

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА


на тему Вебтехнологія для автоматизованої системи управління
готельними комплексами


Галузь знань 12 – Інформаційні технології
Шифр і назва галузі знань

Спеціальність 122 – Комп'ютерні науки
Шифр і назва спеціальності

Освітня програма Комп'ютерні науки
Назва освітньої програми

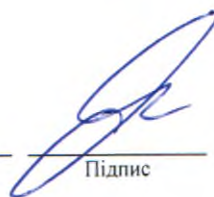
Виконав: студент 4 курсу, група КН-18-1  Е.Р. Муляр
Курс, група виконавця Підпис Ініціали, прізвище

Керівник: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

Нормоконтроль: к.т.н., доцент кафедри КН  Р.О. Багрій
Науковий ступінь, посада Підпис Ініціали, прізвище

До захисту допускаю:

Зав. кафедри КН, д.т.н., професор


Підпис

О.В. Бармак

Ініціали, прізвище

_____ 2022 р.

Анотація

Тема кваліфікаційної роботи бакалавра: «Вебтехнологія для автоматизованої системи управління готельними комплексами»

Виконавець кваліфікаційної роботи бакалавра: студент групи КН-18-1 Муляр Едуард Русланович

Керівник кваліфікаційної роботи бакалавра: к.т.н., доцент кафедри КН Багрій Руслан Олександрович

Кваліфікаційна робота бакалавра містить:

Пояснювальна записка				Кількість додатків
Сторінок	Рисунків	Таблиць	Джерел інформації	
101	66	5	48	4

Метою кваліфікаційної роботи бакалавра є створення вебтехнології для автоматизованої системи управління готельними комплексами. Для написання програмного продукту було обрано платформу розробки Laravel, СКБД – MySQL Server, мову програмування PHP, а також мову запитів SQL.

Результатом виконання кваліфікаційної роботи бакалавра є програмний продукт, який призначений для автоматизованої системи управління готельними комплексами і передбачає зрозумілий та привабливий інтерфейс, можливість реєстрації, автентифікації та авторизації користувача, операції користувачів зі своїми даними, можливість пошуку та перегляду інформації про готельні комплекси, можливість бронювання кімнат для клієнтів, реалізація модуля управління інформацією для адміністраторів, реалізація методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів. Програмний продукт розроблений на фреймворку Laravel із використанням СКБД – MySQL Server, мови програмування PHP та мови запитів SQL.

Виконавець: студент 4 курсу, група КН-18-1 М. Е. Муляр Е.Р. Муляр
Курс, група виконавця Підпис Ініціали, прізвище

Зміст

Перелік скорочень	3
Вступ.....	4
Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізації	6
1.1 Аналіз інформаційних моделей.....	6
1.2 Огляд теоретичних підходів до розв’язку подібних задач	15
1.3 Аналіз існуючих програмних рішень.....	25
1.4 Аналіз сучасних засобів створення програмного забезпечення	31
1.5 Мета, задачі та вимоги до реалізації інформаційної системи	33
Розділ 2 Проектування інформаційної системи.....	34
2.1 Аналіз та автоматизація обробки потоків даних	34
2.1.1 Вебтехнологія для автоматизованої системи управління готельними комплексами	34
2.1.2 Метод підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів.....	35
2.1.3 Функціональна структура інформаційної системи	42
2.2 Інформаційна структура системи.....	58
2.2.1 Проектна архітектура системи та взаємозв’язок компонентів.....	58
2.2.2 Інформаційна модель.....	59
2.3 Вибір засобів розробки інформаційної системи.....	61
Розділ 3 Програмна реалізація інформаційної системи	64
3.1 Структура та функціональне призначення програмних складових системи.....	64
3.2 Особливості реалізації програмних складових системи.....	70
3.3 Тестування інформаційної системи	91
3.4 Особливості реалізації методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів.....	94
3.5 Вимоги до розгортання інформаційної системи.....	100
Висновки	101
Перелік посилань.....	102
Додатки	

Перелік скорочень

Скорочення, термін, позначення	Пояснення
АІС	Автоматизована інформаційна система
БД	База даних
ІТ	Інформаційні технології
КРБ	Кваліфікаційна робота бакалавра
КН	Комп'ютерні науки
ПЗ	Програмне забезпечення
ПП	Програмний продукт
СКБД	Система керування базами даних
СБКГ	Система бронювання кімнат у готелях

Вступ

Кваліфікаційна робота бакалавра присвячена розробці вебтехнології для автоматизованої системи управління готельними комплексами.

Актуальність теми.

У наш час подорожі стали невід'ємною частиною життя людей. Кожного дня люди пересуваються з одної локації в іншу, рухаються з одної країни в іншу. Під час подорожі кожна людину бентежить таке питання, як місце де можна зупинитись переночувати та відпочити. Дане питання допомагає вирішити процес бронювання кімнат у готелях.

Бронювання кімнати у готелі є процесом суть якого полягає у тому що, подорожуючий домовляється з адміністрацією готелю про те, що певний номер буде зарезервованний на його ім'я. Це, як правило, гарантує наявність кімнати на час прибуття. При бронюванні подорожуючий, як правило, повідомляє свої особисті дані. Більшість готелів також вимагають кредитну картку як гарантію платоспроможності клієнта [1].

Інколи у готелях бувають періоди величезної кількості напливу клієнтів, особливо це помітно у пік сезони (літо, зима). В дані періоди відбувається величезне навантаження на працівників готелів, які займаються резервуванням кімнат для клієнтів. Внаслідок навантаження та людських факторів такі як неуважність та квапливість дуже часто виникають такі проблеми як плутанина з особистою інформацією клієнтів під час резервування кімнати, недодавання особистої інформації клієнта до реєстру (бази клієнтів), випадкове скасування резервування кімнати. Для вирішення даних проблем на допомогу приходять інформаційні технології.

Інформаційні технології є сукупністю методів і засобів, що використовуються для збору, зберігання, обробки і поширення інформації. В даний час діяльність людини стала сильно залежати від цих технологій, вони потребують постійного розвитку. Над розробками в галузі інформатики працюють безліч фахівців, які називаються ІТ-фахівцями або «айтішниками» [2]. Їх робота

так чи інакше пов'язана з комп'ютерами. Інформаційні технології дозволять полегшити величезну кількість організаційних моментів.

Завдяки інформаційним технологіям можна автоматизувати величезну кількість процесів для клієнтів та адміністрації готелів. Для клієнтів можна автоматизувати реєстрацію та авторизацію у системі бронювання кімнат, створити механізм для резервування кімнат у певному готелі на певний період, механізм порівняння готелів. Для адміністрації є можливість автоматизувати механізм взаємодії з особистою інформацією клієнтів, механізм отримання інформації про те, що було заброньовано кімнату, функціонал стосовно репрезентації готелю для клієнтів, механізм оформлення резервування кімнат.

Мета полягає у розробці вебтехнології для автоматизованої системи управління готельними комплексами.

Об'єкт – процес автоматизації управління готельними комплексами.

Предмет – моделі, методи та технології для системи управління готельними комплексами.

Створення веб-технології для автоматизованої системи управління готельними комплексами, де клієнти зможуть швидко бронювати кімнати у будь-якому готелі на будь-який період, переглядати та порівнювати готелі, писати відгуки на рахунок умов проживання, де адміністрація готелів має можливість швидко та інформативно оформити бронювання кімнат, взаємодіяти з репрезентацією готелю та кімнат для клієнтів є актуальною задачею впровадження інформаційних технологій у діяльність автоматизованої системи управління готельними комплексами.

Розділ 1 Характеристика предметної області: аналіз моделей, методів та реалізації

1.1 Аналіз інформаційних моделей

У 21 столітті Інтернет є невід’ємною частиною людського суспільства. Всесвітня павутина охопила практично всі сфери людського життя. Не залишився осторонь і туризм [3]. Сьогодні людині достатньо зробити кілька кліків для того, щоб придбати квитки на автобус або літак і розпочати свою подорож. Але перед початком подорожі у кожної людини виникає питання, у якому місці можна буде зупинитися для того, щоб переночувати та відпочити. Місце, яке задовольняє потребу людини в зупинці для того, щоб переночувати та відпочити називається готель.

Готель представляє собою підприємство, що надає людям, які знаходяться поза домівкою, комплекс послуг найважливішими серед яких є послуга розміщення і харчування [4]. Сьогодні сучасний готель є не тільки місцем проживання, але й досить часто виступає як частина номерного фонду, яка використовується під офіси, тимчасові представництва різних фірм і закладів. Тому існує певний різновид готелів:

- для ділових людей (загального типу, відомчі, для нарад тощо);
- для відпочинку (туристські, курортні, для автотуристів, мотелі, хостели, кемпінги);
- спеціальні типи готелів (для транзитних пасажирів, для спортсменів тощо) [5].

Одним з основних атрибутів, яким цікавиться турист під час вибору готелю для відпочинку, є зірки на його фасаді. Як правило кількість зірок готелю є показником рівня комфорту і сервісу в готелі. Класифікація готелів за «зірковістю» – це загальноприйнята система вимог, що висуваються до готелів Всесвітньою туристичною організацією в державах Європи, а також в окремих країнах Азії [6].

Класифікація готелів в залежності від кількості зірок, де:

– 1 зірка – туристичний клас, найдешевший готель, зазвичай, без додаткових послуг або ж з мінімальним їх набором. Всі «апартаменти» однакові, з аскетичними меблями. Туалет і душ, як правило, загального користування, і розташовані на поверсі. Телевізор і холодильник відсутні;

– 2 зірки – стандартний клас, бюджетний готель «без надмірностей» і довгого набору послуг, в числі яких, однак, числиться щоденне прибирання, хоч навіть і чисто формальне, раз на 3 доби;

– 3 зірки – комфортний клас, середньостатистичний і найбільш поширений тип готелів зі стандартним комплексом послуг. У всіх номерах свій санвузол, кондиціонер, фен, холодильник або міні-бар і телевізор. Прибирання, як правило, виконується щодня;

– 4 зірки – перший клас, готель високого класу, в якому представлене все вище перераховане, а також присутні спеціальні послуги, зокрема, спа-процедури, масажі, наявність декількох конференц-залів чи ресторанів;

– 5 зірок – категорія люкс, готель найвищого класу. Найрозкішніші готелі, які пропонують гостям все, що їм може знадобитися. Тут представлений найширший набір додаткових послуг, навіть ексклюзивних. Також присутні дорогі інтер'єри, може бути кілька сучасних телевізорів з кабельним. У ванній кімнаті представлений широкий вибір туалетного приладдя, аж до косметики і парфумів, банні халати, капці, джакузі. Відповідно й ціни на відпочинок в такому готелі деколи вищі, ніж в усіх попередніх, разом узятих [7].

Кожний готель є цільовим комплексом зі своєрідною структурою. Коли клієнт зупиняється в готелі або резервує готель, то як правило мається на увазі місце де клієнт зупиниться для того, щоб переночувати та відпочити. Даним місцем виступає основний елемент структури готелю кімната.

Кімната або номер (готелю) є окремим вмебльованим приміщенням, яке оснащено обладнанням та інвентарем для надання послуг з тимчасового розміщення [8]. Відповідно до різновиду готелів розрізняють також різні види готельних номерів. Існують різні критерії, що враховуються при класифікації готельних номерів:

1) За кількістю місць в номері:

– одномісний номер – номер, призначений для розміщування однієї особи та оснащений одним односпальним ліжком. У практичній діяльності широко застосовують аббревіатуру SNGL, яка походить від англійського слова «single»;

– двомісний номер з одним двоспальним ліжком – номер є призначеним для розміщування двох осіб, оснащений одним двоспальним ліжком. У практичній діяльності широко застосовують аббревіатуру DBL, яка походить від англійського слова «double»;

– двомісний номер з двома односпальними ліжками – номер, призначений для розміщування двох осіб, оснащений двома односпальними ліжками. У практичній діяльності широко застосовують аббревіатуру TWN, яка походить від англійського слова «twin»;

– багатомісний номер – номер, призначений для розміщування трьох і більше осіб, оснащений односпальними ліжками за кількістю осіб, які проживають [8].

2) за рівнем комфорту можна виділити наступні типи номерів:

– президентський апартамент – номер з трьох і більше житлових кімнат, міні-кухні або кухонної ніші, одного повного та одного чи більше додаткових санвузлів для гостей; розрахований на проживання однієї-двох осіб. Номер містить такі житлові кімнати: спальня, кабінет, вітальня або їдальня. Повний санвузол містить умивальник, ванну або джакузі, душ, унітаз, біде;

– апартамент – номер з двох і більше житлових кімнат, кухонної ніші, одного повного та одного додаткового санвузлів для гостей; розрахований на проживання однієї-двох осіб. Містить такі житлові кімнати: спальня, вітальня або їдальня, або кабінет. Повний санвузол містить умивальник, ванну, душ, унітаз;

– люкс – номер з двох і більше житлових кімнат та повного санвузла; розрахований на проживання однієї-двох осіб. Містить такі житлові кімнати: спальня, вітальня або кабінет. Повний санвузол містить умивальник, ванну або душ, унітаз;

- дуплекс – двоповерховий номер з двох і більше житлових кімнат із внутрішніми стаціонарними сходами, та повним санвузлом; розрахований на проживання однієї-двох осіб. Повний санвузол містить умивальник, ванну, душ, унітаз;
- напівлюкс – номер з однієї житлової кімнати з плануванням, яке дає змогу використовувати частину приміщення як вітальню або місце для готування та приймання їжі, та повного санвузла; розрахований на проживання однієї-двох осіб. Повний санвузол містить умивальник, ванну або душ та унітаз;
- стандарт – номер з однієї житлової кімнати та повного санвузла; розрахований на проживання однієї-двох осіб. Повний санвузол містить умивальник, ванну або душ та унітаз [8].

Перед тим як потрапити до готельного номеру клієнту необхідно здійснити обов'язкову операцію, яка надасть змогу заселитися до кімнати та можливість використовувати всі готельні послуги. Даною операцією виступає процес бронювання.

Бронювання є попереднім замовленням місць і номерів у готелі [9]. Процес бронювання виступає стартовим рушієм в обслуговуванні гостей в готелі. Бронюванням номерів як правило займаються менеджери відділу бронювання або служби прийому і розміщення. Саме до цих підрозділів надходять заявки на бронювання від клієнтів. Крім формального збору заявок, відділ бронювання повинен вивчати попит на готельні місця. Вивчаючи багаторічний досвід роботи свого готелю та готелів конкурентів, прогножуючи попит на місця та аналізуючи завантаження номерів у минулі і поточний періоди, відділ бронювання разом зі службою маркетингу планує діяльність готелю [9].

Всю інформацію стосовно аналізу та діяльності готелю як правило представляють у вигляді таблиць та графіків для кращого розуміння ситуацій. Приклад такого аналізу представлено на рисунках 1.1 та 1.2.

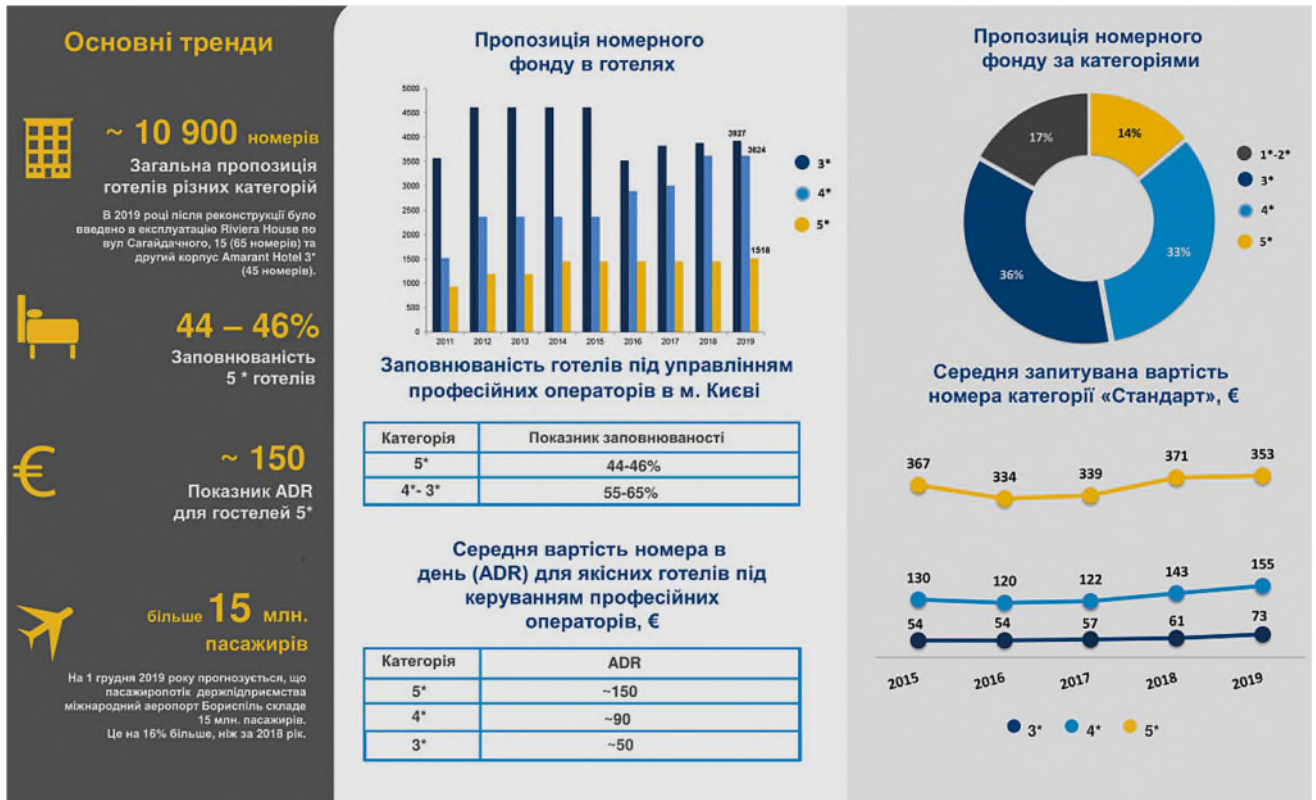


Рисунок 1.1 – Статистика попиту клієнтів на готельні номери

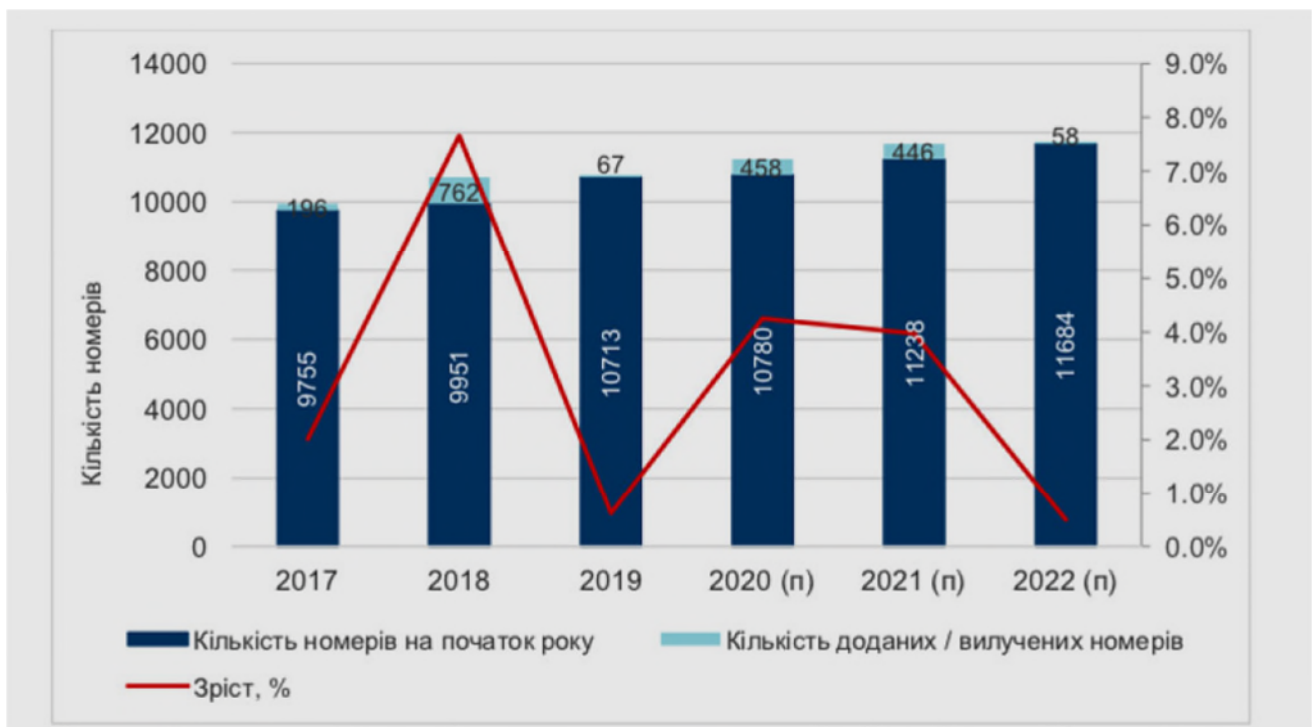


Рисунок 1.2 – Динаміка зміни номерного фонду готелів

Відповідно до різновиду готелів та готельних номерів існує різновид бронювання. Всього існує три основних види резервування номерів:

– Гарантоване бронювання являється видом бронювання, який припускає, що номер у готелі буде триматися вільним до тих пір, поки клієнт не приїде заселятися. Таким чином, у туриста немає жорстких обмежень у часі, і йому можна не поспішати заселитися, щоб не залишитися без житла. Правда, якщо у нього трапиться форс-мажор, із-за якого він відмовиться від номера, то доведеться відшкодувати витрати;

– Негарантоване бронювання припускає, що номер у готелі буде триматися вільним тільки до 18.00, і якщо до того часу клієнт у нього не заселиться, то броня автоматично знімається;

– Подвійне бронювання передбачає резервування вже раніше зарезервовані номери. Тобто якщо клієнт хоче оселитися в певному готелі, а там місць немає, він все одно може зробити бронь в надії, що той, хто раніше забронював номер, відмовиться від нього [10].

Сьогодні більшість подорожуючих окрім традиційного бронювання масово використовують онлайн-бронювання.

Онлайн-бронювання є бронюванням через Інтернет, в інтерактивному режимі. Як правило даний тип бронювання здійснюється за допомогою онлайн систем в яких відображено наявність реальних номерів в готелях, і в яких інформування готелю про бронювання номера клієнтом відбувається менш ніж за одну хвилину з моменту бронювання [11].

Існує два види онлайн-бронювання:

– Істинне онлайн-бронювання – коли клієнт вибирає, бронює і оплачує номер самостійно через системи ADS або безпосередньо на сайті готелю. Для оплати клієнт використовує пластикову картку, дані якої передаються в готель безпосередньо, і через кілька секунд отримує ваучер, що підтверджує бронь.

– Псевдоонлайн-бронювання (також «бронювання за запитом») – коли клієнт заповнює форму заявки на сайті готелю і відправляє її адміністратору. Протягом деякого часу клієнт чекає дзвінка співробітника служби бронювання для підтвердження броні [11].

Якщо порівнювати традиційне бронювання та онлайн-бронювання, онлайн-бронювання матиме більше переваг навпроти традиційному. До переваг онлайн-бронювання можна віднести:

- можливість клієнта миттєво зарезервувати кімнату у будь-якому готелі у будь-який час доби;
- можливість комфортно переглянути та порівняти різні готелі;
- швидка оплата за резервування кімнати;
- готелям немає необхідності зв'язуватися з клієнтом, так як бронювання проходить в автоматичному режимі без участі адміністратора [11];
- готелі мають можливість комфортно визначати квоти для бронювання онлайн, ціни, набір послуг.

Основною та найбільш значимою перевагою традиційного бронювання навпроти онлайн можна відзначити побудову кращих взаємовідносин з клієнтами. Якщо правильно використати дану перевагу, тобто запропонувати клієнтам цікаві послуги, чимось приємно вразити, у клієнта виникне бажання знову повернутися до готелю навіть ще не будучи там раніше. Ця перевага має достатньо важливе значення і це та річ якою не може похвалитися онлайн-бронювання.

Важливим атрибутом у функціонуванні готелю є персонал. Як правило персонал готелю розділений на певні служби з метою забезпечити максимальну ефективність роботи готелю. Основними службами у готелі є служба управління номерним фондом, адміністративна служба, служба громадського харчування, комерційна служба, служба обслуговування, інженерна служба, служба безпеки, додаткові служби [12].

Отже, основними параметрами предметної області для задачі, що розглядається, є:

Параметр предметної області	Опис
Готелі	
Назва	Коротке найменування (назва) готелю
Адреса	Місце розміщення готелю

Опис	Детальна інформація про готель
Рейтинг	Інформація про кількість зірок готелю
Номер типу готелю	Номер, слугує для ідентифікації певного типу готелю
Кількість номерів	Інформація про кількість вільних номерів
Типи готелів	
Назва	Коротке найменування (назва) типу готелю
Кімнати	
Номер	Номер кімнати у готелі
Номер готелю	Номер, слугує для ідентифікації певного готелю
Номер типу кімнати	Номер, слугує для ідентифікації певного типу кімнати
Номер типу комфортності кімнати	Номер, слугує для ідентифікації певного типу комфортності кімнати
Номер типу ліжка у кімнаті	Номер, слугує для ідентифікації певного типу ліжка у кімнаті
Опис	Детальна інформація про кімнату
Статус	Інформація про те, чи зарезерована кімната
Ціна	Ціна кімнати за одну ніч
Номер додаткової послуги	Номер, слугує для ідентифікації певної додаткової послуги
Типи кімнат	
Назва	Коротке найменування (назва) типу кімнати
Типи комфортності кімнат	

Назва	Коротке найменування (назва) типу комфортності кімнати
Додаткові послуги	
Назва	Коротке найменування (назва) додаткової послуги
Типи ліжок у кімнатах	
Назва	Коротке найменування (назва) типу ліжка у кімнаті
Клієнти	
Ім'я	Ім'я клієнта
Прізвище	Прізвище клієнта
Дата народження	Дата народження клієнта
Телефон	Номер мобільного телефону клієнта
Електронна адреса	Особова інформація, слугує для реєстрації/авторизації на сайті, є унікальною
Пароль	Таємне слово, слугує для реєстрації та авторизації на сайті
Бронювання	
Номер клієнта	Номер, слугує для ідентифікації конкретного клієнта
Номер готелю	Номер, слугує для ідентифікації певного готелю
Номер кімнати	Номер, слугує для ідентифікації певної кімнати належної до вибраного готелю
Кількість ночей	Інформація про кількість ночей на яку клієнт зупиниться у готелі
Дата прибуття та від'їзду	Дата та час прибуття та від'їзду клієнта
Рахунок	Загальна ціна за бронювання

1.2 Огляд теоретичних підходів до розв'язку подібних задач

Сьогодні готельні комплекси у більшості випадків організовують свою діяльність у мережі Інтернет. Впровадження веб-технологій у системи керування готельними комплексами дозволило суттєво скоротити час на обслуговування клієнтів, забезпечити швидке та зручне онлайн резервування кімнат для клієнтів, знизити собівартість послуг.

У процесі дослідження було виявлено певний спектр технологій, які доцільно використовують для створення онлайн систем керування готельними комплексами.

Для побудови «скелету», веб-сайти, як правило використовують наступні архітектурні шаблони проектування: MVC, MVP, MVVM.

Архітектурний шаблон MVC

MVC – архітектурний шаблон проектування, який передбачає поділ системи на три взаємопов'язані частини:

- модель даних – це компонент, який представляє собою бізнес-логіку програмного продукту і включає методи вибору, обробки та надання різних даних;
- вигляд (інтерфейс користувача) – компонент, який використовується для завдання зовнішнього відображення даних;
- контролер – компонент, який відповідає за зв'язок між моделлю даних та виглядом. Контролер визначає, як програмний продукт повинен реагувати на зовнішні дії користувача [13].

Шаблон MVC дозволяє створювати програми, різні аспекти яких (логіка введення, бізнес-логіка та логіка інтерфейсу) розділені, але досить тісно взаємодіють один з одним. Ця схема вказує розташування кожного виду логіки у додатку. Логіка інтерфейсу користувача відноситься до вигляду. Логіка введення відноситься до контролера. Бізнес-логіка розміщується у моделі. Цей поділ дозволяє працювати зі складними структурами при створенні програми, оскільки забезпечує одночасну реалізацію лише одного аспекту [14].

Використання даного шаблону для розробки та підтримки онлайн системи керування готельними комплексами є гарним та практичним рішенням:

1. Шаблон дозволяє розділити весь об'єм інформаційної системи на три компоненти. Тим самим забезпечити охайну організацію модулів системи та забезпечити майбутню масштабованість проєкту, так як зникає залежність модулів один від одного.

2. Дозволяє легко інтегрувати схему основного контролера, коли запити веб-програми обробляються через один контролер. Це дозволяє створювати програми, які підтримують розширену інфраструктуру маршрутизації.

3. Облегшує підтримку проєкту та тестування коду.

На рисунку 1.3 представлено принцип роботи веб-сайту на базі шаблону MVC.

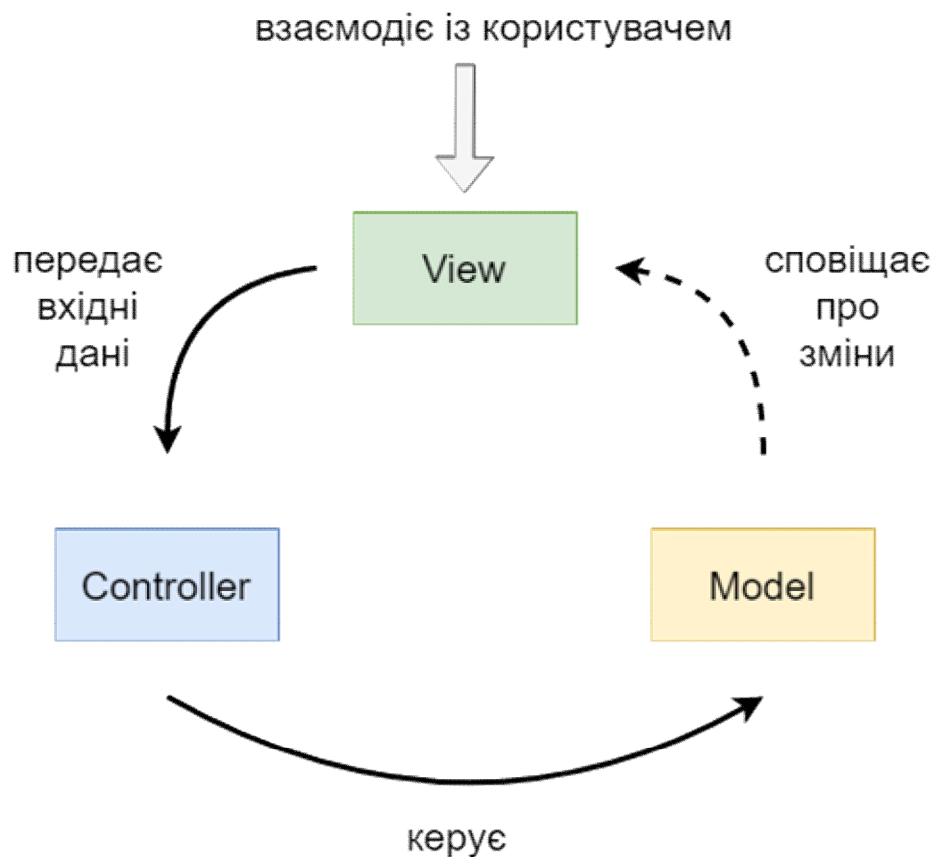


Рисунок 1.3 – Принцип роботи веб-сайту на базі шаблону MVC

Архітектурний шаблон MVP

MVP є архітектурним шаблоном проектування, що відділяє візуальне відображення та поведінку обробки подій у різні класи. Даний шаблон складається з наступних частин:

- Модель являє собою клас для визначення даних, які будуть відображатися або над якими будуть проводитися інші дії у інтерфейсі користувача;
- Вид реалізує відображення даних (з Моделі), звертається до Презентера за оновленнями, перенаправляє події від користувача до Презентера;
- Презентер реалізує взаємодію між Моделлю та Видом і містить у собі всю логіку уявлення даних про предметну область [15].

Принцип роботи шаблону полягає у наступному:

1. Вигляд будує інтерфейс і додає дані з моделі.
2. Користувач бачить інформацію та взаємодіє з інтерфейсом.
3. Вигляд перехоплює події та передає (делегує) їх представнику.
4. Представник обробляє дані (не завжди) та передає їх моделі.
5. Модель виконує певні операції та оновлюється (змінює ті чи інші властивості).
6. Представник отримує оновлену модель та передає її виду.
7. Вигляд створює інтерфейс з новими даними [16].

Використання даного шаблону для розробки та підтримки онлайн системи керування готельними комплексами є цікавим рішенням:

- Модель і вид повністю розділені, тому є можливість змінити вигляд, не торкаючись моделі.
- Всі взаємодії відбуваються в одному місці – усередині Презентера, модель можна використовувати більш ефективно.
- Якщо помістити логіку в Презентер, то можна протестувати логіку без інтерфейсу користувача (модульне тестування).

На рисунку 1.4 представлено принцип роботи веб-сайту на базі шаблону MVP.

