4A"/>
        <Button Margin="2,2,2,2" Width="186"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20" Content="Просунутий

```

```

пошук" VerticalAlignment="Center" Command="{Binding
AdvAccomSearchCommand}" Foreground="#FF4F4A4A" />
        <Button Margin="2,2,2,2" Width="186"
Content="Бронювання" Command="{Binding AccomBookCommand}"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20"
VerticalAlignment="Center" Foreground="#FF4F4A4A"/>
        <Button Margin="2,2,2,2" Width="186" Content="Житло"
Command="{Binding AccomManageCommand}" Style="{StaticResource
DerevButtonStyle}" HorizontalAlignment="Center" Height="50"
FontFamily="Segoe UI" FontSize="20" VerticalAlignment="Center"
Foreground="#FF4F4A4A"/>
        <Button Margin="2,2,2,2" Width="186"
Content="Пепениска" Command="{Binding PeopleChatCommand}"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20"
VerticalAlignment="Center" Foreground="#FF4F4A4A"/>
        <Button Margin="2,2,2,2" Width="186"
Content="Відгуки" Command="{Binding AccomReviewsCommand}"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20"
VerticalAlignment="Center" Foreground="#FF4F4A4A"/>
        <Button Margin="2,2,2,2" Width="186"
Content="Профіль" Command="{Binding UserProfileCommand}"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20"
VerticalAlignment="Center" Foreground="#FF4F4A4A"/>
        <Button Margin="2,2,2,2" Width="186" Content="Вихід"
Command="{Binding SignOutCommand}" Style="{StaticResource
DerevButtonStyle}" HorizontalAlignment="Center" Height="50"
FontFamily="Segoe UI" FontSize="20" VerticalAlignment="Center"
Foreground="#FF4F4A4A"/>
    </StackPanel>

    <Label FontWeight="Bold" Grid.ColumnSpan="2"
Margin="5,0" Grid.Row="1" Grid.Column="1" TextBlock.FontFamily="Segoe
UI" HorizontalAlignment="Center" VerticalAlignment="Center"
TextBlock.FontSize="24" Content="Binding AccomNameWithPoints"></Label>

    <ListBox Width="120"
ScrollViewer.VerticalScrollBarVisibility="Hidden"
ScrollViewer.HorizontalScrollBarVisibility="Hidden"
IsSynchronizedWithCurrentItem="True" Margin="5,0" Height="220"

```

```

Grid.Row="2" Grid.Column="3" BorderBrush="{DynamicResource {x:Static
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
BorderThickness="2,2,2,2" Foreground="#FF4F4A4A"
x:Name="ConsumerOrders" ItemsSource="{Binding Photos}"
VirtualizingPanel.ScrollUnit="Pixel"
HorizontalAlignment="Stretch" FontFamily="Segoe UI"
FontSize="14" SelectedItem="{Binding Path=SelectedAccomPhoto,
Mode=TwoWay}>
        <ListBox.ItemTemplate>
            <DataTemplate>
                <Border Background="{StaticResource
DerevElem}" BorderThickness="2,2,2,2">
                    <Grid Height="75" ClipToBounds="True">
                        <Grid.ColumnDefinitions>
                            <ColumnDefinition Width="100" />
                        </Grid.ColumnDefinitions>
                        <StackPanel Grid.Row="0"
VerticalAlignment="Center" Orientation="Horizontal" Margin="2">
                            <Image Source="{Binding Route}"
VerticalAlignment="Center" Visibility="Visible" />
                        </StackPanel>
                    </Grid>
                </Border>
            </DataTemplate>
        </ListBox.ItemTemplate>
    </ListBox>

    <Image Grid.ColumnSpan="2" Grid.Row="2" Grid.Column="1"
Source="{Binding Path= SelectedAccomPhoto.Route}"
VerticalAlignment="Center" Visibility="Visible" />

    <Label Grid.ColumnSpan="2" Margin="5,0" Grid.Row="3"
Grid.Column="1" TextBlock.FontFamily="Segoe UI"
HorizontalAlignment="Center" VerticalAlignment="Center"
TextBlock.FontSize="20" Content="{Binding City}"></Label>

    <Label Margin="5,0" Grid.Row="4" Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
Region}"></Label>

```

```
<Label      Margin="5,0" Grid.Row="5"  Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
Acc_TypeStr}"></Label>
```

```
<Label      Margin="5,0" Grid.Row="6"  Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
StringFormat=Кімнат: {0}, Path=Rooms}"></Label>
```

```
<Label      Margin="5,0" Grid.Row="7"  Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
StringFormat=Гостей максимально: {0}, Path=Guests}"></Label>
```

```
<Label      Margin="5,0" Grid.Row="8"  Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
StringFormat=Найдовша оренда: {0} діб, Path=MaxStay}"></Label>
```

```
<Label      Margin="5,0" Grid.Row="9"  Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
StringFormat=Поверхів: {0}, Path=Floors}"></Label>
```

```
<Label      Margin="5,0" Grid.Row="10" Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
StringFormat=Спальень: {0}, Path=MaxStay Bedrooms}"></Label>
```

