ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра Інформаційних систем та технологій

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ІНФОРМАЦІЙНА СИСТЕМА БРОНЮВАННЯ ГОТЕЛЬНИХ НОМЕРІВ З ВИКОРИСТАННЯМ МОВИ ПРОГРАМУВАННЯ PYTHON»

на здобуття освітнього ступеня бакалавр

зі спеціальності 126 Інформаційні системи та технології

освітньо-професійної програми Інформаційні системи та технології

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ Станіслав РОМАНЧУК
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Виконав: здобувач вищої освіти
групи ІСД - 41
Станіслав РОМАНЧУК

Керівник: Оксана ТКАЛЕНКО

Рецензент: _____

Київ – 2026

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інформаційних систем та технологій
Ступінь вищої освіти бакалавр
Спеціальність 126 Інформаційні системи та технології
Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІСТ
_____Каміла СТОРЧАК

“____” _____2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Романчук Станіслав Дмитрович
(прізвище, ім’я, по батькові здобувача)

1.Тема кваліфікаційної роботи: Інформаційна система бронювання готельних номерів з використанням мови програмування Python
керівник кваліфікаційної роботи: Оксана ТКАЛЕНКО к.т.н., доцент
(ім’я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від “ 20 ” лютого 2026 р. № 51

2. Строк подання кваліфікаційної роботи «01» червня 2026 р.

3. Вихідні дані кваліфікаційної роботи:

1. Теоретичні матеріали щодо автоматизації процесів бронювання готельних номерів.
2. Мова програмування Python для розробки програмного забезпечення.
3. Графічна бібліотека Tkinter для створення інтерфейсу користувача.
4. Система керування базами даних SQLite для збереження інформації.
5. Методи та засоби проєктування інформаційних систем.
6. Технології створення desktop-додатків.
7. Засоби візуалізації даних та графічного представлення номерного фонду.
8. Матеріали щодо організації обліку номерів і реєстрації клієнтів.

9. Методи реалізації та контролю бронювання готельних номерів.
10. Засоби формування аналітичної інформації та експорту даних у формат CSV.

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

1. Аналіз основних можливостей та технологій іноформаційних систем бронювання
2. Проектування інформаційної системи
3.Прорграмна реалізація та тестування інформаційної системи.

5.Перелік ілюстраційного матеріалу: *презентація*

6. Дата видачі завдання «20» лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Підбір технічної літератури	24.02-03.03.26	
2.	Аналіз основних можливостей та технологій інформаційних систем бронювання	04.03-17.03.26	
3.	Проектування інформаційної системи	18.03-31.03.26	
4.	Прорграмна реалізація та тестування інформаційної системи	01.04-24.04.26	
5.	Висновки по роботі	25.04-16.05.26	
6.	Розробка демонстраційних матеріалів, доповідь.	19.05-20.05.26	
7.	Оформлення бакалаврської роботи	21.05-30.05.26	

Здобувач вищої освіти _____ Станіслав РОМАНЧУК
(підпис) *(ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Оксана ТКАЛЕНКО
(підпис) *(ім'я, ПРІЗВИЩЕ)*

ЗМІСТ

ВСТУП

Особливо важливою є автоматизація процесу бронювання готельних номерів, оскільки саме цей процес є одним із центральних у діяльності будь-якого готелю. Він охоплює перевірку наявності вільних номерів, вибір категорії номера, фіксацію дат заїзду та виїзду, внесення даних клієнта, розрахунок вартості проживання, зміну статусу бронювання та формування інформації для подальшої роботи адміністратора. Якщо ці операції виконуються вручну, зростає ризик помилок, а час обробки заявки збільшується. Тому створення інформаційної системи, яка дозволяє автоматизувати ці дії, є актуальним і практично значущим завданням.

У межах цієї дипломної роботи розробляється інформаційна система бронювання готельних номерів з використанням мови програмування Python. Система реалізована як desktop-додаток, що працює з локальною базою даних SQLite та має графічний інтерфейс користувача, створений засобами Tkinter. Такий підхід є доцільним для навчального проєкту, оскільки дозволяє продемонструвати повний цикл роботи прикладної інформаційної системи: від проєктування структури бази даних і функціональних модулів до реалізації інтерфейсу, алгоритму перевірки доступності номерів, створення бронювань, скасування записів, експорту даних та формування аналітичних показників.

Метою дипломної роботи є проєктування та програмна реалізація інформаційної системи бронювання готельних номерів з використанням мови програмування Python, яка забезпечує облік номерного фонду, роботу з клієнтами, створення та скасування бронювань, перевірку доступності номерів, візуальне відображення карти номерів і формування базових аналітичних показників.

Для досягнення поставленої мети необхідно виконати такі завдання:

– розкрити поняття, призначення та класифікацію інформаційних систем у сфері готельного обслуговування;

– охарактеризувати особливості організації процесу бронювання готельних номерів;

– проаналізувати сучасні програмні рішення для автоматизації бронювання номерного фонду;

– обґрунтувати вибір Python, Tkinter та SQLite для розробки інформаційної системи;

– сформулювати постановку задачі та визначити вимоги до майбутньої системи;

– спроєктувати функціональну структуру інформаційної системи бронювання;

– розробити структуру бази даних для збереження інформації про номери, клієнтів і бронювання;

– описати алгоритм перевірки доступності номерів на вибраний період;

– реалізувати графічний інтерфейс користувача, карту номерів, модуль бронювання та аналітичний модуль;

Об’єктом дослідження є процес автоматизації бронювання готельних номерів у сфері готельного обслуговування.

Предметом дослідження є методи, алгоритми та програмні засоби розробки інформаційної системи бронювання готельних номерів з використанням мови програмування Python, графічної бібліотеки Tkinter і бази даних SQLite.

Методи дослідження. У роботі використано методи аналізу та узагальнення для вивчення теоретичних засад створення інформаційних систем у готельному бізнесі; метод системного підходу для визначення структури та функціональних модулів програми; метод моделювання для проєктування бази даних, алгоритму перевірки доступності номерів і логіки роботи системи; метод програмної реалізації для створення desktop-додатка мовою Python; метод тестування для перевірки працездатності розробленої інформаційної системи.

Практичне значення роботи полягає у створенні працездатного програмного прототипу інформаційної системи бронювання готельних номерів.

Розроблена програма дозволяє переглядати каталог номерів, працювати з картою номерного фонду за поверхами, створювати бронювання, перевіряти доступність номерів на вибрані дати, розраховувати вартість проживання, скасовувати бронювання, виконувати пошук записів, експортувати дані у CSV-файл і переглядати базові аналітичні показники. Така система може бути використана як навчальний приклад або основа для подальшого розширення функціональності, зокрема додавання авторизації користувачів, календаря бронювань, вебінтерфейсу, модуля оплат або інтеграції з онлайн-сервісами.

Наукова новизна роботи полягає у проєктуванні та реалізації цілісного навчального прототипу інформаційної системи бронювання готельних номерів, який поєднує облік номерного фонду, локальну базу даних, алгоритм перевірки перетину дат, графічну карту номерів, модуль керування бронюваннями та елементи аналітики в одному desktop-додатку.

Структура дипломної роботи відповідає поставленій меті та завданням. Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел і додатків. У першому розділі розглянуто теоретичні засади створення інформаційних систем бронювання в готельному бізнесі, охарактеризовано процес бронювання номерів, проаналізовано сучасні програмні рішення та обґрунтовано вибір технологій розробки. У другому розділі здійснено проєктування інформаційної системи: визначено вимоги, функціональну структуру, структуру бази даних, алгоритм перевірки доступності номерів і підхід до побудови графічного інтерфейсу. У третьому розділі описано програмну реалізацію системи, роботу з базою даних, модулі керування номерним фондом, бронюваннями й аналітикою, а також проведено тестування працездатності програми та подано інструкцію користувача.

Апробація результатів та публікації. Основні положення і результати бакалаврської роботи доповідались на науково практичних конференціях, що проходили на базі Державного університету інформаційно-комунікаційних технологій.

1. Романчук С.Д. «ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ “РОЗУМНОГО БУДИНКУ” ІС МОНІТОРИНГУ». ІІІ «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу», ДУІКТ, Київ, 18 листопада 2025р., С, 38-39.
2. Романчук С.Д. « Особливості Інформаційної системи бронювання готельних номерів з використанням мови програмування Python » VII Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT», ДУІКТ, Київ 20 квітня 2026р., С, 53-54.

РЕФЕРАТ

Текстова частина кваліфікаційна робота на здобуття освітнього ступня бакалавр: 66 сторінок, 18 таблиць, 19 рисунків, 25 джерел.

Мета роботи – розробка та програмна реалізація інформаційної системи бронювання готельних номерів з використанням мови програмування Python для автоматизації процесів обліку номерного фонду, створення бронювань та керування інформацією про клієнтів.

Об'єкт дослідження – процес автоматизації бронювання готельних номерів у сфері готельного обслуговування.

Предмет дослідження – методи, алгоритми та програмні засоби розробки інформаційної системи бронювання готельних номерів із використанням Python, Tkinter та SQLite.

Короткий зміст роботи. У дипломній роботі досліджено особливості функціонування інформаційних систем у сфері готельного бізнесу та проаналізовано сучасні програмні рішення для автоматизації бронювання номерного фонду. Визначено основні вимоги до інформаційної системи бронювання готельних номерів, розроблено структуру бази даних та функціональну модель системи.У роботі обґрунтовано вибір мови програмування Python, графічної бібліотеки Tkinter і системи керування базами даних SQLite для реалізації desktop-додатка. Спроєктовано модулі каталогу номерів, карти номерного фонду, створення та скасування бронювань, а також модуль аналітики.Реалізовано алгоритм перевірки доступності номерів на основі аналізу перетину дат бронювання, що дозволяє уникнути подвійного резервування одного номера. Створено графічний інтерфейс користувача для зручної роботи адміністратора готелю та забезпечено локальне збереження інформації у базі даних SQLite.Розробленасистема дозволяє переглядати інформацію про номери, працювати з клієнтськими даними, створювати та скасовувати бронювання, здійснювати пошук записів, розраховувати вартість проживання та формувати базову аналітичну інформацію щодо стану номерного фонду.Проведено тестування основних функціональних можливостей системи та підтверджено працездатність розробленого програмного забезпечення.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, БРОНЮВАННЯ ГОТЕЛЬНИХ НОМЕРІВ, PYTHON, TKINTER, SQLITE, DESKTOP-ДОДАТОК, БАЗА ДАНИХ, ГОТЕЛЬНИЙ БІЗНЕС, АВТОМАТИЗАЦІЯ, АНАЛІТИКА.

РОЗДІЛ 1. ТЕОРЕТИЧНІ ЗАСАДИ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ БРОНЮВАННЯ В ГОТЕЛЬНОМУ БІЗНЕСІ

1.1 Поняття, призначення та класифікація інформаційних систем у сфері готельного обслуговування

У сучасних умовах розвитку готельного бізнесу інформаційні системи стали одним із ключових інструментів організації обслуговування гостей, управління номерним фондом, ведення бронювань, контролю завантаженості готелю та формування управлінської звітності. Якщо раніше значна частина операцій у невеликих готелях виконувалася вручну, з використанням журналів реєстрації, паперових заявок або електронних таблиць, то сьогодні навіть малі засоби розміщення поступово переходять до автоматизованих програмних рішень. Це пояснюється тим, що процес бронювання готельних номерів пов’язаний із великою кількістю змінних: датами заїзду та виїзду, типами номерів, цінами, категоріями клієнтів, статусами оплат, каналами надходження заявок, скасуваннями, змінами умов проживання та потребою швидко перевіряти доступність номерів.

Інформаційну систему у сфері готельного обслуговування доцільно розглядати як сукупність програмних, технічних, інформаційних і організаційних засобів, які забезпечують збирання, збереження, обробку, передачу та відображення даних, необхідних для ефективного функціонування готелю. Основним об’єктом обробки в такій системі є інформація про номерний фонд, гостей, бронювання, оплату, послуги, персонал і результати діяльності закладу. На практиці інформаційна система дає змогу замінити розрізнені ручні операції єдиним цифровим середовищем, у якому адміністратор може швидко отримати актуальні дані та виконати потрібну дію.

У готельному бізнесі інформаційні системи виконують кілька важливих функцій. По-перше, вони забезпечують оперативний облік номерного фонду.

Працівник готелю має бачити, які номери є вільними, які зайняті, які заброньовані, а які тимчасово недоступні через ремонт або прибирання. По-друге, такі системи автоматизують процес реєстрації клієнтів і створення бронювань. По-третє, вони зменшують ризик помилок, пов’язаних із подвійним бронюванням одного й того самого номера. По-четверте, інформаційні системи формують основу для аналітики: можна оцінювати рівень завантаженості, кількість активних бронювань, плановий дохід, популярність окремих типів номерів і динаміку попиту.

У професійній готельній практиці центральне місце серед таких рішень займають PMS-системи — Property Management Systems, тобто системи управління готельною власністю або операційною діяльністю готелю. Такі системи, як Oracle OPERA Cloud, позиціонуються як комплексні хмарні PMS-рішення, що централізують дані, підтримують операції фронт-офісу, інтеграції та доступ до інформації з різних робочих місць [1]. Oracle також окремо підкреслює роль програмного забезпечення для бронювання номерів, яке використовує централізовану базу даних номерного фонду і спрощує управління різними типами бронювань [2].

За функціональним призначенням інформаційні системи готелю можна поділити на кілька груп. До першої групи належать системи бронювання, які відповідають за прийом заявок, перевірку доступності номерів, фіксацію дат проживання та формування запису бронювання. До другої групи належать PMS-системи, що охоплюють ширший спектр операцій: поселення, виселення, управління номерним фондом, облік гостей, звітність, взаємодію з іншими сервісами готелю. До третьої групи можна віднести CRM-системи, орієнтовані на роботу з клієнтами, їхніми контактами, історією проживання та персоналізованими пропозиціями. До четвертої групи належать системи онлайн-бронювання, які інтегруються із сайтом готелю або зовнішніми каналами продажів. Наприклад, Cloudbeds описує свою платформу як об’єднану систему управління готелем, що поєднує PMS, інструменти операційного управління та рішення для покращення роботи з гостями [3].

За способом розгортання готельні інформаційні системи можна класифікувати на локальні, клієнт-серверні, веборієнтовані та хмарні. Локальні системи встановлюються безпосередньо на комп’ютері користувача і можуть працювати без постійного підключення до Інтернету. Саме такий підхід є доцільним для навчального програмного продукту, розробленого мовою Python, оскільки він дозволяє продемонструвати основну логіку бронювання без складної серверної інфраструктури. Клієнт-серверні системи використовують окремий сервер бази даних і кілька робочих місць користувачів. Веборієнтовані системи запускаються через браузер. Хмарні рішення розміщуються на стороні постачальника послуги та надають доступ до функцій системи через Інтернет.

За масштабом використання можна виокремити інформаційні системи для малих готелів, середніх закладів розміщення, великих готельних комплексів і мережевих готелів. Для малого готелю достатньою може бути система, яка містить модулі обліку номерів, бронювання, клієнтів і базової звітності. Для великого готелю потрібна складніша система, що підтримує роботу багатьох адміністраторів, інтеграцію з платіжними сервісами, рестораном, службою прибирання, бухгалтерією, каналами онлайн-продажів і системами електронних ключів. У межах цієї дипломної роботи ми розглядаємо саме навчальний прототип локальної інформаційної системи, який відображає базову бізнес-логіку бронювання готельних номерів і може бути розширений у майбутньому.