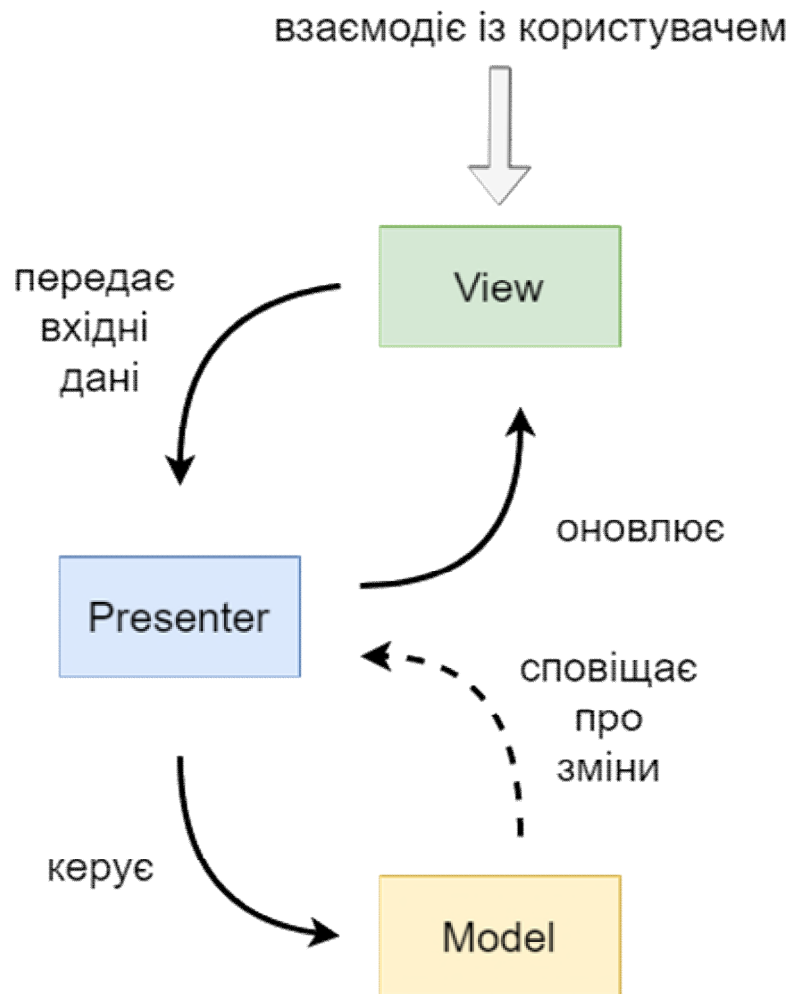


Рисунок 1.4 – Принцип роботи веб-сайту на базі шаблону MVP

Архітектурний шаблон MVVM

MVVM є архітектурним шаблоном проектування, що дозволяє полегшити відокремлення розробки графічного інтерфейсу від розробки бізнес логіки, відомої як модель. Модель представлення є частиною, яка відповідає за перетворення даних для їх подальшої підтримки і використання. З цієї точки зору, модель представлення більше схожа на модель, ніж на представлення і оброблює більшість, якщо не всю, логіку відображення даних. Модель представлення може також реалізовувати патерн медіатор, організовуючи доступ до бек-енд логіки навколо множини правил використання, які підтримуються представленням.

MVVM зручно використовувати замість класичного MVC та йому подібних у тих випадках, коли на платформі, де ведеться розробка, присутнє «зв'язування даних». В MVC/MVP зміни у користувацькому інтерфейсі не

впливають безпосередньо на модель, а йдуть через Контролер/Презентер. У таких технологіях, як WPF та Silverlight, присутня концепція «зв'язування даних», що дозволяє зв'язувати дані із візуальними елементами в обидві сторони [17].

Даний шаблон складається з 3 частин:

- **Модель.** Описує дані, що використовуються в додатку. Моделі можуть містити логіку, безпосередньо пов'язану з цими даними, наприклад, логіку валідації властивостей моделі. У той же час модель не повинна містити жодної логіки, пов'язаної з відображенням даних та взаємодією з візуальними елементами керування.

- **Представлення.** Визначає візуальний інтерфейс, через який користувач взаємодіє з програмним додатком.

- **Модель представлення.** Пов'язує модель та уявлення через механізм прив'язки даних. Модель представлення містить логіку отримання даних з моделі, які потім передаються у представлення, а також визначає логіку щодо оновлення даних у моделі. Оскільки елементи уявлення, тобто візуальні компоненти типу кнопок, не використовують події, уявлення взаємодіє з моделлю представлення за допомогою команд [18].

Використання даного шаблону для розробки та підтримки онлайн системи керування готельними комплексами є гарним рішенням:

- **Можливість ітеративного довільного стилю програмування.** Ізольовані зміни менш небезпечні та зручніші для експериментування.

- **Спрощення модульного тестування.** Блоки коду, ізольовані один від одного, можна тестувати окремо та поза робочими середовищами.

- **Підвищення зручності обслуговування.** Усунення помилок у незв'язаному коді знижує можливість виникнення регресії в іншому коді.

На рисунку 1.5 представлено принцип роботи веб-сайту на базі шаблону MVVM.

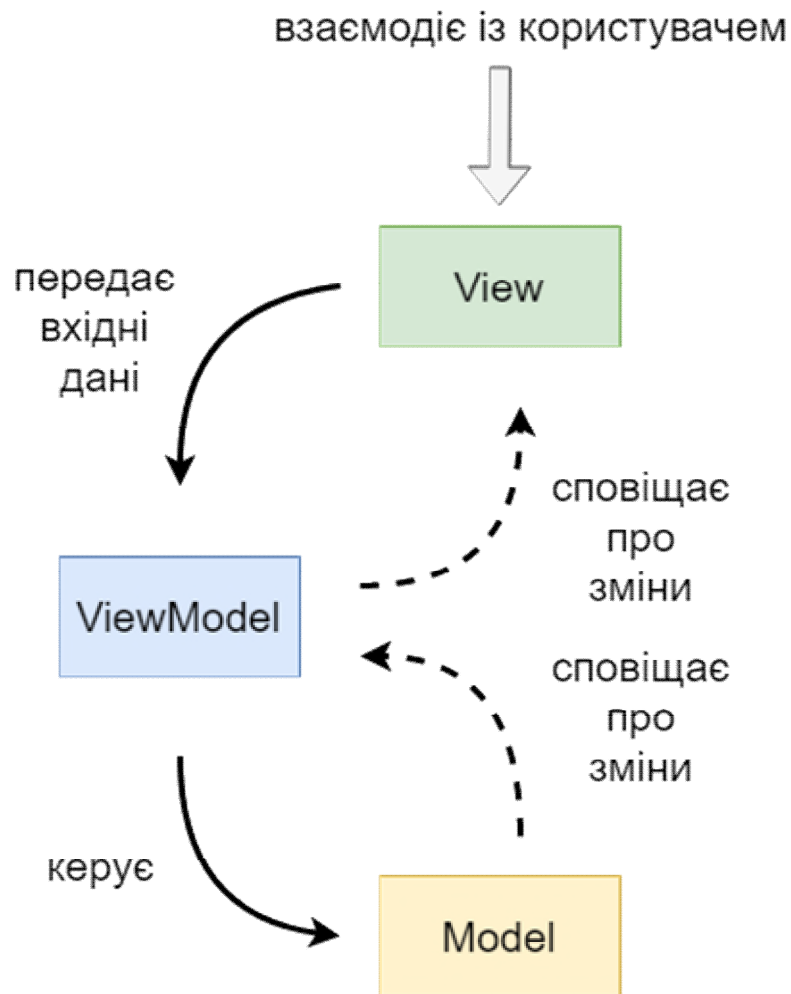


Рисунок 1.5 – Принцип роботи веб-сайту на базі шаблону MVVM

Для вирішення задач даної предметної області найбільш доцільно буде використати шаблон MVC, так як даний шаблон є найпростішим в плані реалізації серед представлених, а також має більшу гнучкість в плані взаємодії між компонентами системи.

Система керування готельними комплексами є додатком, який зберігає та оперує величезну кількість даних. Для виконання даних операцій, як правило використовують бази даних.

База даних є іменованою сукупністю даних, що відбиває стан об'єктів і їхнє відношення в розглянутій предметній області [19]. Об'єднання великої кількості даних в єдину базу дає змогу для формування безлічі варіації групування інформації — особисті дані клієнта, історія резервувань, реєстр пропозицій готелів та будь-що інше. Головною перевагою БД є швидкість внесення та використання потрібної інформації. Завдяки спеціальним алгоритмам, які

використовуються для баз даних, можна легко знаходити необхідні дані всього за декілька секунд. Також в базі даних існує певний взаємозв'язок інформації: зміна в одному рядку може спричинити зміни в інших рядках — це допомагає працювати з інформацією простіше і швидше [20].

Існує відповідний різновид баз даних. Кожний вид слугує для вирішення конкретного спектру задач. Для даної предметної області доцільно буде використати реляційну базу даних.

Реляційною називається база даних, у якій всі дані, доступні користувачу, організовані у вигляді таблиць, а всі операції над даними зводяться до операцій над цими таблицями. Для представлення реляційних баз даних розроблена формальна теорія баз даних, теоретичну основу якої складає алгебра та математична логіка. Одною із особливостей реляційної бази даних є нормалізація даних, що дозволяє видаляти надлишкові дані з кожної таблиці бази даних. Процес нормалізації переслідує подвійну мету: видалити надлишкові копії даних і забезпечити максимальну гнучкість, як в структурах таблиць, так і в її інтерфейсних програмах на випадок можливих майбутніх змін баз даних [21].

У процесі розробки онлайн системи керування готельними комплексами необхідно писати величезну кількість запитів до бази даних. Але написання такої кількості запитів займатиме величезну кількість часу, а також присутній ризик написання «поганих» запитів, які в свою чергу будуть негативно впливати на продуктивність додатку.

Для того, щоб уникнути даних негативних факторів, як правило використовують технологію ORM.

ORM Eloquent

ORM є технологією, що дозволяє автоматично пов'язати базу даних із кодом. Даний механізм намагається приховати існування бази даних настільки, наскільки це можливо. Натомість розробнику дають можливість оперувати даними в базі через спеціальний інтерфейс. Замість побудови SQL-запитів розробник викликає прості методи, а всю решту роботи бере на себе ORM.

Eloquent являє собою відповідний тип ORM і базується на ідеї, що кожній таблиці у додатку відповідає один клас (модель) [22]. Основним призначенням даного типу є зміна схеми бази даних: додавання, видалення та модифікація таблиць. Робиться це, як правило, не на чистому SQL, а за допомогою спеціальної мови. Це дозволяє працювати з ORM, не відволікаючись на особливості конкретних баз даних. ORM сама створює правильний SQL-запит, що підходить під конкретну базу даних.

У Eloquent використовується підхід Database First. Це означає, що для створення нових моделей або зміни поведінки старих потрібно спочатку змінити базу даних, а ORM автоматично підхоплює зміни і працює з ними. Наприклад, для додавання нової властивості достатньо додати нову колонку. У коді нічого змінювати не потрібно, ORM автоматично починає працювати.

Використання ORM для розробки та підтримки автоматизованої системи є гарним та практичним рішенням:

1. Пришвидшення розробки автоматизованої системи у декілька десятків разів. Так як система є досить об'ємною, написання коректних запитів для взаємодії з базою даних займатиме величезну кількість часу. Використовуючи ORM, декілька SQL запитів до бази даних можна буде замінити 1-2 методами.

2. ORM запобігає такому явищу як SQL ін'єкція (впровадження в SQL запит небезпечного коду).

3. Легко інтегрується з будь-якою базою даних.

4. Наявність механізму кешування об'єктів, що зменшує навантаження на базу даних.

На рисунку 1.6 представлено принцип роботи ORM.

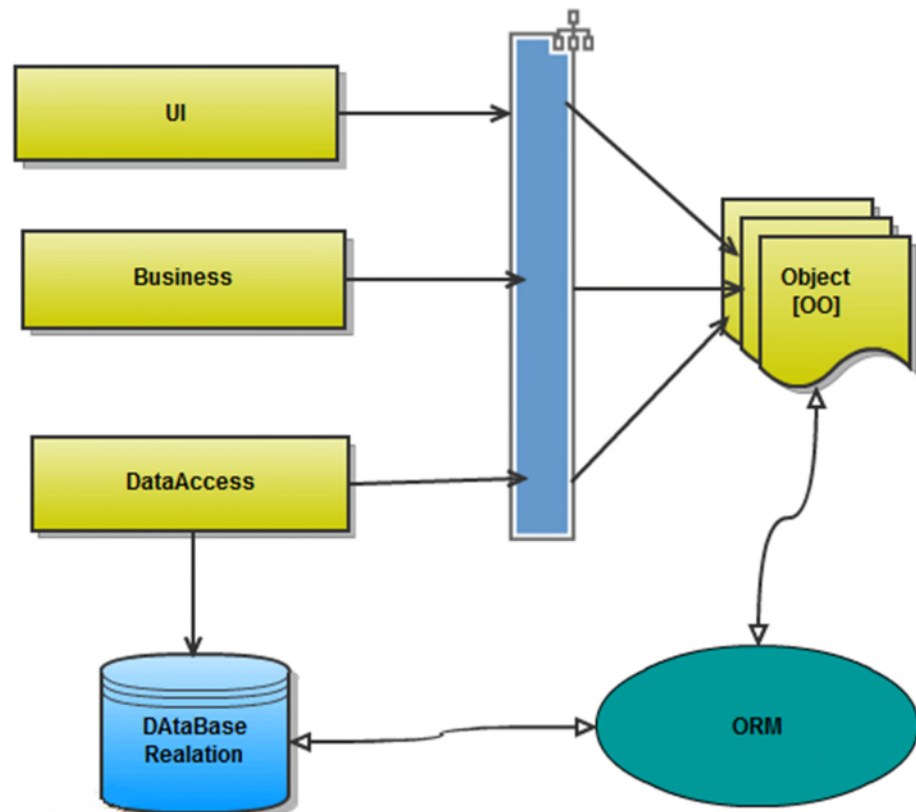


Рисунок 1.6 – Принцип роботи ORM

Онлайн системою керування готельними комплексами користується як правило величезна кількість користувачів. Тому такі речі як реєстрація, авторизація, надання та управління доступом, конфіденційність особової інформації користувача є обов'язковими атрибутами таких систем.

Пакет інструментів керування доступом та станом користувачів

Jetstream

Jetstream — стартовий набір додатків платформи Laravel. Jetstream надає реалізацію для входу у систему, реєстрацію, перевірки електронної пошти, двофакторну автентифікацію, керування сесансами [23].

У контексті використання автоматизованої системи, Jetstream дозволить:

1. Створити реєстрацію та авторизацію у системі.
2. Можливість розділення користувачів на різні категорії з відповідними правами доступу до інформації.
3. Забезпечити безпеку особистих даних користувачів за рахунок двофакторної автентифікації на базі номеру телефону або QR-коду.

4. Керування профілями користувачів є ключовою особливістю Jetstream, яка відрізняє його від інших. Тут користувач у реальному часі може легко оновлювати особисту інформацію: зображення профілю, ім'я, електронну адресу та пароль. Присутня можливість перегляду сеансів браузера та виходити з них. Користувачі можуть повністю видалити свій обліковий запис.

Для пришвидшення розробки автоматизованих систем як правило використовують шаблонізатори, так як вони надають готовий функціонал для створення візуального вигляду додатку.

Шаблонізатор Blade

Blade є потужним движком шаблонів. На відміну від деяких механізмів PHP шаблонів, Blade не обмежує у використанні простого PHP-коду у своїх шаблонах. Насправді, усі шаблони Blade компілюються у простий PHP-код і кешуються, поки вони не будуть змінені, тобто Blade не додає до програмного застосунку накладних витрат [24].

У процесі розробки автоматизованої системи даний шаблонізатор дозволить:

- Розділити код на певні секції і тим самим забезпечити читабельність та гнучкість.
- Наявність механізму наслідування шаблонів дозволяє створити каркас додатку і потім у процесі розробки лише змінювати контенту частину цього каркасу.
- Наявність вбудованих функцій для взаємодії з PHP кодом: вивід контенту, цикли, пагінація тощо.

Онлайн системи керування готельними комплексами оперують такими речами як фотографії готелів, номерів, клієнтів, забезпечують можливість створювати звітні файли. Для таких випадків, як правило використовують технологію файлової системи.

Файлова система Flysystem

Flysystem є потужною файловою системою для зберігання файлів. Інтеграція Laravel Flysystem надає прості драйвери для роботи з локальними

файловими системами, SFTP та Amazon S3. Навіть краще, перемикається між цими параметрами зберігання, між локальною машиною розробки та виробничим сервером надзвичайно просто, оскільки API залишається однаковим для кожної системи [25].

У контексті функціонування автоматизованої системи технологія Flysystem дозволить:

1. Налаштувати автоматичне завантаження фотографій готельних комплексів в онлайн простір.
2. Наявність механізму стиснення файлів для економії місця зберігання.
3. Можливість завантаження файлів у цифрову хмару.
4. Створити каталоги під конкретні типи файлів.
5. Задавати область видимості файлів.

Отже, під час дослідження було розглянуто різний спектр технологій, які використовуються для вирішення різних задач даної предметної області. Для створення онлайн системи керування готельними комплексами будуть використовуватися наступні технології: архітектурний шаблон MVC, бази даних, ORM Eloquent, шаблонізатор Blade, файлова система Flysystem, пакет інструментів керування доступом та станом користувачів Jetstream.

1.3 Аналіз існуючих програмних рішень

У процесі дослідження було виявлено певний різновид онлайн-систем керування готельними комплексами. Як правило дані онлайн-системи класифікують за двома напрямками: зовнішня система (комп'ютерні системи резервування (CRS)) [26] та внутрішня автоматизація процесів (система керування готелем (PMS)) [27].

Прикладом ресурсу категорії «комп'ютерні системи резервування» є система онлайн-бронювання «Booking.com» [28], головна сторінка якої зображена на рисунку 1.3.

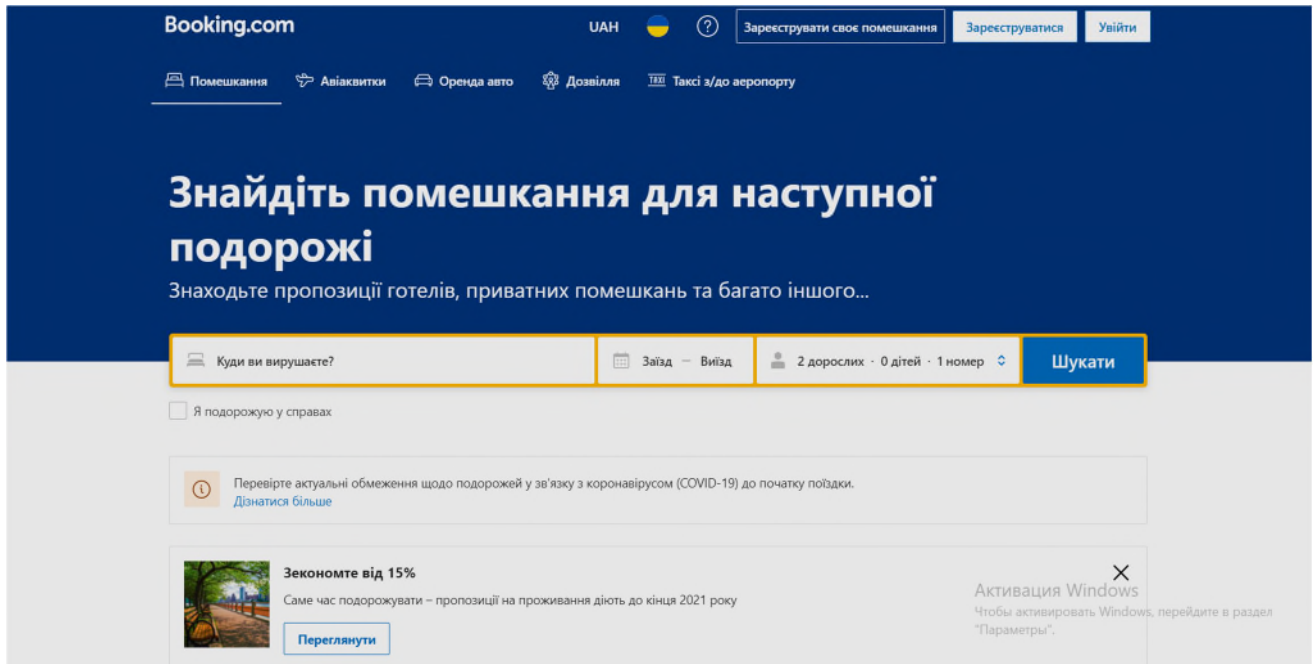


Рисунок 1.3 – Головна сторінка системи онлайн-бронювання «Booking.com»

Booking.com – система інтернет-бронювання готелей. Дана система пропонує подорожуючим вибір більше ніж 320 тисяч готелей в 184 країнах світу [29].

Система забезпечує можливість швидкого та гнучкого пошуку готелів, перегляду детальної інформації про певний готель, перегляду детальної інформації про кімнати відповідного готелю, резервування кімнат, можливість перегляду розміщення готелів на Google картах, наявність особистого кабінету клієнта, можливість додати певний готель до списку улюблених, наявність історії резервувань, можливість оцінювати та залишати відгуки про готелі для клієнтів.

Недоліком даної системи є відсутність функціоналу стосовно зручного порівняння послуг готелей один з одним.

Наступним представником веб-ресурсу є система онлайн-бронювання «Hotels24.ua» [30], головна сторінка якої представлено на рисунку 1.4.

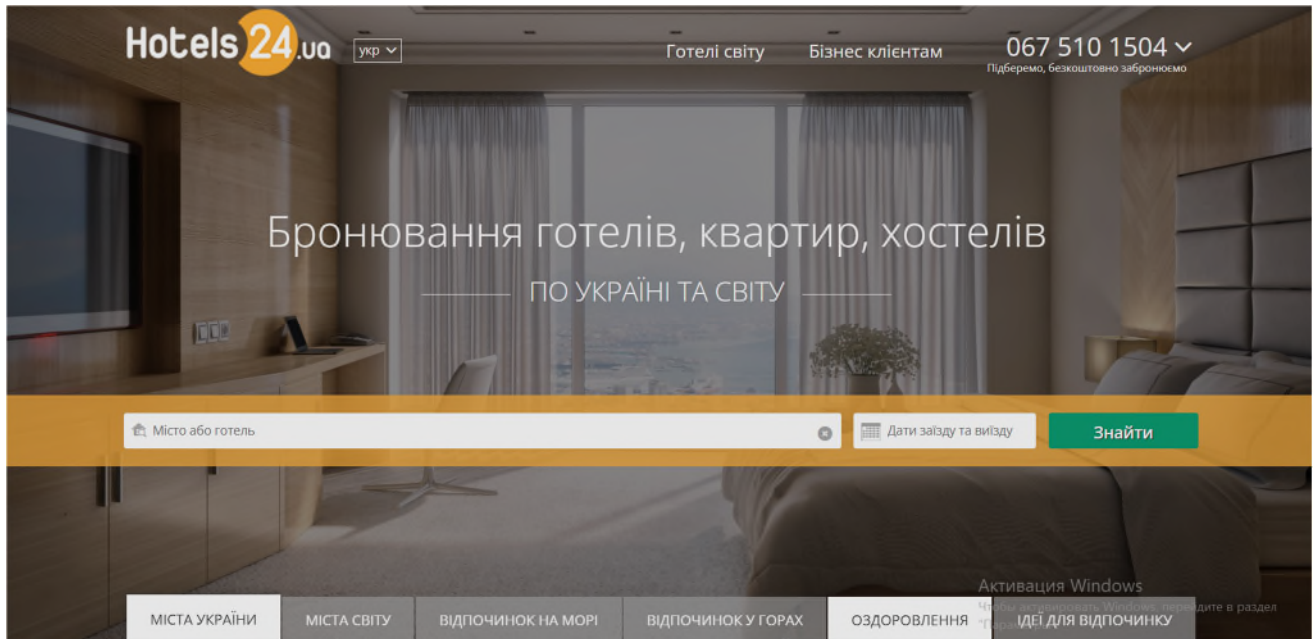


Рисунок 1.4 – Головна сторінка системи онлайн-бронювання готелей
«Hotels24.ua»

Hotels24.ua – безкоштовний сервіс бронювання готелей по всій Україні. Система пропонує зручний вибір з-поміж 4921 готелів у 257 містах України [31].

Серед переваг даної системи можна відмітити гнучкий пошук готелів, можливість перегляду детальної інформації про готелі та готельні номери, швидке та зручне резервування готельних номерів.

Недоліками даної системи є відсутність особистого кабінету, можливості переглянути розміщення готелів на Google картах, перегляду історії резервувань, механізму додавання певного готелю до списку улюблених, функціоналу стосовно зручного порівняння послуг готелей один з одним, можливості оцінювати та залишати відгуки про готелі для клієнтів.

Прикладом ресурсу категорії «системи керування готелем» є онлайн система «innRoad», головна сторінка якої зображена на рисунку 1.5.

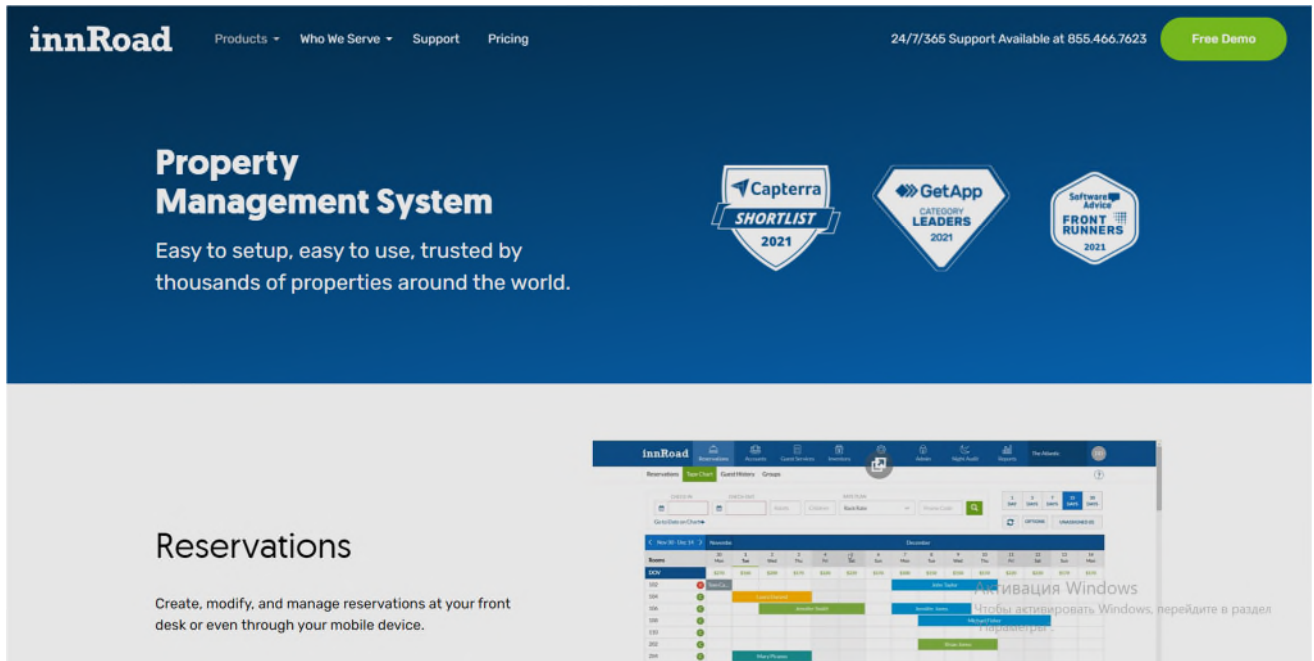


Рисунок 1.5 – Головна сторінка системи управління готелем «innRoad»

innRoad є системою управління готелем, що забезпечує створення, модифікацію та управління бронюванням знаходячись у будь-якій точці світу та використовуючи будь-який електронний пристрій з доступом до мережі Інтернет [32].

Перевагами даної система є можливість комфортної взаємодії з шахматкою, управління номерним фондом, керування справами, наявність особистого кабінету.

Недоліком даної системи є відсутність гнучкого керування тарифами.

Наступним представником веб-ресурсу є система керування готелем «Hotel Connect» [33], головна сторінка якої представлено на рисунку 1.6.

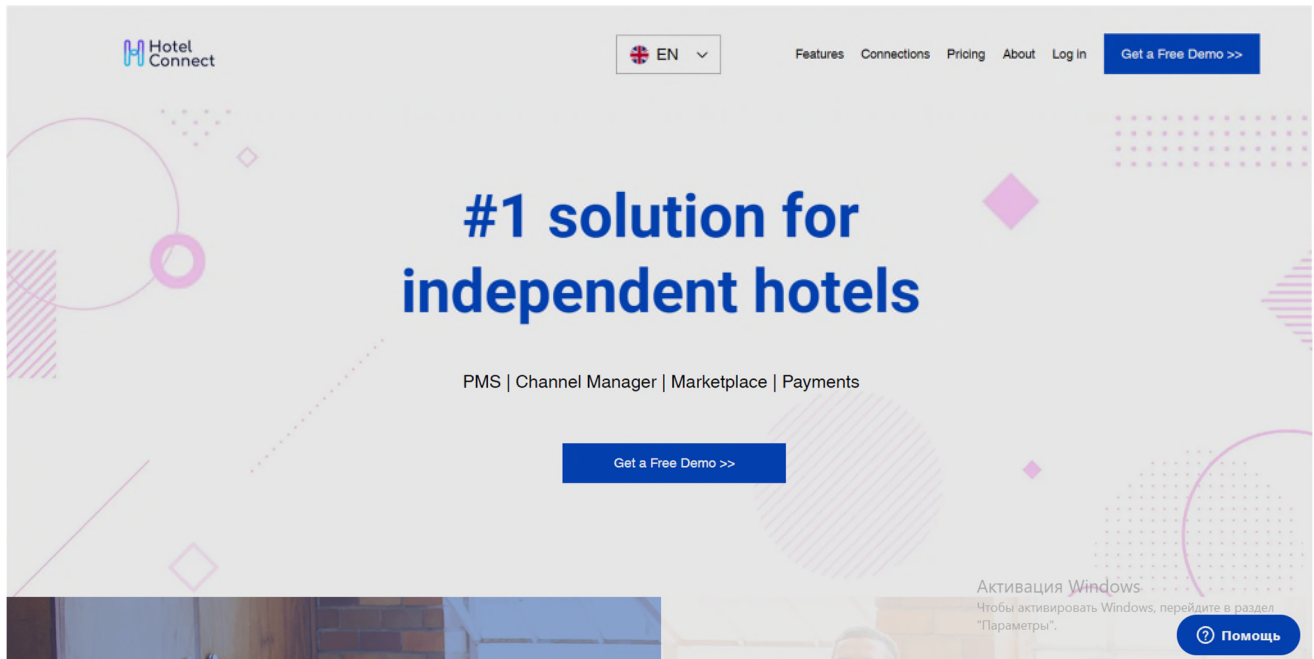


Рисунок 1.6 – Головна сторінка системи управління готелем «Hotel Connect»

Hotel Connect є системою управління готелем, яка надає можливість швидко створити нове бронювання та зареєструвати клієнта. Система пропонує різноманітний функціонал у сфері управління готельним бізнесом [34].

Серед переваг даної системи можна відмітити наявність особистого кабінету, можливість управління номерним фондом, керування справами.

Недоліками даної системи є відсутність комфортної взаємодії з шахматкою, модуля керування тарифами.

Більш наглядне порівняння систем продемонстровано в таблицях 1 (комп'ютерні системи резервування) та 2 (системи керування готелем).

Таблиця 1 – Порівняння комп'ютерних систем резервування

Критерій	«Booking.com»	«Hotels24.ua»
Особистий кабінет користувача	+	-
Перегляд репрезентативної інформації про готель	+	+
Перегляд репрезентативної	+	+

інформації про кімнати певного готелю		
Історія резервувань	+	-
Пошук готелей	+	+
Бронювання готельних номерів	+	+
Геолокація готелей	+	-
Можливість додати готель до списку улюблених	+	-
Можливість залишати відгуки та оцінку стосовно готелів	+	-
Модуль порівняння послуг готелей один з одним	-	-
<i>Разом</i>	9	4

Таблиця 2 – Порівняння систем керування готелем

Критерій	«innRoad»	«Hotel Connect»
Особистий кабінет користувача	+	+
Управління номерним фондом	+	+
Керування тарифами	-	-
Шахматка	+	-
Керування справами	+	+
<i>Разом</i>	4	3

Отже, проаналізувавши системи онлайн-бронювання кімнат у готельних комплексах було виділено наступні елементи, які необхідно реалізувати для автоматизованої системи управління готельними комплексами. А саме: особистий кабінет, реєстр готелей, перегляд детальної інформації про готелі та готельні номери, пошук готелей, резервування кімнат, геолокацію готелей, список улюблених готелей, рейтинг та відгуки готелів, історію резервувань, зіставлення послуг готелей, управління номерним фондом.

1.4 Аналіз сучасних засобів створення програмного забезпечення

Розробка та проектування програмного продукту, як правило розпочинається з вибору середовища (платформи) у якому буде працювати додаток. Основними платформами є десктоп, веб і мобільні пристрої.

Десктоп платформа є сукупністю технологій, призначених для управління ресурсами комп'ютера та організації взаємодії з користувачем [35]. Дана платформа як правило працює під такими операційними системами як Windows, Linux або MacOS і слугує площадкою, на якій розробляються програмні продукти для настільних/переносних комп'ютерів.

Переваги десктоп платформи: безпека, доступність (отримання користувачем певної інформації без Інтернету), швидкість роботи додатку.

Недоліки десктоп платформи: некомфортне оновлення додатку, відсутність доступу до інформації з будь-якої точки світу, дороговартісна розробка.

Веб-платформа є багатоцільовою платформою для розробки веб-проектів та керування вмістом. Платформа являє собою набір комплексних рішень, спрямованих на легку і успішну розробку веб-сайту і його підтримку [36]. Дана платформа на сьогодні вважається найбільш популярною за рахунок використання мережі Інтернет.

Переваги веб-платформи: доступ до інформації з будь-якого пристрою і в будь-якій точці світу, комфортне оновлення додатку, немає потреби в установці додатку, швидкість та дешевизна розробки.

Недоліки веб-платформи: залежність від мережі Інтернет, необхідна адаптивність додатків під різні браузері, пристрої; погана стійкість до хакерських атак.

Платформа мобільні пристрої являє собою програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Як правило програмні продукти які розробляються для даної платформи повинні бути сумісними з наступними операційними системами Android і IOS [37].

Сьогодні кожна людина має смартфон, або іншу переносну електроніку, тим самим дана платформа не поступається у популярності веб.

Переваги мобільної платформи: популярність смартфонів, комфортність у використанні додатку, часткова залежність від мережі Інтернет, швидкість.

Недоліки мобільної платформи: створення додатку під декілька операційних систем, дороговартісна підтримка додатку, погана стійкість до хакерських атак.

Визначивши платформу, на якій буде працювати програмний продукт, необхідно обрати програмну платформу, на якій буде розроблятися продукт. На сьогодні популярними програмними платформами є .Net, PHP, Java [38].

Платформа .Net є програмною технологією, запропонованою компанією Microsoft як платформа для створення як звичайних програм, так і веб-застосунків [39].

Переваги платформи .Net: кроссплатформеність (підтримка різних платформ), тисячі готових модулів для розробки додатку, безпека, швидкість, гнучкість, комфортність розробки високонавантажених додатків.

Недоліки платформи .Net: Потреба великої кількості оперативної пам'яті, потреба в ліцензії для доступу різноманітних модулів.

Платформа PHP є найпопулярнішою платформою для веб розробки завдяки своїй масштабованості, підтримці великого співтовариства та відкритим вихідним кодом. Платформа створена Расмусом Лердорфом. Більше 80% сайтів в світі написанні на даній платформі [40].

Переваги платформи PHP: швидкість та дешевизна розробки, гнучкість, безкоштовне розповсюдження, комфортна взаємодія з базами даних.

Недоліки платформи PHP: розробка лише під веб, безпека, некомфортна розробка високонавантажених додатків.