```
<Label      Margin="5,0" Grid.Row="11" Grid.Column="1"
TextBlock.FontFamily="Segoe UI" HorizontalAlignment="Left"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="{Binding
StringFormat=Ліжок: {0}, Path=MaxStay Beds}"></Label>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="5" IsChecked="{Binding Conditioner}"
Grid.Column="2" HorizontalAlignment="Left" TextBlock.FontFamily="Segoe
UI" VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Кондиціонер"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="6" IsChecked="{Binding Parking}"
```

```
Grid.Column="2" TextBlock.FontFamily="Segoe UI"
VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Паркінг"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="7" IsChecked="{Binding TV}"
Grid.Column="2" HorizontalAlignment="Left" TextBlock.FontFamily="Segoe
UI" VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Телебачення"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="8" IsChecked="{Binding WiFi}"
Grid.Column="2" TextBlock.FontFamily="Segoe UI"
VerticalAlignment="Center" TextBlock.FontSize="16" Content="Wi-
Fi"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="9" IsChecked="{Binding Washer}"
Grid.Column="2" HorizontalAlignment="Left" TextBlock.FontFamily="Segoe
UI" VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Пральна машина"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="10" IsChecked="{Binding Elevator}"
Grid.Column="2" TextBlock.FontFamily="Segoe UI"
VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Підйомник"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="11" IsChecked="{Binding Bathroom}"
Grid.Column="2" HorizontalAlignment="Left" TextBlock.FontFamily="Segoe
UI" VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Ванна"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="12" IsChecked="{Binding Kitchen}"
Grid.Column="2" TextBlock.FontFamily="Segoe UI"
VerticalAlignment="Center" TextBlock.FontSize="16"
Content="Кухня"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"
Focusable="False" Grid.Row="13" IsChecked="{Binding Pets}"
Grid.Column="2" HorizontalAlignment="Left" TextBlock.FontFamily="Segoe
```

```
UI" VerticalAlignment="Center" TextBlock.FontSize="16" Content="Можна  
з тваринами"></CheckBox>
```

```
<CheckBox Margin="5,0" IsHitTestVisible="False"  
Focusable="False" Grid.Row="14" IsChecked="{Binding Heating}"  
Grid.Column="2" TextBlock.FontFamily="Segoe UI"  
VerticalAlignment="Center" TextBlock.FontSize="16"  
Content="06irpib"></CheckBox>
```

```
<Label Margin="5,0" Grid.Row="3" Grid.Column="3"  
TextBlock.FontFamily="Segoe UI" VerticalAlignment="Center"  
TextBlock.FontSize="16" Content="Прибуття"></Label>
```

```
<DatePicker Grid.RowSpan="2" Margin="0,0, 2, 0"  
Grid.Row="4" Padding="0,5" Height="50" Grid.Column="3" Width="138"  
SelectedDateFormat="Short" SelectedDate="{Binding CurrDate}"  
FontSize="20"  
DisplayDateStart="{Binding CurrMonthFDate}"  
DisplayDateEnd="{Binding CurrMonthEDate}"  
BorderBrush="{DynamicResource {x:Static  
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource  
{x:Static SystemColors.ControlLightLightBrushKey}}"  
BorderThickness="2,2,2,2" Foreground="#FF4F4A4A"  
FirstDayOfWeek="Monday"  
HorizontalAlignment="Center" VerticalAlignment="Center" Language="uk-  
UA" />
```

```
<Label Margin="5,0" Grid.Row="6" Grid.Column="3"  
TextBlock.FontFamily="Segoe UI" VerticalAlignment="Center"  
TextBlock.FontSize="16" Content="Відбуття"></Label>
```

```
<DatePicker Grid.RowSpan="2" Margin="2,0" Grid.Row="7"  
Padding="0,5" Height="50" Grid.Column="3" Width="138"  
SelectedDateFormat="Short" SelectedDate="{Binding CurrPDate}"  
FontSize="20"  
DisplayDateStart="{Binding CurrMonthFDate}"  
DisplayDateEnd="{Binding CurrMonthEDate}"  
BorderBrush="{DynamicResource {x:Static  
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource  
{x:Static SystemColors.ControlLightLightBrushKey}}"  
BorderThickness="2,2,2,2" Foreground="#FF4F4A4A"
```

```

        FirstDayOfWeek="Monday"
HorizontalAlignment="Center" VerticalAlignment="Center" Language="uk-
UA" />

```

```

        <Label Margin="5,0" Grid.Row="9" Grid.Column="3"
TextBlock.FontFamily="Segoe UI" VerticalAlignment="Center"
TextBlock.FontSize="16" Content="Гостей"></Label>

```

```

        <TextBox Grid.RowSpan="2" Grid.Row="10" Grid.Column="3"
Padding="2,5" Width="138" MinHeight="50" MaxHeight="50"
FontFamily="Segoe UI" FontSize="20" HorizontalAlignment="Center"
BorderBrush="{DynamicResource {x:Static
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
BorderThickness="2,2,2,2" Foreground="#FF4F4A4A" />

```