Отже, інформаційна система бронювання готельних номерів є важливим інструментом автоматизації готельної діяльності. Вона забезпечує точність обліку, зручність роботи адміністратора, швидкість обробки заявок і можливість аналізу поточного стану номерного фонду. Для нашої роботи особливо важливо, що така система може бути реалізована у вигляді desktop-додатка з локальною базою даних, графічним інтерфейсом, картою номерів і модулем створення бронювань.

1.2 Особливості організації процесу бронювання готельних номерів

Процес бронювання готельних номерів є одним із центральних бізнес-процесів у діяльності готелю. Саме з бронювання часто починається взаємодія клієнта із закладом розміщення. Від того, наскільки швидко, точно і зручно організований цей процес, залежить не лише рівень завантаженості номерного фонду, а й загальне враження гостя від сервісу. Бронювання можна визначити як попереднє закріплення певного номера або типу номера за клієнтом на конкретний період часу з фіксацією основних умов проживання.

У класичному вигляді процес бронювання складається з кількох послідовних етапів. Спочатку клієнт звертається до готелю або використовує онлайн-канал для перевірки доступності номерів. Далі адміністратор або система визначає, чи є вільні номери потрібної категорії на зазначені дати. Після цього фіксуються персональні дані клієнта, обирається номер, розраховується орієнтовна або повна вартість проживання, визначається статус бронювання та зберігається відповідний запис у базі даних. У разі зміни планів клієнта бронювання може бути скасоване або відредаговане.

Найважливішою задачею інформаційної системи на цьому етапі є перевірка доступності номера. Якщо така перевірка виконується вручну, існує ризик помилок, особливо коли адміністратор працює з великою кількістю заявок. Наприклад, один і той самий номер може бути випадково заброньований двома різними клієнтами на періоди, що перетинаються. Щоб уникнути такої ситуації, система повинна автоматично аналізувати вже наявні активні бронювання та порівнювати їх із новим періодом проживання.

У програмній реалізації ця логіка зазвичай базується на перевірці перетину дат. Якщо нове бронювання має дату заїзду check_in і дату виїзду check_out, то система повинна знайти всі активні бронювання для вибраного номера та визначити, чи не перетинаються вони з новим інтервалом. У нашій програмі використовується принцип: номер вважається вільним, якщо для нього немає активного бронювання, яке перетинається з новими датами. Тобто попереднє

бронювання не створює конфлікту лише тоді, коли воно повністю завершується до початку нового бронювання або починається після завершення нового бронювання. Такий підхід є зрозумілим, логічним і придатним для реалізації в SQL-запиті.

Важливою особливістю бронювання є також робота зі статусами. Бронювання може бути активним, скасованим, завершеним або очікувати підтвердження. У навчальній системі достатньо використати два основні статуси: активне і скасоване. Активне бронювання бере участь у перевірці доступності номера, а скасоване — ні. Це означає, що після скасування бронювання номер знову може бути доступним для інших клієнтів у відповідний період. Така логіка дозволяє не видаляти запис із бази даних, а зберігати історію операцій.

Окремого значення набуває розрахунок вартості проживання. У найпростішому варіанті вартість визначається як добуток кількості ночей на ціну номера за одну ніч. Наприклад, якщо номер коштує 1500 грн за ніч, а клієнт бронює його на три ночі, то загальна сума становитиме 4500 грн. У складніших системах можуть враховуватися сезонність, знижки, тарифи вихідного дня, додаткові послуги, передоплата, податки або комісії. Проте для навчального прототипу достатньо базової формули, оскільки вона демонструє основний принцип автоматичного розрахунку.

Особливістю сучасного бронювання є також багатоканальність. Клієнт може забронювати номер через сайт готелю, телефоном, через адміністратора, через онлайн-агентство або спеціалізовану платформу. У професійних рішеннях важливо синхронізувати дані про доступність номерів між усіма каналами, щоб не виникало дублювання. Наприклад, документація Cloudbeds щодо booking engine інтеграцій зазначає, що такі інтеграції можуть працювати з інформацією про актуальний номерний фонд, тарифи та правила бронювання [4]. Для нашого навчального проєкту ця ідея є перспективним напрямом розвитку: у майбутньому локальна desktop-система може бути доповнена вебмодулем або API для онлайн-бронювання.

У межах нашої інформаційної системи процес бронювання організовано через кілька взаємопов’язаних модулів. Каталог номерів дозволяє переглянути доступні номери, їхній тип, місткість і вартість. Карта номерів подає номерний фонд у більш наочній формі за поверхами. Модуль нового бронювання дає змогу вибрати номер, ввести дані клієнта, зазначити дати заїзду та виїзду, перевірити доступність і створити бронювання. Модуль перегляду бронювань дозволяє знаходити потрібні записи, скасовувати їх і експортувати дані. Така структура відповідає реальній логіці роботи адміністратора готелю.

Отже, процес бронювання готельних номерів є не просто операцією запису даних, а комплексним інформаційним процесом, що включає перевірку доступності, роботу з клієнтами, облік номерного фонду, розрахунок вартості, зміну статусів і формування історії бронювань. Саме тому його автоматизація є доцільною та практично значущою.

1.3 Аналіз сучасних програмних рішень для автоматизації бронювання номерного фонду

Сучасний ринок програмного забезпечення для готельного бізнесу пропонує велику кількість рішень, орієнтованих на автоматизацію бронювання, управління номерним фондом, взаємодію з гостями, аналітику та інтеграцію з онлайн-каналами продажів. Найбільш поширеними є PMS-системи, booking engine, channel manager, CRM-рішення та комплексні hotel management systems. Кожен із цих класів програмного забезпечення виконує окрему роль, але в реальній практиці вони часто об’єднуються в єдину цифрову екосистему готелю.

PMS-система є ядром цифрового управління готелем. Вона забезпечує облік номерного фонду, управління бронюваннями, поселення і виселення гостей, облік оплат, формування звітів, взаємодію з іншими службами готелю. Наприклад, Oracle OPERA Cloud подається як хмарна система управління готелем, що централізує дані, підтримує мобільну роботу персоналу та інтегрується з іншими рішеннями готельної інфраструктури [1]. Це свідчить про

те, що сучасна PMS-система вже не є просто електронним журналом бронювань, а виконує роль операційної платформи для всього закладу розміщення.

Booking engine, або модуль онлайн-бронювання, орієнтований насамперед на взаємодію з гостем через сайт готелю або інший цифровий канал. Його головна функція — дозволити клієнту самостійно вибрати дати, тип номера, тариф, додаткові послуги та оформити бронювання. Cloudbeds, наприклад, позиціонує booking engine як інструмент для прямих бронювань через сайт готелю, а також як частину ширшої платформи управління готельним бізнесом [5]. Для готелю це важливо, оскільки пряме бронювання зменшує залежність від посередників і дозволяє краще контролювати взаємодію з клієнтом.

Channel manager відповідає за синхронізацію доступності номерів і тарифів між різними каналами продажів. Якщо готель одночасно працює з власним сайтом, онлайн-агентствами та іншими платформами, йому потрібно забезпечити єдину актуальну інформацію про номерний фонд. Інакше можливе подвійне бронювання або неправильне відображення цін. У комплексних системах channel manager може бути інтегрований із PMS і booking engine, що дозволяє автоматично оновлювати дані в різних каналах.

CRM-системи у готельному бізнесі призначені для управління взаємовідносинами з клієнтами. Вони можуть зберігати історію проживань, контактні дані, побажання гостей, інформацію про лояльність, відгуки та комунікації. Для великих готелів це важливо, оскільки повторний клієнт може отримувати персоналізовані пропозиції, а адміністрація — краще розуміти потреби цільової аудиторії. У невеликому навчальному проєкті CRM-функціональність може бути представлена базовою таблицею клієнтів, яка зберігає ПІБ, телефон та електронну пошту.

Порівняно з професійними системами, навчальний програмний продукт має обмежену функціональність, однак він дозволяє відтворити основну логіку автоматизації. Наша система не претендує на рівень промислової PMS-платформи, але демонструє базові принципи: створення каталогу номерів, перевірку доступності, реєстрацію клієнтів, створення бронювань, скасування

записів, експорт даних і перегляд аналітики. Саме ці функції є фундаментом, на якому надалі можна будувати складнішу систему.

Під час аналізу сучасних програмних рішень можна визначити кілька вимог, які варто враховувати під час розробки власної інформаційної системи. По-перше, система повинна мати зрозумілий інтерфейс, оскільки з нею працює адміністратор, який має швидко виконувати операції під час спілкування з клієнтом. По-друге, база даних повинна зберігати інформацію структуровано, щоб уникнути дублювання і втрати даних. По-третє, логіка бронювання має запобігати конфліктам дат. По-четверте, бажано передбачити наочне відображення номерного фонду, наприклад карту номерів за поверхами. По-п’яте, система повинна мати хоча б базові аналітичні можливості.

Сучасні професійні рішення також активно використовують мобільність, хмарні технології та інтеграції. Oracle Hospitality наголошує на мобільній роботі персоналу, централізації даних та інтеграції з іншими операційними сервісами готелю [1]. У нашій дипломній роботі ми не реалізуємо хмарну архітектуру, однак можемо використати цей підхід як орієнтир для подальшого розвитку системи. Наприклад, desktop-додаток у майбутньому можна трансформувати у вебзастосунок, додати серверну частину, авторизацію, ролі користувачів і синхронізацію з онлайн-бронюванням.

Отже, аналіз сучасних програмних рішень показує, що автоматизація бронювання номерного фонду є комплексним завданням, яке охоплює не лише створення запису бронювання, а й управління даними, візуалізацію стану номерів, роботу з клієнтами, звітність та інтеграцію з іншими каналами. Розроблена в межах роботи система є спрощеним, але функціонально цілісним прототипом, який відображає основні принципи таких рішень.

1.4 Обґрунтування вибору Python, Tkinter та SQLite для розробки інформаційної системи

Вибір технологій для розробки інформаційної системи бронювання готельних номерів має відповідати меті дипломної роботи, складності завдання, вимогам до запуску програми та можливості продемонструвати повний цикл роботи системи. Оскільки розроблюваний програмний продукт є навчальним desktop-додатком, важливо було обрати такі засоби, які дозволяють швидко створити графічний інтерфейс, організувати локальне збереження даних і реалізувати бізнес-логіку без потреби в складній серверній інфраструктурі. Саме тому для реалізації системи обрано мову програмування Python, бібліотеку Tkinter і базу даних SQLite.

Python є високорівневою мовою програмування, яка добре підходить для швидкої розробки прикладних програм. Офіційний опис Python підкреслює, що це інтерпретована, об’єктно-орієнтована мова високого рівня з динамічною семантикою, зручна для швидкої розробки застосунків і поєднання різних компонентів [6]. Для дипломного проєкту це має суттєве значення, оскільки Python дозволяє зосередитися не лише на синтаксичних деталях, а насамперед на логіці інформаційної системи: створенні класів, функцій, обробці подій, роботі з базою даних і побудові інтерфейсу.

Перевагою Python є також його читабельність. Код, написаний цією мовою, зазвичай легше пояснювати в дипломній роботі, демонструвати у вигляді лістингу та супроводжувати коментарями. Це особливо важливо для розділу програмної реалізації, де потрібно показати не тільки кінцевий результат, а й логіку побудови системи. У нашій програмі Python використовується для створення головного вікна, вкладок інтерфейсу, обробки дій користувача, перевірки дат, взаємодії з базою даних, формування списків номерів, створення бронювань, скасування записів і експорту даних.

Для створення графічного інтерфейсу було обрано Tkinter. Це стандартний інтерфейс Python до графічного інструментарію Tcl/Tk, доступний на основних

операційних системах, зокрема Windows, macOS і Unix-платформах [7]. Така характеристика є важливою, оскільки програма має запускатися без встановлення великої кількості додаткових бібліотек. Tkinter дозволяє створювати вікна, вкладки, кнопки, таблиці, поля введення, текстові блоки, випадаючі списки та графічні елементи canvas.

У межах нашої системи Tkinter використовується не лише для простого введення даних, а й для створення більш наочного інтерфейсу. Зокрема, реалізовано вкладки «Каталог номерів», «Карта номерів», «Нове бронювання», «Бронювання» та «Аналітика». Використання canvas дозволило створити схематичний вигляд номерів, а кнопки на карті поверхів — подати номерний фонд у візуальній формі. Це робить програму більш зрозумілою для користувача, оскільки адміністратор може не тільки читати таблицю, а й бачити структуру номерів за поверхами.

Для збереження даних обрано SQLite. Це вбудована база даних, яка не потребує окремого серверного процесу, налаштування користувачів або складного адміністрування. Офіційна документація SQLite визначає її як самодостатній, безсерверний, zero-configuration SQL-рушій баз даних [8]. Крім того, SQLite описується як невелика, швидка, самодостатня і надійна SQL-база даних, що широко використовується в різних програмних продуктах [9]. Для локальної desktop-системи бронювання це є оптимальним варіантом, оскільки вся інформація може зберігатися в одному файлі hotel_booking.db.

SQLite має кілька переваг саме для навчального проєкту. По-перше, вона не потребує встановлення окремого сервера бази даних, тому програму легше запустити на іншому комп’ютері. По-друге, SQLite підтримує стандартні SQL-запити, що дозволяє продемонструвати нормальну структуру бази даних із таблицями rooms, clients і bookings. По-третє, вона добре інтегрується з Python через стандартні засоби. По-четверте, файл бази даних можна легко переносити разом із програмою.

У нашій інформаційній системі SQLite використовується для збереження трьох основних груп даних. Таблиця rooms містить інформацію про номери:

номер кімнати, поверх, тип, місткість, ціну, опис і зручності. Таблиця clients зберігає інформацію про клієнтів: ПІБ, телефон і електронну пошту. Таблиця bookings фіксує бронювання: номер, клієнта, дати заїзду й виїзду, суму, статус і примітки. Така структура є достатньою для реалізації основної логіки бронювання та демонструє зв’язок між сутностями системи.

Комбінація Python, Tkinter і SQLite є виправданою з погляду простоти, доступності та функціональності. Python забезпечує логіку роботи програми, Tkinter відповідає за інтерфейс, а SQLite — за збереження даних. Разом ці технології дозволяють створити повноцінний desktop-додаток без потреби в зовнішньому сервері, вебхостингу або складних залежностях. Для дипломної роботи це особливо зручно, оскільки програму можна передати в архіві, запустити з командного рядка та продемонструвати її роботу на звичайному комп’ютері.

Водночас потрібно розуміти й обмеження такого технологічного вибору. Tkinter не забезпечує такого рівня сучасного дизайну, як спеціалізовані фреймворки для веб- або мобільної розробки. SQLite добре підходить для локальної роботи, але для великого готелю з багатьма одночасними користувачами доцільніше використовувати серверні СУБД, наприклад PostgreSQL або MySQL. Desktop-архітектура також обмежує можливості онлайн-доступу. Проте для навчального прототипу, який має продемонструвати принципи автоматизації бронювання, обраний стек є достатнім і методично обґрунтованим.

У першому розділі було розглянуто теоретичні засади створення інформаційних систем бронювання в готельному бізнесі. Встановлено, що інформаційна система у сфері готельного обслуговування є інструментом автоматизації процесів, пов’язаних з обліком номерного фонду, роботою з клієнтами, створенням бронювань, контролем зайнятості номерів і формуванням аналітичної інформації. Такі системи підвищують точність роботи адміністратора, зменшують ризик помилок і забезпечують швидкий доступ до актуальних даних.

Було визначено, що процес бронювання готельних номерів має чітку логіку: вибір дат, перевірка доступності, внесення даних клієнта, розрахунок вартості, створення запису та подальше керування його статусом. Особливе значення має алгоритм перевірки перетину дат, оскільки саме він запобігає подвійному бронюванню одного номера. Для розроблюваної системи ця логіка є центральною.

