

Хмельницький національний університет
Факультет інформаційних технологій
Кафедра комп'ютерної інженерії та інформаційних систем

КВАЛІФІКАЦІЙНА РОБОТА

бакалавр

Освітній рівень

Веб-орієнтована інформаційна система управління готелем

Назва теми

КвРІСТ.200189.20.01.05 ПЗ

Шифр

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Виконав: студент III курсу, група ICTc-20-1

Підпис

В.Р. Андрєєв

Ініціали, прізвище

Керівник

Підпис, дата

Є.Г. Гнатчук

Ініціали, прізвище

Нормоконтролер

Підпис, дата

С.М. Лисенко

Ініціали, прізвище

До захисту допускаю:

Зав. кафедри комп'ютерної
інженерії та інформаційних
систем

Підпис

Т.О. Говоруценко

Ініціали, прізвище

«1» червня 2023 р.

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Освітній рівень БАКАЛАВР

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Зав. кафедри Т.О.Говорущенко

“ 10 ” 01 2023 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Андрєв Владислав Русланович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Веб-орієнтована інформаційна система управління готелем

Керівник проекту (роботи) Гнатчук Є.Г. к.т.н., доцент.

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

Затверджена наказом ректора університету від 01.03.2023 р. № 5

2. Строк подання студентом проекту (роботи) на кафедру 01.06.2023 р.

3. Вихідні дані до проекту (роботи) Завдання на дипломне проектування

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

Дослідження предметної області та постановка задачі

Проектування веб-орієнтованої інформаційної системи

Програмна реалізація та тестування веб-орієнтованої інформаційної системи

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





Архітектура веб-орієнтованої інформаційної системи готелю

UML-діаграми

Інтерфейсні вікна інформаційної системи

Лістинг програми запуску головного вікна розробленого програмного забезпечення

6. Консультанти розділів дипломного проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Лисенко С.М., професор кафедри КПС		
Антиплагіат	Нічепорук А.О., доцент кафедри КПС		

7. Дата видачі завдання « 01 » 03 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	23.01.2023	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	02.02.2023	виконано
3	Робота над розділом 1 – Дослідження предметної області та постановка задачі	10.02.2023	виконано
4	Робота над розділом 2 – Проектування веб-орієнтованої інформаційної системи	23.03.2023	виконано
5	Робота над розділом 3 – Програмно реалізація та тестування веб-орієнтованої інформаційної системи управління готелем	28.04.2023	виконано
6	Оформлення пояснювальної записки згідно вимог	20.05.2023	виконано
7	Попередній захист ВКР	26.05.2023	виконано
8	Захист ВКР на засіданні ЕК	Червень 2023 року	

Студент


Підпис


Підпис

Андрєєв В.Р.

Ініціали, прізвище

Керівник проекту (роботи)

Гнатчук Є.Г.

Ініціали, прізвище

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Веб-орієнтована інформаційна система управління готелем».

Автор роботи: Андреев Владислав Русланович.

Керівник роботи: Гнатчук Єлизавета Геннадіївна.

Пояснювальна записка: 64 с., 19 рис., 6 табл., 2 дод., 64 джерел.


Графічна частина: 3 креслення.

ІНФОРМАЦІЙНА СИСТЕМА, СИСТЕМА ГОТЕЛЮ, ВЕБ-СИСТЕМА.

Мета: Ефективне керування готельними процесами, поліпшення обслуговування клієнтів і підвищення загальної продуктивності готельного бізнесу шляхом розроблення веб-орієнтованої інформаційної системи для готелю.

Об'єкт дослідження: Готельна діяльність, процеси управління готелем, взаємодія з клієнтами та різними структурними підрозділами готелю.

Предмет дослідження: Веб-орієнтована інформаційна система для готелю, включаючи архітектуру, функціонал, базу даних, інтерфейси користувача та інші компоненти, які допомагають в ефективному керуванні готельними процесами.


Підпис студента

30.05.2023
Дата

ЗМІСТ

СКРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	4
ВСТУП.....	5
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ	7
1.1 Загальна інформація про веб-орієнтовану інформаційну систему управління готелем	7
1.2 Веб-орієнтована інформаційна система управління готелем.....	12
1.3 Вимоги до веб-орієнтованої інформаційної системи управління готелем	18
1.4 Висновки	21
2 ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	23
2.1 Архітектура ІТ-інфраструктури підприємств	23
2.2 Функційні вимоги.....	34
2.3 Нефункційні вимоги.....	37
2.4 Висновки	39
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	41
3.1 Обґрунтування вибору засобів розробки веб-орієнтованої інформаційної системи	41
3.2 Вимоги до апаратного і програмного забезпечення.....	44
3.3 UML діаграми	46
3.4 Структура бази даних	52
3.5 Програмна реалізація веб-орієнтованої інформаційної системи	55
3.6 Інтерфейс веб-орієнтованої інформаційної системи.....	56
3.7 Висновки	66
ВИСНОВКИ.....	67
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ	69

					КвРІСТ.200189.20.01.05 ПЗ			
Зм.	Арк.	№докум.	Підпис	Дата	Веб-орієнтована інформаційна система управління готелем Пояснювальна записка	Літера	Аркуш	Аркушів
Виконав		Андрєєв В.Р.		30.05				
Перевір.		Гнатчук Є.Г.		30.05			2	64
Н.контр.		Лисенко С.М		30.05		ХНУ, ІСТс-20-1		
Затвер.		Говорущенко Т.О		30.05				

ДОДАТОК А.....	75
Копія креслення «Архітектура веб-орієнтованої інформаційної системи готелю».....	75
ДОДАТОК Б	76
Копія креслення «UML-діаграми»	76
ДОДАТОК В	77
Копія креслення «Інтерфейсні вікна інформаційної системи»	77
ДОДАТОК Г	78
«Лістинг програми запуску головного вікна розробленого програмного забезпечення»	78

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД - база даних

ІС – інформаційна система

ІСГ – інформаційна система готелю

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		4

ВСТУП

Веб-орієнтована інформаційна система готелю - це комплекс програмних та апаратних засобів, які допомагають автоматизувати процеси управління та обслуговування готелю. Завдяки цій системі, готелі можуть забезпечити швидкий та якісний сервіс для своїх клієнтів, оптимізувати процеси бронювання, обліку фінансів та інші важливі операції.

Веб-орієнтована інформаційна система готелю може включати в себе такі компоненти, як системи бронювання, автоматизовані розрахунки та операції з готівкою, системи контролю доступу та безпеки, системи управління запасами та обладнанням готелю, системи збору та аналізу даних про клієнтів та багато інших.

Веб-орієнтована інформаційна система готелю є необхідним елементом сучасного готельного бізнесу, оскільки допомагає готелям підтримувати високу якість обслуговування та ефективно управляти бізнес-процесами.

Веб-орієнтована інформаційна система готелю може забезпечувати такі функції, як:

1) управління бронюваннями - система дозволяє клієнтам забронювати номер в готелі через Інтернет або по телефону. Інформація про бронювання автоматично заноситься в базу даних і доступна менеджерам готелю для подальшого управління;

2) операції з готівкою - система дозволяє автоматично розраховувати суму, яку клієнти повинні заплатити за проживання, харчування та інші послуги. Також, система може забезпечувати оплату карткою, безготівковий розрахунок та інші способи оплати;

3) системи контролю доступу та безпеки - система може включати в себе камери спостереження, електронні ключі та інші засоби контролю доступу, що допомагають забезпечити безпеку клієнтів та співробітників готелю;

4) управління запасами та обладнанням готелю - система дозволяє вести облік запасів, замовляти та отримувати необхідні матеріали та обладнання для готелю;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		5

5) системи збору та аналізу даних про клієнтів - система дозволяє збирати і аналізувати дані про клієнтів, такі як історія бронювань, звички споживання та інші дані, що допомагають підвищити якість обслуговування та задоволеність клієнтів.

Інформаційна система готелю може бути індивідуальною для кожного готелю, оскільки вона повинна відповідати потребам конкретного готелю та його клієнтів. Це може включати в себе різні функції та налаштування, які підходять для конкретного типу готелю.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		6

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Загальна інформація про веб-орієнтовану інформаційну систему управління готелем

Веб-орієнтовані інформаційні системи - це комплексні технологічні рішення, що забезпечують збір, обробку, зберігання, передачу та використання інформації з метою забезпечення ефективності та оптимізації різних процесів в організації або управлінні певними сферами діяльності.

Веб-орієнтовані інформаційні системи можуть використовуватись в різних галузях: бізнесі, науці, охороні здоров'я, громадському секторі тощо. Вони можуть бути розроблені для автоматизації різних процесів, від фінансових транзакцій до виробничої лінії, або для забезпечення обробки та аналізу великих обсягів даних.

Веб-орієнтовані інформаційні системи складаються з різних компонентів, включаючи апаратне забезпечення, програмне забезпечення та людські ресурси. Вони можуть бути локальними, розподіленими або хмарними, залежно від того, як забезпечується доступ до даних та програм.

Оскільки інформаційні системи грають все більш важливу роль у сучасному світі, розуміння їх функціонування та можливостей стає все більш важливим для бізнесу та відомства. Для того, щоб успішно використовувати інформаційні системи, потрібно мати знання та навички в різних областях, таких як програмування, бази даних, мережі та безпека інформації.

Однією з ключових переваг веб-орієнтованих інформаційних систем є можливість автоматизації рутинних операцій, що дозволяє відволіктись від них та зосередитись на більш складних завданнях, таких як аналіз даних та прийняття стратегічних рішень. Інформаційні системи також можуть забезпечувати швидкий та точний доступ до інформації, що зменшує ризик помилок та покращує продуктивність.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		7

Веб-орієнтовані інформаційні системи можуть бути розроблені для різних цілей та задач, наприклад, для підтримки управління бізнесом, для обробки та аналізу даних, для автоматизації виробництва та управління ланцюгом постачання, для підтримки медичної діагностики та лікування, для підтримки навчального процесу та інших.

Одним з ключових компонентів веб-орієнтованої інформаційної системи є база даних, що містить інформацію про різні об'єкти та їх взаємодію. База даних може бути структурованою, наприклад, у вигляді таблиць, або неструктурованою, наприклад, у вигляді текстових файлів. Для роботи з базою даних використовуються спеціальні мови запитів, наприклад SQL.

Веб-орієнтовані інформаційні системи також мають важливу роль у забезпеченні безпеки даних та інформації. За допомогою спеціальних засобів, таких як шифрування, аутентифікація та авторизація, інформаційні системи забезпечують захист від несанкціонованого доступу до даних та інформації.

Усі ці фактори зробили інформаційні системи ключовим елементом сучасного бізнесу та управління, що дозволяє забезпечувати ефективну та ефективну роботу організацій та сприяти їх розвитку.

Веб-орієнтовані інформаційні системи також грають важливу роль у забезпеченні комунікації та співпраці між різними групами людей, що працюють в організації. Зокрема, вони дозволяють швидко та легко обмінюватися даними, спілкуватися та координувати роботу між різними відділами та підрозділами.

Одним з найважливіших аспектів інформаційних систем є їх здатність до адаптації до змін. Швидкий темп змін в бізнесі та технологіях вимагає від інформаційних систем здатності до швидкого реагування та адаптації до нових умов та вимог. Це може включати в себе розробку нових функціональних можливостей, модифікацію існуючих процесів та даних, та інші аспекти.

На жаль, інформаційні системи також можуть бути піддаються ризикам та загрозам, таким як хакерські атаки, віруси та інші види злочинної діяльності. Тому безпека та захист інформації є надзвичайно важливим аспектом розробки та експлуатації інформаційних систем.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		8

Узагальнюючи, інформаційні системи стали необхідністю для бізнесу та управління в сучасному світі, що дозволяє автоматизувати рутинні операції, забезпечувати ефективність та ефективність, покращувати комунікацію та співпрацю, а також забезпечувати безпеку та захист інформації.

На сьогоднішній день існує багато різних інформаційних систем, які використовуються в різних галузях і для різних цілей. Однак, деякі з них вважаються найбільш популярними і широко використовуються в бізнесі та управлінні. Давайте розглянемо декілька найпопулярніших інформаційних систем і порівняємо їх за деякими критеріями:

1) ERP системи (Enterprise Resource Planning) - це комплексні інформаційні системи, які інтегрують в собі всі аспекти бізнесу, такі як фінанси, виробництво, логістика, продажі та інші. Ці системи забезпечують стандартизацію процесів і даних, спрощують комунікацію та забезпечують цілісність інформації. У цієї системи є свої переваги та недоліки зображенні на таблиці 1.1. Деякі з найбільш популярних ERP систем на сьогоднішній день - SAP, Oracle, Microsoft Dynamics, Infor.

2) CRM системи (Customer Relationship Management) - це інформаційні системи, які дозволяють управляти взаєминами з клієнтами, включаючи управління продажами, маркетингом та обслуговуванням клієнтів. Вони дозволяють збирати та аналізувати дані про клієнтів, покращувати комунікацію з ними та забезпечувати краще обслуговування. У цієї системи є свої переваги та недоліки зображенні на таблиці 1.2. Найбільш популярні CRM системи на сьогоднішній день - Salesforce, Microsoft Dynamics CRM, Zoho CRM, Hubspot.

3) SCM системи (Supply Chain Management) - це інформаційні системи, які дозволяють управляти логістикою та постачанням продукції від постачальника до клієнта. Вони забезпечують ефективне управління інвентарем, логістикою та виробництвом та дозволяють знизити витрати та покращити ефективність постачального ланцюга в цілому. У цієї системи є свої переваги та недоліки зображенні на таблиці 1.3 Деякі з найбільш популярних SCM систем на сьогоднішній день - SAP Supply Chain Management, Oracle Supply Chain Management, JDA Software.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		9

4) ВІ системи (Business Intelligence) - це інформаційні системи, які дозволяють збирати та аналізувати дані про діяльність бізнесу з метою прийняття кращих управлінських рішень. Вони забезпечують візуалізацію даних та роблять їх доступними для аналізу та інтерпретації, дозволяють проводити прогнозування та виявляти тенденції. У цієї системи є свої переваги та недоліки зображенні на таблиці 1.4. Деякі з найбільш популярних ВІ систем на сьогоднішній день - Tableau, QlikView, Microsoft Power BI, IBM Cognos.

Кожна з цих інформаційних систем має свої переваги та недоліки, і вибір конкретної залежить від бізнес-потреб, функціональних можливостей та бюджету. Проте, всі вони допомагають покращити ефективність управління бізнесом та дозволяють бути більш конкурентоспроможними на ринку.

Таблиця 1.1 – ERP (Enterprise Resource Planning) системи

Переваги	Недоліки
Централізована система управління, що дозволяє забезпечити максимальну ефективність роботи організації	Висока вартість впровадження та підтримки
Інтеграція різних функціональних підрозділів організації та автоматизація бізнес-процесів	Потребує великих затрат на навчання персоналу
Забезпечення стандартизації діяльності організації, що дозволяє уникнути помилок та покращити якість продукції/послуг	Ризик залежності від виробника та недостатня гнучкість в адаптації до змін у потребах організації
Інформаційний потік в режимі реального часу, що дозволяє приймати швидкі та обґрунтовані рішення	

Таблиця 1.2 – CRM (Customer Relationship Management) системи

Переваги	Недоліки
Забезпечують можливість ведення комплексної інформації про клієнтів	Потребують значних затрат на впровадження та підтримку
Дозволяють забезпечити повну контроль над процесами продажу та підтримки клієнтів	Ризик недостатньої використаності системи через недостатню мотивацію персоналу
Забезпечують можливість аналізу ефективності маркетингових та рекламних кампаній	Ризик залежності від одного постачальника технологій
Дозволяють збільшити продуктивність та ефективність роботи команди продажів	Ризик, що дані про клієнтів можуть бути недостатньо захищені

Таблиця 1.3 – SCM (Supply Chain Management) системи

Переваги	Недоліки
Дозволяють ефективно управляти ланцюгом постачання товарів та послуг;	Висока вартість впровадження та підтримки;
Забезпечують моніторинг стану запасів, оптимізацію доставок та скорочення витрат;	Складна процедура інтеграції з іншими системами, що може призвести до затримок у виконанні завдань;
Покращують взаємодію між різними партнерами у ланцюгу постачання;	Ризик виникнення проблем зі зберіганням та обробкою даних.
Забезпечують можливість прогнозування попиту	

Таблиця 1.4 – BI (Business Intelligence) системи

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		11

Переваги	Недоліки
Забезпечують широкі можливості аналізу даних та вивчення трендів в режимі реального часу	Висока вартість впровадження та підтримки
Дозволяють забезпечити цілеспрямоване прийняття рішень на основі даних	Складність процесу налаштування та інтеграції з іншими системами
Забезпечують можливість виявлення недоліків та резервів для покращення бізнес-процесів	Ризик необхідності значних затрат на навчання персоналу та впровадження нових практик у діяльність організації
Дозволяють збільшити ефективність та продуктивність роботи команди аналітиків	

1.2 Веб-орієнтована інформаційна система управління готелем

Веб-орієнтована інформаційна система готелю - це програмний комплекс, який забезпечує автоматизацію бізнес-процесів готелю, включаючи бронювання, реєстрацію гостей, облік розрахунків, облік запасів та управління персоналом.

Основні компоненти веб-орієнтованої інформаційної системи готелю можуть включати:

- 1) модуль бронювання, який забезпечує можливість здійснювати бронювання номерів з вибором дати заїзду та виїзду, кількості гостей тощо;
- 2) модуль реєстрації, який дозволяє швидко та зручно реєструвати гостей готелю, включаючи формування картки гостя та видачу електронного ключа для доступу до номеру;
- 3) модуль обліку розрахунків, який забезпечує облік готівки та безготівкових розрахунків гостей, формування звітів про виконані операції, контроль за взаєморозрахунками з контрагентами тощо;

4) модуль управління запасами, який дозволяє контролювати кількість та рух товарів, складати замовлення на постачання товарів, проводити інвентаризацію та аналізувати звіти про запаси;

5) модуль управління персоналом, який забезпечує контроль за графіком роботи, зарплатою, навчанням та іншими аспектами роботи персоналу готелю.

Переваги використання інформаційної системи готелю:

1) зниження часу та витрат на обробку даних та здійснення операцій;

2) покращення якості обслуговування гостей та збільшення рівня задоволеності клієнтів;

3) забезпечення точного обліку грошових потоків та контролю за запасами;

4) забезпечення оперативного прийняття рішень та аналізу даних для оптимізації бізнес-процесів;

5) покращення комунікації між різними департаментами готелю та зменшення можливості помилок в обробці даних.

Недоліки використання інформаційної системи готелю:

1) високі витрати на впровадження та налаштування системи;

2) необхідність постійної підтримки та оновлення системи;

3) ризик можливості виникнення технічних проблем та несправностей;

4) потенційна загроза конфіденційності даних та можливість кібератак.

Незважаючи на недоліки, використання інформаційної системи готелю є дуже важливим для забезпечення ефективного та якісного управління готелем та задоволення потреб клієнтів.

Інформаційна система готелю може включати в себе такі модулі, як:

1) модуль бронювання - дозволяє клієнтам бронювати номери готелю через Інтернет або через контакт-центр. Модуль повинен бути інтегрований з системою керування номерами та їхнім прибиранням, щоб запобігти подвійному бронюванню або неправильному прибиранню номерів;

2) модуль ресторану - дозволяє клієнтам замовляти їжу та напої через Інтернет або через квитанцію, яку вони отримали під час реєстрації. Модуль

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						13
Зм..	Арк.	№докум.	Підпис	Дата		

повинен бути інтегрований з системою кухні, щоб забезпечити швидке та точне приготування страв;

3) модуль управління номерами - дозволяє готелю контролювати стан кожного номера, щоб відстежувати, коли номер звільняється, та призначати його наступному гостю. Модуль повинен бути інтегрований з модулем бронювання, щоб забезпечити правильну розподільну політику та уникнення конфліктів;

4) модуль управління персоналом - дозволяє готелю керувати графіком роботи, оплатою праці та іншими аспектами, пов'язаними з персоналом готелю. Модуль також може включати тренінгову платформу для підвищення кваліфікації персоналу.

Переваги використання інформаційної системи готелю:

- 1) забезпечення більш ефективного та швидкого обслуговування гостей;
- 2) оптимізація роботи готелю та зменшення ризику помилок;
- 3) збільшення продуктивності праці персоналу та скорочення часу на адміністративні процеси;
- 4) покращення контролю над бізнесом
- 5) забезпечення більш точної та швидкої аналітики щодо роботи готелю та його прибутковості;
- 6) покращення взаємодії з гостями та забезпечення високого рівня задоволення клієнтів.

Недоліки використання інформаційної системи готелю:

- 1) високі витрати на впровадження та підтримку системи;
- 2) потреба у навчанні персоналу та зміна робочих процесів;
- 3) ризик технічних проблем, таких як збої та відмови обладнання, які можуть призвести до перебоїв у роботі готелю;
- 4) ризик порушення безпеки даних та конфіденційності.

Незважаючи на ці недоліки, використання інформаційної системи готелю має значний потенціал для забезпечення ефективної та продуктивної роботи готелю, а також для поліпшення взаємодії з гостями та підвищення їхнього задоволення. Важливо зробити відповідні дослідження та ретельно обрати систему, щоб забезпечити її оптимальну працездатність та ефективність.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		14

Інформаційні системи готелів можуть включати різноманітні програмні продукти, які забезпечують автоматизацію та оптимізацію різних аспектів роботи готелю. До основних систем, які використовуються у готельній галузі, належать:

1) Property Management System (PMS) - це програмне забезпечення, яке використовується для управління готельними операціями, такими як бронювання, реєстрація гостей, управління номерами, збір платежів, облік інвентарю тощо;

2) Central Reservation System (CRS) - це програмне забезпечення, яке дозволяє готелю розміщувати свої номери на різних онлайн-бронювальних сайтах та дистриб'юторських системах, таких як Booking.com, Expedia тощо;

3) Revenue Management System (RMS) - це програмне забезпечення, яке використовується для оптимізації ціноутворення на номери готелю залежно від попиту та сезонності;

4) Customer Relationship Management (CRM) - це програмне забезпечення, яке використовується для збору та аналізу інформації про клієнтів готелю, що дозволяє покращити взаємодію з ними та збільшити рівень їхнього задоволення;

5) Point of Sale (POS) - це програмне забезпечення, яке використовується для управління ресторанами, баром та іншими готельними послугами, що дозволяє вести облік продажів та інших операцій;

6) Business Intelligence (BI) - це програмне забезпечення, яке дозволяє збирати, аналізувати та відображати дані про діяльність готелю, що дозволяє забезпечити ефективніше управління та приймати стратегічні рішення.

Всі ці системи допомагають готелям покращити ефективність своєї роботи та забезпечити високу якість обслуговування для своїх клієнтів.

Property Management System (PMS):

Переваги:

1) удосконалює процеси управління готелем, що дозволяє збільшувати ефективність роботи;

2) спрощує процес резервування номерів та контролює їх наявність, що дозволяє уникнути зайвих непорозумінь з клієнтами;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		15

3) забезпечує повний контроль над фінансовими операціями готелю та обліком витрат.

Недоліки:

- 1) висока вартість впровадження та підтримки;
- 2) система може не відповідати потребам окремих готелів та їх специфіці.

Central Reservation System (CRS):

Переваги:

- 1) забезпечує максимально швидкий доступ до інформації про доступність номерів, тарифи та іншу інформацію;
- 2) скорочує час для обробки бронювань та підтвердження їх з клієнтами;
- 3) дозволяє готелям отримувати замовлення з будь-якої точки світу, що збільшує їхню клієнтську базу.

Недоліки:

- 1) висока вартість впровадження та підтримки;
- 2) можливі проблеми з безпекою даних клієнтів.

Revenue Management System (RMS): Переваги:

- 1) допомагає готелям максимально використовувати їхні ресурси та максимізувати дохід;
- 2) дозволяє готелям більш точно прогнозувати свої доходи та розподіляти їх ресурси;
- 3) стимулює готелі до прийняття більш інформованих та обґрунтованих рішень.

Недоліки:

- 1) висока вартість впровадження та підтримки;
- 2) можливі проблеми з точністю прогнозування при недостатній якості вхідних даних.

Customer Relationship Management (CRM):

Переваги:

- 1) збільшення лояльності клієнтів завдяки персоналізованому обслуговуванню та відповідному підходу до кожного клієнта;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		16

2) збільшення обсягів продажу через краще розуміння потреб клієнтів та аналіз їхньої поведінки;

3) зменшення затрат на маркетинг завдяки можливості зосереджуватися на цільових групах та індивідуальних потребах клієнтів.

Недоліки:

1) високі витрати на імплементацію та підтримку системи;

2) необхідність інтеграції з іншими системами, що може бути часомістким та дорогим;

3) необхідність тренування персоналу для ефективного використання системи.