```

        <Button Grid.Row="12" Grid.Column="3" Grid.RowSpan="2"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20" Content="Обчислити"
VerticalAlignment="Center" Command="{AccPriceCalc}"
Foreground="#FF4F4A4A" Padding="17,0,17,0"/>

```

```

        <Label Grid.ColumnSpan="1" Margin="5,0" Grid.Row="14"
Grid.Column="3" TextBlock.FontFamily="Segoe UI"
HorizontalAlignment="Left" VerticalAlignment="Center"
TextBlock.FontSize="16" Content="{Binding StringFormat=Вартість: {0}
грн., Path=BookingPrice}"></Label>

```

```

        <TextBox Grid.RowSpan="2" IsReadOnly="True"
TextWrapping="Wrap" Text="{Binding Description}" Grid.ColumnSpan="2"
Grid.Row="15" Grid.Column="1" Padding="2,2" Width="410"
MinHeight="195" MaxHeight="195" FontFamily="Segoe UI" FontSize="20"
HorizontalAlignment="Center" BorderBrush="{DynamicResource {x:Static
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
BorderThickness="2,2,2,2" Foreground="#FF4F4A4A" />

```

```

        <Button Grid.Row="15" VerticalAlignment="Top"
Margin="0,10" Grid.Column="3" Grid.RowSpan="2" Style="{StaticResource
DerevButtonStyle}" HorizontalAlignment="Center" Height="50"
FontFamily="Segoe UI" FontSize="20" Content="Забронювати"
Command="{AccomBookCommand}" Foreground="#FF4F4A4A"
Padding="17,0,17,0"/>

```

```

        <ListBox Grid.ColumnSpan="2" Grid.RowSpan="3"
Margin="5,2" Height="205" Grid.Row="17" Grid.Column="1"
BorderBrush="{DynamicResource {x:Static
SystemColors.ScrollBarBrushKey}}" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
BorderThickness="2,2,2,2" Foreground="#FF4F4A4A"
x:Name="AccomodationReviews" ItemsSource="{Binding Reviews}"
VirtualizingPanel.ScrollUnit="Pixel"
HorizontalContentAlignment="Stretch"
ScrollViewer.HorizontalScrollBarVisibility="Disabled"
FontFamily="Segoe UI" FontSize="14">
            <ListBox.ItemTemplate>
                <DataTemplate>
                    <Border Background="{StaticResource
DerevElem}" BorderThickness="2,2,2,2">
                        <Grid Height="190" ClipToBounds="True">
                            <Grid.ColumnDefinitions>
                                <ColumnDefinition Width="130" />
                                <ColumnDefinition Width="100" />
                                <ColumnDefinition Width="150"
/>>
                            </Grid.ColumnDefinitions>
                            <StackPanel Grid.Column="0"
VerticalAlignment="Center" Margin="2">
                                <Image Source="{Binding
UserPhoto}" VerticalAlignment="Center" Visibility="Visible" />
                                <Button Width="127"
Style="{StaticResource DerevButtonStyle}" HorizontalAlignment="Center"
Height="50" FontFamily="Segoe UI" FontSize="20" Content="Написать"
Command="{AddMessagesCommand}" Foreground="#FF4F4A4A"/>
                            </StackPanel>
                            <StackPanel Grid.Column="1"
VerticalAlignment="Center" Margin="2">
                                <TextBlock Text="{Binding
UserSurnameName}" FontWeight="Bold" HorizontalAlignment="Center"
FontFamily="Segoe UI" FontSize="16" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
Foreground="#FF4F4A4A"/>
                                <TextBlock Text="{Binding
Fixed}" HorizontalAlignment="Center" FontFamily="Segoe UI"
FontSize="16" Background="{DynamicResource {x:Static
SystemColors.ControlLightLightBrushKey}}" Foreground="#FF4F4A4A"/>

```

```

        <TextBlock Text="{Binding
PointViews}" HorizontalAlignment="Center" FontFamily="Segoe UI"
FontSize="16" Background="{DynamicResource {x:Static
SystemColors.ControlLightLightBrushKey}}" Foreground="#FF4F4A4A"/>
    </StackPanel>
    <StackPanel Grid.Column="2"
HorizontalAlignment="Center" VerticalAlignment="Center" Margin="2">
        <TextBlock TextWrapping="Wrap"
Text="{Binding Message}" HorizontalAlignment="Center"
FontFamily="Segoe UI" FontSize="16" Background="{DynamicResource
{x:Static SystemColors.ControlLightLightBrushKey}}"
Foreground="#FF4F4A4A"/>
    </StackPanel>
</Grid>
</Border>
</DataTemplate>
</ListBox.ItemTemplate>
</ListBox>

<Image Grid.RowSpan="1" Grid.Row="17" Grid.Column="3"
Source="{Binding OwnerPhoto}" VerticalAlignment="Center"
Visibility="Visible" />