Платформа Java є рядом програмних продуктів і специфікацій створених компанією Sun Microsystems, сьогодні розвитком цієї платформи займається компанія Oracle, які спільно надають систему для розробки прикладного програмного забезпечення та вбудовування її в будь-яке крос-платформне

програмне забезпечення. Java використовується в різних комп'ютерних платформах від вбудованих пристроїв і мобільних телефонів в нижньому ціновому сегменті, до корпоративних серверів і суперкомп'ютерів у вищому ціновому сегменті [41].

Переваги платформи Java: кроссплатформеність, безпека, гнучкість, величезна кількість готових модулів.

Недоліки платформи Java: платне комерційне використання, низька продуктивність.

Таким чином, за результатом проведеного аналізу необхідно розробити веб-технологію для автоматизованої системи управління готельними комплексами.

1.5 Мета, задачі та вимоги до реалізації інформаційної системи

Метою кваліфікаційної роботи бакалавра є розробка вебтехнології для автоматизованої системи управління готельними комплексами.

Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- *розробити функціональну та інформаційну структуру системи управління готельними комплексами;*
- *розробити інформаційну модель: структуру БД та її об'єктно-реляційну проєкцію;*
- *розробити метод підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів;*
- *спроєктувати та розробити веб-технологію для автоматизованої системи управління готельними комплексами з можливістю підбору готельних номерів зі словесного опису побажань клієнтів.*

Розділ 2 Проктування інформаційної системи

2.1 Аналіз та автоматизація обробки потоків даних

2.1.1 Вебтехнологія для автоматизованої системи управління готельними комплексами

Розробка вебтехнології для автоматизованої системи управління готельними комплексами забезпечить процес автоматизації функціональних можливостей системи, а також спростить використання системи для користувачів.

В узагальненому вигляді вебтехнологію можна подати наступною послідовністю кроків:



Рисунок 2.1 – Схема роботи вебтехнології для автоматизованої системи управління готельними комплексами

Початковим етапом у функціонуванні технології є здійснення автентифікації користувача у системі, для цього використовується особова інформація: електронна пошта та пароль. Після здійснення автентифікації користувачу необхідно у відповідну форму пошуку готельних комплексів ввести наступні атрибути: локація, дата в'їзду та виїзду, кількість дорослих та дітей.

Наступним кроком є здійснення пошуку готельних номерів використовуючи вхідні атрибути. Система на основі атрибутів будує запит до бази даних інтеграційних сервісів. У разі успішного виконання запиту, система отримає у відповідь від сервісів та відобразить список готелів з наступною інформацією: назва, ціна за період проживання, кількість зірок, рейтинг, репрезентативна інформація, фотографії, список кімнат.

Наступний крок – підбір готельних номерів на базі побажань користувача. Користувач у відповідну форму вводить свої побажання використовуючи поставлені правила. Система в свою чергу за допомогою методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів здійснює аналіз вхідної інформації та за певним алгоритмом знаходить кімнати, які найбільш доцільно відповідають побажанням.

Останній крок – представлення користувачу список кімнат, які максимально задовольняють задані побажання.

Отже, представлено та описано вебтехнологію для автоматизованої системи управління готельними комплексами.

2.1.2 Метод підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів

У процесі експлуатації системи керування готельними комплексами доволі часто виникає наступна проблема: у клієнта під час процесу резервування кімнати виникає бажання: щоб з вікна майбутньої кімнати було видно гори або парк, кімната знаходилась на 2 поверсі, у наявності був кондиціонер та телевізор. Клієнт може зв'язатися з адміністратором готельного комплексу та вказати на дані

потреби. Адміністратору в свою чергу необхідно буде оглянути n-кімнат, щоб знайти ту кімнату, яка зможе задовольнити усі потреби клієнта. Весь процес займатиме величезну кількість часу. Для того, щоб вирішити дану проблему необхідно розробити метод, який автоматично зможе обрати кімнату на основі побажань клієнта, а також якщо кімната за деяких причин не доступна для проживання, запропонувати іншу враховуючи побажання клієнта.

Суть даного методу полягає у синтаксичному аналізі побажань клієнта, де за певним алгоритмом відшукуються відповідні кімнати, а далі на базі теорії ймовірностей визначається список номерів у низхідному порядку (тобто вверху списку буде кімната, яка найбільше підходить клієнту у відсотковому еквіваленті). Метод підбору готельних номерів можна подати наступним чином:

1. Лексичний аналіз вхідних даних;
2. Створення дерева розбору;
3. Створення та корегування запиту до бази даних;
4. Визначення найбільш відповідного готельного номеру.



Рисунок 2.2 – Схема роботи методу підбору готельних номерів

Опис функціонування методу

Першим етапом у функціонуванні даного методу є лексичний аналіз, за допомогою якого потік вхідних даних (текст) поділяється на значущі набори символів (токени), визначені граматиною регулярних виразів. Для прикладу користувач ввів наступний текст: «Я хотів би, щоб з вікна було видно *гори. Також є побажання: кімната повинна знаходитись на 2 поверсі, в кімнаті повинні бути присутні *кондиціонер&телевізор.» В даному тексті зустрічаються наступні спеціальні символи («*», «&», «<, >, >=, <=, =», «:», «,», «.»), ці символи є визначені граматиною регулярних виразів (усі правила використання спеціальних символів будуть завчасно відображенні користувачу). Кожен з цих символів має своє визначення та пріоритет:

- «*» - вказує на важливість слова. Має найвищий пріоритет (1);
- «&» - вказує на неявну важливість побажання. Має пріоритет (2);
- «<, >, >=, <=, =» - слугують для вказання діапазону цін готельних номерів. Мають пріоритет (3);
- «:» - вказує на те, що далі буде йти перерахування побажань (слів) клієнта. Має пріоритет (4);
- «,» - слугує для відокремлення набору побажань (має сенс лише у контексті з символом «:»). Має пріоритет (5);
- «.» - вказує на закінчення речення. Має найнижчий пріоритет (6).

Другий етап – це створення дерева розбору на базі створених токенів.

Дерево розбору є своєрідною структурою даних, мета якої полягає у відокремленні основних (суто важливих) частин тексту, які необхідні для роботи алгоритму та побудова запиту до бази даних.

Принцип побудова дерева є подібним до такої структури даних як бінарне дерево, в якому кожній вершині зіставлене певне значення. В даному випадку дерево будується на основі отриманих токенів, де кожен токен є окремою вершиною з певним зіставленим значенням.

Перевагами використання даної структури даних є можливість використання алгоритму центрованого обходу дерева, який передбачає відвідування (перевірку або модифікацію) кожної вершини дерева рівно один раз, тим саме дозволяє автоматизувати послідовну побудову запитів до бази даних. Також перевагою даної структури даних є можливість звернутися до будь-якої вершини дерева без зайвих проблем за рахунок подібності даної структури до принципу об'єктно-орієнтованого програмування «наслідування».

На рисунку 2.3 зображено структуру даних – Дерево розподілу на основі заданого прикладу.

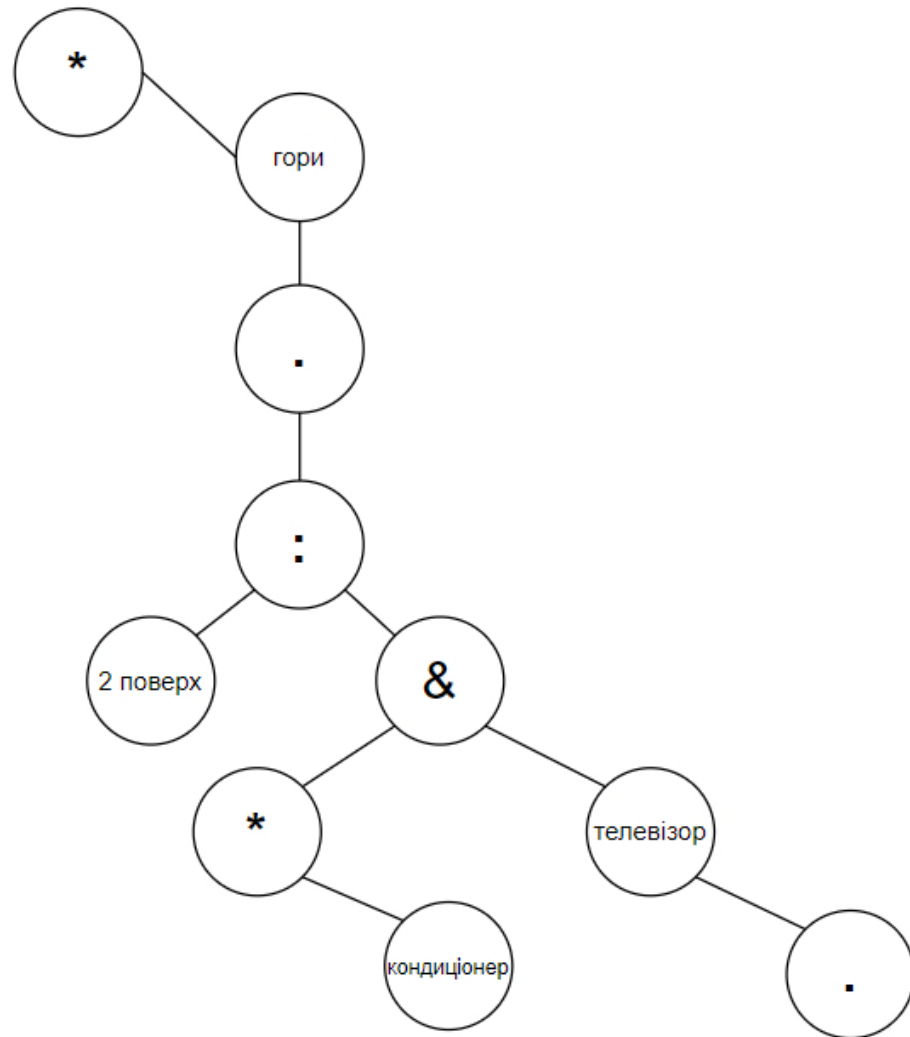


Рисунок 2.3 – Дерево розбору на основі заданого прикладу

Частина 2 є ключовою в даному етапі, так як від неї залежить логіка побудова запиту. Як раніше було зазначено дерево розбору складається з вершин, які в свою чергу представляють токени утворенні на початкову етапі функціонування алгоритму.

Кожний токен має своє значення, а також пріоритет. Опираючись на пріоритети створюється структура запиту. Чим вищий пріоритет у токена, тим точніше відбуватиметься побудова запиту з метою отримати кімнату, яка ідеально задовольнятиме потреби клієнта.

Третя частина пов'язана з побудовою та корегуванням запиту до бази даних. Запит у випадку даного алгоритму будується сам за рахунок команд, які вже

раніше були записані у файл. Тобто принцип побудову запиту полягає в наступному: в залежності від пріоритету токена підставляється певний набір команд, які вже раніше були записані у файл. І так як для створення автоматизованої системи було зазначено використання технології ORM, то набір команд об'єднуються в одну-дві команди з легким та зрозумілим синтаксисом.

Четвертим заключним етапом є визначення найбільш відповідного готельного номеру серед набору отриманих кімнат відповідно до побажань клієнта. Необхідність визначення найбільш відповідної кімнати виявляється у тому випадку коли кімната, яка ідеально підходить по всім параметрам за деяких причин стає не доступною для проживання і тому необхідно клієнту запропонувати певну альтернативу.

Алгоритм визначення найбільш відповідної кімнати серед набору кімнат базується на теорії ймовірностей. Використання принципів та алгоритмів теорії ймовірностей дозволить легко створити алгоритм обрахунку вибірки, а також забезпечити мінімальну затрату продуктивності автоматизованої системи. Для обрахування вибірки було обрано принцип «розподіл Гумбеля».

Розподіл Гумбеля (подвійний експоненціальний розподіл) є граничним розподілом, який використовується для визначення найбільшого елемента вибірки випадкової величини. Даний розподіл ідеально підходить вирішення поставленої задачі. Розподіл Гумбеля розраховується за наступною формулою:

$$f(x) = \frac{1}{\beta} \exp \left[\frac{\alpha - x}{\beta} - \exp \left(\frac{\alpha - x}{\beta} \right) \right]$$

Де α і β є параметрами, які в граничному випадку наближено визначаються за формулами:

$$\alpha = \frac{P_2 + 0,4 * n}{P_1 + \ln(k) * 0,6} * 0,1$$

Де P_2 – ціна альтернативної кімнати з врахуванням періоду майбутнього проживання клієнта;

P_1 – ціна ідеальної кімнати, яка задовольняє всі побажання, з врахуванням період майбутнього проживання клієнта;

k – кількість обов'язкових побажань клієнта (опирається на токени з пріоритетом 1);

n – загальна кількість побажань без врахування обов'язкових (опирається на токени з пріоритетом більше 1);

Коефіцієнти 0,4; 0,6; 0,1 – є стандартно заданими у розподілі Гумбеля.

$$\beta = \frac{k * 0,4}{n * 0,6}$$

k – кількість обов'язкових побажань клієнта (опирається на токени з пріоритетом 1);

n – загальна кількість побажань без врахування обов'язкових (опирається на токени з пріоритетом більше 1);

Коефіцієнти 0,4; 0,6; 0,1 – є стандартно заданими у розподілі Гумбеля.

Параметр α – слугує для визначення коефіцієнта відповідності, β – слугує для корегування коефіцієнта відповідності.

Параметр x – коефіцієнт шуканого максимального значення випадкової величини. Визначається за наступною формулою:

$$x = \alpha - \beta * \ln(-\ln(F))$$

Де α – коефіцієнт відповідності;

β – коефіцієнт коригування відповідності;

F – інтегральна функція розподілу Гумбеля, визначається за формулою:

$$F = \exp \left[-\exp \left(\frac{\alpha}{\beta} \right) \right]$$

Обрахувавши параметри кожної кімнати за даним розподілом, буде отримано впорядкований низхідний набір кімнат, де кожна кімната має своє значення від 0 до 1. Якщо кімната, яка найкраще підходила по заданим параметрам за певних причин буде не доступною, система зможе запропонувати альтернативу з даного набору. Для більш наглядного розуміння, у якості прикладу на рисунку 2.4 представлено діаграму, яка відображає кімнати та їх значення відповідності після розрахунків. Опираючись на діаграму, видно, що найкращою альтернативою ідеальної кімнати може стати кімната №4, так як має найвище значення відповідності.

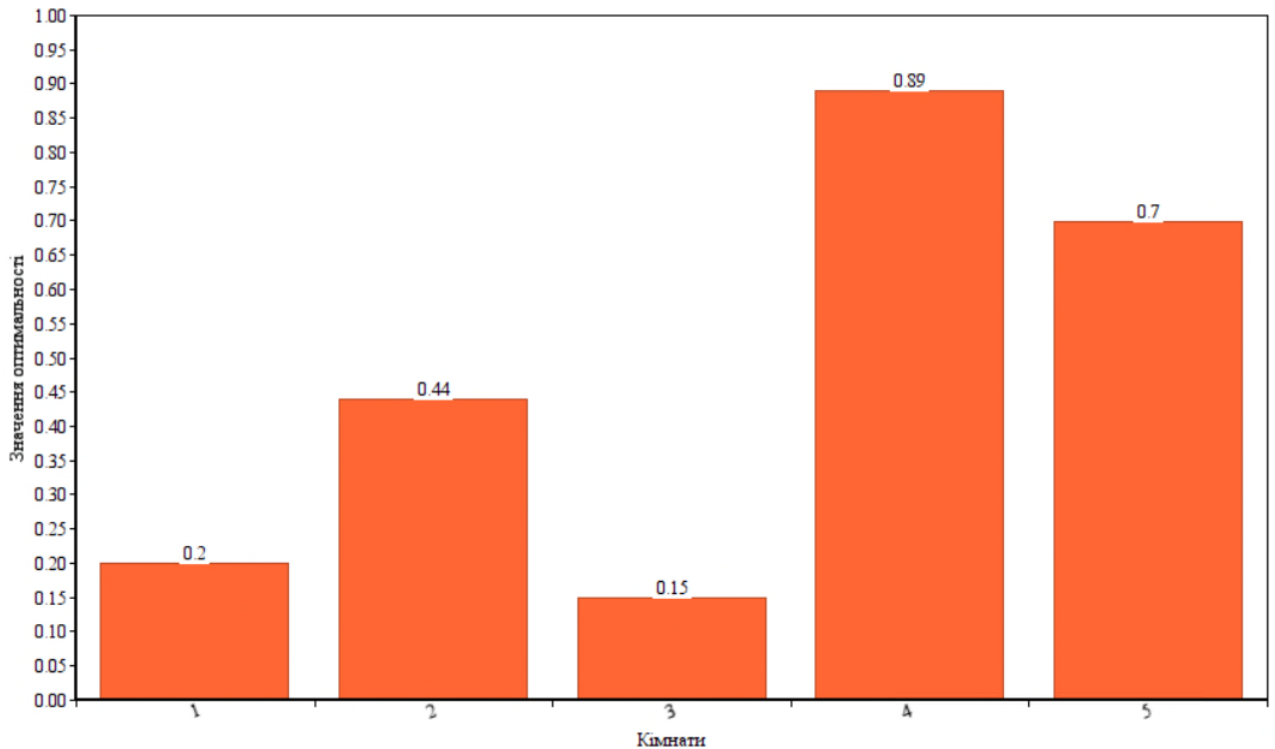


Рисунок 2.4 – Діаграма відповідності кімнат

Отже, створення методу підбору готельного номеру на базі синтаксичного аналізу побажань клієнта при бронюванні дозволить отримати набір номерів, які максимально задовольняють побажання клієнтів.

2.1.3 Функціональна структура інформаційної системи

Автоматизована система управління готельними комплексами передбачає розділення користувачів системи на три окремі ролі: клієнт, адміністратор та відвідувач. Для кожної ролі необхідно автоматизувати відповідний функціонал. Автоматизація функціоналу забезпечить комфортну, гнучку та різноманітну взаємодію з системою для кожного виду користувачів.

Процес автоматизації, як правило базується на розробці груп функцій, які відповідають функціональним частинам системи. Групи функцій, які необхідно розробити для кожного виду користувачів представлено на рисунку 2.5.

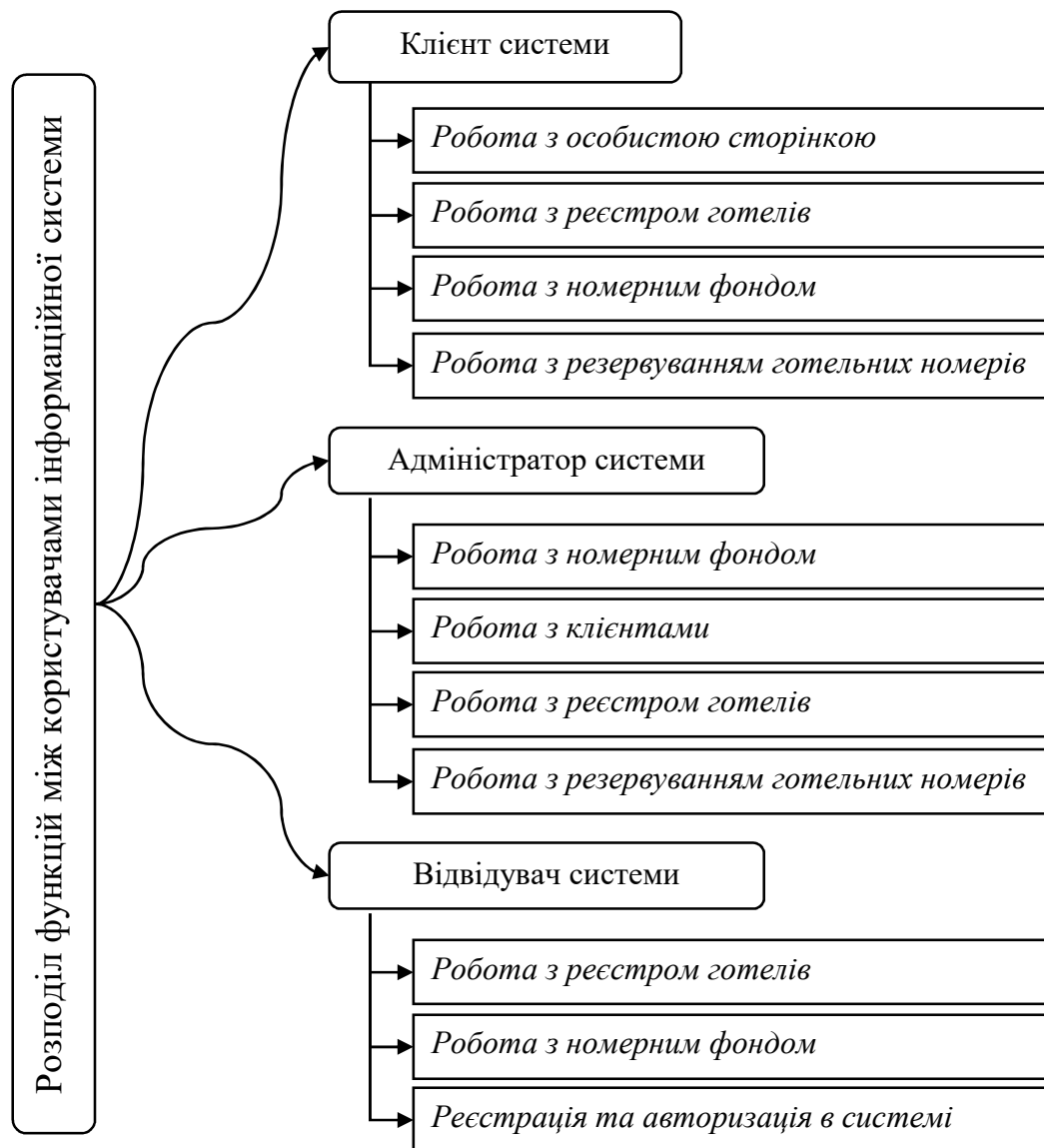


Рисунок 2.5 – Розподіл функцій між користувачами автоматизованої системи

Відповідно до розподілу функцій між клієнтом, адміністратором та відвідувачем, необхідно автоматизувати відповідні бізнес-процеси, бізнес-процеси представлено в таблиці 2.1.

Таблиця 2.1 – Бізнес-процеси

Номер бізнес-процесу	Назва бізнес-процесу
1	Робота клієнта з особистою сторінкою
2	Робота клієнта з реєстром готелів
3	Робота клієнта з номерним фондом готелів
4	Робота клієнта з резервуванням готельних номерів
5	Робота адміністратора з реєстром готелів
6	Робота адміністратора з управлінням номерним фондом

7	Робота адміністратора з особовою інформацією клієнтів
8	Робота адміністратора з резервуванням готельних номерів
9	Робота відвідувача з реєстром готелів
10	Робота відвідувача з номерним фондом готелів
11	Робота відвідувача з реєстрацією та авторизацією в системі

Бізнес-процес «Робота клієнта з особистою сторінкою»

Бізнес-процес надає можливість клієнту взаємодіяти (маніпулювати) зі своєю особистою інформацією, а також забезпечує можливість взаємодіяти з активними резервуваннями та списком улюблених готелів:

1. Додавлення особистої інформації: прізвище, ім'я, по батькові, дата народження, фотографія, номер телефону, стать.
2. Редагування особистої інформації: прізвище, ім'я, по батькові, дата народження, фотографія, номер телефону, стать.
3. Перегляд списку улюблених готелів: назва готелю, адрес, кількість зірок, фотографія, ціна (опирається на ціну кімнати за одну ніч), рейтинг.
4. Видалення готелю із списку улюблених.
5. Перегляд активного резервування кімнати: назва готелю, адреса готелю, номер кімнати, назва та тип кімнати, кількість ліжок, кількість дорослих, кількість дітей, дата прибуття і відбуття, ціна.
6. Відміна активного резервування.

Діаграма дій бізнес-процесу «Робота клієнта з особистою сторінкою» зображена на рисунку 2.6.

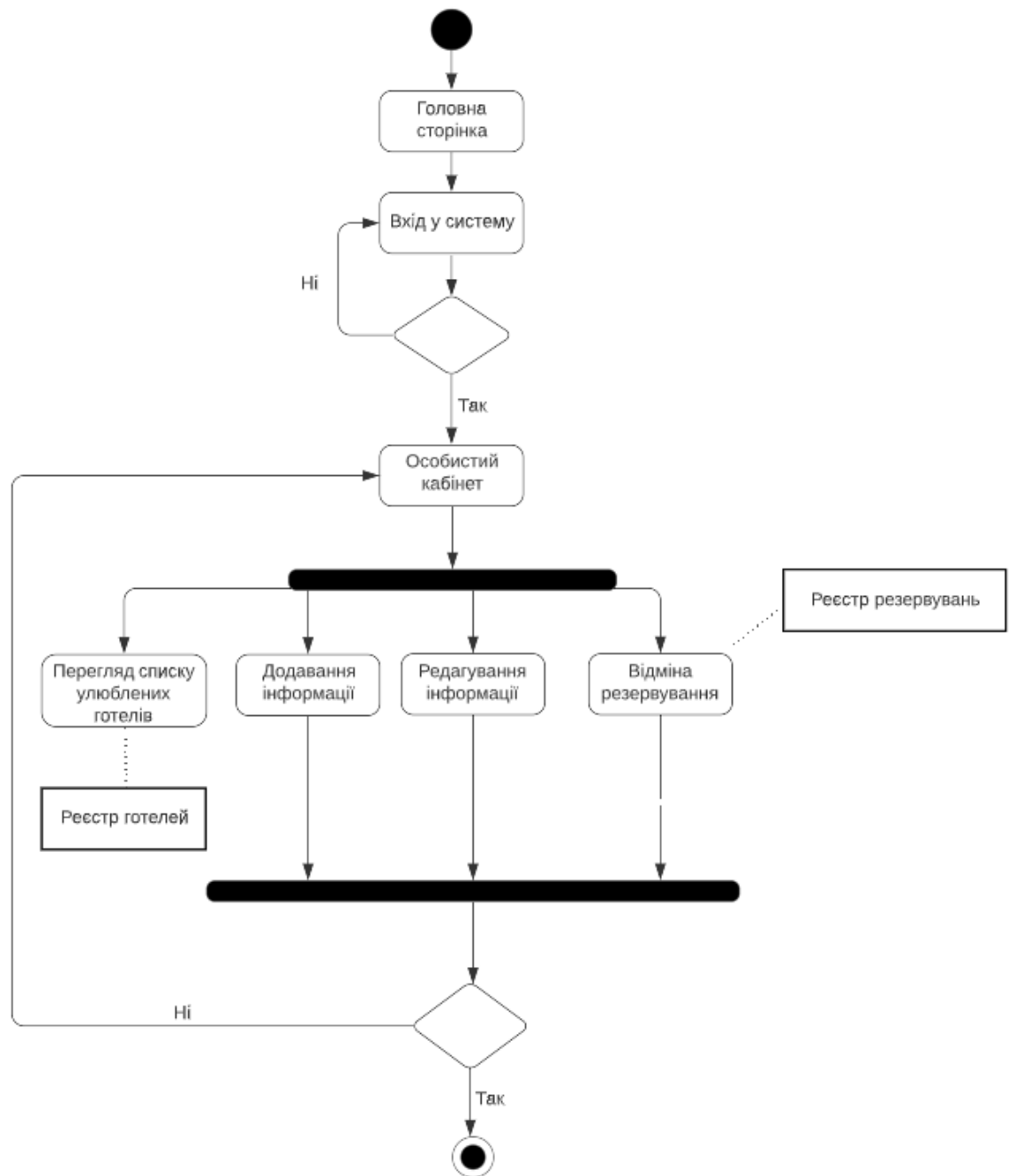


Рисунок 2.6 – Діаграма дій бізнес-процесу «Робота клієнта з особистою сторінкою»

Бізнес-процес «Робота клієнта з реєстром готелів»

Бізнес-процес надає можливість клієнту взаємодіяти з реєстром готелів, а також забезпечує можливість гнучкого пошуку готелів та взаємодію з списком улюблених готелів:

1. Перегляд реєстру готелів (перегляд репрезентативної інформації готелю): назва, адрес, кількість зірок, фотографія, ціна (добуток ціни за одну ніч і періоду перебування клієнта), рейтинг.
2. Пошук готелів з використанням фільтрів: локація, дата прибуття, дата відбуття, кількість дорослих людей, кількість дітей, кількість зірок, рейтинг.
3. Додавлення певного готелю до списку улюблених.

Діаграма дій бізнес-процесу «Робота клієнта з реєстром готелів» зображена на рисунку 2.7.

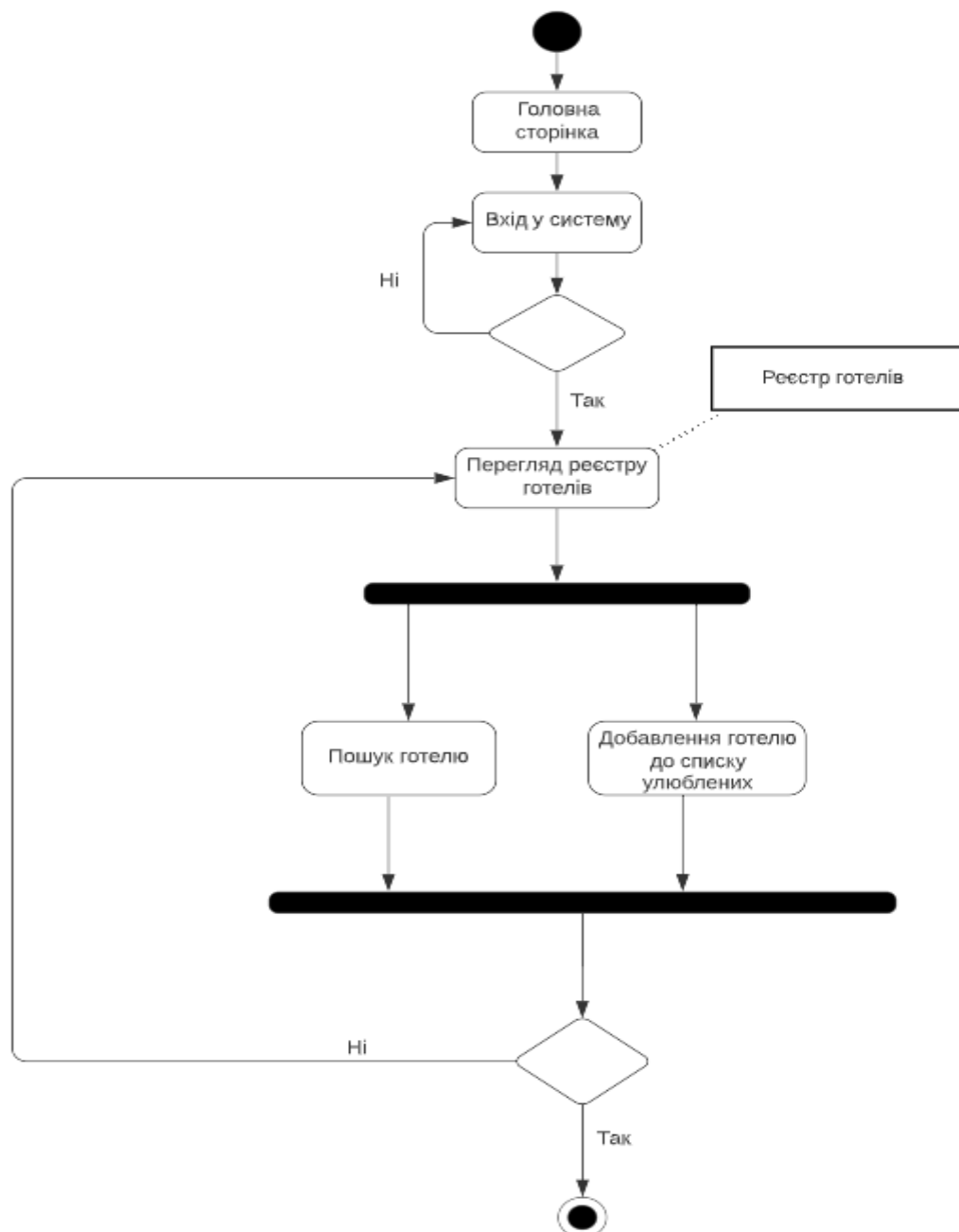


Рисунок 2.7 – Діаграма дій бізнес-процесу «Робота клієнта з реєстром готелів»

Бізнес-процес «Робота клієнта з номерним фондом готелів»

Бізнес-процес надає можливість клієнту взаємодіяти з детальною інформацією відповідного готелю та кімнат, а також забезпечує можливість написання та перегляду коментарів, виставлення рейтингу:

1. Перегляд детальної інформації певного готелю: назва, адрес, фотографії, розгорнута інформація про діяльність готелю, розташування готелю на онлайн-карті, погода до відповідного місцезнаходження готелю.

2. Перегляд репрезентативної інформації кімнат: назва, тип кімнат, кількість ліжок, ціна (добуток ціни за одну ніч і періоду перебування клієнта), харчування, фотографії, кількість дорослих, кількість дітей.

3. Перегляд коментарів клієнтів: тема коментарю, опис, ім'я та фотографія клієнта.

4. Додавання коментарю: тема коментаря, опис, ім'я та фото.

5. Виставлення рейтингу готелю: оцінка сервісу готелю від 0 до 10.

Діаграма дій бізнес-процесу «Робота клієнта з номерним фондом готелів» зображена на рисунку 2.8.