Point of Sale (POS):

Переваги:

1) підвищення ефективності роботи, зменшення часу на обробку замовлень та збільшення точності даних;

2) відслідковування продажів та запасів, що дозволяє уникнути зайвих затрат та управляти інвентарем;

3) можливість простого створення звітів та аналізу продажів.

Недоліки:

1) високі витрати на імплементацію та підтримку системи;

2) необхідність тренування персоналу та підтримки системи;

3) необхідність інтеграції з іншими системами, такими як система управління готелем.

Business Intelligence (BI)

Переваги:

1) дозволяє аналізувати інформацію про клієнтів, прибутки та витрати, тенденції ринку, що дозволяє покращити стратегії розвитку готельного бізнесу;

2) допомагає виявляти потенційні можливості для підвищення прибутковості та оптимізації витрат;

3) дозволяє відстежувати ефективність маркетингових кампаній та прогнозувати їхні результати;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						17
Зм.	Арк.	№докум.	Підпис	Дата		

4) надає можливість швидко реагувати на зміни в ринкових умовах та вирішувати проблеми.

Недоліки:

- 1) висока вартість розробки та впровадження системи ВІ;
- 2) необхідність в наявності високої якості та точності даних для ефективної роботи системи;
- 3) потребує підготовки та навчання персоналу для використання системи;
- 4) недостатня регулярність та своєчасність оновлення даних може призвести до недостовірних результатів аналізу.

1.3 Вимоги до веб-орієнтованої інформаційної системи управління готелем

Розробка власної веб-орієнтованої інформаційної системи управління готелем може бути складним завданням, яке вимагає уважної оцінки вимог та потреб бізнесу. Ось деякі загальні вимоги, які можуть бути корисними для розробки ефективної інформаційної системи готелю:

- 1) функціональність: Інформаційна система готелю має забезпечувати всі потреби, пов'язані з керуванням готелем. До основних функцій системи можуть належати реєстрація гостей, розподіл кімнат, управління запасами, бронювання кімнат і послуг, електронні ключі для входу в номери і так далі;
- 2) безпека: Інформаційна система готелю має забезпечувати захист конфіденційної інформації гостей, такої як особисті дані, фінансові транзакції та інші важливі дані;
- 3) відповідність: Інформаційна система готелю повинна відповідати всім вимогам і стандартам, що застосовуються у готельній галузі. Також важливо враховувати законодавчі вимоги, що регулюють зберігання і обробку конфіденційної інформації;
- 4) інтеграція: Інформаційна система готелю повинна бути здатна інтегруватися з іншими системами, які використовуються у готелі. Наприклад,

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		18

інтеграція з POS-системою може дозволити автоматично збирати і обробляти дані про продажі продуктів харчування;

5) простота використання: Інформаційна система готелю повинна бути простою у використанні та надавати користувачам зручний інтерфейс для роботи з системою;

6) масштабованість: Інформаційна система готелю повинна бути здатна масштабуватися в залежності від потреб готелю;

7) аналітика та звіти: система повинна мати можливість збирати та аналізувати дані, щоб надавати звіти та статистику про роботу готелю;

8) підтримка користувачів: виробник системи повинен надавати достатню підтримку користувачів, щоб допомогти в разі виникнення проблем;

9) вартість: вартість системи повинна бути в межах бюджету готелю та не повинна перевищувати її можливостей.

Основні функції, які повинні бути вбудовані в інформаційну систему готелю, включають наступне:

1) керування бронюваннями та прибуттям гостей: ця функція дозволяє гостям забронювати номери та вести облік їх прибуття. Вона повинна також включати можливість попереднього замовлення послуг та страв;

2) управління фінансами: ця функція дозволяє готелю контролювати свої фінанси, включаючи облік витрат та доходів, виставлення рахунків та оплату;

3) управління запасами: ця функція дозволяє готелю контролювати свої запаси, включаючи поповнення запасів та відстеження використання матеріалів;

4) управління персоналом: ця функція дозволяє готелю керувати своїм персоналом, включаючи розклади роботи, зарплатні виплати та навчання;

5) управління відносинами з клієнтами: ця функція дозволяє готелю зберігати інформацію про своїх гостей, включаючи їх персональні дані та історію відвідувань;

6) управління послугами: ця функція дозволяє готелю керувати послугами, які надаються гостям, такі як ресторани, басейни, фітнес-центри та інші;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		19

7) аналітика та звітність: ця функція дозволяє готелю збирати та аналізувати дані про своїх гостей та бізнес-процеси з метою прийняття кращих рішень. Інформація, що збирається, може включати такі дані, як статистика бронювання номерів, витрати гостей в ресторанах та барах, популярність різних послуг готелю та багато іншого.

Безпека є однією з ключових вимог до інформаційної системи готелю. Система повинна забезпечувати захист конфіденційної інформації про клієнтів, включаючи особисті дані, платежі, замовлення та інші дані. Для цього система повинна мати вбудовані механізми автентифікації, авторизації та шифрування даних.

Іншою важливою вимогою є відповідність системи готелю законодавчим та регуляторним вимогам, включаючи GDPR, PCI DSS та інші вимоги до обробки особистих даних та безпеки платежів.

Інтеграція з іншими системами є також важливою вимогою до інформаційної системи готелю. Система повинна бути легко інтегрована з іншими системами, такими як система бронювання, система управління витратами, система відеоспостереження та інші. Інтеграція допомагає забезпечити безперебійну роботу системи, запобігає втраті даних та сприяє збільшенню ефективності бізнесу.

Отже, система готелю повинна мати високу функціональність, забезпечувати високий рівень безпеки та відповідність вимогам законодавства, а також бути легко інтегрованою з іншими системами. Ці вимоги допоможуть забезпечити ефективну роботу готелю та задоволення потреб клієнтів.

Розробка власної інформаційної системи готелю має багато переваг для бізнесу. Ось деякі з них:

1) відповідність потребам бізнесу: розробка власної інформаційної системи дозволяє точно відповідати потребам конкретного готелю, уникнути надлишковості та зайвих витрат;

2) ефективність: власна інформаційна система може бути налаштована таким чином, щоб забезпечувати максимальну ефективність в роботі готелю та зниження витрат на операції;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		20

3) конкурентна перевага: розробка власної інформаційної системи може надати готелю конкурентну перевагу на ринку, покращити його репутацію та забезпечити високий рівень обслуговування для клієнтів;

4) більш точний аналіз: власна інформаційна система дозволяє готелю збирати та аналізувати дані про клієнтів, що дозволяє розуміти їхні потреби та бажання.

Отже, розробка власної інформаційної системи готелю є дуже важливою для забезпечення ефективності та конкурентоспроможності готельного бізнесу, забезпечення якісного обслуговування гостей, оптимізації бізнес-процесів, покращення взаємодії з клієнтами та збільшення прибутку. Розробка власної інформаційної системи дозволяє готелям виходити на новий рівень управління та підвищувати ефективність своєї роботи.

1.4 Висновки

У сучасному світі веб-орієнтовані інформаційні системи займають ключове місце в різних галузях, включаючи готельний бізнес. Веб-орієнтовані інформаційні системи допомагають автоматизувати бізнес-процеси, збільшують ефективність управління готелем, підвищують якість обслуговування гостей та допомагають збільшити прибуток.

Найбільш популярними інформаційними системами для готелів є Property Management System (PMS), Central Reservation System (CRS), Revenue Management System (RMS), Customer Relationship Management (CRM), Point of Sale (POS) та Business Intelligence (BI). Кожна з цих систем має свої переваги та недоліки, які слід враховувати при виборі.

Окрім того, розробка власної веб-орієнтованої інформаційної системи готелю може бути дуже важливою, оскільки це дозволяє підлаштувати систему під конкретні потреби та вимоги готелю. При розробці власної інформаційної системи слід враховувати багато факторів, таких як функціональність, безпеку, відповідність та інтеграцію з іншими системами.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		21

У загальному, веб-орієнтовані інформаційні системи допомагають готелям підвищити ефективність управління та якість обслуговування гостей, що дозволяє збільшити прибуток та конкурентність на ринку готельного бізнесу. Тому, використання інформаційних систем є необхідністю для сучасних готелів, а розробка власної інформаційної системи є важливим кроком у покращенні ефективності та конкурентоспроможності.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		22

2 ПРОЄКТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Архітектура ІТ-інфраструктури підприємств

Архітектура ІТ-інфраструктури підприємства - це план, який описує склад та взаємозв'язки компонентів ІТ-системи, які забезпечують роботу компанії. Ця архітектура дозволяє розробляти та управляти ІТ-інфраструктурою, забезпечувати її ефективне функціонування та забезпечувати відповідність бізнес-потребам підприємства.

Архітектура ІТ-інфраструктури має на меті забезпечити інформаційну та технічну підтримку бізнес-процесів. Вона включає в себе компоненти, такі як апаратне забезпечення, програмне забезпечення, мережі та комунікаційні засоби, бази даних, інтерфейси користувача та інші складові.

При розробці архітектури ІТ-інфраструктури підприємства необхідно враховувати бізнес-потреби компанії, технічні можливості та ресурси, а також вимоги до безпеки та надійності системи. Крім того, архітектура повинна бути гнучкою та легко змінюватися для адаптації до змін у бізнес-потребах та технологічних можливостях.

Для ефективного управління ІТ-інфраструктурою підприємства використовуються різні підходи, такі як ІТІЛ (Information Technology Infrastructure Library) та СОВІТ (Control Objectives for Information and Related Technology). Ці підходи дозволяють організувати управління ІТ-інфраструктурою, забезпечити її ефективну роботу та надійність, а також вирішувати проблеми, які виникають під час експлуатації системи.

В загальному, архітектура ІТ-інфраструктури підприємства має вирішувати завдання забезпечення безперебійної та ефективної роботи всіх компонентів ІТ-системи, що дозволяє досягати поставлених бізнес-цілей, забезпечувати високу продуктивність роботи, оптимізувати витрати на обслуговування та розвиток ІТ-інфраструктури, а також зменшувати ризики збоїв і вразливостей в системі. Ефективно побудована ІТ-інфраструктура дозволяє забезпечити стійкість та

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		23

надійність в роботі системи, зменшити час простою, збільшити продуктивність роботи працівників та підвищити загальну ефективність бізнесу підприємства.

Однак, при розробці архітектури ІТ-інфраструктури необхідно враховувати також потреби конкретного підприємства. Наприклад, якщо підприємство веде бізнес в галузі електронної комерції, то важливо мати масштабовану та безпечну інформаційну систему, що дозволить забезпечити високу доступність сайту та безпеку транзакцій.

Крім того, при розробці архітектури ІТ-інфраструктури важливо враховувати технічні та фінансові обмеження, які можуть вплинути на вибір конкретних рішень та технологій. Наприклад, якщо підприємство має обмежені бюджетні кошти, то важливо забезпечити оптимальне використання наявних ресурсів та технологій.

Таким чином, архітектура ІТ-інфраструктури підприємства є важливим елементом успішної діяльності та конкурентоспроможності на ринку. Правильний підхід до її розробки дозволяє забезпечити ефективність та оптимізацію роботи підприємства, зниження витрат та ризиків, а також підвищення якості обслуговування клієнтів.

Наслідком такого підходу є забезпечення швидкої реакції на зміни в бізнес-процесах, підвищення ефективності роботи співробітників та підвищення загальної продуктивності підприємства.

У загальному, успішна архітектура ІТ-інфраструктури підприємства включає в себе цілісну систему, яка забезпечує швидкий доступ до інформації, підвищує продуктивність, знижує витрати та підвищує конкурентоспроможність підприємства. При розробці архітектури ІТ-інфраструктури необхідно враховувати бізнес-процеси та потреби підприємства, щоб забезпечити її ефективність та оптимальність використання ресурсів. Крім того, необхідно підтримувати архітектуру в актуальному стані, відповідному сучасним вимогам та технологіям.

До складу архітектури ІТ-інфраструктури можуть входити такі елементи, як сервери, комутатори, маршрутизатори, засоби зберігання даних, програмне забезпечення, бази даних, системи забезпечення безпеки, системи моніторингу та

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		24

управління, технічна підтримка та інше. Кожен елемент повинен бути обраний з урахуванням потреб підприємства та забезпечувати високу ефективність та безпеку роботи всієї системи.

Основна архітектура ІТ-інфраструктури для Веб-орієнтованої інформаційної системи управління готелем може бути зображена наступною діаграмою (рис 2.1).

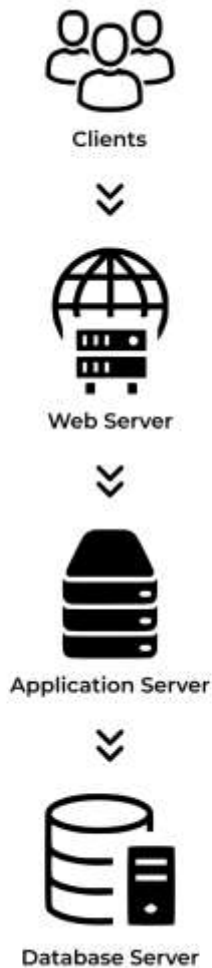


Рисунок 2.1 – Діаграма взаємодіє клієнтів з Веб-сервером

Діаграма показує, що клієнти взаємодіють з Веб-сервером, що забезпечує зовнішній доступ до веб-додатку. Веб-сервер звертається до серверу додатків для обробки бізнес-логіки, а сервер додатків, в свою чергу, отримує інформацію з бази даних. База даних містить усю необхідну інформацію про готель, номери, бронювання, клієнтів і т.д.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		25

Ця архітектура забезпечує гнучкість та масштабованість системи, а також можливість забезпечення безпеки та доступності інформації для клієнтів та працівників готелю.

У зв'язку зі швидким розвитком технологій та поширенням хмарних сервісів, архітектура ІТ-інфраструктури підприємства може включати в себе і хмарні сервіси, що дозволяє збільшити ефективність та масштабованість роботи компанії. Використання хмарних сервісів дозволяє знизити витрати на зберігання та обробку даних, а також забезпечує більш гнучкий та швидкий доступ до інформації з будь-якого місця з доступом до Інтернету. Однак, при використанні хмарних сервісів, необхідно враховувати питання безпеки та конфіденційності даних, оскільки вони зберігаються на серверах провайдера хмарних послуг. Враховуючи ці питання та вибравши оптимальну архітектуру ІТ-інфраструктури, підприємство зможе забезпечити ефективну роботу та масштабованість своєї діяльності, що в свою чергу сприятиме збільшенню прибутків та конкурентоспроможності на ринку.

ІТ-інфраструктура підприємства може бути розглянута як комплекс систем, пристроїв та програмного забезпечення, які забезпечують збереження та обробку даних, зв'язок між різними відділами та підрозділами підприємства, а також взаємодію з зовнішнім середовищем. Архітектура ІТ-інфраструктури підприємства повинна відповідати потребам та цілям бізнесу та бути розроблена з урахуванням таких факторів, як обсяг даних, розмір підприємства, доступність ресурсів, безпека, резервування, масштабованість та інші. Для ефективної роботи ІТ-інфраструктури підприємства необхідно мати належну організацію, контроль та підтримку систем. Організаційна структура ІТ-відділу підприємства може включати різні команди, такі як команда адміністраторів систем, команда розробників, команда технічної підтримки та інші. Основні складові архітектури ІТ-інфраструктури підприємства включають: Апаратне забезпечення: сервери, комп'ютери, пристрої зберігання даних, мережеве обладнання тощо. Програмне забезпечення: операційні системи, бази даних, програми обробки даних, програми зв'язку, програми захисту тощо. Мережева інфраструктура: мережеві компоненти, такі як маршрутизатори, комутатори, мережеві кабелі та інші, що забезпечують

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		26

зв'язок між пристроями. Захист інформації: системи захисту даних, резервні копії даних, антивірусне програмне забезпечення та інші заходи, що забезпечують безпеку та захист від злочинних дій та несанкціонованого доступу до даних.

Більш детальний огляд архітектури ІТ-інфраструктури підприємства для Веб-орієнтованої інформаційної системи управління готелем (рис 2.2).

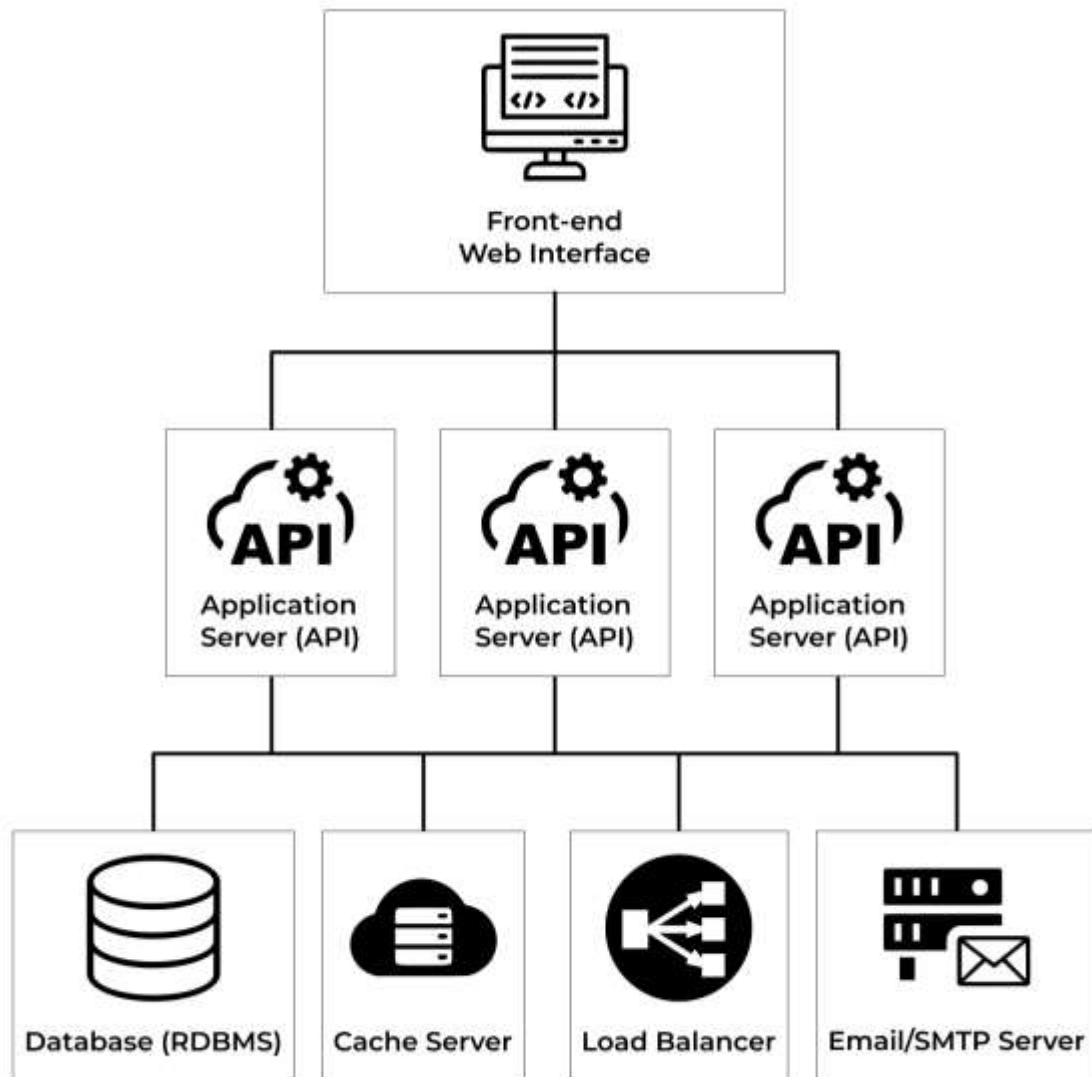


Рисунок 2.2 – Діаграма основних компонентів ІТ-інфраструктури

Основні компоненти ІТ-інфраструктури підприємства для Веб-орієнтованої інформаційної системи управління готелем включають:

- 1) Front-end - інтерфейс користувача, що забезпечує доступ до функцій системи через веб-браузер.

Зм.	Арк.	№докум.	Підпис	Дата

- 2) Application Server - сервер, що відповідає за обробку бізнес-логіки, отримання запитів з фронтенду, зв'язок з базою даних та кеш-сервером.
- 3) Database - реляційна база даних для зберігання даних про готель, клієнтів, номерів, бронювань тощо.
- 4) Cache Server - сервер кешування, що забезпечує швидкий доступ до даних та зменшує навантаження на базу даних.
- 5) Load Balancer - сервер, що розподіляє навантаження між декількома екземплярами сервера додатків для забезпечення високої доступності та швидкодії.
- 6) Email/SMTP Server - сервер електронної пошти, що відповідає за відправку електронних листів.

Архітектура IT-інфраструктури підприємства є ключовим елементом для ефективного функціонування Веб-орієнтованої інформаційної системи управління готелем. IT-інфраструктура має забезпечити необхідні вимоги до продукту, такі як швидкодія, надійність, масштабованість, ефективність та продуктивність. Для досягнення цих вимог необхідно враховувати різноманітні аспекти, такі як безпека, надійність, сумісність, наявність підтримки, легкість використання, а також сумісність з законодавством.

Веб-орієнтована інформаційна система управління готелем має складну структуру, яка включає в себе різноманітні складові частини, такі як веб-сервер, базу даних, сервери зберігання даних, мережеві комутатори, маршрутизатори та інші компоненти. Для забезпечення безперебійної роботи системи необхідно мати високоякісну IT-інфраструктуру з надійними засобами зберігання даних, а також швидкою мережею з високою пропускнуою здатністю та масштабованістю.

Для забезпечення безпеки даних важливо мати відповідну політику захисту інформації, яка включає в себе різні заходи, такі як захист мережі, захист даних та захист додатків. Також необхідно забезпечити належну підтримку та післяпродажне обслуговування системи, щоб забезпечити безперебійну роботу і уникнути простоїв в роботі системи.

Важливим етапом при створенні архітектури IT-інфраструктури є вибір технологій, які використовуються в системі. Для веб-орієнтованої інформаційної

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						28
Зм.	Арк.	№докум.	Підпис	Дата		

системи управління готелем можуть бути використані такі технології, як HTML, CSS та JavaScript для розробки клієнтської частини системи, а також PHP, Python або Java для розробки серверної частини.

Для забезпечення безпеки інформації в системі можуть бути використані різні методи, такі як шифрування даних, аутентифікація та авторизація користувачів, контроль доступу до ресурсів інформаційної системи.

Крім того, ефективність і продуктивність системи можуть бути покращені за допомогою використання таких технологій, як кешування даних, оптимізація запитів до бази даних, використання CDN для швидкої доставки контенту до користувачів.

Важливо також враховувати масштабованість системи, тобто можливість збільшення обсягів обробки даних і кількості користувачів без значного погіршення продуктивності і доступності системи.

Таким чином, архітектура IT-інфраструктури підприємства для веб-орієнтованої інформаційної системи управління готелем повинна бути добре пророблена і забезпечувати надійність, безпеку, продуктивність, масштабованість та ефективність системи.

Python - це потужна, високорівнева мова програмування з великою кількістю бібліотек та фреймворків, що дозволяють швидко та ефективно розробляти веб-орієнтовані інформаційні системи. Деякі з переваг використання Python для створення архітектури IT-інфраструктури підприємства для системи управління готелем включають:

1) простота вивчення: Python має простий та лаконічний синтаксис, що робить його дуже легким для вивчення та розуміння. Це зменшує час, необхідний для розробки системи, і сприяє зниженню витрат на її створення;

2) багата бібліотека та фреймворки: Python має велику кількість бібліотек та фреймворків, що дозволяють швидко розробляти веб-додатки та системи. Наприклад, для створення веб-додатків можна використовувати фреймворки, такі як Django або Flask;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		29

3) підтримка багатьох операційних систем: Python підтримує багато операційних систем, включаючи Windows, Linux та MacOS. Це дозволяє запускати розроблену систему на будь-якому сервері з підтримкою Python;

4) велика спільнота розробників: Python має велику та активну спільноту розробників, які створюють та підтримують бібліотеки, фреймворки та інші інструменти, що значно полегшують розробку веб-додатків та систем;

5) широкі можливості інтеграції: Python добре інтегрується з багатьма іншими технологіями та інструментами, що дозволяє створювати більш складні та функціональні системи.

Python має багато популярних фреймворків, які допомагають швидко розробляти веб-додатки, забезпечують безпеку та зручність використання. Ось декілька найпопулярніших фреймворків для Python.