<Label Grid.ColumnSpan="1" Margin="5,0" Grid.Row="18"
Grid.Column="3" TextBlock.FontFamily="Segoe UI"
HorizontalAlignment="Center" VerticalAlignment="Top"
TextBlock.FontSize="16" Content="{Binding OwnerSurnameName}"></Label>

<Button Grid.Row="19" VerticalAlignment="Top"
Margin="0,5" Grid.Column="3" Style="{StaticResource
DerevButtonStyle}" HorizontalAlignment="Center" Height="50"
FontFamily="Segoe UI" FontSize="20" Content="Переписка"
Command="{OwnerMessageCommand}" Foreground="#FF4F4A4A"
Padding="17,0,17,0"/>

</Grid>
</Window>

```

Б.2 Текст файла AddRentalViewModel.cs

```

using System;
using System.Collections.Generic;

```

```

using System.Collections.ObjectModel;
using System.IO;
using System.Linq;
using System.Net.Sockets;
using System.Net;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using AccomodationRental.Models;
using AccomodationRental.Views;

namespace AccomodationRental.ViewModels
{
    public class AddRentalViewModel: INotifyPropertyChanged
    {

        DataBaseConnector connec;
        User Owner;
        private RelayCommand privadder;
        private RelayCommand privimageadder;

        public AddRentalViewModel (User owner)
        {
            Owner = owner;
            connec = new DataBaseConnector ();
            connec.Cities.Load();
            ObservableCollection<City> Cities =
connec.Cities.Local.ToBindingList();
            connec.Countries.Load();
            ObservableCollection<Country> Countries =
connec.Countries.Local.ToBindingList();
        }

        private ObservableCollection<City> privCities;
        public ObservableCollection<City> Cities
        {
            get { return privCities.Where((country) =>
country.Title.Equals(AccomCountry.Title));
            }
            set
            {
                privCities = value;
                OnPropertyChanged("Cities");
            }
        }
    }
}

```

```

    }
}

private ObservableCollection<Country> privCountries;
public ObservableCollection<Country> Countries
{
    get { return privCountries; }
    set
    {
        privCountries = value;
        OnPropertyChanged("Countries");
    }
}

private string privtitle;
public string Title
{
    get { return privtitle; }
    set
    {
        privtitle = value;
        OnPropertyChanged("Title");
    }
}

private string privdescription;
public string Description
{
    get { return privdescription; }
    set
    {
        privdescription = value;
        OnPropertyChanged("Description");
    }
}

private ObservableCollection<Photo> privPhotos;
public ObservableCollection<Photo> Photos
{
    get { return privPhotos; }
    set
    {

```

```

        privPhotos = value;
        OnPropertyChanged("Photos");
    }
}

private string privregion;
public string Region
{
    get { return privregion; }
    set
    {
        privregion = value;
        OnPropertyChanged("Region");
    }
}

private string privaddress;
public string Address
{
    get { return privaddress; }
    set
    {
        privaddress = value;
        OnPropertyChanged("Address");
    }
}

private string privphotoroute;
public string PhotoRoute
{
    get { return privphotoroute; }
    set
    {
        privphotoroute = value;
        OnPropertyChanged("PhotoRoute");
    }
}

private int privrooms;
public int Rooms
{
    get { return privrooms; }
    set

```

```

        {
            privrooms = value;
            OnPropertyChanged("Rooms");
        }
    }

    private int privfloors;
    public int Floors
    {
        get { return privfloors; }
        set
        {
            privfloors = value;
            OnPropertyChanged("Floors");
        }
    }

    private bool privbathroom;
    public bool Bathroom
    {
        get { return privbathroom; }
        set
        {
            privbathroom = value;
            OnPropertyChanged("Bathroom");
        }
    }

    private int privbeds;
    public int Beds
    {
        get { return privbeds; }
        set
        {
            privbeds = value;
            OnPropertyChanged("Beds");
        }
    }

    private bool privwifi;
    public bool WiFi
    {
        get { return privwifi; }
        set
    }

```

```

        {
            privwifi = value;
            OnPropertyChanged("WiFi");
        }
    }

    private bool privparking;
    public bool Parking
    {
        get { return privparking; }
        set
        {
            privparking = value;
            OnPropertyChanged("Parking");
        }
    }

    private bool privconditioner;
    public bool Conditioner
    {
        get { return privconditioner; }
        set
        {
            privconditioner = value;
            OnPropertyChanged("Conditioner");
        }
    }

    private bool privtv;
    public bool TV
    {
        get { return privtv; }
        set
        {
            privtv = value;
            OnPropertyChanged("TV");
        }
    }

    private int privacctype;
    public int Acc_Type
    {
        get { return privacctype; }
        set
    }

```

```

        {
            privacctype = value;
            OnPropertyChanged("Acc_Type");
        }
    }

    private bool privheating;
    public bool Heating
    {
        get { return privheating; }
        set
        {
            privheating = value;
            OnPropertyChanged("Heating");
        }
    }

    private bool privkitchen;
    public bool Kitchen
    {
        get { return privkitchen; }
        set
        {
            privkitchen = value;
            OnPropertyChanged("Kitchen");
        }
    }

    private bool privbathroom;
    public bool Bathroom
    {
        get { return privbathroom; }
        set
        {
            privbathroom = value;
            OnPropertyChanged("Bathroom");
        }
    }

    private int privmaxstay;
    public int MaxStay
    {
        get { return privmaxstay; }
        set
    }

```

