

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

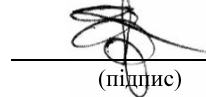
" ____ " _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему: Системний аналіз користувацьких сценаріїв та оптимізація UX
вебзастосунків

Виконав: студент групи 6145 м



(підпис)

Леус С.О.

Керівник роботи:

доцент кафедри ІУСТ, к.т.н.,

(посада, науковий ступінь вчене звання)



(підпис)

Ворона М.В.

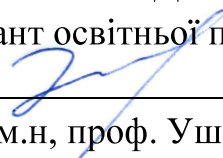
м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 124 "Системний аналіз"
Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми


д.ф-м.н, проф. Ушкац М.В.

"___" _____ 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Леусу Сергію Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Системний аналіз користувацьких сценаріїв та оптимізація UX вебзастосунків

Керівник роботи Ворона Михайло Владиславович

Затверджені наказом ректора № 1355-уч. від "14" листопада 2025 року.

2. Термін подання роботи: 15 грудня 2025 р.

3. Вихідні дані до проєкту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

Дослідження користувацьких сценаріїв та оптимізація UX вебзастосунків.
Проектування ІС користувацьких сценаріїв та оптимізація UX вебзастосунків.
Розробка ІС користувацьких сценаріїв та оптимізація UX вебзастосунків.
Техніко-економічне обґрунтування розробки. Охорона праці. Охорона навколишнього середовища

5. Перелік презентаційних матеріалів Постановка задачі; Існуюча схема функціонування ІС; Концепція системи; Діаграма варіантів використання ІС; 6-Фізична модель бази даних; Діаграма розгортання ІС; Екранні форми розробки; Апробація дослідження; Висновки.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Ворона М.В. викладач	05.09.2024	22.12.2024
2	Ворона М.В. викладач	23.12.2024	15.06.2025
3	Ворона М.В. викладач	15.06.2025	25.09.2025
4	Ворона М.В. викладач	25.09.2025	14.11.2025
5	Ворона М.В. викладач	25.09.2025	14.11.2025
6	Ворона М.В. викладач	25.09.2025	14.11.2025

7. Дата видачі завдання 05 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2024	
2	Розробка концепції	15.03.2025	
3	Розділ з розробки проєктних рішень по системі (та її частинах)	15.06.2025	
4	Реалізація проєкту та розробка робочої документації	25.09.2025	
5	Обчислення економічної ефективності	14.11.2025	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2025	
7	Оформлення пояснювальної записки	14.11.2025	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2025	
9	Подання магістерської роботи рецензенту	20.11.2025	
10	Подання магістерської роботи на державний захист	15.12.2025	

Студент


(підпис)

Леус С.О.
(ПІБ)

Керівник роботи


(підпис)

Ворона М.В.
(ПІБ)

АНОТАЦІЯ

Магістерська робота присвячена дослідженню та вдосконаленню користувацького досвіду (UX) вебзастосунків на основі системного аналізу користувацьких сценаріїв. У роботі розглянуто теоретичні засади UX-дизайну, методи аналізу поведінки користувачів та підходи до побудови інтерфейсів, орієнтованих на потреби та очікування цільової аудиторії. Особливу увагу приділено моделюванню сценаріїв взаємодії (User Flow, Customer Journey Map), аналізу навігаційних структур, когнітивного навантаження та визначенню критичних точок, що впливають на якість взаємодії.

У роботі проведено збір та аналіз UX-даних, включаючи юзабіліті-тестування, A/B-експерименти та інструменти аналітики користувацької поведінки. На основі отриманих результатів розроблено рекомендації щодо оптимізації інтерфейсу, підвищення ефективності виконання ключових дій, покращення доступності та адаптивності вебзастосунку. Запропоновано прототип оновленої інтерфейсної моделі, що демонструє застосування розроблених підходів на практиці.

Результати дослідження можуть бути використані під час проєктування сучасних вебзастосунків, удосконалення існуючих цифрових сервісів, а також у процесі навчання фахівців з UX/UI-дизайну та Web-інженерії. Робота підтверджує, що системний аналіз у поєднанні з UX-методологіями дозволяє значно підвищити якість взаємодії користувачів зі складними вебсистемами.