1) Django - це повнофункціональний веб-фреймворк, який надає різноманітні можливості для розробки веб-додатків. Django має широкий спектр вбудованих функцій, таких як аутентифікація, адміністративна панель, маршрутизація URL, створення форм, шаблонів та багато іншого. Крім того, Django є дуже популярним веб-фреймворком серед розробників, що дозволяє швидко знайти відповіді на різноманітні питання та проблеми завдяки великій кількості ресурсів та спільноти розробників, які активно допомагають одне одному. Крім того, Django надає можливість розробляти веб-додатки відповідно до патерну Model-View-Controller (MVC), що дозволяє розділити логіку додатку на три окремі компоненти, що спрощує розробку та підтримку коду;

2) Flask - це легкий веб-фреймворк, який дозволяє розробляти веб-додатки швидко та ефективно. Flask має простий та зрозумілий синтаксис та надає можливість розширення функціональності за допомогою плагінів. Також слід зазначити, що Flask не нав'язує певного формату організації проекту, що дає розробникам більшу свободу при роботі з ним. Flask є чудовим вибором для створення простих веб-додатків, які не потребують великих обсягів коду та використовуються в основному для прототипування та швидкого розгортання простих проектів. Flask також має багато різноманітних розширень, що

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		30

дозволяють забезпечити різноманітність функціональних можливостей веб-додатків;

3) Pyramid - це гнучкий веб-фреймворк, який підтримує різні технології та підходи до розробки веб-додатків. Pyramid надає велику свободу вибору та дозволяє розробникам вибирати найкращі інструменти для конкретної задачі. Він надає можливість розробникам використовувати різні підходи до розробки, такі як Model-View-Controller (MVC), Model-View-ViewModel (MVVM), або інші. Крім того, Pyramid має багато розширень та бібліотек, які допомагають розробникам виконувати різні завдання, такі як авторизація, кешування, робота з базами даних та інші. Також, Pyramid має вбудовану підтримку RESTful API, що дозволяє розробляти додатки, які можуть взаємодіяти з іншими додатками за допомогою стандартних протоколів та форматів даних;

4) CherryPy - це мінімалістичний веб-фреймворк, який дозволяє створювати веб-додатки за допомогою мінімального коду. CherryPy має вбудовану підтримку статичних та динамічних контентів, а також можливість роботи з різними форматами даних. CherryPy є одним з найпростіших веб-фреймворків Python. Він має досить низький рівень абстракції, тому дозволяє розробникам використовувати мінімальну кількість коду для створення веб-додатків. Завдяки цьому CherryPy є досить швидким і ефективним веб-фреймворком. CherryPy підтримує статичний контент (такий як зображення, стилі та скрипти), динамічний контент (такий як HTML-сторінки, які генеруються додатком) та динамічну маршрутизацію URL. Фреймворк також має вбудовану підтримку JSON, XML та інших форматів даних. Основна перевага CherryPy полягає в тому, що він має дуже маленький розмір та мінімальну кількість залежностей. Це дозволяє розробникам зосередитись на функціональності додатку, а не на встановленні та налагодженні фреймворку. Крім того, CherryPy дозволяє розробникам розширювати функціональність фреймворку за допомогою власних бібліотек та плагінів. Однак, через низький рівень абстракції та мінімалістичність, розробка складних веб-додатків в CherryPy може бути важкою та вимагати більшої кількості коду порівняно з іншими веб-фреймворками, такими як Django чи Flask;

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		31

5) Tornado - це веб-фреймворк, який спеціалізується на розробці веб-додатків, що вимагають великої кількості одночасних з'єднань. Tornado має вбудовану підтримку асинхронної обробки запитів та дозволяє розробникам ефективно створювати високопродуктивні веб-додатки. Він часто використовується для створення різних типів додатків, таких як чат-боти, стрімінгові сервіси, а також веб-сервери, що обробляють великий потік запитів. Також він пропонує широкий спектр вбудованих інструментів для роботи з HTTP, WebSockets та іншими протоколами, що дозволяє легко та ефективно розробляти веб-додатки зі складним функціоналом. Tornado використовується в багатьох веб-проектах, в тому числі в Dropbox, FriendFeed, Pinterest та інших компаніях. Він є чудовим вибором для розробки веб-додатків, які потребують високої продуктивності та можливості обробки великої кількості запитів.

Отже, використання Python для створення архітектури IT-інфраструктури підприємств для веб-орієнтованої інформаційної системи управління готелем має також перевагу у забезпеченні високої швидкодії та масштабованості системи. Python є досить швидкою мовою програмування, яка може ефективно обробляти великі обсяги даних та працювати з розподіленими системами. Крім того, Python має багато бібліотек та фреймворків, які сприяють швидкому та ефективному розробленню веб-додатків, таких як Django, Flask, Pyramid тощо. Таким чином, використання Python для створення архітектури IT-інфраструктури підприємства для веб-орієнтованої інформаційної системи управління готелем може допомогти забезпечити високу продуктивність та масштабованість системи.

На діаграмі можна зобразити загальну структуру роботи Python із програмними модулями та залежностями. Основна структура роботи Python може бути поділена на наступні етапи:

- 1) інтерпретація: на першому етапі програмний код написаний на мові Python інтерпретується за допомогою вбудованого інтерпретатора Python;
- 2) виконання: після інтерпретації коду, Python запускає виконання програми та завантажує необхідні модулі та бібліотеки;
- 3) перевірка модулів: під час виконання програми Python перевіряє наявність та правильність використання залежних модулів та бібліотек;

4) збірка сміття: Python включає автоматичний збірник сміття, який відслідковує використання об'єктів та прибирає непотрібні об'єкти під час роботи програми;

5) робота з файлами та мережею: Python може працювати з файловою системою та мережевими протоколами для зчитування та запису даних;

6) робота з базами даних: Python має багато вбудованих та зовнішніх бібліотек для роботи з реляційними та нереляційними базами даних;

7) робота з графічним інтерфейсом користувача: Python має багато бібліотек для розробки графічного інтерфейсу користувача, таких як PyQt, Tkinter, wxPython та ін.

Нижче наведено діаграму, що ілюструє загальну структуру роботи Python (рис 2.3).



Рисунок 2.3 – Діаграма основного процесу виконання коду на Python

Ця діаграма є повною і відображає основний процес виконання коду на Python. Однак, варіанти діаграм можуть відрізнятися залежно від того, які деталі включені в процес виконання, і які етапи більше виділяються. Наприклад, деякі діаграми можуть включати додаткові кроки, такі як аналіз коду під час інтерпретації, або збір інформації про пам'ять, витрачену програмою під час виконання. Однак, основні етапи виконання залишаються незмінними.

ІТ-інфраструктура підприємства є критично важливою складовою бізнесу та повинна бути розглянута як інвестиція в ефективну та безпечну роботу підприємства. Правильно спроектована та належно підтримувана ІТ-інфраструктура може допомогти підприємству досягти більшої продуктивності, забезпечити більшу доступність та надійність даних, знизити витрати та підвищити конкурентоспроможність.

Отже, архітектура ІТ-інфраструктури підприємства для веб-орієнтованої інформаційної системи управління готелем повинна бути розроблена з урахуванням потреб користувачів, розміру та потужності мережі, вимог до безпеки та доступності даних, а також можливості масштабування та резервування системи. Діаграми архітектури допоможуть візуалізувати та розуміти структуру та взаємодію компонентів системи, що сприятиме ефективному проектуванню та управлінню ІТ-інфраструктурою підприємства.

2.2 Функційні вимоги

Функційні вимоги - це вимоги, що визначають функції або операції, які повинні бути виконані системою, програмою або продуктом. Ці вимоги зазвичай описуються у вигляді списку функцій або операцій, які повинні бути реалізовані у системі.

Функційні вимоги зазвичай формулюються на основі вимог бізнесу та потреб користувачів. Наприклад, для інтернет-магазину функційні вимоги можуть включати можливість здійснення замовлень, оплати, відстеження статусу замовлення, роботу зі списком бажань тощо.

Для того, щоб вимоги були чітко сформульовані, вони повинні відповідати так званим SMART-критеріям (Specific, Measurable, Achievable, Relevant, Time-bound). Це допомагає уникнути недорозумінь та помилок під час розробки системи.

Функційні вимоги також можуть бути класифіковані за рівнем важливості або пріоритету. Наприклад, деякі функції можуть бути визначені як критичні,

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		34

тобто вони повинні бути реалізовані в першу чергу, тоді як інші можуть мати менший пріоритет і виконуватися в другому етапі розробки.

Основна мета функційних вимог - забезпечити повну та коректну роботу системи, продукту або програми та задовольнити потреби користувачів та бізнесу (див. табл. 2.1).

Таблиця 2.1 – Функційні вимоги

Функції	Опис
Бронювання номерів	Система повинна забезпечувати можливість онлайн-бронювання номерів готелю, включаючи вибір типу номера, дати приїзду та виїзду, кількість гостей тощо
Облік та управління номерами	Система повинна забезпечувати можливість управління вільними та зайнятими номерами готелю, додавання нових номерів до системи, редагування та видалення існуючих
Управління резерваціями	Система повинна дозволяти переглядати, редагувати та видаляти наявні резервації, а також здійснювати повернення грошей за скасовані бронювання
Управління клієнтами	Система повинна зберігати дані про клієнтів готелю, включаючи особисту інформацію та історію перебування в готелі
Управління платежами	Система повинна забезпечувати можливість оплати за проживання в готелі, включаючи оплату карткою, електронними коштами та готівкою
Забезпечення безпеки	Система повинна забезпечувати безпеку персональних даних гостей, а також забезпечувати безпеку оплат та транзакцій

Кінець таблиці 2.1 – Функційні вимоги

Інтеграція з платіжними системами	Система повинна мати можливість підключення до платіжних систем для оплати бронювань та інших послуг, які надає готель
Можливість роботи на різних пристроях та браузерах	Система повинна бути адаптована до роботи на різних пристроях та браузерах, щоб забезпечити зручний доступ для користувачів з будь-якого пристрою та забезпечити максимальну доступність системи
Звітність	Система повинна забезпечувати можливість створення звітів про фінансову діяльність готелю, статистики про роботу готелю та інші важливі показники, що допоможуть власникам та керівникам готелю приймати управлінські рішення та планувати діяльність на майбутнє
Управління замовленнями	Система повинна забезпечувати можливість обробки замовлень на резервування номерів, зміни, скасування замовлень

Ці функційні вимоги дозволяють забезпечити ефективну роботу готельної інформаційної системи та забезпечити якісне обслуговування клієнтів готелю. Наприклад, бронювання номерів дозволяє клієнтам зарезервувати номери заздалегідь, що забезпечує готелю більш ефективне планування роботи та зменшення кількості порожніх номерів. Управління платежами та забезпечення безпеки допомагають захистити фінансові інтереси готелю та його клієнтів. Звітність та управління замовленнями дозволяють відслідковувати ефективність роботи готелю та дозволяють планувати подальші дії щодо його розвитку. Всі ці функціональні вимоги є важливими для успішної роботи готелю та його інформаційної системи.

2.3 Нефункційні вимоги

Нефункційні вимоги - це вимоги, що визначають характеристики та властивості системи, програми або продукту, які не стосуються їх функціональності. Ці вимоги можуть включати такі характеристики, як надійність, безпека, швидкість, масштабованість, доступність, сумісність, легкість використання та інші.

Нефункційні вимоги допомагають забезпечити коректну та ефективну роботу системи, програми або продукту в різних умовах використання, а також задовольнити потреби користувачів та бізнесу. Наприклад, надійність та безпека є критичними нефункційними вимогами для банківської системи, тоді як швидкість та масштабованість можуть бути важливі для веб-сайту з великою кількістю відвідувачів.

Враховання нефункційних вимог є важливою складовою процесу розробки системи, програми або продукту. Оскільки ці вимоги не стосуються безпосередньо функціональності системи, їх визначення та формулювання можуть бути трохи складнішими, але вони не менш важливі для успішної реалізації проекту.

Нефункційні вимоги можуть бути визначені на основі стандартів, правил та рекомендацій, а також враховувати вимоги безпеки та захисту даних. Важливо, щоб нефункційні вимоги були чітко сформульовані та відповідали потребам бізнесу та користувачів. Крім того, їх слід враховувати під час проектування та розробки системи, програми або продукту, а також під час тестування та оцінки їх якості (див. табл. 2.2).

Таблиця 2.2 – Нефункційні вимоги

Фунції	Опис
Надійність	Система повинна бути надійною та стабільною, з низьким рівнем відмов та можливістю відновлення після відмови.

Кінець таблиці 2.2 – Нефункційні вимоги

Безпека		система повинна мати високий рівень безпеки, забезпечуючи захист від несанкціонованого доступу до даних, збереження конфіденційності та цілісності даних.
Швидкодія		система повинна бути швидкою та ефективною в роботі, з короткими термінами відповіді на запити користувачів.
Легкість використання		система повинна бути легкою використовувати та інтуїтивно зрозумілою для користувачів, що забезпечить їхню задоволеність від користування.
Сумісність законодавством	з	система повинна відповідати вимогам законодавства щодо захисту персональних даних та інших вимог щодо збереження даних.
Наявність підтримки		система повинна мати належну технічну підтримку та документацію для користувачів.
Ефективність продуктивність	та	система повинна працювати швидко та ефективно навіть при великій кількості користувачів та обсязі даних. Також, час відповіді системи на запити користувачів повинен бути прийнятним та не займати більше певного часу.
Масштабованість		система повинна бути масштабованою, з можливістю легкої розширення та підтримки великої кількості користувачів.
Надійність та стійкість		система повинна бути стійкою до помилок та відмов, а також забезпечувати збереження даних та їх відновлення у випадку аварійної ситуації.

Ці нефункційні вимоги визначають якість та ефективність системи. Надійність та стійкість забезпечують безперебійну роботу системи та захист від можливих помилок та збоїв. Безпека забезпечує захист інформації та даних

клієнтів. Швидкодія та ефективність дозволяють користувачам швидко та легко взаємодіяти з системою. Легкість використання допомагає користувачам швидко зрозуміти, як користуватися системою. Сумісність з законодавством дозволяє системі відповідати всім необхідним правовим вимогам. Наявність підтримки забезпечує користувачам необхідну допомогу та підтримку у використанні системи. Масштабованість та сумісність забезпечують можливість розширення та адаптації системи до змінних потреб користувачів. Звітність дозволяє готелю отримувати необхідну інформацію про свою діяльність та планувати подальші кроки. Управління замовленнями допомагає готелю керувати своєю діяльністю та оптимізувати роботу.

2.4 Висновки

У процесі аналізування принципів та методів проектування веб-орієнтованої інформаційної системи було висвітлено багато аспектів, які потрібно враховувати при розробці системи. Зокрема, було досліджено функційні та нефункційні вимоги до веб-орієнтованої інформаційної системи управління готелем.

Функційні вимоги визначають функції, які повинні бути виконані системою, програмою або продуктом. Нефункційні вимоги відносяться до характеристик системи, таких як швидкість роботи, безпека, доступність і т.д.

Також було визначено організаційні вимоги до продукту, такі як вимоги до кваліфікації персоналу, можливість інтеграції з іншими системами та інші.

Важливо враховувати SMART-критерії при формулюванні вимог, щоб уникнути недорозумінь та помилок під час розробки системи.

При проектуванні веб-орієнтованої інформаційної системи управління готелем потрібно враховувати всі вимоги та розробити оптимальну архітектуру, яка задовольнить потреби користувачів та бізнесу. Також важливо розробити план тестування та забезпечення якості продукту.

У процесі проектування важливо також дотримуватись принципів SOLID та Clean Architecture, щоб забезпечити простоту, масштабованість та підтримуваність системи в майбутньому.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						39
Зм..	Арк.	№докум.	Підпис	Дата		

Отже, при проектуванні веб-орієнтованої інформаційної системи управління готелем потрібно враховувати багато аспектів, щоб забезпечити якість та ефективність системи. Важливо бути уважним при формулюванні вимог та виборі технологій, а також розробити оптимальну архітектуру системи, яка забезпечить її стійкість, масштабованість та безпеку. Крім того, необхідно враховувати вимоги користувачів, бізнесу та ринку, а також дотримуватися процесів розробки програмного забезпечення та стандартів якості. У процесі розробки необхідно проводити тестування та внесення змін, які забезпечать оптимальну роботу системи. В цілому, успішне проектування програмно-технічного засобу потребує комплексного підходу та залучення фахівців з різних областей.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		40

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ВЕБ-ОРІЄНТОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Обґрунтування вибору засобів розробки веб-орієнтованої інформаційної системи

Вибір засобів розробки веб-орієнтованої інформаційної системи (ІС) зазвичай залежить від різних факторів, таких як тип ІС, розмір проекту, бюджет, наявність необхідних навичок та ресурсів, терміни реалізації проекту, наявність попереднього досвіду з розробкою ІС і так далі.

Одним з найпоширеніших засобів розробки ІС є мова програмування Java. Java є мовою з відкритим вихідним кодом і має велику спільноту розробників, що забезпечує доступ до великої кількості бібліотек і фреймворків. Java також має високу портативність, що означає, що програми, написані на Java, можуть працювати на різних платформах, що робить його ідеальним вибором для ІС.

Ще одним популярним засобом розробки ІС є мова програмування Python. Python також має відкритий вихідний код та велику спільноту розробників. Python є дуже простою та зрозумілою мовою програмування, що робить його вибором для швидкої розробки прототипів та маленьких проектів.

Також можна використовувати фреймворки, такі як Ruby on Rails, Django, Flask і Laravel. Ці фреймворки роблять процес розробки більш швидким і ефективним, забезпечують велику кількість функціональності та вбудовані інструменти для роботи з базами даних та іншими компонентами.

Крім того, для розробки ІС можна використовувати такі інструменти, як Node.js, Angular, React та Vue.js для розробки фронтенду. Вони забезпечують багато функціональності та допомагають створювати користувацький інтерфейс ІС, що дуже важливо для забезпечення користувачам зручного та простого використання системи.

Вибір конкретних засобів розробки ІС залежить від вимог проекту та вмінь розробників. Наприклад, якщо проект потребує великої кількості обчислень та маніпулювання даними, то може бути краще використовувати мову

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		41

програмування з низькорівневим доступом до пам'яті, таку як C++ або Rust. Якщо ж проект потребує швидкої розробки та простоти коду, то можна використовувати мову програмування з високорівневим доступом до функцій, наприклад, Python або Ruby.

Також важливо звернути увагу на підтримку та розвиток обраних засобів розробки. Наявність активної спільноти розробників та регулярні оновлення допоможуть забезпечити стабільну роботу ІС протягом тривалого часу та знизити ризик виникнення помилок та проблем в майбутньому.

Узагалі, вибір засобів розробки ІС повинен бути обґрунтований та здійснений на основі аналізу вимог проекту та доступних ресурсів та навичок команди розробників.

Я обрав Python. Тому що обрання Python для розробки інформаційної системи може мати свої переваги. Python є високорівневою мовою програмування з простим синтаксисом, що дозволяє швидко та ефективно розробляти програми. Крім того, Python має широкий набір сторонніх бібліотек та фреймворків, що дозволяє з легкістю реалізувати різноманітні функції та функціональність системи.

Python також має хорошу підтримку серед спільноти розробників, що означає, що завжди можна знайти багато корисних ресурсів, які допоможуть розв'язувати проблеми та вирішувати завдання під час розробки. Крім того, наявність безкоштовних та відкритих інструментів розробки та інтегрованих середовищ (IDE) таких як PyCharm, Visual Studio Code, Jupyter Notebook, робить розробку в Python зручною та простою.

Однак, слід зазначити, що Python має свої обмеження, зокрема швидкість виконання програм, тому якщо ваша система має бути високопродуктивною та масштабованою, можливо, буде необхідно поєднувати Python з іншими мовами програмування або використовувати спеціалізовані фреймворки.

Загалом, обрання Python для розробки інформаційної системи може бути хорошим варіантом, якщо він відповідає вимогам вашого проекту та вам зручно працювати з цією мовою програмування.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		42

Технології які було використано при розробці інформаційної системи готелю:

1) Tkinter - це бібліотека інтерфейсу користувача (UI), що дозволяє розробляти графічні інтерфейси користувача для програм в Python. Tkinter є стандартним модулем Python, що дозволяє з легкістю створювати вікна, кнопки, текстові поля та інші елементи інтерфейсу.

2) ttk (Themed Tkinter) - це розширення Tkinter, що дозволяє створювати більш сучасний та стильний вигляд інтерфейсу користувача за допомогою тем оформлення.

3) time та datetime - ці модулі Python дозволяють отримувати та обробляти часову інформацію. Модуль time містить функції для роботи з часом у форматі UNIX, а модуль datetime надає класи для роботи з датою та часом в різних форматах.

4) PIL (Python Imaging Library) - це бібліотека для роботи з зображеннями у форматі растрової графіки, що дозволяє завантажувати, зберігати та обробляти зображення в різних форматах.

5) os - це модуль Python, що дозволяє взаємодіяти з операційною системою. Він надає функції для створення та редагування файлів та каталогів, а також для взаємодії з оточенням операційної системи.

6) sqlite3 - це модуль Python для роботи з базами даних SQLite. SQLite - це легковага, вбудовувана база даних, яка дозволяє зберігати та організувати дані в інтерактивному режимі.

7) messagebox - це модуль Tkinter, що дозволяє створювати вікна повідомлень, що містять текст та інші елементи інтерфейсу.

Технології, які були використані при розробці інформаційної системи готелю, є оптимальними з кількох причин:

1) Tkinter та ttk є стандартними бібліотеками для розробки графічних інтерфейсів користувача в Python, що дозволяє забезпечити простоту та зручність розробки інтерфейсу. Оскільки обидві бібліотеки розроблені для роботи разом, використання ttk для створення сучасного вигляду інтерфейсу є більш простим та ефективним.

2) Модулі time та datetime дозволяють легко та точно отримувати, обробляти та зберігати часову інформацію, що є необхідним для роботи з датами та часом у системі готелю.

3) PIL дозволяє легко та швидко робити операції з зображеннями, такі як завантаження, збереження та обрізання, що є необхідним для роботи з фотографіями готелю.

4) Модуль os дозволяє взаємодіяти з операційною системою, що є необхідним для доступу до файлової системи та зчитування конфігураційних файлів системи готелю.

5) База даних SQLite є легкою та вбудовуваною, що дозволяє легко та швидко зберігати та організовувати дані системи готелю.

6) Модуль messagebox дозволяє легко створювати вікна повідомлень та інші елементи інтерфейсу, що дозволяє забезпечити зручну та логічну взаємодію користувача з системою готелю.

У цілому, використання цих технологій дозволяє забезпечити швидку та ефективну розробку інформаційної системи готелю з високоякісним та зручним користувацьким інтерфейсом, що в свою чергу забезпечує зручну та ефективну роботу з системою для користувачів.

Загалом, вибір конкретних технологій для розробки інформаційної системи готелю залежить від багатьох факторів, таких як потреби бізнесу, технічні можливості, досвід розробника та багато інших. Проте, використання вже зарекомендованих технологій та бібліотек, які підтримуються та оновлюються спільнотою розробників, може значно спростити та прискорити розробку проекту, зменшити кількість помилок та забезпечити високу якість програмного забезпечення.

3.2 Вимоги до апаратного і програмного забезпечення

Для оптимальної роботи інформаційної системи готелю необхідне відповідне апаратне та програмне забезпечення. Нижче наведено загальні вимоги

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						44
Зм..	Арк.	№докум.	Підпис	Дата		

до апаратного та програмного забезпечення для роботи інформаційної системи готелю:

Апаратне забезпечення:

1) Процесор: Intel Core i3 або вище - вибір процесора залежить від вимог програмного забезпечення, яке використовується в системі. Процесор з можливістю гіпертрейдингу, який дозволяє збільшити продуктивність системи, є перевагою.

2) Оперативна пам'ять: 4 ГБ або більше - чим більше оперативної пам'яті в системі, тим ефективніше вона працює. Для інформаційної системи готелю достатньо 4 ГБ пам'яті.

3) Жорсткий диск: 500 ГБ або більше - розмір жорсткого диска повинен відповідати обсягу даних, що зберігаються в системі. Для інформаційної системи готелю достатньо 500 ГБ дискового простору.

4) Монітор: роздільна здатність 1024x768 пікселів або більше - монітор повинен мати достатню роздільну здатність для зручної роботи з системою.

5) Миша та клавіатура - ці компоненти не мають спеціальних вимог і можуть бути звичайними комп'ютерними пристроями.

Програмне забезпечення:

1) Операційна система: Windows 10 або Linux - вибір операційної системи залежить від більшості факторів, таких як потреби бізнесу, досвід розробника та бібліотек, які використовуються в системі.

2) Python 3.x - це мова програмування, яка використовується для розробки інформаційної системи готелю.

3) Бібліотеки: Tkinter, ttk, time, datetime, PIL, os, sqlite3, messagebox - ці бібліотеки необхідні для розробки графічного інтерфейсу, забезпечення роботи зі зображеннями, роботи з базою даних та іншими завданнями, які повинна виконувати система.

Отже, ці вимоги до апаратного і програмного забезпечення є оптимальними для стабільної та надійної роботи інформаційної системи готелю, тому що використання такого апаратного та програмного забезпечення забезпечує стабільну та надійну роботу інформаційної системи готелю. Використання

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		45

оптимального обладнання і програмного забезпечення дозволить забезпечити швидку та ефективну роботу системи, яка буде легкою у використанні та обслуговуванні. Всі ці вимоги дозволяють забезпечити оптимальну роботу готельної інформаційної системи та забезпечити комфорт для користувачів, що дозволяє підвищити рівень обслуговування та забезпечити задоволення гостей готелю.