```

        {
            privmaxstay = value;
            OnPropertyChanged("MaxStay");
        }
    }

    private bool privwasher;
    public bool Washer
    {
        get { return privwasher; }
        set
        {
            privwasher = value;
            OnPropertyChanged("Washer");
        }
    }

    private bool privelevator;
    public bool Elevator
    {
        get { return privelevator; }
        set
        {
            privelevator = value;
            OnPropertyChanged("Elevator");
        }
    }

    private int privrefund;
    public int Refund
    {
        get { return privrefund; }
        set
        {
            privrefund = value;
            OnPropertyChanged("Refund");
        }
    }

    private int privguests;
    public int Guests
    {
        get { return privguests; }
        set
    }

```

```

        {
            privguests = value;
            OnPropertyChanged("Guests");
        }
    }

    private bool privpets;
    public bool Pets
    {
        get { return privpets; }
        set
        {
            privpets = value;
            OnPropertyChanged("Pets");
        }
    }

    private City privcity;
    public City AccomCity
    {
        get { return privcity; }
        set
        {
            privcity = value;
            OnPropertyChanged("City");
        }
    }

    private Country privcountry;
    public Country AccomCountry
    {
        get { return privcountry; }
        set
        {
            privcountry = value;
            OnPropertyChanged("AccomCountry");
        }
    }

    private int privpoints;
    public int Points
    {
        get { return privpoints; }
        set
    }

```

```

        {
            privpoints = value;
            OnPropertyChanged("Points");
        }
    }

    private int privviews;
    public int Views
    {
        get { return privviews; }
        set
        {
            privviews = value;
            OnPropertyChanged("Views");
        }
    }

    public RelayCommand ImageAdd
    {
        get
        {
            return privimageadder
?? (new RelayCommand(obj =>
            {
                Stream ifs = File.OpenRead(PhotoRoute);
                byte[] barr=new byte[ifs.Length];
                ifs.Read(barr, 0, (int) ifs.Length);
                TcpClient tc = new
TcpClient("derevjankome.me", 1577);
                NetworkStream ns = tc.GetStream();
                ns.Write(barr,0, barr.GetLength(0));
                ns.Close();
                Photo result =
connec.Photo.Add(PhotoRoute);
                connec.Save();
                if (!result)
                {
                    MessageBox.Show("Виникла проблема.
Зображення не завантажено", "Error", MessageBoxButton.OK,
MessageBoxImage.Error);
                }
                else
                {

```

```

        Photos.Add(result);
    }
    }));
}

}

public RelayCommand Add
{
    get
    {
        return privadder ?? (new RelayCommand(obj =>
        {
            bool result =
connec.Accomodations.Add(Owner, Title, Description, Region, Address,
Rooms, Floors, Bathroom, Beds, WiFi, Parking, Conditioner, TV,
Acc_Type, Heating, Kitchen, Bathroom, MaxStay, Washer, Elevator,
Refund, Guests, Pets, AccomCity, Points, Views, Photos);
            connec.Save();