Ключові слова: системний аналіз, сценарії користувачів, UX-оптимізація, зручність використання, веб-застосунки, користувацький досвід, аналітика поведінки користувачів, дизайн взаємодії, UI/UX, взаємодія людини з комп'ютером.

ABSTRACT

The master's thesis focuses on the systematic analysis of user scenarios and the optimization of user experience (UX) in modern web applications. The research examines theoretical foundations of UX design, methods of analyzing user behavior, and approaches to creating user-centered interfaces. Particular attention is given to modeling interaction scenarios (User Flows, Customer Journey Maps), evaluating navigation structures, analyzing cognitive load, and identifying critical usability issues affecting user efficiency and satisfaction.

The study includes the collection and analysis of UX data using usability testing, A/B experiments, heatmap-based behavior analytics, and session recordings. Based on the obtained results, a set of recommendations for UX optimization was developed, aimed at improving interface clarity, reducing friction in key user tasks, enhancing accessibility, and ensuring responsive design across devices. A prototype of the improved interface demonstrates the practical implementation of the proposed methodological approach.

The results of this research can be applied to the development of new web applications, the improvement of existing digital services, and the training of specialists in UX/UI design and web engineering. The work confirms that system analysis combined with modern UX methodologies significantly increases the quality and efficiency of user interaction with complex web systems.

Keywords: system analysis, user scenarios, UX optimization, usability, web applications, user experience, user behavior analytics, interaction design, UI/UX, human-computer interaction.

ЗМІСТ

ВСТУП.....	2
РОЗДІЛ 1. ДОСЛІДЖЕННЯ КОРИСТУВАЦЬКИХ СЦЕНАРІЇВ ТА ОПТИМІЗАЦІЯ UX ВЕБЗАСТОСУНКІВ	4
1.1 Аналіз предметної області та постановка задачі.....	4
1.1.1 Характеристика об'єкта і результати його функціонування	5
1.1.2 Опис діючої ІС	5
1.1.3 Опис недоліків діючої інформаційної системи. Обґрунтування необхідності створення удосконалення діючої ІС	7
1.1.4 Цілі та функції ІС. Попередні економічні результати.....	8
1.2 Розробка концепції інформаційної системи	8
1.2.1 Опис результатів вивчення об'єкта автоматизації.....	8
1.2.2 Опис і оцінка переваг і недоліків розроблених альтернативних варіантів створення інформаційної системи	10
1.2.3 Порівняльний аналіз вимог користувача до ІС і варіантів концепції ІС на предмет задоволення потреб користувача	11
1.2.4 Очікувані результати і ефективність реалізації обраного варіанту концепції ІС	12
1.2.5 Обґрунтування вибору оптимального варіанту концепції і опис запропонованої інформаційної системи	15
1.3 Математичне забезпечення розробки.....	16
1.4 Постановка задачі.....	20
Висновки до розділу 1.....	21
РОЗДІЛ 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ПО СИСТЕМНОМУ АНАЛІЗУ КОРИСТУВАЦЬКИХ СЦЕНАРІЇВ ТА ОПТИМІЗАЦІЯ UX ВЕБЗАСТОСУНКІВ ІНФОРМАЦІЙНОГО ПОРТАЛУ МІСТА	22
2.1 Загальносистемні рішення.....	22
2.1.1 Схема організаційної структури	22
2.1.2 Основні функції і зв'язки між окремими службовими особами їх підпорядкованість	22
2.1.3 Схема функціональної структури.....	23
2.1.4 Опис постановки задачі.....	34
2.1.5 Загальний опис системи.....	38
2.2 Рішення з організаційного забезпечення	39
2.2.1 Опис організаційної структури	39
2.3 Склад інформаційного забезпечення.....	39
2.3.1 Організація інформаційного забезпечення.....	45