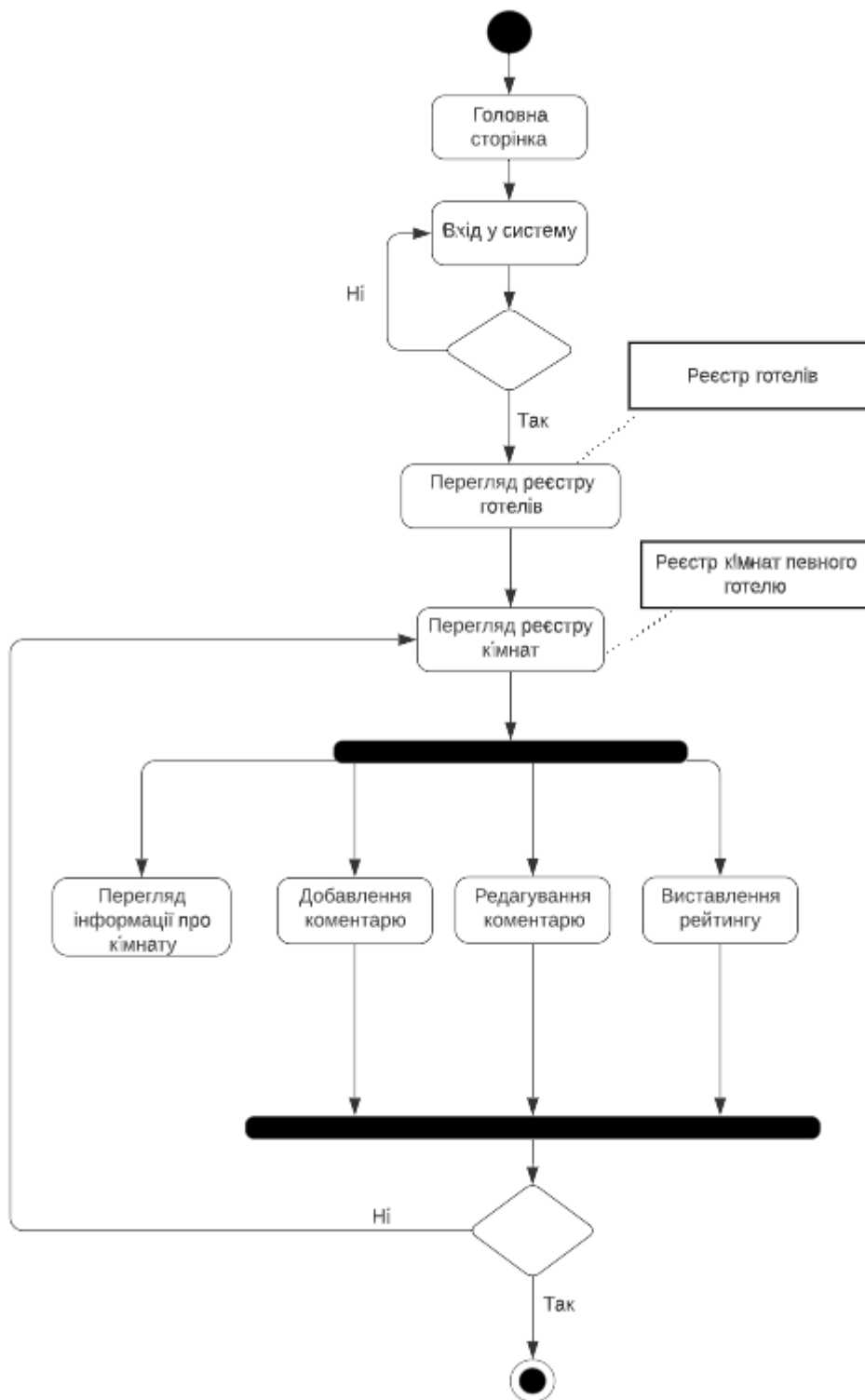


Рисунок 2.8 – Діаграма дій бізнес-процесу «Робота клієнта з номерним фондом готелів»

Бізнес-процес «Робота клієнта з резервуванням готельних номерів»

Бізнес-процес надає можливість клієнту здійснити бронювання обраної кімнати відповідного готелю:

1. Додавання та редагування особистої інформації стосовно резервування: прізвище, ім'я, по батькові, дата народження, стать, номер телефону.

2. Додавання та редагування інформації по резервуванню: назва готелю, адреса готелю, назва та тип кімнати, кількість дорослих, кількість дітей, дата прибуття і відбуття, ціна.

3. Створення резервування.

Діаграма дій бізнес-процесу «Робота клієнта з резервуванням готельних номерів» зображена на рисунку 2.9.

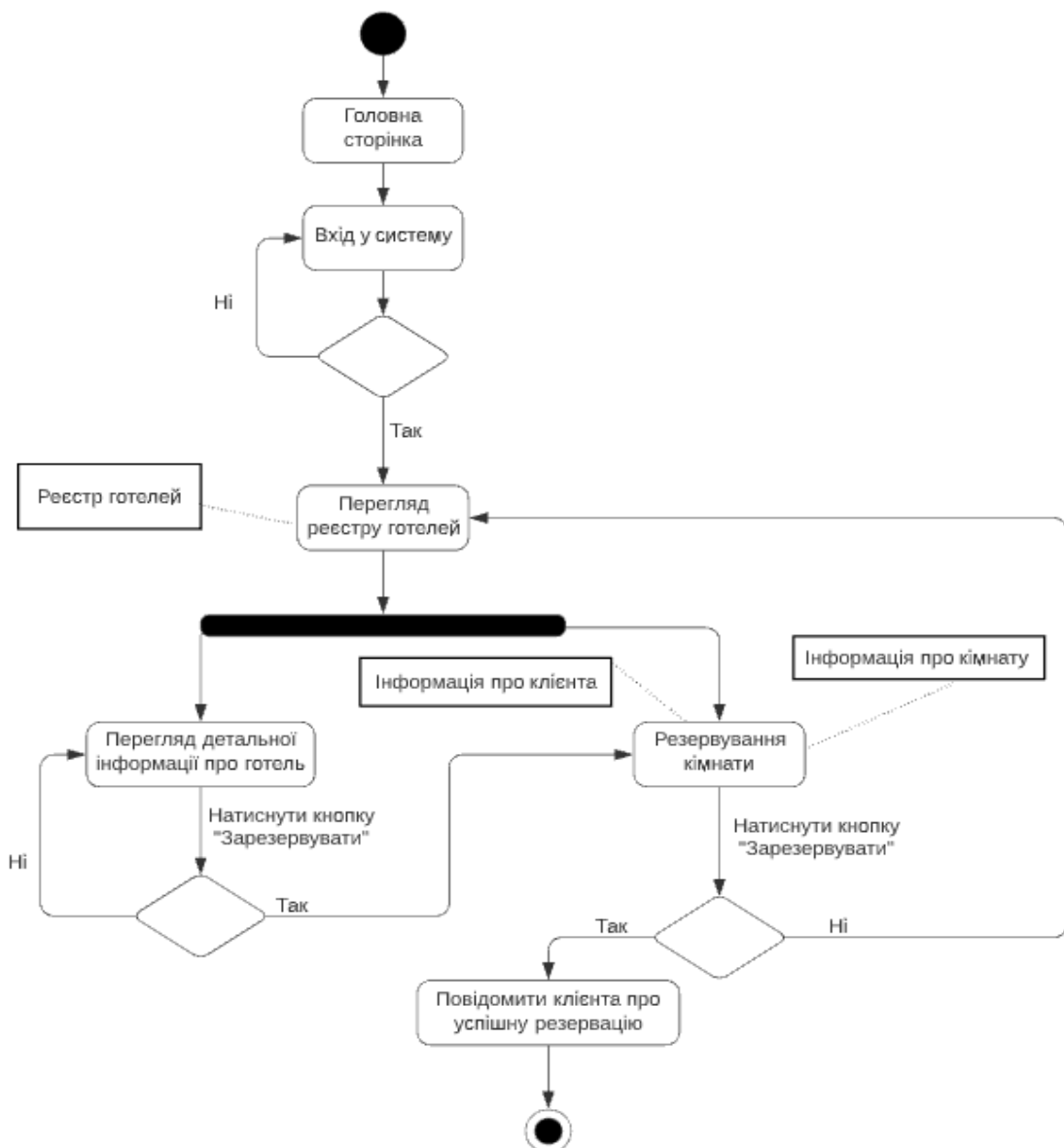


Рисунок 2.9 – Діаграма дій бізнес-процесу «Робота клієнта з резервуванням готельних номерів»

Бізнес-процес «Робота адміністратора з реєстром готелів»

Бізнес-процес надає можливість адміністратору взаємодіяти (маніпулювати) з реєстром готелів та їх репрезентативною інформацією:

1. Перегляд реєстру (списку) готелей (перегляд репрезентативної інформації готелю): назва, адрес, кількість зірок, фотографія, ціна (опирається на ціну кімнати за одну ніч), рейтинг.
2. Додавання інформації про новий готель: назва, адрес, фотографії, кількість зірок, рейтинг, інформація про діяльність готелю, розташування на онлайн-карті.
3. Редагування інформації відповідного готеля: назва, адрес, фотографії, кількість зірок, рейтинг, інформація про діяльність готелю, розташування на онлайн-карті.
4. Видалення готелю із реєстру.

Діаграма дій бізнес-процесу «Робота адміністратора з реєстром готелів» зображена на рисунку 2.10.

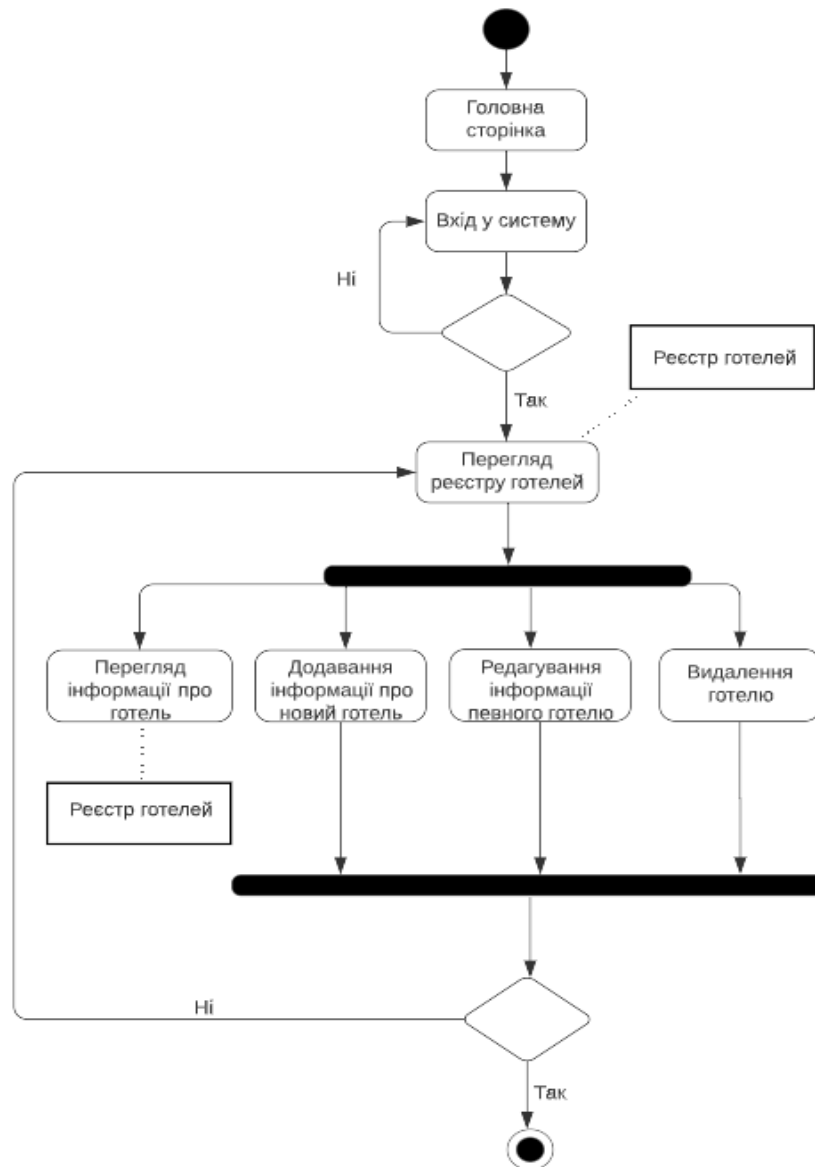


Рисунок 2.10 – Діаграма дій бізнес-процесу «Робота адміністратора з реєстром готелів»

Бізнес-процес «Робота адміністратора з управлінням номерним фондом готелів»

Бізнес-процес надає можливість адміністратору взаємодіяти з реєстром кімнат відповідного готелю та їх детальною інформацією:

1. Перегляд реєстру (списку) кімнат відповідного готелю: номер, назва, тип кімнати, ціна за одну ніч.

2. Перегляд детальної інформації кімнати: номер, назва, тип кімнати, кількість ліжок, опис, ціна за одну ніч, фотографії, харчування, кількість дорослих, кількість дітей.

3. Додавання інформації про нову кімнату: номер, назва, тип кімнати, кількість ліжок, опис, ціна за одну ніч, фотографії, харчування, кількість дорослих, кількість дітей.

4. Редагування інформації кімнати: номер, назва, тип кімнати, кількість ліжок, опис, ціна за одну ніч, фотографії, харчування, кількість дорослих, кількість дітей.

5. Видалення кімнати.

Діаграма дій бізнес-процесу «Робота адміністратора з управління номерним фондом готелів» зображена на рисунку 2.11.

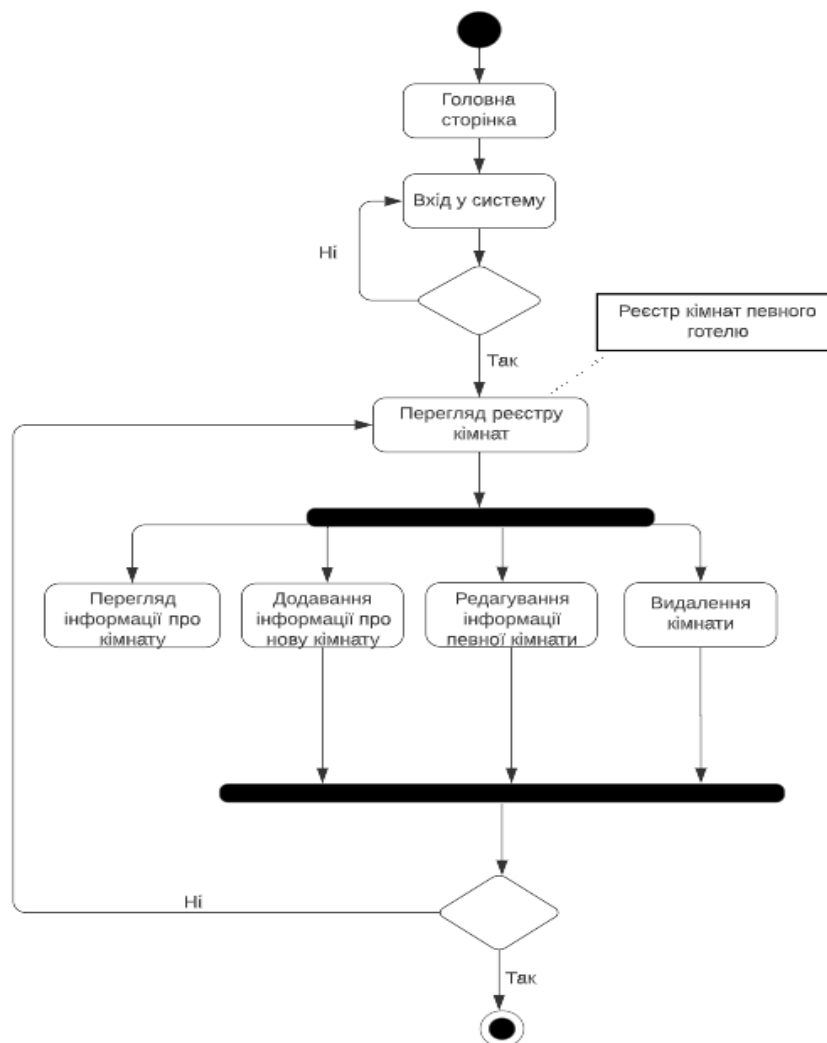


Рисунок 2.11 – Діаграма дій бізнес-процесу «Робота адміністратора з управління номерним фондом готелів»

Бізнес-процес «Робота адміністратора з особою інформацією клієнтів»

Бізнес-процес надає можливість адміністратору взаємодіяти з реєстром клієнтів та їх особистою інформацією:

1. Перегляд реєстру (списку) клієнтів (перегляд репрезентативної інформації клієнтів): прізвище, ім'я, по батькові, дата народження, стать, фотографія.
2. Перегляд детальної інформації клієнта: прізвище, ім'я, по батькові, дата народження, фотографія, номер телефону, стать, електронний адрес, пароль.
3. Додавання інформації про нового клієнта: прізвище, ім'я, по батькові, дата народження, фотографія, номер телефону, стать, електронний адрес, пароль.
4. Редагування інформації клієнта: прізвище, ім'я, по батькові, дата народження, фотографія, номер телефону, стать, електронний адрес, пароль.
5. Видалення клієнта із реєстру.

Бізнес-процес «Робота адміністратора з резервуванням готельних номерів»

Бізнес-процес надає можливість адміністратору взаємодіяти з реєстром резервувань та детальною інформацією по бронюванню:

1. Перегляд реєстру (списку) резервувань: назва готелю, номер та назва кімнати, дата прибуття, дата відбуття, ціна, позначення про підтвердження резервування.
2. Перегляд детальної інформації по резервуванню: особова інформація клієнта (прізвище, ім'я, по батькові, дата народження, стать, номер телефону), назва готелю, адрес готелю, номер кімнати, назва кімнати, тип кімнати, кількість ліжок, кількість дорослих, кількість дітей, харчування, дата прибуття, дата відбуття, ціна, позначення про підтвердження резервування.
3. Редагування резервування: особова інформація клієнта (прізвище, ім'я, по батькові, дата народження, стать, номер телефону), назва готелю, адрес готелю, номер кімнати, назва кімнати, тип кімнати, кількість ліжок, кількість

дорослих, кількість дітей, харчування, дата прибуття, дата відбуття, ціна, позначення про підтвердження резервування.

4. Підтвердження резервування.
5. Видалення резервування.

Діаграма дій бізнес-процесу «Робота адміністратора з резервуванням готельних номерів» зображена на рисунку 2.12.

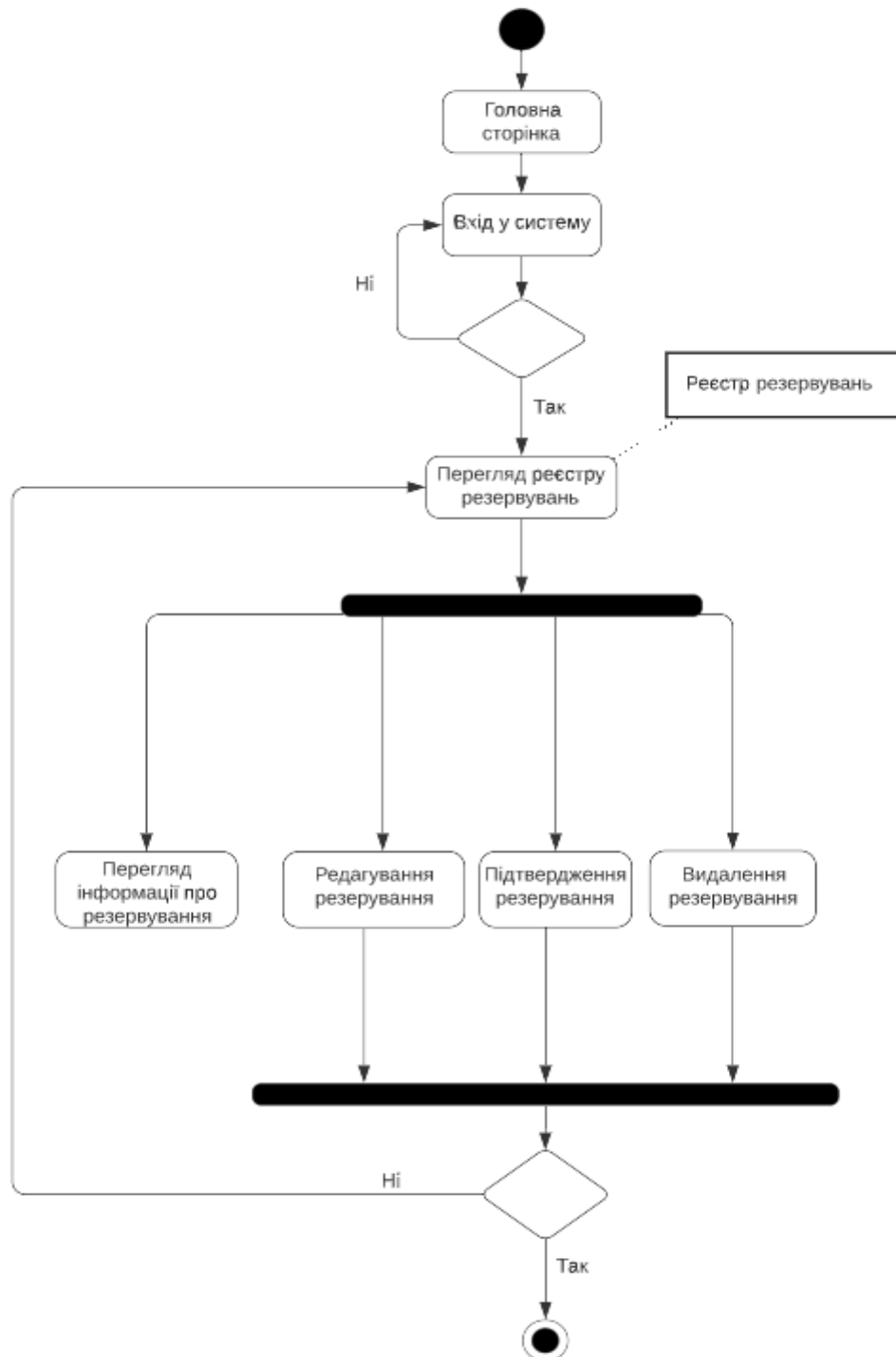


Рисунок 2.12 – Діаграма дій бізнес-процесу «Робота адміністратора з резервуванням готельних номерів»

Бізнес-процес «Робота відвідувача з реєстром готелів»

Бізнес-процес надає можливість відвідувачу взаємодіяти з реєстром готелів, а також забезпечує можливість гнучкого пошуку готелів:

1. Перегляд реєстру готелів (перегляд репрезентативної інформації готелю): назва, адрес, кількість зірок, фотографія, ціна (добуток ціни за одну ніч і періоду перебування), рейтинг.
2. Пошук готелів з використанням фільтрів: локація, дата прибуття, дата відбуття, кількість дорослих людей, кількість дітей.

Бізнес-процес «Робота відвідувача з номерним фондом готелів»

Бізнес-процес надає можливість відвідувачу взаємодіяти з детальною інформацією про відповідний готель та кімнати, а також забезпечує можливість перегляду коментарів та рейтингу:

1. Перегляд детальної інформації про певний готель: назва, адрес, фотографії, розгорнута інформація про діяльність готелю, розташування готелю на онлайн-карті, погода до відповідного місцезнаходження готелю.
2. Перегляд репрезентативної інформації кімнат: назва, тип кімнат, кількість ліжок, ціна (добуток ціни за одну ніч і періоду перебування), харчування, фотографії, кількість дорослих, кількість дітей.
3. Перегляд коментарів.

Діаграма дій бізнес-процесу «Робота відвідувача з номерним фондом готелів» зображена на рисунку 2.13.

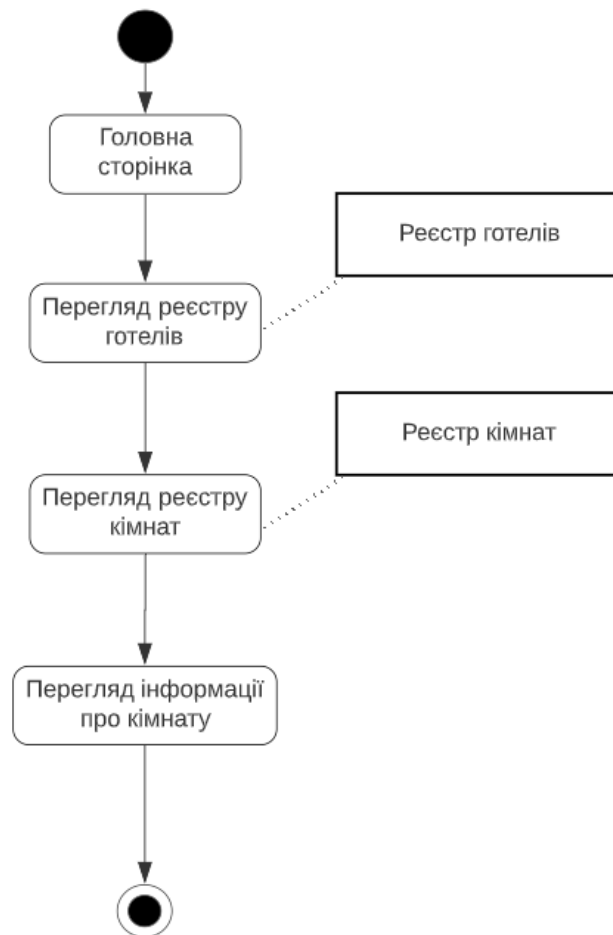


Рисунок 2.13 – Діаграма дій бізнес-процесу «Робота відвідувача з номерним фондом готелів»

Бізнес-процес «Робота відвідувача з реєстрацією та авторизацією в системі»

Бізнес-процес надає можливість відвідувачу зареєструватись та авторизуватись в системі:

1. Створення акаунту клієнта: електронний адрес, пароль.
2. Вхід у систему для зареєстрованих клієнтів та адміністраторів: електронний адрес, пароль.

Діаграма дій бізнес-процесу «Робота відвідувача з реєстрацією та авторизацією в системі» зображена на рисунку 2.14.

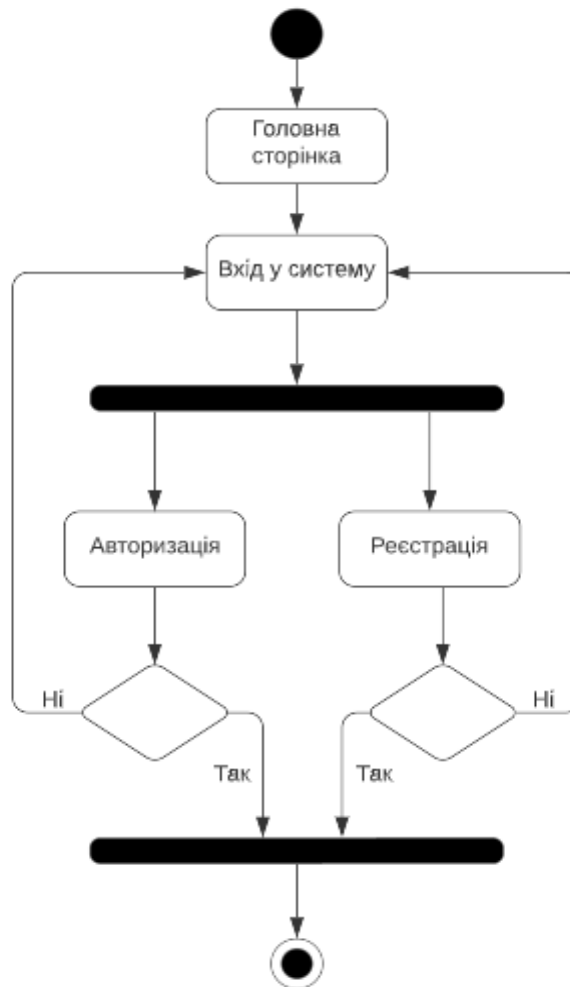


Рисунок 2.14 – Діаграма дій бізнес-процесу «Робота відвідувача з реєстрацією та авторизацією в системі»

Зважаючи на наведені бізнес-процеси, можна розробити структуру вебтехнології, а також відповідної бази даних, призначеної для збереження даних про клієнтів, готелі, готельні номери, резервування клієнтів, готелі в яких клієнт бажає у майбутньому зупинитися.

2.2 Інформаційна структура системи

2.2.1 Проектна архітектура системи та взаємозв'язок компонентів

Для забезпечення автоматизації функціональних можливостей системи управління готельними комплексами запропоновано наступну проектну архітектуру (рис. 2.15):

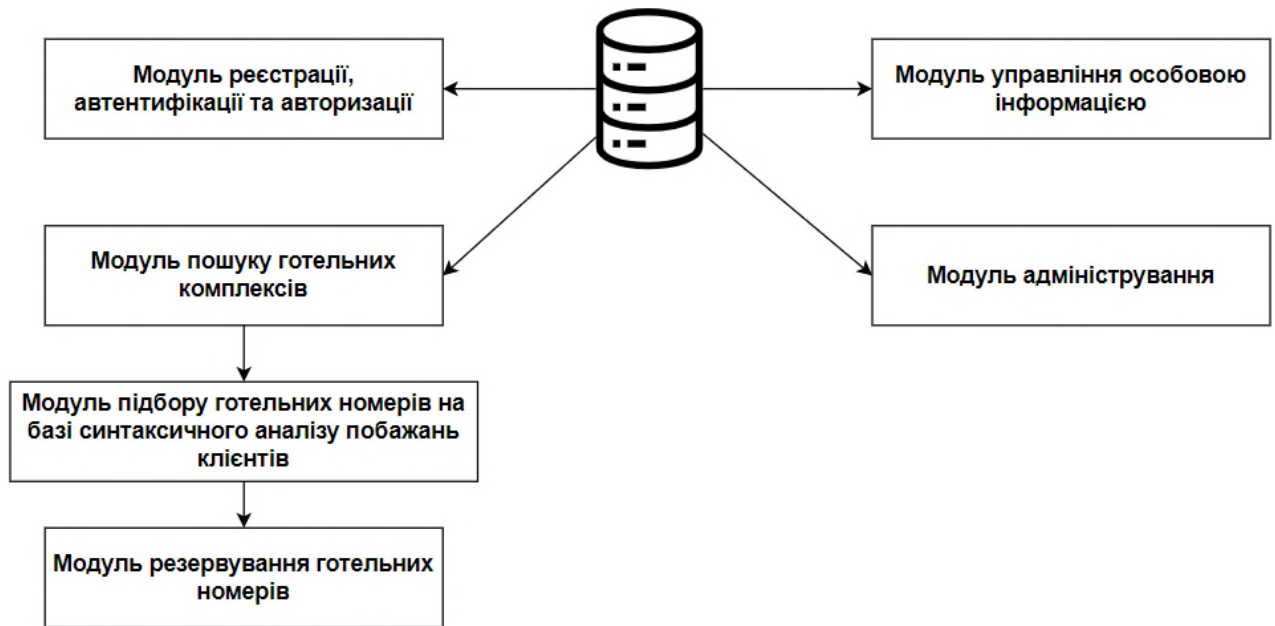


Рисунок 2.15 – Проектна архітектура системи управління готельними комплексами

Архітектура системи складається з наступних компонентів:

- *Модуль реєстрації, автентифікації та авторизації* – забезпечує можливість реєстрації та автентифікації користувача у системі. Також відповідає за здійснення авторизації, тобто за надання доступу користувачам до відповідних функціональних можливостей системи;

- *Модуль пошуку готельних комплексів* – забезпечує пошук готельних комплексів;

- *Модуль управління особовою інформацією* – надає можливість користувачам керувати (додавати, редагувати, видаляти) своєю особовою інформацією;

- Модуль резервування готельних номерів – слугує для здійснення бронювання кімнат;
- Модуль підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів – здійснює підбір готельних номерів аналізуючи побажання клієнтів (текстова інформація);
- Модуль адміністрування – надає можливість адміністраторам керувати (додавати, редагувати, видаляти) інформацією клієнтів, готельних комплексів та кімнат, резервувань.

Отже, представлена проектна архітектура системи забезпечить процес автоматизації функціональних компонентів системи.

2.2.2 Інформаційна модель

У ході розробки автоматизованої системи управління готельними комплексами необхідно розробити та описати структуру бази даних, а також визначити основні сутності предметної області та їх властивості. На рисунку 2.16 зображена даталогічна модель бази даних автоматизованої системи управління готельними комплексами, яка передбачає такі таблиці: англ: Users, Sexes, Countries, Hotels, Cities, Stars, Rates, Hotel_images, Comments, Roles, Rooms, Favourite_hotels, Periods, Reservations.

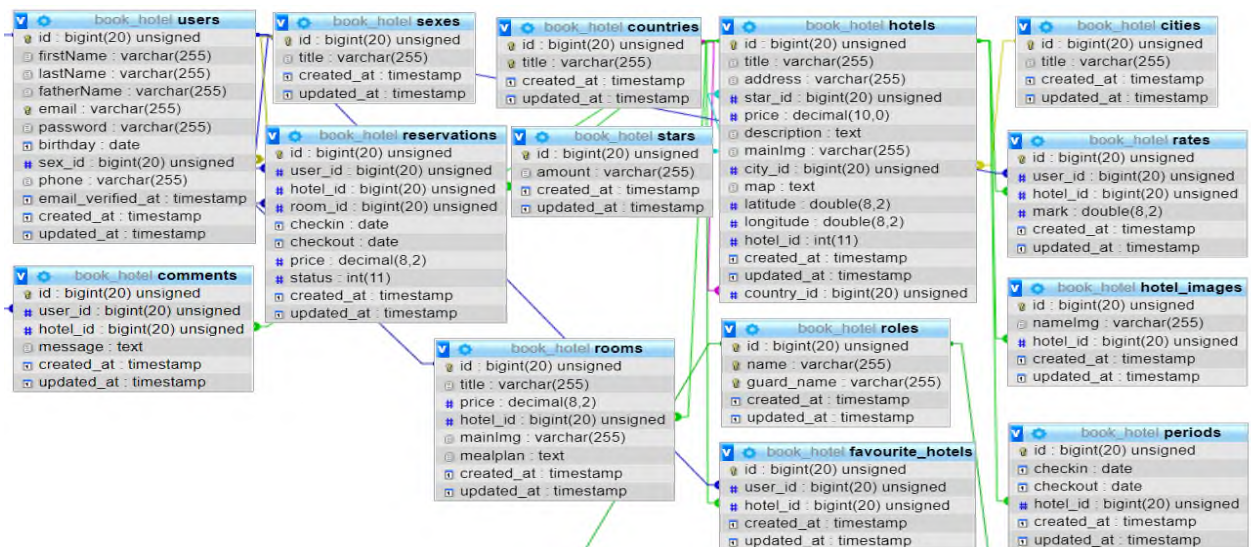


Рисунок 2.16 – Даталогічна модель бази даних автоматизованої системи управління готельними комплексами

Кожна з таблиць БД є необхідною для збереження даних окремої сутності. Властивості сутностей є атрибутами таблиць. Більш детальний опис таблиць та їх атрибутів наведено у додатку «А».