3.3 UML діаграми

Діаграма діяльності (Activity diagram) - це один з типів діаграм UML, який використовується для моделювання послідовності дій або процесів, що відбуваються в системі. Ця діаграма може бути корисною для відображення послідовності дій в інформаційній системі готелю, наприклад, для процесу бронювання номеру, реєстрації гостя, оформлення замовлення на харчування та інших дій.

На рисунку 3.1 зображено основні процеси, які відбуваються в системі управління готелем:

- 1) Клієнт приходить до стійки готелю або відкриває онлайн-портал;
- 2) Пошук номерів за датами та типом номеру;
- 3) Клієнт вносить оплату за заброньований номер;
- 4) Пошук інформації про клієнта;
- 5) Бронювання кімнати;
- 6) Клієнт оплачує номер;
- 7) Система підтверджує оплату та позначає заброньований номер як оплачений;
- 8) Клієнт отримує чек про оплату.

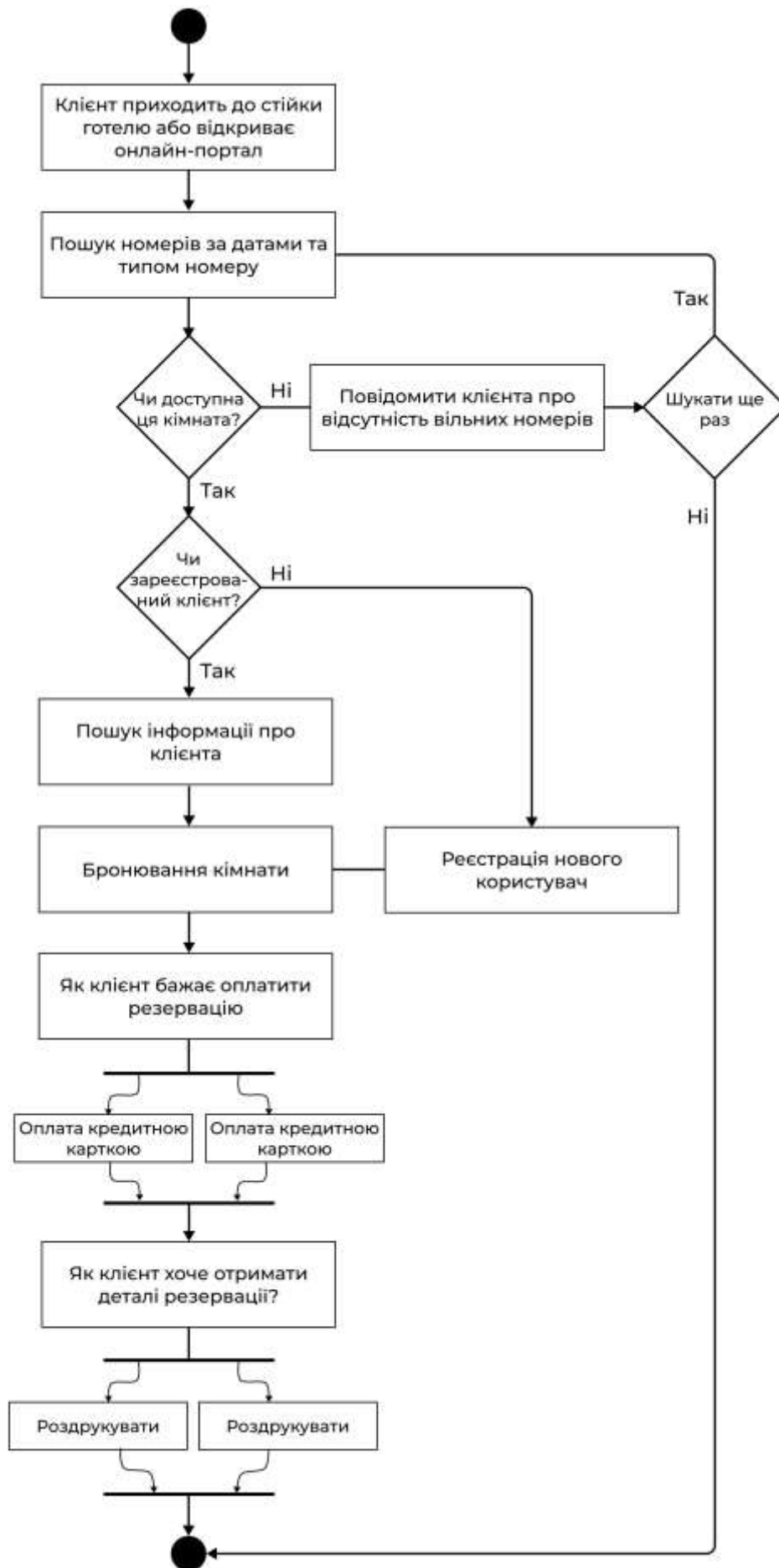


Рисунок 3.1 – Діаграма діяльності системи управління готелем

Діаграма послідовності (Sequence diagram) - це UML-діаграма, яка відображає взаємодію об'єктів в рамках певної операції або функції. Рисунок 3.2 показує послідовність повідомлень, які відправляються між об'єктами, та як ці повідомлення обробляються.



Рисунок 3.2 – Діаграма послідовності (Sequence diagram)



Рисунок 3.3 – Діаграма станів (State diagram)

Діаграма станів (State diagram) - це UML-діаграма, яка відображає стани об'єкту і переходи між ними. Вона використовується для моделювання систем, які мають складну логіку зміни станів об'єктів. Нижче наведена діаграма станів для інформаційної системи готелів.

На рисунку 3.3 зображено сім станів:

- 1) "Вільний" (Free) - номер доступний для бронювання.
- 2) "Заброньований" (Reserved) - номер заброньований, але ще не підтверджений.
- 3) "Очікування" (On hold) - стан, в якому бронювання знаходиться в очікуванні підтвердження.
- 4) "Підтверджений" (Confirmed) – номер підтверджений.
- 5) "Зайнятий" (Occupied) - номер зайнятий гостем.
- 6) "Перепланування" (Replanning) – Зміна планів гостя
- 7) "Відмінено" (Cancelled) - бронювання відмінено.

Також на діаграмі показані переходи між станами:

1. Бронювання номеру переводить номер зі стану "Вільний" в стан "Заброньований".
2. Підтвердження бронювання номеру переводить номер зі стану "Заброньований" в стан "Очікування".
3. Перепланування бронювання зі стану "Очікування" переводить номер у стан "Заброньований" або "Відмінено".
4. Підтвердження бронювання зі стану "Очікування" переводить номер у стан "Зайнятий".
5. Відміна бронювання номеру переводить номер зі стану "Заброньований" у стан "Вільний" або зі стану "Очікування" у стан "Відмінено".

Діаграма компонентів (Component diagram) - це один з типів діаграм UML, який використовується для моделювання архітектури системи та компонентів, з яких вона складається. У контексті інформаційної системи готелю, діаграма компонентів може допомогти візуалізувати складові частини системи та залежності між ними, наприклад, базу даних, веб-сервер, додатки для реєстрації гостей та бронювання номерів.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						49
Зм..	Арк.	№докум.	Підпис	Дата		

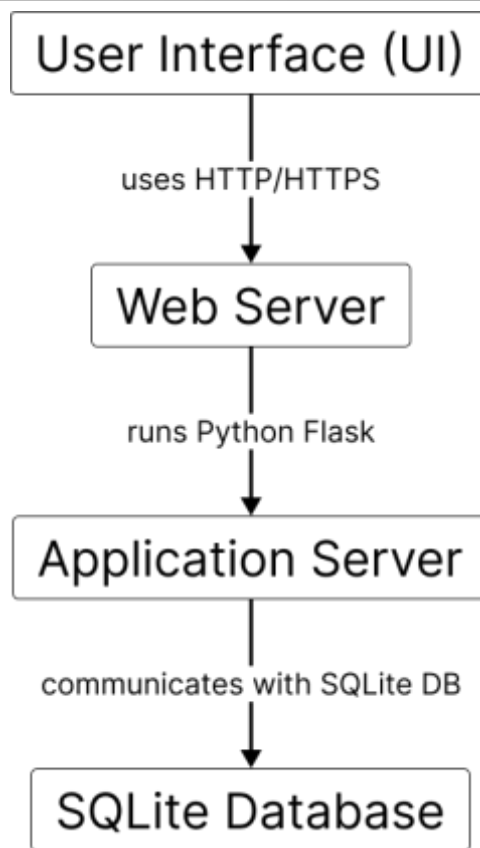


Рисунок 3.4 – Діаграма компонентів (Component diagram)

На рисунку 3.4 зображено чотири компоненти системи - інтерфейс користувача (UI), веб-сервер, сервер додатків та база даних SQLite. Якщо порівнювати з попереднім прикладом, то відсутній сервер додатків та додатково з'явився Python-фреймворк Flask.

Користувач взаємодіє з системою через інтерфейс користувача (UI), який забезпечує можливість бронювання кімнат, перевірки наявності кімнат, відміни бронювання тощо. Інтерфейс користувача використовує веб-сервер, який запускає фреймворк Python Flask для обробки запитів.

Python Flask обробляє запити користувачів та передає їх на базу даних SQLite для зберігання та отримання інформації про кімнати, клієнтів, операції тощо.

Загалом, діаграма компонентів показує, які компоненти є частинами інформаційної системи готелів на базі Python та бази даних SQLite, та як вони взаємодіють між собою для забезпечення роботи системи в цілому. Ця діаграма дозволяє розуміти, як працює система та які компоненти можуть бути замінені

або покращені для поліпшення функціоналу системи. Наприклад, в майбутньому можна було б підвищити продуктивність системи шляхом заміни бази даних SQLite на більш потужну реляційну базу даних, таку як PostgreSQL чи MySQL.

Діаграма компонентів також допомагає розбити систему на окремі компоненти, що полегшує її розробку та тестування. Наприклад, розробник може працювати над окремим компонентом, таким як інтерфейс користувача, не втручаючись в роботу інших компонентів.

У загальному, діаграма компонентів є потужним інструментом для розуміння та розробки складних систем, таких як інформаційні системи готелів, та допомагає забезпечити якість та ефективність розробки.

Діаграма варіантів використання (Use Case Diagram) є графічним зображенням функціональності або можливостей системи з точки зору її акторів. Вона використовується для моделювання взаємодії між користувачами системи (акторами) та самою системою. Діаграма варіантів використання дозволяє уявити, як система буде використовуватись в реальному світі та які функціональності будуть доступні різним акторам (рис 3.5).



Рисунок 3.5 – Діаграма варіантів використання (Use Case Diagram)

На цій діаграмі показано, що інформаційна система готелю має різні варіанти використання, доступні як користувачу, так і адміністратору. Обидва актори можуть взаємодіяти з такими функціями як "Статус готелю", "Доступні Кімнати", "Резервація", "Інформація про платіжи" та "Контакти працівників готелю".

У додаток до цього, адміністратор має додаткові можливості, такі як "Додавання/Видалення кімнати", "Зміна статусу кімнати" та "Редагування бази даних", що дозволяє йому керувати інформацією про кімнати та забезпечувати правильну роботу системи.

Ця діаграма варіантів використання допомагає краще зрозуміти, як користувачі та адміністратори взаємодіють з системою та які функції доступні для кожної ролі.

3.4 Структура бази даних

База даних - це організована колекція даних, які зберігаються та управляються в електронному форматі. Вона дозволяє ефективно зберігати, організовувати та отримувати доступ до інформації.

Бази даних бувають різних типів, таких як реляційні, об'єктно-орієнтовані, графові та інші. Реляційні бази даних - найпоширеніший тип, де дані зберігаються у вигляді таблиць зі зв'язками між ними. Управління такими базами даних відбувається за допомогою мов SQL (Structured Query Language).

SQLite - це реляційна база даних, яка зберігає дані у локальних файлах, не вимагаючи окремого сервера баз даних. Вона є легкою у використанні, швидкою та надійною, що робить її ідеальним вибором для невеликих проектів, де потрібно зберігати обмежену кількість даних.

Обрання SQLite для інформаційної системи готелю є оптимальним рішенням, оскільки вона забезпечує швидкий та надійний доступ до даних, має невисокі вимоги до ресурсів, а також не потребує додаткового програмного забезпечення для своєї роботи (рис 3.6).

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		52

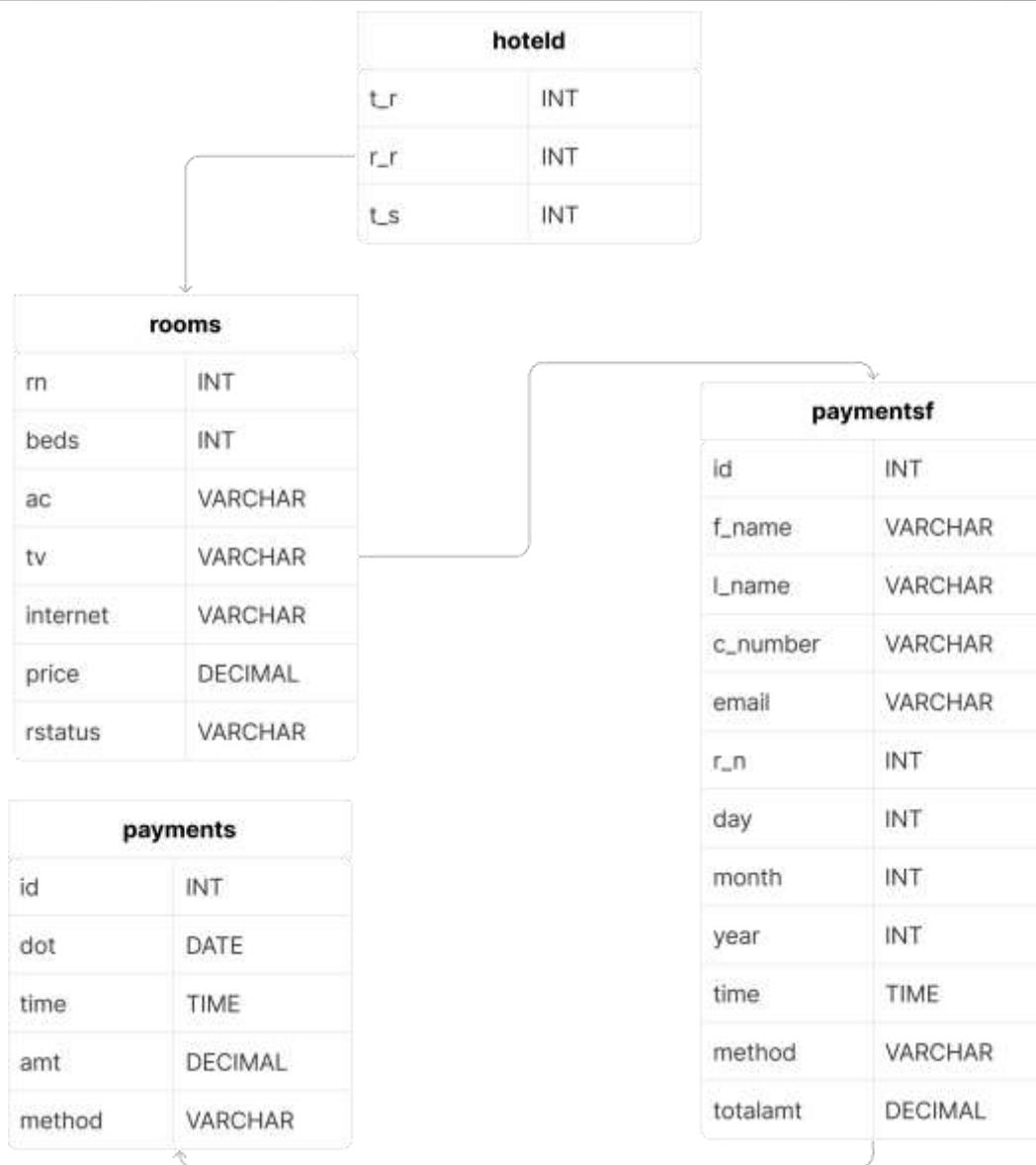


Рисунок 3.6 – Схема бази даних SQL

Складові структури бази даних для Інформаційної системи готелю:

Таблиця "hoteld"

- 1) t_r (скільки всього кімнат в готелі);
- 2) r_r (скільки кімнат зарезеровано);
- 3) t_s (скільки всього персоналу в готелі).

Таблиця "payments"

- 1) id (унікальний ідентифікатор платежу);
- 2) dot (дата оплати);
- 3) time (час оплати);
- 4) amt (сума оплати);

5) method (метод оплати).

Таблиця "paymentsf"

- 1) "id" - унікальний ідентифікатор платежу (первинний ключ);
- 2) "f_name" - ім'я платника;
- 3) "l_name" - прізвище платника;
- 4) "c_number" - номер кредитної картки платника;
- 5) "email" - електронна адреса платника;
- 6) "r_n" - номер кімнати, за яку здійснюється оплата;
- 7) "day" - день оплати;
- 8) "month" - місяць оплати;
- 9) "year" - рік оплати;
- 10) "time" - час оплати;
- 11) "method" - метод оплати (наприклад, "кредитна картка", "готівка" і т.д.);
- 12) "totalamt" - загальна сума оплати.

Таблиця "rooms"

- 1) "rn" - номер кімнати (первинний ключ);
- 2) "beds" - кількість ліжок у кімнаті;
- 3) "ac" - наявність кондиціонера (так/ні);
- 4) "tv" - наявність телевізора (так/ні);
- 5) "internet" - наявність Інтернету (так/ні);
- 6) "price" - вартість номеру за ніч;
- 7) "rstatus" – статус номеру.

Таким чином, Інформаційна система готелю має такі таблиці в базі даних: "hoteld", "payments", "paymentsf", та "rooms". Кожна таблиця містить певну інформацію про кімнати, платежі та клієнтів готелю, а зв'язки між таблицями реалізуються через унікальні ідентифікатори. За допомогою такої бази даних можна ефективно зберігати та організувати інформацію про клієнтів та їх бронювання, забезпечуючи легкий доступ та швидкий пошук інформації, а також можливість створення звітів та аналітики для прийняття рішень.

3.5 Програмна реалізація веб-орієнтованої інформаційної системи

Програмна реалізація інформаційної системи готелю базується на мові програмування Python та використовує різноманітні бібліотеки, такі як Tkinter, ttk, time, datetime, PIL, os, sqlite3, messagebox.

Для забезпечення роботи з базою даних використовується SQLite. Усі дані про кімнати, клієнтів, персонал та бронювання зберігаються в таблицях бази даних. В програмі забезпечена можливість додавання, видалення та зміни даних у базі даних.

Інтерфейс користувача реалізований за допомогою бібліотеки Tkinter та її додаткової тематичної бібліотеки ttk. За допомогою цих бібліотек було розроблено графічний інтерфейс користувача зі всіма необхідними елементами: кнопки, поля введення, список, календар і т. д. Всі функції програми доступні через головне меню.

Для забезпечення функції пошуку вільних номерів з вибраними зручностями використовується SQL-запит до бази даних, який повертає список номерів, що відповідають критеріям пошуку.

Загальна структура програмного коду системи містить класи, які відображають структуру бази даних, та класи, що реалізують логіку програми. Для забезпечення максимальної читабельності та ефективності програмного коду використовуються принципи об'єктно-орієнтованого програмування (ООП).

Загалом, програмна реалізація інформаційної системи готелю є ефективною та зручною для користувача завдяки використанню потужних інструментів програмування та баз даних.

Окремо варто зупинитись на деяких ключових моментах програмної реалізації інформаційної системи готелю:

1. Реалізація бронювання номерів: Для реалізації бронювання номерів було створено відповідний модуль, який забезпечує збереження інформації про бронювання в базі даних. Крім того, була реалізована функція перевірки доступності номерів на задані дати.

2. Реалізація пошуку вільних номерів з вибраними зручностями: Для реалізації функції пошуку вільних номерів з вибраними зручностями було розроблено відповідний модуль, який виконує пошук в базі даних за вказаними параметрами (зручності, дати перебування, кількість гостей тощо) і повертає список доступних номерів з відповідними параметрами та цінами.

3. Реалізація інтерфейсу користувача: Для інтерфейсу користувача було використано Tkinter, що дозволяє створювати графічний інтерфейс на Python. У програмі використовується головне вікно, яке містить різні вкладки для навігації, а також вікна для додавання, редагування та перегляду інформації про номери, клієнтів, працівників та інше.

Отже, програмна реалізація інформаційної системи готелю включає реалізацію модулів бронювання номерів, пошуку вільних номерів з вибраними зручностями, керування оплатою та інтерфейсом користувача. Для реалізації було використано мову програмування Python, бібліотеку Tkinter для створення графічного інтерфейсу та базу даних SQLite для збереження інформації про номери, клієнтів, працівників та інше. Програмна реалізація інформаційної системи готелю дозволяє зручно керувати бронюванням номерів, переглядати список вільних номерів з вибраними зручностями, керувати персоналом та зберігати інформацію в базі даних для подальшого аналізу. До основних переваг розробленої системи можна віднести зручність та швидкість роботи, високу точність та надійність зберігання інформації, можливість простої настройки та розширення функціоналу, а також зручний та інтуїтивно зрозумілий інтерфейс для користувачів.

3.6 Інтерфейс веб-орієнтованої інформаційної системи

Інтерфейс веб-орієнтованої інформаційної системи готелю грає дуже важливу роль у забезпеченні комфортного користування та ефективного використання системи. Його головною метою є спрощення та зручність взаємодії користувача з системою. Інтерфейс повинен бути простим у використанні та надавати необхідну інформацію користувачеві в чіткому та доступному форматі.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		56

Екран "Статус готелю" є важливою складовою частиною інформаційної системи готелю. Цей екран містить наступну інформацію:

- 1) Усі кімнати: цей розділ містить повний перелік кімнат, які належать готелю.
- 2) Доступні кімнати: цей розділ містить інформацію про ті кімнати, які доступні для бронювання в даний момент.
- 3) Зарезервовано: цей розділ містить інформацію про ті кімнати, які були заброньовані.
- 4) Кількість гостей: цей розділ містить інформацію про загальну кількість гостей, які знаходяться в готелі на даний момент.
- 5) Кількість персоналу: цей розділ містить інформацію про кількість працівників готелю, які знаходяться на роботі на даний момент.
- 6) Скільки номерів на ремонті: цей розділ містить інформацію про кількість кімнат, які потребують ремонту, або які знаходяться у стадії ремонту.

В цілому, екран "Статус готелю" є важливим інструментом для керування готелем і дозволяє персоналу готелю ефективно керувати ресурсами та забезпечити задоволення потреб гостей (рис 3.7).

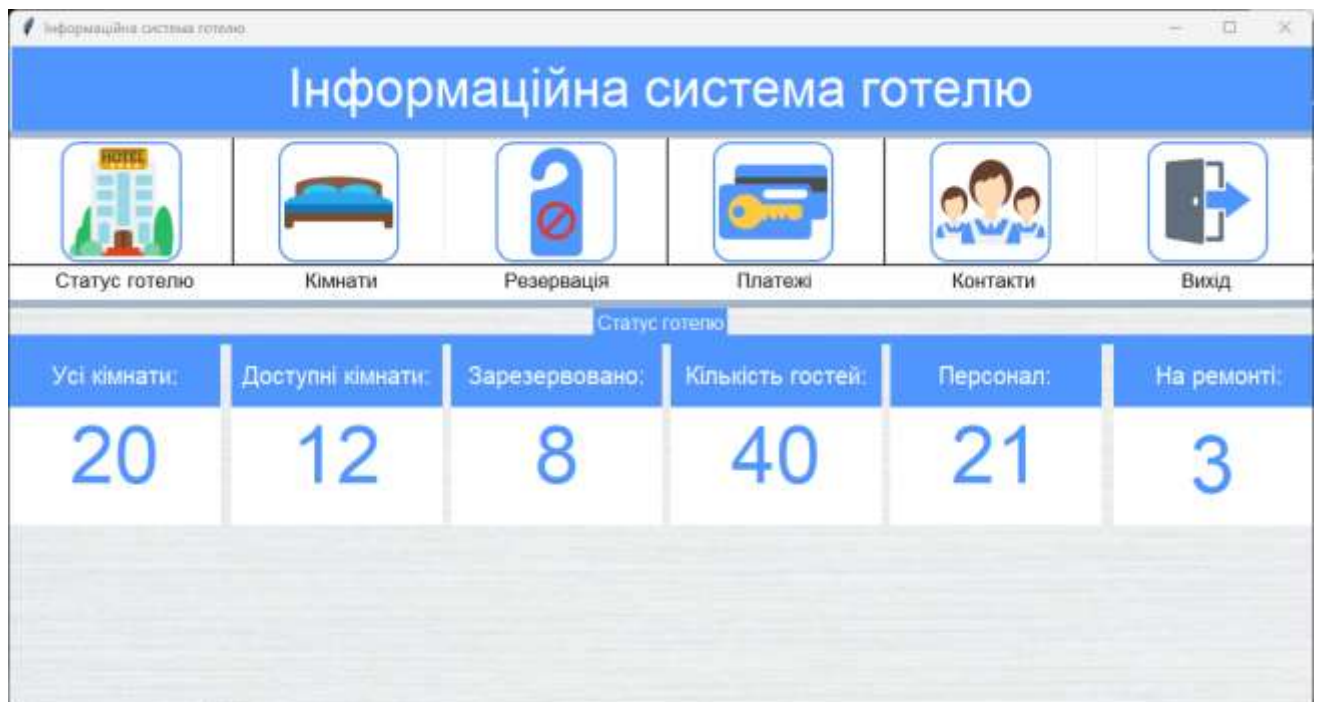


Рисунок 3.7 – Екран "Статус готелю"

Екран "Кімнати" із переліком всіх номерів готелю є важливим компонентом інформаційної системи готелю. Цей екран містить інформацію про кожен номер, що може бути корисним для гостей, які шукають зручний і комфортний номер для перебування в готелі.