2.3.2 Опис організації інформаційної бази	46
2.4 Рішення з технічного забезпечення	55
2.4.1 Схема автоматизації.....	55
2.5 Призначення і характеристика	58
2.5.1 Використана інформація	58
2.5.2 Алгоритм рішення	59
2.6 Методи і засоби програмного забезпечення	60
2.6.1 Структура програмного забезпечення	63
Висновки до розділу 2.....	68
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КОРИСТУВАЦЬКИХ СЦЕНАРІЇВ ТА ОПТИМІЗАЦІЯ UX ВЕБЗАСТОСУНКІВ.....	70
Висновки до розділу 3.....	70
РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ.....	72
4.1. Мета та принципи техніко-економічного обґрунтування	72
4.2. Технічні передумови та ресурсне забезпечення	72
4.2.1. Апаратне забезпечення	72
4.2.2. Програмне забезпечення	72
4.2.3. Кадрові ресурси.....	72
4.3. Оцінка витрат на розробку	73
4.3.1. Витрати на оплату праці.....	73
4.3.2. Витрати на програмні засоби та інструменти.....	73
4.3.3. Накладні витрати	73
4.3.4. Загальна сума витрат	73
4.4. Економічна доцільність та ефективність впровадження	73
4.4.1. Прямі економічні вигоди	73
4.4.2. Непрямі вигоди.....	74
4.5. Аналіз окупності проекту.....	74
Висновки до розділу 4.....	74
РОЗДІЛ 5. ОХОРОНА ПРАЦІ.....	75
5.1 Загальні положення	75
4.2 Аналіз шкідливих та небезпечних факторів	75
4.3 Розрахунок системи штучного освітлення	78
Висновки до розділу 5.....	82
РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА	83
6.1. Загальні положення та актуальність екологічної безпеки у сфері ІТ	83
6.2. Екологічні аспекти розробки вебзастосунків	83
6.3. Енергоефективність та екологічні принципи у проектуванні вебзастосунку	84
6.4. Вимоги техніки безпеки та охорони праці при роботі з комп'ютерною технікою	84
6.5. Заходи щодо зменшення негативного впливу ІТ-	

інфраструктури на довкілля.....	85
Висновки до розділу 6.....	85
ВИСНОВКИ.....	87
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	88
ДОДАТОК А – Технічне завдання	90
ДОДАТОК Б – Фрагмент програмного коду	93
ДОДАТОК В – Опис системи.....	99
ДОДАТОК Г – Інструкція користувача	101
ДОДАТОК Д –Програма і методика випробувань.....	102
ДОДАТОК Є- Екранні форми розробки.....	104

ВСТУП

Стрімкий розвиток цифрових технологій та зростання кількості вебзастосунків у різних сферах діяльності зумовлюють потребу у створенні ефективних, зручних та інтуїтивних інтерфейсів для користувачів. У сучасних умовах якість користувацького досвіду (UX) стає одним із ключових конкурентних факторів, що визначає успішність вебсервісів незалежно від їх призначення — комерційного, інформаційного, освітнього чи адміністративного. Низький рівень юзабіліті, перевантаженість інтерфейсу, нелогічна навігація або надмірна складність виконання сценаріїв призводять до зниження ефективності роботи користувачів, збільшення кількості помилок і відмов, а також зменшення довіри до вебзастосунку.

Недоліки, характерні для багатьох сучасних вебзастосунків, полягають у відсутності чіткого моделювання користувацьких сценаріїв, недостатній увазі до поведінкових патернів аудиторії, відсутності системного аналізу UX-даних, а також у фрагментарному використанні інструментів UX-дизайну без комплексного підходу. Часто інтерфейси проєктуються інтуїтивно, без врахування когнітивного навантаження, що призводить до дезорієнтації користувача та зниження ефективності роботи з вебресурсом. У таких умовах виникає потреба у формалізації процесів аналізу користувацьких сценаріїв на основі системного підходу.

Актуальність дослідження полягає в необхідності розроблення комплексної методики системного аналізу користувацьких сценаріїв та оптимізації UX вебзастосунків, яка дозволить підвищити якість взаємодії користувачів із цифровими сервісами та забезпечити їх ефективність у реальних умовах використання. Це особливо важливо для ресурсів з високою інтенсивністю взаємодій, складною структурою або великою кількістю різних груп користувачів [1].

