

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ  
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ  
Факультет інформаційних технологій

УДК 004.4:655.425

«ПОГОДЖЕНО»

Декан факультету  
інформаційних технологій

Болбот І. М., д.п.н., професор

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук

Голуб Б.Л., к.т.н., доцент

\_\_\_\_\_ 2024 р.

\_\_\_\_\_ 2024 р.

**МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА**

на тему Програмне забезпечення експертної системи букінгу в Україні

Спеціальність 121 «Інженерія програмного забезпечення»

(код і назва)

Освітня програма Програмне забезпечення інформаційних систем

(назва)

Орієнтація освітньої програми освітня-професійна

(освітньо-професійна або освітньо-наукова)

**Гарант освітньої програми**

професор, д.т.н.  
(науковий ступінь та вчене звання)

(підпис)

Семко В. В.  
(ПІБ)

**Керівник магістерської кваліфікаційної роботи**

ст. викладач  
(науковий ступінь та вчене звання)

(підпис)

Бородкін Г. О.  
(ПІБ)

**Виконав**

(підпис)

Наумов В.В.  
(ПІБ студента)

КИЇВ-2024

## ЗМІСТ

ВСТУП .....	3
РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	5
1.1 Опис предметної області.....	5
1.2 Огляд існуючих рішень .....	10
1.3 Постановка задачі .....	14
1.4 Висновки до першого розділу .....	16
РОЗДІЛ 2. МОДЕЛЮВАННЯ СИСТЕМИ.....	17
2.1 Функціональне моделювання .....	17
2.2 Об'єктно-орієнтоване моделювання.....	19
2.3 Порівняння підходів до моделювання .....	20
2.4 Висновки до другого розділу.....	22
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ .....	24
3.1 Архітектура системи.....	24
3.2 Архітектура підсистем.....	25
3.3 Алгоритми обробки інформації.....	27
3.4 Технології реалізації .....	29
3.5 Висновки до третього розділу .....	36
РОЗДІЛ 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ.....	37
4.1 Апаратні вимоги.....	37
4.2 Програмні вимоги .....	37
4.3 Опис виконання дослідження .....	38
4.4 Аналіз отриманих результатів .....	52
4.5 Висновки до четвертого розділу .....	53
ВИСНОВКИ.....	55
ПЕРЕЛІК ПОСИЛАНЬ.....	57

## ВСТУП

В умовах розвитку цифрових технологій автоматизація процесів бронювання (букінгу) стає важливою складовою ефективного управління у багатьох галузях, включаючи туризм, готельний бізнес, транспорт та інші сфери послуг. Створення експертної системи букінгу в Україні є актуальним завданням, оскільки вона дозволяє підвищити швидкість, точність і зручність обробки даних, забезпечуючи користувачам швидкий доступ до необхідних послуг.

Актуальність. Зважаючи на стрімкий розвиток туристичної індустрії та збільшення попиту на онлайн-бронювання, виникає необхідність у впровадженні експертних систем, здатних оптимізувати цей процес. Експертна система букінгу може значно підвищити ефективність взаємодії між користувачами та сервісами, забезпечити зручність вибору і бронювання послуг, а також мінімізувати людські помилки.

Об'єкт дослідження - процеси автоматизації систем бронювання, що застосовуються в туристичній і готельній сферах України.

Предмет дослідження - експертні системи букінгу та їх застосування для автоматизації процесів бронювання в Україні.

Мета дослідження - розробка експертної системи букінгу, що оптимізує процеси бронювання послуг в Україні, підвищуючи їх ефективність, зручність та точність для користувачів.

Завдання дослідження:

- проаналізувати існуючі системи бронювання, які використовуються в Україні;
- дослідити методи та алгоритми, які можуть бути застосовані для створення експертної системи;
- розробити архітектуру та модель експертної системи букінгу;
- оцінити ефективність запропонованої системи через тестування на реальних даних.

Для вирішення поставлених завдань використовуватимуться методи системного аналізу, моделювання, алгоритмізації та тестування на реальних даних.

Розроблено експертну систему букінгу, яка враховує специфіку українського ринку послуг, забезпечує більш ефективний і швидкий процес бронювання з можливістю адаптації до різних вимог користувачів та сервісів.

## РОЗДІЛ 1. СИСТЕМНИЙ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Опис предметної області

У науковій роботі О.П. Просович та Ю.С. Боцман досліджено українські маркетплейси з позиції цифрового маркетингу. Вони класифікують лідерів цього сегмента на такі чотири групи: інтернет-магазини, прайс-агрегатори, маркетплейси та класифайди [1, с. 32]. Т.М. Ткачук аналізує популяризацію віртуального простору для розвитку туристичних мереж, зокрема через франчайзинг. Вона зазначає, що віртуальні туристичні мережі мають численні переваги, однак стикаються з технічними (вірусні атаки, брак знань персоналу), юридичними та інтелектуальними викликами [2, с. 72]. С.П. Гурська досліджує суть і переваги маркетплейсів як нового сегмента електронної комерції [3, с. 26]. На сьогодні досліджень, присвячених функціонуванню маркетплейсів в Україні, небагато, хоча з огляду на їхній стрімкий розвиток такі роботи є вкрай необхідними. Стаття ставить за мету аналіз діяльності Booking Holdings як одного з провідних маркетплейсів на світовому туристичному ринку.

Маркетплейс (від англ. Marketplace – «ринкова площа») – це онлайн-торговий майданчик, який об'єднує продавців і покупців для взаємодії [3, с. 33]. Його модель заснована на Інтернет-платформі, де користувачі з усього світу можуть переглядати пропозиції різних постачальників. Вибравши найкращу, клієнт робить покупку, а частина платежу йде на користь власникам маркетплейсу.

Ця модель вигідна для обох сторін: постачальники отримують доступ до великої кількості клієнтів, а покупці мають можливість вибору серед широкого асортименту. Власник маркетплейсу, на відміну від інтернет-магазину, не продає товари безпосередньо, а лише надає платформу для інших бізнесів, забезпечує трафік і маркетингову підтримку, отримуючи комісію з продажів.

Присутність багатьох постачальників робить маркетплейс більш складною структурою в порівнянні з традиційною моделлю e-commerce. Завдяки глобальній доступності та можливості пропонувати широкий спектр туристичних послуг і товарів, маркетплейси мають великий потенціал для швидкого розширення та зростання.

Booking Holdings є лідером серед туристичних маркетплейсів, отримуючи 41% доходу всіх онлайн-туристичних компаній. За ним слідують Expedia та Strip з частками 32% і 13% відповідно [2]. Розглянемо детальніше діяльність Booking Holdings як лідера туристичних онлайн-продажів.

Booking Holdings управляє одним із найбільших інтернет-туристичних агентств — Booking.com, через який щодня бронюється понад 1,5 мільйона ночей у готелях. Компанія має понад 200 офісів у більш ніж 70 країнах. Штаб-квартира Booking Holdings розташована у США, а сама компанія активно використовує інформаційні технології. Booking.com було засновано в 1996 році в Амстердамі [5].

Метою Booking Holdings є полегшення організації подорожей для клієнтів. Компанія пропонує рішення для бронювання рейсів, готельних номерів, столиків у ресторанах та прокату автомобілів. До брендів Booking Holdings належать:

- Booking.com – найбільший портал для бронювання готелів у світі, з офісом у Нідерландах, що також пропонує туристичні тури;
- KAYAK – провідний онлайн-сервіс для порівняння цін на готелі, авіаквитки та туристичні пакети;
- Priceline – головний портал у Північній Америці для бронювання авіарейсів, готелів і оренди автомобілів;
- Agoda – лідер серед онлайн-платформ для бронювання проживання в Азіатсько-Тихоокеанському регіоні;
- Rentalcars.com – глобальний сервіс для бронювання автомобілів;
- OpenTable – платформа для онлайн-бронювання столиків у ресторанах [2].

Booking Holdings є світовим лідером у сфері онлайн-подорожей і ресторанних бронювань. Її стратегічні напрями включають:

- надання споживачам найкращих пропозицій для відпочинку та цін у будь-який час і на будь-якому пристрої;
- спрощення процесу пошуку та бронювання подорожей;
- забезпечення партнерів функціональними платформами;
- акцент на інноваціях, цифрових технологіях і розширенні ринків;
- підвищення якості сервісу, розширення вибору місць для проживання;
- надання корисного контенту, включаючи фотографії, огляди та деталі про ресторани.

Отже, використовуючи бренди Booking Holdings, можна забронювати широкий спектр варіантів проживання (готелі, мотелі, курорти, будинки, квартири, гостьові будинки, бунгало та інші об'єкти нерухомості), орендувати автомобіль або замовити таксі в аеропорт, забронювати столик у ресторані, круїз, авіаквиток, туристичний пакет чи екскурсію. Туристи також мають доступ до метапошукових сервісів, що дозволяють швидко порівняти варіанти бронювання (наприклад, авіаквитки, номери в готелях, прокат автомобілів) із сотень онлайн-платформ. Окрім цього, надаються й інші послуги, такі як страхування та управління ресторанными резервуваннями.

Модель з декількома брендами дозволяє Booking Holdings надавати послуги для різних категорій споживачів, використовуючи різноманітні маркетингові та бізнес-стратегії, підтримувати інновації та експерименти, а також зосереджуватися на різних ринках. Це приносить користь усім брендам завдяки можливості обміну досвідом та співпраці.

Booking.com доступний на 43 мовах і пропонує понад 28 мільйонів зареєстрованих об'єктів проживання, з яких понад 6,2 мільйона — це будинки, апартаменти й інші унікальні варіанти житла [4]. Компанія Booking Holdings продовжує активно зміцнювати позиції на ринку альтернативного житла, щоб успішно конкурувати з Airbnb у цій зростаючій категорії. Станом на 2018 рік 80%

об'єктів на Booking.com складала будинки, квартири та інші нетрадиційні варіанти проживання (у 2017 році цей показник становив 75%). У 2023 році кількість таких об'єктів зросла на 47%, тоді як зростання номерного фонду готелів склало 10% (табл. 1.1).

Таблиця 1.1 – Зареєстровані на Booking Holding засоби розміщення

Засоби розміщення	2021	2022	2023	2023/2022, %
Індивідуальні засоби розміщення (будинки, квартири та ін.)	776000	1190000	1744000	47
Колективні засоби розміщення (готелі, мотелі й ін.)	339000	396000	436000	10
Всього	1115000	1586000	2180000	37

Фінансові результати Booking Holdings значною мірою визначаються операційними показниками, зокрема кількістю бронювань та іншою діяльністю, пов'язаною з туристичними послугами. Найбільший приріст зафіксовано у кількості заброньованих номерів — 12,9%, що пояснюється розширенням варіантів проживання та загальним зростанням туристичної галузі, а також переходом від офлайн-до онлайн-бронювання. У 2023 році було здійснено 760 мільйонів бронювань готельних номерів і 7 мільйонів авіаквитків (табл. 1.2).

Таблиця 1.2 – Основні кількісні показники діяльності Booking Holdings

Показники діяльності	2021	2022	2023	Приріст, %
Кількість ночівель	556	673	760	12,9
Кількість днів прокату автомобілів	67	73	73	0,3
Кількість заброньованих авіаквитків	7	7	7	3,8

Валові бронювання — це поширений операційний та статистичний показник, що використовується в туристичній індустрії. Він представляє

загальну вартість у доларах США, зазвичай з урахуванням усіх податків і зборів, усіх туристичних послуг, придбаних споживачами через наші онлайн-сервіси для бронювання подорожей, за вирахуванням скасувань (рис.1.1).

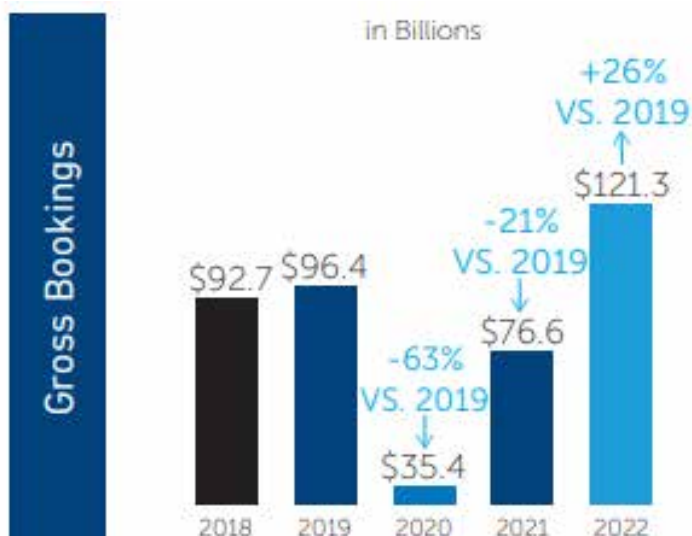


Рис. 1.1 Валові бронювання

Бронювання на ніч за 2021 рік сягнули більше 52% (рис. 1.2).

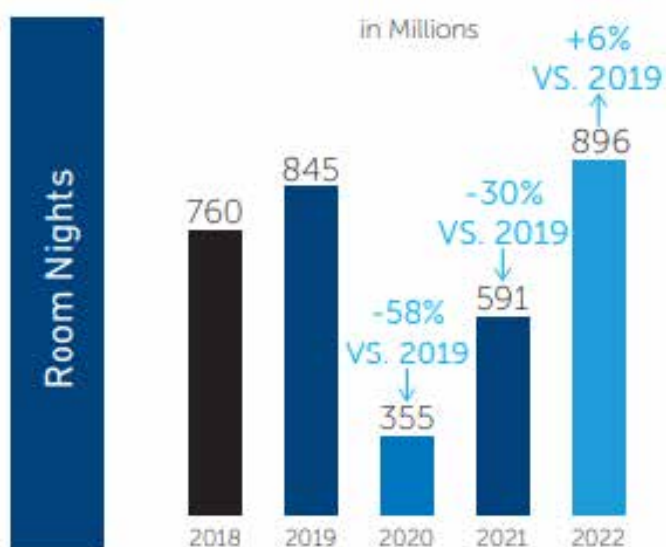


Рис. 1.2 Бронювання за ніч

Таким чином, результат аудиту супровідних консолідованих балансів Booking Holdings Inc. та її дочірніх компаній ("Компанія") станом на 31 грудня 2022 та 2021 років, відповідних консолідованих звітів про результати операцій, сукупний дохід, зміни в капіталі акціонерів та рух грошових коштів за кожен з трьох років, що завершилися 31 грудня 2022 року, включаючи відповідні примітки (разом "фінансові звіти"). На нашу думку, фінансові звіти достовірно відображають, у всіх суттєвих аспектах, фінансовий стан Компанії станом на 31 грудня 2022 та 2021 років, а також результати її операцій та рух грошових коштів за кожен з трьох років, що завершилися 31 грудня 2022 року, відповідно до загальноприйнятих принципів бухгалтерського обліку в Сполучених Штатах Америки.

А також результат аудиту внутрішнього контролю Компанії за фінансовою звітністю станом на 31 грудня 2022 року відповідно до стандартів Ради з нагляду за бухгалтерською діяльністю публічних компаній (РСАОВ) США, на основі критеріїв, встановлених у документі "Внутрішній контроль - інтегрована структура" (2013), виданому Комітетом спонсорських організацій Комісії Тредвея, і в нашому звіті від 23 лютого 2023 року висловили безумовну думку щодо внутрішнього контролю Компанії за фінансовою звітністю [5].

## 1.2 Огляд існуючих рішень

Засоби та методи прикладної інформатики переважно застосовуються у сфері управління. Нові технології, засновані на використанні обчислювальної техніки, потребували повної зміни організаційної структури управління, процедур, регламентів, кадрового складу, а також системи документообігу, обліку та передачі інформації.