При виборі номера на екрані "Кімнати", інформаційна система може показати:

- 1) кількість ліжок в номері
- 2) наявність кондиціонера
- 3) наявність телевізора
- 4) наявність WI-FI
- 5) ціну за номер
- 6) чи заброньований цей номер на даний момент

Ця інформація допоможе гостям готелю прийняти відповідне рішення щодо вибору номера та зробити бронювання відповідно до своїх потреб і бюджету.

Загалом, екран "Кімнати" є важливою складовою інформаційної системи готелю яка допомагає гостям та персоналу готелю керувати процесом бронювання кімнат та покращувати загальний рівень задоволеності гостей (рис 3.8).

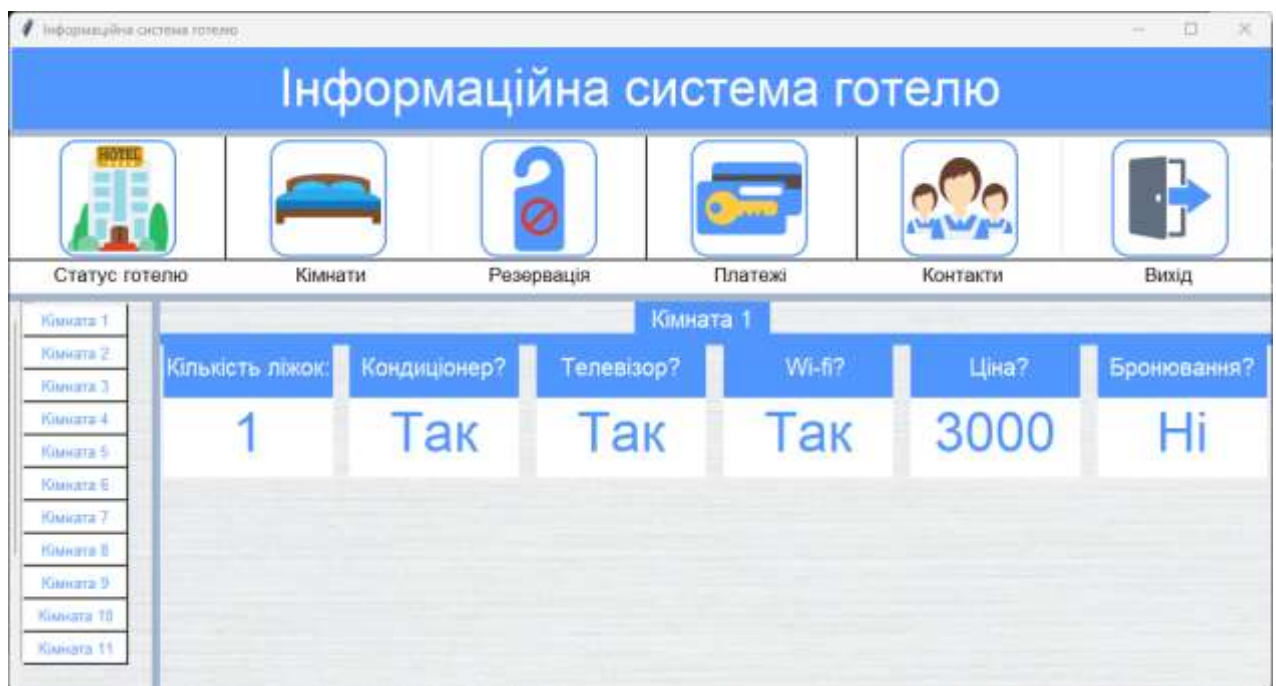


Рисунок 3.8 – Екран "Кімнати"

Екран "Резервація" дозволяє гостям здійснювати бронювання номерів у готелі. Для цього гостям потрібно заповнити форму з особистою і контактною інформацією, а також інформацією про бронювання.

У формі резервації гості можуть ввести своє повне ім'я, по батькові та прізвище, контактну інформацію, таку як номер телефону, електронну пошту та адресу проживання. Також гості повинні вказати інформацію про бронювання, таку як кількість дітей та дорослих, кількість днів проживання та номер кімнати, який вони хотіли б забронювати.

Праворуч на екрані є фільтр, який дозволяє гостям підібрати номер за кількістю ліжок, наявністю кондиціонера, наявністю телевізора та наявністю Wi-Fi. Це допомагає гостям знайти оптимальний варіант номера, що відповідає їхнім потребам та побажанням.

Загалом, екран "Резервація" дозволяє гостям легко та швидко забронювати номер в готелі з урахуванням їхніх потреб та побажань, а також дозволяє персоналу готелю ефективно керувати процесом бронювання та забезпечувати якість обслуговування гостей (рис 3.9).

Рисунок 3.9 – Екран "Резервація"

Зм..	Арк.	№докум.	Підпис	Дата

На екрані "Резервація" зображено роботу фільтра, який дозволяє підібрати номер за різними параметрами. Фільтр знаходиться з правого боку екрану та містить наступні параметри:

- 1) Кількість ліжок - дозволяє обрати номер з потрібною кількістю ліжок.
- 2) Кондиціонер - дозволяє обрати номер з наявністю кондиціонера.
- 3) Телевізор - дозволяє обрати номер з наявністю телевізора.
- 4) WI-FI - дозволяє обрати номер з наявністю WI-FI.

Ці параметри можуть бути використані окремо або в комбінації один з одним. Наприклад, користувач може обрати номер з одним ліжком, кондиціонером і WI-FI, або номер з двома ліжками і телевізором.

Після вибору необхідних параметрів, користувач може натиснути на кнопку "Знайти кімнати", щоб отримати список вільних номерів, які відповідають заданим критеріям. Якщо ж користувач вже обрав номер, він може просто ввести номер кімнати у відповідному полі і продовжити процес резервації (рис 3.10).

The screenshot shows a web application window titled "Інформаційна система готелю". The interface is divided into several sections. At the top, there is a navigation bar with icons for "Статус готелю", "Кімнати", "Резервація", "Платежі", "Контакти", and "Вихід". The main content area is split into two columns. The left column contains a reservation form with sections for "Персональна інформація" (Name, Surname, Title), "Контактна інформація" (Phone, Email, Guest Address), and "Інформація про бронювання" (Children, Adults, Stay Duration). The right column features a "Фільтр" (Filter) panel with dropdown menus for "Кількість ліжок" (1), "Кондиціонер" (Так), "Телевізор" (Так), and "Wi-Fi" (Так). Below the filter is a "Знайти кімнати" button and a list of room options: "Номер кімнати 1 - Ціна - 3000", "Номер кімнати 2 - Ціна - 3500", and "Номер кімнати 10 - Ціна - 3500".

Рисунок 3.10 – Екран "Резервація", робота фільтра

На екрані "Резервація" після того, як користувач ввів необхідну інформацію про своє резервування, з'являється вибір способу оплати. Користувач може обрати

один з чотирьох варіантів оплати: готівка, картка, Bitcoin або PayPal. Для вибору способу оплати необхідно натиснути на відповідну кнопку з інформацією про кожен варіант оплати. Після вибору способу оплати, користувач зможе підтвердити своє резервування та здійснити оплату, натиснувши кнопку "Оплатити". Якщо користувач обрав оплату готівкою, на екрані з'явиться повідомлення з інструкціями щодо сплати за резервування при прибутті на місце (рис 3.11).

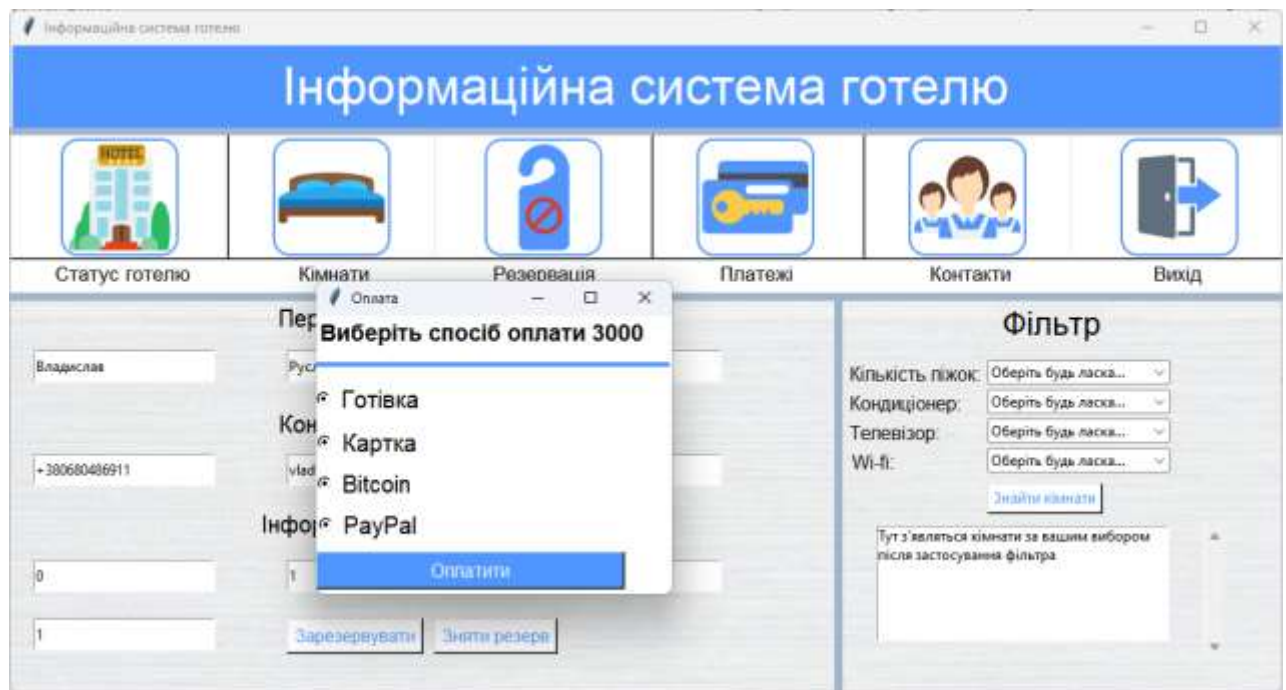


Рисунок 3.11 – Екран "Резервація", оплата

На екрані "Резервація" після того, як користувач успішно здійснив оплату та зарезервував номер, з'являється підтвердження про успішну резервацію. Підтвердження успішної резервації є важливою складовою користувацького досвіду, оскільки воно надає впевненість та забезпечує відчуття успіху після виконання операції (рис 3.12).

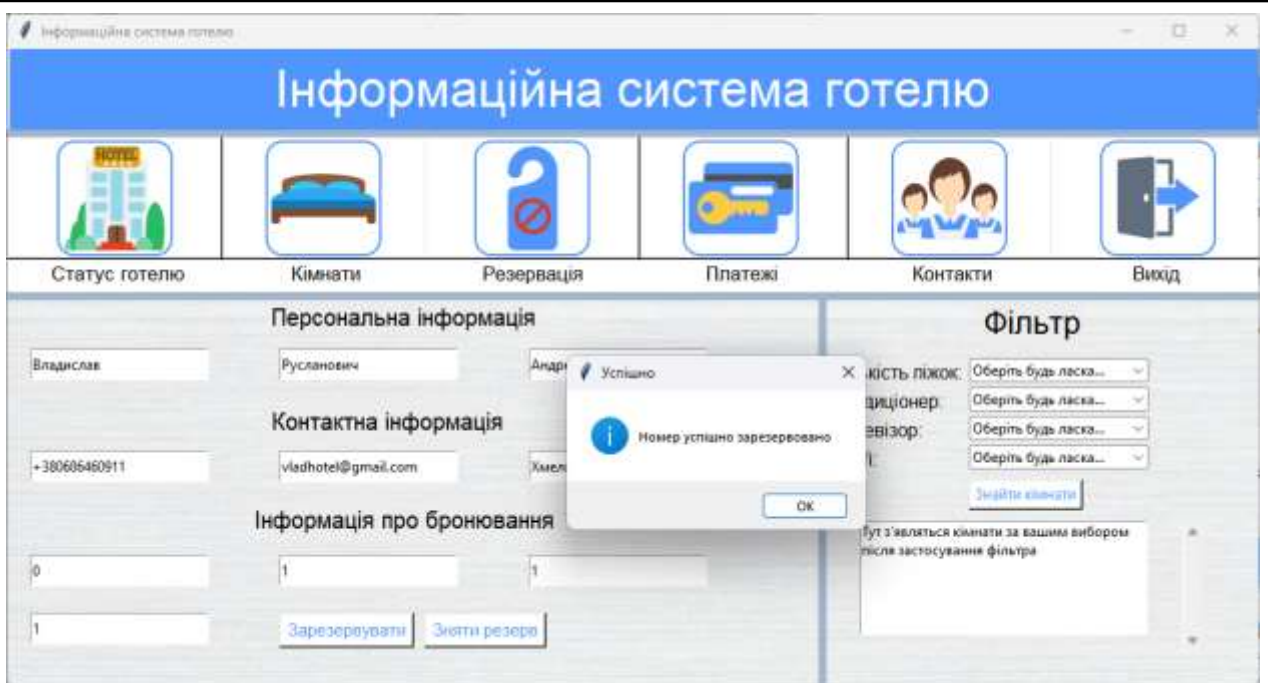


Рисунок 3.12 – Екран "Резервація", успішне зарезервування номеру

На екрані "Резервація" після успішної оплати та зарезервування номера з'являється опція роздрукувати чек. Це може бути важливо для користувачів, які бажають мати паперову копію документів про оплату та резервацію (рис 3.13).

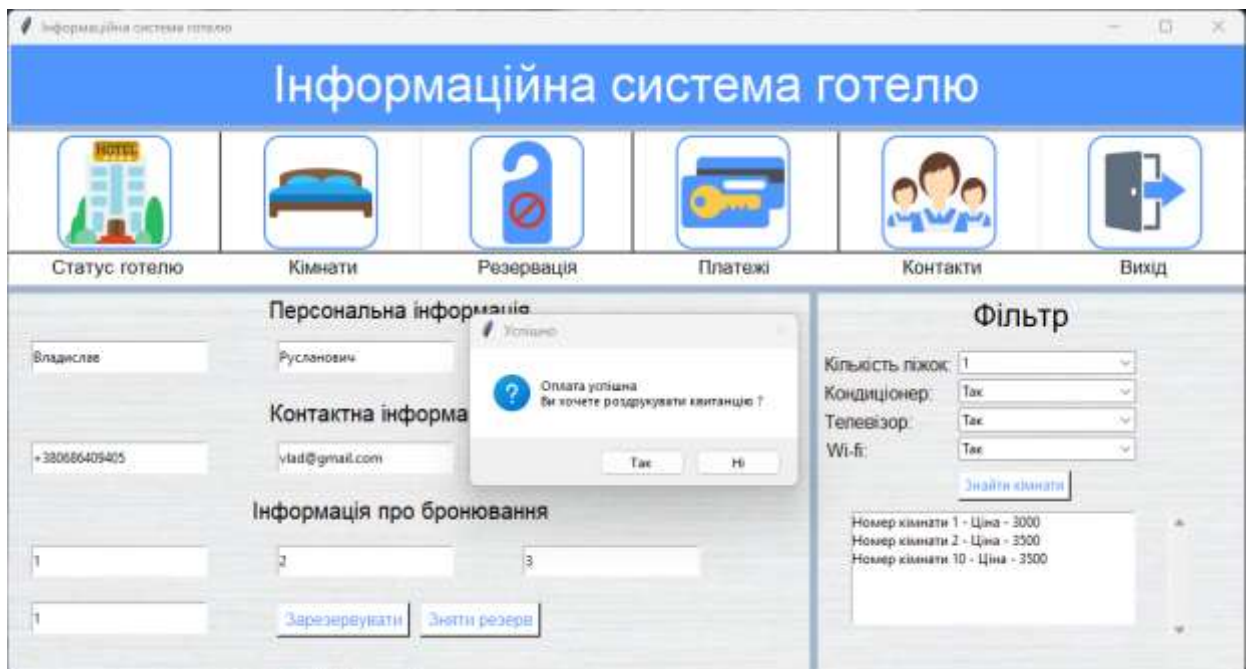


Рисунок 3.13 – Екран "Резервація", можливість роздрукувати чек після успішної оплати

На екрані "Платежі" можна перевірити деталі платежу, які були здійснені раніше від імені гостя. Для перевірки платежу, користувач повинен ввести унікальний ідентифікатор платежу, який був наданий під час оплати. Після цього на екрані будуть відображені: дата операції, час транзакції, сума оплати та спосіб оплати. Крім того, користувач може роздрукувати квитанцію про здійснення платежу, яка буде містити всі деталі операції.

Цей екран є корисним для гостей, які бажають перевірити, чи була проведена оплата за їхнє проживання, а також для готелів, які потребують підтвердження здійсненої оплати. Він дозволяє зручно керувати та відстежувати всіма платіжними операціями, пов'язаними з резерваціями та проживанням. Забезпечувати доступ до цієї інформації користувачам допомагає зменшити можливість незрозумінь та сприяє взаєморозумінню між гостями та готелями (рис 3.14).

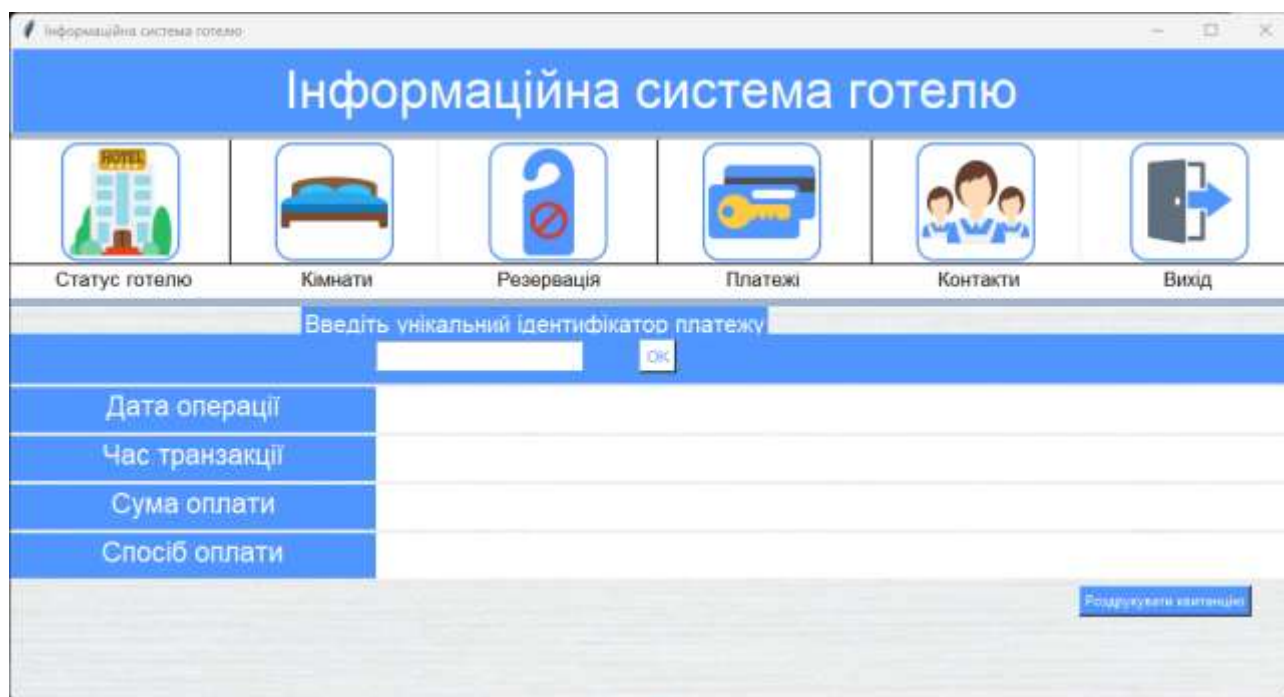


Рисунок 3.14 – Екран "Платежі"

На екрані "Платежі" після того, як користувач ввів унікальний ідентифікатор платежу, з'явилася інформаційна таблиця з деталями платежу. Вона містить наступні дані: дату операції, час транзакції, суму оплати та спосіб оплати (рис 3.15).

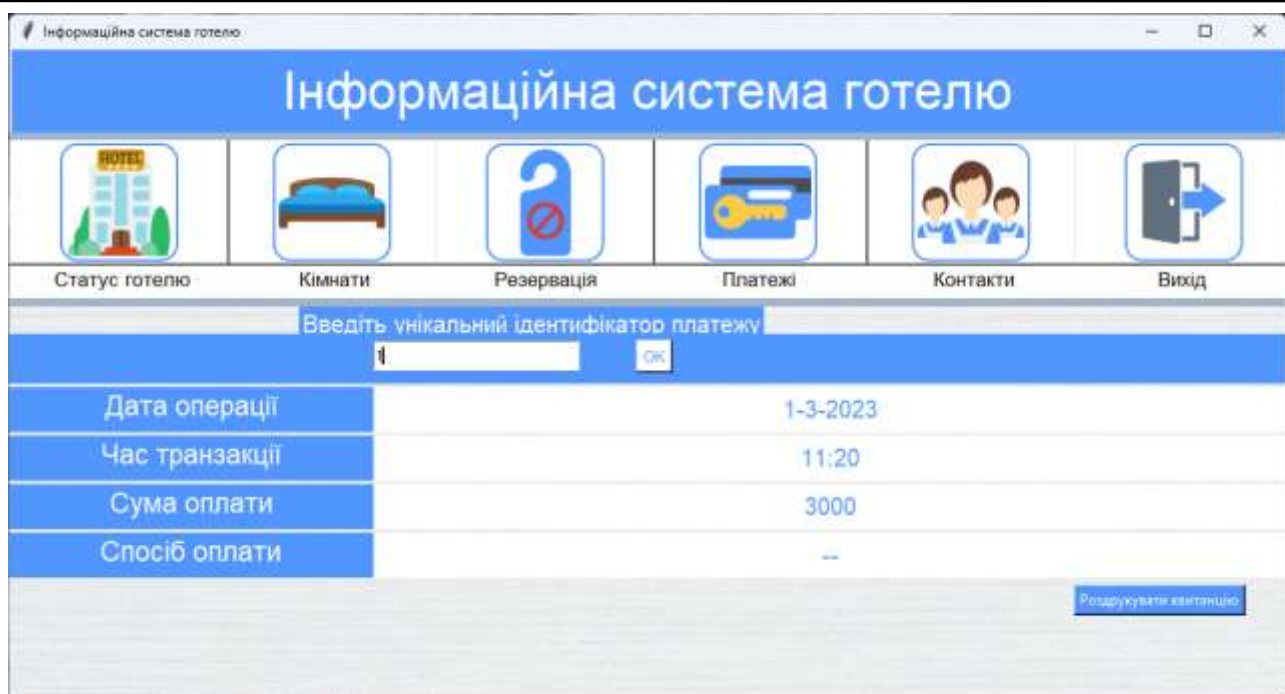


Рисунок 3.15 – Екран "Платежі", перевірка платежу

Екран "Контакти" містить інформацію про персонал готелю:

- 1) Менеджер: ім'я та прізвище менеджера, адреса електронної пошти та унікальний номер співробітника.
- 2) Ресторан: ім'я та прізвище шеф-кухаря, адреса електронної пошти та унікальний номер співробітника.
- 3) Сервіс підтримки: ім'я та прізвище представника служби підтримки, адреса електронної пошти та унікальний номер співробітника.
- 4) Прибирання: ім'я та прізвище головної прибиральниці, адреса електронної пошти та унікальний номер співробітника.

На екрані можуть бути представлені контактні дані кожного співробітника, що може бути корисним для гостей готелю, які мають питання або потребують допомоги. Наприклад, гость може звернутися до менеджера готелю, якщо виникли питання щодо бронювання номеру або оплати послуг. Або звернутися до представника служби підтримки, якщо потрібна технічна підтримка (рис 3.16).