Аналіз сучасних програмних рішень показав, що професійні PMS- і booking-системи орієнтовані на централізацію даних, управління номерним фондом, інтеграцію з каналами продажів і підвищення якості обслуговування гостей. Водночас для навчального проєкту доцільним є створення спрощеного desktop-прототипу, який реалізує базові функції таких систем: каталог номерів, карту номерного фонду, створення бронювань, скасування записів, експорт і аналітику.

Обґрунтовано вибір Python, Tkinter та SQLite як технологічної основи розробки. Python забезпечує зручну реалізацію логіки програми, Tkinter дозволяє створити графічний інтерфейс без додаткових складних залежностей, а SQLite забезпечує локальне збереження даних у простій і надійній формі. Отже, обраний технологічний стек відповідає меті роботи та дозволяє реалізувати функціональну інформаційну систему бронювання готельних номерів.

РОЗДІЛ 2. ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ БРОНЮВАННЯ ГОТЕЛЬНИХ НОМЕРІВ

2.1 Постановка задачі та визначення вимог до інформаційної системи

Проєктування інформаційної системи бронювання готельних номерів передбачає попереднє визначення її призначення, кола користувачів, основних функцій, структури даних і логіки взаємодії між окремими модулями. У межах цієї дипломної роботи нами розробляється desktop-додаток мовою програмування Python, призначений для автоматизації процесу обліку номерного фонду, створення бронювань, перевірки доступності номерів, ведення інформації про клієнтів і формування базових аналітичних показників роботи готелю.

Основна проблема, яку має вирішувати система, полягає в тому, що ручне ведення бронювань або використання звичайних електронних таблиць не забезпечує достатнього рівня надійності, зручності та швидкості роботи. У разі ручного обліку адміністратор може помилитися під час перевірки дат, випадково створити дублююче бронювання, втратити інформацію про клієнта або неправильно розрахувати вартість проживання. Тому завданням проєктованої системи є створення єдиного програмного середовища, у якому всі основні операції з бронювання номерів виконуються автоматизовано.

Метою проєктування є визначення такої структури системи, яка дозволить адміністратору готелю швидко переглядати номери, бачити їхній стан, створювати нові бронювання, скасовувати наявні записи, перевіряти зайнятість номерів на конкретні дати та отримувати коротку інформацію про завантаженість готелю. З огляду на навчальний характер роботи система не орієнтована на промислове використання у великій готельній мережі, однак її функціональність відображає базову логіку реальних систем бронювання.

Користувачем системи виступає адміністратор готелю. Він працює з графічним інтерфейсом, у якому доступні вкладки каталогу номерів, карти

номерного фонду, нового бронювання, списку бронювань та аналітики. Система повинна бути простою для запуску, не вимагати складного встановлення серверного програмного забезпечення та зберігати дані локально. Саме тому для її реалізації використовується SQLite як вбудована база даних.

Основні вимоги до інформаційної системи подано в таблиці 2.1.

Таблиця 2.1

Основні вимоги до інформаційної системи бронювання готельних номерів

Група вимог	**Зміст вимоги**	**Реалізація у системі**
Функціональні вимоги	Перегляд номерного фонду	Вкладка «Каталог номерів» із таблицею номерів
Функціональні вимоги	Перевірка доступності номера	Автоматична перевірка перетину дат бронювання
Функціональні вимоги	Створення бронювання	Форма введення даних клієнта, номера та дат
Функціональні вимоги	Скасування бронювання	Зміна статусу бронювання з active на cancelled
Функціональні вимоги	Візуалізація номерів	Карта номерів за поверхами
Функціональні вимоги	Аналітика	Показ кількості номерів, активних бронювань, доступних номерів і доходу
Інформаційні вимоги	Збереження даних про номери	Таблиця rooms у базі SQLite
Інформаційні вимоги	Збереження даних про клієнтів	Таблиця clients
Інформаційні вимоги	Збереження даних про бронювання	Таблиця bookings
Інтерфейсні вимоги	Зручний графічний інтерфейс	Використання Tkinter і вкладкової структури
Надійність	Захист від подвійного бронювання	SQL-перевірка конфлікту дат
Простота запуску	Робота без серверної частини	Локальна база SQLite у файлі hotel_booking.db

Під час постановки задачі було визначено, що система повинна виконувати не лише функцію запису бронювання, а й забезпечувати контроль логіки процесу. Це означає, що перед створенням запису система має перевірити коректність введених дат, наявність клієнта, вибір номера, а також доступність номера у зазначений період. Якщо будь-яка з умов не виконується, програма повинна повідомити користувача про помилку та не створювати некоректний запис.

Важливою вимогою є також збереження історії бронювань. Тому при скасуванні бронювання запис не видаляється з бази даних, а лише змінює свій статус. Такий підхід є більш правильним, оскільки дає змогу зберегти інформацію про всі дії, які виконувалися в системі. У майбутньому це може бути використано для формування статистики, аналізу скасувань або створення звітності.

Для візуального представлення загальної постановки задачі доцільно подати діаграму варіантів використання системи.

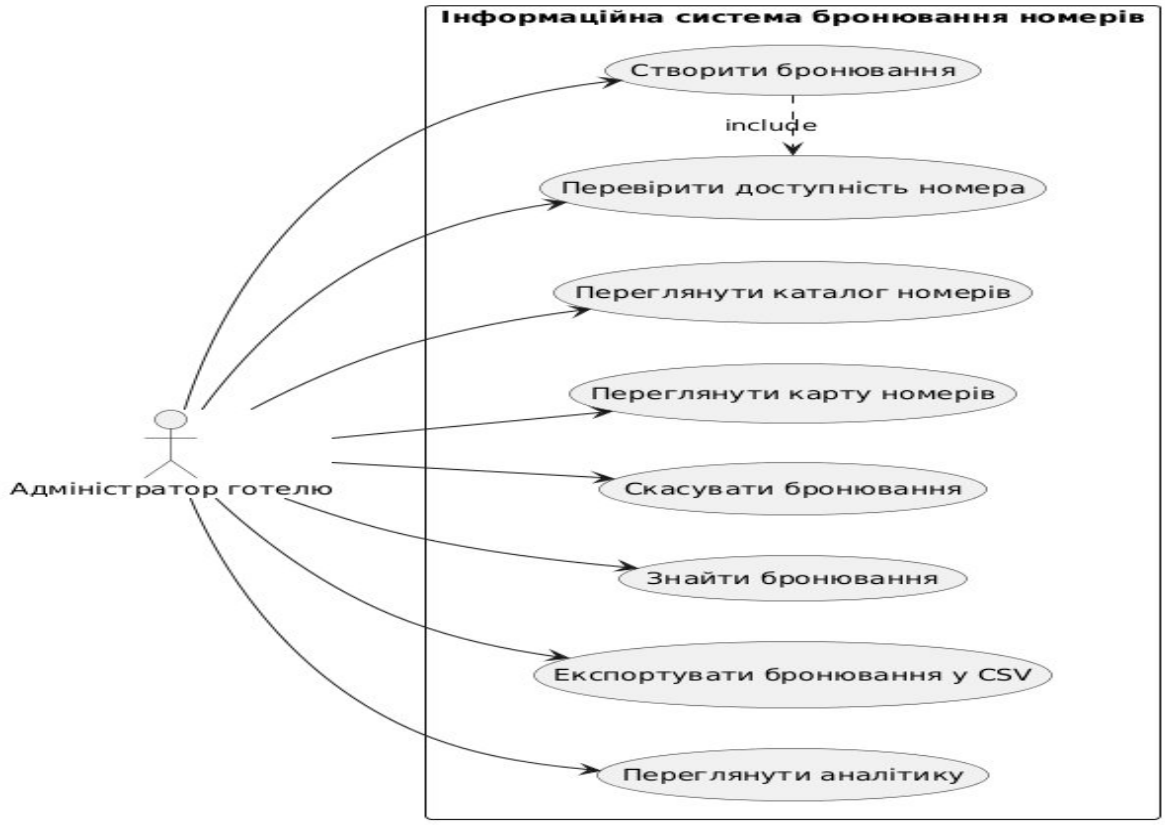


Рис. 2.1 Діаграма варіантів використання інформаційної системи

Отже, на етапі постановки задачі було визначено, що система має бути локальним desktop-додатком для адміністратора готелю, який забезпечує повний базовий цикл роботи з номерним фондом і бронюваннями.

2.2 Проєктування функціональної структури системи бронювання

Функціональна структура інформаційної системи визначає склад її основних модулів і взаємозв’язки між ними. Для зручності користувача система поділена на кілька логічних блоків, кожен з яких відповідає за окремий напрям роботи. Такий підхід дозволяє зробити програму зрозумілою, полегшує її подальше тестування та створює основу для майбутнього розширення.

У розробленій системі можна виділити п’ять основних функціональних модулів: модуль каталогу номерів, модуль карти номерів, модуль створення бронювання, модуль керування бронюваннями та модуль аналітики. Окремо слід виділити модуль роботи з базою даних, який забезпечує збереження та отримання інформації з таблиць SQLite.

Модуль каталогу номерів призначений для відображення списку всіх номерів готелю. У ньому користувач бачить номер кімнати, поверх, тип номера, місткість, ціну за одну ніч і статус доступності. Додатково передбачено фільтрацію за типом номера, мінімальною місткістю та датами проживання. Це дозволяє адміністратору швидко знайти номер, який відповідає запиту клієнта.

Модуль карти номерів забезпечує візуальне представлення номерного фонду за поверхами. На відміну від звичайної таблиці, карта дає змогу швидше зорієнтуватися в розміщенні номерів. У системі номери відображаються у вигляді кнопок, колір яких залежить від статусу номера на вибрані дати. Зелений колір означає, що номер вільний, червоний — що номер зайнятий, а сірий — що перевірка доступності ще не виконувалася або дати не задані.

Модуль створення бронювання реалізує основний бізнес-процес системи. Адміністратор обирає номер, вводить ПІБ клієнта, телефон, електронну пошту,

дати заїзду та виїзду, а також за потреби додає примітки. Після цього система може перевірити доступність номера, розрахувати вартість проживання та створити бронювання в базі даних.

Модуль керування бронюваннями призначений для перегляду вже створених записів. У ньому адміністратор може знайти бронювання за ПІБ клієнта або номером кімнати, переглянути статус запису, скасувати активне бронювання або експортувати список бронювань у CSV-файл. Такий функціонал важливий для практичного використання системи, оскільки бронювання не є статичним записом — воно може змінювати статус протягом життєвого циклу.

Модуль аналітики забезпечує короткий управлінський огляд. Він показує загальну кількість номерів у системі, кількість активних бронювань, кількість доступних номерів у заданий період і сумарний плановий дохід за активними бронюваннями. Навіть базова аналітика підвищує практичну цінність системи, оскільки адміністратор бачить не лише окремі записи, а й загальний стан номерного фонду.

Таблиця 2.2

Функціональні модулі інформаційної системи

Модуль системи	**Основне призначення**	**Основні операції**
Каталог номерів	Перегляд номерного фонду	Перегляд, фільтрація, вибір номера
Карта номерів	Візуальне представлення номерів	Перегляд поверхів, вибір номера, відображення статусу
Нове бронювання	Створення бронювання	Введення клієнта, вибір номера, перевірка дат, розрахунок ціни
Бронювання	Керування створеними записами	Пошук, перегляд, скасування, експорт
Аналітика	Оцінка стану системи	Підрахунок номерів, активних бронювань, доходу
База даних	Збереження інформації	Запис, читання, оновлення даних

Функціональну структуру системи можна подати у вигляді компонентної діаграми.

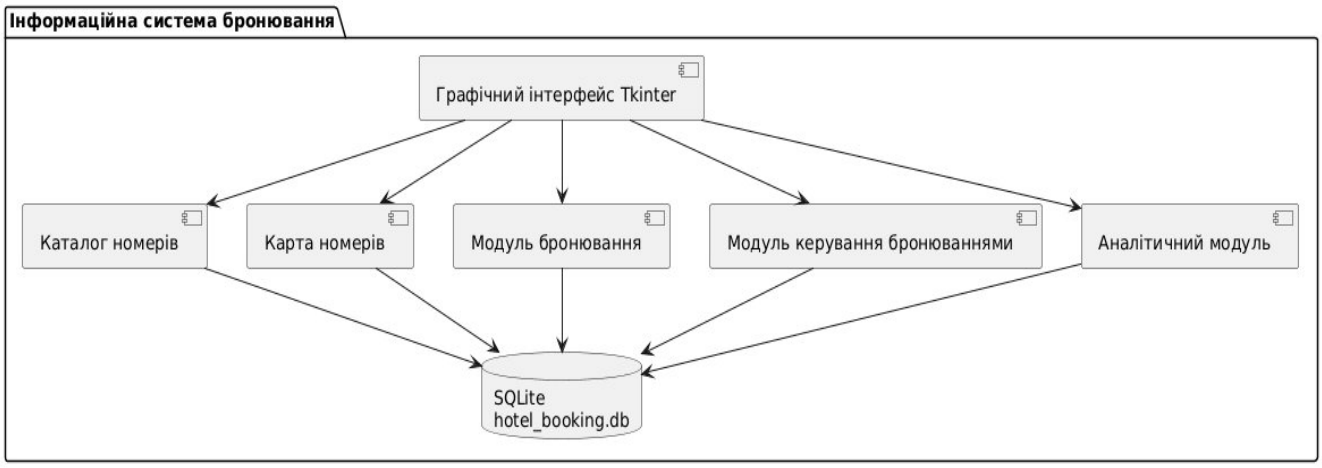


Рис. 2.2 Функціональна структура інформаційної системи бронювання

З наведеної діаграми видно, що центральним елементом системи є графічний інтерфейс, через який адміністратор взаємодіє з усіма функціональними модулями. Усі модулі звертаються до єдиної бази даних SQLite. Це забезпечує цілісність інформації, оскільки каталог номерів, карта номерів, бронювання та аналітика використовують спільні дані.

2.3 Розробка структури бази даних для обліку номерів, клієнтів і бронювань

База даних є основою інформаційної системи, оскільки саме вона забезпечує збереження інформації між запусками програми. У проєктованій системі використовується локальна база SQLite, яка зберігається у файлі hotel_booking.db. Такий варіант є доцільним для desktop-додатка, оскільки не потребує окремого сервера та складного адміністрування.

Під час проєктування бази даних було визначено три основні сутності: номер, клієнт і бронювання. Сутність «номер» описує номерний фонд готелю. Сутність «клієнт» містить контактну інформацію про особу, яка здійснює бронювання. Сутність «бронювання» пов’язує конкретного клієнта з конкретним номером на визначений період часу.

Таблиця rooms призначена для збереження інформації про готельні номери. Вона містить номер кімнати, поверх, тип номера, місткість, ціну за одну

ніч, перелік зручностей і текстовий опис. Таке наповнення дозволяє використовувати дані не лише для бронювання, а й для формування картки номера в інтерфейсі.

Таблиця clients містить дані клієнтів: ПІБ, телефон і електронну пошту. Для навчального прототипу цього достатньо, оскільки система не реалізує паспортний облік, оплату або складну CRM-функціональність. У майбутньому таблицю можна розширити, додавши поля для адреси, документа, історії проживань або категорії клієнта.

Таблиця bookings є центральною таблицею системи. Вона містить посилання на номер і клієнта, дати заїзду та виїзду, загальну вартість проживання, статус бронювання, примітки та дату створення запису. Саме ця таблиця використовується для перевірки зайнятості номерів.