Одним зі способів впровадження нових інформаційних технологій є створення автоматизованої системи управління (АСУ). При розробці економічної моделі управлінської організації важливим завданням стає

оптимізація організаційних відносин таким чином, щоб структура управління відповідала реальному стану підприємства, зокрема, в готельному господарстві.

Повна інтеграція автоматизованих процесів управління охоплює такі інформаційні процеси, як комунікація, збір, зберігання та доступ до необхідної інформації, її аналіз, підготовка документації, підтримка дій, програмування та вирішення спеціальних завдань. Для створення інформаційного середовища потрібно об'єднати всі шість функцій системи автоматизації в центральній точці управління та на кожному робочому місці.

Сучасні технологічні засоби автоматизації інформаційно-управлінської діяльності включають персональні комп'ютери, електронні друкарські машини, копіювальні пристрої, засоби зв'язку, телефонне обладнання, "електронну пошту" та інтегровані готельні мережі.

Деякі готелі та готельні комплекси в Україні мають добре організовані централізовані та складні системи бронювання, що включають три взаємопов'язані підсистеми: планування інженерного обладнання, оперативно-диспетчерський зв'язок та реєстрація й бронювання вільних місць.

Готельна індустрія використовує системи управління власністю (PMS), а ресторанний бізнес — системи управління ресторанами (RMS). Традиційний готельний цикл (прибуття-проживання-від'їзд) трансформувався в низку подій, розділених на три етапи: до підписання договору про поселення, під час укладання договору та після його підписання.

Цей цикл охоплює весь період, від першого контакту потенційного клієнта з готелем до моменту оплати рахунку. Він починається з подій до поселення (бронювання, передоплата, відкриття рахунку), продовжується етапами, пов'язаними з проживанням (номер, їжа, напої), і завершується процесом виїзду (розрахунок, закриття рахунків, завершення обліку клієнтів).

В Україні система "FIDELIO" успішно впроваджується в великих готельних мережах.

Серед світових готельних мереж найбільш популярною версією є «FIDELIO FO v.6», яка працює на системі DOS і відповідає вимогам висококласних готелів. Для окремих готелів частіше використовують версію «FIDELIO FO v.7», розроблену для Windows Revise. Наразі в розробці перебуває версія «FIDELIO Suite 8», яку позиціонують як нове покоління готельного програмного забезпечення. Вона базується на популярній платформі управління базами даних Oracle, яка інтегрована в FIDELIO.

Серед програмних рішень «МІКРОС-ФІДЕЛІО» найбільш поширеним є комплекс для автоматизації процесів прийому та розміщення гостей «ФІДЕЛІО Фронт-офіс» (ФІДЕЛІО ФО).

Система «FIDELIO FO» забезпечує функції індивідуального та групового бронювання номерів, реєстрації, розміщення та виселення гостей, а також управління номерним фондом. Вона є центральною ланкою інформаційної мережі готелю, отримуючи дані з інших модулів, таких як тарифи на телефон, платне телебачення, ресторани сервіси тощо. «FIDELIO FO» спрощує та прискорює процес бронювання, поселення та виселення гостей, зберігаючи дані про попередніх клієнтів. Система дозволяє завжди мати актуальну інформацію про наявність номерів, час прибуття та виїзду гостей, а також аналізувати та встановлювати цінову політику. Вона ефективно працює з великим потоком туристів.

«FIDELIO FO» забезпечує:

- Бронювання та розміщення гостей;
- Управління вартістю проживання та інших послуг;
- Ведення обліку неоплачених рахунків клієнтів;
- Формування попередніх та остаточних рахунків для розрахунків з клієнтами;
- Облік безготівкових розрахунків;
- Генерацію фінансової та статистичної звітності.

Основні функціональні групи, що відповідають за облік клієнта та його дані, включають:

- Збереження історії клієнтів;
- Бронювання;
- Розрахунки;
- Облік та звітність;
- Безготівкові розрахунки.

Система також відстежує стан номерного фонду та роботу покоївок, надсилає попередження щодо незавершеного прибирання та надає звіти про стан номерів. Вона використовує такі поняття, як «Картка гостя», «Бронювання», «Рахунки», «Код послуги» та «Переадресація».

Система «Lodging Touch LIBICA» має кілька переваг [7]:

- Широкий функціонал для управління як бізнес-готелями, так і туристичними готелями.
- Модульність, що охоплює всі процеси обслуговування гостей та управління готелем. Це включає базові модулі (консьєрж, бронювання, реєстрація гостей, прибирання, інженерні послуги), модуль історії гостей, модуль управління номерним фондом та модуль туристичних агентств.
- Потужний звітний модуль для створення різних звітів.
- Інтеграція з ресторанными та іншими готельними системами.

Недоліки системи «Lodging Touch LIBICA» полягають у наступному:

1. Відсутня інтеграція між модулями системи та іншими системами, що стосуються даних клієнтів, таких як кількість бронювань, тривалість перебування, скасування та дохід, отриманий від кожного гостя чи компанії.
2. Система не має функції обробки кредитних карток, що ускладнює введення кодів підтвердження для всіх польських клієнтів у базу даних та відстеження кредитного статусу кожного рахунка.

Цю систему можна використовувати як у невеликих готелях, так і в великих готельних компаніях. Для малих готелів доступна система «Ерітоте Експрес», яка оптимізована для готелів з 15 до 85 номерів.

Відмінною рисою системи Lodging Touch LIBICA є те, що вона поєднує інструменти для управління готельними операціями з можливостями корпоративного управління готельною компанією та потужними функціями онлайн-комерції. Завдяки інтеграції всіх даних, що надходять до мережі готелів, система пропонує широкий спектр можливостей для корпоративного управління та централізованого бронювання номерів.

Ключовою складовою системи «Lodging Touch LIBICA» є можливість виконання різноманітних аналітичних процедур, які дозволяють багатовимірно аналізувати дані, отримані з системи управління, як на рівні окремого готелю, так і на рівні компанії. Це дозволяє порівнювати показники діяльності готелів, встановлювати тенденції та прогнозувати розвиток бізнесу, а також оцінювати ефективність маркетингових заходів.

### 1.3 Постановка задачі

Якість продуктів і послуг сьогодні є одним із ключових факторів успіху будь-якого готельного підприємства. Вона дозволяє залучати нових клієнтів, ефективно конкурувати з іншими готелями, а також підвищувати привабливість закладу для інвесторів та формувати його позитивний імідж. У сучасних умовах ринок готельних послуг висуває жорсткі вимоги до організації якості обслуговування, тому задоволення потреб гостей стає одним із основних пріоритетів у сфері гостинності, поряд із отриманням прибутку. Від першої взаємодії клієнта з готелем через службу бронювання залежить його перше враження про заклад, що в свою чергу впливає на подальші відносини з клієнтами.

Якість наданих послуг є одним із найважливіших чинників, який визначає успіх готельного бізнесу, зокрема у підвищенні рейтингу закладів та залученні нових клієнтів. Ефективність системи бронювання безпосередньо відображається на якості обслуговування гостей, оскільки цей сервіс виконує роль прямого продажу готельних продуктів. Тому удосконалення процесу бронювання — це стратегічно важливе завдання для кожного готелю.

Процес бронювання є складовою частиною замкнутого циклу туристичної послуги. Він передбачає встановлення співпраці між гостем і персоналом готелю, від замовлення номера до надання повного спектра послуг. Спосіб бронювання може відрізнитися, включаючи телефон, Інтернет, пошту або факс, але важливим є те, наскільки ефективно працює система. Професіоналізм і навички менеджерів з бронювання, зокрема в сфері телемаркетингу та комунікації, є важливими для створення позитивного враження про готель та переконання клієнта обрати саме цей заклад. Оскільки турист зазвичай переглядає кілька варіантів перед остаточним вибором, менеджери мають підкреслювати конкурентні переваги своїх послуг, щоб переконати клієнта.

Сучасні електронні системи бронювання забезпечують швидку та ефективну обробку замовлень, створення заявок, перевірку звітів та інші важливі функції, що сприяють технічній і якісній роботі готелю. Для великих готельних мереж, особливо міжнародних, використання сучасних інформаційних технологій, включаючи автоматизовані системи управління та онлайн-бронювання, стає важливим елементом успішної діяльності.

Такі системи дозволяють готелям оперативно взаємодіяти з туристичними агентствами та інтегруватися в глобальні системи бронювання через спеціалізовані портали. Однак невеликі готелі та немережеві заклади часто стикаються з труднощами через високу вартість обслуговування таких систем, тоді як великі готельні мережі витрачають значні кошти на підтримку ефективної роботи своїх сервісів бронювання. У таких умовах кожен готель самостійно

обирає систему бронювання, яка найкраще відповідає його фінансовим можливостям, системі управління та потребам щодо конкурентоспроможності.

#### 1.4 Висновки до першого розділу

У першому розділі було проведено системний аналіз предметної області, що стосується автоматизації готельних послуг, зокрема процесів бронювання, реєстрації та обслуговування гостей. Описано ключові аспекти роботи готельних систем, які включають значення якості наданих послуг для залучення нових клієнтів і підтримання конкурентоспроможності на ринку.

В огляді існуючих рішень проаналізовано популярні готельні програмні продукти, такі як «FIDELIO FO», «Lodging Touch LIBICA» та їх функціональні можливості, а також недоліки, пов'язані з інтеграцією і взаємодією з іншими системами. Особливу увагу було приділено важливості ефективної організації процесу бронювання, що суттєво впливає на загальну якість обслуговування клієнтів.

Постановка задачі вказує на необхідність розробки нової системи, яка б враховувала існуючі проблеми та відповідала сучасним вимогам ринку. Системний підхід до аналізу предметної області дозволив виявити ключові фактори, що впливають на успішність готельного бізнесу, і сформулювати основні завдання для подальшого дослідження та розробки.

Таким чином, перший розділ окреслив актуальність теми, вказав на основні проблеми, які потребують вирішення, та надав чітке уявлення про напрямки подальшої роботи в межах кваліфікаційної роботи.

## РОЗДІЛ 2. МОДЕЛЮВАННЯ СИСТЕМИ

### 2.1 Функціональне моделювання

IDEF0 — це методологія функціонального моделювання та графічного опису процесів, яка призначена для формалізації та опису бізнес-процесів. Особливістю IDEF0 є акцент на ієрархічному представленні об'єктів, що значно спрощує розуміння предметної області. У IDEF0 акцентується увага на логічних зв'язках між завданнями, а не на їхній часовій послідовності (Workflow).

Крім того, в моделі відображаються всі управлінські сигнали. Ця модель вважається однією з найсучасніших і використовується для організації бізнес-проектів та проектів, що базуються на моделюванні всіх процесів, як адміністративних, так і організаційних [8].

Функціональну модель наведено на рис. 2.1.

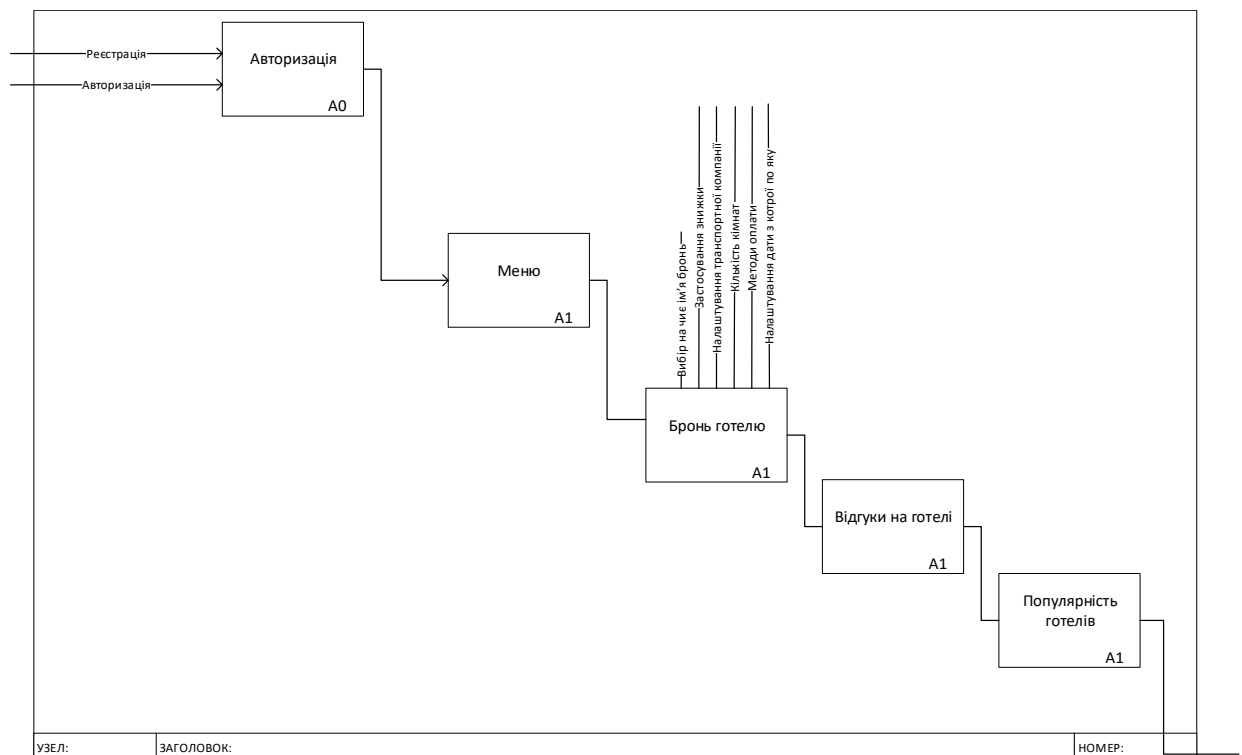


Рис. 2.1 Функціональна модель

Функціональна модель, зображена на діаграмі, що описує процес бронювання готелів і взаємодію користувача з системою. Опишемо основні блоки та їх зв'язки:

Авторизація - користувач починає роботу із системою, пройшовши процедуру авторизації або реєстрації, якщо новий користувач.

Меню - після авторизації користувач отримує доступ до меню, де представлені різні функції для взаємодії із системою бронювання готелів.

Відгуки на готелі - система дозволяє переглядати відгуки на готелі, що допомагає користувачеві зробити вибір.

Оцінка та відгук - користувач також може залишити свій відгук і оцінку після проживання в готелі.

Популярність готелів - користувач може переглянути рейтинг або популярність готелів, що може вплинути на вибір.

Бронювання готелів - основний процес, що містить декілька підпроцесів:

- вибір на ім'я чи номер броні - користувач може здійснити пошук і вибір готелю за ім'ям чи номером броні.

- методи оплати - вибір і підтвердження методу оплати для бронювання;

- налаштування дати з і по яку - користувач вибирає дати для бронювання готелю;

- кількість кімнат - можливість вибору кількості кімнат при бронюванні;

- застосування знижки - якщо доступні акції або промокоди, користувач може застосувати знижки;

- налаштування транспортної компанії – обирання транспортної компанії (потяг чи літак).

Модель описує процеси, необхідні для бронювання готелю, і показує, як користувач взаємодіє з системою на кожному етапі — від авторизації до вибору методу оплати та подальшої взаємодії після бронювання (відгуки, оцінка).

## 2.2 Об'єктно-орієнтоване моделювання

Об'єктно-орієнтоване моделювання (ООМ) — це методологія моделювання програмного застосунку, яка застосовується на початкових етапах життєвого циклу програмного забезпечення в контексті об'єктно-орієнтованого підходу до розробки.