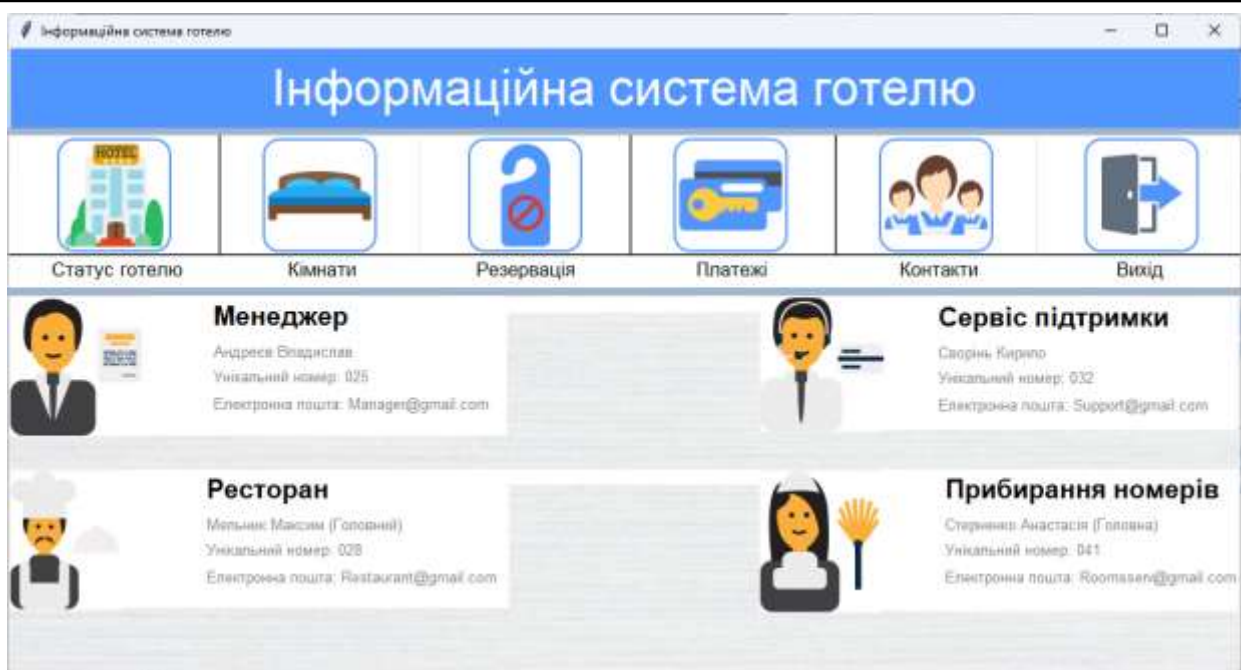


Рисунок 3.16 – Екран "Контакти"

Екран "Вихід" містить запит на підтвердження виходу з програми. На цьому екрані можна побачити запитання: "Ви впевнені, що хочете вийти?" і кнопки "Так" та "Ні". Користувач має можливість вибрати "Так", щоб підтвердити вихід, або "Ні", щоб залишитися в програмі (рис 3.17).

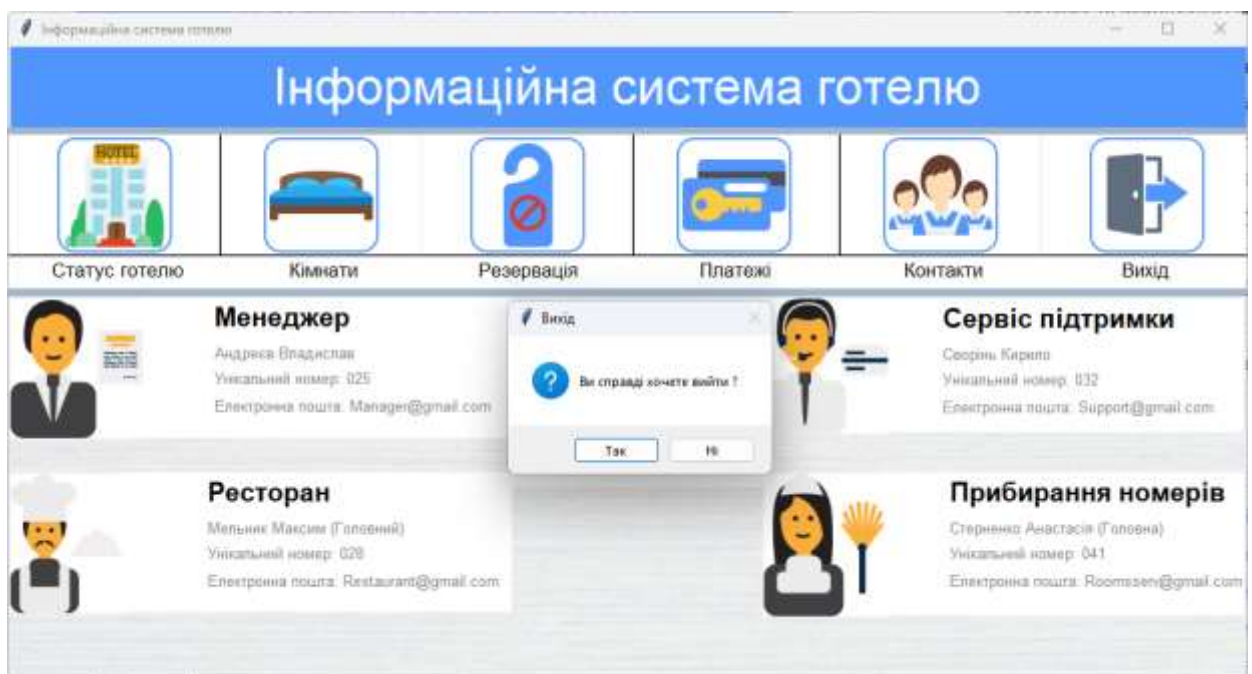


Рисунок 3.17 – Екран "Вихід"

Зм..	Арк.	№докум.	Підпис	Дата

3.7 Висновки

Розробка веб-орієнтованої інформаційної системи готелю є складним завданням, яке потребує вивчення і використання різних засобів і технологій. Під час обґрунтування вибору засобів розробки було розглянуто різні аспекти, такі як мови програмування, фреймворки, бази даних та інші. В результаті було обрано Python як основну мову програмування, фреймворк Django для розробки веб-додатків та базу даних SQLite для збереження інформації про готель.

Для успішної реалізації інформаційної системи готелю також потрібні вимоги до апаратного та програмного забезпечення. Апаратне забезпечення повинно відповідати вимогам розробленої системи, зокрема, мати достатній обсяг пам'яті та процесорну потужність. Щодо програмного забезпечення, то необхідно мати наявність певних програмних засобів, таких як веб-браузери, редактори коду тощо.

Під час розробки програмної реалізації інформаційної системи готелю було вирішено багато ключових завдань, зокрема, реалізація бронювання номерів, пошуку вільних номерів з вибраними зручностями, керування персоналом та інтерфейсу користувача. Всі ці модулі були успішно інтегровані в одну програму та виконують свої функції відповідно до поставлених вимог.

Отже, успішна розробка інформаційної системи готелю потребує комплексного підходу та використання різних технологій та засобів. Дотримання вимог до апаратного і програмного забезпечення та реалізація необхідного функціоналу дозволяють створити ефективну та зручну для користувачів інформаційну систему готелю, яка допоможе підвищити рівень обслуговування гостей та забезпечить ефективне управління готелем. Вибір відповідних засобів розробки та технологій, а також комплексний підхід до розробки системи є ключовими факторами для успішного впровадження інформаційної системи готелю

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
						66
Зм..	Арк.	№докум.	Підпис	Дата		

ВИСНОВКИ

У сучасному світі веб-орієнтовані інформаційні системи грають важливу роль у готельному бізнесі, допомагаючи автоматизувати процеси, підвищувати ефективність управління та покращувати обслуговування гостей. Використання популярних інформаційних систем, таких як Property Management System (PMS), Central Reservation System (CRS), Revenue Management System (RMS), Customer Relationship Management (CRM), Point of Sale (POS) та Business Intelligence (BI), дозволяє готелям досягти позитивних результатів.

Однак, розробка власної веб-орієнтованої інформаційної системи може бути важливою стратегічною перевагою для готелю. Це дозволяє створити систему, яка повністю відповідає специфічним потребам та вимогам готелю. Врахування функціональних, безпекових та інтеграційних аспектів, а також SMART-критеріїв при формулюванні вимог допомагає побудувати оптимальну інформаційну систему для готелю.

Успішна розробка веб-орієнтованої інформаційної системи готелю вимагає комплексного підходу. Необхідно враховувати вимоги користувачів, бізнесу та ринку, розробити оптимальну архітектуру, забезпечити тестування та якість продукту. Дотримання принципів SOLID та Clean Architecture сприяє створенню простої, масштабованої та підтримуваної системи. Залучення фахівців з різних областей допомагає забезпечити успішне проектування програмно-технічного засобу.

Розробка веб-орієнтованої інформаційної системи готелю включає вибір засобів розробки, таких як мови програмування, фреймворки та бази даних. У даному випадку, Python, фреймворк tkinter та база даних SQLite були обрані для розробки системи. Python є однією з популярних мов програмування, яка використовується для розробки веб-додатків. Вона має чистий синтаксис, що сприяє зрозумілості коду і полегшує розробку. Python також володіє великою кількістю бібліотек і фреймворків, які допомагають прискорити процес розробки.

Ще одним важливим аспектом розробки веб-орієнтованої інформаційної системи готелю є питання безпеки. З огляду на чутливу природу даних, таких як

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм..	Арк.	№докум.	Підпис	Дата		67

персональна інформація гостей та фінансові дані, необхідно забезпечити надійний рівень захисту. Це включає в себе застосування шифрування даних, механізми аутентифікації та авторизації, а також заходи проти зловживання системою. При розробці системи слід дотримуватись найкращих практик безпеки і використовувати актуальні методи та технології для забезпечення надійності системи.

Крім того, після успішної розробки і впровадження веб-орієнтованої інформаційної системи готелю необхідно забезпечити постійну підтримку та обслуговування. Це включає в себе виявлення і усунення помилок, оновлення та покращення функціональності, а також надання технічної підтримки користувачам системи. Постійна підтримка допоможе забезпечити безперебійну роботу системи та задоволення потреб користувачів.

Узагальнюючи, розробка веб-орієнтованої інформаційної системи готелю є складним процесом, який вимагає уваги до деталей, комплексного підходу та використання різноманітних засобів та технологій. Вибір відповідних інформаційних систем, формулювання вимог, розробка архітектури, забезпечення безпеки та підтримка - всі ці аспекти мають важливе значення для успішного впровадження системи та підвищення ефективності готельного бізнесу.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		68

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Боброва Н.Ю. Комплексна інформаційна система для готелю та актуальні проблеми економіки: конспект лекцій. Київ: ЦУЛ, 2018. С. 196-204.
2. Дуб О.М. Система автоматизованого бронювання готельних послуг на платформі Lagavel. *Сучасні проблеми математики, фізики та інформатики*: матеріали доповідей XV Всеукраїнської науково-практичної конференції. Київ, 2018. С. 75-77.
3. Коломієць А.О. Автоматизована інформаційна система готелю. *Комп'ютерні науки*. Том 1. Київ, 2018. С. 78-85.
4. Мартиненко І.С. Підприємець Ю.Ю. Використання системи бронювання готельних послуг на основі технології РНР. *Інформаційні технології в науці, освіті, техніці та економіці*. Київ, 2018. С. 103-105.
5. Роздол І.Ю. Комплексна інформаційна система готелю на базі CRM. *Східноєвропейський журнал передових технологій*. Київ, 2018. С. 17-20.
6. Чорна Ю.В. Розробка автоматизованої інформаційної системи для готелю. *Науковий світ*. № 12 (64). Київ, 2019. С. 12-15.
7. Корнієнко О. Розвиток готельного бізнесу в Україні: тенденції та перспективи. *Маркетинг і менеджмент інновацій*. Київ, 2018. С. 8-16.
8. Ніконов В.В. Інформаційні технології в готельному бізнесі. *Економіка. Фінанси. Право*. Київ, 2018. С. 98-100.
9. Підгурський В. Інформаційні системи в готельному бізнесі. *Готельний бізнес*. Київ, 2019. С. 28-30.
10. Садовничий В. Інформаційні технології в готельному бізнесі. *Туризм: теорія і практика*. Київ, 2019. С. 63-67.
11. Стаднік А. Особливості впровадження інформаційних технологій у готельному бізнесі. *Вісник ЖДТУ*. Київ, 2018. С. 191-195.
12. Шрамко А. Інформаційні технології в готельному бізнесі. *Інноваційна економіка*. Київ, 2018. С. 124-129.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		69

13. Altamimi A., Aldakhil A. M., Al-Kwifi O. S., Alqarni A. M. Factors affecting the adoption of hotel information systems in Saudi Arabia. *Journal of Hospitality and Tourism Technology*. 2020. Vol. 11, No 2. P. 167-183.
14. Anohah A., Wali N. A. An empirical investigation of hotel management information systems and hotel performance in Malaysia. *Journal of Hospitality and Tourism Technology*. 2019. Vol. 10, No 2. P. 94-106.
15. Chen Y. C., Wang D., Xie H. K. An intelligent decision support system for hotel room pricing. *Journal of Hospitality and Tourism Technology*. Vol. 10, No 2. P. 52-64.
16. Deng J., Hu X., Zhang Y. The impact of hotel management information systems on hotel service delivery in China. *Journal of Hospitality and Tourism Technology*. 2018. Vol. 9, No 2. P. 151-167.
17. Juwaheer T. D. The use of information technology in the hotel industry: a case study of Hilton hotel corporation. *Journal of Tourism, Hospitality and Sports*. 2018. Vol. 16. P. 1-10.
18. Kim D., Lee H. G. Effect of hotel information systems on hotel operational efficiency: Focused on hotel central reservation system. *International Journal of Information Management*. 2018. Vol. 38, No 1. P. 157-166.
19. Kwok C. K., Yip L. S., Ngai E. W. A conceptual model of factors affecting the adoption of hotel information systems in the Hong Kong hotel industry. *Journal of Hospitality and Tourism Technology*. 2018. Vol. 8, No 1. P. 91-105.
20. Li X., Lu L., Su C. Information technology capability and hotel performance: Evidence from the Chinese hotel industry. *Journal of Hospitality and Tourism Technology*. 2018. Vol. 9, No 3. P. 316-331.
21. Ma Y., Zhang Y. Impact of information technology capabilities on hotel performance: An empirical study. *Journal of Hospitality and Tourism Technology*. 2018. Vol. 8, No 3. P. 370-384.
22. Yoo D. K., & Lee J. Effects of hotel information systems on hotel performance: Focusing on hotel size and hotel chain affiliation. *International Journal of Information Management*. 2018. Vol. 37, No 6. P. 631-642.

23. The Ultimate Guide to Hotel Management Software URL: <https://hoteltechreport.com/guides/hotel-management-software>. (дата звернення: 12.01.2023)

24. Channel Manager for Hotels. URL: <https://www.siteminder.com/platform/channel-manager/> (дата звернення: 12.01.2023)

25. Kaur H., & Singh K. D. A review of hotel revenue management research: What is the next step?. *Journal of Hospitality and Tourism Management*. 2020. Vol. 33. P. 152-161.

26. Laudon K. C., & Laudon, J. P. Management Information Systems: Managing the Digital Firm. Pearson Education Limited. 2021. Vol. 3. P. 1-10.

27. Чернявський В.І., Жуковський, О.М., Герман, І.І. Архітектура інформаційно-технічної інфраструктури підприємства. *Вісник Харківського національного технічного університету сільського господарства імені Петра Василенка*. Київ, 2018. С. 147-154.

28. Кропивницький В. Архітектура ІТ-інфраструктури як інструмент ефективного управління інформаційними ресурсами підприємства. *Наукові записки Національного університету «Острозька академія», серія «Економіка»*. Київ, 2018. С. 88-92.

29. Кунанець Н. Роль архітектури ІТ-інфраструктури у розвитку бізнесу. *Маркетинг і менеджмент інновацій*. Львів, 2018. С. 80-86.

30. Information Technology Infrastructure Library. URL: <https://www.axelos.com/best-practice-solutions/itil> (дата звернення: 13.01.2023)

31. Control Objectives for Information and Related Technology (COBIT). URL: <https://www.isaca.org/resources/cobit> (дата звернення: 13.01.2023)

32. Hawryszkiewicz I. T. The art of enterprise information architecture. A *systems-based approach for unlocking business insight*. 2018. Vol. 8, No 5. P. 61-68.

33. Bonneau J. Cybersecurity and infrastructure protection: Background, current issues, and options for Congress. Congressional Research Service. 2019. Vol. 26, No 3. P. 250-275.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		71

34. Теплякова О.В., Хоменко І.І. "Технології управління готелями та ресторанами: збірник матеріалів IV Міжнародної науково-практичної конференції". Київ: 2018. 320 с.

35. Sivakumar S. та Arjunan R. Нефункціональні вимоги до веб-базової інформаційної системи управління готелями. *Міжнародний журнал комп'ютерних додатків*. 2018. С. 6-11.

36. Ali N. M., Khan A. H. та Khan S. Проектування системи управління готелем: підхід на основі вивчення випадку. *Міжнародний журнал комп'ютерних наук та мережевої безпеки*. 2018. С. 81-91.

37. Aziz N. A. та Mohd Zaki N. N. Аналіз вимог та проектування системи управління готелем. *Міжнародний журнал інженерії та технології*. 2018. С. 634-640.

38. Wallace L. G. та Hendricks, L. G. Каркас нефункціональних вимог до інформаційних систем. *Журнал управління інформаційними технологіями*. 2019. С. 16-30.

39. Mahdavi-Hezavehi M., Abrishami, A. та Jalali, M. (2016). Інженерія нефункціональних вимог: систематичний огляд. *Журнал систем і програмного забезпечення*. 2018. С. 273-293.

40. Web-Based Hotel Reservation System. URL: https://www.academia.edu/9645615/Web-Based_Hotel_Reservation_System (дата звернення: 13.01.2023)

41. Офіційна документація Python. URL: <https://docs.python.org/> (дата звернення: 18.01.2023)

42. Офіційна документація Django. URL: <https://docs.djangoproject.com/> (дата звернення: 18.01.2023)

43. Офіційна документація Flask. URL: <https://flask.palletsprojects.com/> (дата звернення: 18.01.2023)

44. Офіційна документація CherryPy. URL: <https://docs.cherrypy.org/> (дата звернення: 18.01.2023)

45. Офіційна документація Tornado. URL: <https://www.tornadoweb.org/en/stable/> (дата звернення: 18.01.2023)

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		72

46. Офіційна документація Pyramid. URL: <https://trypyramid.com/> (дата звернення: 18.01.2023)
47. Laudon K. C., & Laudon, J. P. Management information systems: managing the digital firm. *Pearson Education*. 2018. Vol. 8. P. 2-10 .
48. Sigala M. Exploiting smart tourism innovation ecosystem for destination competitiveness: the case of Visit Greece digital journey. *Information Technology & Tourism*. 2018. Vol. 16, No 3. P. 233-262.
49. Бутенко С.В., Верес Ю.В., Гаврилова Т.В. Моделювання та проектування інформаційних систем. *Видавничий дім "Ліра"*. 2016. С. 48-53.
50. Мішкевич І.В. "Інформаційні системи та технології." *Центр учбової літератури*. 2018. С. 34-42.
51. Сайт компанії Oracle. URL: <https://www.oracle.com/ua/database/what-is-database.html> (дата звернення: 18.01.2023)
52. Zhang Z., Law R., Qi S. Examining the effects of online reviews on travelers' decision-making: A comparison of Chinese and Western travelers. *Journal of Travel Research*. 2019. Vol. 39, No 1. P. 188-202.
53. Gretzel U., Sigala M., Xiang Z., Koo C. Smart tourism: foundations and developments (2nd edition). 2019. Vol. 25, No 3. P. 179-188.
54. Xiang Z., Du Q., Fan W., Wang D. Social media analytics and big data in tourism: Current status and future directions. *Journal of Travel Research*. 2020. Vol. 58. P. 51-65.
55. Sigala M., Robinson R. N. Sharing economy and the future of tourism. *Routledge*. 2018. Vol. 29, No 9. P. 279-297.
56. Li X. R., Wang D. A study on cloud computing-based hotel information management system. *Journal of Physics: Conference Series*. 2018. Vol. 29. P. 25-29.
57. O'Connor P., Frew A. J., Morrison A. M. Revenue management and pricing analytics in the sharing economy: Research themes and future directions. *Journal of Revenue and Pricing Management*. 2018. Vol. 15, No 1. P. 270-283.
58. Vrana V., Mokoš M., Mokoš M. Mobile applications in hospitality and tourism—A review of 10 years of research. *Current Issues in Tourism*. 2019. Vol. 22, No 9. P. 1039-1059.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		73

59. Gupta A., Kumar R. Hotel management systems: A study of usage in medium-sized hotels. *Journal of hospitality and tourism management*. 2018. Vol. 31. P. 99-110.

60. Kim H., Lee J., Choi B. The impact of hotel information technology on service quality and business performance: *A case study of hotels in Korea*. 2018. Vol. 10, No 5. P. 1365 p.

61. Xu F., Buhalis D., Weber J. Serious games and the gamification of tourism. *Tourism Management*. Guangzhou: 2019. Vol. 60. P. 244-256.

62. Abdou A. A., Hafeez K. The impact of big data analytics on hotel guest loyalty: The mediating role of hotel guest satisfaction. *Journal of Travel Research*. 2018. Vol. 57, No 8. P. 1093-1108.

63. Prayag G., Hosany S., Odeh K. The role of tourists' emotional experiences and satisfaction in understanding behavioral intentions. *Journal of Destination Marketing & Management*. 2019. Vol. 6, No 3. P. 183-191.

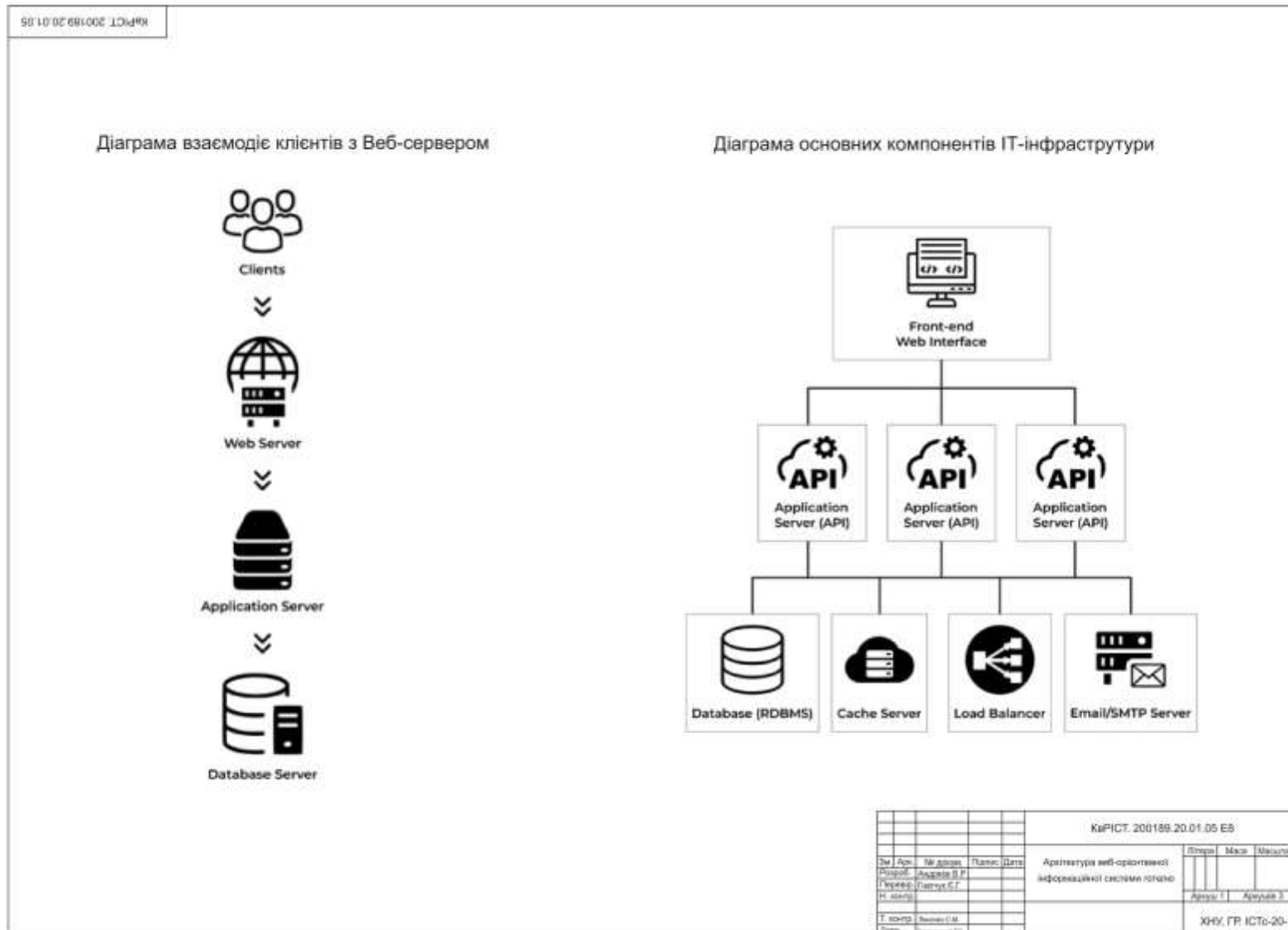
64. Ramanathan U., Ramanathan R. The impact of e-commerce on the small size hotels in the UK. *International Journal of Hospitality Management*. 2020. Vol. 46. P. 177-188.