            if (!result)
            {
                MessageBox.Show("Виникла проблема.
Запис не збережено", "Error", MessageBoxButton.OK,
MessageBoxImage.Error);
            }
        }));
    }
}
}
}

```

ДОДАТОК В
Слайди презентації

Національний університет «Запорізька політехніка»

кафедра програмних засобів

Дипломна кваліфікаційна робота

**Програмне забезпечення короткострокової
оренди житла
Software for Short-Term Accommodation Rental**

Виконав
ст. гр. КНТ-218сп

Дерев'янка Є.В.

Керівник
к.т.н., доц.

Льовкін В.М.

2021

Рисунок В.1 – Слайд № 1

Об'єкт роботи – процес оренди житла.

Предмет роботи – програмне забезпечення короткострокової оренди житла.

Мета роботи – розроблення програмного забезпечення короткострокової оренди житла для забезпечення взаємодії власників житла з орендарями.

► 2

Рисунок В.2 – Слайд № 2

Завдання дипломної кваліфікаційної роботи

- визначення вимог до програмного забезпечення короткострокової оренди житла;
- вибір мови і засобів проєктування і розробки;
- проєктування програмного забезпечення;
- реалізація і налагодження програмного забезпечення.

► 3

Рисунок В.3 – Слайд № 3

Вибір мови програмування

Критерій	Java	C#
Об'єктно-орієнтована парадигма програмування	Так	Так
Корпоративний рівень взаємодії з базою даних	Так	Так
Створення кросплатформних рішень	Так	Так, за рахунок платформи .NET Core
Підтримка фреймворків розробки	Так	Так
Виконання програм	Середня швидкість	Висока швидкість
Проектування і розробка графічного інтерфейсу	Так	Зручна у WPF-рішеннях за рахунок відділення етапу

► 4

Рисунок В.4 – Слайд № 4

Діаграма прецедентів

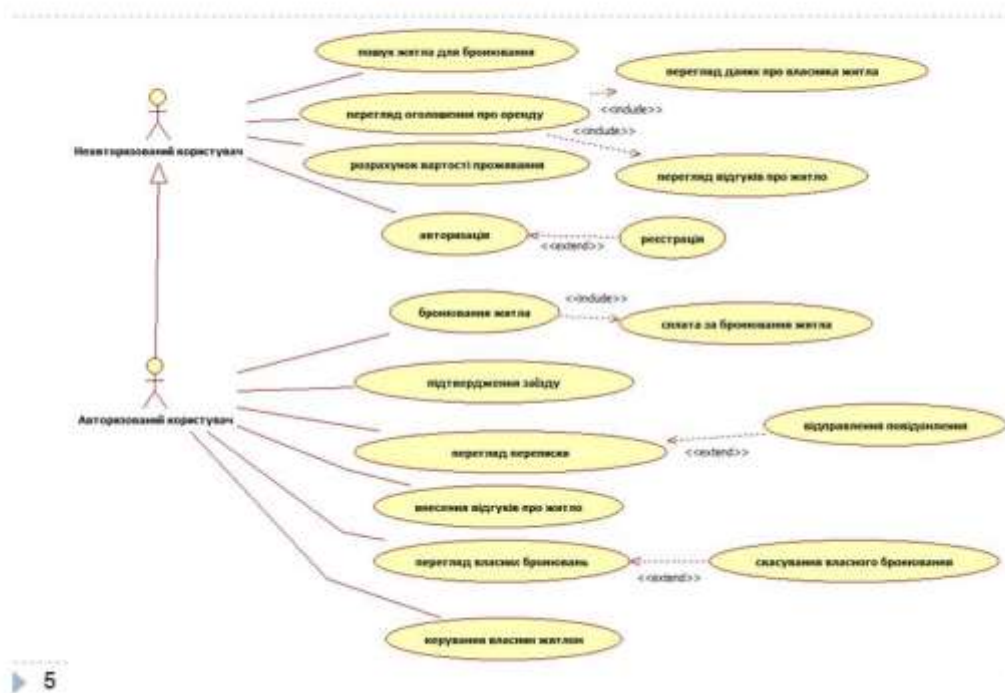


Рисунок В.5 – Слайд № 5

Схема бази даних

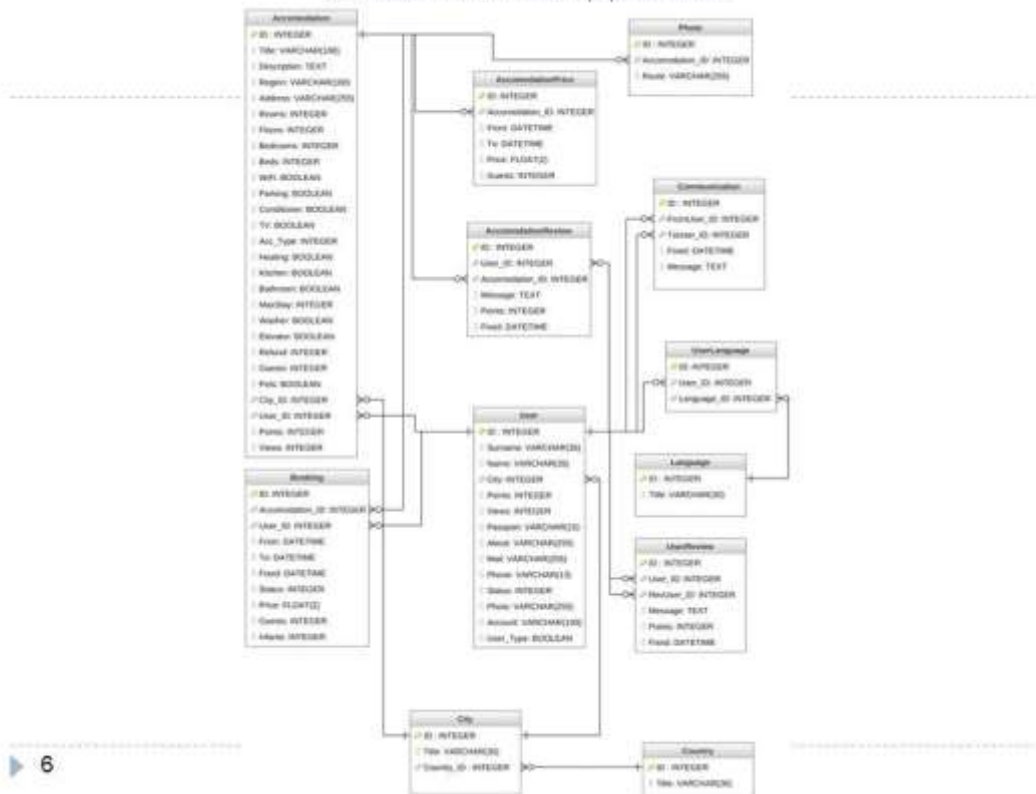


Рисунок В.6 – Слайд № 6

Архітектурний шаблон MVVM

Пошук житла для оренди

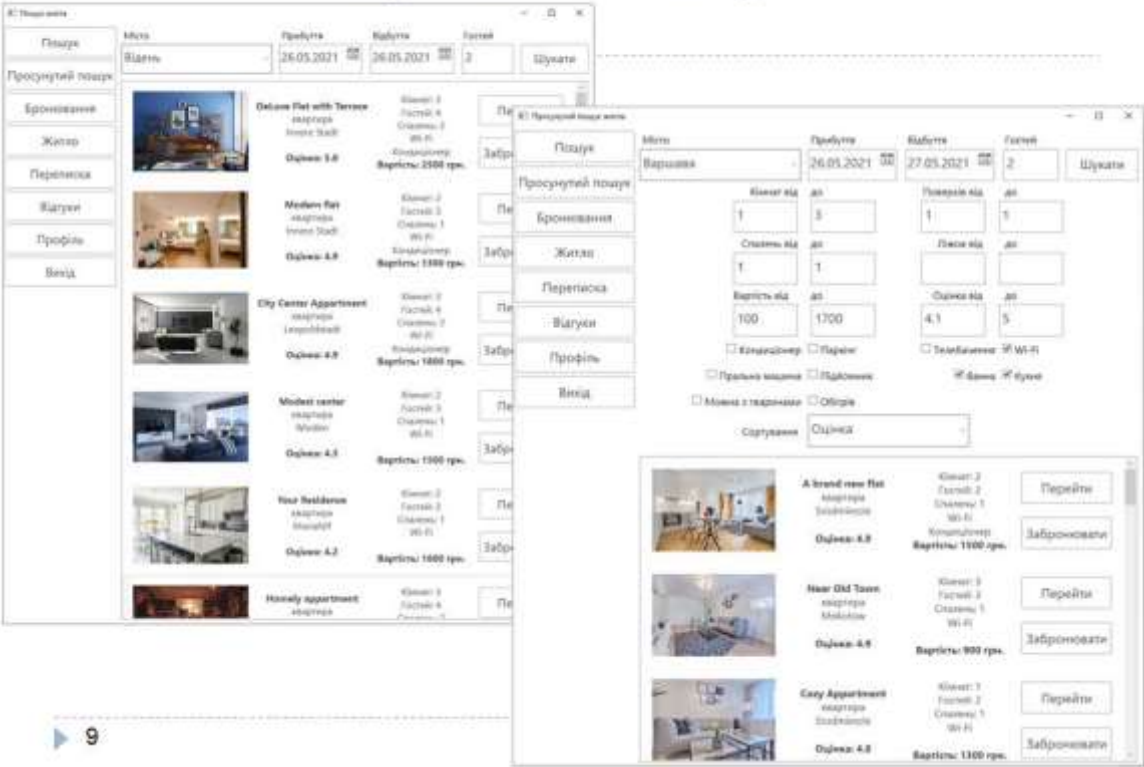


Рисунок В.9 – Слайд № 9

Бронювання житла

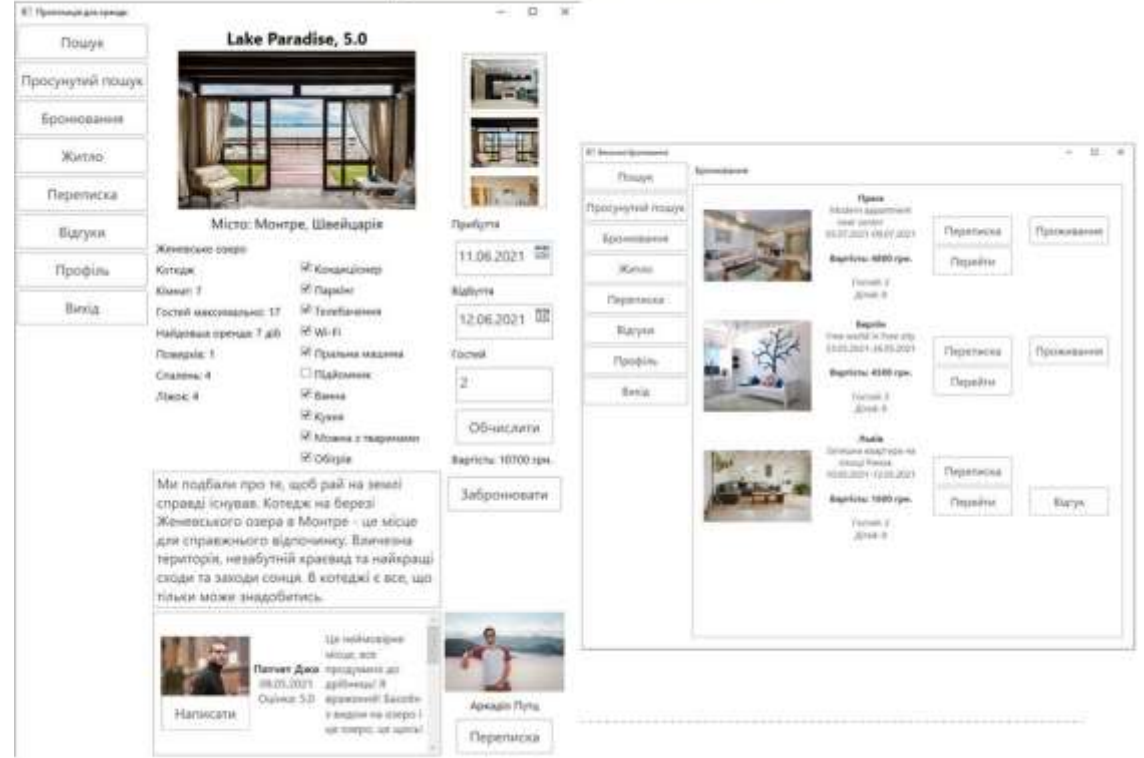


Рисунок В.10 – Слайд № 10

Керування власними пропозиціями житла

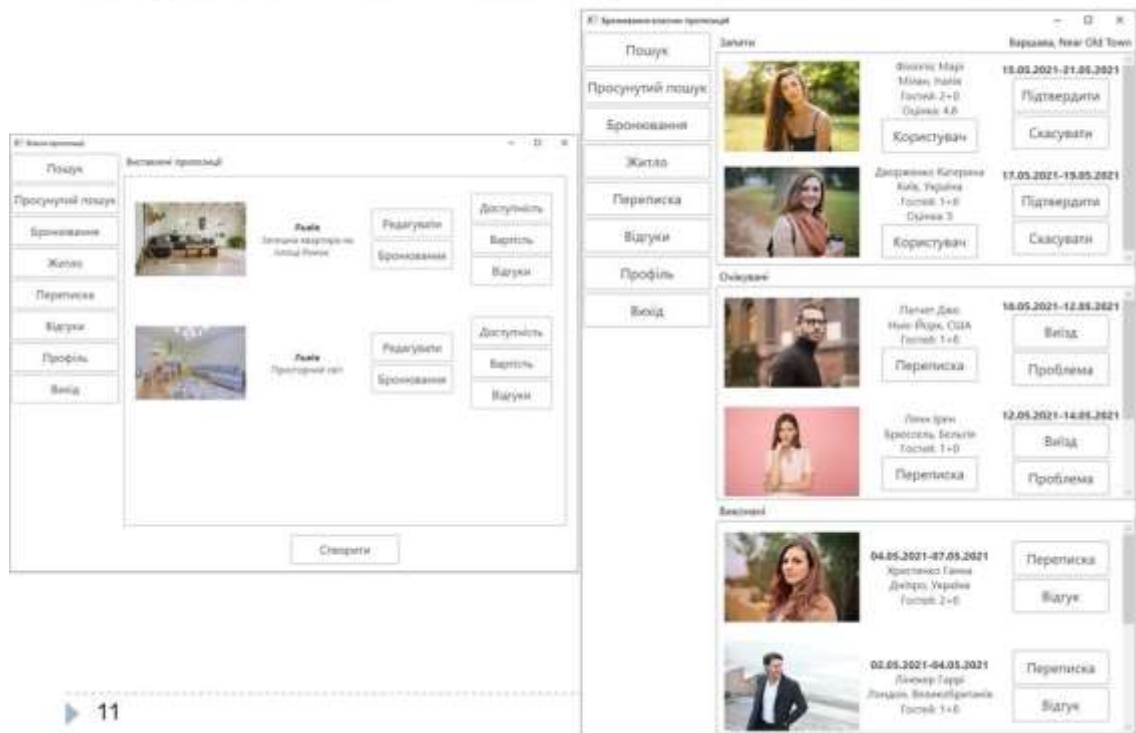


Рисунок В.11 – Слайд № 11

Висновки

Дипломну кваліфікаційну роботу бакалавра присвячено проблемі розробки програмного забезпечення короткострокової оренди житла. Для цього розглянуто особливості короткострокової оренди порівняно з довгостроковою, особливості організації програмного забезпечення для підтримки даного процесу, виконано аналіз програмних аналогів, що включають Airbnb, VrBO, Flipkey.

Реалізацію програми було вирішено виконувати за допомогою мови програмування C#, для чого було виконано порівняння альтернативних варіантів та створено таблицю порівняння. Виконано моделювання сценаріїв роботи авторизованого та неавторизованого користувачів з програмою, в результаті чого побудовано відповідні діаграми прецедентів. Виконано проектування бази даних для забезпечення роботи з даними. Обрано архітектурний шаблон MVVM для створення програмного забезпечення короткострокової оренди житла.

Розроблено програму і описано структуру програмного забезпечення з представленням структурної схеми. Створений програмний застосунок дозволив реалізувати програмну платформу, де власники житла можуть взаємодіяти з орендарем без участі посередників.

Усі завдання, які було визначено у дипломній кваліфікаційній роботі бакалавра, виконано у повному обсязі.

Рисунок В.12 – Слайд № 12