Життєвий цикл програмного забезпечення зазвичай складається з кількох етапів: від абстрактного опису проблеми до проектування, а потім — до кодування, тестування і, в кінцевому рахунку, розгортання. Моделювання виконується на початку цього процесу. Основні причини для моделювання системи перед написанням коду:

**Комунікація.** Користувачі зазвичай не мають змоги зрозуміти мову програмування або код. Діаграми моделі можуть бути більш зрозумілими, що дозволяє користувачам давати розробникам зворотний зв'язок щодо відповідної структури системи. Ключова мета об'єктно-орієнтованого підходу полягає в зменшенні «семантичної прогалини» між системою та реальним світом, використовуючи терміни, що відповідають функціям, які виконують користувачі. Моделювання є важливим інструментом для досягнення цієї мети.

**Абстракція.** Більшість методологій програмного забезпечення намагаються спочатку відповісти на запитання «що», а потім — «як». Тобто, спочатку визначають функціональність, яку має надати система, не беручи до уваги обмеження реалізації, а потім розглядають, як перетворити цей абстрактний опис на реальний дизайн і код, враховуючи такі обмеження, як технології та бюджет. Моделювання забезпечує можливість створення абстрактних описів процесів і об'єктів, які визначають їх основну структуру та поведінку [9].

Об'єктно-орієтована модель наведена на рис. 2.2.

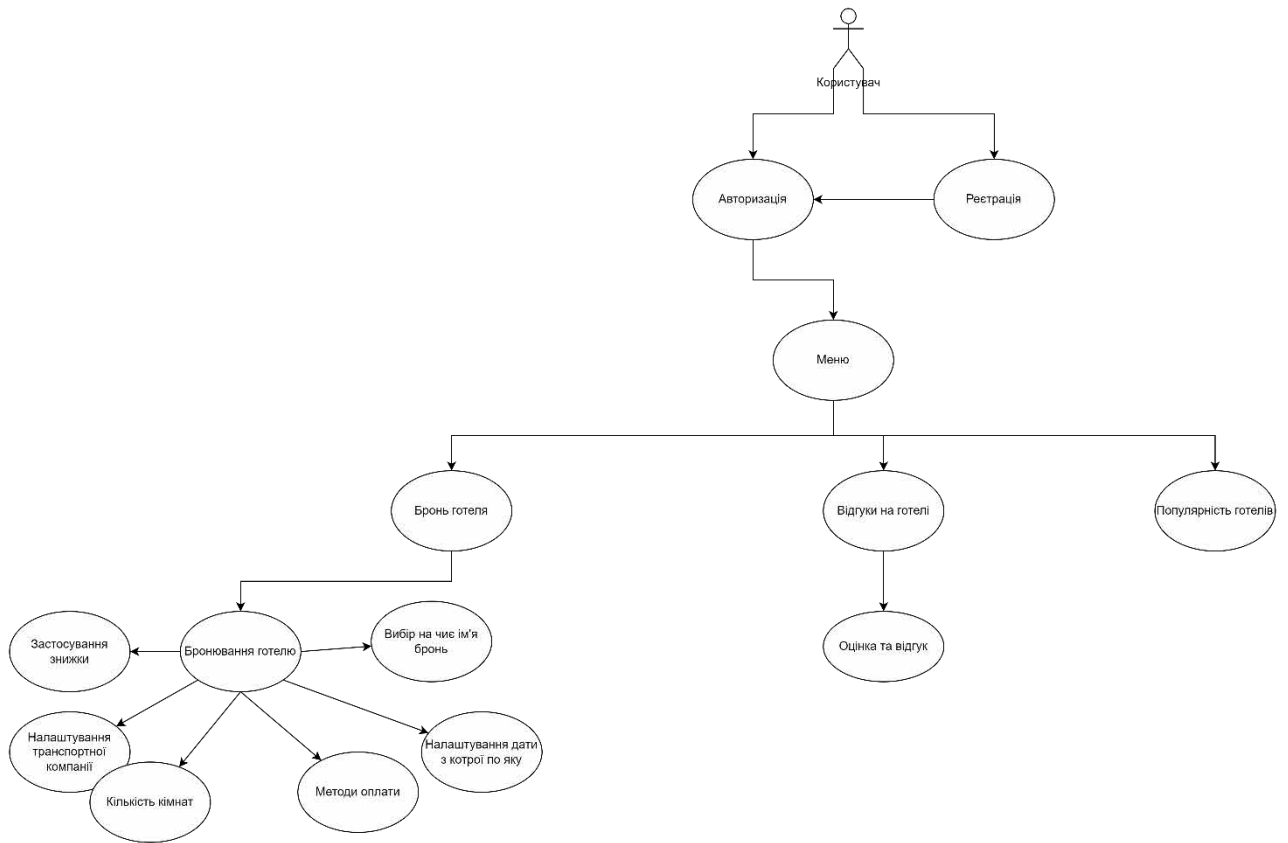


Рис. 2.2 Об'єктно-орієнтована модель

### 2.3 Порівняння підходів до моделювання

Об'єктно-орієнтоване (ООМ) та функціональне моделювання (ФМ) є двома популярними підходами до моделювання систем, які мають свої особливості, переваги та недоліки. Нижче наведено порівняння цих двох підходів.

Об'єктно-орієнтоване моделювання - сфокусоване на об'єктах, які є основними елементами системи. Кожен об'єкт має свої атрибути (дані) та методи (функції), що дозволяють взаємодіяти з іншими об'єктами.

Використовує поняття класів та ієрархії спадкування, що дозволяє створювати гнучкі й масштабовані системи.

Функціональне моделювання - сфокусоване на функціях або процесах, які система виконує. Основна увага приділяється тому, що система повинна робити, а не як це реалізується.

Зазвичай використовує діаграми, які показують вхідні дані, вихідні дані та трансформації.

Об'єктно-орієнтоване моделювання використовує UML (Unified Modeling Language) для графічного представлення системи, де об'єкти, класи та їх взаємозв'язки показуються через діаграми класів, об'єктів тощо.

Підтримує ієрархічну структуру, що спрощує управління складними системами.

Функціональне моделювання. Зазвичай використовує нотацію IDEF0 або діаграми потоків даних (DFD) для відображення процесів та їх взаємозв'язків.

Більше фокусується на вхідних та вихідних даних функцій, ніж на їх внутрішньому функціонуванні.

Об'єктно-орієнтоване моделювання. Підкреслює важливість взаємодії між об'єктами, що дозволяє моделювати складні системи та їх поведінку.

Забезпечує високу рівень абстракції, що дозволяє уникнути зайвої деталізації.

Функціональне моделювання. Підкреслює важливість послідовності виконання процесів, що може бути корисно для аналізу бізнес-процесів.

Може бути легшим для розуміння не технічними спеціалістами, оскільки фокусується на тому, що система повинна робити.

Об'єктно-орієнтоване моделювання. Часто використовується в розробці програмного забезпечення, особливо в середовищах, де потрібна гнучкість і підтримка складних систем.

Функціональне моделювання. Застосовується в бізнес-аналізі, системному аналізі та процесному управлінні, особливо у випадках, коли важливо відстежувати функціональність системи.

## 5. Переваги та недоліки

Об'єктно-орієнтоване моделювання:

Переваги: гнучкість, повторне використання коду, полегшене управління складними системами.

Недоліки: може бути складнішим для розуміння на початкових етапах.

Функціональне моделювання:

Переваги: простота, легкість у спілкуванні з не технічними сторонами, фокус на процесах.

Недоліки: обмежена гнучкість, може бути менш ефективним у складних системах [10-11].

Обидва підходи до моделювання мають свої сильні та слабкі сторони, і вибір між ними залежить від специфіки проекту, потреб замовника та бажаних результатів. У ряді випадків ефективно використовувати комбінацію обох методів, що дозволить максимально реалізувати переваги кожного з них.

#### 2.4 Висновки до другого розділу

У другому розділі було розглянуто процес моделювання системи, зокрема функціональне та об'єктно-орієнтоване моделювання, а також проведено їх порівняння.

Функціональне моделювання надає можливість чітко визначити функції та процеси, які система повинна виконувати. Використання діаграм, таких як IDEF0 та діаграми потоків даних, дозволяє наочно представити взаємозв'язки між функціями та даними. Цей підхід є особливо корисним на етапі аналізу бізнес-процесів, оскільки фокусується на тому, що система повинна робити.

Об'єктно-орієнтоване моделювання акцентує увагу на об'єктах, які є основними елементами системи, що надає можливість створювати гнучкі та масштабовані рішення. Завдяки використанню UML, цей підхід забезпечує високу ступінь абстракції та можливість повторного використання коду, що є важливим фактором у сучасній розробці програмного забезпечення.

Порівняння обох підходів продемонструвало, що кожен з них має свої переваги та недоліки. Функціональне моделювання простіше для розуміння не технічними фахівцями, тоді як об'єктно-орієнтоване моделювання пропонує більшу гнучкість та можливості для роботи з складними системами.

Вибір між цими підходами залежить від специфіки проекту, його вимог і потреб замовника. У деяких випадках доцільно комбінувати обидва методи для досягнення найкращих результатів.

Узагальнюючи, моделювання є критично важливим етапом у розробці систем, оскільки дозволяє зрозуміти їх структуру, функціональність та взаємозв'язки, що, в свою чергу, сприяє успішному проектуванню та реалізації програмного забезпечення.

## РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ

### 3.1 Архітектура системи

Архітектура системи визначає структурні компоненти системи, їх взаємозв'язки та принципи організації. Вона слугує основою для розробки, реалізації та подальшого розвитку програмного забезпечення. У цьому розділі розглянемо основні аспекти архітектури системи, включаючи її компоненти, шарову структуру та взаємодію між елементами.

Архітектура системи складається з кількох ключових компонентів:

- клієнтська частина. Відповідає за взаємодію з користувачем. Це може бути веб-додаток, мобільний додаток або десктопний застосунок. Клієнтська частина забезпечує зручний інтерфейс для користувачів і виконує запити до серверної частини системи;

- серверна частина. Обробляє запити від клієнтської частини, виконує бізнес-логіку, здійснює доступ до бази даних і повертає результати. Сервер може бути реалізований за допомогою різних технологій, таких як RESTful API або SOAP;

- база даних. Зберігає всю інформацію, яка використовується системою. Це може бути реляційна база даних (наприклад, MySQL, PostgreSQL). База даних повинна бути оптимізована для забезпечення швидкого доступу до даних та їх безпеки.

Архітектура системи може бути організована за шаровим принципом, що дозволяє ізолювати різні елементи системи:

- презентаційний шар, що відповідає за взаємодію з користувачем і відображення інформації. У цьому шарі реалізуються інтерфейси користувача;

- логічний шар містить бізнес-логіку та правила, які визначають, як дані повинні оброблятися. Цей шар виконує основні функції системи;

- шар доступу до даних забезпечує взаємодію між логічним шаром та базою даних. Він відповідає за виконання запитів до бази даних та обробку результатів.

Взаємодія між компонентами системи здійснюється через добре визначені інтерфейси. Клієнтська частина надсилає запити до серверної частини, яка, в свою чергу, взаємодіє з базою даних для отримання необхідної інформації. Результати обробки повертаються клієнту.

Архітектура системи забезпечує кілька переваг:

- модульність. Окремі компоненти можна розробляти, тестувати і модифікувати незалежно один від одного;

- гнучкість. Зміни в одній частині системи не впливають на інші компоненти, що спрощує обслуговування;

- масштабованість. Система може бути легко масштабована, додавши нові компоненти або вдосконаливши існуючі.

Архітектура системи є критично важливим етапом у процесі розробки програмного забезпечення, оскільки вона визначає структуру, компоненти та принципи організації системи. Чітке визначення архітектури дозволяє досягти ефективності, гнучкості та зручності в подальшому обслуговуванні програми.

### 3.2 Архітектура підсистем

Архітектура підсистем є важливою складовою загальної архітектури системи. Вона визначає структуру і взаємодію окремих підсистем, які складають систему в цілому. Кожна підсистема виконує конкретні функції та може бути розроблена, протестована та підтримувана незалежно від інших. У цьому розділі розглянемо ключові аспекти архітектури підсистем, включаючи їх функціональні компоненти, зв'язки та способи інтеграції.

Кожна підсистема виконує визначену роль у рамках системи і може містити такі компоненти:

Контролери - відповідають за управління логікою підсистеми, обробку запитів та координацію дій між різними компонентами. Вони виконують бізнес-правила і приймають рішення на основі даних, отриманих від інших компонентів.

Сервіси - забезпечують виконання конкретних функцій, таких як обробка даних, виконання запитів до бази даних, інтеграція з зовнішніми системами тощо. Сервіси можуть бути реалізовані у вигляді RESTful API або інших інтерфейсів.

Моделі даних визначають структуру даних, які використовуються в підсистемі. Це можуть бути об'єкти, які представляють дані, або схеми, що описують, як дані зберігаються і взаємодіють.

Інтерфейси описують, як підсистема взаємодіє з іншими підсистемами або зовнішніми системами. Інтерфейси можуть бути реалізовані через API, повідомлення або інші механізми комунікації.

Взаємодія між підсистемами відбувається через чітко визначені інтерфейси. Основні типи зв'язків включають:

1. Синхронні запити – один компонент викликає функцію в іншій підсистемі і чекає на відповідь. Цей підхід використовується, коли необхідно отримати результат для подальшої обробки.

2. Асинхронні повідомлення – один компонент надсилає повідомлення до іншої підсистеми без очікування відповіді. Цей спосіб забезпечує гнучкість і дозволяє підсистемам працювати незалежно.

3. Події - підсистеми можуть використовувати подієву модель, де одні компоненти реагують на події, що генеруються іншими компонентами. Це дозволяє легко масштабувати систему і реалізувати складні бізнес-процеси.

Інтеграція підсистем може бути реалізована за допомогою різних підходів:

Підсистеми можуть бути реалізовані як незалежні сервіси, які взаємодіють через API. Це дозволяє забезпечити незалежність і масштабованість.

Кожна підсистема реалізується як окремий мікросервіс, що відповідає за певну функціональність. Мікросервісна архітектура дозволяє легко розширювати систему та знижує ризики, пов'язані з відмовами.

Для управління взаємодією між підсистемами можна використовувати системи оркестрації, які контролюють порядок виконання завдань і передачу даних між компонентами.

Архітектура підсистем забезпечує ряд переваг:

1. Модульність підсистеми можна розробляти і тестувати окремо, що спрощує процес підтримки.

2. Гнучкість зміни в одній підсистемі не впливають на роботу інших підсистем, що дозволяє швидше реагувати на зміни вимог.

3. Масштабованість підсистеми можна масштабувати незалежно одна від одної, що дозволяє адаптувати систему до зростаючих навантажень.

Архітектура підсистем є ключовим аспектом загальної архітектури системи, що визначає структуру, взаємозв'язки та інтеграцію різних компонентів. Чітке визначення архітектури підсистем дозволяє досягти гнучкості, модульності та масштабованості, що є критично важливим для успішної розробки та підтримки програмного забезпечення.

### 3.3 Алгоритми обробки інформації

Для обробки інформації користувачу необхідно зробити цілий алгоритм дій (рис. 3.1).

Опишемо даний алгоритм:

Запуск програми – користувач відкриває програму для бронювання готелів.

Введення логіна та пароля або реєстрація. Якщо користувач вже має обліковий запис, він вводить свої дані (логін і пароль). Якщо облікового запису немає, користувач проходить процедуру реєстрації.

Вибір дії – після успішної авторизації користувачеві пропонується обрати подальшу дію із кількох можливих варіантів:

- бронювання готелю;
- відгук та оцінка;
- перегляд популярності готелів;
- вихід.

Бронювання готелю. Якщо обрано цей варіант, користувач переходить до наступного кроку:

Обрати необхідний готель - користувач вибирає готель, який бажає забронювати.