Таблиця 2.3

Структура таблиці rooms

Поле	**Тип даних**	**Призначення**
id	INTEGER PRIMARY KEY	Унікальний ідентифікатор номера
room_number	TEXT UNIQUE	Номер кімнати
floor	INTEGER	Поверх розміщення
room_type	TEXT	Тип номера
capacity	INTEGER	Кількість місць
price_per_night	REAL	Ціна за одну ніч
amenities	TEXT	Перелік зручностей
description	TEXT	Опис номера

Таблиця 2.4

Структура таблиці clients

Поле	**Тип даних**	**Призначення**
id	INTEGER PRIMARY KEY	Унікальний ідентифікатор клієнта
full_name	TEXT	ПІБ клієнта
phone	TEXT	Номер телефону
email	TEXT	Електронна пошта

Таблиця 2.5

Структура таблиці bookings

Поле	**Тип даних**	**Призначення**
id	INTEGER PRIMARY KEY	Унікальний ідентифікатор бронювання
room_id	INTEGER	Посилання на номер
client_id	INTEGER	Посилання на клієнта
check_in	TEXT	Дата заїзду
check_out	TEXT	Дата виїзду
total_price	REAL	Загальна вартість проживання
status	TEXT	Статус бронювання
notes	TEXT	Додаткові примітки
created_at	TEXT	Дата і час створення запису

Зв’язок між таблицями побудовано за принципом «один до багатьох». Один номер може мати багато бронювань у різні періоди часу. Один клієнт також потенційно може мати багато бронювань. У межах поточної реалізації при кожному новому бронюванні створюється новий запис клієнта, однак у майбутньому систему можна вдосконалити, додавши пошук уже наявного клієнта за телефоном або електронною поштою.

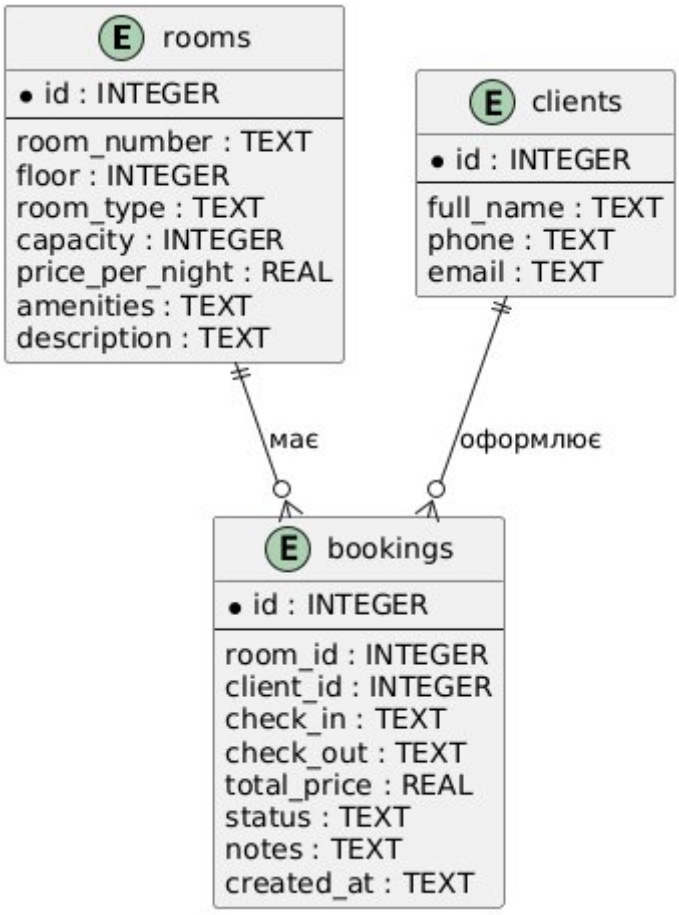


Рис. 2.3 ER-діаграма бази даних системи

У цій структурі таблиця bookings виконує роль зв’язувальної сутності між клієнтами та номерами. Вона дозволяє визначити, хто саме забронював номер, на який період і на яку суму. Наявність поля status дає змогу не видаляти бронювання фізично, а змінювати його стан. Це важливо для збереження історії операцій.

2.4 Проєктування алгоритму перевірки доступності номерів

Алгоритм перевірки доступності номера є ключовим елементом інформаційної системи бронювання. Його завдання полягає в тому, щоб визначити, чи можна створити нове бронювання для конкретного номера на заданий період. Якщо алгоритм працює неправильно, система може допустити подвійне бронювання, що є критичною помилкою для готельного бізнесу.

Вхідними даними алгоритму є ідентифікатор номера, дата заїзду та дата виїзду. Перед перевіркою система повинна переконатися, що дата виїзду є пізнішою за дату заїзду. Якщо користувач вводить некоректні дати, система не повинна виконувати подальшу перевірку та має вивести повідомлення про помилку.

Основний принцип перевірки полягає у виявленні перетину часових інтервалів. Нехай існує нове бронювання з датами:

new_check_in — дата заїзду нового бронювання;

new_check_out — дата виїзду нового бронювання.

У базі даних уже можуть бути активні бронювання з датами:

check_in — дата заїзду наявного бронювання;

check_out — дата виїзду наявного бронювання.

Нове бронювання не конфліктує з наявним лише у двох випадках: якщо наявне бронювання завершується до початку нового або якщо наявне бронювання починається після завершення нового. У SQL-логіці це можна подати так:

NOT (check_out <= new_check_in OR check_in >= new_check_out)

Тобто конфлікт виникає тоді, коли не виконується умова повного неперетину інтервалів. Якщо кількість таких конфліктних активних бронювань дорівнює нулю, номер вважається доступним. Якщо хоча б одне активне бронювання перетинається з новим періодом, номер вважається зайнятим.

Таблиця 2.6

Приклади перевірки перетину дат бронювання

Наявне бронювання	**Нове бронювання**	**Результат**
01.06–05.06	05.06–08.06	Конфлікту немає, номер звільняється 05.06
01.06–05.06	04.06–07.06	Конфлікт є, періоди перетинаються
10.06–15.06	01.06–10.06	Конфлікту немає
10.06–15.06	12.06–14.06	Конфлікт є, новий період усередині наявного

Продовження таблиці 2.6

10.06–15.06	08.06–17.06	Конфлікт є, новий період охоплює наявний
10.06–15.06	16.06–20.06	Конфлікту немає

У програмі перевірка доступності виконується до створення бронювання. Якщо номер зайнятий, користувач отримує повідомлення про помилку, і запис не додається до бази даних. Якщо номер вільний, система розраховує вартість проживання та створює запис у таблиці bookings.

Вартість проживання визначається за формулою:

$$\mathbf{В = Кн \times Цн},$$

де **В** — загальна вартість бронювання;

Кн — кількість ночей;

Цн — ціна номера за одну ніч.

Наприклад, якщо номер коштує 2300 грн за ніч, а клієнт бронює його на 3 ночі, то загальна сума становить:

$$\mathbf{В = 3 \times 2300 = 6900\ грн}.$$

Алгоритм створення бронювання можна подати за допомогою блок-схеми.

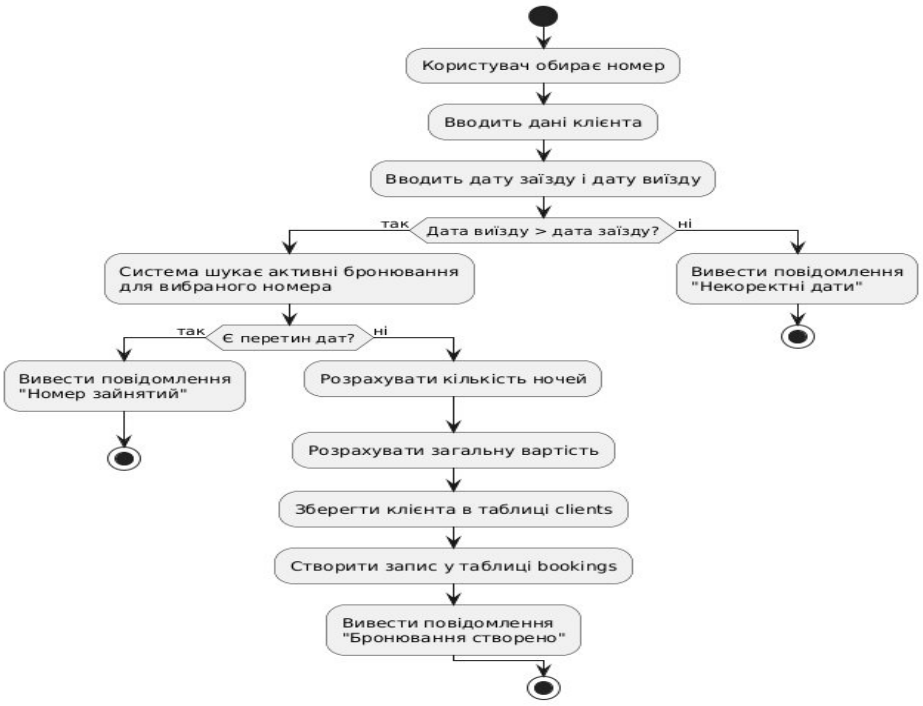


Рисунок 2.4 — Алгоритм перевірки доступності номера та створення бронювання

З наведеної схеми видно, що система не дозволяє створити бронювання без попередньої перевірки. Це підвищує надійність роботи програми та забезпечує коректність даних у базі.

2.5 Проєктування графічного інтерфейсу користувача, карти номерів та модуля аналітики

Графічний інтерфейс є важливою складовою інформаційної системи, оскільки саме через нього користувач взаємодіє з програмою. Для адміністратора готелю інтерфейс має бути зрозумілим, логічним і швидким у використанні. Саме тому у проєкті обрано вкладкову структуру, яка дозволяє розділити функції системи за окремими напрямами.

У програмі передбачено п’ять основних вкладок: «Каталог номерів», «Карта номерів», «Нове бронювання», «Бронювання» та «Аналітика». Такий поділ відповідає типовому порядку роботи адміністратора. Спочатку він може переглянути номерний фонд, потім перейти до карти номерів, обрати конкретний номер, створити бронювання, а пізніше знайти або скасувати запис. Аналітична вкладка використовується для швидкого огляду загального стану системи.

Таблиця 2.7

Структура графічного інтерфейсу користувача

Вкладка	**Призначення**	**Основні елементи**
Каталог номерів	Перегляд і фільтрація номерів	Таблиця номерів, фільтри, картка номера
Карта номерів	Візуальне відображення номерів	Поверхи, кнопки номерів, кольорові статуси
Нове бронювання	Оформлення бронювання	Поля клієнта, вибір номера, дати, примітки

Продовження таблиці 2.7

Бронювання	Робота зі створеними бронюваннями	Таблиця бронювань, пошук, скасування, експорт
Аналітика	Огляд показників системи	Картки показників, текстовий аналітичний блок

Особливо важливим елементом інтерфейсу є карта номерів. Вона дозволяє відійти від суто табличного представлення інформації та зробити систему більш наочною. Кожен номер подається у вигляді окремого графічного елемента. При натисканні на номер користувач бачить його опис, тип, поверх, місткість, ціну та перелік зручностей.

Карта номерів також виконує функцію оперативного контролю доступності. Користувач вводить дати заїзду та виїзду, після чого система перевіряє кожен номер і змінює його колір відповідно до результату. Такий спосіб відображення є зручним, оскільки адміністратор одразу бачить загальну картину по поверхах.

Рис. 2.5 Структура навігації графічного інтерфейсу

Модуль аналітики проєктується як спрощений дашборд. Він не замінює повноцінну систему бізнес-аналітики, однак дає користувачу швидке розуміння поточного стану. У ньому виводяться чотири основні показники: загальна кількість номерів, кількість активних бронювань, кількість доступних номерів на вибрані дати та плановий дохід за активними бронюваннями.

Таблиця 2.8

Показники аналітичного модуля

Показник	**Джерело даних**	**Призначення**
Усього номерів	Таблиця rooms	Оцінка розміру номерного фонду
Активні бронювання	Таблиця bookings	Поточне навантаження системи
Доступні номери	rooms + bookings	Оцінка вільного фонду на вибраний період
Плановий дохід	bookings.total_price	Оцінка очікуваної виручки за активними бронюваннями

Під час проєктування інтерфейсу було враховано потребу в мінімізації зайвих дій користувача. Наприклад, якщо адміністратор обрав номер у каталозі або на карті, він може одразу перенести його у форму бронювання. Це скорочує час роботи та зменшує ймовірність помилки під час ручного вибору номера.

Крім того, у системі передбачено картку номера та схематичний вигляд номера. Картка містить текстову інформацію, а схема на canvas умовно показує розміщення основних зон: ліжка, санвузла, столу, дивана або кухонної зони залежно від типу номера. Це не є архітектурним кресленням, але виконує демонстраційну функцію та робить програму більш завершеною.

Загалом проєктування інтерфейсу спрямоване на те, щоб система була не лише функціональною, а й зрозумілою для користувача. Вкладкова структура,

карта номерів, картка номера та аналітичний блок формують логічну модель роботи адміністратора готелю.

У другому розділі було здійснено проєктування інформаційної системи бронювання готельних номерів. Спочатку було сформульовано постановку задачі та визначено основні вимоги до системи. Встановлено, що програмний продукт має забезпечувати перегляд номерного фонду, створення бронювань, перевірку доступності номерів, роботу з клієнтами, скасування бронювань, експорт даних і перегляд базових аналітичних показників.

Було спроєктовано функціональну структуру системи, яка включає модуль каталогу номерів, карту номерів, модуль створення бронювання, модуль керування бронюваннями, аналітичний модуль і модуль роботи з базою даних. Такий поділ дозволяє логічно організувати програму та забезпечити зручність її подальшої реалізації.

Окрему увагу приділено проєктуванню бази даних. Було визначено три основні таблиці: rooms, clients і bookings. Таблиця rooms зберігає інформацію про номерний фонд, таблиця clients — дані клієнтів, а таблиця bookings — інформацію про бронювання. Зв’язок між цими таблицями забезпечує цілісну модель обліку номерів, клієнтів і періодів проживання.

Також було розроблено алгоритм перевірки доступності номера. Його основою є перевірка перетину дат між новим і вже наявними активними бронюваннями. Запропонований алгоритм дозволяє уникнути подвійного бронювання одного номера та забезпечує коректність роботи системи.

На завершальному етапі було описано проєктування графічного інтерфейсу користувача, карти номерів і модуля аналітики. Визначено, що інтерфейс системи має вкладкову структуру, а карта номерів забезпечує наочне представлення номерного фонду за поверхами. Аналітичний модуль дозволяє швидко оцінити кількість номерів, активні бронювання, доступність номерів і плановий дохід. Отже, результати проєктування створюють достатню основу для програмної реалізації інформаційної системи бронювання готельних номерів у наступному розділі.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Характеристика середовища розробки та використаних програмних засобів

Програмна реалізація інформаційної системи бронювання готельних номерів виконана мовою програмування Python. Основна мета практичної частини роботи полягає у створенні desktop-додатка, який дозволяє адміністратору готелю працювати з номерним фондом, створювати бронювання, перевіряти доступність номерів, переглядати карту номерів, скасовувати бронювання, експортувати дані та аналізувати основні показники роботи системи.

Для реалізації було обрано локальну desktop-архітектуру, оскільки вона є зручною для навчального проєкту, не потребує серверної частини, вебхостингу або складного налаштування середовища. Програма запускається безпосередньо на комп'ютері користувача, а всі дані зберігаються у локальному файлі бази даних hotel_booking.db.