Таблиці бази даних, як правило взаємодіють з конкретною групою функцій:

1. Для реалізації бізнес-процесу «Робота клієнта з особистою сторінкою» будуть застосовуватись таблиці «Users», «Favourite_hotels», «Reservations»;

2. Для реалізації бізнес-процесу «Робота клієнта з реєстром готелів» будуть застосовуватись таблиці «Hotels», «Favourite_hotels», «Countries», «Cities», «Periods»;

3. Для реалізації бізнес-процесу «Робота клієнта з номерним фондом готелів» будуть застосовуватись таблиці «Hotels», «Rooms», «Countries», «Cities», «Hotel_images»;

4. Для реалізації бізнес-процесу «Робота клієнта з резервуванням готельних номерів» будуть застосовуватись таблиці «Reservations», «Users», «Hotels», «Rooms», «Countries», «Cities»;

5. Для реалізації бізнес-процесу «Робота адміністратора з реєстром готелів» будуть застосовуватись таблиці «Hotels», «Countries», «Cities»;

6. Для реалізації бізнес-процесу «Робота адміністратора з управлінням номерним фондом» будуть застосовуватись таблиці «Rooms», «Hotels», «Hotel_images», «Countries», «Cities»;

7. Для реалізації бізнес-процесу «Робота адміністратора з особовою інформацією клієнтів» буде застосовуватись таблиця «Users»;

8. Для реалізації бізнес-процесу «Робота адміністратора з резервуванням готельних номерів» будуть застосовуватись таблиці «Users», «Rooms», «Hotels», «Hotel_images», «Countries», «Cities»;

9. Для реалізації бізнес-процесу «Робота відвідувача з реєстром готелів» будуть застосовуватись таблиці «Hotels», «Countries», «Cities», «Periods»;

10. Для реалізації бізнес-процесу «Робота відвідувача з номерним фондом готелів» будуть застосовуватись «Hotels», «Rooms», «Countries», «Cities», «Hotel_images»;

11. Для реалізації бізнес-процесу «Робота відвідувача з реєстрацією та авторизацією в системі» буде застосовуватись таблиця «Users».

Таким чином, представлена база даних підходить для функціонування автоматизованої системи управління готельними комплексами. Далі необхідно обрати засоби для розробки даної системи.

2.3 Вибір засобів розробки інформаційної системи

Для забезпечення швидкої та гнучкої розробки автоматизованої системи управління готельними комплексами було обрано технологію, яка надасть широкий спектр інструментів для розробки та сформує загальну функціональну структуру системи. Даною технологією виступає програмний фреймворк Laravel.

Laravel є безкоштовним РНР фреймворком з відкритим кодом, який надає набір інструментів та ресурсів для створення сучасних веб-додатків. Даний фреймворк заснований на базі паттерна Model-View-Controller. Головні його особливості, це простота, елегантність, та хороша документація. Також він звільняє від масивного і заплутаного коду, допомагає створювати багатофункціональні веб-додатки, використовуючи простий і виразний синтаксис.

Фреймворк Laravel надає потужні інструменти для взаємодії з базами даних, включаючи технологію ORM (Object Relational Mapper) під назвою Eloquent, а також вбудовані механізми для створення міграцій бази даних. Даний програмний каркас має у своєму арсеналі вбудовані інструменти реєстрації, автентифікації та авторизації. За допомогою інструмента командного рядка Artisan розробники можуть створювати нові моделі, контролери та інші компоненти програми, що в свою чергу прискорює загальну розробку програми [42]. Основною перевагою Laravel є шаблонізатор Blade. Blade – простий, але

потужний шаблонізатор, що входить до складу Laravel. На відміну від інших шаблонізаторів, він не обмежує вас у використанні конструкцій PHP всередині шаблонів. Шаблони Blade компілюються в PHP-код і кешуються фреймворком - Blade не вносить додаткових гальм в роботу фреймворка [43].

Переваги Laravel:

- вбудований шаблонізатор Blade;
- технологія ORM Eloquent для взаємодії з базами даних;
- комфортна система міграцій;
- вбудовані системи реєстрації, автентифікації та авторизації;
- гнучкість (підтримка різноманітних сторонніх бібліотек, плагінів, програмних пакетів та сервісів);
- надійний захист у роботі з СКБД;
- швидкість у розробці програмних продуктів.

До недоліків фреймворку Laravel можна віднести:

– більшість функціоналу працює через фасади і інтегровані середовища розробки не розпізнають методів і властивостей в деяких класах.

Laravel має достатньо широку екосистему функціональних можливостей. Особливу увагу можна приділити взаємодії фреймворку з різними позасистемними пакетами та інтеграційними сервісами.

Під інтеграційними сервісами розуміється набір чітко визначених підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення. Як правило їх прийнято іменувати як прикладний програмний інтерфейс (API). API надає розробнику засоби для швидкої розробки програмного забезпечення [44].

Інтегрування допоміжного API в автоматизовану систему забезпечить розширення функціоналу системи та надасть широкий спектр інструментів для комфортної та гнучкої розробки. Принцип роботи API з системою полягає у "спілкуванні" системи із віддаленим сервером відповідного сервісу: система надсилає запит на сервер певного сервісу. Отримавши запит сервіс формує відповідну інформацію. Сформувавши інформацію сервіс використовуючи сервер

надсилає інформацію системі. Система отримавши вхідну інформацію розпочинає процедуру обробки даних.

Відповідно до предметної області для реалізації функціоналу автоматизованої системи було обрано наступні API:

- Booking.com API – даний сервіс реалізовує функціонал пошуку та резервування готельних номерів. Сервіс надає величезну базу даних готельних комплексів з детальною інформацією (репрезентативна інформація готелю, фотографії, готельні номери, ціни, послуги, рейтинги, відгуки клієнтів) та забезпечує можливість здійснення операції резервування (фактичне та тестове) готельного номеру [45];

- Google Maps Geocoding – даний сервіс реалізовує функціонал інтегрування з картографічними сервісами. Сервіс надає можливість отримати дані стосовно розташування відповідного готелю на місцевості (фотографії локації, координати, особливості місцевості) [46];

- Open Weather Map – сервіс реалізовує функціонал прогноз погоди. Сервіс надає детальну інформацію стосовно атмосферних умов до відповідного розташування певного готелю (тип погоди, температура, тиск, швидкість вітру, час заходу та сходу сонця) [47];

- Stripe – сервіс реалізовує функціонал платіжних систем. Даний сервіс надає можливість здійснювати оплату при резервуванні готельного номеру (можливість фактичної оплати та тестової) [48].

Отже, під час процесу дослідження та опису було розроблено інформаційну структуру для автоматизованої системи управління готельними комплексами, що передбачає реалізацію вебтехнології із використанням методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів, модулю реєстрації, автентифікації та авторизації користувача, модулю управління особою інформацією, модулю пошуку готельних комплексів, модулю бронювання кімнат, модулю адміністрування. Автоматизована система розроблена на фреймворку Laravel із використанням СКБД – MySQL Server, мови програмування PHP, мови запитів SQL та інтеграційних сервісів.

Розділ 3 Програмна реалізація інформаційної системи

3.1 Структура та функціональне призначення програмних складових системи

У процесі розробки автоматизованої системи управління готельними комплексами використовувався фреймворк Laravel. Головною перевагою Laravel є базування своєї функціональності на базі шаблону MVC. Основна ідея шаблону MVC полягає в розділенні програми на моделі, контролери та представлення. Контролер реалізовує зв'язок між моделями і представленнями та приймає вхідні дані, які надходять від користувача. Модель представляє та реалізовує бізнес-логіку програмного продукту. Представлення відображає зовнішній вигляд застосунку.

Дана автоматизована система реалізовує наступні моделі:

- User – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про користувачів.
- Role – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з ролями користувачів автоматизованої системи.
- Sex – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної із статтю користувачів.
- UserImage – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про фотографії клієнта.
- Hotel – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про готельні комплекси.
- Start – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про кількість зірок готельних комплексів.
- Rate – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про рейтинг готельних комплексів.
- Country – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про країни, де розташовані готельні комплекси.

- City – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про міста, де розташовані готельні комплекси.
- HotelImage – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про фотографії готельних комплексів.
- FavoriteHotel – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про список улюблених готельних комплексів, які відмітив для себе клієнт.
- Room – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про готельні номери.
- RoomImage – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про фотографії готельних номерів.
- RoomType – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про типи готельних номерів.
- Reservation – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про резервування клієнтом готельного номеру.
- Period – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про дати проживання (в'їзду та виїзду) в кімнатах готельних комплексів.
- Comment – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про коментарі клієнтів, які проживали у певному готельному комплексі.
- Syntax – є моделлю, яка відповідає за реалізацію бізнес-логіки пов'язаної з інформацією про роботу алгоритму синтаксичного аналізу на базі побажань клієнта.

Дана система реалізовує наступні представлення:

- Home – відповідає за відображення головної сторінки автоматизованої системи.
- Login – відповідає за відображення сторінки авторизації.
- Register – відповідає за відображення сторінки реєстрації.
- Загальне представлення User складається з наступних представлень:

- Index – відображає особистий кабінет користувача.
- Create – відображає сторінку створення особової інформації користувача.
- Edit – відображає сторінку оновлення особової інформації користувача.
- Favorite – відображає сторінку улюблених готельних комплексів користувача.
- Offer – відображає сторінку готельних комплексів.
- Single – відображає сторінку конкретного готельного комплексу.
- Reservation – відображає сторінку резервування.
- Загальне представлення Admin складається з наступних категорій

представлень:

- Категорія User складається з наступних представлень:
 - Index – відображає сторінку користувачів.
 - Create – відображає сторінку створення нового користувача.
 - Edit – відображає сторінку редагування інформації певного користувача.
- Категорія Hotel складається з наступних представлень:
 - Index – відображає сторінку готельних комплексів.
 - Create – відображає сторінку створення нового готельного комплексу.
 - Edit – відображає сторінку редагування інформації певного готельного комплексу.
- Категорія Room складається з наступних представлень:
 - Index – відображає сторінку кімнат певного готельного комплексу.
 - Create – відображає сторінку створення нової кімнати певного готельного комплексу.

- Edit – відображає сторінку редагування інформації певного кімнати відповідного готельного комплексу.
- Категорія Reservation складається з наступних представлень:
 - Index – відображає сторінку резервувань.
 - Edit – відображає сторінку редагування інформації певного резервування.

Дана система реалізовує наступні контролери:

- HomeController – відповідає за обробку головної сторінки автоматизованої системи.
- UserController – відповідає за обробку інформації пов'язаної з користувачами.
- AuthController – відповідає за обробку інформації пов'язаної з реєстрацією та авторизацією.
- OfferController – відповідає за обробку інформації пов'язаної з готельними комплексами.
- SinglePageController – відповідає за обробку інформації пов'язаної з конкретним готельним комплексом та його кімнатами.
- ReservationController – відповідає за обробку інформації пов'язаної з резервуванням та оплатою готельного номеру.
- AdminController – відповідає за обробку інформації пов'язаної з діяльністю адміністрації.
- AdminUserController – відповідає за обробку інформації пов'язаної з діяльністю адміністрації по відношенню до клієнтів.
- AdminHotelController – відповідає за обробку інформації пов'язаної з діяльністю адміністрації по відношенню до готельних комплексів.
- AdminRoomController – відповідає за обробку інформації пов'язаної з діяльністю адміністрації по відношенню до кімнат готельних комплексів.
- AdminReservationController – відповідає за обробку інформації пов'язаної з діяльністю адміністрації по відношенню до бронювань клієнтів.

Отже, наведені моделі, представлення та контролери забезпечать функціонал автоматизованої системи. Подібне розділення програмного продукту на три компоненти дозволить розробляти та оновлювати програмний продукт швидко, ефективно та безпечно.

Для більшої наглядності на рисунку 3.1 зображено вигляд структури класів проекту.

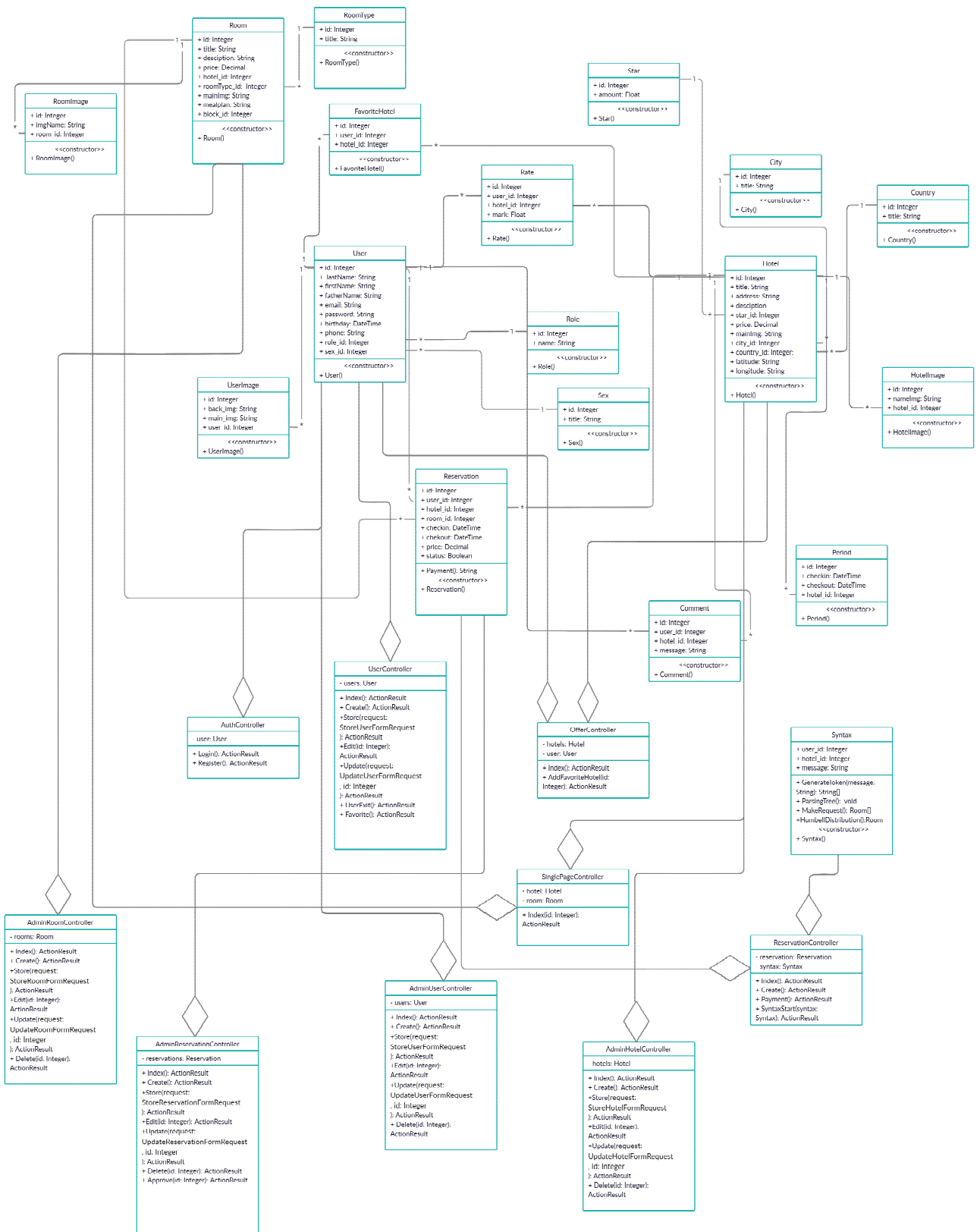


Рисунок 3.1 – Діаграма класів автоматизованої системи управління
ГОТЕЛЬНИМИ КОМПЛЕКСАМИ

3.2 Особливості реалізації програмних складових системи

Головними та найбільш важливими елементами програмного продукту є наступні модулі:

- Головна сторінка автоматизованої системи;
- Реєстрації та авторизації користувача;
- Особистий кабінет користувача;
- Операції перегляду, додавання та редагування особових даних клієнта;
- Пошук та пропозиції готельних комплексів;
- Перегляд детальної інформації про певний готельний комплекс;
- Перегляд детальної інформації про певну кімнату готельного комплексу;
- Панель керування адміністратора;
- Резервування готельних номерів.

Точкою входу у будь-яку систему є головна сторінка, яка відображає основну суть програмного продукту для користувачів. На рисунку 3.2 представлено головну сторінку автоматизованої системи управління готельними комплексами. На рисунку 3.2.1 представлено фрагмент лістингу реалізації головної сторінки.

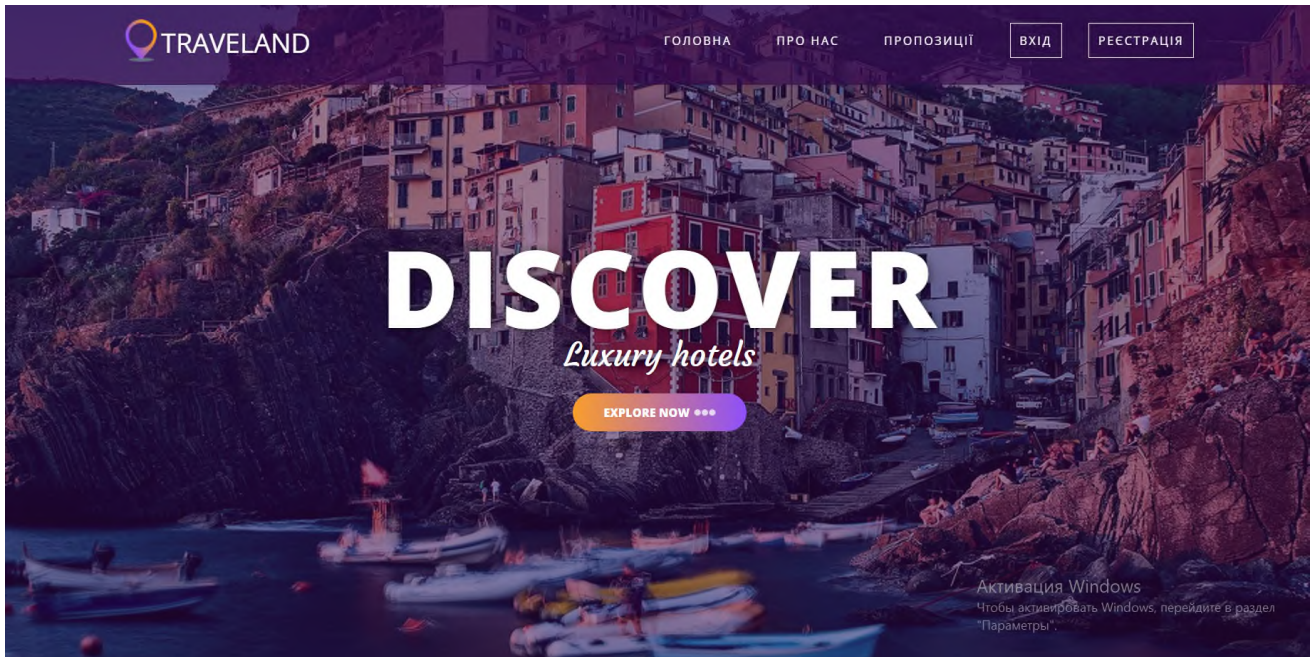


Рисунок 3.2 – Головна сторінка автоматизованої системи управління готельними комплексами

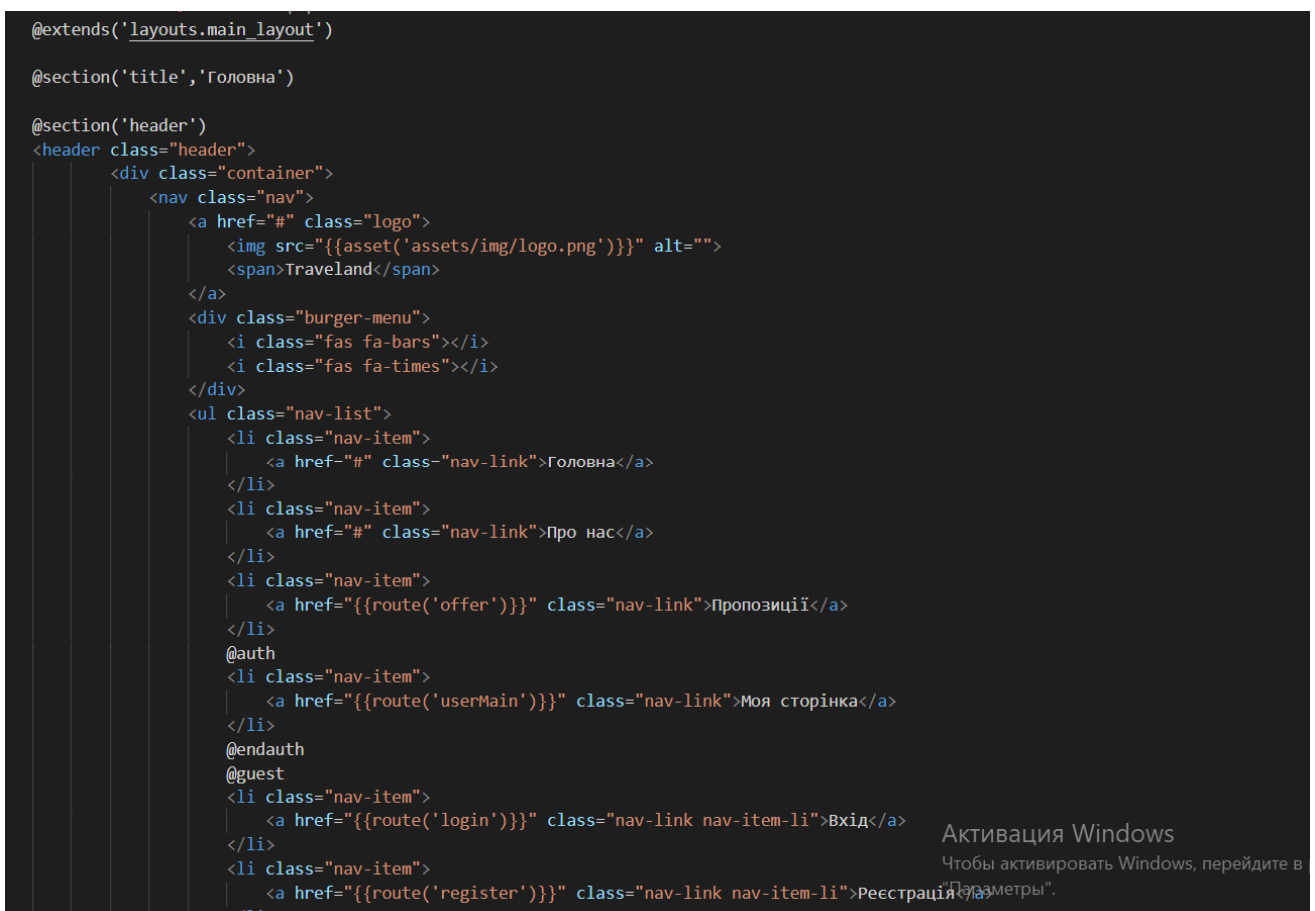
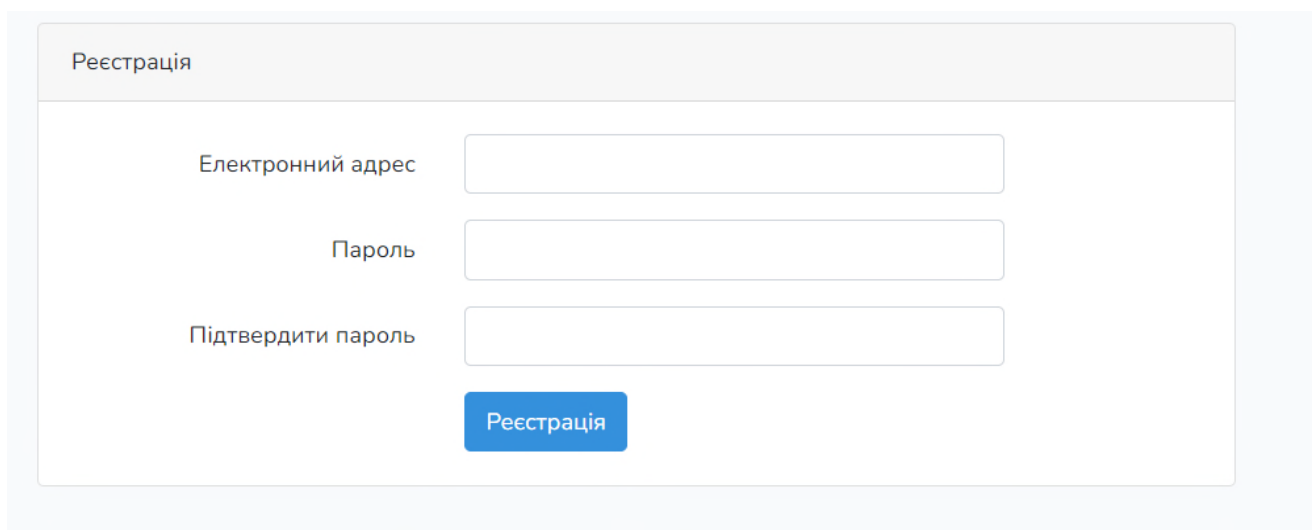


Рисунок 3.2.1 – Фрагмент лістингу головної сторінки автоматизованої системи управління готельними комплексами

Реєстрація в автоматизованій системі виконується за допомогою методу Create контролера AuthController. Обов'язком даного методу є виконання процесу реєстрації користувача в системі. Для валідації даних при спробі зареєструватися відповідає метод Validator контролера AuthController. Даний метод слідкує за коректністю введених даних.

Метод Create та Validator опираються на бізнес-процес «Робота відвідувача з реєстрацією та авторизацією в системі». Результат виконання методу Create зображено на рисунку 3.2.2, методу Validator – на рисунку 3.2.3. Фрагмент лістингу – на рисунку 3.2.4.



Реєстрація

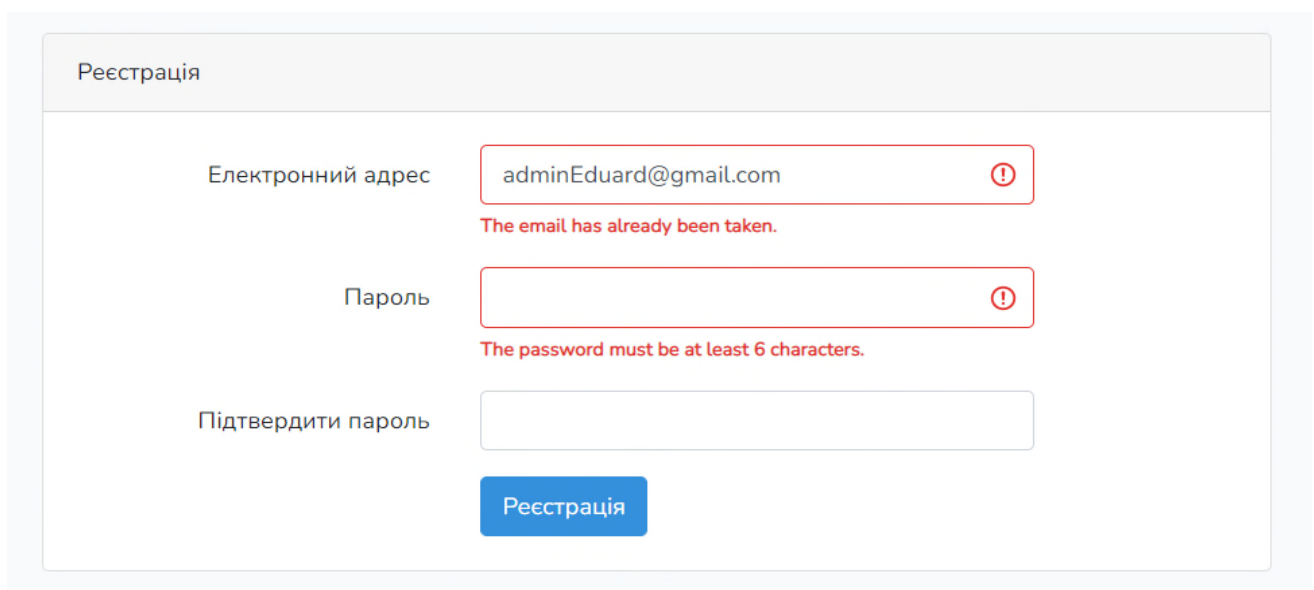
Електронний адрес

Пароль

Підтвердити пароль

Реєстрація

Рисунок 3.2.2 – Форма реєстрації



Реєстрація

Електронний адрес !
The email has already been taken.

Пароль !
The password must be at least 6 characters.

Підтвердити пароль

Реєстрація

Рисунок 3.2.3 – Валідація даних при спробі зареєструватися

```

public function __construct()
{
    $this->middleware('guest');
}

/**
 * Get a validator for an incoming registration request.
 *
 * @param array $data
 * @return \Illuminate\Contracts\Validation\Validator
 */
protected function validator(array $data)
{
    return Validator::make($data, [
        'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
        'password' => ['required', 'string', 'min:6', 'confirmed'],
    ]);
}

/**
 * Create a new user instance after a valid registration.
 *
 * @param array $data
 * @return \App\Models\User
 */
protected function create(array $data)
{
    $user = User::create([
        'email' => $data['email'],
        'password' => Hash::make($data['password']),
    ]);
    $user->assignRole('user');
    return $user;
}

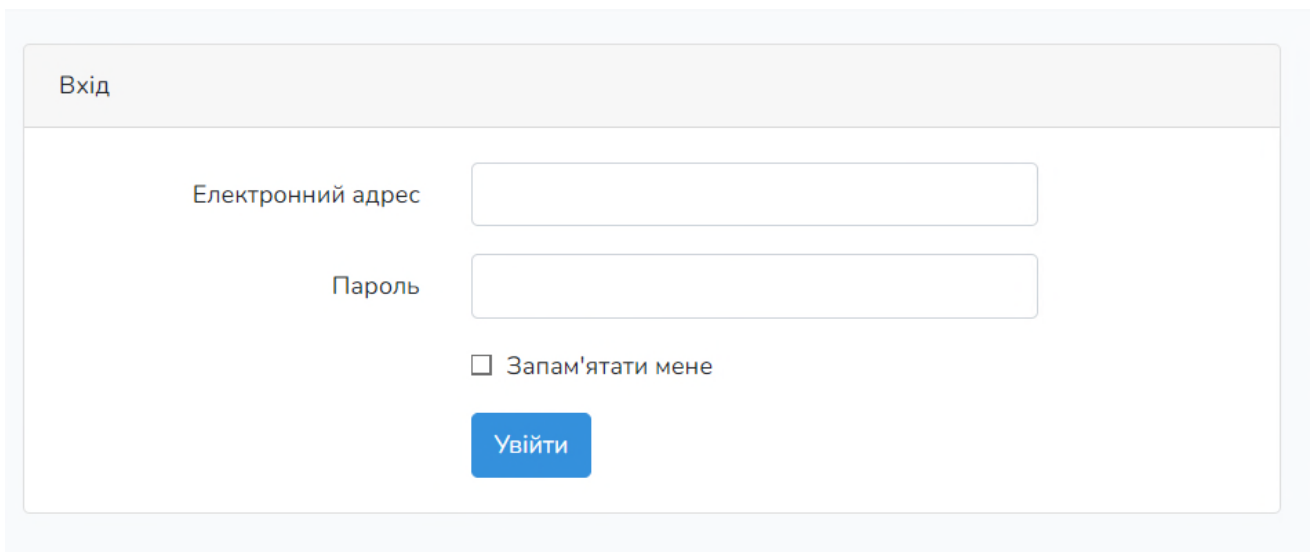
```

Рисунок 3.2.4 – Фрагмент лістингу валідації даних при реєстрації

Авторизація в автоматизованій системі виконується за допомогою методу `Authenticated` контролера `AuthController`. Обов'язком даного методу є виконання процесу авторизації користувача. Для валідації даних при спробі авторизуватися відповідає метод `Validator` трейта `AuthenticatesUser`. Даний метод слідкує за коректністю введених даних.

Метод `Authenticated` та `Validator` опираються на бізнес-процес «Робота відвідувача з реєстрацією та авторизацією в системі». Результат виконання методу

Authenticated зображено на рисунку 3.2.5, методу Validator – на рисунку 3.2.6. Фрагмент лістингу представлено на рисунку 3.2.7.



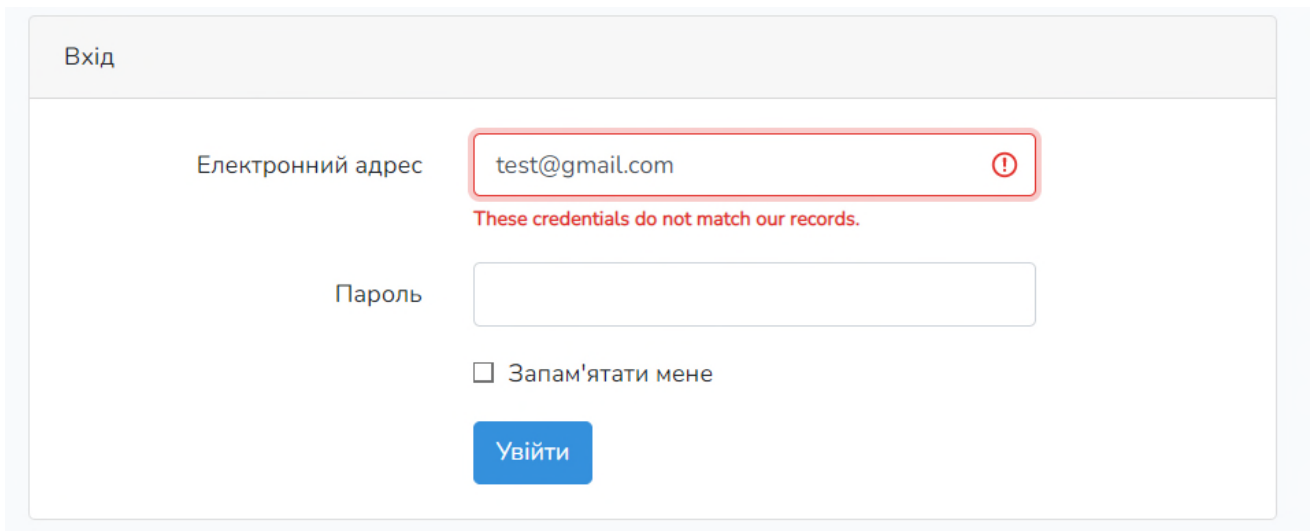
Вхід

Електронний адрес


Пароль

Запам'ятати мене

Рисунок 3.2.5 – Форма авторизації



Вхід

Електронний адрес 

These credentials do not match our records.

Пароль

Запам'ятати мене

Рисунок 3.2.6 – Валідація даних при спробі авторизуватися

```

use AuthenticatesUsers;

/**
 * Where to redirect users after login.
 *
 * @var string
 */
protected function authenticated(Request $request, User $user)
{
    if($user->hasRole('admin')){
        return redirect()->route('adminMain');
    }
    return redirect()->route('userMain');
}

/**
 * Create a new controller instance.
 *
 * @return void
 */
public function __construct()
{
    $this->middleware('guest')->except('logout');
}

```

Рисунок 3.2.7 – Фрагмент лістингу процесу авторизації

Особистий кабінет користувача є місцем, де клієнт може створити власний профіль з особовою інформацією, слідкувати за улюбленими готельними комплексами, переглядати історію резервувань. На рисунку 3.2.8 представлено особистий кабінет користувача, який щойно зареєструвався, на рисунку 3.2.9 – користувача, який вже ввів особову інформацію про себе. Фрагмент лістингу особистого кабінету користувача представлено на рисунку 3.10.