Забронювати готель - після вибору готелю відбувається процес бронювання.

Відгук та оцінка - користувач має можливість залишити відгук або оцінку після користування послугами готелю.

Залишити відгук - користувач пише відгук і додає оцінку, яка може бути корисною для інших користувачів.

Перегляд популярності готелів. Користувач може переглядати рейтинг або популярність різних готелів, щоб краще зорієнтуватися при виборі.

Вихід. Якщо користувач завершив роботу із програмою, він може обрати цей варіант для виходу з системи.

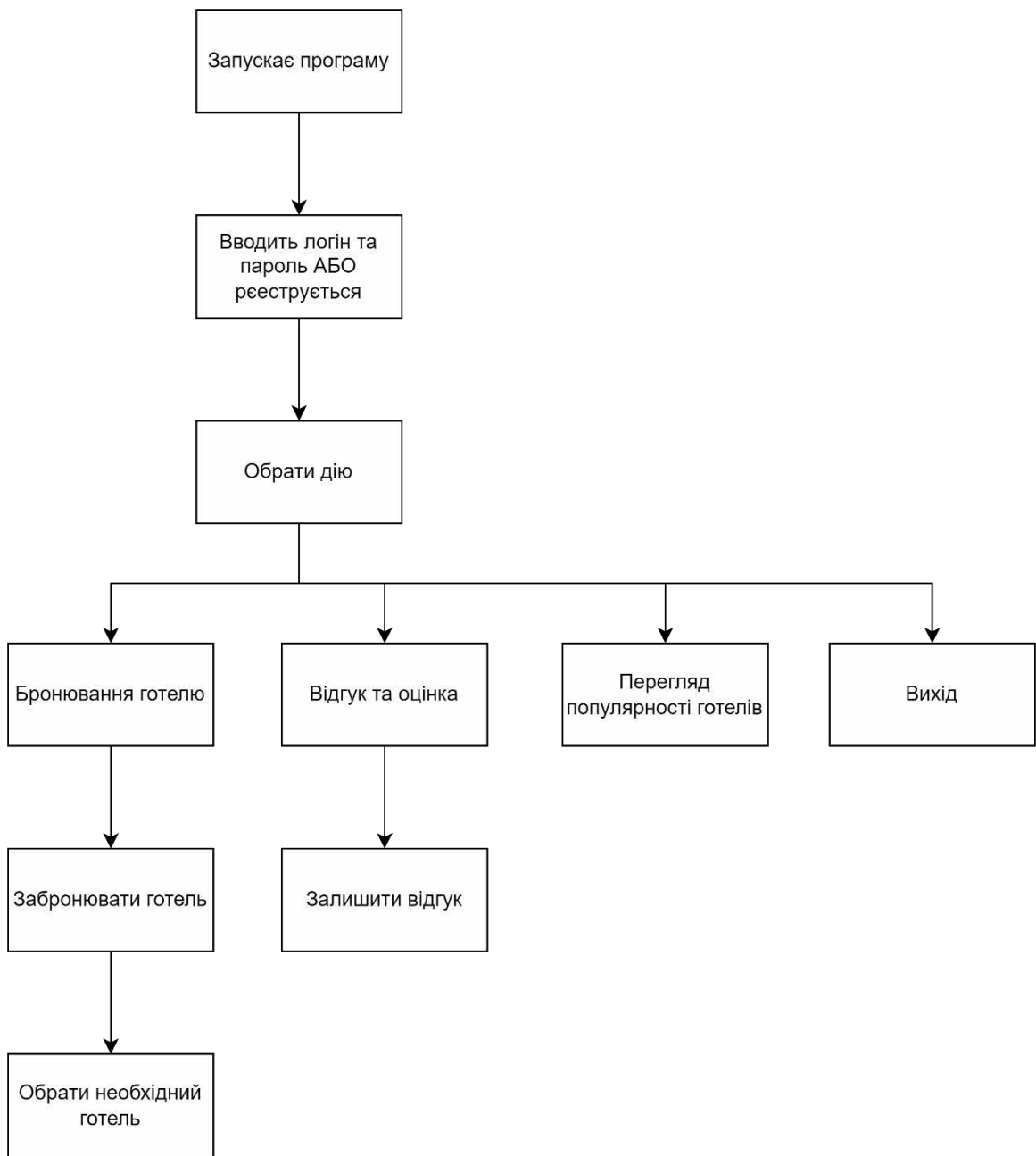


Рис. 3.1 Алгоритм дій

### 3.4 Технології реалізації

Обраними технологіями реалізації є мова програмування C# та використання баз даних MySQL.

Стрімкий розвиток комп'ютерних технологій та поширення Інтернету у всіх сферах життя ставлять нові вимоги до програмного забезпечення. Сучасні

програми функціонують у величезній мережі, що об'єднує мільйони комп'ютерів, тому вони мають підтримувати можливість інтеграції компонентів, створених на різних мовах програмування, які можуть працювати у різних операційних системах. Однак, існуючі мови програмування важко адаптувати до взаємодії між собою, оскільки вони залежать від конкретних операційних систем або навіть різних версій однієї ОС.

Також варто зазначити, що складні програми рідко створюються однією людиною; зазвичай над ними працюють команди програмістів, які використовують різні технології та мови програмування. Особливо це стосується програм, що призначені для роботи в мережі. До появи технології Microsoft .NET Framework така розробка вимагала складної координації між різними частинами програм, оскільки, попри схожий синтаксис мов, вони базувалися на різних принципах.

Microsoft .NET Framework — це технологія, розроблена компанією Microsoft для створення програм різних типів, зокрема веб-застосунків. Однією з ключових ідей цієї технології є забезпечення сумісності між різними мовами програмування, що використовуються для розробки. Програми на платформі .NET можуть взаємодіяти з класами та методами, створеними різними мовами, завдяки тому, що середовище розробки .NET компілює код кожної програми у спеціальну проміжну мову MSIL (Microsoft Intermediate Language), яка потім може бути перетворена у платформозалежний код за допомогою JIT-компілятора (just-in-time). Цей процес контролюється середовищем виконання CLR (Common Language Runtime), яке відповідає за розгортання програм, керування пам'яттю та інші низькорівневі завдання.

Крім того, .NET Framework включає систему типів CTS (Common Type System) та розширену бібліотеку базових класів, яку використовують усі мови платформи .NET [12].

Таким чином, платформа .NET є сучасною моделлю для розробки програмних застосунків, яка, серед інших переваг, надає можливість повної

взаємодії між мовами програмування і використання потужної бібліотеки базових об'єктів.

Хоча платформа .NET підтримує різні мови програмування (такі як C++ і Visual Basic), спеціально для неї була розроблена нова мова програмування — C#. Команда розробників Microsoft створила цю мову, щоб максимально використовувати переваги технології .NET. C# є своєрідним гібридом декількох мов програмування, вбираючи найкращі риси від кожної з них.

Як спадкоємиця мови C++, C# має схожий синтаксис, але при цьому позбавлена неоднозначностей, що можуть виникати в компіляторах C++. Водночас вона має багато спільного з Java, яка була створена як мова, що забезпечує можливість виконання застосунків на різних операційних системах. Java вирішує проблему сумісності програм за допомогою двоетапної компіляції: спочатку Java-компілятор генерує машиннонезалежний байт-код, який виконується віртуальною машиною Java (JVM). Це дозволяє запускати Java-програми на будь-якій платформі, де реалізовано JVM. Однак, Java не вирішує питання взаємодії між різними мовами програмування в одній програмі.

На відміну від Java, C# генерує код, який може виконуватися лише в середовищі .NET, що називається керованим кодом (managed code). Це передбачає автоматичне управління пам'яттю і усуває необхідність працювати з вказівниками на пам'ять, що знижує ймовірність помилок під час розробки і використання програми [13].

Ще одна перевага C# — це можливість одночасно створювати програму та документацію до неї у форматі XML. За допомогою спеціальних коментарів можна автоматично генерувати файл документації в процесі розробки (ця функція вбудована у середовище розробки).

Microsoft Visual Studio, інтегроване середовище для розробки, надає зручний інтерфейс для написання програм на C#, що робить процес створення застосунків ще більш комфортним.

Інтегроване середовище розробки (Integrated Development Environment, IDE) Visual Studio 2019 є універсальним середовищем для всіх мов програмування платформи .NET. Це означає, що незалежно від того, яку мову програмування з набору .NET ви використовуєте, для створення проєктів будь-якого типу ви працюватимете в одному й тому ж середовищі розробки. Більшість функцій Visual Studio можна опанувати самотійно, а зараз розглянемо лише основні з них.

Перш за все, Visual Studio працює з проєктами. Слово «проєкт» походить від латинського і означає «те, що виступає вперед». У цьому контексті проєкт — це вся сукупність програмних засобів для вирішення певного завдання. Проєкт може містити кілька програмних файлів, бази даних, класи тощо.

Опис класу визначає його члени, зокрема дані та методи, які обробляють і керують цими даними. Клас може також містити спеціальні члени, такі як статичні поля, методи, властивості та події. Для початку розглянемо основні члени класу. Синтаксис опису класу починається зі службового слова `class`, після якого слідує блок із визначенням членів класу [14].

Отже, загальний синтаксис визначення класу виглядає так (порядок розміщення членів класу може бути довільним):

```
class <назва_класу> {
    // оголошення полів класу
    <специфікатор_доступу> <тип> <назва_змінної_1>;
    <специфікатор_доступу> <тип> <назва_змінної_2>;
    //...
    // оголошення методів класу
    <специфікатор_доступу>    <тип_результату>    <назва_методу_1>
    (<параметри>) {
        // код методу
    }
    //...
    <специфікатор_доступу>    <тип_результату>    <назва_методу_N>
    (<параметри>) {
        // код методу
    }
}
```

Примітка: Специфікатор доступу визначає, як член класу може використовуватися за його межами, і може мати значення: `public` (для загальнодоступних членів), `protected` або `private` (для закритих членів). Відкриті члени доступні з будь-якої частини коду, де доступний клас, тоді як закриті — лише зсередини самого класу. Якщо специфікатор доступу не вказаний, за замовчуванням використовується `private` [].

База даних — це сукупність взаємопов'язаних даних, організованих відповідно до певної структури (схеми), що дозволяє користувачам взаємодіяти з ними (ДСТУ 2874-94) [10]. В іншому визначенні, база даних — це зібрання інформаційних елементів, що відображають дані з певної предметної області та постійно зберігаються в пам'яті обчислювальної системи (ОС), забезпечуючи доступ до них для різних прикладних програм [15].

Предметна область (ПО) — це частина реального світу, дані про яку зберігаються в базі даних. Вона описується в термінах класів об'єктів і їхніх взаємозв'язків, які зрозумілі кінцевим користувачам та фахівцям у відповідній сфері, навіть якщо вони не знайомі з деталями організації баз даних.

Модель предметної області відображає знання про цю область. У процесі її дослідження інформація подається на трьох рівнях: інфологічному, даталогічному та фізичному.

На інфологічному рівні визначають, яку інформацію про предметну область потрібно зберігати і обробляти в комп'ютері. Інфологічна модель описує дані незалежно від програмного чи апаратного забезпечення, яке буде використане для їхньої обробки та зберігання.

На даталогічному рівні дані представлені у вигляді логічних зв'язків між ними, з урахуванням програмного забезпечення (наприклад, систем управління базами даних — СУБД), але без прив'язки до конкретних технічних засобів. Існують різні види даталогічних моделей даних: мережева, ієрархічна, реляційна, об'єктна, документо-орієнтована тощо.

На фізичному рівні визначаються місце та спосіб зберігання даних на фізичних носіях. Фізична модель даних описує їхню структуру з урахуванням особливостей конкретної СУБД.

Крім того, для відображення інформації про предметну область з точки зору окремого користувача або групи користувачів можуть бути створені зовнішні моделі першого та другого рівнів — інфологічні та даталогічні.

Зовнішній рівень моделей — кожна модель на цьому рівні надає специфічне уявлення про дані. Він визначає, як різні програми бачать базу даних, дозволяючи кожному додатку працювати тільки з тими даними, які йому необхідні.

Концептуальний рівень — це узагальнене уявлення бази даних, яке поєднує всі дані, що використовуються різними додатками, що взаємодіють з базою. Він відображає спільну модель предметної області, для якої створено базу даних.

Фізичний (внутрішній) рівень — це реальне розташування даних у файлах або структурованих блоках на зовнішніх носіях.

Системи управління базами даних (СУБД) поділяються на персональні та багатокористувацькі за характером використання. Залежно від місця розташування елементів СУБД розрізняють локальні (автономні) бази даних та віддалені (мережеві або багатокористувацькі). Перелік і особливості бази даних наведено на рис. 3.2.



Рис. 3.2 Перелік і особливості баз даних

Система баз даних — це комп'ютеризована система для зберігання однорідних записів. Базу даних можна уявити як електронну картотеку, тобто сховище або контейнер для файлів даних, що зберігаються на комп'ютері.

Система баз даних складається з кількох ключових компонентів: дані, апаратне забезпечення, програмне забезпечення та користувачі.

Моделі даних використовуються для формалізації представлення інформації. Модель даних — це абстрактне, логічне визначення об'єктів, операторів і інших елементів, які в сукупності утворюють систему доступу до даних. Об'єкти дозволяють моделювати структуру даних, а оператори — їх поведінку. Це визначення допомагає відрізнити модель даних від її реалізації.

Реалізація моделі даних — це фізичне втілення компонентів моделі на комп'ютері. Модель описує те, що повинно бути зрозумілим користувачам, а реалізація — це внутрішній механізм, яким користувачі не повинні займатися.

Модель даних складається з трьох аспектів:

Аспект структури: визначає логічну будову бази даних.

Аспект маніпуляції: описує способи зміни стану бази даних і методи отримання даних.

Аспект цілісності: встановлює правила для забезпечення коректності бази даних.

Модель даних можна визначити як інтегрований набір елементів для опису даних, зв'язків між ними та обмежень, які накладаються на структуру інформації. Дані організовуються таким чином, щоб забезпечити можливість розширення додатків.

У процесі розвитку баз даних було створено кілька класичних моделей даних: ієрархічну, мережеву та реляційну [16].

### 3.5 Висновки до третього розділу

У третьому розділі було розглянуто архітектуру системи та її підсистем, що забезпечують функціонування всієї системи. Детально описано алгоритми обробки інформації, які використовуються для забезпечення ефективної роботи програми, а також проаналізовано технології, що застосовувалися при її реалізації.

Завдяки впровадженню алгоритмів обробки даних, система досягає високої точності та швидкості в опрацюванні запитів користувачів. Обрані технології дозволили забезпечити стабільність, масштабованість та гнучкість системи, що є важливим аспектом для її подальшого розвитку та підтримки.

Таким чином, розроблена система відповідає вимогам до сучасних інформаційних систем і має всі необхідні технічні характеристики для подальшої ефективної експлуатації в реальних умовах.

## РОЗДІЛ 4. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

### 4.1 Апаратні вимоги

Оскільки наша база даних розташована локально, то її апаратні вимоги будуть:

Процесор: Intel Core i5 3 Gb або вище;

Рекомендований обсяг оперативної пам'яті: 8 Gb або більше;

Жорсткий диск: наявність 20 Gb вільної дискової пам'яті;

Роздільна здатність екрану: Мін. 1366x768; Рек. 1920x1080;

Операційна система: Windows 8.1 (не нижче) Windows 10 [17].