У процесі розробки використано такі програмні засоби:

Таблиця 3.1

Використані програмні засоби для розробки інформаційної системи

Програмний засіб	Призначення у проєкті
Python 3.x	Основна мова програмування
Tkinter	Створення графічного інтерфейсу користувача
SQLite	Локальна база даних для збереження номерів, клієнтів і бронювань
Модуль sqlite3	Взаємодія Python-програми з базою SQLite
Модуль datetime	Перевірка дат заїзду та виїзду
Модуль csv	Експорт бронювань у CSV-файл
PowerShell / командний рядок	Запуск програми
Visual Studio Code / IDLE / PyCharm	Можливе середовище редагування коду

Вибір Python зумовлений тим, що ця мова дозволяє швидко розробляти прикладні програми, має зрозумілий синтаксис і підтримує роботу з графічним інтерфейсом та базами даних без необхідності встановлення великої кількості зовнішніх бібліотек. У межах цієї роботи Python використовується для опису основної логіки системи: створення вікна програми, обробки дій користувача, перевірки введених даних, виконання SQL-запитів, розрахунку вартості проживання та формування аналітичних показників.

Для створення інтерфейсу обрано бібліотеку Tkinter. Вона входить до стандартного набору Python, тому користувачеві не потрібно окремо встановлювати графічні фреймворки. За допомогою Tkinter реалізовано головне вікно програми, вкладки, таблиці, кнопки, поля введення, випадаючі списки, текстові блоки та графічні елементи для схематичного відображення номерів.

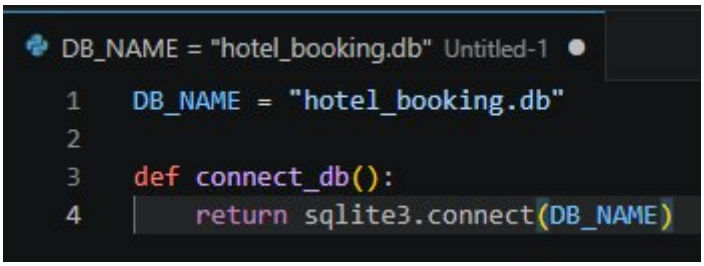
Для збереження інформації використано SQLite. Це вбудована база даних, яка зберігається в одному файлі й не потребує запуску окремого сервера. Такий підхід є достатнім для невеликої інформаційної системи бронювання, оскільки всі основні дані — номери, клієнти та бронювання — можна зберігати локально.

Структурно програма реалізована у вигляді одного основного Python-файлу hotel_booking_system_enhanced_fixed.py. У цьому файлі зосереджено функції роботи з базою даних, клас графічного інтерфейсу, логіку створення бронювання, механізм перевірки доступності номерів, модуль карти номерів, експорт у CSV та аналітичний блок.

3.2 Реалізація бази даних і механізму збереження інформації

База даних є основою інформаційної системи, оскільки саме в ній зберігаються всі відомості, необхідні для роботи програми. У розробленій системі база даних створюється автоматично під час першого запуску програми. Якщо файл hotel_booking.db відсутній, програма створює його та формує необхідні таблиці.

Для підключення до бази використовується функція connect_db(), яка створює з’єднання з файлом SQLite.



Лістинг 3.1 — Підключення до бази даних

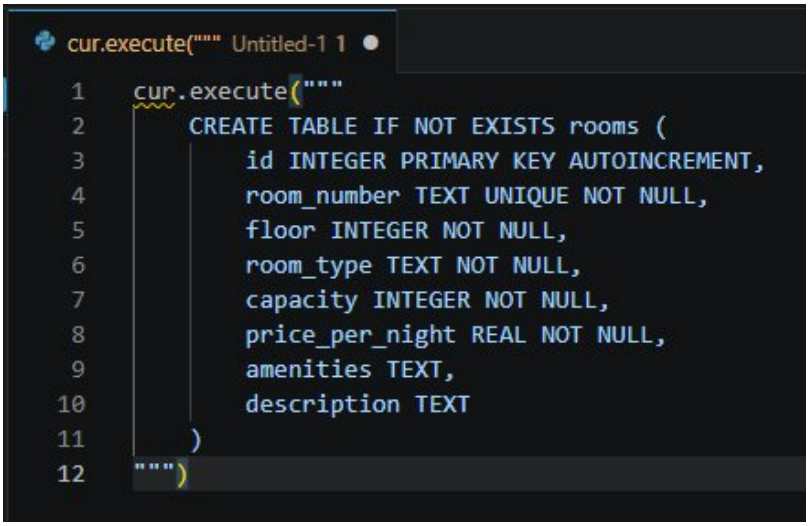
Після запуску програми викликається функція init_db(). Вона створює таблиці rooms, clients і bookings, якщо вони ще не існують. Також у системі передбачено механізм автоматичного оновлення старої бази даних. Це потрібно на випадок, якщо раніше запускалася базова версія програми, у якій ще не було полів floor, amenities, description або notes.

Таблиця 3.2

Таблиці бази даних інформаційної системи

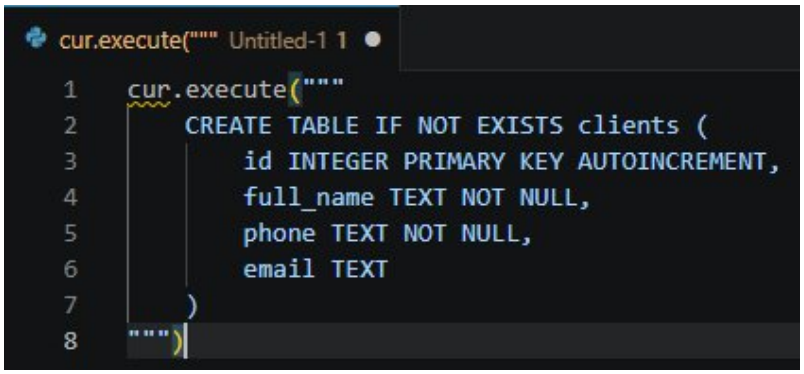
Таблиця	**Призначення**
rooms	Збереження інформації про готельні номери
clients	Збереження інформації про клієнтів
bookings	Збереження інформації про бронювання

Таблиця rooms містить інформацію про номерний фонд готелю. У ній зберігаються номер кімнати, поверх, тип номера, місткість, ціна за одну ніч, зручності та опис. Це дозволяє використовувати одну таблицю як для відображення каталогу номерів, так і для формування картки номера та карти поверхів.



Лістинг 3.2 Створення таблиці rooms

Таблиця clients призначена для збереження даних про клієнтів. У поточній версії програми вона містить ПІБ, телефон та електронну пошту. Такої структури достатньо для навчального прототипу системи бронювання.



Лістинг 3.3 Створення таблиці clients

Таблиця bookings є центральною таблицею системи. Вона зберігає інформацію про те, який клієнт забронював який номер, на який період, на яку суму та з яким статусом.

```
cur.execute(""" Untitled-1 1
cur.execute("""
    CREATE TABLE IF NOT EXISTS bookings (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        room_id INTEGER NOT NULL,
        client_id INTEGER NOT NULL,
        check_in TEXT NOT NULL,
        check_out TEXT NOT NULL,
        total_price REAL NOT NULL,
        status TEXT NOT NULL DEFAULT 'active',
        notes TEXT,
        created_at TEXT NOT NULL,
        FOREIGN KEY(room_id) REFERENCES rooms(id),
        FOREIGN KEY(client_id) REFERENCES clients(id)
```

Лістинг 3.4 Створення таблиці bookings

У таблиці bookings використано зовнішні ключі room_id і client_id. Поле room_id пов’язує бронювання з конкретним номером, а поле client_id — з конкретним клієнтом. Завдяки цьому база даних має логічну структуру, у якій бронювання не зберігається ізольовано, а пов’язане з іншими сутностями системи.

Під час першого запуску програма автоматично додає демонстраційні номери. Це зроблено для того, щоб після запуску користувач одразу міг тестувати систему, не створюючи вручну номерний фонд. У базу додаються номери різних категорій: «Стандарт», «Комфорт», «Люкс» та «Апартаменти».

Таблиця 3.3

Приклад демонстраційних номерів у системі

Номер	**Поверх**	**Тип**	**Місткість**	**Ціна за ніч, грн**
101	1	Стандарт	1	1200
102	1	Стандарт	2	1500
103	1	Комфорт	2	2100
201	2	Комфорт	2	2300
203	2	Люкс	2	3900
302	3	Апартаменти	4	6200
303	3	Апартаменти	4	6700

Таким чином, механізм збереження інформації реалізовано через локальну базу SQLite, яка автоматично створюється, оновлюється та наповнюється початковими даними. Це забезпечує стабільну роботу системи та дозволяє зберігати інформацію між різними запусками програми.

3.3 Реалізація модуля керування номерним фондом і карти номерів

Модуль керування номерним фондом призначений для перегляду всіх номерів готелю, фільтрації номерів за параметрами, відображення картки номера та переходу до створення бронювання. У програмі цей модуль реалізовано у вкладці «Каталог номерів».

У каталозі користувач бачить таблицю номерів, у якій відображаються такі дані: ідентифікатор, номер кімнати, поверх, тип номера, місткість, ціна за одну ніч і статус доступності. Для зручності адміністратора передбачено фільтри за датами, типом номера та мінімальною місткістю.

Таблиця 3.4

Основні елементи вкладки «Каталог номерів»

Елемент інтерфейсу	**Призначення**
Поле дати заїзду	Введення початкової дати проживання
Поле дати виїзду	Введення кінцевої дати проживання
Фільтр типу номера	Вибір категорії номера
Фільтр місткості	Відбір номерів за кількістю місць
Таблиця номерів	Перегляд номерного фонду
Картка номера	Детальний опис вибраного номера
Схема номера	Умовне графічне відображення планування

Для отримання номерів із бази використовується функція fetch_rooms(). Вона виконує SQL-запит до таблиці rooms і повертає список номерів, упорядкованих за поверхом і номером кімнати.

```python
def fetch_rooms(self):
    conn = connect_db()
    cur = conn.cursor()
    cur.execute("""
        SELECT id, room_number, floor, room_type, capacity,
               price_per_night, amenities, description
        FROM rooms
        ORDER BY floor, room_number
    """)
    data = cur.fetchall()
    conn.close()
    return data
```

Лістинг 3.5 Отримання списку номерів із бази даних

Після вибору номера в таблиці програма відображає його картку. У картці показуються номер кімнати, поверх, тип, місткість, ціна, опис та перелік зручностей. Це дозволяє адміністратору швидко отримати повну інформацію про номер без переходу до інших вікон.

Окремо реалізовано схематичний вигляд номера. Для цього використовується елемент Canvas, на якому програма малює умовне планування приміщення. Залежно від типу номера змінюється набір зон: для стандартного номера відображається ліжко, стіл, санвузол і шафа; для люксу — спальня та вітальня; для апартаментів — спальня, зона відпочинку та кухонна зона.

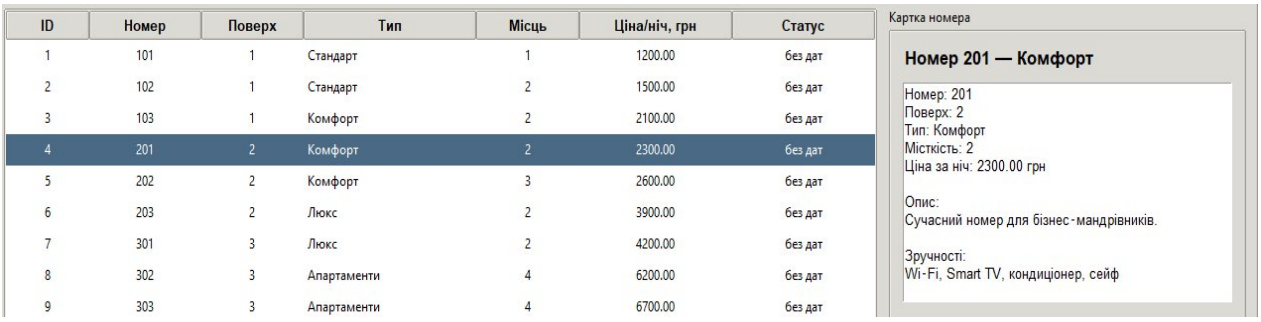


Рис.3.1 Вкладка «Каталог номерів» інформаційної системи

Другим важливим елементом модуля керування номерним фондом є карта номерів. Вона реалізована у вкладці «Карта номерів». Карта відображає номери за поверхами у вигляді кнопок. Кожна кнопка відповідає окремому номеру.

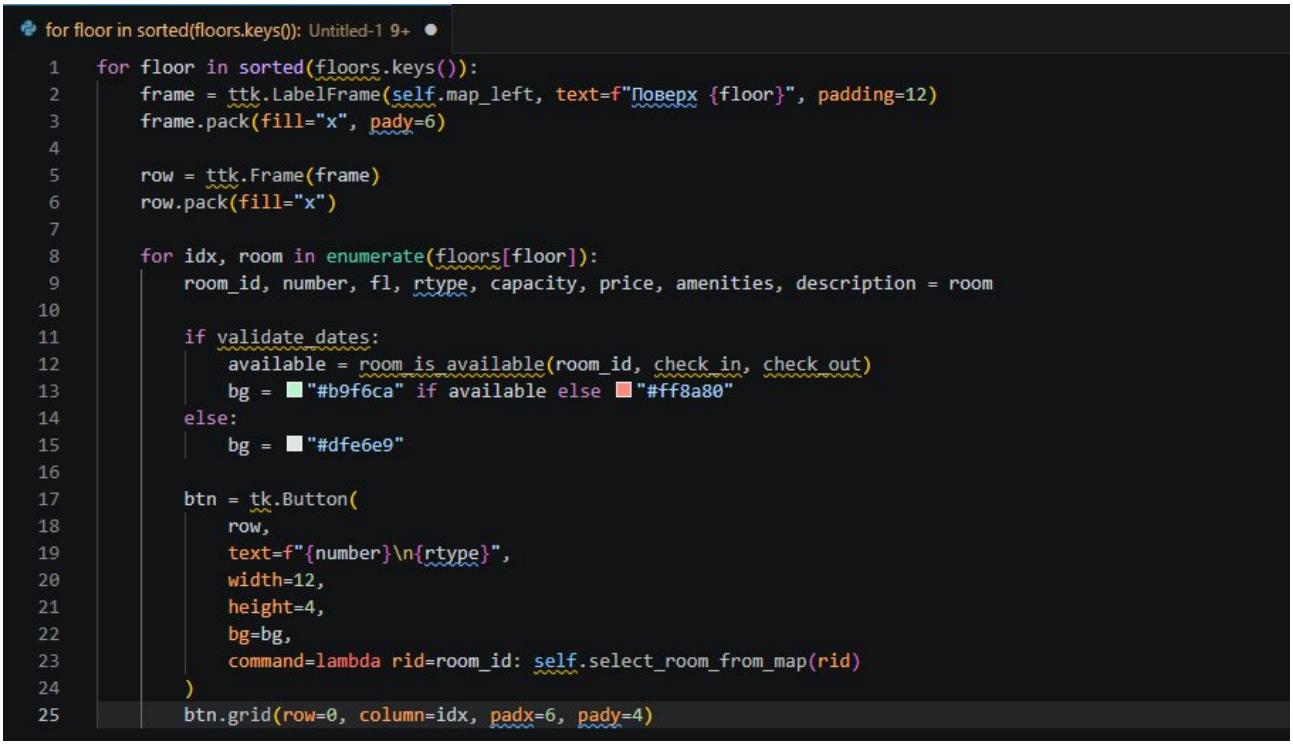
Колір кнопки залежить від доступності номера на вибрані дати:

Таблиця 3.5

Кольорове позначення номерів на карті

Колір	Значення
Зелений	Номер вільний на вибраний період
Червоний	Номер зайнятий на вибраний період
Сірий	Дати не задані або перевірка не виконана

Формування карти номерів відбувається динамічно. Програма отримує всі номери з бази, групує їх за поверхами, а потім створює для кожного номера окрему кнопку. При натисканні на кнопку в правій частині вікна відображається інформація про вибраний номер.