Об’єкт дослідження — процеси взаємодії користувача з вебзастосунком у контексті виконання користувацьких сценаріїв.

Предмет дослідження — методи системного аналізу та підходи до оптимізації користувацького досвіду (UX) у вебзастосунках.

Мета дослідження — розробити та обґрунтувати системний підхід до

аналізу користувацьких сценаріїв і оптимізації UX вебзастосунків з метою підвищення ефективності, зручності та якості користувацької взаємодії.

Для досягнення поставленої мети необхідно виконати такі завдання дослідження.

1. Проаналізувати теоретичні основи UX-дизайну, системного аналізу та методів моделювання користувацьких сценаріїв.
2. Дослідити існуючі підходи до оцінювання юзабіліті та поведінки користувачів у вебзастосунках.
3. Визначити та описати ключові користувацькі сценарії досліджуваного вебзастосунку.
4. Здійснити збір та аналіз UX-даних (юзабіліті-тестування, A/B-тестування, аналітика поведінки).
5. Виявити основні проблеми та «вузькі місця» у функціонуванні інтерфейсу та навігаційної структури.
6. Розробити методологію оптимізації UX на основі системного аналізу сценаріїв.
7. Створити прототип або концептуальну модель оптимізованого інтерфейсу та обґрунтувати внесені зміни.
8. Провести оцінювання ефективності запропонованих рішень шляхом порівняння до- та після-метрик UX.

Наукова новизна роботи полягає у формуванні комплексної моделі системного аналізу користувацьких сценаріїв, яка інтегрує методи поведінкової аналітики, моделювання сценаріїв та UX-оптимізації в єдину структуру. Запропоновані підходи дозволяють більш точно визначати проблемні точки взаємодії та формувати науково обґрунтовані рішення для покращення UX у вебзастосунках [2].

РОЗДІЛ 1. ДОСЛІДЖЕННЯ КОРИСТУВАЦЬКИХ СЦЕНАРІЇВ ТА ОПТИМІЗАЦІЯ UX ВЕБЗАСТОСУНКІВ

1.1 Аналіз предметної області та постановка задачі

Сучасні вебзастосунки є ключовим інструментом взаємодії користувачів з цифровими сервісами — від електронної комерції до освітніх платформ і корпоративних інформаційних систем. Підвищення складності функціональних можливостей та розширення цільових аудиторій зумовлюють необхідність системного підходу до проектування користувацького досвіду (UX). Саме UX визначає ефективність, зручність, швидкість взаємодії та загальний рівень задоволеності користувача.

Предметна область охоплює методи системного аналізу, теорію взаємодії людини з комп'ютером (HCI), дизайн інтерфейсів, аналіз поведінкових моделей та оптимізацію користувацьких сценаріїв. Центральне місце в дослідженні займає моделювання сценаріїв взаємодії (User Flows, Customer Journey Maps), визначення точок фрустрації, аналіз когнітивного навантаження та оптимізація навігаційних структур[3].

Особливу увагу приділено методам збору та аналізу даних про користувачів: юзабіліті-тестуванню, А/В-експериментам, тепловим картам, аналізу кліків, записам сесій та іншим інструментам UX-аналітики. Ці методи дозволяють виявити проблемні ділянки інтерфейсу, визначити фактори, що сповільнюють виконання сценаріїв, або ті, що призводять до помилок чи відмови від дії.

Системний аналіз у контексті UX передбачає цілісне бачення вебзастосунку як складної людиноцентричної системи, що включає: технічну архітектуру, інформаційну структуру, логіку бізнес-процесів, поведінку користувачів та їхні потреби. Оптимізація UX ґрунтується на принципах логічності, мінімізації когнітивних витрат, адаптивності та доступності [4].

Вивчення предметної області дозволяє визначити основні проблеми, притаманні сучасним вебзастосункам: надмірна складність інтерфейсу, невпорядко-

вана структура навігації, нечіткість сценаріїв, низька продуктивність, недостатня адаптивність під різні пристрої та відсутність персоналізації. Їх розв'язання вимагає інтеграції системного підходу, UX-методологій та інструментів аналітики, що забезпечує підвищення ефективності виконання ключових дій користувачами та покращення загального користувацького досвіду.