### 4.2 Програмні вимоги

Для того, щоб встановити MySQL на локальному комп'ютері потрібно дотриматись таких системних вимог:

- 512 МБ оперативної пам'яті;
- Приблизно 500 МБ вільного дискового місця;
- Win2K, WinXP (x86/x64), Win2K3 (x86/x64), Vista (x86/x64), Win2K8 (x86/x64) і вище;
- Для проведення повноцінного адміністрування, заповнення таблиць, а також доступу до даних із середовища програмування Visual Studio нам знадобиться:
  - mysql Server (безпосередньо сам mysql)
  - MySQL /NET connector (для виконання запитів в Visual Studio);
  - MySQL for Visual Studio (компонента через яку Visual Studio може отримати доступ до структури даних);
  - MySQL-Front (для наповнення, створення, перегляду таблиць користувачів) [18].

### 4.3 Опис виконання дослідження

Під час проходження практики було здійснено дослідження на тему кваліфікаційної роботи та його завізуалізовано. Для візуалізації роботи було використано MS Excel. У якій було створено графічний застосунок, що демонструє бронювання (рис. 4.1), ціну (рис. 4.2), тривалість (рис.4.3) та двохкритеріальне значення.

	PriceCategory	DurationCategory	TotalBookings	Клас
▶	Low	Short	803	L
	Medium	Short	1618	L
	Medium	Long	3757	H
	Medium	Medium	2174	H
	Low	Medium	1098	L
	High	Short	801	L
	High	Medium	1007	L
	Low	Long	1835	H
	High	Long	1907	H
*				

Рис. 4.1 Представлення у MS Excel (Бронювання)

Клас розраховуємо наступним чином: якщо більше 1700 бронювань то клас High якщо менше то low.

	PriceCategory	H	L	Всього	Клас	Ймовірність	Помилка
▶	Low	1	2	3	L	66.66666666666666...	1 / 3
	Medium	2	1	3	H	66.66666666666666...	1 / 3
	High	1	2	3	L	66.66666666666666...	1 / 3
*							

Рис. 4.2 Представлення у MS Excel (Ціна)

Бачимо що всі три рівня ціни мають однакову ймовірність, тобто якщо ціна низька або висока то з ймовірністю 66% кількість бронювань буде низькою, а якщо середня то кількість бронювань буде в класі Н з ймовірністю теж 66%.

А для тривалості бачимо, що якщо тривалість на короткий термін, то кількість бронювань буде відноситись до класу L з ймовірністю 100%

DurationCategory	H	L	Всього	Клас	Ймовірність	Помилка
Short	0	3	3	L	100	0 / 3
Long	3	0	3	H	100	0 / 3
Medium	1	2	3	L	66.666666666666...	1 / 3

Рис. 4.3 Представлення у MS Excel (Тривалість)

Після того як я дослідив можливості вирішення задач Data Mining процесів у середовищі BI MS SQL Server та MS Excel, то мені необхідно було побудувати класи з використанням алгоритму Naïve Bayes, щоб розв'язати задачу класифікації.

Процес розв'язку задачі полягав в визначенні двохкритеріального значення, тобто, треба було визначити ціну та тривалість їхню ймовірність, а також привести у спільній таблиці ймовірність на основі попередньо визначених значень (рис. 4.4).

PriceCategory	Ймовірність H	Ймовірність L
Low	33.333333333333...	66.666666666666...
Medium	66.666666666666...	33.333333333333...
High	33.333333333333...	66.666666666666...
	100	

DurationCategory	Ймовірність H	Ймовірність L
Short	0	100
Long	100	0
Medium	33.333333333333...	66.666666666666...
	100	

PriceCategory	DurationCategory	Ймовірність H	Ймовірність L
Low	Short	0	1
Low	Long	1	0
Low	Medium	0.2	0.8
Medium	Short	0	1

Рис. 4.4 Двохкритеріальне значення

На основі отриманого значення можна зробити висновок, що якщо тривалість бронювання коротка і ціна низька то з ймовірністю 100% кількість бронювань буде знаходитись в класі Low (низька кількість бронювань).

Якщо тривалість Long і ціна низька то ймовірність що кількість бронювань буде висока – 100%.

Для наочного подання побудовано в Microsoft Analysis Service діаграму (рис. 4.5).



Рис. 4.5 Побудова діаграми у Microsoft Analysis Service

Таким чином, якщо ціна висока і тривалість довга, то з ймовірністю 100% кількість бронювань буде більше 16 (середня кількість) що співпадає з результатами що ми отримали в програмі (рис. 4.6).

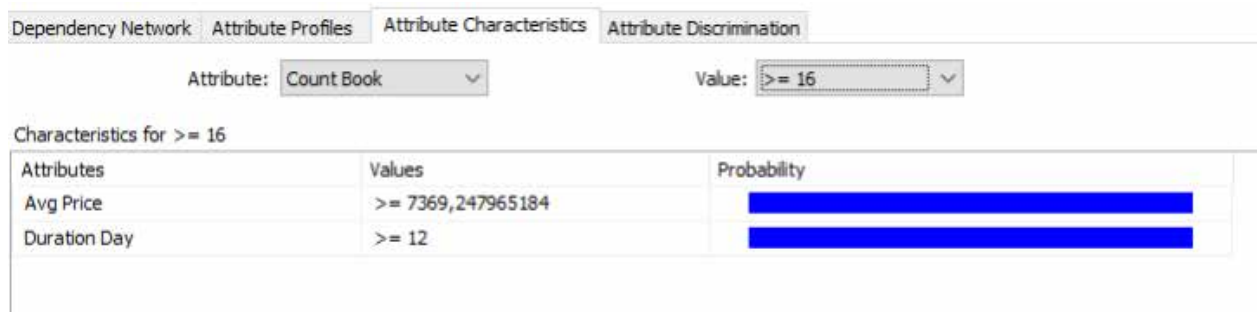


Рис. 4.6 – Отримані значення у програмі

І тепер нам необхідно здійснити пошук асоціативних правил, тобто, для реалізації пошуку, ми відберемо відповідні параметри у SSAS (рис. 4.7).

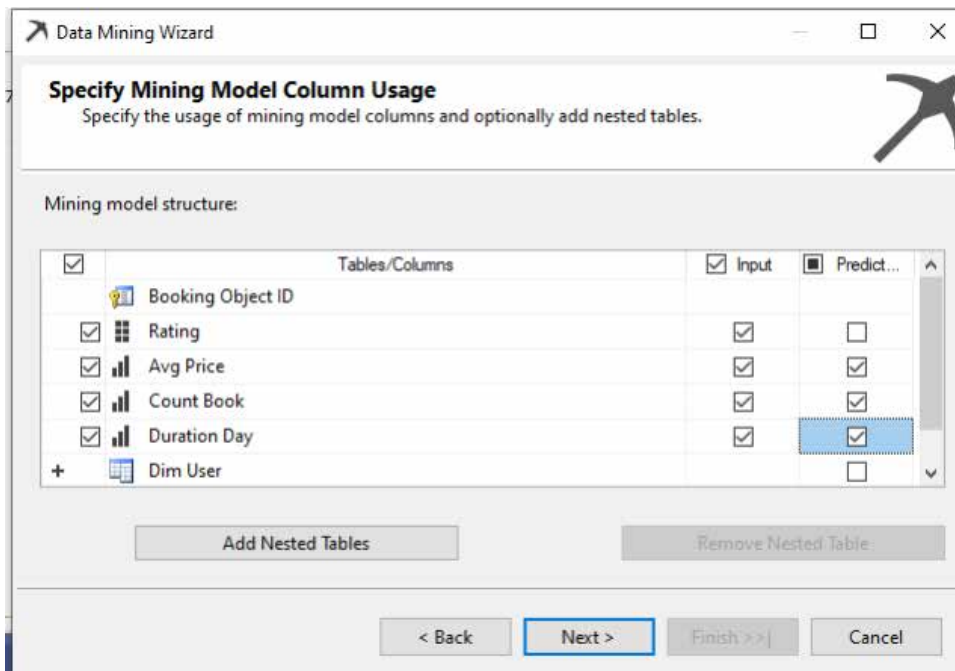


Рис. 4.7 Відбір параметрів

Після того, як ми відібрали параметри, ми погодимося та побудуємо асоціативні правила, що включають всі елементи, що зберігаються у вибраних параметрах (рис.4.8-4.9).



Рис. 4.8 Асоціативні правила для вибраних параметрів (графічне подання)

Pr...	Importance	Rule
1.000	-0.058	Rating = 3,23 - 3,85, 5 = Чоловік -> 9 >= 7369,247965184
1.000	-0.118	Rating = 3,85 - 4,07 -> 9 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 8 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 7 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 6 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 5 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 4 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 3 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 2 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 1 >= 13
1.000	-0.118	Rating = 3,85 - 4,07 -> 10 >= 9
1.000	-0.118	Rating = 3,85 - 4,07 -> 9 >= 9
1.000	-0.118	Rating = 3,85 - 4,07 -> 8 >= 9
1.000	-0.118	Rating = 3,85 - 4,07 -> 7 >= 9

Рис. 4.9 Асоціативні правила для вибраних параметрів (подання у вигляді правил)

Таким чином, по першому правилу бачимо що якщо рейтинг готелю від 3,32 до 3,85 і бронювали чоловіки, то ціна бронювання на 9 днів буде більшою за 7369 грн. Важливість цього правила -0.058.

Далі отримаємо графік точності наших асоціативних правил (рис. 4.10).

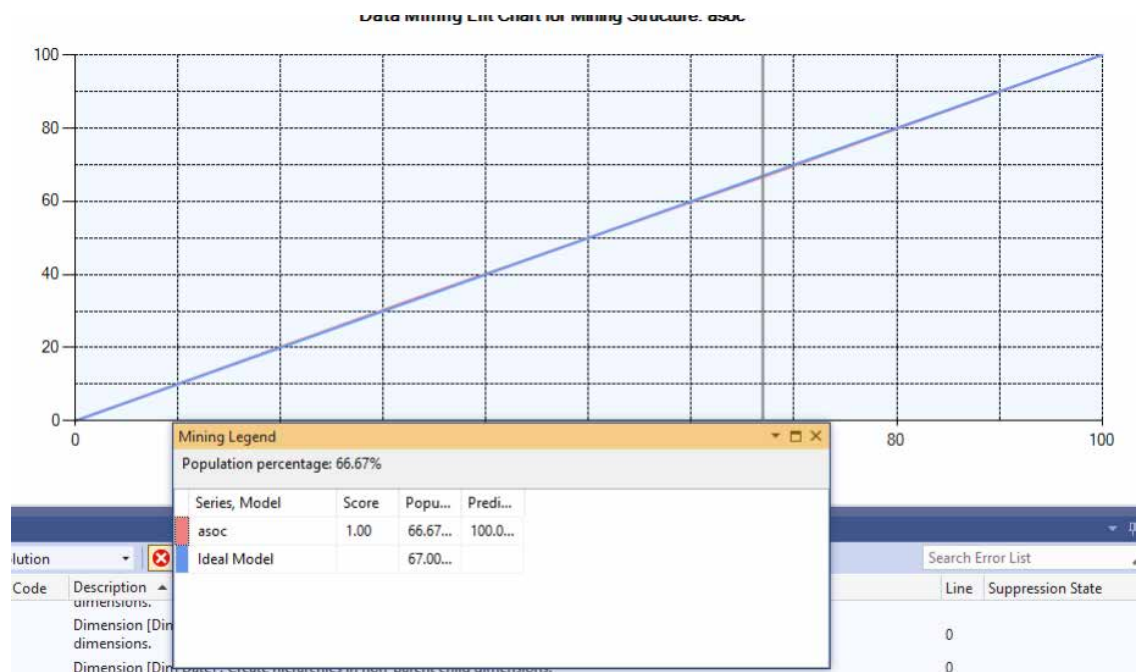


Рис. 4.10 Графік точності асоціативних правил

Як видно з отриманого повідомлення на рис 4.10, то наша модель є майже ідеальною для передбачення правил.

Наприклад, розглянемо таку модель, що у нас є чотири кластери, що складають середній вік чоловіків, середній вік жінок, відхилення у віці для чоловіків та жінок, та середній вік чоловіків та жінок. Візуально представимо це на рис. 4.11. І в цій же моделі врахуємо, що в нас є втрачені дані, тобто, винесемо цей п'ятий кластер окремо.

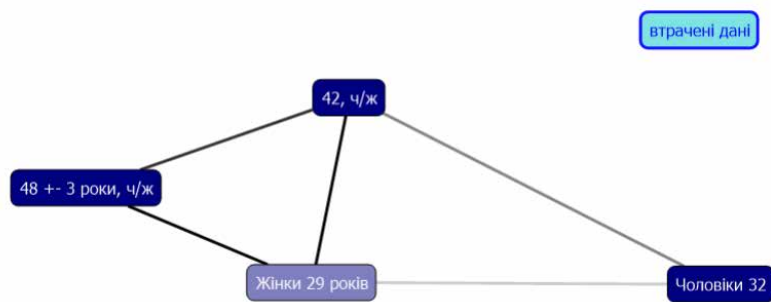


Рис. 4.11 Модель середнього віку

На основі цих даних побудуємо гістограму, що включає в себе: вік, середнє значення бронювання, середню тривалість, середню ціну та стать (рис. 4.12).

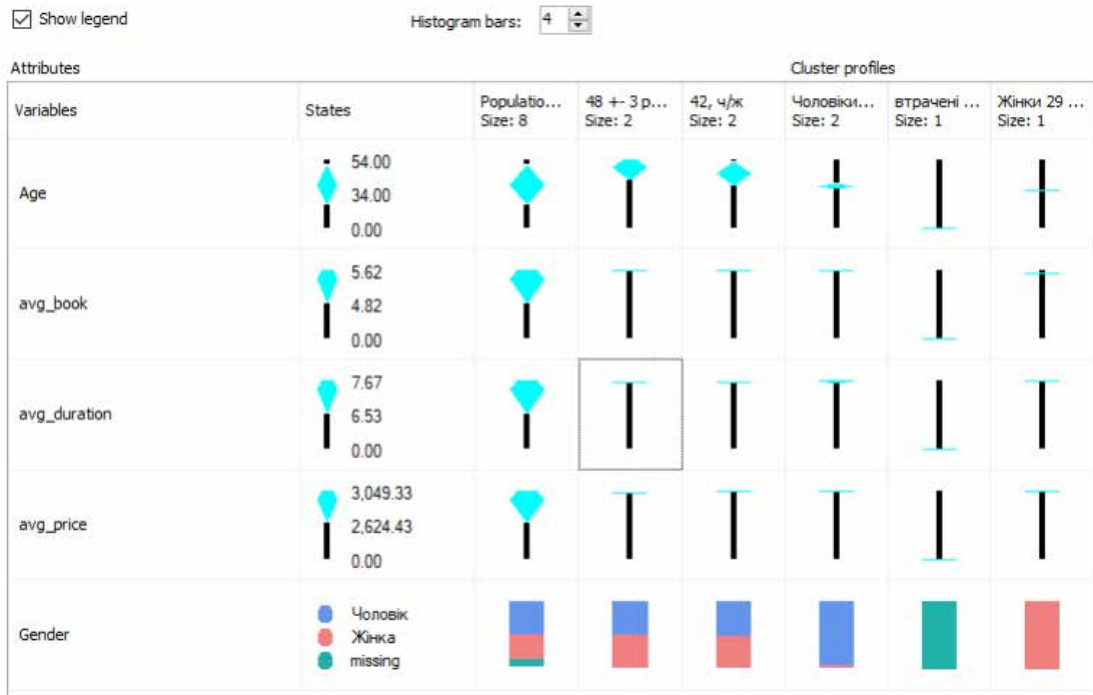


Рис. 4.12 Визначення параметрів гістограми для побудови

Та безпосередньо представлення у вигляді гістограми, що у свою чергу поділене на кластери (рис. 4.13).