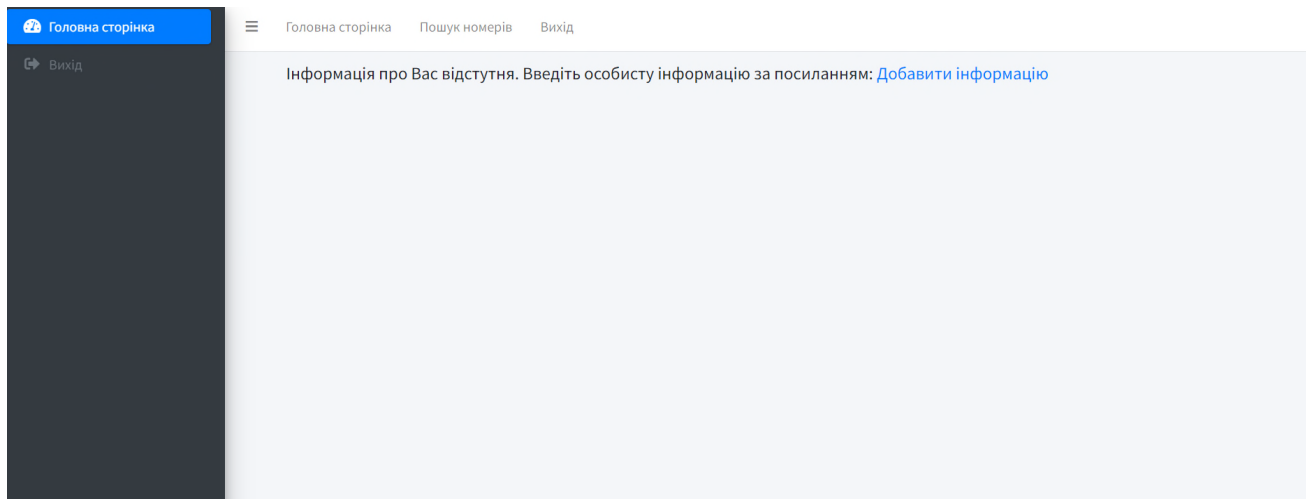


Рисунок 3.2.8 – Особистий кабінет користувача, який щойно зареєструвався

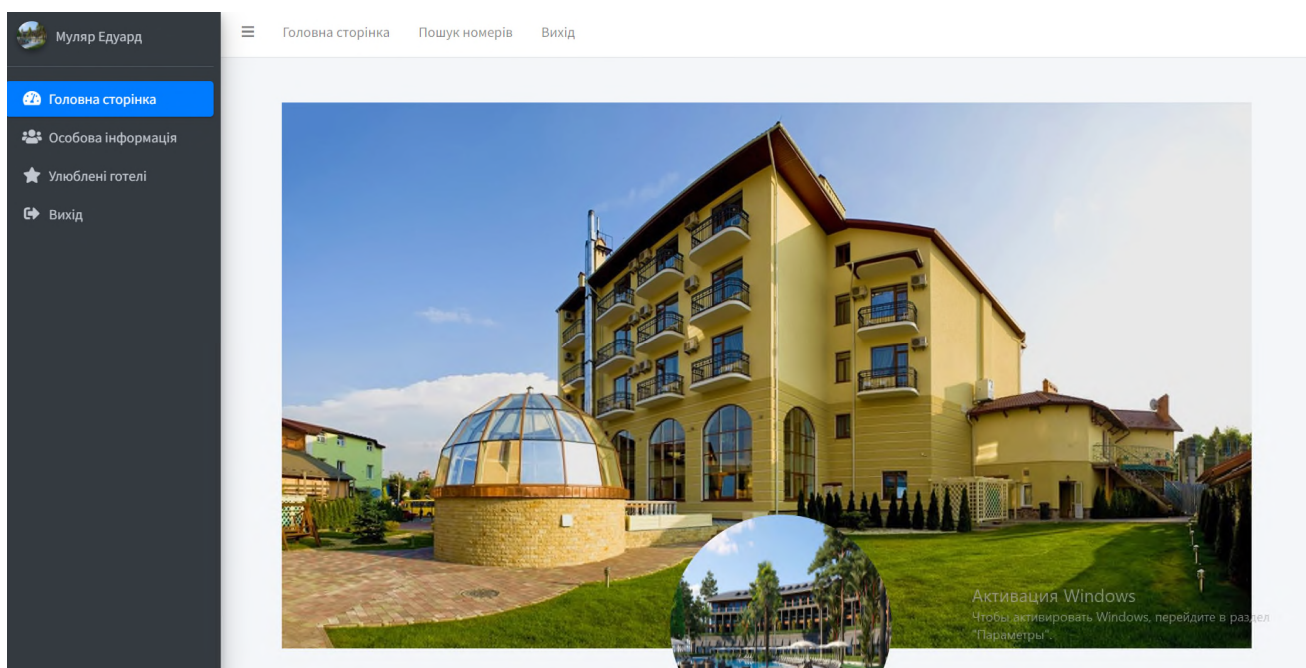


Рисунок 3.2.9 – Особистий кабінет користувача, який ввів особові дані

```

@section('content')
<!-- Content Wrapper. Contains page content -->
<div class="content-wrapper">
  <div class="content">

    @foreach ($user as $us)

      @if(strlen($us->firstName) > 0)
        <div class="content_user_photo">
          <div class="content_user_photo_bg">
            @foreach ($userImage as $image)
              
              <div class="content_user_photo_avatar">
                
              </div>
            @endforeach
          </div>
        </div>
      </div>
      <div class="content_user_name">
        <h2>{{ $us->lastName}} {{ $us->firstName}}</h2>
      </div>
      <div class="content_menu_info">
        <ul class="content_menu_info_nav">
          <li class="menu_info_item"><i class="fas fa-envelope"></i> {{ $us->email}}</li>
          <li class="menu_info_item"><i class="fas fa-birthday-cake"></i> {{ $us->birthday}}</li>
          <li class="menu_info_item"><i class="fas fa-genderless"></i> {{ $us->sexes['title']}}</li>
          <li class="menu_info_item"><i class="fas fa-phone-alt"></i> {{ $us->phone}}</li>
        </ul>
      </div>
      @else
        <h5 style="padding-top: 12px; font-weight: 500;">
        Інформація про Вас відсутня. Введіть особисту інформацію за посиланням:
        <a href="{{route('userCreateInfo')}}">Добавити інформацію</a>
      </h5>
      @endif
    </div>
  </div>

```

Рисунок 3.2.10 – Фрагмент лістингу особистого кабінету користувача

Додавання особової інформації клієнта в автоматизованій системі виконується за допомогою методів Create та StoreDataClient контролера ClientController. Метод Create відповідає за відображення форми додавання особової інформації, метод StoreDataClient – за логіку додавання особової інформації клієнта.

Метод Create та StoreDataClient опираються на бізнес-процес «Робота клієнта з особистою сторінкою». Результат виконання методу Create та Store зображено на рисунку 3.2.11. Фрагмент лістингу – на рисунку 3.2.12.


☰ Головна сторінка Пошук номерів Вихід

Особова інформація користувача

Ім'я

Прізвище

По батькові

Дата народження
 

Стать

Моб. телефон

Задній фон профіля
 Файл не выбран

Активация Win
Чтобы активировать
"Параметры".

Головне фото профіля

Файл не выбран

Рисунок 3.2.11 – Форма додавання особової інформації клієнта

```

public function storeDataClient($validData, $request)
{
    $userCurrent = Auth::user()->email;
    $user = User::where('email', $userCurrent)->get();
    $userId = 0;
    foreach ($user as $key => $value) {
        $userId = $value->id;
    }
    $userCreate = User::find($userId);
    $userCreate->update([
        'firstName'=>$validData['firstName'],
        'lastName'=>$validData['lastName'],
        'fatherName'=>$validData['fatherName'],
        'birthday'=>$validData['birthday'],
        'sex_id'=>$validData['sex_id'],
        'phone'=>$validData['phone'],
    ]);
    $userCreate->save();

    $back_img = '';
    $main_img = '';

    $backImgInput = $request->file('back_img');
    $back_img = $backImgInput->getClientOriginalName();
    $backImgInput->move(public_path().'/assets/images-user', $back_img);

    $mainImgInput = $request->file('main_img');
    $main_img = $mainImgInput->getClientOriginalName();
    $mainImgInput->move(public_path().'/assets/images-user', $main_img);

    UserImage::create([
        'back_img'=>$back_img,
        'main_img'=>$main_img,
        'user_id'=>$userId
    ]);
}

```

Рисунок 3.2.12 – Фрагмент лістингу операції додавання особової інформації клієнта

Оновлення особової інформації клієнта в автоматизованій системі виконується за допомогою методів Edit та UpdateDataClient контролера ClientController. Метод Edit відповідає за відображення форми оновлення особової

інформації клієнта, метод `UpdateDataClient` – за логіку оновлення особової інформації.

Метод `Edit` та `UpdateDataClient` опираються на бізнес-процес «Робота клієнта з особистою сторінкою». Результат виконання методу `Edit` та `UpdateDataClient` зображено на рисунку 3.2.13. Фрагмент лістингу – на рисунку 3.2.14.

☰
Головна сторінка
Пошук номерів
Вихід

Особова інформація користувача

Ім'я

Прізвище

По батькові

Дата народження

Стать

Моб. телефон

Поточний задній фон профіля

Активация V
 Чтобы активировать
 "Параметры".

Задній фон профіля

Выберите файл

Файл не выбран

Поточне головне фото профіля

Головне фото профіля

Выберите файл

Файл не выбран

Оновити

Рисунок 3.2.13 – Форма оновлення особової інформації клієнта

```

public function updateDataClient($request,$id)
{
    $userCreate = User::find($id);
    $userCreate->update([
        'firstName'=>$request->input('firstName'),
        'lastName'=>$request->input('lastName'),
        'fatherName'=>$request->input('fatherName'),
        'birthday'=>$request->input('birthday'),
        'sex_id'=>$request->input('sex_id'),
        'phone'=>$request->input('phone'),
    ]);
    $userCreate->save();

    $back_img = '';
    $main_img = '';

    if($request->has('back_img'))
    {
        $backImgInput = $request->file('back_img');
        $back_img = $backImgInput->getClientOriginalName();
        $backImgInput->move(public_path().'/assets/images-user', $back_img);
    }
    if($request->has('main_img'))
    {
        $mainImgInput = $request->file('main_img');
        $main_img = $mainImgInput->getClientOriginalName();
        $mainImgInput->move(public_path().'/assets/images-user', $main_img);
    }

    if(strlen($back_img) > 0)
    {
        $updateImageUser = UserImage::where('user_id',$id)->get();

        $updateImageid = 0;
    }
}

```

Рисунок 3.2.14 – Фрагмент лістингу операції оновлення особової інформації клієнта

Пошук готельних комплексів є основним механізмом в автоматизованій системі. Даний модуль дозволяє шукати різноманітні готельні комплекси, які є зареєстровані у мережі Інтернет, відображає певну інформацію про кожний комплекс. Пошук готелів здійснюється за допомогою методів Index та Search контролера OfferController. Метод Index відповідає за відображення знайдених готельних комплексів, метод Search – за логіку пошуку готелів.

Метод Index та Search опираються на бізнес-процес «Робота клієнта з реєстром готелів». Результат виконання методу Index та Search зображено на рисунку 3.2.15. Фрагмент лістингу – на рисунку 3.2.16.

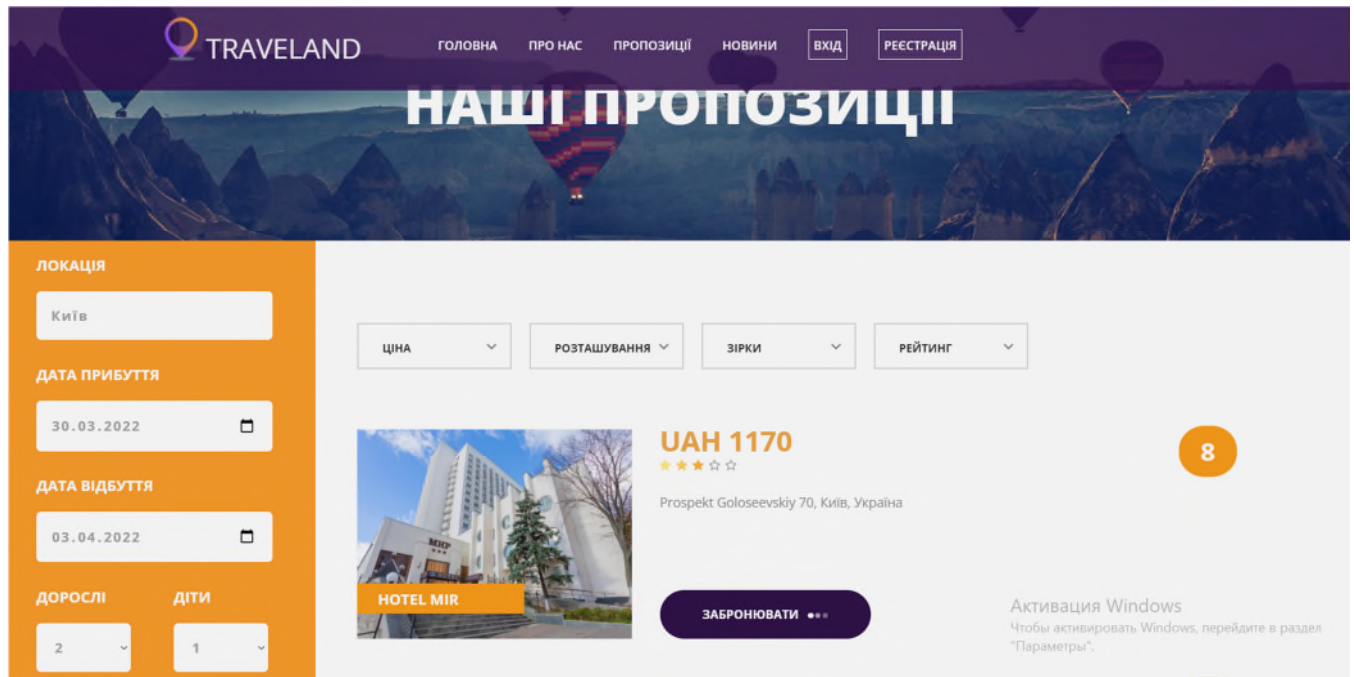


Рисунок 3.2.15 – Сторінка зі знайденими готельними комплексами за заданими параметрами

```

foreach ($result as $item) {
    $country = Country::where('title',$item['country_trans'])->get();
    $city = City::where('cities.title',$item['default_wishlist_name'])->get();

    if(!$country->count()){
        Country::create([
            'title'=>$item['country_trans']
        ]);
    }
    if(!$city->count()){
        City::create([
            'title'=>$item['default_wishlist_name']
        ]);
    }
    $cit = City::where('cities.title',$item['default_wishlist_name'])->get();
    foreach($cit as $c){
        $cityId = $c->id;
    }

    $cot = Country::where('countries.title',$item['country_trans'])->get();
    foreach($cot as $c){
        $cotId = $c->id;
    }

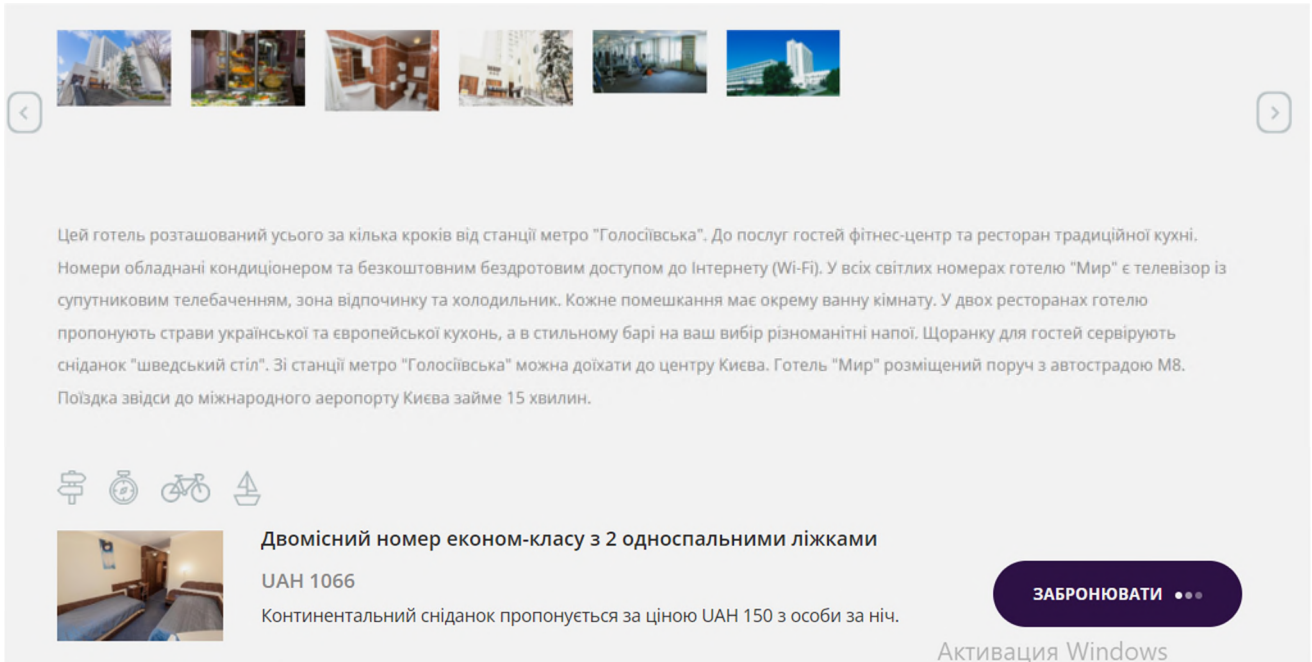
    $star = Star::where('amount',$item['class'])->get();
    $starId = 0;
    foreach($star as $s){
        $starId = $s->id;
    }
    Hotel::create([
        'title'=>$item['hotel_name'],
        'address'=>$item['address'],
        'star_id'=>$starId,
        'price'=>$item['min_total_price'],
        'city_id'=>$cityId
    ]);
}

```


Рисунок 3.2.16 – Фрагмент лістингу пошуку готельних комплексів


Перегляд детальної інформації про певний готельний комплекс є важливим модулем системи, так як забезпечує відображення детальної інформації про певний готельний комплекс: назва, адреса, рейтинг, історія, фотографії, послуги, готельні номери, інтерактивна карта розташування, погода в локації готелю. Модуль перегляду детальної інформації про певний готель працює за допомогою методу Index контролера SinglePageController та методу Display моделі Hotel. Метод Index відповідає за відображення детальної інформації готельного комплексу, метод Display – за логіку обробки та підготовки до відображення інформації певного готельного комплексу.

Метод Index та Display опираються на бізнес-процес «Робота клієнта з реєстром готелів». Результат виконання методу Index та Display зображено на рисунку 3.2.17. Фрагмент лістингу – на рисунку 3.2.18.



Цей готель розташований усього за кілька кроків від станції метро "Голосіївська". До послуг гостей фітнес-центр та ресторан традиційної кухні. Номери обладнані кондиціонером та безкоштовним бездротовим доступом до Інтернету (Wi-Fi). У всіх світлих номерах готелю "Мир" є телевізор із супутниковим телебаченням, зона відпочинку та холодильник. Кожне помешкання має окрему ванну кімнату. У двох ресторанах готелю пропонують страви української та європейської кухонь, а в стильному барі на ваш вибір різноманітні напої. Щоранку для гостей сервірують сніданок "шведський стіл". Зі станції метро "Голосіївська" можна доїхати до центру Києва. Готель "Мир" розміщений поруч з автострадою М8. Поїздка звідси до міжнародного аеропорту Києва займе 15 хвилин.




Двомісний номер економ-класу з 2 односпальними ліжками
 UAH 1066
 Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

[ЗАБРОНЮВАТИ](#) ●●●

Активация Windows

Рисунок 3.2.17 – Сторінка з детальною інформацією певного готельного комплексу

```

$hotelImg = [];
$checkin = session('checkin');
$checkout = session('checkout');
$adult = session('adult');
$children = session('child');

$hotel = Hotel::where('id', $id)->get();

$hotelId = 0;
$hotelDesc = "";

foreach ($hotel as $key => $value) {
    $hotelId = $value->hotel_id;
    $hotelDesc = $value->description;
}

$hotelImgTest = HotelImage::where('hotel_id', $id)->get();

```

Рисунок 3.2.18 – Фрагмент лістингу модуля перегляду детальної інформації про певний готельний комплекс

Перегляд детальної інформації про певну кімнату готельного комплексу є модулем системи, який забезпечує відображення детальної інформації про певну кімнату відповідного готельного комплексу: назва, тип кімнати, ціна за період проживання, опис, фотографії, присутність елементів побуту та послуг. Модуль перегляду детальної інформації про певну кімнату готельного комплексу працює

за допомогою методу Index контролера SinglePageController та методу Display моделі Room. Метод Index відповідає за відображення детальної інформації про певну кімнату, метод Display – за логіку обробки та підготовки до відображення інформації певної кімнати готельного комплексу.

Метод Index та Display опираються на бізнес-процес «Робота клієнта з номерним фондом готелів». Результат виконання методу Index та Display зображено на рисунку 3.2.19. Фрагмент лістингу – на рисунку 3.2.20.

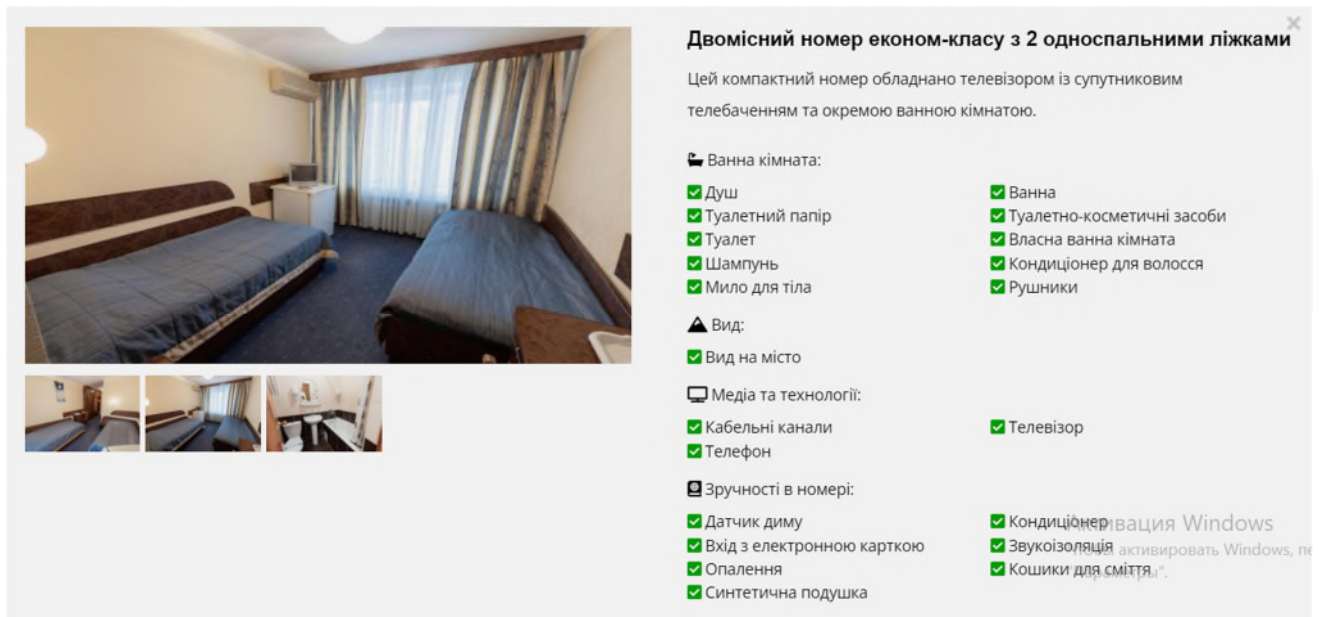


Рисунок 3.2.19 – Сторінка з детальною інформацією про кімнату певного готельного комплексу

```

$roomServicesType = RoomOffersType::all();
$roomsServices = Room::where('hotel_id', $id)->get();

foreach ($roomsServices as $key => $value) {
    foreach ($roomPhoto as $key => $service) {
        if($key == $value['block_id']){
            foreach ($service['facilities'] as $key => $facilities) {
                foreach ($roomServicesType as $key => $roomServicesTypeItem) {
                    if($roomServicesTypeItem->title == $facilities['alt_facilitytype_name'])
                    {
                        RoomOffer::create([
                            'title'=>$facilities['name'],
                            'room_id'=>$value['id'],
                            'room_offers_type_id'=>$roomServicesTypeItem->id
                        ]);
                        break;
                    }
                }
            }
        }
    }
}

foreach ($roomsServices as $value) {
    foreach ($roomPhoto as $key => $service) {
        if($key == $value['block_id']){
            foreach ($service['photos'] as $photoItem) {
                RoomImage::create([
                    'imgName'=>$photoItem['url_original'],
                    'room_id'=>$value->id
                ]);
            }
        }
    }
}

```

Активация Windows
 Чтобы активировать Windows, перейдите
 "Параметры".

Рисунок 3.2.20 – Фрагмент лістингу модуля перегляду детальної інформації про кімнату готельного комплексу

Панель управління є основним механізмом керування діяльністю адміністратора в автоматизованій системі. Дана панель надає можливість керувати обліковими записами клієнтів; додавати, видаляти, оновлювати та переглядати інформацію про кожний готельний комплекс, а також забезпечує можливість керувати резервуваннями клієнтів. Головна сторінка панелі управління адміністратора представлена на рисунку 3.2.21, на рисунку 3.2.22 – фрагмент лістингу.

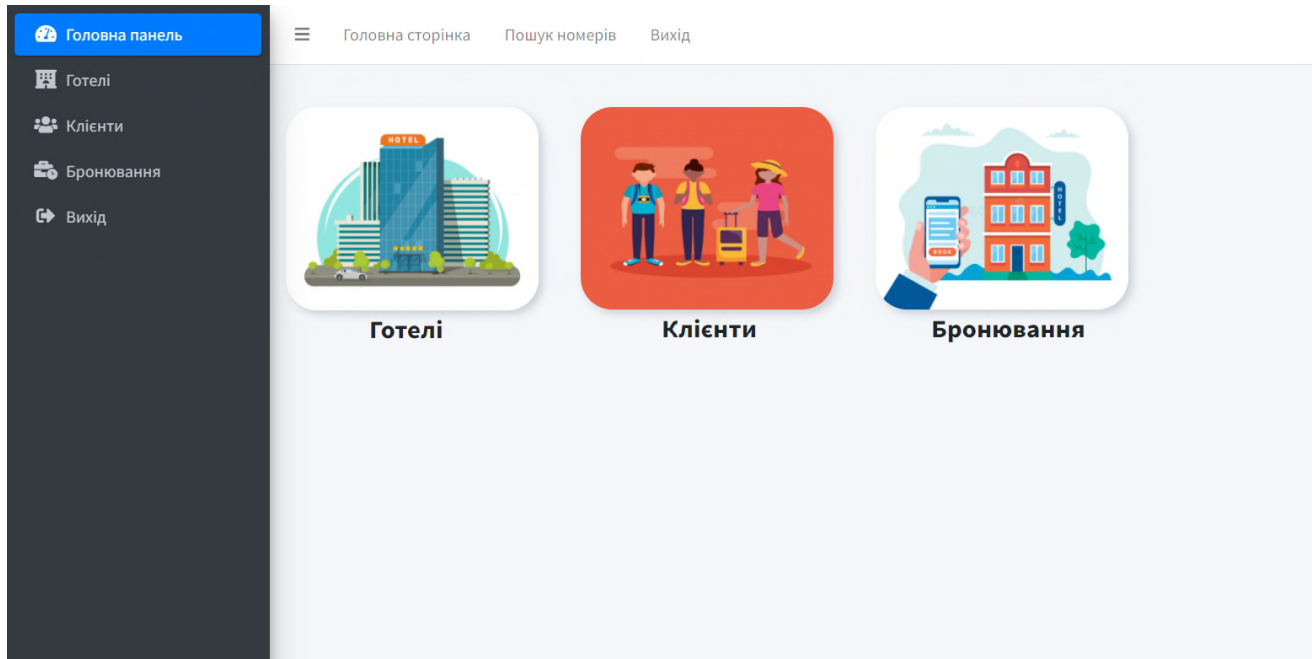


Рисунок 3.2.21 – Панель управління адміністратора

```
@section('content')
<div class="content-wrapper">
  <!-- Content Header (Page header) -->

  <!-- /.content-header -->
  <!-- Main content -->
  <section class="content">
    <div class="container-fluid">
      <!-- Small boxes (Stat box) -->
      <div class="content__panel">
        <div class="content__panel_img">
          <a href="{{route('hotels.index')}}"></a>
          <p style="margin-left: 33%; font-size: 25px; font-weight: bold;">Готелі</p>
        </div>
        <div class="content__panel_img">
          <a href="{{route('users.index')}}"></a>
          <p style="margin-left: 32%; font-size: 25px; font-weight: bold;">Клієнти</p>
        </div>
        <div class="content__panel_img">
          <a href="#"></a>
          <p style="margin-left: 22%; font-size: 25px; font-weight: bold;">Бронювання</p>
        </div>
      </div>
    <!-- /.content -->
  </div>

  <!-- /.content -->
</div>
@endsection
```

Рисунок 3.2.22 – Фрагмент лістингу панелі управління адміністратора

Управління реєстром готелів є модулем системи, який забезпечує відображення реєстру готельних комплексів, надає можливість додавати

інформацію про новий готельний комплекс, оновлювати інформацію відповідного готелю, а також видаляти будь-який готель із реєстру. Даний модуль реалізовується за допомогою методів Index, Create, Store, Edit, Update, Delete контролера AdminHotelController та моделі Hotel. Метод Index відповідає за відображення реєстру готелів, метод Create – за форму додавання інформації про новий готель, Store – за логіку додавання інформації, Edit – за форму оновлення інформації відповідного готелю, Update – за логіку оновлення, Delete – за видалення готельного комплексу із реєстру. За реалізацію даного модуля відповідає бізнес-логіка «Робота адміністратора з реєстром готелів».

На рисунку 3.2.23 представлено реєстр готельних комплексів, на рисунку 3.2.24 – фрагмент лістингу.

#	Назва	Адреса	Кількість зірок	Місто	Країна	Дії
3131	Basseynaya Apart Hotel	Basseynaya Street 3	0	Київ	Україна	👁️ ✏️ 🗑️
3130	Apartment Комфорт Таун	проспект Соборності	0	Київ	Україна	👁️ ✏️ 🗑️
3129	7 Sky on Shchorsa Street	Shchorsa Street 3 (Yevhena Konovaltsia street)	0	Київ	Україна	👁️ ✏️ 🗑️
3128	Hotel Mir	Prospekt Goloseevskiy 70	3	Київ	Україна	👁️ ✏️ 🗑️
3127	BestKievApartment	Different locations in city center	0	Київ	Україна	👁️ ✏️ 🗑️
3126	Fire Inn	Tarasovska Street 4	0	Київ	Україна	👁️ ✏️ 🗑️
3125	Park Inn by Radisson Kyiv Troyitska	55 Velyka Vasylkivska Street	4	Київ	Україна	👁️ ✏️ 🗑️
3124	Апартаменти на Осокорках	182-а Садова вулиця	0	Київ	Україна	👁️ ✏️ 🗑️
3123	Evropeyskiy	Dilova (ex.Dimitrova) Street 14A	0	Київ	Україна	👁️ ✏️ 🗑️
3122	VIP apartment on 14 Vidradnyi Avenue	14 Vidradnyi Avenue 2 этаж 7 парадное	0	Київ	Україна	👁️ ✏️ 🗑️

Рисунок 3.2.23 – Реєстр готельних комплексів

```
|  |
| --- |
|  |

```

Рисунок 3.2.24 – Фрагмент лістингу реєстру готельних комплексів

Управління реєстром клієнтів є модулем системи, який забезпечує відображення реєстру клієнтів, надає можливість додавати інформацію про нового клієнта, оновлювати інформацію відповідного клієнта, а також видаляти будь-якого клієнта із реєстру. Даний модуль реалізовується за допомогою методів Index, Create, Store, Edit, Update, Delete контролера AdminUserController та моделі User. Метод Index відповідає за відображення реєстру клієнтів, метод Create – за форму додавання інформації про нового клієнта, Store – за логіку додавання інформації, Edit – за форму оновлення інформації відповідного клієнта, Update – за логіку оновлення, Delete – за видалення клієнта із реєстру. За реалізацію даного модуля відповідає бізнес-логіка «Робота адміністратора з особовою інформацією клієнтів».

На рисунку 3.2.25 представлено реєстр готельних комплексів, на рисунку 3.2.26 – фрагмент лістингу.

Реєстр користувачів								
Добавити нового користувача								
#	Прізвище	Ім'я	Електронний адрес	Стать	Дата народження	Дії		
2			edikmulyar228@gmail.com					
3			admin@mail.ru					
4			qwerty@gmail.com					
6			orb@gmail.com					
7	Муляр	Інна	ok228@gmail.com	жін	2001-01-18			
8			adminedik@mail.ru					
9	Муляр	Едуард	ediktest@gmail.com	чол	2001-01-19			
13	Стасюк	Роман	romanStasuk@gmail.com	чол	2001-01-20			
14			adminEduard@gmail.com					

Рисунок 3.2.25 – Реєстр клієнтів

```

<tbody>
  @foreach ($users as $user)
    <tr>
      <td>{{ $user->id }}</td>
      <td>{{ $user->lastName }}</td>
      <td>
        | {{ $user->firstName }}
      </td>
      <td>{{ $user->email }}</td>
      <td>{{ $user->sexes['title'] }}</td>
      <td>{{ $user->birthday }}</td>
      <td>
        <div style="display: flex;">
          <a title="Проглянути" style="margin-right: 10px;" href="#"><i class="far fa-eye"></i></a>
          <a title="Редагувати" style="margin-right: 10px; color: green;" href="{{ route('users.edit', $user->id) }}">
            <form action="{{ route('users.destroy', $user->id) }}" method="POST">
              @csrf
              @method('DELETE')
              <button type="submit" onclick="return confirm('Видалити даного користувача?');" style="color: red; background-color: #f00;">
            </form>
          </a>
          <!-- <a title="Видалити" href="#" style="color: red;"><i class="fas fa-trash"></i></a> -->
        </div>
      </td>
    </tr>
  @endforeach
</tbody>

```

Рисунок 3.2.26 – Фрагмент лістингу реєстру клієнтів

Отже, було представлено розробку програмних модулів, які є найбільш важливими та цікавими з точки зору програмного коду та свого призначення.