					КВРІСТ.200189.20.01.05 ПЗ	Арк.
Зм.	Арк.	№докум.	Підпис	Дата		74

ДОДАТОК А

(обов'язковий)

Копія креслення «Архітектура веб-орієнтованої інформаційної системи готелю»



ДОДАТОК В

(обов'язковий)

Копія креслення «Інтерфейсні вікна інформаційної системи»

50 10'02' 691'00С ДЗНБ®

Інформаційна система готелю

Статус готелю | Кімнати | Резервації | Платежі | Контакти | Вихід

Персональна інформація

Ім'я* | По батьку | Прізвище*

Контактна інформація

Мобільний телефон* | Електронна пошта* | Адреса електронної пошти*

Інформація про бронювання

Кількість днів* | Кількість дорослих* | Кількість дітей до трьох років*

Фільтр

Кількість тижнів | Станьте будь-яким...
 Кондиціонер | Станьте будь-яким...
 Телевізор | Станьте будь-яким...
 Wi-Fi | Станьте будь-яким...
 Кнопка фільтрації

Тут ви можете вибрати чи відключити певні параметри фільтрації

Кімната 1

Кімната	Кількість місць	Кондиціонер?	Телевізор?	Wi-Fi?	Ціна?	Бронювання?
1	1	Так	Так	Так	3000	Ні
Кімната 2						
Кімната 3						
Кімната 4						
Кімната 5						
Кімната 6						
Кімната 7						
Кімната 8						
Кімната 9						
Кімната 10						
Кімната 11						
Кімната 12						

Інформаційна система готелю

Статус готелю | Кімнати | Резервації | Платежі | Контакти | Вихід

Персональна інформація

Ім'я* | По батьку | Прізвище*

Контактна інформація

Мобільний телефон* | Електронна пошта* | Адреса електронної пошти*

Інформація про бронювання

Кількість днів* | Кількість дорослих* | Кількість дітей до трьох років*

Фільтр

Кількість тижнів | Станьте будь-яким...
 Кондиціонер | Станьте будь-яким...
 Телевізор | Станьте будь-яким...
 Wi-Fi | Станьте будь-яким...
 Кнопка фільтрації

Тут ви можете вибрати чи відключити певні параметри фільтрації

Введіть унікальний ідентифікатор платежу

Дата операції
 Час транзакції
 Сума оплати
 Спосіб оплати

Кнопка фільтрації

КаРІСТ. 200189.20.01.05.Е8						Відео	Мова	Масштаб
Ім'я	Адреса	№ докум.	Статус	Дата				
Розробник	Курдюк В.Р.							
Програмувальник	Григорук С.Г.							
№ версії						Версія 1	Версія 2	
Сторінка	Всього стор.					ХНУ, ГР ICT-20-1		
Дата	Надруковано							

ДОДАТОК Г

«Лістинг програми запуску головного вікна розробленого програмного забезпечення»

```
#code written by VladAnreiev

from tkinter import *
from tkinter import ttk
import time
import datetime
from PIL import ImageTk,Image
import os
import sqlite3
from tkinter import messagebox
now = datetime.datetime.now()
#----- importing sqlite for server side operations-----
con = sqlite3.Connection('hm_proj.db')
cur = con.cursor()
cur.execute("create table if not exists hoteld(t_r number,r_r number,t_s number)")
cur.execute("create table if not exists roomd(rn number primary key,beds number,ac varchar(10),tv varchar(10),internet
varchar(10),price number(10))")

#pmethod=0

#rstatus extra column
#for i in range (1,21):
#cur.execute("update roomd set tv='Yes' where rn = ? ",(19,))
cur.execute("create table if not exists payments(id number primary key,dot varchar(15),time varchar(10),amt
number,method varchar(10))")
cur.execute("create table if not exists paymentsf(id number primary key,f_name varchar,l_name varchar,c_number
varchar,email varchar , r_n number ,day varchar,month varchar,year varchar,time varchar , method varchar,totalamt
varchar)")
#cur.execute("insert into paymentsf
values(1,'Andreiev','Vladyslav','0686463800','vlad@gmail.com',2,'1','11','2023','11:20:27 PM','Cash','3500)")
#cur.execute("alter table paymentsf add totalamt varchar")
con.commit()
#cur.execute("drop table paymentsf")
#cur.execute("insert into hoteld values(20,11,30)")
con.commit()
cur.execute("select * from payments")
con.commit()
x=cur.fetchall()
con.commit()
#print(x)
#-----splash_screen-----
sroot = Tk()
sroot.minsize(height=422,width=800)
sroot.configure(bg='white')
spath = "images/mypic.jpg"
simg = ImageTk.PhotoImage(Image.open(spath))
my = Label(sroot,image=simg)
my.image = simg
my.place(x=0,y=0)

#----- main project-----
def mainroot():
    #sroot.destroy()
    root = Tk()
    root.geometry('1080x500')
```

```

root.minsize(width=1080,height=550)
root.maxsize(width=1080,height=550)
root.configure(bg='white')
root.title("Інформаційна система готелю")
#-----seperator-----

sep = Frame(height=500,bd=1,relief='sunken',bg='white')
#sep.place(x=20,y=0)
#-----Connection with printer-----

def connectprinter():
    os.startfile("C:/Users/TestFile.txt", "print")
#-----top frame-----

top_frame = Frame(root,height=70,width=1080,bg='orange')
path = "images/newestbg5.jpg"
img = ImageTk.PhotoImage(Image.open(path))
label = Label(top_frame,image = img ,height=70,width=1080)
label.image=img
label.place(x=0,y=0)
top_frame.place(x=0,y=0)
tf_label = Label(top_frame,text='Інформаційна система готелю',font='msserif
33',fg='white',bg='#5196FF',height=70)
tf_label.pack(anchor='center')
top_frame.pack_propagate(False)

#-----DATE TIME-----
def datetime():
    #while(True):
        localtime = now.strftime("%Y-%m-%d %H:%M")
        lblInfo = Label(top_frame,font='helvetica 15',text=localtime,bg='blue',fg='white')
        #lblInfo.place(x=333,y=40)

#bottom frame - hotel status and default page---
def hotel_status():
    global b_frame
    b_frame = Frame(root,height=400,width=1080,bg='gray91')
    b_frame.place(x=0,y=120+6+20+60+11)
    b_frame.pack_propagate(False)
    path = "images/newbg6lf.jpg"
    img = ImageTk.PhotoImage(Image.open(path))
    label = Label(b_frame,image = img ,height=400,width=1080)
    label.image=img
    label.place(x=0,y=0)
    cur.execute("select * from hoteld")
    x = cur.fetchall()
    #print(x)
    cur.execute("select count(rn) from roomd")
    x = cur.fetchone()
    print (x)
    cur.execute("select count(rn) from roomd where rstatus = 'Reserved'")
    y = cur.fetchone()
    print (y)
    tor = x[0]
    rer = y[0]
    tos = 21
    avr = int(tor)-int(rer)
    avr = str(avr)
    #print(tor,rer,tos,avr)
    hts = Label(b_frame,text='Статус готелю',font='msserif 15',fg='black',bg='gray91',height=1)
#-----inner frames of bottom frame-----

```

```

smf1 = Frame(b_frame,height=150,width=175,bg='white')
tr = Label(smf1,text='Усі кімнати:',fg='white',bg='#5196ff',width=100,height=2,font='helvetica 15')
tr.pack(side='top')
smf1.pack_propagate(False)
smf1.place(x=0,y=30)
Label(smf1,text=tor,fg='#5196ff',bg='white',font='msserif 50').pack(anchor='center')

smf2 = Frame(b_frame,height=150,width=175,bg='white')
ar = Label(smf2,text='Доступні кімнати:',fg='white',bg='#5196ff',width=130,height=2,font='helvetica
15')
ar.pack(side='top')
smf2.pack_propagate(False)
smf2.place(x=180+4,y=30)
Label(smf2,text=avr,fg='#5196ff',bg='white',font='msserif 50').pack(anchor='center')

smf3 = Frame(b_frame,height=150,width=175,bg='white')
tre = Label(smf3,text='Зарезервовано:',fg='white',bg='#5196ff',width=130,height=2,font='helvetica
15')
tre.pack(side='top')
smf3.pack_propagate(False)
smf3.place(x=360+6,y=30)
Label(smf3,text=rer,fg='#5196ff',bg='white',font='msserif 50').pack(anchor='center')

smf4 = Frame(b_frame,height=150,width=175,bg='white')
tc = Label(smf4,text='Кількість гостей:',fg='white',bg='#5196ff',width=130,height=2,font='helvetica
15')
tc.pack(side='top')
smf4.pack_propagate(False)
smf4.place(x=540+8,y=30)
Label(smf4,text='40',fg='#5196ff',bg='white',font='msserif 50').pack(anchor='center')

smf5 = Frame(b_frame,height=150,width=175,bg='white')
ts = Label(smf5,text='Персонал:',fg='white',bg='#5196ff',width=130,height=2,font='helvetica 15')
ts.pack(side='top')
smf5.pack_propagate(False)
smf5.place(x=720+10,y=30)
Label(smf5,text=tos,fg='#5196ff',bg='white',font='msserif 50').pack(anchor='center')
redf1 = Frame(b_frame,height=8,width=1080,bg='#5196ff')

smf6 = Frame(b_frame,height=150,width=175,bg='white')
ts = Label(smf6,text='На ремонті:',fg='white',bg='#5196ff',width=130,height=2,font='helvetica 15')
ts.pack(side='top')
smf6.pack_propagate(False)
smf6.place(x=915,y=30)
Label(smf6,text='3',fg='#5196ff',bg='white',font='msserif 50').place(x=60,y=60)
#redf1 = Frame(b_frame,height=8,width=1080,bg='#5196ff')

#Label(b_frame,text='=====',fg='#5196ff').place(x=0,y=20)
redf1.place(x=0,y=22)
Label(b_frame,text='Статус готелю',font='msserif 12',bg='#5196ff',fg='white').pack(anchor='center')
redf1.pack_propagate(False)
#b_frame.pack_propagate(False)

#----- Guests -----
def staff():
    b_frame = Frame(root,height=400,width=1080,bg='white')
    path = "images/newbg6lf.jpg"
    img = ImageTk.PhotoImage(Image.open(path))
    label = Label(b_frame,image = img ,height=400,width=1080)
    label.image=img
    label.place(x=0,y=0)
    l = Label(b_frame,text='Details of Staff will be Available soon')

```

```

#l.place(x=180,y=0)
""smf4 = Frame(b_frame,height=150,width=175,bg='white')
tc = Label(smf4,text='Total Customers:',fg='white',bg='#5196ff',width=130,height=2,font='helvetica
15')

tc.pack(side='top')
smf4.pack_propagate(False)
smf4.place(x=540+8,y=30)
Label(smf4,text='40',fg='#5196ff',bg='white',font='msserif 50').pack(anchor='center')
""

emp1f = Frame(b_frame)
path1 = "images/newman.jpg"
img1 = ImageTk.PhotoImage(Image.open(path1))
emp1 = Label(emp1f,image = img1)
emp1.image=img1
emp1.pack()
emp1f.place(x=0,y=0)
emp1inf = Frame(b_frame,bg='White',height=122,width=320)
Label(emp1inf,text="Менеджер",bg='white',font='msserif 17 bold').place(x=60,y=0)
Label(emp1inf,text="Андрєєв Владислав",bg='white',fg="Grey",font='msserif 10').place(x=60,y=37)
Label(emp1inf,text="Унікальний номер: 025",bg='white',fg="Grey",font='msserif
10').place(x=60,y=59)
Label(emp1inf,text="Електронна пошта: Manager@gmail.com",bg='white',fg="Grey",font='msserif
10').place(x=60,y=83)
emp1inf.pack_propagate(False)
emp1inf.place(x=117,y=1)

emp1f = Frame(b_frame)
path2 = "images/receptionnew.jpg"
img2 = ImageTk.PhotoImage(Image.open(path2))
emp1 = Label(emp1f,image = img2)
emp1.image=img2
emp1.pack()
emp1f.place(x=657,y=0)
emp1inf = Frame(b_frame,bg='White',height=116,width=310)
Label(emp1inf,text="Сервіс підтримки",bg='white',font='msserif 17
bold').place(x=45,y=0)#pack(side='top')
Label(emp1inf,text="Своринь Кирило",bg='white',fg="Grey",font='msserif 10').place(x=45,y=37)
Label(emp1inf,text="Унікальний номер: 032",bg='white',fg="Grey",font='msserif
10').place(x=45,y=59)
Label(emp1inf,text="Електронна пошта: Support@gmail.com",bg='white',fg="Grey",font='msserif
10').place(x=45,y=83)
emp1inf.pack_propagate(False)
emp1inf.place(x=767,y=2)

emp1f = Frame(b_frame)
path3 = "images/fchefnew.jpg"
img3 = ImageTk.PhotoImage(Image.open(path3))
emp1 = Label(emp1f,image = img3)
emp1.image=img3
emp1.pack()
emp1f.place(x=0,y=152)
emp1inf = Frame(b_frame,bg='White',height=121,width=340)
Label(emp1inf,text="Ресторан",bg='white',font='msserif 17 bold').place(x=72,y=0)#pack(side='top')
Label(emp1inf,text="Мельник Максим (Головний)",bg='white',fg="Grey",font='msserif
10').place(x=72,y=37)
Label(emp1inf,text="Унікальний номер: 028",bg='white',fg="Grey",font='msserif
10').place(x=72,y=59)
Label(emp1inf,text="Електронна пошта:
Restaurant@gmail.com",bg='white',fg="Grey",font='msserif 10').place(x=72,y=83)
emp1inf.pack_propagate(False)
emp1inf.place(x=99,y=153)
emp1inf.tkraise()

```

```

emp1f = Frame(b_frame)
path4 = "images/roomservicew.jpg"
img4 = ImageTk.PhotoImage(Image.open(path4))
emp1 = Label(emp1f,image = img4)
emp1.image=img4
emp1.pack()
emp1f.place(x=657,y=152)
emp1inf = Frame(b_frame,bg='White',height=124,width=315)
Label(emp1inf,text="Прибирання номерів",bg='white',font='msserif 17
bold').place(x=55,y=0)#pack(side='top')
Label(emp1inf,text="Стерненко Анастасія (Головна)",bg='white',fg="Grey",font='msserif
10').place(x=55,y=37)
Label(emp1inf,text="Унікальний номер: 041",bg='white',fg="Grey",font='msserif
10').place(x=55,y=59)
Label(emp1inf,text="Електронна пошта:
Roomsserv@gmail.com",bg='white',fg="Grey",font='msserif 10').place(x=55,y=83)
emp1inf.pack_propagate(False)
emp1inf.place(x=763,y=153)
#b_frame.pack_propagate(False)
Frame(b_frame,height=13,width=250,bg='white').place(x=410,y=2)
Frame(b_frame,height=13,width=250,bg='white').place(x=410,y=153)
#Frame(b_frame,height=180,width=13,bg='white').place(x=406,y=20)

b_frame.place(x=0,y=120+6+20+60+11)
b_frame.pack_propagate(False)
b_frame.tkraise()

#----- rooms -----
def rooms():
    b_frame = Frame(root,height=400,width=1080,bg='gray91')
    b_frame.place(x=0,y=120+6+20+60+11)
    b_frame.pack_propagate(False)
    b_frame.tkraise()
    path = "images/newbg6lf.jpg"
    img = ImageTk.PhotoImage(Image.open(path))
    label = Label(b_frame,image = img ,height=400,width=1080)
    label.image=img
    label.place(x=0,y=0)
    sidebuttons = Text(b_frame,width=1,height=19)
    sc = Scrollbar(b_frame,command=sidebuttons.yview,width=10,bg='lightsteelblue3')
    sidebuttons.configure(yscrollcommand=sc.set)
    sc.pack(side='left',fill=Y)
    sidebuttons.place(x=10,y=0)
    def roomdet(rno):
        Label(b_frame,text='Кімната %s'% rno,font='msserif
15',fg='white',bg='#5196ff',width=10).place(x=535,y=0)
        cur.execute("select * from roomd where rn = ?",(rno,))
        rdata=cur.fetchall()
        #print (rdata)
        smf1 = Frame(b_frame,height=120,width=145,bg='white')
        hline = Frame(b_frame,height=10,width=960,bg='#5196ff')
        hline.place(x=122,y=27)
        vline = Frame(b_frame,height=400,width=7,bg='lightsteelblue3')
        vline.place(x=122,y=0)
        tr = Label(smf1,text='Кількість
ліжок:',fg='white',bg='#5196ff',width=100,height=2,font='msserif 15')
        tr.pack(side='top')
        smf1.pack_propagate(False)
        smf1.place(x=129+3,y=30)
        Label(smf1,text=str(rdata[0][1]),fg='#5196ff',bg='white',font='msserif 35').pack()
        smf2 = Frame(b_frame,height=120,width=145,bg='white')

```

```

tr =
Label(smf2,text='Кондиционер?',fg='white',bg='#5196ff',width=100,height=2,font='msserif 15')
tr.pack(side='top')
smf2.pack_propagate(False)
smf2.place(x=140*2+5+3*2,y=30)
Label(smf2,text=str(rdata[0][2]),fg='#5196ff',bg='white',font='msserif 35').pack()
smf2 = Frame(b_frame,height=120,width=145,bg='white')
tr = Label(smf2,text='Телевизор?',fg='white',bg='#5196ff',width=100,height=2,font='msserif
15')
tr.pack(side='top')
smf2.pack_propagate(False)
smf2.place(x=140*3+12+5*2+3*3,y=30)
Label(smf2,text=str(rdata[0][3]),fg='#5196ff',bg='white',font='msserif 35').pack()
smf2 = Frame(b_frame,height=120,width=145,bg='white')
tr = Label(smf2,text=' Wi-fi?',fg='white',bg='#5196ff',width=100,height=2,font='msserif 15')
tr.pack(side='top')
smf2.pack_propagate(False)
smf2.place(x=140*4+12*2+5*3+3*4,y=30)
Label(smf2,text=str(rdata[0][4]),fg='#5196ff',bg='white',font='msserif 35').pack()
smf2 = Frame(b_frame,height=120,width=145,bg='white')
tr = Label(smf2,text=' Ціна?',fg='white',bg='#5196ff',width=100,height=2,font='msserif 15')
tr.pack(side='top')
smf2.pack_propagate(False)
smf2.place(x=140*5+12*3+5*4+3*5,y=30)
Label(smf2,text=str(rdata[0][5]),fg='#5196ff',bg='white',font='msserif 35').pack()
smf2 = Frame(b_frame,height=120,width=145,bg='white')
tr =
Label(smf2,text='Бронювання?',fg='white',bg='#5196ff',width=100,height=2,font='msserif 15')
tr.pack(side='top')
#print (rdata)
smf2.pack_propagate(False)
smf2.place(x=140*6+12*4+5*5+3*6,y=30)
p=""
if rdata[0][6]=='Unreserved':
    p = 'Hi'
else :
    p = 'Так'
Label(smf2,text=p,fg='#5196ff',bg='white',font='msserif 35').pack()

roomdet(1)
#b1 = Button(b_frame,font='mssherif 10', text="Room
1",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(1))
#sidebuttons.window_create("end",window=b)
#sidebuttons.insert("end", "\n")

sidebuttons.configure(state='disabled')
"""loop approach will fail when we connect database with button that's why itni mehnat"""
b1 = Button(b_frame,font='mssherif 10', text="Кімната 1",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(1))
b2 = Button(b_frame,font='mssherif 10', text="Кімната 2",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(2))
b3 = Button(b_frame,font='mssherif 10', text="Кімната 3",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(3))
b4 = Button(b_frame,font='mssherif 10', text="Кімната 4",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(4))
b5 = Button(b_frame,font='mssherif 10', text="Кімната 5",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(5))
b6 = Button(b_frame,font='mssherif 10', text="Кімната 6",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(6))
b7 = Button(b_frame,font='mssherif 10', text="Кімната 7",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(7))

```

```
    b8 = Button(b_frame,font='mssherif 10', text="Кімната 8",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(8))
    b9 = Button(b_frame,font='mssherif 10', text="Кімната 9",
bg='white',fg='#5196ff',width=10,command=lambda:roomdet(9))
    b10 = Button(b_frame,font='mssherif 10', text="Кімната
10",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(10))
    b11 = Button(b_frame,font='mssherif 10', text="Кімната
11",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(11))
    b12 = Button(b_frame,font='mssherif 10', text="Кімната
12",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(12))
    b13 = Button(b_frame,font='mssherif 10', text="Кімната
13",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(13))
    b14 = Button(b_frame,font='mssherif 10', text="Кімната
14",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(14))
    b15 = Button(b_frame,font='mssherif 10', text="Кімната
15",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(15))
    b16 = Button(b_frame,font='mssherif 10', text="Кімната
16",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(16))
    b17 = Button(b_frame,font='mssherif 10', text="Кімната
17",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(17))
    b18 = Button(b_frame,font='mssherif 10', text="Кімната
18",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(18))
    b19 = Button(b_frame,font='mssherif 10', text="Кімната
19",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(19))
    b20 = Button(b_frame,font='mssherif 10', text="Кімната
20",bg='white',fg='#5196ff',width=10,command=lambda:roomdet(20))
    sidebuttons.window_create("end",window=b1)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b2)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b3)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b4)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b5)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b6)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b7)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b8)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b9)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b10)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b11)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b12)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b13)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b14)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b15)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b16)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b17)
    sidebuttons.insert("end","\n")
    sidebuttons.window_create("end",window=b18)
    sidebuttons.insert("end","\n")
```

```
sidebuttons.window_create("end",window=b19)
sidebuttons.insert("end","\n")
sidebuttons.window_create("end",window=b20)
```

```
#----- payments-----
```

```
def payments():
```

```
    b_frame = Frame(root,height=400,width=1080,bg='gray89')
    path = "images/newbg6lf.jpg"
    img = ImageTk.PhotoImage(Image.open(path))
    label = Label(b_frame,image = img,height=400,width=1080)
    label.image=img
    label.place(x=0,y=0)
    l = Label(b_frame,text='Введіть унікальний ідентифікатор платежу',font='msserif
15',bg='#5196ff',fg='white')
    l.place(x=245,y=0)
    b_frame.place(x=0,y=120+6+20+60+11)
    b_frame.pack_propagate(False)
    b_frame.tkraise()
    hline = Frame(b_frame,height=42,width=1080,bg='#5196ff')
    hline.place(x=0,y=23)
    ef = Frame(hline)
    p_id = Entry(ef)
    p_id.pack(ipadx=25,ipady=3)
    ef.place(x=308,y=6)

    fl1=Frame(b_frame,height=38,width=308,bg='#5196ff')
    fl1.place(x=0,y=68)
    l1=Label(fl1,text='Дата операції',bg='#5196ff',fg='white',font='msserif 17')
    l1.pack()
    fl1.pack_propagate(False)

    fr1=Frame(b_frame,height=38,width=1080-308,bg='white')
    fr1.place(x=0+308,y=68)
    l1=Label(fr1,text='Дата операції',bg='#5196ff',fg='white',font='msserif 17')
    #l1.pack()
    fr1.pack_propagate(False)

    fl2=Frame(b_frame,height=38,width=308,bg='#5196ff')
    fl2.place(x=0,y=109)
    fl2.pack_propagate(False)
    l1=Label(fl2,text='Час транзакції',bg='#5196ff',fg='white',font='msserif 17')
    l1.pack()

    fr2=Frame(b_frame,height=38,width=1080-308,bg='white')
    fr2.place(x=0+308,y=109)
    fr2.pack_propagate(False)
    l1=Label(fr2,text='Дата операції',bg='#5196ff',fg='white',font='msserif 17')
    #l1.pack()

    fl3=Frame(b_frame,height=38,width=308,bg='#5196ff')
    fl3.place(x=0,y=150)
    fl3.pack_propagate(False)
    l1=Label(fl3,text='Сума оплати',bg='#5196ff',fg='white',font='msserif 17')
    l1.pack()

    fr3=Frame(b_frame,height=38,width=1080-308,bg='white')
    fr3.place(x=0+308,y=150)
    fr3.pack_propagate(False)
    l1=Label(fr3,text='Дата операції',bg='#5196ff',fg='white',font='msserif 17')
    #l1.pack()
```

```

fl4=Frame(b_frame,height=38,width=308,bg='#5196ff')
    fl4.place(x=0,y=191)
    fl4.pack_propagate(False)
    l1=Label(fl4,text='Спосіб оплати',bg='#5196ff',fg='white',font='ms serif 17')
    l1.pack()

fr4=Frame(b_frame,height=38,width=1080-308,bg='white')
fr4.place(x=0+308,y=191)
fr4.pack_propagate(False)
l1=Label(fr4,text='Дата операції',bg='#5196ff',fg='white',font='ms serif 17')
#l1.pack()
def getid(event=None):
    fl1=Frame(b_frame,height=38,width=308,bg='#5196ff')
    fl1.place(x=0,y=68)
    l1=Label(fl1,text='Дата операції',bg='#5196ff',fg='white',font='ms serif 17')
    l1.pack()
    fl1.pack_propagate(False)

fr1=Frame(b_frame,height=38,width=1080-308,bg='white')
fr1.place(x=0+308,y=68)
l1=Label(fr1,text='Дата операції',bg='#5196ff',fg='white',font='ms serif 17')
#l1.pack()
fr1.pack_propagate(False)

fl2=Frame(b_frame,height=38,width=308,bg='#5196ff')
fl2.place(x=0,y=109)
fl2.pack_propagate(False)
l1=Label(fl2,text='Час транзакції',bg='#5196ff',fg='white',font='ms serif 17')
l1.pack()

fr2=Frame(b_frame,height=38,width=1080-308,bg='white')
fr2.place(x=0+308,y=109)
fr2.pack_propagate(False)
l1=Label(fr2,text='Дата операції',bg='#5196ff',fg='white',font='ms serif 17')
#l1.pack()

fl3=Frame(b_frame,height=38,width=308,bg='#5196ff')
fl3.place(x=0,y=150)
fl3.pack_propagate(False)
l1=Label(fl3,text='Сума оплати',bg='#5196ff',fg='white',font='ms serif 17')
l1.pack()

fr3=Frame(b_frame,height=38,width=1080-308,bg='white')
fr3.place(x=0+308,y=150)
fr3.pack_propagate(False)
l1=Label(fr3,text='Дата операції',bg='#5196ff',fg='white',font='ms serif 17')
#l1.pack()

fl4=Frame(b_frame,height=38,width=308,bg='#5196ff')
fl4.place(x=0,y=191)
fl4.pack_propagate(False)
l1=Label(fl4,text='Спосіб оплати',bg='#5196ff',fg='white',font='ms serif 17')
l1.pack()

fr4=Frame(b_frame,height=38,width=1080-308,bg='white')
fr4.place(x=0+308,y=191)
fr4.pack_propagate(False)
l1=Label(fr4,text='Дата операції',bg='#5196ff',fg='white',font='ms serif 17')
idd = p_id.get()

fl5=Frame(b_frame,height=38,width=308,bg='#5196ff')
    fl5.place(x=0,y=150)