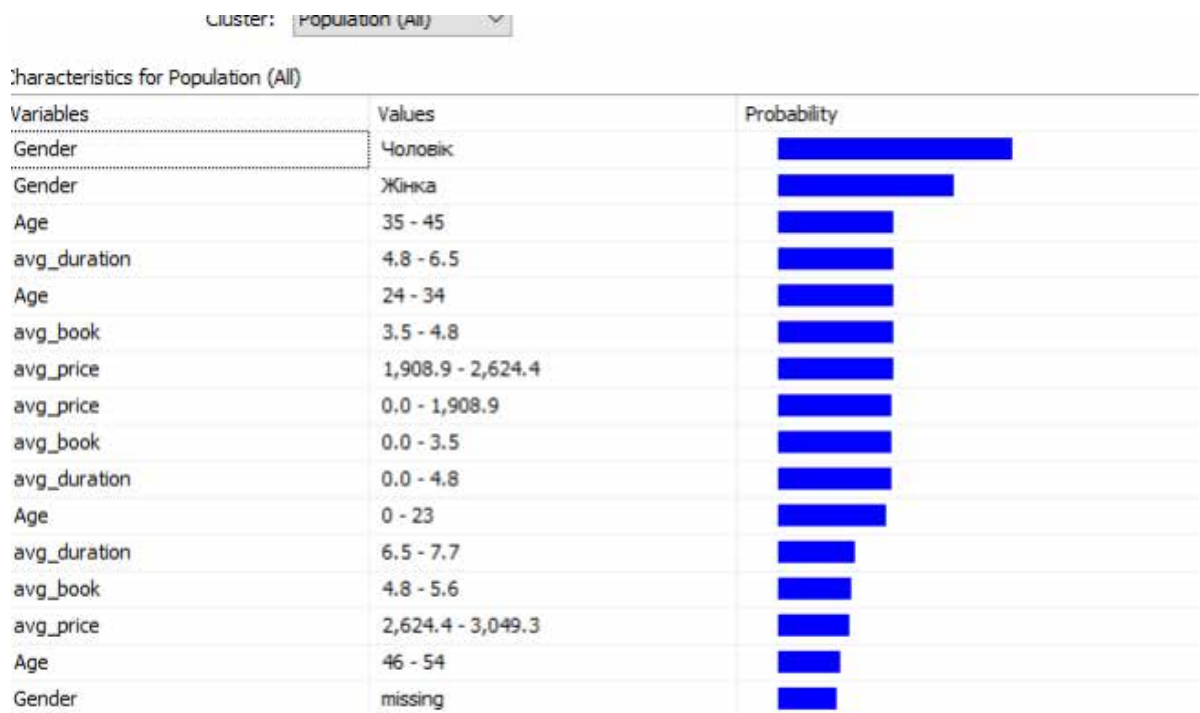


Рис. 4.13 Представлення у вигляді гістограми

Кластер 2 (Чоловіки 32): У цьому кластері переважають чоловіки, вік яких становить приблизно 32 роки. В середньому вони роблять більше бронювань, але середня тривалість і ціна бронювань у цього кластера є середніми.

Кластер 3 (Жінки 29 років): Цей кластер складається в основному з жінок, вік яких становить приблизно 29 років. Вони також роблять менше бронювань, але середня тривалість і ціна бронювань у цього кластера є середніми.

Кластер 4 (42, ч/ж): Цей кластер має середній вік приблизно 42 роки і збалансований розподіл між чоловіками та жінками. Вони роблять менше бронювань, але середня тривалість і ціна бронювань у цього кластера є середніми.

Кластер 5 (48 +/- 3 роки, ч/ж): Цей кластер також має середній вік приблизно 48 років і збалансований розподіл між чоловіками та жінками. Вони

роблять більше бронювань, але середня тривалість і ціна бронювань у цього кластера є середніми.

В рамках розробки дипломної роботи, то було спроектовано базу даних з наступною структурою (рис. 4.14).

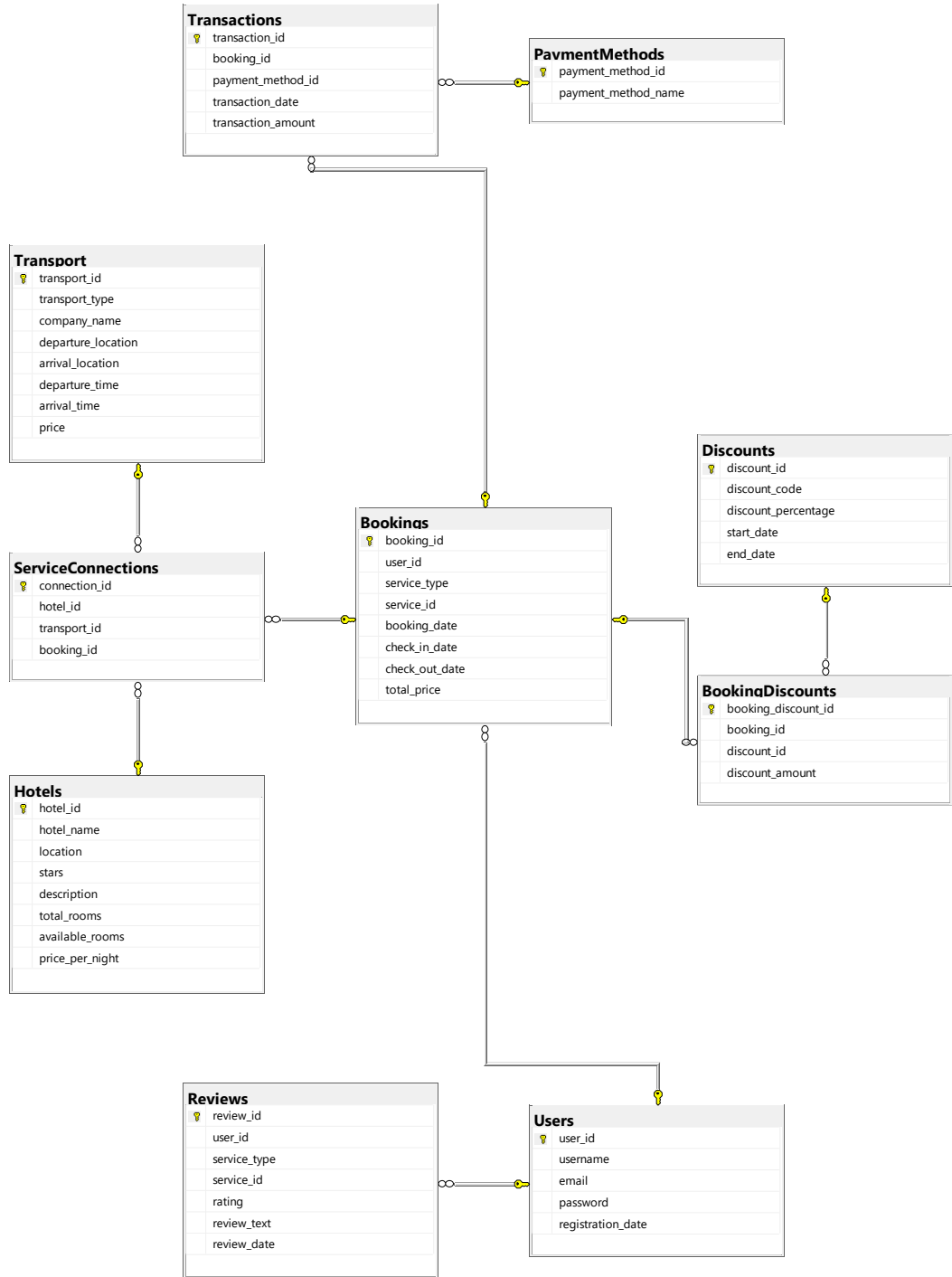
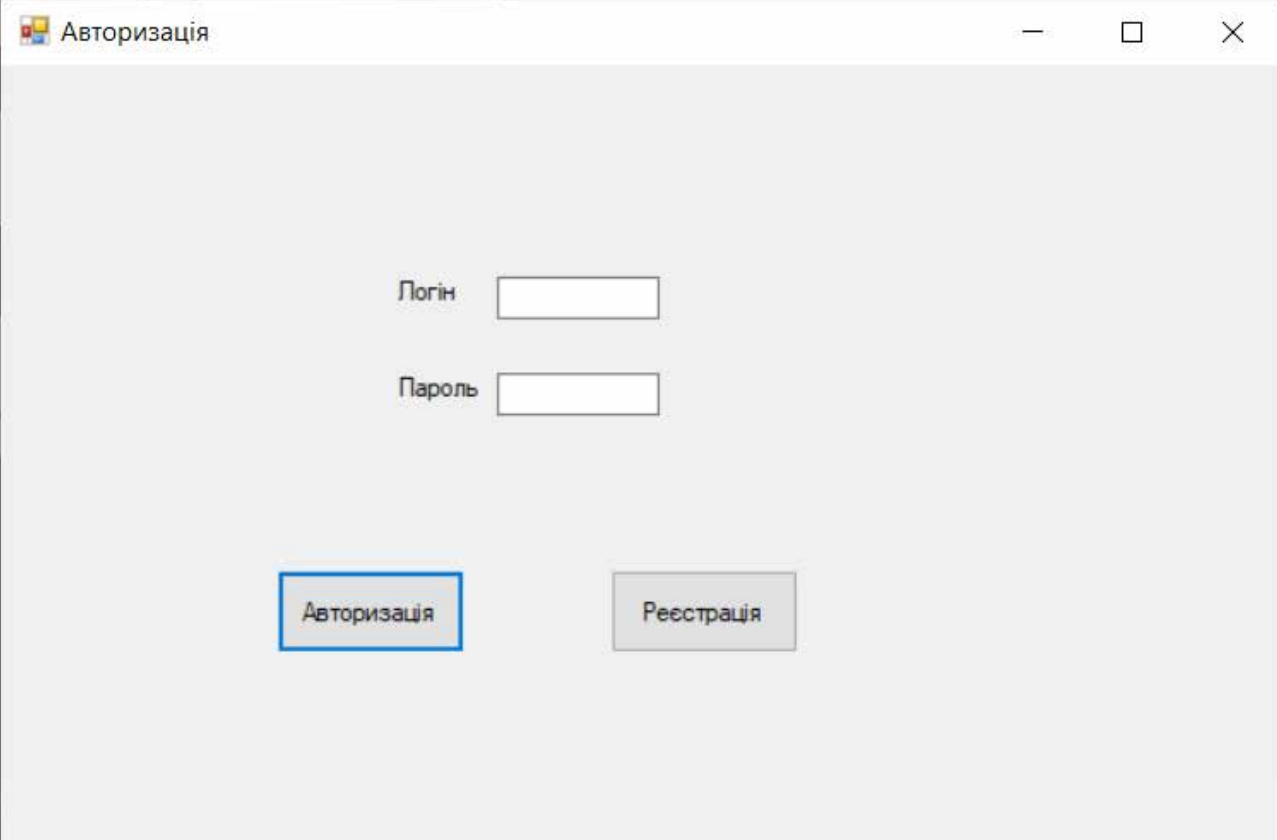


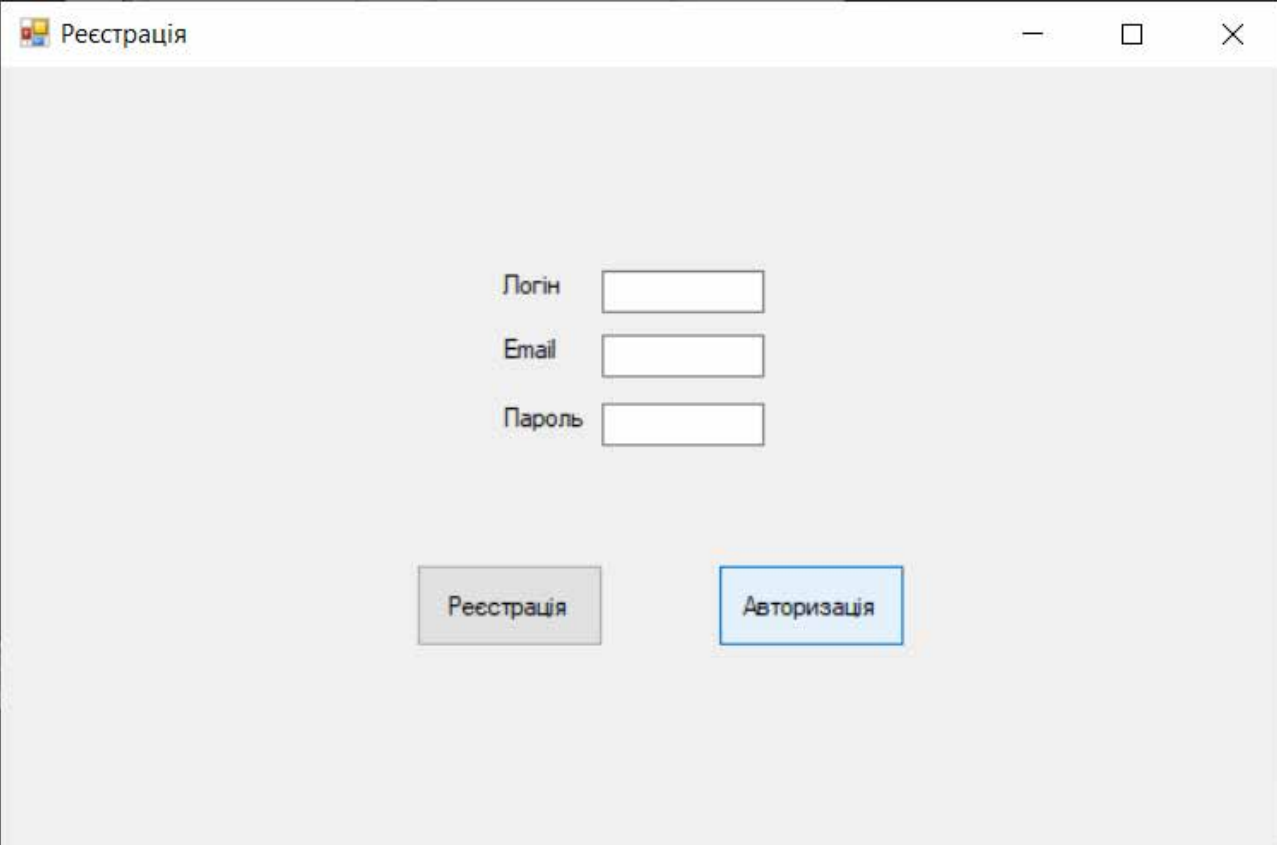
Рисунок 4.14 Структура бази даних

Та розроблено програмний застосунок, який дозволяє працювати з експертною системою. Розглянемо застосунок. Форма авторизації та реєстрації наведені на рис. 4.15-4.16.



The image shows a screenshot of a software application window titled "Авторизація" (Authorization). The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains two input fields: "Логін" (Login) and "Пароль" (Password). Below the input fields are two buttons: "Авторизація" (Authorization) and "Реєстрація" (Registration). The "Авторизація" button is highlighted with a blue border, while the "Реєстрація" button is gray.

Рисунок 4.15 Авторизація



The image shows a window titled "Реєстрація" (Registration) with a standard Windows-style title bar. Inside the window, there are three input fields stacked vertically, labeled "Логін" (Login), "Email", and "Пароль" (Password). Below these fields are two buttons: "Реєстрація" (Registration) on the left and "Авторизація" (Authorization) on the right. The "Авторизація" button is highlighted with a blue border.

Рис. 4.16 Реєстрація

У разі успішної авторизації побачимо наступне повідомлення (рис. 4.17).