1.1.1 Характеристика об'єкта і результати його функціонування

Предметна область є дана інформаційна система розроблена для великих міст, та міст-курортів. Для прикладу розглядається місто N яке є обласним центром, та неподалік біля себе має велику кількість курортних містечок та вихід до моря, що надає можливість жителям та туристам запропонувати зручну, та корисну інформаційну систему яка буде приносити користь, та прибуток .

1.1.2 Опис діючої ІС

Проведемо аналіз стану інформаційної забезпеченості міста.

Інформаційна забезпеченість міст поступово росте, але недостатня для розвитку інфраструктури.

Постачальники курортних послуг викладають інформацію про себе :

- на дошках оголошень;
- в газетах, брошурах, на радіо;

Ці web-сайти надають можливість:

- 1) прочитати статті про переваги міста;
- 2) побачити фото-звіти деяких підприємств;
- 3) дізнатись погоду.

Ці web-сайти не надають можливість:

- а) побачити фото, прочитати статті про всі:
 - 1) готелі;
 - 2) будинки для відпочиваючих(неподалік від міста);
 - 3) лікувально-оздоровчі пансіонати (неподалік від міста);
 - 4) кафе,бари;
 - 5) столові;

- б) нічні клуби;
- 7) інші місця відпочинку, розваг.
- б) замовити будь-яку кімнату у будь-якому готелі на вибір;
- с) прочитати відгуки попередніх клієнтів, залишити свої відгуки;
- д) прийняти замовлення від туриста для готелю, кафе, і т. д. ;
- е) отримати свіжу інформацію про транспорт, нові маршрути, заздалегідь замовити авто;
- ф) отримати зручну докладну карту місцевості;
- г) жителям міста і гостям самостійно наповнювати інформаційну систему: господарям клубів, барів виставляти повідомлення про зміни в розважальній програмі, про приїзд артистів, музикантів, діджеїв, господарів закладів харчування про зміни в меню, та інші новини;
- h) дізнатись ціни послуг.

На даному об'єкті окремо функціонують:

- а) сайт міста представлено на рисунку 1.1;