Лістинг 3.6 Фрагмент створення карти номерів

Карта номерів робить систему більш наочною та зручною. Адміністратор може не тільки переглядати номери в таблиці, а й бачити їхнє розміщення за поверхами. Це наближує програму до реальних готельних систем, у яких часто використовуються графічні схеми номерного фонду.

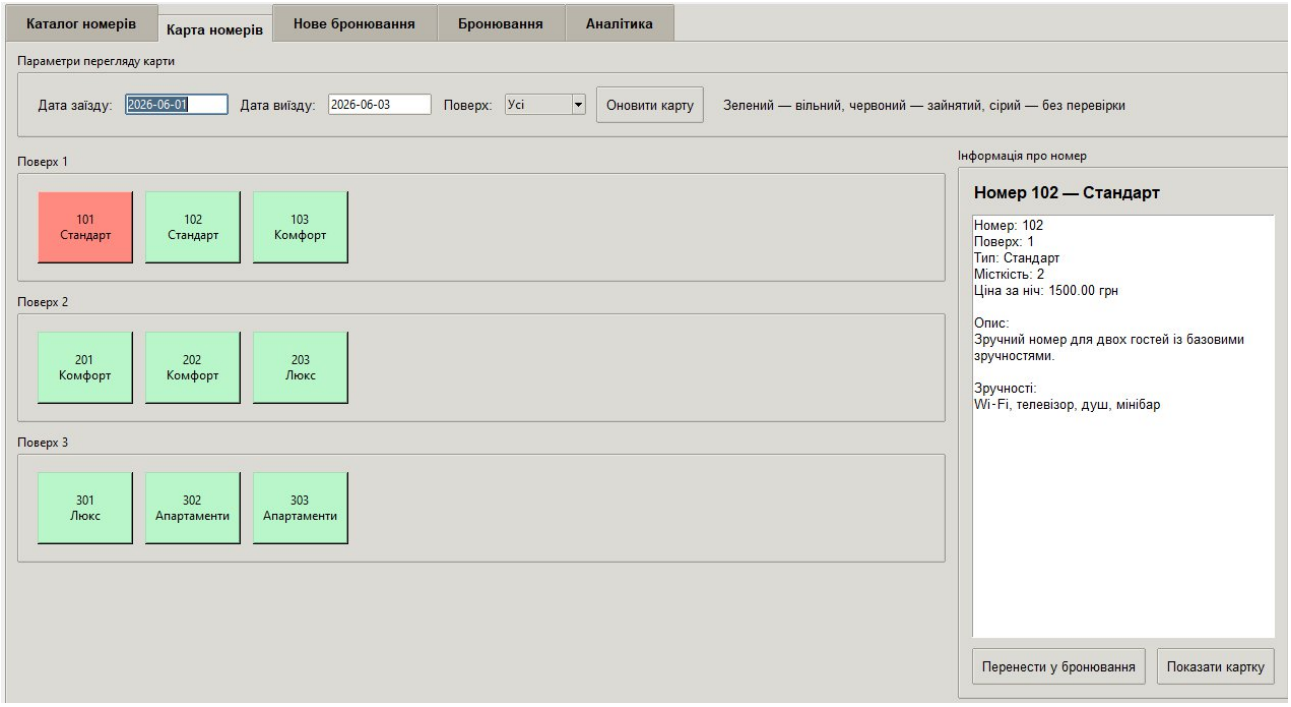


Рис. 3.2 Вкладка «Карта номерів» із відображенням поверхів

3.4. Реалізація модуля створення, пошуку, скасування та експорту бронювань

Модуль бронювання є центральною частиною інформаційної системи, оскільки саме він забезпечує створення записів про проживання клієнтів. У програмі цей модуль реалізовано у вкладці «Нове бронювання». Користувач обирає номер, вводить дані клієнта, зазначає дати заїзду та виїзду, додає примітки, перевіряє доступність номера та створює бронювання.

Перед створенням бронювання система виконує кілька перевірок. По-перше, перевіряється, чи обрано номер. По-друге, контролюється заповнення обов’язкових полів: ПІБ клієнта та телефон. По-третє, перевіряється коректність дат. Дата виїзду повинна бути пізнішою за дату заїзду. По-четверте, система перевіряє доступність номера у вказаний період.

Таблиця 3.6

Перевірки під час створення бронювання

Перевірка	**Умова**	**Результат у разі помилки**
Вибір номера	Номер має бути обраний зі списку	Повідомлення про необхідність вибору номера
ПІБ клієнта	Поле не повинно бути порожнім	Повідомлення про необхідність введення ПІБ
Телефон	Поле не повинно бути порожнім	Повідомлення про необхідність введення телефону
Дати проживання	Дата виїзду пізніша за дату заїзду	Повідомлення про некоректні дати
Доступність номера	Номер не зайнятий у вибраний період	Повідомлення, що номер уже зайнятий

Ключова функція цього модуля — create_booking(). Вона отримує дані з форми, перевіряє їх, обчислює вартість проживання, перевіряє доступність номера та записує інформацію в базу даних.

```python
def create_booking(self):
    room_id = self.get_selected_room_id_from_combo()
    full_name = self.client_name_entry.get().strip()
    phone = self.client_phone_entry.get().strip()
    email = self.client_email_entry.get().strip()
    check_in = self.booking_check_in.get().strip()
    check_out = self.booking_check_out.get().strip()

    if not full_name:
        raise ValueError("Введіть ПІБ клієнта.")

    if not phone:
        raise ValueError("Введіть телефон клієнта.")

    total = self.calculate_booking_price()

    if not room_is_available(room_id, check_in, check_out):
        raise ValueError("Номер уже зайнятий у цей період.")
```

Лістинг 3.7 Фрагмент створення бронювання

Після проходження всіх перевірок програма створює запис у таблиці clients, а потім створює відповідний запис у таблиці bookings.

```python
cur.execute(
    "INSERT INTO clients (full_name, phone, email) VALUES (?, ?, ?)",
    (full_name, phone, email)
)

client_id = cur.lastrowid

cur.execute("""
    INSERT INTO bookings
    (room_id, client_id, check_in, check_out, total_price, status, notes, created_at)
    VALUES (?, ?, ?, ?, ?, 'active', ?, ?)
""", (
    room_id,
    client_id,
    check_in,
    check_out,
    total,
    notes,
    datetime.now().strftime("%Y-%m-%d %H:%M:%S")
))
```

Лістинг 3.8 Запис клієнта та бронювання до бази даних

Розрахунок вартості проживання виконується за простою формулою:

Вартість = кількість ночей × ціна за одну ніч.

Для визначення кількості ночей використовується модуль datetime. Якщо різниця між датами менша або дорівнює нулю, система повідомляє про помилку.

```
def nights_between(check_in, check_out): Untitled-1 2
1   def nights_between(check_in, check_out):
2       start = parse_date(check_in)
3       end = parse_date(check_out)
4       nights = (end - start).days
5
6       if nights <= 0:
7           raise ValueError("Дата виїзду має бути пізнішою за дату заїзду.")
8
9       return nights
```

Лістинг 3.9 Розрахунок кількості ночей

Пошук, перегляд і скасування бронювань реалізовано у вкладці «Бронювання». У ній відображається таблиця всіх бронювань із зазначенням номера, клієнта, телефону, дат проживання, суми та статусу. Користувач може виконати пошук за ПІБ клієнта або номером кімнати, а також відфільтрувати записи за статусом.

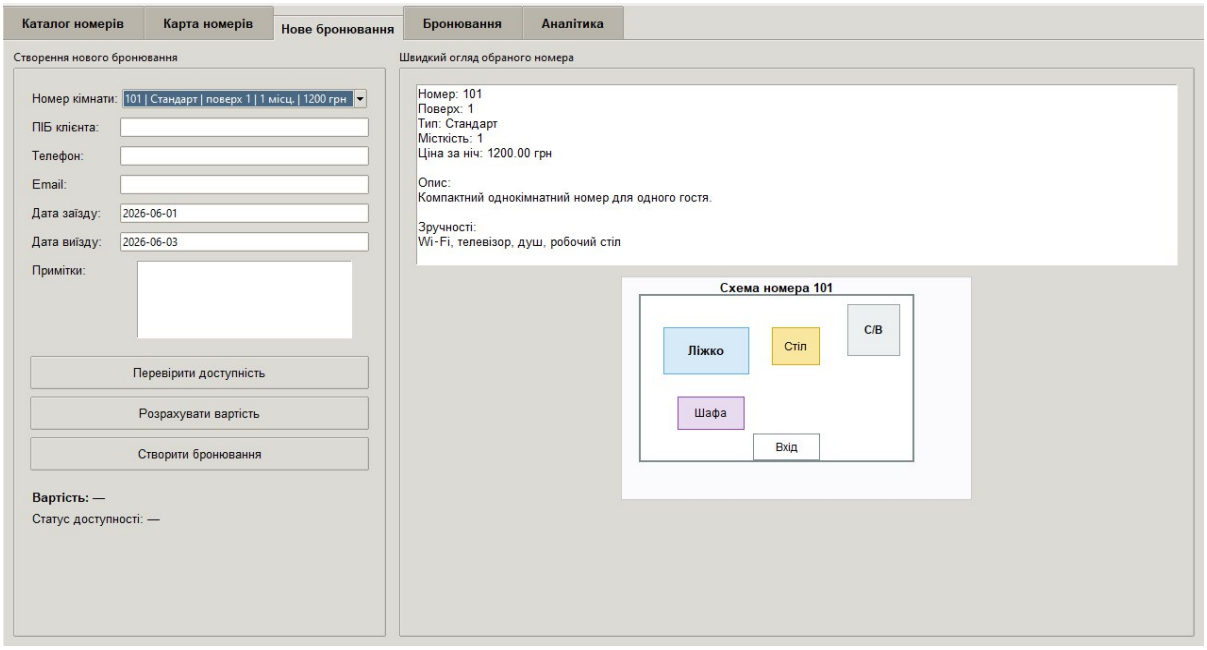


Рис. 3.3 Вкладка створення нового бронювання

Скасування бронювання реалізовано не через видалення запису, а через зміну його статусу з active на cancelled. Такий підхід є правильнішим, оскільки дозволяє зберігати історію дій у системі.

Додатковою функцією є експорт бронювань у CSV-файл. Це дає змогу зберігати дані у форматі, який можна відкрити в Excel або іншому табличному редакторі. Експорт може використовуватися для створення звітів, архівування або подальшого аналізу.

3.5 Реалізація аналітичного модуля інформаційної системи

Аналітичний модуль призначений для швидкого перегляду основних показників роботи системи. Він реалізований у вкладці **«Аналітика»**. На відміну від інших вкладок, цей модуль не створює нові записи, а опрацьовує вже наявні дані з бази та формує короткий інформаційний огляд.

У поточній версії системи аналітичний модуль відображає чотири основні показники:

Таблиця 3.7

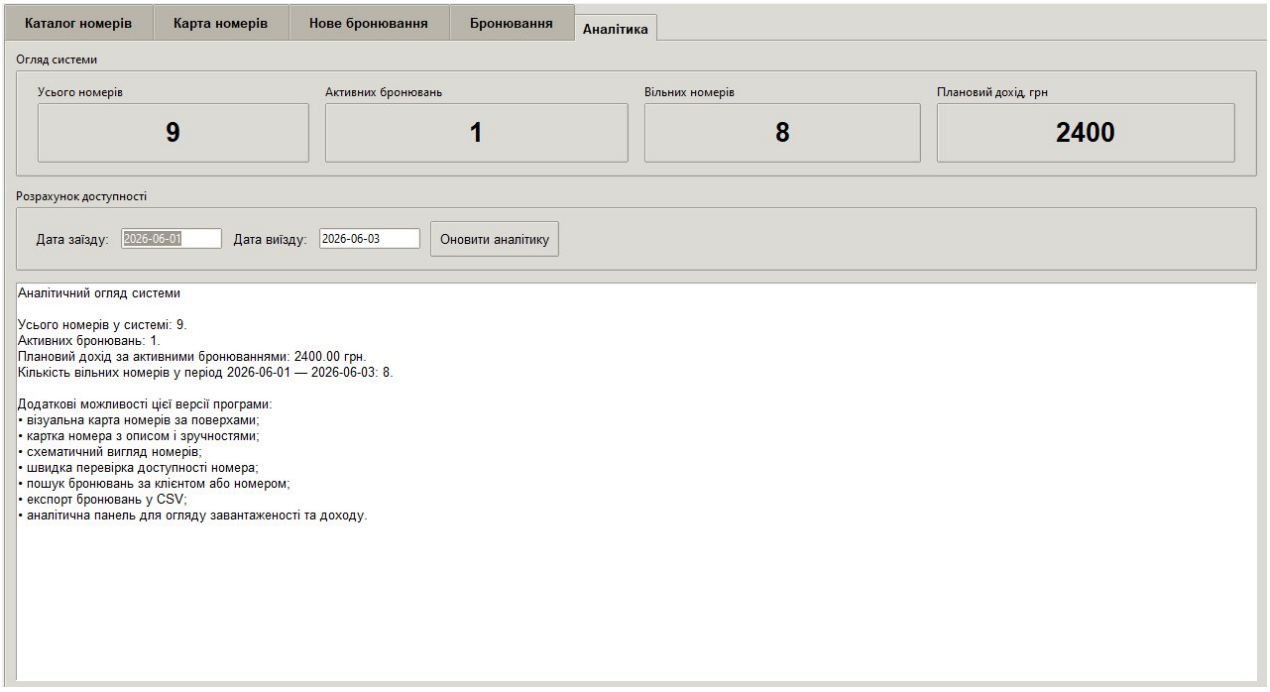
Показники аналітичного модуля

Показник	**Джерело даних**	**Призначення**
Усього номерів	Таблиця rooms	Визначення розміру номерного фонду
Активних бронювань	Таблиця bookings	Оцінка поточного навантаження
Вільних номерів	Таблиці rooms і bookings	Перевірка доступності на вибраний період
Плановий дохід	Поле total_price у bookings	Оцінка очікуваної виручки

Загальна кількість номерів визначається простим SQL-запитом до таблиці rooms. Кількість активних бронювань визначається за таблицею bookings, де статус дорівнює active. Плановий дохід розраховується як сума вартості всіх активних бронювань.

Кількість доступних номерів визначається складніше, оскільки вона залежить від вибраного користувачем періоду. Система отримує всі номери з бази та для кожного з них виконує перевірку доступності за алгоритмом перетину дат. Після цього підраховується кількість вільних номерів.

Аналітичний модуль не є повноцінною системою бізнес-аналітики, однак він виконує важливу функцію швидкого контролю. Адміністратор може одразу побачити, скільки номерів є в системі, скільки бронювань активні, скільки номерів доступні на конкретні дати та яку суму становить очікуваний дохід.



Наявність такого модуля підвищує практичну цінність системи. Навіть у невеликому готелі адміністратору важливо швидко оцінювати поточний стан номерного фонду. У майбутньому цей модуль можна розширити, додавши

графіки завантаженості, статистику за місяцями, аналіз популярності типів номерів і прогноз доходу.