3.3 Тестування інформаційної системи

Обов'язковим етапом при розробці будь-якого програмного продукту є процес тестування функціоналу системи. Проведення процесу тестування забезпечує розробників інформацією про те, що програмний додаток працює коректно до поставлених вимог. Для перевірки коректності роботи автоматизованої системи було проведено модульне тестування (unit-тестування).

Модулем для тестування виступає функція авторизації користувача. Дана функція слугує для того, щоб забезпечити користувача доступом до всього функціоналу автоматизованої системи. Метод, який забезпечує тестування модулю авторизації є `test_authorize`. Даний метод виконується поетапно:

1. Перехід за відповідним посиланням на сторінку форми авторизації;
2. Передача облікових даних користувача для здійснення авторизації;
3. Виконання функції авторизації, яка повертає результат в змінну «response» про те, чи успішно був авторизований користувач;
4. Перевірка результату змінної «response» на еквівалентність статусу коду 200 (успішне виконання).

Лістинг та результат виконання методу `test_authorize` представлено на рисунках 3.3.1 та 3.3.2.

```
public function test_authorize()
{
    $response = $this->post('/login',[
        'email'=>'test@gmail.com',
        'password'=>'test12345'
    ]);

    $this->assertEquals(200, $response->status());
}
```

Рисунок 3.3.1 – Лістинг методу `test_authorize`

```
PASS Tests\Unit\AuthTest
✓ authorize
```

Рисунок 3.3.2 – Результат виконання методу `test_authorize`

Процес тестування модуля авторизації пройдено: успішно.

Наступним модулем для тестування виступає функція пошуку готелів. Дана функція необхідна для здійснення пошуку готельних комплексів по заданим параметрам у базі даних. Метод, який забезпечує тестування модулю пошуку готельних комплексів є `test_search`. Даний метод виконується поетапно:

1. Перехід за відповідним посиланням на сторінку пошуку готельних комплексів;
2. Передача параметрів, які необхідні для здійснення пошуку;
3. Виконання функції пошуку, яка повертає результат в змінну «response» про те, чи успішно був знайдений готельний комплекс;
4. Перевірка результату змінної «response» на еквівалентність статусу коду 200 (успішне виконання).

Лістинг та результат виконання методу `test_search` представлено на рисунках 3.3.3 та 3.3.4.

```
public function test_search()
{
    $response = $this->post('/offers',[
        'location' => 'Kyiv',
        'checkin' => '01.07.2022',
        'checkout'=>'03.07.2022',
        'adults'=>2,
        'children'=>1
    ]);

    $this->assertEquals(200, $response->status());
}
```

Рисунок 3.3.3 – Лістинг методу `test_search`

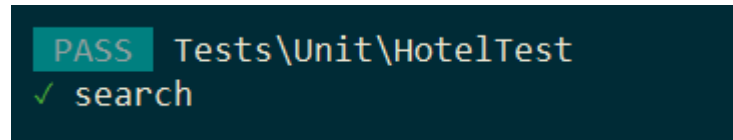


Рисунок 3.3.4 – Результат виконання методу `test_search`

Процес тестування модуля пошуку готельних комплексів пройдено: успішно.

Наступним модулем для тестування виступає функція резервування готельних номерів. Дана функція необхідна для здійснення процесу резервування готельного номеру для клієнта. Метод, який забезпечує тестування модулю резервування готельних номерів є `test_booking`. Даний метод виконується поетапно:

1. Перехід за відповідним посиланням на сторінку резервування готельного номеру;
2. Передача особових даних користувача, а також інформації щодо резервування;
3. Виконання функції резервування, яка повертає результат в змінну «response» про те, чи успішно було здійснено резервування кімнати;
4. Перевірка результату змінної «response» на еквівалентність статусу коду 200 (успішне виконання).

Лістинг та результат виконання методу `test_booking` представлено на рисунках 3.3.5 та 3.3.6.

```

public function test_booking()
{
    $response = $this->post('/reservation/1/room/5', [
        'firstName' => 'Test',
        'lastName' => 'Test',
        'birthday' => '01.07.1990',
        'hotel_id' => 1,
        'room_id' => 5,
        'checkin' => '01.2022',
        'checkout' => '03.07.2022',
        'adults' => 2,
        'children' => 1,
        'price' => 3525
    ]);

    $this->assertEquals(200, $response->status());
}

```

Рисунок 3.3.5 – Лістинг методу test_booking

```

PASS Tests\Unit\ReservationTest
✓ booking

```

Рисунок 3.3.6 – Результат виконання методу test_booking

Процес тестування модуля резервування готельного номеру пройдено: успішно.

Таким чином, було проведено тестування автоматизованої системи управління готельними комплексами, яке продемонструвало коректну роботу функціональних можливостей системи.

3.4 Особливості реалізації методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів

Метод підбору готельних номерів є ключовим елементом функціонуючої структури системи управління готельними комплексами.

Першим кроком у функціонуванні методу є синтаксичний аналіз вхідного тексту клієнта за відповідними правилами та утворення лексичних токенів (ключових слів). На рисунку 3.4.1 продемонстровано форму для введення тексту та відображення правил для коректного вводу. На рисунку 3.4.2 – фрагмент коду, який відповідає за генерацію лексичних токенів. Таблиця 3.1 відображає результат генерації лексичних токенів.

Правила для написання коректного запиту:

* - спец. символ, який вказує на обов'язковість слова (побажання), має найвищий пріоритет 1, приклад: *приклад

& - спец. символ, який вказує на неявну обов'язковість слова (побажання), має пріоритет 2, приклад: &приклад

<, >, >=, <=, = - спец. символи, які слугують для вказання діапазону цін готельних номерів, мають пріоритет 3, приклад: >1500 <=2000

: - спец. символ, який вказує на перерахування побажань, має пріоритет 4, приклад: номер:душ,ванна

Я хотів би номер з видом на місто. В номері обов'язково повинне бути *біде, &паркет. Ціна номеру <3000

ПІДБРАТИ

Рисунок 3.4.1 – Форма введення побажань, відображення правил написання

Таблиця 3.1 – Результат генерація лексичних токенів

Вхідне значення	Я хотів би номер з видом на місто. В номері обов'язково повинне бути *біде, &паркет. Ціна номеру <3000
Вихідне значення	Токен: біде Токен: паркет Токен: 3000

Наступним кроком є створення дерева розбору на базі отриманих лексичних токенів. Дане дерево дозволить утворити своєрідну структуру даних, яка забезпечить можливість гнучкого отримання ключових слів для генерації запиту до бази даних. На рисунку 3.4.3 продемонстровано фрагмент коду, який відповідає за створення дерева розбору та отримання ключових слів для побудови запиту.

```
$parseTree = new ParseTree();
$parseTree->generate($tokens);
$keywordsForRequest = $parseTree->getElements();
```

Рисунок 3.4.3 – Фрагмент коду, який відповідає за створення дерева розбору та отримання ключових слів

Далі на базі отриманих ключових слів необхідно згенерувати запит до бази даних. Використовуючи технологію ORM Eloquent побудова запитів здійснюється максимально гнучко та комфортно, а виконання самих запитів є максимально швидким та легконавантаженим. Запит до бази даних продемонстровано на рисунку 3.4.4.

```

if(count($wordsWithSignPrice) > 0)
{
    for($i = 0; $i<count($starResulter); $i++)
    {
        for($j = 0; $j<count($wordsWithSignPrice); $j++)
        {
            if($lengthSignPrice > 1)
            {
                if($wordsWithSignPrice[$j][0] == '>' || $wordsWithSignPrice[$j][0] == '<')
                {
                    $tmp = Room::where('id',$starResulter[$i]->room_id)->where([[ 'price','>',$priceSignMany[0]],['p
                    array_push($withPriceStars, ...$tmp);
                }
                else
                {
                    $tmp = Room::where('id',$starResulter[$i]->room_id)->where([[ 'price','>=',$priceSignMany[0]],['p
                    array_push($withPriceStars, ...$tmp);
                }
            }
            else if($lengthSignPrice == 1)
            {
                $price = substr($wordsWithSignPrice[$j],1);
                $tmp = Room::where('id',$starResulter[$i]->room_id)->where('price',$wordsWithSignPrice[$j][0],(floo
                array_push($withPriceStars, ...$tmp);
            }
        }
    }
}

```

Активация Windows
Чтобы активировать Windows, перейдите в "Параметры".

```

}
}
else
{
    for($i = 0; $i<count($starResulter); $i++)
    {
        $tmp = Room::where('id',$starResulter[$i]->room_id)->get();
        array_push($withoutPriceStars, ...$tmp);
    }
}
}
}

```

Рисунок 3.4.4 – Генерація запиту та здійснення пошуку готельних номерів

Заключний етап у функціонуванні методу підбору є визначення набору найбільш відповідних готельних номерів, де першим елементом даного набору буде кімната, яка максимально точно задовольняє побажання клієнта. Для визначення такого набору використовується метод `getOptimalRooms`, який

базується на використанні розподілу Гумбеля. На рисунку 3.4.5 продемонстровано фрагмент коду методу визначення набору найбільш відповідних кімнат, на рисунку 3.4.6 – фінальний результат виконання методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів.

```
public function getOptimalRooms($rooms)
{
    foreach ($rooms as $i => $k) {
        Support::checkLimits(['k' => $this->supportLimits['k'][$i]], ['k' => $k]);
    }

    $n = array_sum($rooms);
    $total = array_sum($this->quantities);

    $optimalRooms = array_product(array_map(
        function (int $quantity, int $room) {
            return Combinatorics::gumbel($quantity, $room);
        },
        $this->quantities,
        $rooms
    ));

    return $optimalRooms;
}
```

Рисунок 3.4.5 – Фрагмент коду методу визначення набору найбільш відповідних кімнат



Розкішні апартаменти

UAH 2389

Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

ЗАБРОНЮВАТИ ...



Апартаменти

UAH 2119

Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

ЗАБРОНЮВАТИ ...

Рисунок 3.4.6 – Результат виконання методу підбору готельних номерів

Отже, створення методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів дозволяє користувачам отримати набір найбільш відповідних кімнат до заданих побажань.

3.5 Вимоги до розгортання інформаційної системи

Для коректного розгортання та функціонування автоматизованої системи необхідно:

вимоги до браузера:

- Браузер Google Chrome, від версії 95 – поточна версія;
- Браузер Opera, від версії 78 – поточна версія;
- Браузер Microsoft Edge, від версії 92 – поточна версія;
- Браузер Mozilla Firefox, від версії 91 – поточна версія;
- Браузер Safari, від версії 15.1 – поточна версія.

вимоги до провайдера:

- MySQL Server, від версії 2.0 – поточна версія;
- Дисковий простір – не менше 1000 Мб;
- Підтримування Laravel 8.

Висновки

Під час виконання кваліфікаційної роботи бакалавра проведено аналіз предметної області та визначено основні параметри для розв'язку поставленої задачі. Описано технології, методи та сервіси, які використовувались при розробці систем автоматизації готельних комплексів з визначенням програмних платформ та середовищ в яких вони розроблялись та функціонують.

Спроектовано та розроблено веб-технологію з можливістю підбору готельних номерів зі словесного опису побажань клієнтів. Особливістю даної технології є метод підбору, який здійснює синтаксичний аналіз побажань клієнтів і за певним алгоритмом підбирає найбільш відповідний готельний номер. Також розроблено та описано функціональну структуру системи та представлено її у вигляді груп функцій.

Розроблено структуру бази даних та її об'єктно-реляційну проєкцію. Здійснено опис таблиць та полів бази даних, визначено та представлено взаємодію таблиць з групами функцій.

Описано та графічно представлено реалізацію функціональних складових системи. Продемонстровано роботу методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів. Проведено тестування основних програмних модулів системи.

Відповідно до отриманого результату, можна зробити висновок, що розроблена вебтехнологія для автоматизованої системи управління готельними комплексами працює коректно, а вимоги технічного завдання виконанні в повному обсязі.

Перелік посилань

1. Бронювання готелів. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Бронювання_готелів
2. АПЕПС. [Електронний ресурс]. – Режим доступу: <http://apeps.kpi.ua/shcho-take-informatsiini-technologii/en>
3. Особливості бронювання готелів. [Електронний ресурс]. – Режим доступу: <https://vidpochivay.top/osoblivosti-bronjuvannja-goteliv-cherez-internet/>
4. Характеристика готельних послуг. [Електронний ресурс]. – Режим доступу: https://tourlib.net/books_ukr/pucentejlo91.htm
5. Типи сучасних готелів. [Електронний ресурс]. – Режим доступу: https://tourlib.net/books_ukr/roglev01-5.htm
6. Зірки на готелі. [Електронний ресурс]. – Режим доступу: <http://hotel-hetman.com.ua/stati/zvezdy-na-otele-chto-oni-oznachaiut>
7. Що означає кількість зірок. [Електронний ресурс]. – Режим доступу: <http://tursvit.info/shho-oznachaye-kilkist-zirok-gotelyu/>
8. Класифікація готельних номерів. [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5436979/page:53/>
9. Технологія бронювання місць і номерів. [Електронний ресурс]. – Режим доступу: https://tourlib.net/books_ukr/roglev04-3.htm
10. Види бронювання. [Електронний ресурс]. – Режим доступу: <https://kafedra.com.ua/vydy-bronyuvannya-sposoby-ta-osoblyvosti-protsesu/>
11. Онлайн-бронювання. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Онлайн-бронювання>
12. Служби готелю. [Електронний ресурс]. – Режим доступу: https://pidru4niki.com/14000914/turizm/sluzhbi_gotelyu
13. Що таке MVC? [Електронний ресурс]. – Режим доступу: <https://ru.hexlet.io/blog/posts/chto-takoe-mvc-rasskazyvaem-prostymi-slovami>

14. ASP.NET Обзор MVC. [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/ru-ru/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>
15. Model-View-Presenter. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Model-View-Presenter>
16. Що таке MVP і як це використовувати. [Електронний ресурс]. – Режим доступу: https://skillbox.ru/media/code/chto_takoe_mvp_i_kak_eto_ispolzovat/
17. Model-View-ViewModel. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Model-View-ViewModel>
18. Патерн MVVM. [Електронний ресурс]. – Режим доступу: <https://metanit.com/sharp/wpf/22.1.php>
19. Основні поняття систем баз даних. [Електронний ресурс]. – Режим доступу: https://msn.khnu.km.ua/pluginfile.php/308271/mod_resource/content/3/ОБДЗ%20Тема%20№02%20Основні%20поняття%20й%20архітектура%20систем%20баз%20даних.pdf
20. Що таке база даних? [Електронний ресурс]. – Режим доступу: <http://areps.kpi.ua/shco-take-basa-danykh>
21. Реляційна модель даних та нормалізація. [Електронний ресурс]. – Режим доступу: https://msn.khnu.km.ua/pluginfile.php/308273/mod_resource/content/3/ОБДЗ%20Тема%20№04%20Реляційна%20модель%20та%20нормалізація.pdf
22. PHP: Eloquent (ORM). [Електронний ресурс]. – Режим доступу: https://ru.hexlet.io/courses/php-orm-eloquent/lessons/concepts/theory_unit
23. Laravel Jetstream. [Електронний ресурс]. – Режим доступу: <https://jetstream.laravel.com/2.x/introduction.html>
24. Blade Templates. [Електронний ресурс]. – Режим доступу: <https://laravel.com/docs/9.x/blade>
25. File Storage – Laravel. [Електронний ресурс]. – Режим доступу: <https://laravel.com/docs/9.x/filesystem>

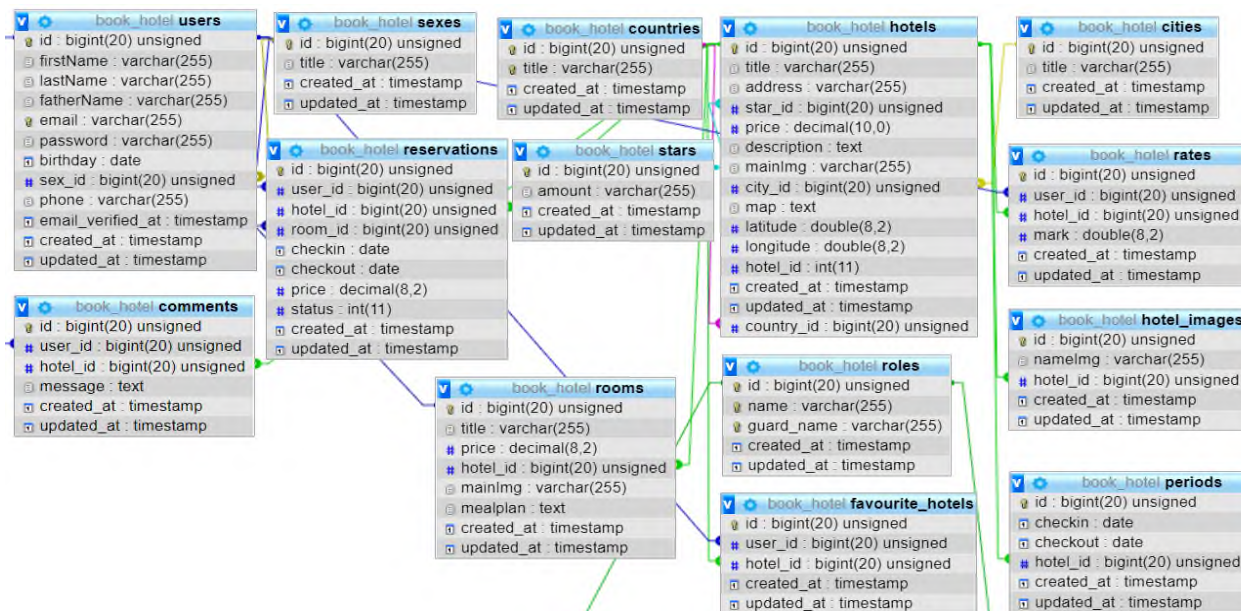
26. Автоматизовані системи бронювання. [Електронний ресурс]. – Режим доступу: https://tourlib.net/statti_ukr/melnychenko7.htm
27. Система керування готелем. [Електронний ресурс]. – Режим доступу: http://cyclowiki.org/wiki/Система_управления_гостиницей
28. Booking.com [Електронний ресурс]. – Режим доступу: https://www.booking.com/index.uk.html?label=gen173nr-1DCAEoggI46AdIM1gEaOkBiAEBmAEpuAEXyAEV2AED6AEBiAIBqAIDuAKRs6yLBsACAdICJDBiYWEzODM4LWQzMjItNDU0YS1iZjVILWU4MTg5MDc5OWYzM9gCBOACAQ;sid=609fe33d187b1de4cd5d9a9f85ec50e5;keep_landing=1;sb_price_type=total;sig=v199MariZ-&
29. Wikipedia: Booking.com. [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/wiki/Booking.com>
30. Hotels24.ua. [Електронний ресурс]. – Режим доступу: <https://hotels24.ua>
31. Hotels24.ua: About. [Електронний ресурс]. – Режим доступу: <https://hotels24.ua/uk/about/>
32. innRoad. [Електронний ресурс]. – Режим доступу: <https://www.innroad.com/product/property-management-system/>
33. Hotel Connect. [Електронний ресурс]. – Режим доступу: <https://www.hotelconnect.me>
34. Hotel Connect: Features. [Електронний ресурс]. – Режим доступу: <https://www.hotelconnect.me/features>
35. Операційна система. Електронний ресурс]. – Режим доступу: https://ru.wikipedia.org/wiki/Операционная_система
36. Веб-Платформа. [Електронний ресурс]. – Режим доступу: <http://x-site.by/products/web-platform>
37. Мобільний застосунок. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Мобильний_застосунок
38. .NET vs PHP vs JAVA. [Електронний ресурс]. – Режим доступу: <https://www.intergy.com.au/net-vs-php-vs-java/>

39. .NET Framework. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/.NET_Framework
40. Чому PHP досі є лідером серед платформ. [Електронний ресурс]. – Режим доступу: <https://msiter.ru/articles/pochemu-php-do-sih-por-yavlyaetsya-liderom-sredi-platform-razrabotki-veb-prilozheniy>
41. Java (програмна платформа). [Електронний ресурс]. – Режим доступу: [https://ru.wikipedia.org/wiki/Java_\(программная_платформа\)](https://ru.wikipedia.org/wiki/Java_(программная_платформа))
42. What is Laravel? | DigitalOcean. [Електронний ресурс]. – Режим доступу: <https://www.digitalocean.com/community/tutorials/what-is-laravel>
43. Шаблонізатор Blade. [Електронний ресурс]. – Режим доступу: <https://laravel.su/docs/5.3/blade>
44. Wikipedia: API. [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Прикладний_програмний_інтерфейс
45. Booking.com API. [Електронний ресурс]. – Режим доступу: <https://english.api.rakuten.net/tipsters/api/booking-com>
46. Google Maps Geocoding. [Електронний ресурс]. – Режим доступу: <https://english.api.rakuten.net/googlecloud/api/google-maps-geocoding/details>
47. Open Weather Map. [Електронний ресурс]. – Режим доступу: <https://english.api.rakuten.net/community/api/open-weather-map>
48. Laravel Cashier (Stripe). [Електронний ресурс]. – Режим доступу: <https://laravel.com/docs/8.x/billing>

ДОДАТКИ

Додаток А

Структура бази даних вебтехнології для автоматизованої системи управління готельними комплексами



Таблиця «Users» призначена для збереження даних про клієнтів та адміністраторів й має наступні атрибути: англ: id, lastName, firstName, fatherName, birthday, email, password, phone, sex_id (таблиця 2.2).

Таблиця 2.2 – Атрибути таблиці «Users»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	lastName	nvarchar(255)	Прізвище
3	firstName	nvarchar(255)	Ім'я
4	fatherName	nvarchar(255)	По батькові
5	birthday	date	Дата народження
6	email	nvarchar(255)	Електронний адрес
7	password	nvarchar(255)	Пароль
8	phone	nvarchar(11)	Номер телефону
9	sex_id	int	Вторинний ключ (стать)

Таблиця «Sexes» призначена для збереження даних про стать клієнтів та адміністраторів й має наступні атрибути: англ: id, title (таблиця 2.3).

Таблиця 2.3 – Атрибути таблиці «Sexes»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	title	nvarchar(255)	Назва статі

Таблиця «Countries» призначена для збереження даних про країни й має наступні атрибути: англ: id, title (таблиця 2.4).

Таблиця 2.4 – Атрибути таблиці «Countries»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	title	nvarchar(255)	Назва країни

Таблиця «Hotels» призначена для збереження даних про готельні комплекси й має наступні атрибути: англ: id, title, address, star_id, price, description, mainImg, city_id, map, latitude, longitude, country_id (таблиця 2.5).

Таблиця 2.5 – Атрибути таблиці «Hotels»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	title	nvarchar(255)	Назва готельного комплексу
3	address	nvarchar(255)	Адреса готельного комплексу
4	star_id	int	Вторинний ключ (зірки)
5	price	money	Добуток ціни найдешевшої кімнати і планований період перебування (дата в'їзду та виїзду) клієнта
6	description	text	Детальна інформація про діяльність готельного комплексу
7	mainImg	nvarchar(255)	Репрезентативна фотографія

8	city_id	int	Вторинний ключ (місто)
9	map	nvarchar(255)	Фотографія готелю на онлайн-карті
10	latitude	double	Широта
11	longitude	double	Довгота
12	country_id	int	Вторинний ключ (країна)

Таблиця «Cities» призначена для збереження даних про міста й має наступні атрибути: англ: id, title (таблиця 2.6).

Таблиця 2.6 – Атрибути таблиці «Cities»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	title	nvarchar(255)	Назва міста

Таблиця «Stars» призначена для збереження даних про кількість зірок готельних комплексів й має наступні атрибути: англ: id, amount (таблиця 2.7).

Таблиця 2.7 – Атрибути таблиці «Stars»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	amount	int	Кількість зірок

Таблиця «Rates» призначена для збереження даних про рейтинг готельних комплексів й має наступні атрибути: англ: id, user_id, hotel_id, mark (таблиця 2.8).

Таблиця 2.8 – Атрибути таблиці «Stars»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	user_id	int	Вторинний ключ (клієнт)
3	hotel_id	int	Вторинний ключ (готель)
4	mark	double	Оцінка готельного комплексу

Таблиця «Hotel_images» призначена для збереження даних про фотографії готельних комплексів й має наступні атрибути: англ: id, nameImg, hotel_id (таблиця 2.9).

Таблиця 2.9 – Атрибути таблиці «Hotel_images»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	nameImg	nvarchar(255)	Фотографія готельного комплексу
3	hotel_id	int	Вторинний ключ (готель)

Таблиця «Comments» призначена для збереження даних про коментарі клієнтів стосовно готельних комплексів й має наступні атрибути: англ: id, user_id, hotel_id, message (таблиця 2.10).

Таблиця 2.10 – Атрибути таблиці «Comments»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	user_id	int	Вторинний ключ (клієнт)
3	hotel_id	int	Вторинний ключ (готель)
4	message	text	Відгук клієнта стосовно готельного комплексу

Таблиця «Roles» призначена для збереження даних про ролі користувачів у системі й має наступні атрибути: англ: id, user_id, (таблиця 2.11).

Таблиця 2.11 – Атрибути таблиці «Roles»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	user_id	int	Вторинний ключ (користувач)

Таблиця «Rooms» призначена для збереження даних про готельні номери й має наступні атрибути: англ: id, title, price, hotel_id, mainImg, mealplan (таблиця 2.12).

Таблиця 2.12 – Атрибути таблиці «Rooms»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	title	nvarchar(255)	Назва кімнати
3	price	money	Ціна кімнати за одну ніч
4	hotel_id	int	Вторинний ключ (готель)
5	mainImg	nvarchar(255)	Фотографія кімнати
6	mealplan	text	Інформація про сніданок

Таблиця «Favourite_hotels» призначена для збереження даних про список улюблених готелів клієнта й має наступні атрибути: англ: id, user_id, hotel_id (таблиця 2.13).

Таблиця 2.13 – Атрибути таблиці «Favourite_hotels»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	user_id	int	Вторинний ключ (клієнт)
3	hotel_id	int	Вторинний ключ (готель)

Таблиця «Periods» призначена для збереження даних про актуальні та доступні дати резервування й має наступні атрибути: англ: id, checkin, checkout, hotel_id (таблиця 2.14).

Таблиця 2.14 – Атрибути таблиці «Periods»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	checkin	date	Дата в'їзду
3	checkout	date	Дата виїзду

4	hotel_id	int	Вторинний ключ (готель)
---	----------	-----	-------------------------

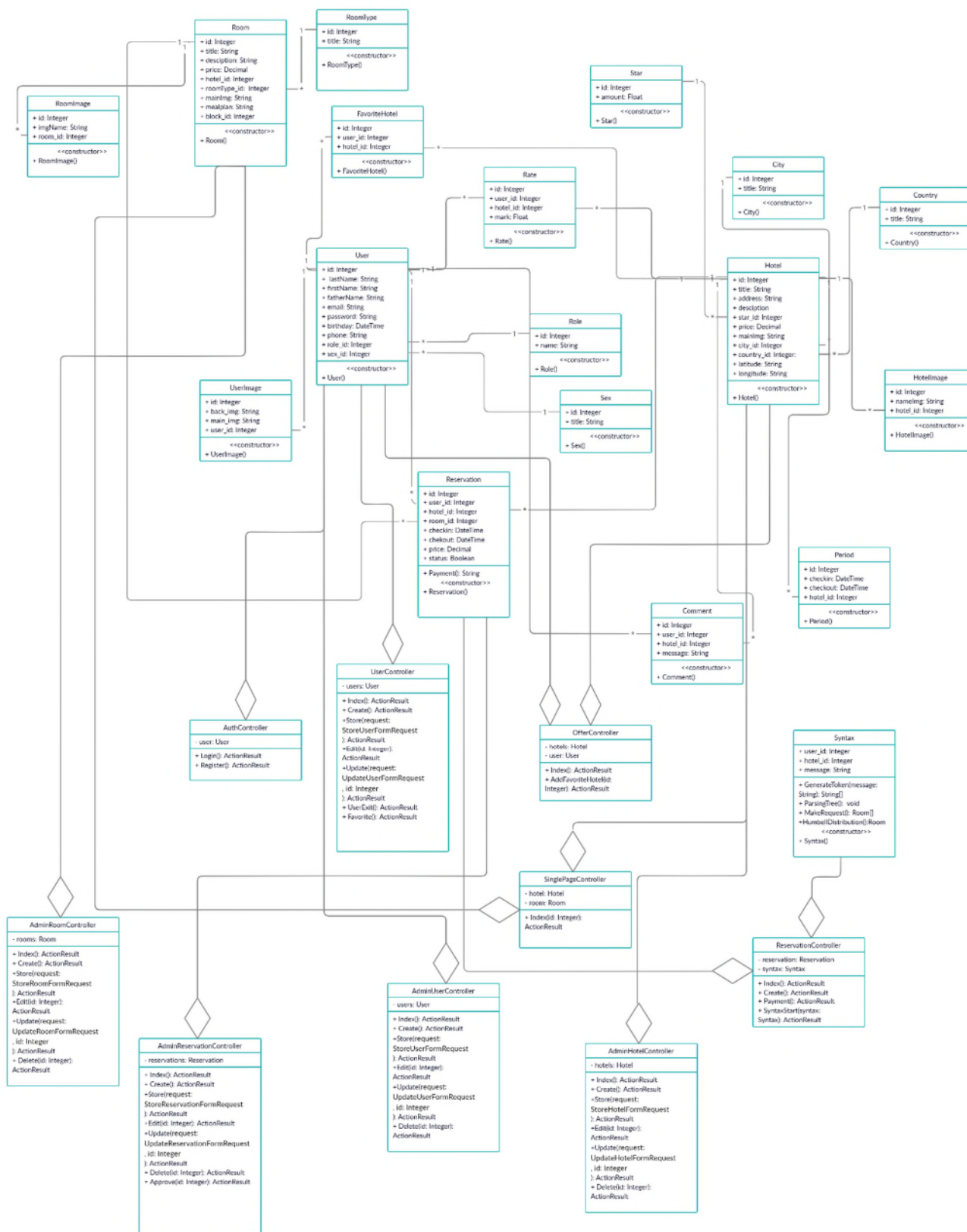
Таблиця «Reservations» призначена для збереження даних про резервування готельних номерів й має наступні атрибути: англ: id, user_id, hotel_id, room_id, checkin, checkout, price, status (таблиця 2.15).