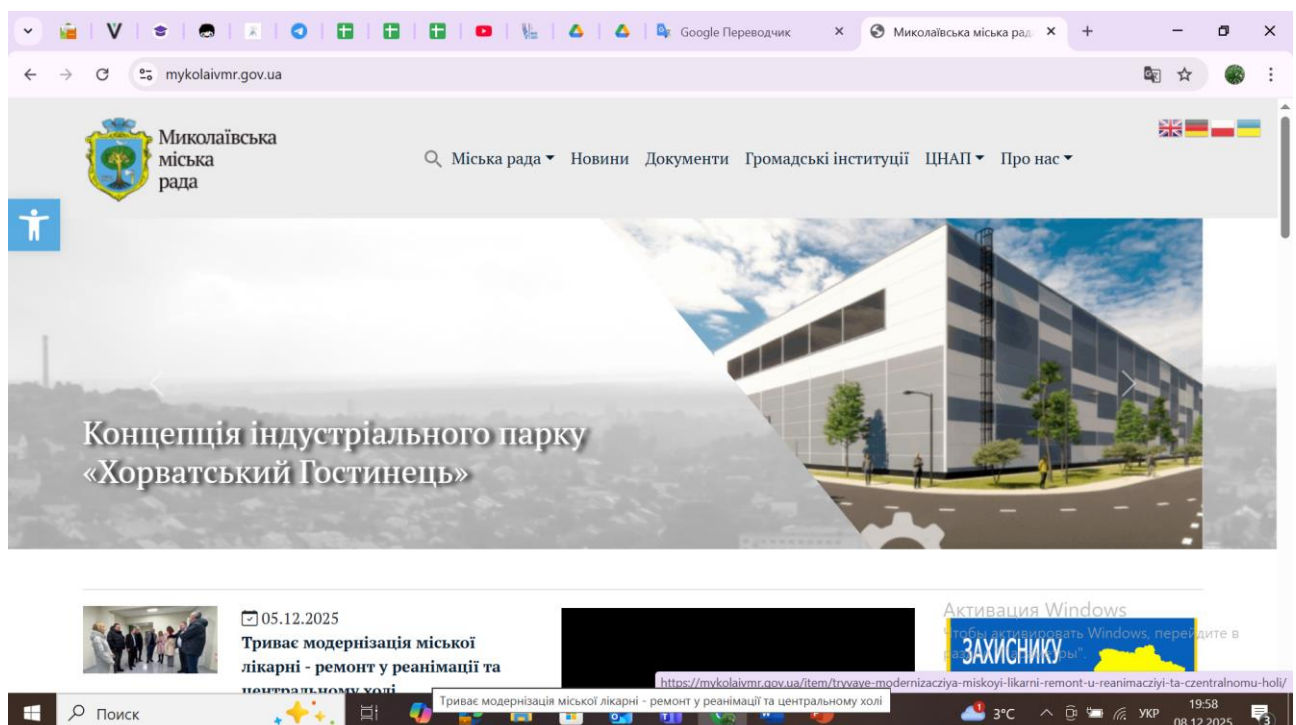


Рисунок 1.1- Діючий сайт міста

В результаті проведення аналізу інформаційної забезпеченості міста, робимо наступний висновок.

На даній предметній області не має діючої інформаційної системи у повному розумінні цього поняття. Адже система- це цілеспрямована сукупність взаємопов'язаних елементів, які взаємодіють паралельно, як єдине ціле. Інформаційна система для збору, зберігання, обробки, видачі інформації використовує програмні і технічні засоби.

Елементи інфраструктури даного об'єкту не взаємопов'язані між собою, не взаємодіють як єдине ціле, кожен з них діє окремо. Готелі, кафе, бари, транспортні служби, жителі не співпрацюють між собою. Деякі елементи не використовують програмні, технічні засоби. Дають оголошення, приймають замовлення окремо. Одна з головних цілей всіх елементів інфраструктури міста спільна- залучення максимальної кількості туристів та пробудити інтерес жителів. Відсутність ІС не дозволяє досягти цієї цілі [5].

1.1.3 Опис недоліків діючої інформаційної системи. Обґрунтування необхідності створення удосконалення діючої ІС

Всі існуючі варіанти не можуть задовольнити користувача у зручності користування функціями, дизайном.

Існуючий сайт даного міста (рисунок 1.1) не надає жодної інформації про готельні, харчувальні, транспортні комплекси і інші комплекси, а лише інформує про декілька приватних квартир для оренди.

Недоліком є те, що елементи даного об'єкту не взаємодіють як єдине ціле, кожен з них діє окремо. Готелі, кафе, бари, транспортні служби, жителі не співпрацюють між собою. Оголошення дають окремо [6].

Недоліком є нездатність задовольнити потреби користувача:

- дізнатись інформацію про місця проживання, ціни, рівень комфорту;
- дізнатись інформацію про транспорт, ціни, розклад маршрутів, зміни;
- дізнатись інформацію про місця харчування, ціни, меню;
- зареєструватись постачальникам послуг;
- постачальникам послуг викласти інформацію про вільні місця;

- бронювати послуги (проживання, транспорт, харчування);
- дізнатись історію міста, переглянути зручну карту, фото, статті, новини місцевості, погоду.

1.1.4 Цілі та функції ІС. Попередні економічні результати

Ціль створення інформаційної системи - підвищення рівня інформативності, а отже рівня послуг і прибутку рідного міста. Для досягнення цілі було проведено аналіз інформаційної системи, буде:

- проведено опис інформаційної системи;
- розроблено проект інформаційної системи;
- розроблено програмне забезпечення інформаційної системи .