3.6 Тестування працездатності програми та аналіз результатів

Після реалізації основних модулів було проведено тестування інформаційної системи. Метою тестування є перевірка правильності роботи програми, виявлення можливих помилок і підтвердження того, що система виконує заявлені функції.

Тестування проводилося вручну шляхом запуску програми, введення тестових даних і перевірки результатів у графічному інтерфейсі. Основна увага приділялася таким функціям: запуск програми, створення бази даних, перегляд номерів, фільтрація, карта номерів, створення бронювання, блокування подвійного бронювання, пошук, скасування, експорт і аналітика.

Таблиця 3.8

Результати функціонального тестування інформаційної системи

№	Тестова дія	Очікуваний результат	Фактичний результат	Статус
1	Запуск програми	Відкривається головне вікно	Головне вікно відкривається	Успішно
2	Перше відкриття програми	Створюється база hotel_booking.db	База створюється автоматично	Успішно
3	Перегляд каталогу номерів	Відображаються демонстраційні номери	Номери показані в таблиці	Успішно
4	Фільтрація за типом номера	Показуються лише номери вибраного типу	Фільтрація працює	Успішно
5	Перегляд картки номера	Виводиться опис і зручності	Картка відображається	Успішно

Продовження таблиці 3.8

6	Перегляд карти номерів	Номери групуються за поверхами	Карта формується	Успішно
7	Перевірка доступності номера	Система визначає вільний/зайнятий номер	Перевірка працює	Успішно
8	Створення бронювання	Запис додається до бази	Бронювання створюється	Успішно
9	Спроба подвійного бронювання	Система блокує запис	Виводиться повідомлення про помилку	Успішно
10	Скасування бронювання	Статус змінюється на cancelled	Бронювання скасовано	Успішно
11	Пошук бронювання	Знаходиться запис за ПІБ або номером	Пошук працює	Успішно
12	Експорт у CSV	Створюється CSV-файл	Файл збережено	Успішно
13	Оновлення аналітики	Показники перераховуються	Аналітика оновлюється	Успішно

Окремо перевірялася коректність роботи алгоритму доступності номерів. Було створено тестове бронювання для номера 201 на період з 2026-06-01 до 2026-06-05. Після цього система перевіряла спроби створення нових бронювань для цього самого номера на різні дати.

Таблиця 3.9

Тестування алгоритму перевірки доступності номера

Наявне бронювання	**Нове бронювання**	**Очікуваний результат**	**Фактичний результат**
01.06–05.06	05.06–08.06	Дозволити бронювання	Дозволено
01.06–05.06	04.06–07.06	Заборонити бронювання	Заборонено

Продовження таблиці 3.9

01.06–05.06	25.05–01.06	Дозволити бронювання	Дозволено
01.06–05.06	02.06–03.06	Заборонити бронювання	Заборонено
01.06–05.06	30.05–10.06	Заборонити бронювання	Заборонено

Результати тестування підтвердили, що алгоритм правильно визначає перетин дат. Система не допускає подвійного бронювання одного номера на періоди, які перетинаються, але дозволяє створювати бронювання, якщо попереднє завершується в день нового заїзду.

Також було протестовано обробку некоректних даних. Якщо користувач залишає порожнім поле ПІБ або телефону, система виводить повідомлення про помилку. Якщо дата виїзду є ранішою за дату заїзду або дорівнює їй, програма не створює бронювання та повідомляє про некоректність дат.

Таблиця 3.10

Тестування обробки помилкових введень

Помилкова ситуація	**Реакція системи**
Не введено ПІБ клієнта	Повідомлення: «Введіть ПІБ клієнта»
Не введено телефон	Повідомлення: «Введіть телефон клієнта»
Дата виїзду раніше дати заїзду	Повідомлення про некоректні дати
Не обрано номер	Повідомлення про необхідність вибору номера
Номер зайнятий	Повідомлення про неможливість створення бронювання

За результатами тестування можна зробити висновок, що програма працює стабільно в межах заявленої функціональності. Усі основні модулі виконують свої завдання, дані зберігаються в базі SQLite, інтерфейс реагує на дії користувача, а логіка перевірки бронювань запобігає конфліктам.

3.7 Інструкція користувача щодо запуску та роботи з програмою

Для запуску інформаційної системи користувачеві потрібно мати встановлений Python версії 3.10 або новішої. Програма не потребує додаткових бібліотек, оскільки використовує стандартні модулі Python: tkinter, sqlite3, datetime і csv.

Файл програми має назву:

hotel_booking_system_enhanced_fixed.py

Для запуску потрібно відкрити папку з програмою, перейти до неї через командний рядок або PowerShell і виконати команду:

python hotel_booking_system_enhanced_fixed.py

Якщо у Windows команда python не спрацьовує, можна використати команду:

py hotel_booking_system_enhanced_fixed.py

Після запуску відкривається головне вікно програми з вкладками: «Каталог номерів», «Карта номерів», «Нове бронювання», «Бронювання» та «Аналітика».

Таблиця 3.11

Послідовність роботи користувача з програмою

Крок	**Дія користувача**	**Результат**
1	Запустити програму	Відкривається головне вікно
2	Перейти до вкладки «Каталог номерів»	Відображається список номерів
3	За потреби ввести дати та фільтри	Система показує відповідні номери
4	Обрати номер	Відображається картка номера
5	Перейти до вкладки «Нове бронювання»	Відкривається форма бронювання
6	Ввести дані клієнта і дати	Дані готові до перевірки
7	Натиснути «Перевірити доступність»	Система показує, чи вільний номер
8	Натиснути «Розрахувати вартість»	Виводиться сума проживання
9	Натиснути «Створити бронювання»	Бронювання записується в базу
10	Перейти до вкладки «Бронювання»	Можна переглянути або скасувати запис
11	Перейти до вкладки «Аналітика»	Виводяться загальні показники

Під час роботи з датами потрібно використовувати формат:

YYYY-MM-DD

Наприклад:

2026-06-01

Якщо користувач введе дату в неправильному форматі, програма не зможе виконати перевірку та повідомить про помилку.

Для створення бронювання потрібно обрати номер, ввести ПІБ клієнта, телефон, за потреби електронну пошту, дату заїзду, дату виїзду та примітку.

Після цього користувач може перевірити доступність номера. Якщо номер вільний, система дозволить створити бронювання. Якщо номер зайнятий, програма виведе повідомлення про неможливість бронювання.

Для скасування бронювання потрібно перейти до вкладки «Бронювання», обрати потрібний запис у таблиці та натиснути кнопку «Скасувати обране». Після підтвердження статус бронювання зміниться на «скасоване». Запис не видаляється з бази, що дозволяє зберігати історію бронювань.

Для експорту даних потрібно натиснути кнопку «Експорт у CSV», вибрати місце збереження файлу та підтвердити операцію. Створений CSV-файл можна відкрити в Microsoft Excel або іншому табличному редакторі.

Якщо під час запуску виникає помилка, пов’язана з базою даних, можна видалити файл hotel_booking.db і запустити програму повторно. Однак у виправленій версії програми передбачено механізм автоматичного оновлення структури бази, тому така дія зазвичай не потрібна.

У третьому розділі було розглянуто програмну реалізацію та тестування інформаційної системи бронювання готельних номерів. Нами було описано середовище розробки, використані програмні засоби та обґрунтовано практичне застосування Python, Tkinter і SQLite для створення desktop-додатка.

Було реалізовано базу даних, яка містить таблиці rooms, clients і bookings. Така структура дозволяє зберігати інформацію про номерний фонд, клієнтів і бронювання. Додатково передбачено автоматичне створення бази під час першого запуску та оновлення її структури у разі використання старішої версії.

У межах програмної реалізації створено модуль керування номерним фондом, який дає змогу переглядати номери, фільтрувати їх, відкривати картку номера та переглядати схематичний вигляд приміщення. Також реалізовано карту номерів за поверхами, яка візуально показує доступність номерів на вибрані дати.

Центральним елементом системи став модуль бронювання. Він забезпечує введення даних клієнта, вибір номера, перевірку дат, перевірку доступності, розрахунок вартості проживання, створення бронювання, пошук записів,

скасування бронювань і експорт даних у CSV. Особливе значення має алгоритм перевірки перетину дат, який запобігає подвійному бронюванню одного номера.

Також було реалізовано аналітичний модуль, який відображає загальну кількість номерів, кількість активних бронювань, кількість доступних номерів у вибраний період і плановий дохід. Це дозволяє адміністратору швидко оцінити поточний стан системи.

Проведене тестування підтвердило працездатність програми. Було перевірено запуск системи, створення бази даних, перегляд номерів, роботу карти номерів, створення бронювань, блокування подвійного бронювання, скасування записів, експорт у CSV та оновлення аналітики. За результатами тестування встановлено, що програма виконує поставлені завдання та може бути використана як навчальний прототип інформаційної системи бронювання готельних номерів.

ВИСНОВКИ

У дипломній роботі було досліджено теоретичні засади, спроєктовано та програмно реалізовано інформаційну систему бронювання готельних номерів з використанням мови програмування Python. У процесі виконання роботи було досягнуто поставленої мети, яка полягала у створенні працездатного desktop-додатка для автоматизації обліку номерного фонду, роботи з клієнтами, створення та скасування бронювань, перевірки доступності номерів, візуалізації карти номерів і формування базових аналітичних показників.

У першому розділі було розглянуто поняття, призначення та класифікацію інформаційних систем у сфері готельного обслуговування. Встановлено, що інформаційні системи є важливим інструментом підвищення ефективності діяльності готелю, оскільки дозволяють автоматизувати процеси обліку номерного фонду, реєстрації клієнтів, контролю бронювань, аналізу завантаженості та формування управлінської інформації. Було визначено, що для готельного бізнесу особливо важливими є системи бронювання та PMS-рішення, які забезпечують централізовану роботу з даними й допомагають уникати помилок, характерних для ручного ведення обліку.

Охарактеризовано особливості організації процесу бронювання готельних номерів. З’ясовано, що бронювання є одним із ключових бізнес-процесів готелю, оскільки саме воно забезпечує попереднє закріплення номера за клієнтом на визначений період. Процес бронювання охоплює вибір номера, перевірку доступності, внесення даних клієнта, фіксацію дат заїзду та виїзду, розрахунок вартості проживання, створення запису в базі даних і подальше керування статусом бронювання. Особливу увагу в роботі приділено проблемі подвійного бронювання, яка виникає у разі відсутності автоматичної перевірки перетину дат.

Було проаналізовано сучасні програмні рішення для автоматизації бронювання номерного фонду. На основі аналізу встановлено, що сучасні готельні системи орієнтовані не лише на збереження інформації про

бронювання, а й на комплексне управління номерним фондом, каналами продажу, клієнтськими даними, оплатами та аналітикою. Водночас для навчального проєкту доцільним є створення локального desktop-прототипу, який реалізує базові функції таких систем: каталог номерів, карту номерного фонду, створення бронювань, пошук, скасування, експорт і аналітичний огляд.

Обґрунтовано вибір Python, Tkinter та SQLite для розробки інформаційної системи. Python було обрано як зручну високорівневу мову програмування, яка дозволяє швидко створювати прикладні програми. Tkinter використано для розробки графічного інтерфейсу користувача, оскільки ця бібліотека входить до стандартного набору Python і не потребує додаткового встановлення. SQLite обрано як локальну базу даних, що не потребує окремого сервера й добре підходить для desktop-додатків навчального типу.

У другому розділі було виконано проєктування інформаційної системи бронювання готельних номерів. Сформульовано постановку задачі та визначено основні вимоги до системи. Встановлено, що розроблюваний програмний продукт має забезпечувати перегляд номерного фонду, фільтрацію номерів, створення бронювань, перевірку доступності, роботу з клієнтами, скасування записів, експорт даних і перегляд базових аналітичних показників. Також було визначено, що система повинна мати простий і зрозумілий графічний інтерфейс, орієнтований на роботу адміністратора готелю.

Спроєктовано функціональну структуру інформаційної системи. До її складу включено модуль каталогу номерів, модуль карти номерів, модуль створення бронювання, модуль керування бронюваннями, аналітичний модуль і модуль роботи з базою даних. Такий поділ дозволив логічно організувати роботу програми й забезпечити зручність подальшої програмної реалізації. Особливістю системи є наявність карти номерів за поверхами, яка робить інтерфейс більш наочним і наближає програму до реальних готельних систем.

Розроблено структуру бази даних для обліку номерів, клієнтів і бронювань. База даних містить три основні таблиці: rooms, clients і bookings. Таблиця rooms зберігає інформацію про готельні номери, їхній тип, поверх, місткість, ціну, опис

і зручності. Таблиця clients містить контактні дані клієнтів. Таблиця bookings забезпечує зв’язок між клієнтом і номером, фіксує період проживання, загальну вартість, статус бронювання та дату створення запису. Така структура забезпечує цілісне збереження інформації й підтримує основну логіку роботи системи.

Описано алгоритм перевірки доступності номерів. Його сутність полягає у перевірці перетину дат нового бронювання з уже наявними активними бронюваннями для того самого номера. Якщо періоди перетинаються, система блокує створення нового запису. Якщо конфлікту немає, бронювання може бути створене. Такий алгоритм дозволяє уникнути подвійного бронювання та забезпечує коректність роботи інформаційної системи.

У третьому розділі було здійснено програмну реалізацію інформаційної системи. Розроблено desktop-додаток мовою Python із використанням Tkinter для графічного інтерфейсу та SQLite для локального збереження даних. Програма реалізована у вигляді єдиного застосунку, який містить вкладки «Каталог номерів», «Карта номерів», «Нове бронювання», «Бронювання» та «Аналітика». Така структура забезпечує послідовну й логічну роботу користувача із системою.

Було реалізовано механізм створення та оновлення бази даних. Під час першого запуску програма автоматично створює файл hotel_booking.db і необхідні таблиці. Крім того, передбачено механізм оновлення структури бази у випадку, якщо користувач раніше запускав базову версію програми. Це підвищує стабільність роботи застосунку та зменшує ризик помилок під час запуску.

Реалізовано модуль керування номерним фондом і карту номерів. У каталозі номерів користувач може переглядати всі номери, фільтрувати їх за типом, місткістю та датами, відкривати картку номера й переглядати його схематичне зображення. Карта номерів відображає номери за поверхами та використовує кольорове позначення для показу їхньої доступності. Це підвищує зручність роботи адміністратора й дозволяє швидко оцінити стан номерного фонду.

Розроблено модуль створення, пошуку, скасування та експорту бронювань. Система дозволяє створювати нові бронювання, вводити дані клієнта, перевіряти доступність номера, розраховувати вартість проживання, зберігати бронювання в базі даних, шукати записи за ПІБ клієнта або номером кімнати, скасовувати активні бронювання та експортувати дані у CSV-файл. Скасування реалізовано шляхом зміни статусу бронювання, а не видалення запису, що дозволяє зберігати історію операцій.

Реалізовано аналітичний модуль інформаційної системи. Він відображає загальну кількість номерів, кількість активних бронювань, кількість доступних номерів на вибрані дати та плановий дохід за активними бронюваннями. Хоча цей модуль має базовий характер, він підвищує практичну цінність програми, оскільки дає змогу адміністратору швидко оцінити поточний стан системи.

Проведено тестування працездатності програми. Було перевірено запуск застосунку, створення бази даних, відображення каталогу номерів, роботу фільтрів, карту номерів, створення бронювань, перевірку доступності, блокування подвійного бронювання, пошук, скасування, експорт у CSV та оновлення аналітичних показників. Результати тестування підтвердили, що програма працює стабільно в межах заявленої функціональності та коректно реагує на типові помилки користувача, зокрема незаповнені поля, некоректні дати або спробу забронювати вже зайнятий номер.