```

```

l1=Label(fl5,text='Paid for room',bg='#5196ff',fg='white',font='msserif 17')
l1.pack()
fl5.pack_propagate(False)

fr5=Frame(b_frame,height=38,width=1080-308,bg='white')
fr5.place(x=0+308,y=150)
l1=Label(fl1,text='Date of transaction',bg='#5196ff',fg='white',font='msserif 17')
#l1.pack()
fr5.pack_propagate(False)
'''

#print (idd)
#cur.execute("create table if not exists paymentsf(id number primary key,f_name varchar,l_name
varchar,c_number varchar,email varchar , r_n number ,day varchar,month varchar,year varchar,time varchar , method
varchar,totalamt varchar)")
cur.execute("select * from payments where id = ?",(idd,))
x = cur.fetchall()
#print(x)
cur.execute("select day,month,year,time,totalamt,r_n from paymentsf where id = ?",(idd,))
yy = cur.fetchone()
print (yy)
#print (x)
if(yy!=None):
    dot = yy[0]+'-'+yy[1]+'-'+yy[2]
    tot = yy[3]
    ap = yy[4]
    pm = '-'
    l1=Label(fr1,text=dot,height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    l2=Label(fr2,text=tot,height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    l3=Label(fr3,text=ap,height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    l4=Label(fr4,text=pm,height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
else:
    l1=Label(fr1,text='Інформація відсутня',height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    l1=Label(fr2,text='Інформація відсутня',height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    l1=Label(fr3,text='Інформація відсутня',height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    l1=Label(fr4,text='Інформація відсутня',height=38,width=1080-308,font='msserif
15',bg='white',fg='#5196ff').pack()
    ok = Button(hline,text='OK',font='msserif
10',bg='white',activebackground='steelblue',fg='#5196ff',command=getid)
    ok.place(x=530,y=5)
    p_id.bind('<Return>',getid)
    def pr():
        messagebox.askyesno("Друк","Ви хочете роздрукувати квитанцію")
    pinv = Button(b_frame,text='Роздрукувати
квитанцію',bg='#5196ff',fg='white',command=pr).place(x=900,y=235)

#-----reserve-----

def reserve():
    b_frame = Frame(root,height=420,width=1080,bg='gray89')
    path = "images/newbg6lf.jpg"
    img = ImageTk.PhotoImage(Image.open(path))
    label = Label(b_frame,image = img ,height=420,width=1080)
    label.image=img
    label.place(x=0,y=0)

```

```

#hline = Frame(b_frame,height=10,width=960,bg='#5196ff')
#hline.place(x=122,y=27)
vline = Frame(b_frame,height=400,width=7,bg='lightsteelblue3')
vline.place(x=700,y=0)

Label(b_frame,text='Персональна інформація',font='msserif 15',bg='gray93').place(x=225,y=0)

fnf = Frame(b_frame,height=1,width=1)
fn = Entry(fnf)

mnf = Frame(b_frame,height=1,width=1)
mn = Entry(mnf)

lnf = Frame(b_frame,height=1,width=1)
ln = Entry(lnf)

fn.insert(0, 'Ім`я *')
mn.insert(0, 'По батькові')
ln.insert(0, 'Прізвище *')

def on_entry_click1(event):
    if fn.get() == 'Ім`я *' :
        fn.delete(0,END)
        fn.insert(0,"")
def on_entry_click2(event):
    if mn.get() == 'По батькові' :
        mn.delete(0,END)
        mn.insert(0,"")
def on_entry_click3(event):
    if ln.get() == 'Прізвище *' :
        ln.delete(0,END)
        ln.insert(0,"")
def on_exit1(event):
    if fn.get()=="":
        fn.insert(0,'Ім`я *')
def on_exit2(event):
    if mn.get()=="":
        mn.insert(0,'По батькові')
def on_exit3(event):
    if ln.get()=="":
        ln.insert(0,'Прізвище *')

fn.bind('<FocusIn>', on_entry_click1)
mn.bind('<FocusIn>', on_entry_click2)
ln.bind('<FocusIn>', on_entry_click3)
fn.bind('<FocusOut>',on_exit1)
mn.bind('<FocusOut>',on_exit2)
ln.bind('<FocusOut>',on_exit3)

fn.pack(ipady=4,ipadx=15)
mn.pack(ipady=4,ipadx=15)
ln.pack(ipady=4,ipadx=15)
fnf.place(x=20,y=42)
mnf.place(x=235,y=42)
lnf.place(x=450,y=42)

Label(b_frame,text='Контактна інформація',font='msserif 15',bg='gray93').place(x=225,y=90)

cnf = Frame(b_frame,height=1,width=1)
cn = Entry(cnf)

```

```

emf = Frame(b_frame,height=1,width=1)
em = Entry(emf)

adf = Frame(b_frame,height=1,width=1)
ad = Entry(adf)

cn.insert(0, 'Номер телефону *')
em.insert(0, 'Електронна пошта *')
ad.insert(0, "Адреса гостя *")

def on_entry_click4(event):
    if cn.get() == 'Номер телефону *' :
        cn.delete(0,END)
        cn.insert(0,"")
def on_entry_click5(event):
    if em.get() == 'Електронна пошта *' :
        em.delete(0,END)
        em.insert(0,"")
def on_entry_click6(event):
    if ad.get() == "Адреса гостя *" :
        ad.delete(0,END)
        ad.insert(0,"")
def on_exit4(event):
    if cn.get()=="":
        cn.insert(0,'Номер телефону *')
def on_exit5(event):
    if em.get()=="":
        em.insert(0,'Електронна пошта *')
def on_exit6(event):
    if ad.get()=="":
        ad.insert(0,"Адреса гостя *")

cn.bind('<FocusIn>', on_entry_click4)
em.bind('<FocusIn>', on_entry_click5)
ad.bind('<FocusIn>', on_entry_click6)
cn.bind('<FocusOut>',on_exit4)
em.bind('<FocusOut>',on_exit5)
ad.bind('<FocusOut>',on_exit6)

cn.pack(ipady=4,ipadx=15)
em.pack(ipady=4,ipadx=15)
ad.pack(ipady=4,ipadx=15)
cnf.place(x=20,y=130)
emf.place(x=235,y=130)
adf.place(x=450,y=130)
#l = Label(b_frame,text='Please Enter The Unique Payment ID',font='msserif
15',bg='#5196ff',fg='white')
#l.place(x=245,y=0)

Label(b_frame,text='Інформація про бронювання',font='msserif
15',bg='gray93').place(x=210,y=175)

nocf = Frame(b_frame,height=1,width=1)
noc = Entry(nocf)

noaf = Frame(b_frame,height=1,width=1)
noa = Entry(noaf)

nodf = Frame(b_frame,height=1,width=1)
nod = Entry(nodf)

noc.insert(0, 'Кількість дітей *')

```

```

noa.insert(0, 'Кількість дорослих *')
nod.insert(0, 'Кількість днів проживання *')

def on_entry_click7(event):
    if noc.get() == 'Кількість дітей *' :
        noc.delete(0,END)
        noc.insert(0,"")
def on_entry_click8(event):
    if noa.get() == 'Кількість дорослих *' :
        noa.delete(0,END)
        noa.insert(0,"")
def on_entry_click9(event):
    if nod.get() == 'Кількість днів проживання *' :
        nod.delete(0,END)
        nod.insert(0,"")
def on_exit7(event):
    if noc.get()=="":
        noc.insert(0,'Кількість дітей *')
def on_exit8(event):
    if noa.get()=="":
        noa.insert(0,'Кількість дорослих *')
def on_exit9(event):
    if nod.get()=="":
        nod.insert(0,'Кількість днів проживання *')

noc.bind('<FocusIn>', on_entry_click7)
noa.bind('<FocusIn>', on_entry_click8)
nod.bind('<FocusIn>', on_entry_click9)
noc.bind('<FocusOut>',on_exit7)
noa.bind('<FocusOut>',on_exit8)
nod.bind('<FocusOut>',on_exit9)

noc.pack(ipady=4,ipadx=15)
noa.pack(ipady=4,ipadx=15)
nod.pack(ipady=4,ipadx=15)
nocf.place(x=20,y=220)
noaf.place(x=235,y=220)
nodf.place(x=450,y=220)

roomnf = Frame(b_frame,height=1,width=1)
roomn = Entry(roomnf)
roomn.insert(0, 'Введіть номер кімнати *')
def on_entry_click10(event):
    if roomn.get() == 'Введіть номер кімнати *' :
        roomn.delete(0,END)
        roomn.insert(0,"")
def on_exit10(event):
    if roomn.get()=="":
        roomn.insert(0,'Введіть номер кімнати *')
roomn.bind('<FocusIn>', on_entry_click10)
roomn.bind('<FocusOut>',on_exit10)
roomn.pack(ipady=4,ipadx=15)
roomnf.place(x=20,y=270)

pmethod = IntVar()
def booking():
    if fn.get() == 'Ім'я *' or ln.get() == 'Прізвище *' or cn.get() == 'Номер телефону *' or
em.get() == 'Електронна пошта *' or ad.get() == "Адреса гостя *" or noc.get() == 'Кількість дітей *' or noa.get() ==
'Кількість дорослих *' or nod.get() == 'Кількість днів проживання *' or roomn.get() == 'Введіть номер кімнати *':
        messagebox.showinfo('Неповна інформація','Заповніть усі поля, позначені *')
    elif fn.get() == " or ln.get() == " or cn.get() == " or em.get() == " or ad.get() == "" or
noc.get() == " or noa.get() == " or nod.get() == " or roomn.get() == " :

```



```

        reserve()
        payroot.destroy()
    else :
        messagebox.showwarning("Не вибрано", "Виберіть
спосіб оплати")
        Button(payroot, text='Оплатити', font='ms serif
12', bg='#5196FF', fg='White', width=28, command=f).place(x=0, y=200)
        Label(payroot, text='Your unique payment id
:', font='ms serif', bg='White')#.place(x=0, y=25)

def unreserve():
    if (roomn.get() == 'Введіть номер кімнати') or (roomn.get()==''):
        messagebox.showerror("Записи не заповнені", "Будь ласка, введіть номер
кімнати")
    else :
        cur.execute("update roomd set rstatus='Unreserved' where rn = ? ", (roomn.get(),))
        messagebox.showinfo("Успішно", "З кімнати успішно знята резервація")
        reserve()
        con.commit()

#right side-----
Label(b_frame, text='Фільтр', font='ms serif 20', bg='gray93').place(x=840, y=0)

nbb = IntVar()
acb = IntVar()
tvb = IntVar()
wifib = IntVar()

style = ttk.Style()
style.map("TCombobox", fieldbackground=[('readonly', 'white')])
Label(b_frame, text='Кількість ліжок:', bg='gray93', font='17').place(x=710, y=50)

#Radiobutton(b_frame, text='1', bg='gray93', variable=nbb, value=1, font='15', width=3).place(x=800, y=50)
#Radiobutton(b_frame, text='2', bg='gray93', variable=nbb, value=2, font='15', width=3).place(x=880, y=50)
#Radiobutton(b_frame, text='3', bg='gray93', variable=nbb, value=3, font='15', width=3).place(x=960, y=50)
nb = ttk.Combobox(b_frame, values=['Оберіть будь ласка...', '1', '2', '3'], state='readonly', width=22)
nb.place(x=830, y=50)
nb.current(0)
Label(b_frame, text='Кондиціонер:', font='17', bg='gray93').place(x=710, y=75)

#Radiobutton(b_frame, text='Так', bg='gray93', variable=acb, value=1, font='15', width=3).place(x=800, y=90)
#Radiobutton(b_frame, text='Hi', bg='gray93', variable=acb, value=0, font='15', width=3).place(x=880, y=90)
ac = ttk.Combobox(b_frame, values=['Оберіть будь ласка...', 'Так', 'Hi'], state='readonly', width=22)
ac.place(x=830, y=75)
ac.current(0)

Label(b_frame, text='Телевізор:', font='17', bg='gray93').place(x=710, y=100)

#Radiobutton(b_frame, text='Так', bg='gray93', variable=tvb, value=1, font='15', width=3).place(x=800, y=130)
#Radiobutton(b_frame, text='Hi', bg='gray93', variable=tvb, value=0, font='15', width=3).place(x=880, y=130)
tv = ttk.Combobox(b_frame, values=['Оберіть будь ласка...', 'Так', 'Hi'], state='readonly', width=22)
tv.place(x=830, y=100)
tv.current(0)

Label(b_frame, text='Wi-fi:', font='17', bg='gray93').place(x=712, y=125)

#Radiobutton(b_frame, text='Так', bg='gray93', variable=tvb, value=1, font='15', width=3).place(x=800, y=130)
#Radiobutton(b_frame, text='Hi', bg='gray93', variable=tvb, value=0, font='15', width=3).place(x=880, y=130)

```

```

wifi = ttk.Combobox(b_frame,values=['Оберіть будь ласка...','Так','Hi'],state='readonly',width=22)
wifi.place(x=830,y=125)
wifi.current(0)
#roomd(rn number primary key,beds number,ac varchar(10),tv varchar(10),internet varchar(10),price
number(10))
listofrooms = Listbox(b_frame,height=6,width=41)
listofrooms.place(x=735,y=190)
listofrooms.insert(END,'Тут з'являться кімнати за вашим вибором')
listofrooms.insert(END,'після застосування фільтра')
def findrooms():
    cur.execute('select rn,price,rstatus from roomd where beds = ? and ac = ? and tv = ? and
internet = ? order by price asc',((nb.get()),ac.get(),tv.get(),wifi.get()))
    x = cur.fetchall()
    #print (x)
    listofrooms.delete(0,END)
    if x == []:
        listofrooms.insert(END,'Нічого не знайдено')
    for i in x :
        listofrooms.insert(END,'Номер кімнати '+str(i[0])+' - Ціна - '+str(i[1]))
Res = Button(b_frame,text='Зарезервувати',bg='white',fg='#5196ff',font='timenewroman
11',activebackground='#5196FF',command=booking).place(x=235,y=270)
unres = Button(b_frame,text='Зняти резерв',bg='white',fg='#5196ff',font='timenewroman
11',activebackground='#5196FF',command=unreserve).place(x=360,y=270)
findrooms = Button(b_frame,text='Знайти кімнати',bg='white',fg='#5196ff',font='timenewroman
9',activebackground='#5196FF',command = findrooms).place(x=830,y=155)

scrollbar = Scrollbar(b_frame, orient="vertical")
scrollbar.config(command=listofrooms.yview)
scrollbar.place(x=1014,y=191,height=111)
listofrooms.config(yscrollcommand=scrollbar.set)
b_frame.place(x=0,y=120+6+20+60+11)
b_frame.pack_propagate(False)
b_frame.tkraise()

#-----login module-----
def login():
    q = messagebox.askyesno("Вихід", "Ви справді хочете вийти ?")
    if(q):
        root.destroy()
#-----2nd top frame-----

sl_frame = Frame(root,height=130,width=1080,bg='white')
sl_frame.place(x=0,y=70+6)
path = "images/rooms.png"
img = ImageTk.PhotoImage(Image.open(path))
b1 = Button(sl_frame,image=img,text='b1',bg='white',width=180,command=rooms)
b1.image = img
b1.place(x=180,y=0)
path2 = "images/hotelstatus.png"
img1 = ImageTk.PhotoImage(Image.open(path2))
b2 = Button(sl_frame,image=img1,text='b2',bg='white',width=180,command=hotel_status)
b2.image = img1
b2.place(x=0,y=0)
path3='images/guests.png'
img3 = ImageTk.PhotoImage(Image.open(path3))
b3 = Button(sl_frame,image=img3,text='b2',bg='white',width=180,command=staff)
b3.image = img3
b3.place(x=180*4,y=0)
path4='images/payments.png'
img4 = ImageTk.PhotoImage(Image.open(path4))
b4 = Button(sl_frame,image=img4,text='b2',bg='white',width=180,command = payments)

```

```

b4.image = img4
b4.place(x=180*3,y=0)
path5='images/logout.png'
img5 = ImageTk.PhotoImage(Image.open(path5))
b5 = Button(sl_frame,image=img5,text='b2',bg='white',width=180,height=100,command=login)
b5.image = img5
b5.place(x=180*5,y=0)
path6='images/Bookroom.png'
img6 = ImageTk.PhotoImage(Image.open(path6))
b6 = Button(sl_frame,image=img6,text='b2',bg='white',width=180,height=100,command=reserve)
b6.image = img6
b6.place(x=180*2,y=0)
Label(sl_frame,text='Статус готелю',font='msserif 13',bg='white').place(x=35,y=106)
Label(sl_frame,text='Кімнати',font='msserif 13',bg='white').place(x=242,y=106)
Label(sl_frame,text='Резервація',font='msserif 13',bg='white').place(x=407,y=106)
Label(sl_frame,text='Контакти',font='msserif 13',bg='white').place(x=778,y=106)
Label(sl_frame,text='Платежі',font='msserif 13',bg='white').place(x=600,y=106)
Label(sl_frame,text='Вихід',font='msserif 13',bg='white').place(x=968,y=106)
sl_frame.pack_propagate(False)
#-----extra frame-----
redf = Frame(root,height=6,width=1080,bg='lightsteelblue3')
redf.place(x=0,y=70)
redf1 = Frame(root,height=40,width=1080,bg='lightsteelblue3')
redf1.place(x=0,y=210)
#-----
#hotel_status() # calling the bottom frame for default page
#login()
#rooms()
#payments()
#cur.execute("select * from roomd ")
#x = cur.fetchall()
#print (x)
reserve()
#staff()
datetime()
mainloop()
def call_mainroot():
    sroot.destroy()
    mainroot()
sroot.after(1000,call_mainroot)
mainloop()

```

Ім'я користувача:
Кафедра КІ

ID перевірки:
1015239039

Дата перевірки:
25.05.2023 06:28:40 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
25.05.2023 06:34:18 EEST

ID користувача:
100005591

Назва документа: Андреев_Веб-орієнтована інформаційна система готелю

Кількість сторінок: 79 Кількість слів: 13374 Кількість символів: 106467 Розмір файлу: 2.63 MB ID файлу: 1014915261

1.82% Схожість

Найбільша схожість: 0.72% з джерелом з Бібліотеки (ID файлу: 1014517564)

1.38% Джерела з Інтернету 61 Сторінка 81

1.12% Джерела з Бібліотеки 99 Сторінка 81

0.44% Цитат

Цитати 4 Сторінка 82

Посилання 1 Сторінка 82

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи 1

Anti-Plagiarism v-15.257

Максимальне співпадіння з одним документом 0.0%

Словники перевірки: en_US, ru_RU, ua_UA. Помилки в документах: 7%

ID: 113971 Назва: БКР Веб-орієнтована інформаційна система готелю Додано в БД: 2023-05-25 Автора: В.Р. Андрєєв Керівник: С.Г. Гнатчук Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	98016	788	2046 (2%)	26 (3%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Андреев Владислав Русланович

Тема: Веб-орієнтована інформаційна система управління готелем

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 64

1. Короткий зміст роботи та прийнятих рішень: Розроблено веб-орієнтовану інформаційну систему управління готелем для ефективного керування готельними процесами, поліпшення обслуговування клієнтів і підвищення загальної продуктивності готельного бізнесу.

2. Висновок про відповідність роботи дипломному завданню: Дипломна робота відповідає поставленому завданню

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: У першому розділі проведено дослідження предметної області та представлено огляд існуючих рішень, що дозволило сформулювати постановку задачі та вимоги до веб-орієнтованої системи управління готелем. В другому розділі проведено вибір засобів реалізації та спроектовано веб-орієнтовану інформаційну систему управління готелем з врахуванням сформованих вимог. У третьому розділі реалізовано прототип веб-орієнтованої інформаційної системи готелю та представлено інтерфейс розробленої системи.

4. Позитивні сторони роботи: Запропоновано концептуальну модель бази даних та реалізовано веб-орієнтовану інформаційну систему управління готелем.

5. Негативні сторони роботи: _____

6. Оцінка графічного оформлення та пояснювальної записки роботи: графічне оформлення та пояснювальна записка оформлена згідно вимог.

7. Відгук про роботу в цілому: Робота виконана на достатньому технічному рівні


8. Інші зауваження: _____

9. Оцінка дипломної роботи: Розглянувши представлену роботу, вважаю, що
робота заслуговує оцінки добре В
(4.5)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Яшица Оксана Миколаївна, к.т.н., доцент, доцент
кафедри інженерії програмного забезпечення

"29" 05 2023 р.

 (підпис)

Завідувачу кафедри КІС
д-р.техн.наук, проф. Говорущенко Т. О.

Андрєєва Владислава Руслановича

ПІБ здобувача вищої освіти

ФІТ, 3 курсу, групи ІСТс-20-1

ЗАЯВА

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

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

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

30.05.2023

дата


підпис

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

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

Назва: Веб-орієнтована інформаційна система управління готелем

Автор: Андреев Владислав Русланович

Спеціальність: 126 – Інформаційні системи та технології

Освітня програма: освітньо-професійна

Науковий керівник: Гнатчук Є.Г. к.н.т доцент

Після аналізу звіту подібності зроблено такий висновок:

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

Підтвердження:

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

- 1) запозичення розміщені в розділах аналізу існуючих аналогів та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи;
- 2) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 3) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з 10-40 джерелами на один фрагмент речення;
- 4) в якості запозичень в окремих місцях системою зафіксовано послідовності чотирьохрозрядних двійкових кодів, які є вхідними даними до великої кількості задач і не можуть розглядатися як об'єкт авторських прав і, відповідно, їх порушення;
- 5) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 1.82% і адресується до 61,99 першоджерел, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

Керівник роботи

Гарант ОП

Завідувач кафедри КІС



Є. Г. Гнатчук

Є. Г. Гнатчук

Т. О. Говорущенко