Таблиця 2.15 – Атрибути таблиці «Reservations»

№ п/п	Назва	Тип даних	Опис
1	id	int	Унікальний ідентифікатор
2	user_id	int	Вторинний ключ (клієнт)
3	hotel_id	int	Вторинний ключ (готель)
4	room_id	int	Вторинний ключ (кімната)
5	checkin	date	Дата в'їзду
6	checkout	date	Дата виїзду
7	price	money	Загальна ціна резервування
8	status	int	Статус підтвердження бронювання

Додаток Б

Розгорнута структура класів вебтехнології для автоматизованої системи управління готельними комплексами



Додаток В

Програмні коди

```

<?php
namespace App\Http\Controllers\SinglePage;

use App\Models\Room;
use App\Models\Hotel;
use App\Models\RoomOffer;
use Illuminate\Http\Request;
use App\Services\Offer\Service;
use App\Http\Controllers\Controller;
use Illuminate\Support\Facades\Http;

class SingleIndexController extends BaseController
{
    public function index($id)
    {
        session(['hotelId'=>$id]);
        $serv = new Service();
        $resArr = $this->service->search($serv,$id);

        return view('single.home',['hotelSingle'=>$resArr[0],
            'hotelPhoto'=>$resArr[1],
            'rooms'=>$resArr[2],
            'weatherCurrent'=>$resArr[3],
            'weather'=>$resArr[4],
            'weatherLocation'=>$resArr[5],
            'serviceOffers'=>$resArr[6],
            'roomImages'=>$resArr[7]]);
    }

    public function search(Request $request)
    {
        $userMessage = $request->input('searchRoom');
        $userMessage = explode(' ', $userMessage);

        $wordsWithStar = [];
        $wordsWithAmpersand = [];
        $wordsWithSignPrice = [];

        $keyWords = [];
        $temp = '';
        $count = 1;

        for($i = 0; $i<count($userMessage); $i++)
            if($userMessage[$i][0] == '*')
            {
                if(stripos($userMessage[$i], '&') != false)
                {
                    $pos = stripos($userMessage[$i], '&');
                    $wordsWithStar[] =
                        substr($userMessage[$i], 1, $pos);
                    $wordsWithAmpersand[] =
                        substr($userMessage[$i], $pos);
                }
                else
                {
                    $wordsWithStar[] =
                        substr($userMessage[$i], 1);
                }
            }
            else if($userMessage[$i][0] == '&')
            {
                if(stripos($userMessage[$i], '*') != false)
                {
                    $pos = stripos($userMessage[$i], '*');
                    $wordsWithStar[] =
                        substr($userMessage[$i], $pos);
                    $wordsWithAmpersand[] =
                        substr($userMessage[$i], 1, $pos);
                }
                else
                {
                    $wordsWithAmpersand[] =
                        substr($userMessage[$i], 1);
                }
            }
            else if($userMessage[$i][0] == '>' ||
                $userMessage[$i][0] == '<' || ($userMessage[$i][0] == '>' &&
                $userMessage[$i][1] == '=') || ($userMessage[$i][0] == '<' &&
                $userMessage[$i][1] == '=') || $userMessage[$i][0] == '=')
            {
                $wordsWithSignPrice[] = $userMessage[$i];
            }
            else if(stripos($userMessage[$i], ':') != false)
            {

```

```

    $pos = strpos($userMessage[$i], ':');

    $wordsTemp = substr($userMessage[$i], $pos+1);
    $wordsArr = explode(' ', $wordsTemp);

    for($k = 0; $k < count($wordsArr); $k++)
    {
        if($wordsArr[$k][0] == '*')
        {
            $wordsWithStar[] =
substr($wordsArr[$k], 1);
        }
        else if($wordsArr[$k][0] == '*&')
        {
            $wordsWithAmpersand[] =
substr($wordsArr[$k], 1);
        }
        else
        {
            $keyWords[] = $wordsArr[$k];
        }
    }
}

}

$temp = preg_replace('/[*&.:,]/', ' ', $keyWords);
$signStarArr = preg_replace('/[*&.:,]/', ' ',
$wordsWithStar);
$ampersandUnsignedWords = preg_replace('/[*&.:,]/', '
', $wordsWithAmpersand);
$res = [];
$resStars = [];
$resultAmpersand = [];

$keyWordsDig = [];

for($i = 0; $i < count($temp); $i++)
{
    $tmp = explode(' ', trim($temp[$i]));
    array_push($res, ...$tmp);
}

for($i = 0; $i < count($signStarArr); $i++)
{
    $tmp = explode(' ', trim($signStarArr[$i]));
    array_push($resStars, ...$tmp);
}

for($i = 0; $i < count($res); $i++)
{
    if($res[$i] == '')
    {
        continue;
    }
    else
    {
        $keyWordsDig[] = $res[$i];
    }
}

$rooms =
Room::where('hotel_id', session('hotelId'))->get();

$resulter = [];
$starResulter = [];
$ampersandResulter = [];
$count = 0;
$countStar = 0;
$countAmpersand = 0;

for($i = 0; $i < count($rooms); $i++)
{
    for($j = 0; $j < count($resStars); $j++)
    {
        $tmp = RoomOffer::select('room_id')-
>where('room_id', $rooms[$i]->id)-
>where('title', 'LIKE', "%{$resStars[$j]}%")->get();

        array_push($starResulter, ...$tmp);

        if(count($starResulter) > $countStar)
        {

```



```

        if($lengthSignPrice > 1)
        {
            $tmp = Room::where('id',$starResultler[$i]-
                if($wordsWithSignPrice[$j][0] >room_id)->get();
            == '>' || $wordsWithSignPrice[$j][0] == '<')
                array_push($withoutPriceStars, ...$tmp);
            {
                $tmp =
            }
            Room::where('id',$starResultler[$i]->room_id)-
            >where([[ 'price', '>', $priceSignMany[0]], [ 'price', '<', $priceSig
            nMany[1]])->get();
            array_push($withPriceStars
            , ...$tmp);
        }
        else
        {
            $tmp =
            Room::where('id',$starResultler[$i]->room_id)-
            >where([[ 'price', '>=', $priceSignMany[0]], [ 'price', '<=', $priceS
            ignMany[1]])->get();
            array_push($withPriceStars
            , ...$tmp);
        }
    }
    else if($lengthSignPrice == 1)
    {
        $price =
        substr($wordsWithSignPrice[$j],1);
        $tmp =
        Room::where('id',$starResultler[$i]->room_id)-
        >where('price',$wordsWithSignPrice[$j][0],(float)$price)-
        >get();
        array_push($withPriceStars,
        ...$tmp);
    }
}
else
{
    for($i = 0; $i<count($starResultler); $i++)
    {
        if($wordsWithSignPrice[$j][0]
            == '>' || $wordsWithSignPrice[$j][0] == '<')
            {
                $tmp =
                Room::where('id',$ampersandResultler[$i]->room_id)-
                >where([[ 'price', '>', $priceSignMany[0]], [ 'price', '<', $priceSig
                nMany[1]])->get();
                array_push($withPriceAmper
                sand, ...$tmp);
            }
            else
            {
                $tmp =
                Room::where('id',$ampersandResultler[$i]->room_id)-
                >where([[ 'price', '>=', $priceSignMany[0]], [ 'price', '<=', $priceS
                ignMany[1]])->get();
                array_push($withPriceAmper
                sand, ...$tmp);
            }
        }
    }
    else if($lengthSignPrice == 1)

```

```

        {
            $price =
            substr($wordsWithSignPrice[$j],1);
            $tmp =
            Room::where('id',$ampersandResultler[$i]->room_id)-
            >where('price',$wordsWithSignPrice[$j][0],(float)$price)-
            >get();
            array_push($withPriceAmpersand
            , ...$tmp);
        }
    }
}
else
{
    for($i = 0; $i<count($ampersandResultler);
    $i++)
    {
        $tmp =
        Room::where('id',$ampersandResultler[$i]->room_id)->get();
        array_push($withoutPriceAmpersand,
        ...$tmp);
    }

    if(count($wordsWithSignPrice) > 0)
    {
        array_push($selectedRooms, ...$withPriceStars);
    }
    else
    {
        array_push($selectedRooms, ...$withoutPriceStars);
    }

    if(count($wordsWithSignPrice) > 0)
    {
        array_push($selectedRooms,
        ...$withPriceAmpersand);
    }
    else
    {
        array_push($selectedRooms,
        ...$withoutPriceAmpersand);
    }
}
}
}

        array_push($selectedRooms,
        ...$withoutPriceAmpersand);
    }
}

    for($i = 0; $i<count($resultler); $i++)
    {
        if(count($wordsWithSignPrice)>0)
        {
            for($j = 0; $j<count($wordsWithSignPrice);
            $j++)
            {
                if($wordsWithSignPrice[$j][0] == '>' ||
                $wordsWithSignPrice[$j][0] == '<' ||
                ($wordsWithSignPrice[$j][0] == '>' &&
                $wordsWithSignPrice[$j][1] == '=') ||
                ($wordsWithSignPrice[$j][0] == '<' &&
                $wordsWithSignPrice[$j][1] == '=') ||
                $wordsWithSignPrice[$j][0] == '=')
                {
                    if($lengthSignPrice > 1)
                    {
                        if($wordsWithSignPrice[$j][0] ==
                        '>' || $wordsWithSignPrice[$j][0] == '<')
                        {
                            $tmp =
                            Room::where('id',$resultler[$i]->room_id)-
                            >where([[ 'price', '>', $priceSignMany[0]], [ 'price', '<', $priceSig
                            nMany[1]]])->get();
                            array_push($selectedRooms,
                            ...$tmp);
                        }
                        else
                        {
                            $tmp =
                            Room::where('id',$resultler[$i]->room_id)-
                            >where([[ 'price', '>=', $priceSignMany[0]], [ 'price', '<=', $priceS
                            ignMany[1]]])->get();
                            array_push($selectedRooms,
                            ...$tmp);
                        }
                    }
                    else if($lengthSignPrice == 1)

```

```

        {
            $price =
substr($wordsWithSignPrice[$j],1);
            $tmp =
Room::where('id',$resulter[$i]->room_id)-
>where('price',$wordsWithSignPrice[$j][0],(float)$price)-
>get();
            array_push($selectedRooms,
...$tmp);
        }
    }

}

else
{
        $tmp = Room::where('id',$resulter[$i]-
>room_id)->get();
        array_push($selectedRooms, ...$tmp);
    }
}

return redirect()->back()->with('selectedRooms',
$selectedRooms);
}
}

```

Додаток Г

Презентаційний матеріал

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Вебтехнологія для автоматизованої системи управління готельними комплексами

Виконав: студент 4 курсу, група КН-18-1, Е.Р. Муляр

Керівник: к.т.н., доцент кафедри КН, Р.О. Багрій

Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

Актуальність

На сьогоднішній день існує величезна кількість різноманітних вебсистем для здійснення резервувань готельних номерів. Основним недоліком систем даного типу є проблеми з пошуком номерів, які б максимально точно задовольняли би потреби клієнта. У більшості випадків клієнт витрачає більшу кількість часу на перегляд списку кімнат, які пропонує відповідний готельний комплекс для того, щоб знайти кімнату, яка найбільш відповідним чином задовольнить його потреби.

Створення вебтехнології, яка дозволить підібрати кімнату або набір кімнат для клієнта максимально швидко та точно до його побажань є актуальною задачею:

- Вебтехнологія включає в себе модуль, який використовуючи текстову інформацію клієнта дозволить за певним алгоритмом підібрати набір кімнат, які найбільш відповідним чином відповідатимуть заданим побажанням;
- Вебтехнологія включає в себе модулі, які дозволять перетворити будь-яку локальну систему резервувань кімнат в повноцінну вебсистему з різноманітним, гнучким та зручним функціоналом.

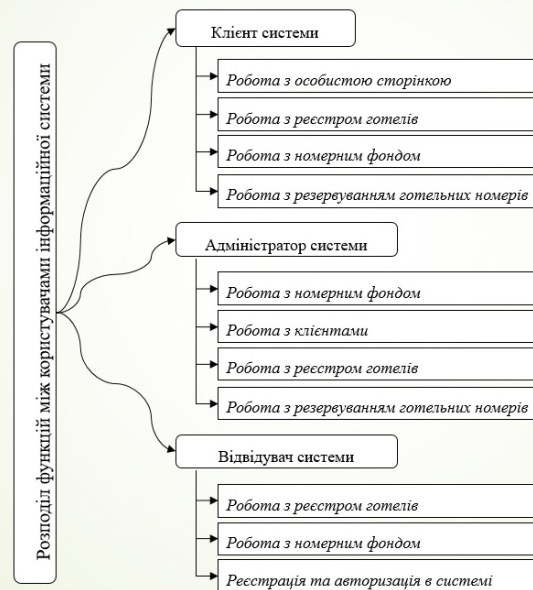
Активация Windows
Чтобы активировать Windows, перейдите в раздел
"Параметры".

Завдання

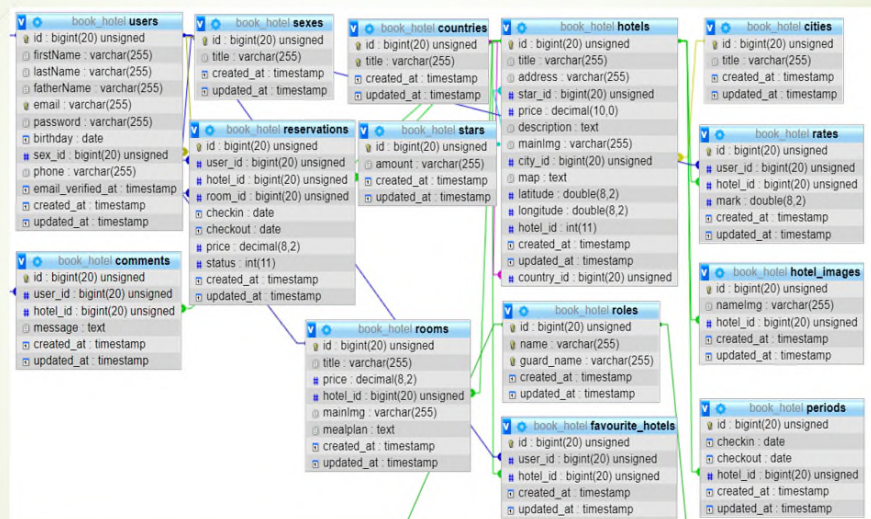
Метою кваліфікаційної роботи бакалавра є розробка вебтехнології для автоматизованої системи управління готельними комплексами. Для досягнення поставленої мети необхідно реалізувати виконання наступних задач:

- розробити функціональну та інформаційну структуру системи управління готельними комплексами;
- розробити інформаційну модель: структуру БД та її об'єктно-реляційну проєкцію;
- розробити метод підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів;
- спроектувати та розробити вебтехнологію з можливістю підбору готельних номерів зі словесного опису побажань клієнтів.

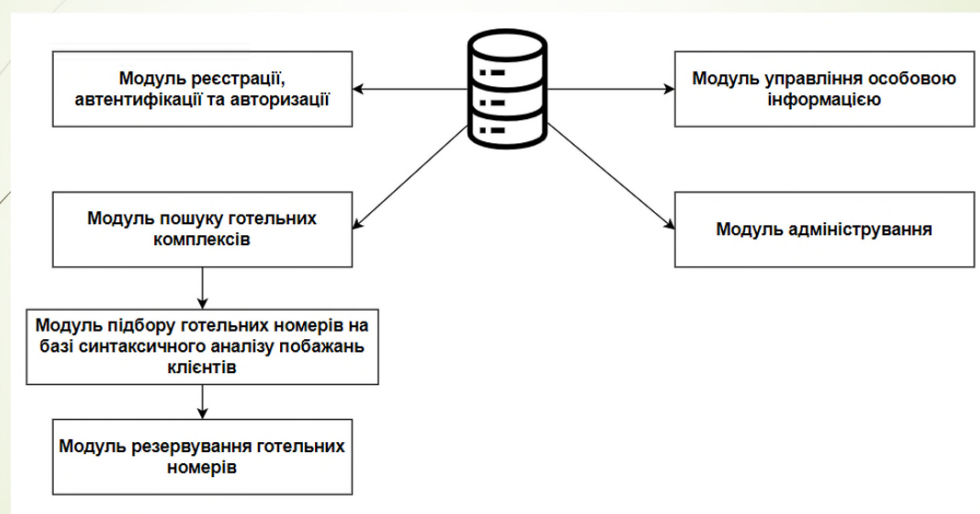
Функціональна структура системи



Даталогічна модель бази даних



Архітектура вебтехнології



Структура модуля підбору готельних номерів



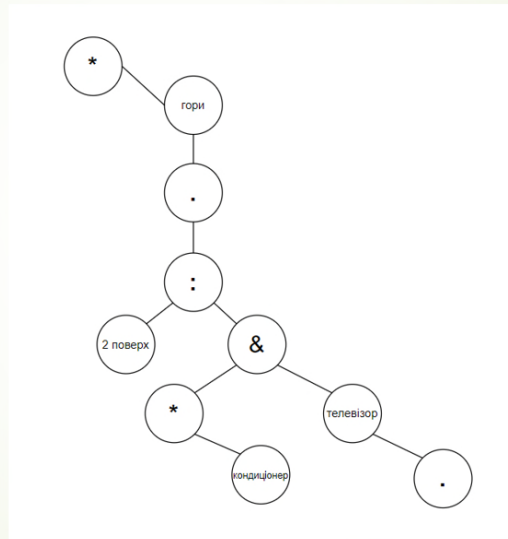
Лексичний аналіз вхідних даних

Першим етапом у функціонуванні даного методу є лексичний аналіз, за допомогою якого потік вхідних даних (текст) поділяється на значущі набори символів (токени), визначені граматикою регулярних виразів. Для прикладу користувач ввів наступний текст: «Я хотів би, щоб з вікна було видно *гори. Також є побажання: кімната повинна знаходитись на 2 поверсі, в кімнаті повинні бути присутні *кондиціонер&телевізор.» В даному тексті зустрічаються наступні спеціальні символи («*», «&», «<, >, >=, <=, =», «:», «(,», «.)), ці символи є визначені граматикою регулярних виразів (усі правила використання спеціальних символів будуть завчасно відображенні користувачу).

Кожен з цих символів має своє визначення та пріоритет:

- «*» - вказує на важливість слова. Має найвищий пріоритет (1);
- «&» - вказує на неявну важливість побажання. Має пріоритет (2);
- «<, >, >=, <=, =» - слугують для вказання діапазону цін готельних номерів. Мають пріоритет (3);
- «:» - вказує на те, що далі буде йти перерахування побажань (слів) клієнта. Має пріоритет (4);
- «(,» - слугує для відокремлення набору побажань (має сенс лише у контексті з символом «:»). Має пріоритет (5);
- «.» - вказує на закінчення речення. Має найнижчий пріоритет (6).

Дерево розбору



Генерація запиту до бази даних

Запит до бази даних утворюється опираючись на дерево розбору та використовуючи ORM Eloquent. Процес побудови запиту відбувається наступним чином:

1. Обхід дерева розбору використовуючи цикл;
2. Побудова структури запиту використовуючи елементи ORM;
3. Додавання до структури значення дерева розбору.

Після побудови здійснюється запит до бази даних, в результаті чого буде отримано набір відповідних готельних номерів.

Визначення найбільш відповідного готельного номеру

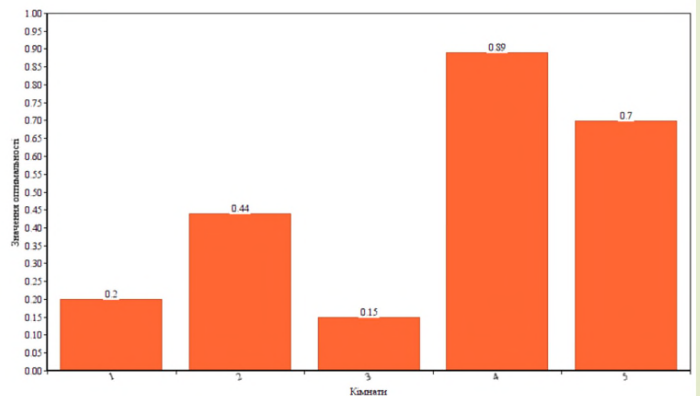
Алгоритм визначення найбільш відповідної кімнати серед набору кімнат базується на теорії ймовірностей. Використання принципів та алгоритмів теорії ймовірностей дозволить легко створити алгоритм обчислення вибірки, а також забезпечити мінімальну затрату продуктивності системи. Для обчислення вибірки було обрано принцип «розподіл Гумбеля».

Розподіл Гумбеля (подвійний експоненціальний розподіл) є граничним розподілом, який використовується для визначення найбільшого елемента вибірки випадкової величини. Даний розподіл ідеально підходить для вирішення поставленої задачі. Розподіл Гумбеля розраховується за наступною формулою:

$$f(x) = \frac{1}{\beta} \exp \left[\frac{a-x}{\beta} - \exp \left(\frac{a-x}{\beta} \right) \right]$$

Визначення найбільш відповідного готельного номеру

Обрахувавши параметри кожної кімнати за даним розподілом, буде отримано впорядкований низхідний набір кімнат, де кожна кімната має своє значення від 0 до 1. Тобто вверху списку буде знаходитися кімната, яка найбільш відповідним чином відповідає поставленим побажанням клієнта.



Демонстрація роботи програми

Правила для написання коректного запиту:

- * - спец. символ, який вказує на обов'язковість слова (побаження), має найвищий пріоритет 1, приклад: *приклад
- & - спец. символ, який вказує на наявну обов'язковість слова (побаження), має пріоритет 2, приклад: &приклад
- <>>=<=<=> - спец. символи, які слугують для вказання діапазону цін готельних номерів, мають пріоритет 3, приклад: >1500 <=2000
- :- спец. символ, який вказує на перечислення побажань, має пріоритет 4, приклад: номер:душ,ванна

Я хочу зарезервувати кімнату. Вид з кімнати повинен виходити на &місто. В номері повинен бути присутній:*біде,кондиціонер. Ціна <3000

ПІДБРАТИ



Розкішні апартаменти

UAN 2389

Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

ЗАБРОНЮВАТИ ...



Двомісний номер економ-класу з 2 односпальними ліжками

UAN 1066

Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

ЗАБРОНЮВАТИ ...



Стандартний двомісний номер з 2 односпальними ліжками

UAN 1552

Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

ЗАБРОНЮВАТИ ...



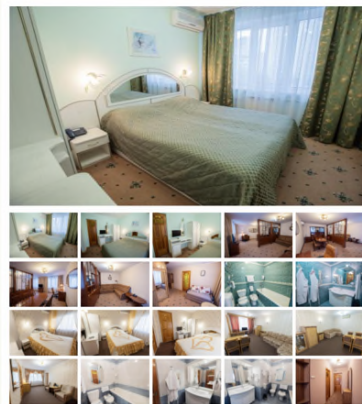
Напівлюкс

UAN 1705

Континентальний сніданок пропонується за ціною UAH 150 з особи за ніч.

ЗАБРОНЮВАТИ ...

Демонстрація роботи програми



Розкішні апартаменти

Апартаменти обладнано кондиціонером, телевізором із супутниковим телебаченням, робочим столом та окремою ванною кімнатою. У розпорядженні гостей також перебуватиме окрема вітальня та додаткова ванна кімната.

Ванна кімната:

- Душ
- Гостьовий туалет
- Туалетний папір
- Туалетно-косметичні засоби
- Біде
- Капці
- Кондиціонер для волосся
- Рушники
- Ванна
- Фен
- Халат
- Туалет
- Власна ванна кімната
- Шампунь
- Мило для тіла

Вид:

- Вид на місто

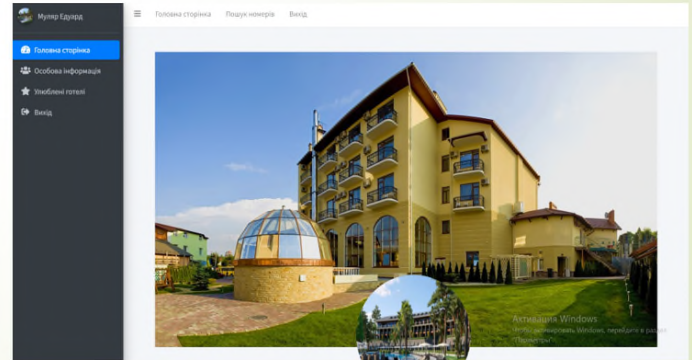
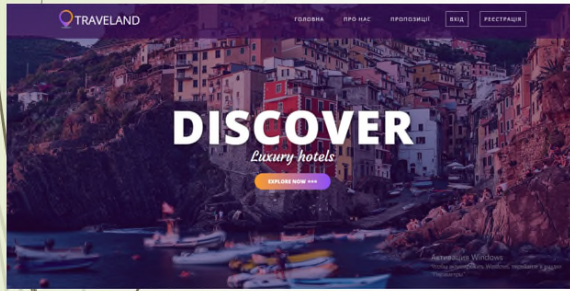
Медіа та технології:

- Кабельні канали
- Телевізор
- Телефон
- Телевізор з плоским екраном

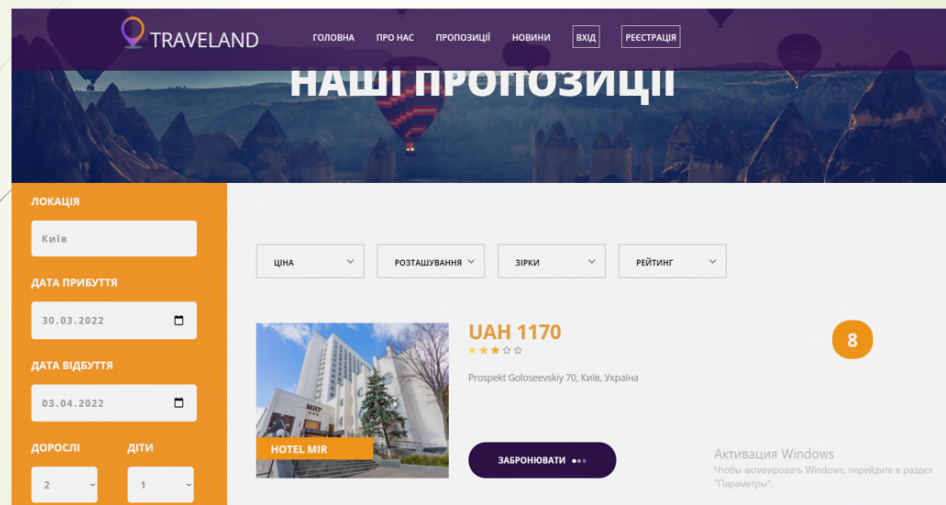
Зручності в номері:

- Датчик диму
- Вид з електронною картою
- Опалення
- Синтетична подушка
- Кондиціонер
- Звукоізоляція
- Кошики для сміття

Демонстрація роботи програми



Демонстрація роботи програми



Демонстрація роботи програми

Novotel Amsterdam City
★★★★☆
Europaboulevard 10, Амстердам, Нідерланди

[ЗАБРОНЮВАТИ](#)

[Подивіться на карті](#)

Активізація Windows

Цей чотиризірковий готель розташований менш ніж за 10 хвилин ходьби від Центру виставок і конгресів RAI. До послуг гостей сучасні номери з безкоштовним Wi-Fi. Крім того, у готелі Novotel працюють тренажерний зал і сауна. Для гостей, які подорожують на автомобілі, на території облаштовано простору паркування. У кожному просторому номері готелю Novotel Amsterdam City облаштовано зону відпочинку. Серед інших сучасних зручностей: телевизор із 55-дюймовим плоским екраном, міні-бар, кавоарка Nespresso та прилади для приготування чаю. За додаткову плату щоранку гостям сервірують континентальний сніданок. Також у готелі працює ресторан із широким вибором сезонних страв по меню. До послуг гостей прокат велосипедів. Готель розміщений неподалік від парку Амстелпарк і за 5 хвилин ходьби від станції метро лінії Північ-Південь, звідси можна без пересадок дістатися до центру міста. Поїздка на автомобілі або громадському транспорті до аеропорту Схіпгол триватиме не більш ніж 15 хвилин.

Демонстрація роботи програми

РОЗМІЩЕННЯ НА КАРТІ

ПОГОДА

Saturday
28 May, 2022 Amsterdam, Netherlands

12 °C
Breezy

28, Sat	29, Sun	30, Mon	31, Tue	1, Wed	2, Thu	3, Fri
☀	☁	☁	☁	☁	☁	☁
15° / 10°	13° / 8°	13° / 6°	17° / 7°	20° / 9°	17° / 17°	20° / 20°

Активізація Windows
Щоб активувати Windows, перейдіть в «Параметри».

[🏠](#)
[🚗](#)
[🚲](#)

- Стандартний номер із 1 двоспальним ліжком розміру "queen-size"**
UAN 681
Континентальний сніданок пропонується за ціною UAN 664 з особи за ніч.

[ЗАБРОНЮВАТИ](#)
- Оновлений стандартний номер із ліжком розміру queen-size**
UAN 713
Континентальний сніданок пропонується за ціною UAN 664 з особи за ніч.

[ЗАБРОНЮВАТИ](#)
- Стандартний номер із 1 двоспальним ліжком розміру "queen-size"**
UAN 738
Континентальний сніданок пропонується за ціною UAN 664 з особи за ніч.

[ЗАБРОНЮВАТИ](#)
- Оновлений стандартний номер із ліжком розміру queen-size**
UAN 770
Континентальний сніданок пропонується за ціною UAN 664 з особи за ніч.

[ЗАБРОНЮВАТИ](#)

Активізація Windows
Щоб активувати Windows, перейдіть в «Параметри».

Демонстрація роботи програми

Прізвище	Ім'я	По батькові
Муляр	Едуард	Антонович
Дата народження		
19.01.2001		
Електронна адреса		
ediktest@gmail.com		
Номер телефону		
0961324567		
Готель		
Novotel Amsterdam City		
Кімната		
Оновлений стандартний номер із ліжком розміру queen-size		
Активация Windows		
Дата в'їзду	Дата виїзду	
04.06.2022	07.06.2022	
Ціна		
UAN 712.94		
ЗАБРОНЮВАТИ		


Демонстрація роботи програми

Головна панель


- Готелі
- Клієнти
- Бронювання
- Вихід

Головна сторінка Пошук номерів Вихід


Готелі



Клієнти



Бронювання



Висновки

Під час виконання кваліфікаційної роботи бакалавра проведено аналіз предметної області та визначено основні параметри для розв'язку поставленої задачі. Описано технології, методи та сервіси, які використовувались при розробці системи управління готельними комплексами з визначенням програмної платформи та середовищ в якому система розроблялась та функціонує.

Відповідно до отриманого результату, можна зробити висновок, що розроблена вебтехнологія для автоматизованої системи управління готельними комплексами працює коректно, а вимоги технічного завдання виконанні в повному обсязі.



**Дякую за
увагу!**

Anti-Plagiarism v-15.257

Максимальное совпадение с одним документом 3.0%

Словари проверки: en_US, ru_RU, ua_UA. **Ошибок в документах: 10%**

ID: 103896 Название: КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА Вебтехнологія для автоматизованої системи управління готельними комплексами Добавлено в БД: 2022-05-24 Авторы: Е.Р. Муляр Руководители: Р.О. Багрій Консультанты: Оponentы:	Документ		Суммарное совпадение по Базе Данных	
	Символы	Лексемы	Символы	Лексемы
	88271	1333	6348 (7%)	98 (7%)

Источник плагиата

ID	Описание	Наличие плагиата в документе	
		Символы	Лексемы

Ім'я користувача:
Кафедра КН

ID перевірки:
1011323162

Дата перевірки:
24.05.2022 18:30:22 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
24.05.2022 18:32:08 EEST

ID користувача:
100005671

Назва документа: Муляр_ЗАПИСКА_short

Кількість сторінок: 109 Кількість слів: 13824 Кількість символів: 112044 Розмір файлу: 4.63 MB ID файлу: 1011209601

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

10.1% Схожість

Найбільша схожість: 1.95% з Інтернет-джерелом (<https://lifar-o-a.at.ua/gotel/Lekcii/lekcija-2.pdf>)

7.46% Джерела з Інтернету

257

Сторінка 111

3.46% Джерела з Бібліотеки

123

Сторінка 113

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0% Вилучень

Немає вилучених джерел

Модифікації

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

Замінені символи

1

Підозріле форматування

39
сторінок

РІШЕННЯ ЕКСПЕРНОЇ КОМІСІЇ
КАФЕДРИ КОМП'ЮТЕРНИХ НАУК
ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ ЗА РЕЗУЛЬТАТАМИ
АНАЛІЗУ ЗВІТУ ПОДІБНОСТІ

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

Назва: Вебтехнологія для автоматизованої системи управління готельними комплексами

Автор: студент групи КН-18-1 Муляр Едуард Русланович

Спеціальність: 122 – Комп'ютерні науки

Освітня програма: освітньо-професійна

Науковий керівник: к.т.н., доц. Багрій Р.О.

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) за програмою Anti-Plagiarism виявлені 3% є фрагментарними – містять поширені конструкції, загальновідомі терміни, скорочення та визначення.
- 2) За програмою UNICHECK виявлені 10.1%, що є запозиченнями, які розміщені в розділах аналізу існуючих технологій та прототипів, які не описують безпосередньо авторське дослідження і не стосуються результатів роботи.


Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ідентичності/схожості, складає 3% і 10.1% відповідно, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.


Керівник роботи

Гарант ОП

Завідувач кафедри КН







Руслан Багрій

Олександр Мазурець

Олександр Бармак



РЕЦЕНЗІЯ

на кваліфікаційну роботу бакалавра

студента гр. КН-18-1 Муляра Едуарда Руслановича

за темою: Вебтехнологія для автоматизованої системи управління готельними комплексами

1. Актуальність обраної теми

На сьогоднішній день існує величезна кількість різноманітних вебсистем для здійснення резервувань готельних номерів. Основним недоліком систем даного типу є проблеми з пошуком номерів, які б максимально точно задовольняли би потреби клієнта. У більшості випадків клієнти витрачають більшу кількість свого вільного часу на пошук та перегляд списку кімнат, які пропонує відповідний готельний комплекс з метою знайти кімнату, яка найбільш відповідним чином задовольнить його потреби. Створення вебтехнології, яка дозволить підібрати кімнату або набір кімнат для клієнта швидко та максимально точно до його побажань є актуальною задачею.

2. Повнота розкриття мети та завдань роботи

Метою кваліфікаційної роботи бакалавра було розробити вебтехнологію для автоматизованої системи управління готельними комплексами. Для досягнення поставленої мети реалізовано виконання наступних задач:

- розроблено функціональну та інформаційну структуру системи управління готельними комплексами;
- розроблено інформаційну модель: структуру бази даних та її об'єктно-реляційну проєкцію;
- розроблено метод підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів;
- спроєктовано та розроблено вебтехнологію з можливістю підбору готельних номерів зі словесного опису побажань клієнтів.

Усі завдання опрацьовані в повному обсязі, що дозволяє зробити висновок про повноту опрацювання теми роботи.

3. Зміст кожного розділу роботи

Перший розділ присвячений проведенню аналізу предметної області та визначенню основних параметрів для розв'язку поставленої задачі.

Другий розділ присвячений проєктуванню функціональної структури інформаційної системи та вебтехнології.

Третій розділ присвячений програмній реалізації спроєктованої вебтехнології та функціональної структури інформаційної системи.

4. Оцінка розробленої інформаційної системи, її практична цінність

Спроєктована та розроблена інформаційна система дозволяє здійснювати підбір готельних номерів опираючись на побажання клієнтів, реєстрацію та автентифікацію, пошук готельних комплексів, резервування готельних номерів, а також адміністрування. Цінність даної системи полягає в повній автоматизації її функціональних можливостей, а також у можливості функціонувати у Інтернет середовищі, тим саме забезпечуючи певну кроссплатформеність.

5. Якість оформлення кваліфікаційної роботи бакалавра

Робота виконана на належному науково-методичному рівні та відповідає встановленим вимогам щодо оформлення такого роду праць.

6. Недоліки кваліфікаційної роботи бакалавра

Кваліфікаційна робота бакалавра виглядала б привабливіше, якби у ній було розглянуто більше математичних підходів до визначення найбільш відповідного готельного номеру.

7. Загальний висновок (допускається чи не допускається до захисту), та оцінка на яку заслуговує кваліфікаційна робота.

Враховуючи рівень виконання та забезпечення усіх необхідних вимог, робота може бути допущена до захисту. Рекомендована оцінка «відмінно».

Рецензент професор, д.т.н., комп'ютер. _____
к.інженерії ХНУ Лисенко Сергій Миколайович



ВІДГУК КЕРІВНИКА

на кваліфікаційну роботу бакалавра

гр. КН-18-1 Муляра Едуарда Руслановича за темою: Вебтехнологія для автоматизованої системи управління готельними комплексами

1. Актуальність теми

Актуальність теми достатньо обґрунтована, оскільки створення вебтехнології дозволить автоматизувати систему управління готельними комплексами. Особливістю теми є можливість підбору готельних номерів за словесним описом побажань клієнтів.

2. Відповідність роботи предметній області спеціальності 122 Комп'ютерні науки та загальним вимогам до наукових робіт

Теми кваліфікаційної роботи "Вебтехнологія для автоматизованої системи управління готельними комплексами" відповідає предметній області спеціальності 122 Комп'ютерні науки та вимогам до кваліфікаційної роботи бакалавра, оскільки метою роботи є розробка веб-технології для автоматизованої системи управління готельними комплексами та методу підбору готельних номерів на базі синтаксичного аналізу побажань клієнтів. При вирішенні поставленої задачі використано математичні моделі, методи та алгоритми розв'язання теоретичних і прикладних задач, що виникають при розробці інформаційних технологій.

3. Професійні та особистісні якості магістранта

Муляр Е.Р. під час роботи над кваліфікаційною роботою бакалавра продемонстрував високий рівень знань та умінь за спеціальністю "Комп'ютерні науки".

4. Ступінь самостійності під час виконання кваліфікаційної роботи

Робота виконана самостійно, академічного плагіату не виявлено, стосовно всіх запозичень наведено відповідні посилання на джерела.

5. Ступінь оволодіння методами дослідження

Студент Муляр Е.Р. показав достатній рівень компетентностей та володіння необхідними інструментами та обладнанням, методами, методиками та технологіями предметної області комп'ютерних наук.

6. Повнота та якість розкриття теми роботи

Мета роботи повністю розкрита, виконані усі поставлені задачі та розроблено програмне забезпечення для підтвердження запропонованого метода.

7. Логічність, послідовність, аргументованість, літературна грамотність викладу матеріалу

Викладення матеріалу логічне, послідовне та аргументоване. Мова і стиль викладення кваліфікаційної роботи відповідають стандартам, що забезпечує доступність сприймання матеріалу і відповідає вимогам до сучасних наукових робіт.

8. Можливість практичного застосування кваліфікаційної роботи, окремих її частин

Може мати практичне значення при управлінні готельними комплексами.

9. Висновок про можливість допуску кваліфікаційної роботи до захисту, на яку оцінку заслуговує робота

Вважаю, що кваліфікаційна робота студента Муляра Едуарда Руслановича може бути рекомендована до захисту та заслуговує на оцінку "відмінно".

Керівник _____  к.т.н., доц. Руслан Багрій