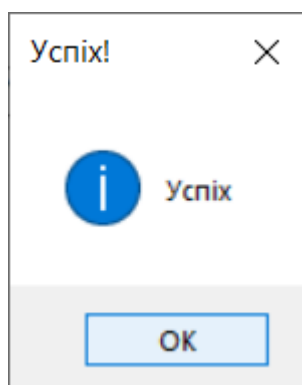


Рис. 4.17 Повідомлення про успіх

Головне меню програми наведено на рис. 4.18.

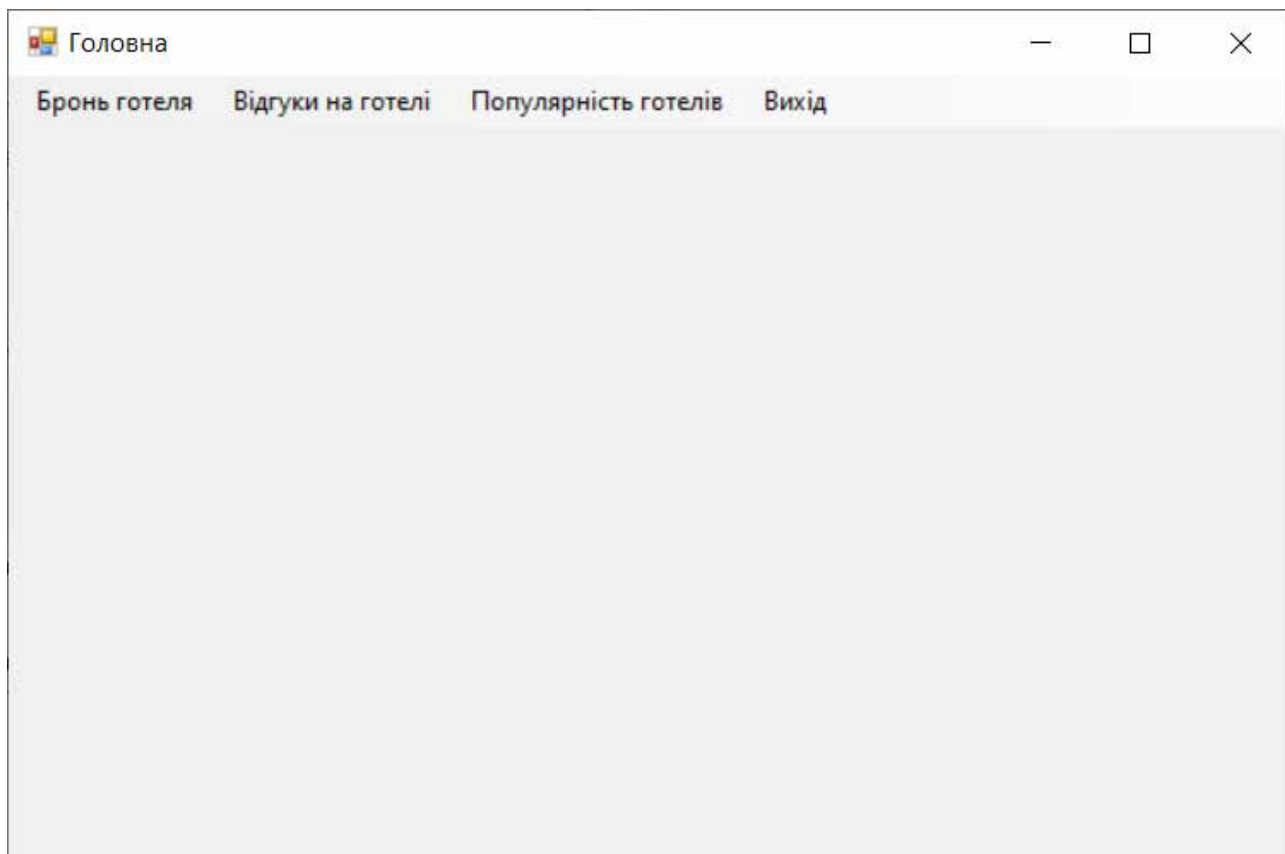


Рис. 4.18 Меню програми

Переглянемо форму «Бронь готеля» (рис. 4.19). Натиснувши на рядок, то дані про готель перенесуться у текстові поля (рис. 4.20). У випадаючих списках можна переглянути, всю доступну інформацію стосовно певного рядка (рис.4.21). У одному з випадаючих полів зберігається інформація стосовно користувачів бази даних, і щоб забронювати номер, то треба обрати себе у цьому списку (рис. 4.22).

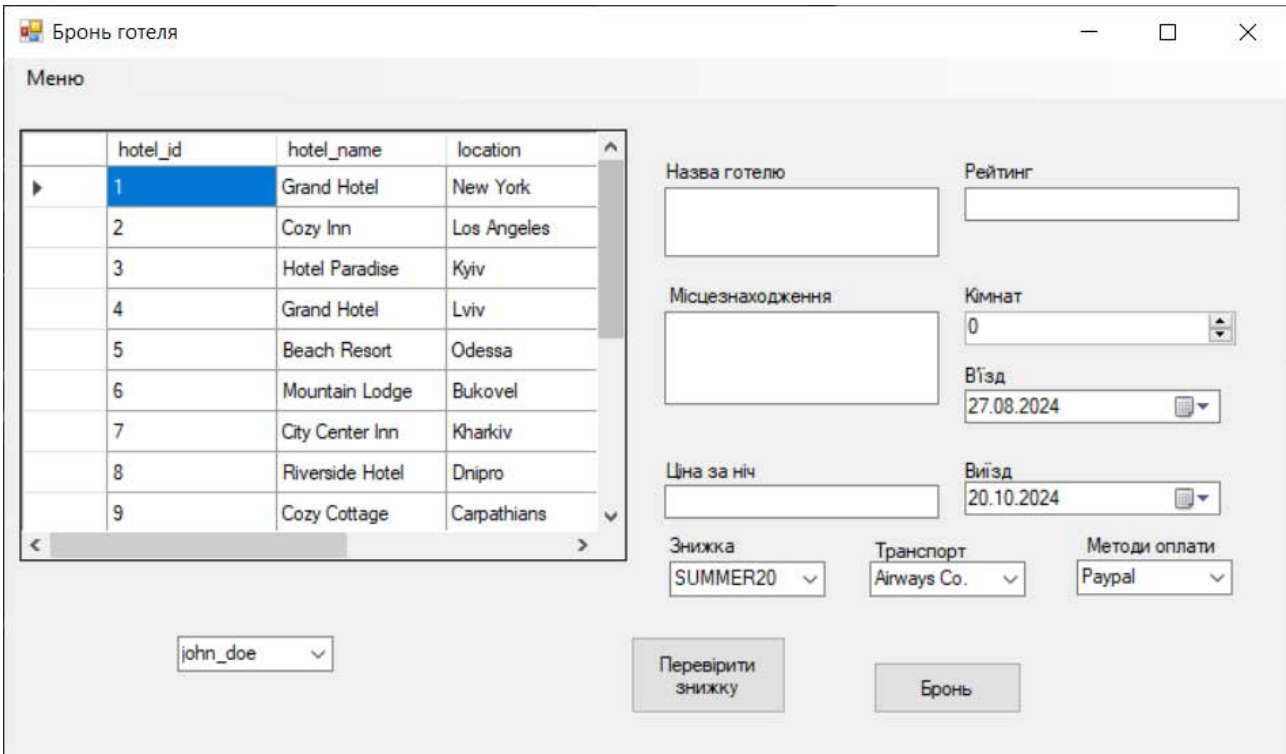


Рис. 4.19 Бронь готеля

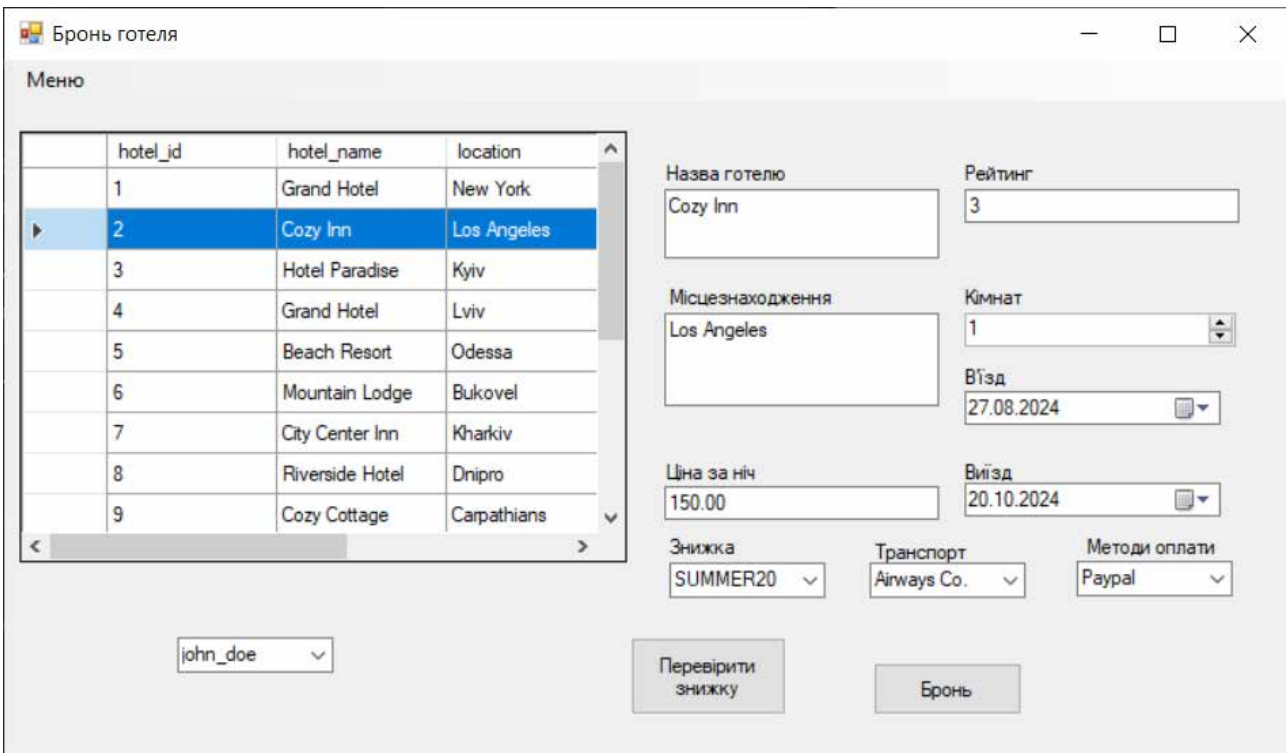


Рис. 4.20 Обрана інформація про готель

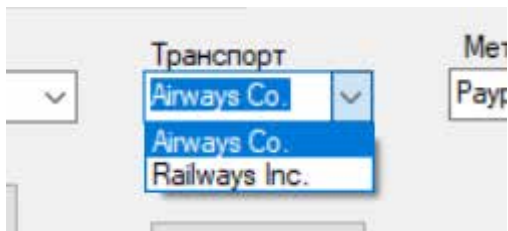


Рис. 4.20 Випадаюче поле з інформацією

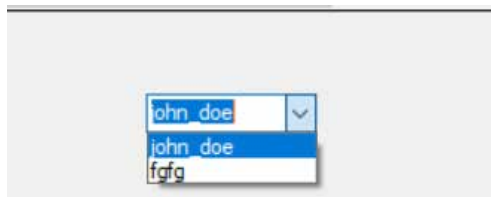


Рис. 4.21 Випадаюче поле з інформацією про користувачів

Якщо обрати певний діапазон дат, під які не підпадає знижка, то отримаємо наступне повідомлення (рис. 4.22).

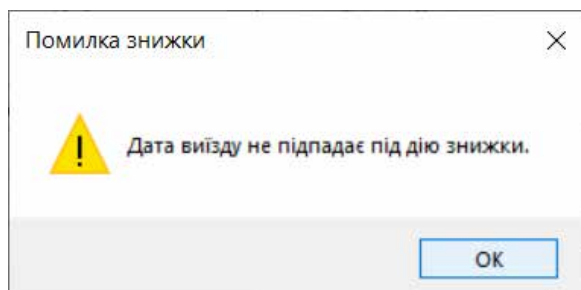


Рис. 4.22 Помилка знижки

Якщо здійснити бронювання і воно буде успішним, то з'явиться наступне повідомлення (рис. 4.23).

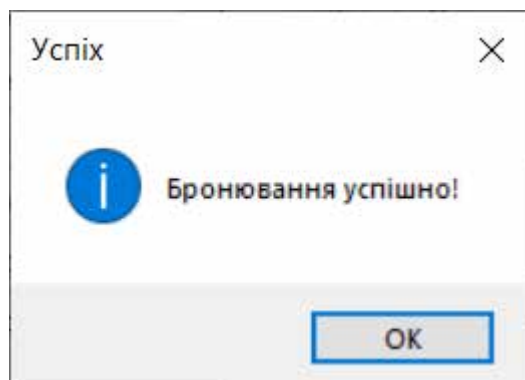
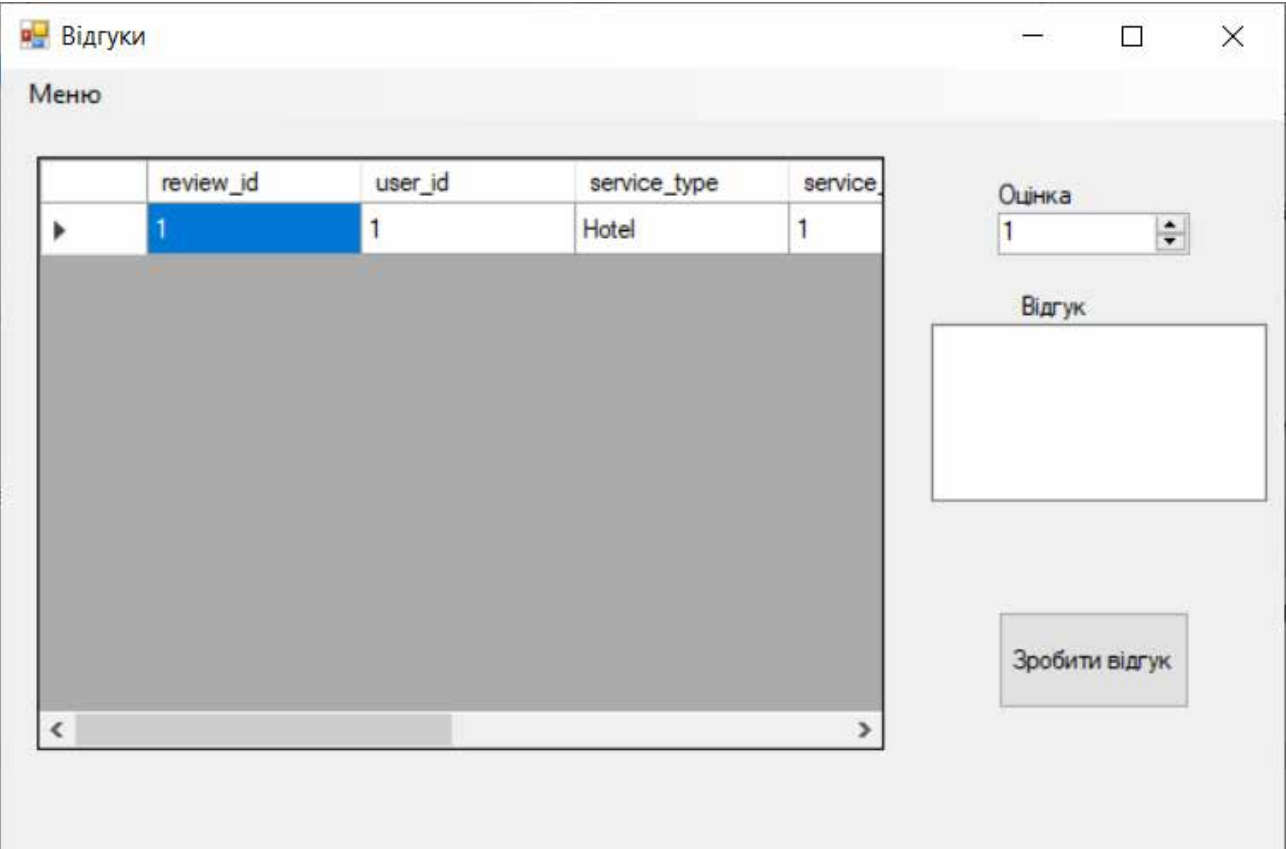


Рис. 4.23 – Успіх при бронюванні

Для того, щоб залишити відгук на готель, то треба поставити йому оцінку та увести сам відгук (рис.4.24).



	review_id	user_id	service_type	service
▶	1	1	Hotel	1

Оцінка  
1

Відгук

Зробити відгук

Рис. 4.24 Відгуки на готелі

Інфографіка популярності готелів наведена на рис. 4.25. Популярність готелів визначається шляхом найчастішого бронювання.