Практичне значення роботи полягає в тому, що було створено працездатний програмний прототип інформаційної системи бронювання готельних номерів. Розроблена система може бути використана як навчальний приклад реалізації прикладної інформаційної системи, а також як основа для подальшого розвитку. У майбутньому програму можна доповнити авторизацією користувачів, ролями адміністратора та менеджера, календарем бронювань, модулем оплат, формуванням рахунків, експортом звітів у PDF, вебінтерфейсом або інтеграцією з онлайн-сервісами бронювання.

ПЕРЕЛІК ПОСИЛАНЬ

1. Про туризм : Закон України від 15.09.1995 № 324/95-ВР. URL: https://zakon.rada.gov.ua/laws/show/324/95-%D0%B2%D1%80 (дата звернення: 15.05.2026).
2. Про захист персональних даних : Закон України від 01.06.2010 № 2297-VI. URL: https://zakon.rada.gov.ua/laws/show/2297-17 (дата звернення: 15.05.2026).
3. Правила користування готелями й аналогічними засобами розміщення та надання готельних послуг : наказ Державної туристичної адміністрації України від 16.03.2004 № 19. URL: https://zakon.rada.gov.ua/laws/show/z0413-04 (дата звернення: 15.05.2026).
4. ДСТУ 4268:2003. Послуги туристичні. Засоби розміщування. Загальні вимоги. Київ : Держспоживстандарт України, 2004. 13 с.
5. ДСТУ 4269:2003. Послуги туристичні. Класифікація готелів. Київ : Держспоживстандарт України, 2004. 15 с.
6. Buhalis D., Law R. Progress in information technology and tourism management: 20 years on and 10 years after the Internet. Tourism Management. 2008. Vol. 29, № 4. P. 609–623.
7. Buhalis D., Leung R. Smart hospitality — Interconnectivity and interoperability towards an ecosystem. International Journal of Hospitality Management. 2018. Vol. 71. P. 41–50.
8. Walker J. R. Introduction to Hospitality. 8th ed. Harlow : Pearson, 2021. 432 p.
9. Ivanov S. Hotel Revenue Management: From Theory to Practice. Varna : Zangador, 2014. 204 p.
10. Laudon K. C., Laudon J. P. Management Information Systems: Managing the Digital Firm. 17th ed. Harlow : Pearson, 2022. 656 p.
11. O’Brien J. A., Marakas G. M. Management Information Systems. 10th ed. New York : McGraw-Hill/Irwin, 2011. 711 p.
12. Sommerville I. Software Engineering. 10th ed. Harlow : Pearson, 2016. 816 p.

13. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2020. 705 p.
14. Fowler M. Patterns of Enterprise Application Architecture. Boston : Addison-Wesley, 2002. 560 p.
15. Connolly T., Begg C. Database Systems: A Practical Approach to Design, Implementation, and Management. 6th ed. Harlow : Pearson, 2015. 1440 p.
16. Silberschatz A., Korth H. F., Sudarshan S. Database System Concepts. 7th ed. New York : McGraw-Hill Education, 2020. 1376 p.
17. Coronel C., Morris S. Database Systems: Design, Implementation, & Management. 13th ed. Boston : Cengage Learning, 2019. 816 p.
18. Date C. J. Database Design and Relational Theory: Normal Forms and All That Jazz. 2nd ed. Sebastopol : O'Reilly Media, 2019. 448 p.
19. Python Software Foundation. Python 3 Documentation. URL: https://docs.python.org/3/ (дата звернення: 15.05.2026).
20. Python Software Foundation. tkinter — Python interface to Tcl/Tk. URL: https://docs.python.org/3/library/tkinter.html (дата звернення: 15.05.2026).
21. Python Software Foundation. sqlite3 — DB-API 2.0 interface for SQLite databases. URL: https://docs.python.org/3/library/sqlite3.html (дата звернення: 15.05.2026).
22. Python Software Foundation. datetime — Basic date and time types. URL: https://docs.python.org/3/library/datetime.html (дата звернення: 15.05.2026).
23. Python Software Foundation. csv — CSV File Reading and Writing. URL: https://docs.python.org/3/library/csv.html (дата звернення: 15.05.2026).
24. SQLite. About SQLite. URL: https://sqlite.org/about.html (дата звернення: 15.05.2026).
25. SQLite. SQLite Documentation. URL: https://sqlite.org/docs.html (дата звернення: 15.05.2026).
26. SQLite. Query Language Understood by SQLite. URL: https://sqlite.org/lang.html (дата звернення: 15.05.2026).

27. Matthes E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. 3rd ed. San Francisco : No Starch Press, 2023. 552 p.
28. Lutz M. Learning Python. 5th ed. Sebastopol : O'Reilly Media, 2013. 1648 p.
29. Oracle. Hotel Cloud Property Management System (PMS). URL: https://www.oracle.com/ua/hospitality/hotel-property-management/hotel-pms-software/ (дата звернення: 15.05.2026).
30. Oracle. Hotel Property Management and POS Solutions. URL: https://www.oracle.com/ua/hospitality/hotel-property-management/ (дата звернення: 15.05.2026).
31. Oracle. OPERA Hotel Property Management Solutions (PMS). URL: https://www.oracle.com/ua/hospitality/opera-property-services/ (дата звернення: 15.05.2026).
32. Cloudbeds. Hospitality Management System. URL: https://www.cloudbeds.com/ (дата звернення: 15.05.2026).
33. Cloudbeds. Commission-Free Booking Engine. URL: https://www.cloudbeds.com/booking-engine/ (дата звернення: 15.05.2026).
34. Cloudbeds Developers. Booking Engine. URL: https://developers.cloudbeds.com/docs/booking-engine (дата звернення: 15.05.2026).
35. Nielsen J., Budiu R. Mobile Usability. Berkeley : New Riders, 2013. 216 p.
36. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. 3rd ed. Berkeley : New Riders, 2014. 216 p.
37. Myers G. J., Sandler C., Badgett T. The Art of Software Testing. 3rd ed. Hoboken : John Wiley & Sons, 2012. 256 p.
38. McKinney W. Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter. 3rd ed. Sebastopol : O'Reilly Media, 2022. 579 p.
39. Beazley D., Jones B. K. Python Cookbook. 3rd ed. Sebastopol : O'Reilly Media, 2013. 706 p.

40. ISO/IEC/IEEE 29119-1:2022. Software and systems engineering — Software testing — Part 1: General concepts. Geneva : International Organization for Standardization, 2022. 66 p.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
Навчально-науковий інститут інформаційних технологій
Кафедра інформаційних систем та технологій

Інформаційна система бронювання готельних номерів

з використанням мови програмування Python

Виконав: студент групи ІСД-41 Станіслав Романчук
Керівник: к.т.н., доцент Оксана Ткаленко

Київ – 2026

Актуальність теми та мета роботи

АКТУАЛЬНІСТЬ

1. Активна цифровізація готельного бізнесу
2. Ручне ведення бронювань спричиняє помилки та дублювання
3. Потреба в автоматизації перевірки доступності номерів
4. Відсутність доступних навчальних прототипів локальних ІС бронювання

МЕТА РОБОТИ

Проєктування та програмна реалізація інформаційної системи бронювання готельних номерів на Python із забезпеченням обліку номерного фонду, роботи з клієнтами, перевірки доступності та базової аналітики.

ОБ'ЄКТ ДОСЛІДЖЕННЯ

Процес автоматизації бронювання готельних номерів у сфері готельного обслуговування.

ПРЕДМЕТ ДОСЛІДЖЕННЯ

Методи та програмні засоби розробки ІС бронювання з використанням Python, Tkinter і SQLite.

Завдання дипломної роботи

3

1. Розкрити поняття та класифікацію ІС у готельному бізнесі
2. Охарактеризувати особливості організації процесу бронювання
3. Проаналізувати сучасні програмні рішення (Oracle OPERA, Cloudbeds тощо)
4. Обґрунтувати вибір Python, Tkinter та SQLite
5. Сформулювати вимоги та спроєктувати функціональну структуру ІС
6. Розробити структуру бази даних (rooms, clients, bookings)
7. Описати алгоритм перевірки доступності номерів
8. Реалізувати GUI, карту номерів, модуль бронювань та аналітики
9. Провести функціональне тестування системи

Теоретичні засади: класифікація ІС у готельному бізнесі

4

Системи бронювання	PMS-системи	CRM-системи	Booking Engine
Прийом заявок, перевірка доступності, фіксація дат	Управління готелем, поселення, звітність (Oracle OPERA Cloud)	Робота з клієнтами, персоналізовані пропозиції, лояльність	Онлайн-бронювання через сайт готелю (Cloudbeds)

КЛАСИФІКАЦІЯ ЗА СПОСОБОМ РОЗГОРТАННЯ

Локальні	Клієнт-серверні	Веборієнтовані	Хмарні
Desktop-додаток, працює без Інтернету (наш проєкт)	Окремий сервер БД, кілька робочих місць	Запуск через браузер, доступ з будь-якого ПК	Розміщення у хмарі, постачальник SaaS

Аналіз сучасних програмних рішень

5

Oracle OPERA Cloud	Cloudbeds	Наша ІС (прототип)
PMS-система	*Hotel Mgmt Platform*	*Desktop-додаток (Python)*
Хмарна PMS-платформа	PMS + Booking Engine	Локальна БД SQLite
Централізація даних	Channel Manager	Графічний інтерфейс Tkinter
Мобільна робота персоналу	Аналітика та звітність	Карта номерів по поверхах
Інтеграція з POS, Revenue Mgmt	Прямі бронювання з сайту	Експорт CSV, аналітика

Наша система є спрощеним, але функціонально цілісним прототипом, що реалізує основні принципи зазначених рішень.

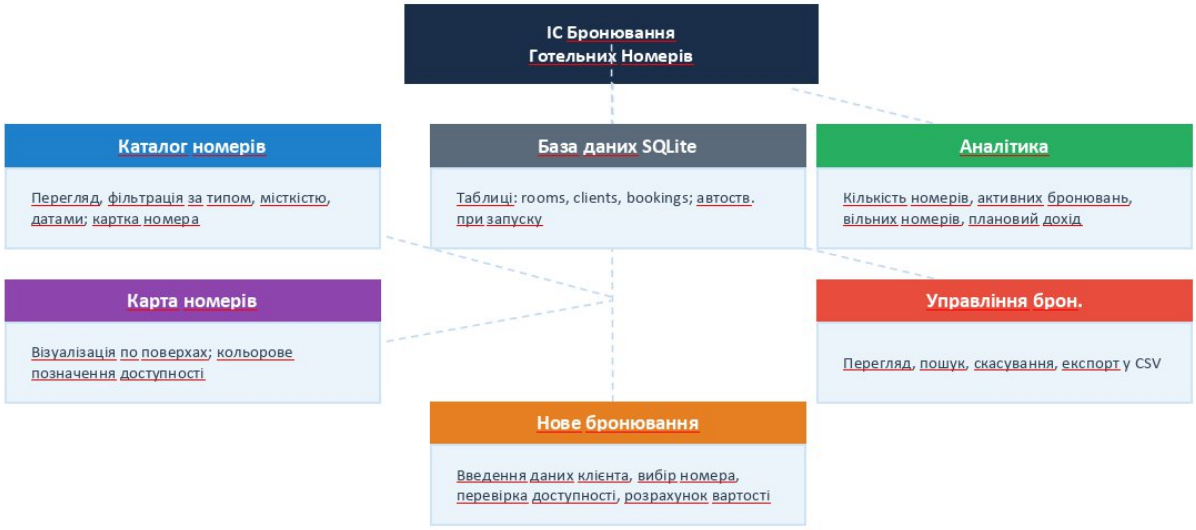
Обґрунтування вибору технологій розробки

6

Python 3.10+	Tkinter	SQLite 3
Мова програмування	*GUI-бібліотека*	*СУБД*
Висока читабельність коду	Вбудована в стандартний Python	Файлова БД без окремого сервера
Стандартна бібліотека без додаткових залежностей	Підтримка вікон, вкладок, форм, Canvas	Стандартний модуль sqlite3 в Python
Широка підтримка та навчальні ресурси	Кросплатформенність (Windows/Linux/macOS)	Підтримка зовнішніх ключів і транзакцій
Повний цикл desktop-розробки	Достатня для навчального прототипу	Ідеальна для desktop-додатків

7

Функціональна структура інформаційної системи



9

Алгоритм перевірки доступності номерів

ТЕСТОВІ ВИПАДКИ

Існуюче: 01.06 – 05.06

05.06–08.06	Дозволено
04.06–07.06	Заборонено
25.05–01.06	Дозволено
02.06–03.06	Заборонено
30.05–10.06	Заборонено

10

Програмна реалізація: модулі системи

Каталог номерів

– Таблиця з фільтрами (тип, місткість, дати)

– Картка вибраного номера

– Схематичний вигляд приміщення (Canvas)

Управл. бронюваннями

– Таблиця з пошуком за ПІБ / номером

– Скасування → зміна статусу на cancelled

– Експорт у CSV (сумісний з Excel)

Карта номерів

– Кнопки-номери по поверхах

– Зелений — вільний, Червоний — зайнятий

– Натискання → деталі номера

Аналітика

– Усього номерів / активних бронювань

– Вільних номерів на вибраний період

– Плановий дохід за активними брон.

Нове бронювання

– Форма: ПІБ, телефон, email, дати, примітка

– Перевірка доступності + розрахунок вартості

– Вартість = к-сть ночей × ціна за ніч

База даних

– Автоств. hotel_booking.db при першому запуску

– 3 таблиці: rooms, clients, bookings

– Демо-номери 4 категорій при ініціалізації

11

Тестування працездатності системи

№	Тестова дія	Результат	Статус
1	Запуск програми та автостворення БД		Успішно ✓
2	Відображення каталогу демонстраційних номерів		Успішно ✓
3	Фільтрація номерів за типом та місткістю		Успішно ✓
4	Перевірка доступності номера на вибрані дати		Успішно ✓
5	Створення бронювання та запис до SQLite		Успішно ✓
6	Блокування подвійного бронювання (конфлікт дат)		Успішно ✓
7	Скасування бронювання (зміна статусу на cancelled)		Успішно ✓
8	Пошук бронювань за ПІБ клієнта або номером кімнати		Успішно ✓
9	Експорт бронювань у CSV-файл		Успішно ✓
10	Оновлення аналітичних показників		Успішно ✓

12

ВИСНОВКИ

1. Досліджено теоретичні засади ІС бронювання: класифікацію, функції, сучасні рішення (Oracle OPERA Cloud, Cloudbeds)
2. Обґрунтовано вибір Python 3.10+, Tkinter та SQLite як оптимального стеку для навчального desktop-прототипу
3. Спроєктовано БД (3 таблиці) та алгоритм перевірки перетину дат, що унеможливлює подвійне бронювання
4. Реалізовано 5 модулів: каталог, карта номерів, нове бронювання, управління, аналітика
5. Проведено функціональне тестування — 10 сценаріїв підтвердили коректну роботу системи
6. Практичне значення: прототип може слугувати основою для розширення (веб, авторизація, оплати, CSV-звіти)

Дякую за увагу!

13

Апробація результатів дослідження

1

«Інформаційна система моніторингу “розумного будинку” ІС моніторингу»

Конференція: III Міжнародна науково-практична конференція «Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу»

2

«Особливості інформаційної системи бронювання готельних номерів з використанням мови програмування Python»

Конференція: VII Міжнародна науково-технічна конференція «Сучасний стан та перспективи розвитку IoT»