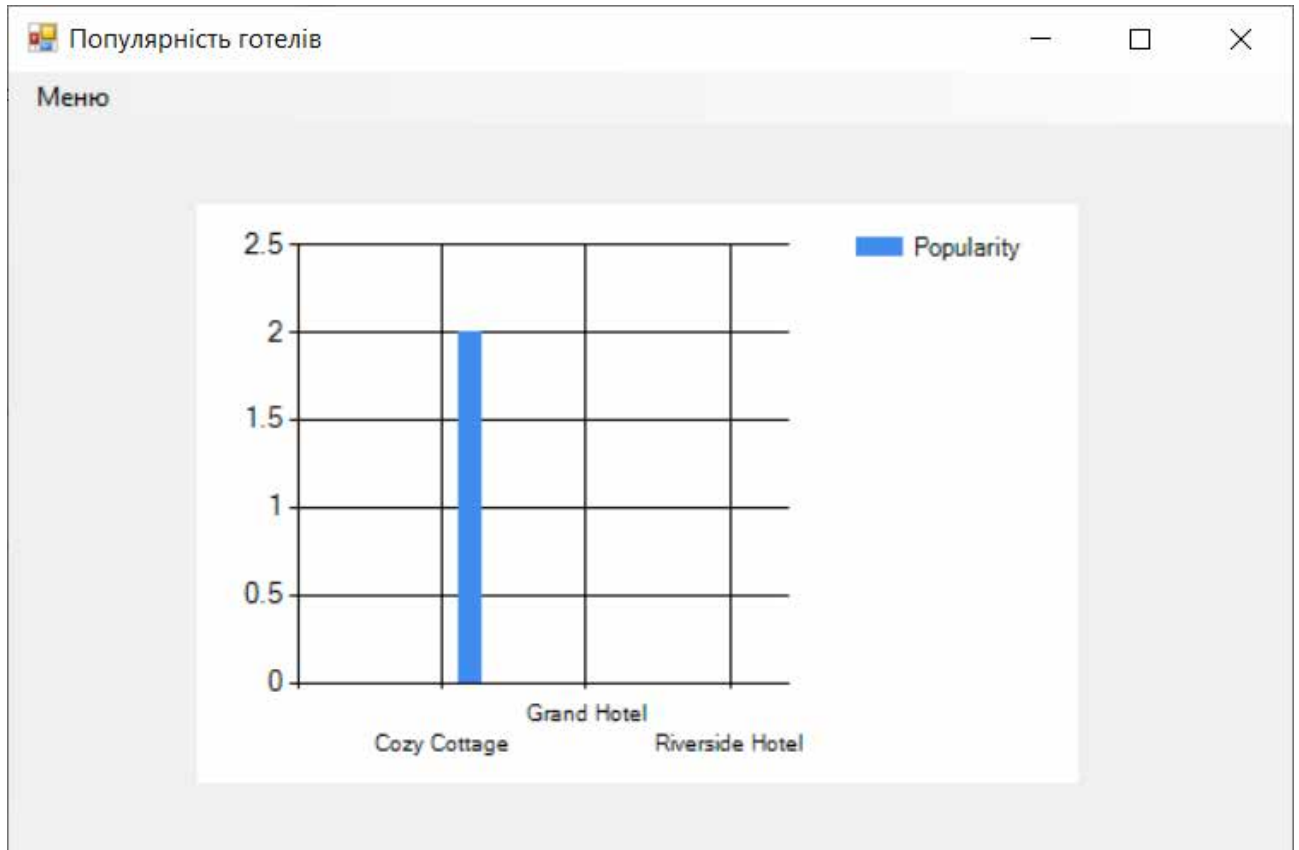


Рис. 4.25 Популярність готелів

#### 4.4 Аналіз отриманих результатів

Аналіз отриманих результатів демонструє успішне застосування різних методів обробки даних і візуалізації для вирішення завдань, поставлених у рамках кваліфікаційної роботи. Під час дослідження було розглянуто можливості системи MS Excel для побудови графічних застосунків, які візуалізують кількість бронювань, ціни та тривалість перебування в готелях. Це дозволило виявити певні закономірності, такі як ймовірність низької кількості бронювань за низької ціни або короткої тривалості перебування.

Крім того, побудова двокритеріальних моделей дозволила оцінити ймовірності різних сценаріїв, таких як залежність кількості бронювань від ціни та тривалості. Модель на основі алгоритму Naïve Bayes підтвердила ці

закономірності та надала чітке розуміння, як різні фактори впливають на попит на готелі.

Використання Microsoft Analysis Services дозволило поглибити аналіз за допомогою асоціативних правил. Це дало змогу знайти залежності між характеристиками бронювань і цінами, наприклад, визначення правил для різних типів клієнтів, що використовують послуги готелів. Отримані асоціативні правила також підкреслили важливість факторів, таких як стать і вік клієнтів.

Усі отримані результати дозволяють робити висновки про ефективність розроблених моделей і використовуваних інструментів. Система, розроблена для бронювання готелів, показала високу точність прогнозування та здатність до адаптації під різні умови.

#### 4.5 Висновки до четвертого розділу

У даному розділі було визначено необхідні апаратні та програмні ресурси для реалізації системи бронювання готелів, що забезпечує високу продуктивність і стабільність роботи.

Проведене дослідження демонструє ефективність використання методів візуалізації даних та алгоритмів машинного навчання для аналізу та класифікації інформації про бронювання. Використання MS Excel та Microsoft Analysis Services дозволило отримати чіткі результати, які легко інтерпретувати.

Зроблено висновок про суттєвий вплив ціни та тривалості перебування на кількість бронювань. Результати дослідження підтверджують наявність чіткої залежності між цими факторами, що важливо для оптимізації цінової політики готелів.

Розроблені моделі на основі алгоритму Naïve Bayes показали високу точність прогнозування кількості бронювань, а також дозволили виявити ключові фактори, що впливають на рішення клієнтів.

Результати пошуку асоціативних правил підтвердили, що певні характеристики клієнтів (стать, вік) суттєво впливають на поведінку щодо бронювання, що може бути використано для цілеспрямованого маркетингу.

В майбутньому планується розширення системи шляхом інтеграції нових функцій, які забезпечать ще більш детальний аналіз даних і можливість прогнозування попиту на основі історичних даних.

## ВИСНОВКИ

Проведене дослідження автоматизації готельних послуг продемонструвало важливість системного підходу до розробки та впровадження ефективних інформаційних систем, здатних відповідати сучасним вимогам ринку. У результаті аналізу та моделювання, проведеного в межах роботи, були досягнуті такі результати:

У першому розділі було детально проаналізовано предметну область автоматизації готельних послуг, що дало змогу виявити ключові проблеми, які потребують вирішення. Визначено, що якість наданих послуг і ефективність процесів бронювання є критичними для залучення нових клієнтів і підтримки конкурентоспроможності на ринку. У зв'язку з цим, обґрунтовано необхідність розробки нової системи, яка враховувала б наявні проблеми та відповідала сучасним вимогам користувачів.

Другий розділ зосередився на процесі моделювання системи, зокрема на порівнянні функціонального та об'єктно-орієнтованого підходів. Функціональне моделювання, через простоту візуалізації функцій системи, виявилось ефективним для початкового аналізу бізнес-процесів. Натомість об'єктно-орієнтоване моделювання надало гнучкість та можливість повторного використання коду, що є важливим аспектом для сучасної розробки програмного забезпечення. Порівняння цих підходів показало, що комбінація обох методів може забезпечити оптимальні результати у розробці складних систем.

У третьому розділі було детально розглянуто архітектуру розробленої системи та її підсистеми, що забезпечують стабільну роботу всього рішення. Обрані алгоритми обробки інформації забезпечили високу швидкість і точність в обробці запитів, що є критично важливим для готельного бізнесу. Вибір технологій, які підтримують масштабованість та гнучкість системи, забезпечив можливість подальшого розвитку та вдосконалення програми.

Четвертий розділ зосередився на аналізі результатів, отриманих внаслідок дослідження. Використання методів візуалізації даних та алгоритмів машинного

навчання, зокрема Naïve Bayes, дозволило не лише точно прогнозувати кількість бронювань, але й виявити ключові фактори, що впливають на прийняття рішень клієнтами. Це дозволяє готелям оптимізувати свої ціни та адаптувати пропозиції до потреб різних сегментів споживачів. Результати також показали, що характеристики клієнтів, такі як стать і вік, істотно впливають на поведінку щодо бронювання, що відкриває нові можливості для цілеспрямованого маркетингу.

Отримані результати вказують на потенціал для подальшого розвитку системи. Планується інтеграція нових функцій, які дозволять здійснювати ще більш детальний аналіз даних та прогнозування попиту на основі історичних даних. Це допоможе адаптувати систему до змінних умов ринку, зокрема до сезонних коливань попиту.

У підсумку, результати дослідження свідчать про доцільність і необхідність впровадження сучасних технологій в сфері готельних послуг. Розроблена система відповідає вимогам сучасних інформаційних систем і має всі необхідні технічні характеристики для ефективної експлуатації в реальних умовах. Вона не лише підвищує ефективність роботи готелів, але й сприяє підвищенню якості обслуговування клієнтів, що, в свою чергу, позитивно впливає на їх задоволеність та лояльність. В подальшому, система може стати основою для створення комплексних рішень в сфері управління готельним бізнесом, що сприятиме успішному розвитку даної галузі.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Просович О.П., Боцман Ю.С. Маркетплейс як дієвий інструмент цифрового маркетингу. Вісник Національного університету “Львівська політехніка”. Серія: Проблеми економіки та управління. 2018. № 897(2). С. 32–38.
2. Грабар М.В. Інформаційні системи та технології на туристичному ринку: сучасність та перспективи. Інфраструктура ринку. 2019. №39 URL: <http://www.market-infr.od.ua/uk/archive> (дата звернення: 07.10.2024).
3. Gurskaya S.P. (2017) Marketpleysi – novyyi segment v e-commerce. [Marketplace – a new segment in e-commerce] Problemy i perspektivy elektronnoho biznesa : sbornik nauchnyih statey mezhdunarodnoy nauchno-prakticheskoy konferentsii (pp. 26–30).
4. Booking.com. URL: [https://www.booking.com/index.uk.html?label=gen173nr-1BCAEoggI46AdIM1gEaOkBiAEBmAEPuAEXyAEM2AEB6AEBiAIBqAIDuAKFw9S4BsACAdICJGVmMDcyNGRiLTI4MTktNDAwOC1hY2NmLTUwYmM2YjRjYzA4ZNgCBeACAQ&sid=f5125f0860ad4761f31b88453bdbcbe4&keep\\_landing=1&sb\\_price\\_type=total&](https://www.booking.com/index.uk.html?label=gen173nr-1BCAEoggI46AdIM1gEaOkBiAEBmAEPuAEXyAEM2AEB6AEBiAIBqAIDuAKFw9S4BsACAdICJGVmMDcyNGRiLTI4MTktNDAwOC1hY2NmLTUwYmM2YjRjYzA4ZNgCBeACAQ&sid=f5125f0860ad4761f31b88453bdbcbe4&keep_landing=1&sb_price_type=total&) (дата звернення: 07.10.2024).
5. Booking Holdings Inc. Annual Report on Form 10-K for the Year Ended December 31, 2023 pp. 236
6. Система управління готелями та ресторанами FIDELIO “HOTEL & RESTAURANT SYSTEM (HRS)”. URL: <https://repository.ldufk.edu.ua/bitstream/34606048/10657/1/%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%8F%2015.pdf> (дата звернення: 07.10.2024).
7. Казарінова А.О. Автоматизація управління діяльністю готелю. URL: [https://tourlib.net/statti\\_ukr/kazarinova.htm](https://tourlib.net/statti_ukr/kazarinova.htm) (дата звернення: 08.10.2024).
8. Функціональне моделювання. Методології функціонального моделювання. URL:

<https://moodle.gi.edu.ua/mod/book/tool/print/index.php?id=36262&chapterid=571>

(дата звернення: 08.10.2024).

9. Об'єктно-орієнтоване моделювання. URL: [https://www.wikidata.uk-ua.nina.az/%D0%9E%D0%B1%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5\\_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F.html](https://www.wikidata.uk-ua.nina.az/%D0%9E%D0%B1%D1%94%D0%BA%D1%82%D0%BD%D0%BE-%D0%BE%D1%80%D1%96%D1%94%D0%BD%D1%82%D0%BE%D0%B2%D0%B0%D0%BD%D0%B5_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F.html) (дата звернення: 08.10.2024).

10. Методології моделювання. Сімейство IDEF. URL: [https://elib.lntu.edu.ua/sites/default/files/elib\\_upload/%D0%9A%D0%BE%D0%BD%D0%B4%D1%96%D1%83%D1%81%20%20%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%B0/page5.html](https://elib.lntu.edu.ua/sites/default/files/elib_upload/%D0%9A%D0%BE%D0%BD%D0%B4%D1%96%D1%83%D1%81%20%20%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%B0/page5.html) (дата звернення: 08.10.2024).

11. Висоцька В.А. Об'єктно-орієнтоване моделювання та проектування систем електронної комерції: Методичні вказівки до лабораторної роботи № 4. – Львів: Видавництво Національного університету "Львівська політехніка", 2017. – 113 с.

12. Омельчук Л.Л. Об'єктно-орієнтоване програмування. Лабораторний практикум: навчальний посібник / Л.Л. Омельчук, А.С. Белова. – Київ: 2022. - 271 с.

13. Роберт С. Мартін. Чиста архітектура. – Фабула. 2019 – 368 с.

14. Дібрівний О.А., Гребенюк В.В. Вступ до об'єктно орієнтованого програмування С#: Навчальний посібник. – К.: Державний університет телекомунікацій, 2018, - 190 с.

15. Балик Н.Р., Мандзюк В.І. Бази даних MySQL: Навчальний посібник. — Тернопіль: Навчальна книга – Богдан, 2010.— 160 с.

16. Лосєв М. Ю. Бази даних : навчально-практичний посібник для самостійної роботи студентів. – Харків : ХНЕУ ім. С. Кузнеця, 2018. – 233 с.

17. Технічні та програмні вимоги. URL: <https://ispro.ua/uploads/usermanuals/July2022/UWNP21BLQlvRBB7l93Dt.pdf> (дата звернення: 08.10.2024).

18. Лекція 4. Семестр 1. Налаштування середовища MySQL на локальному комп'ютері. URL: <https://www.e-helper.com.ua/node/931> (дата звернення: 08.10.2024).