Функції ІС:

- 1) надавати можливість постачальникам послуг викласти інформацію про вільні місця;
- 2) надавати можливість зареєструватись постачальникам послуг;
- 3) надавати можливість бронювати послуги (проживання, транспорт, харчування);
- 4) показати історію міста, переглянути зручну карту, фото, статті, новини місцевості, надати інформацію про погоду;
- 5) надати інформацію про місця проживання, ціни, рівень комфорту;
- 6) надати інформацію про транспорт, ціни, розклад маршрутів, зміни;
- 7) надати інформацію про місця харчування, ціни, меню.

Дана ІС надасть можливість обмінюватись інформацією туристам та жителям між собою та з постачальниками послуг [7] .

1.2 Розробка концепції інформаційної системи

1.2.1 Опис результатів вивчення об'єкта автоматизації

В результаті вивчення предметної області було розроблено варіанти концепції створення інформаційної системи. Варіант 1 концепції створення інформаційної системи представлено на рисунку 1.5.

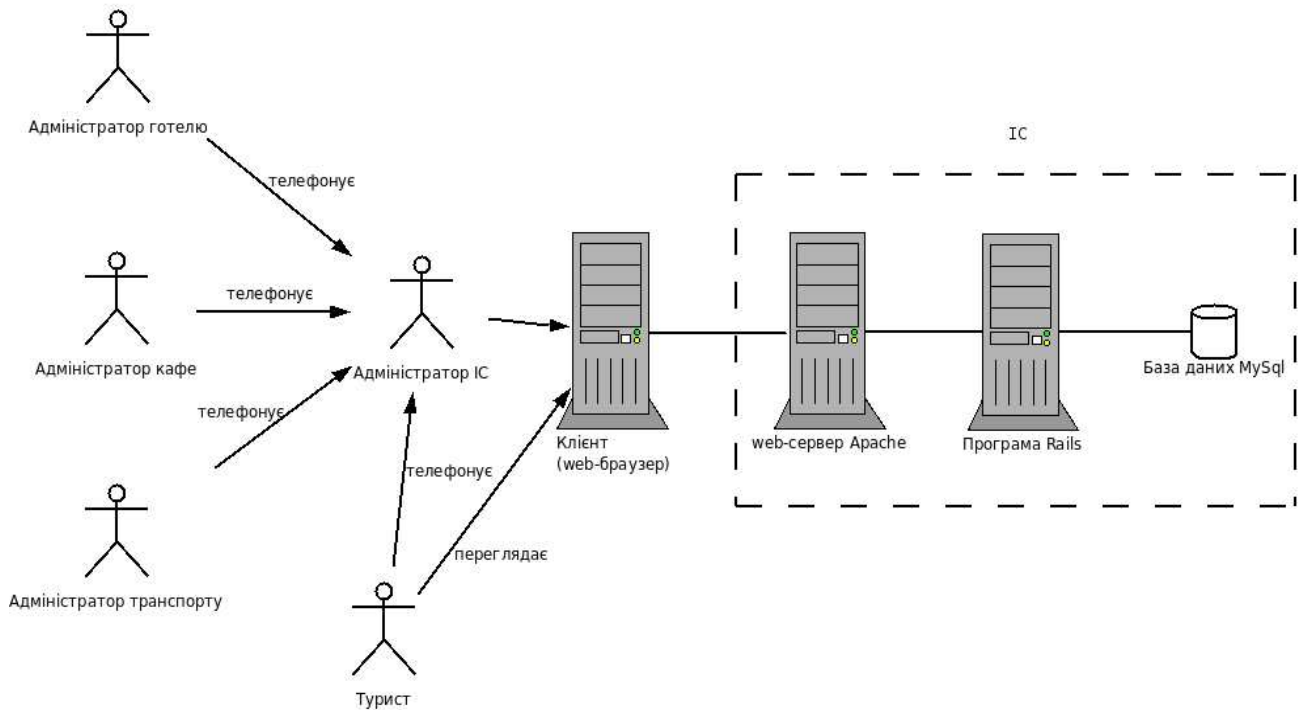


Рисунок 1.5- Варіант 1 концепції створення інформаційної системи

Варіант 2 концепції створення інформаційної системи представлено на рисунку

1.6.

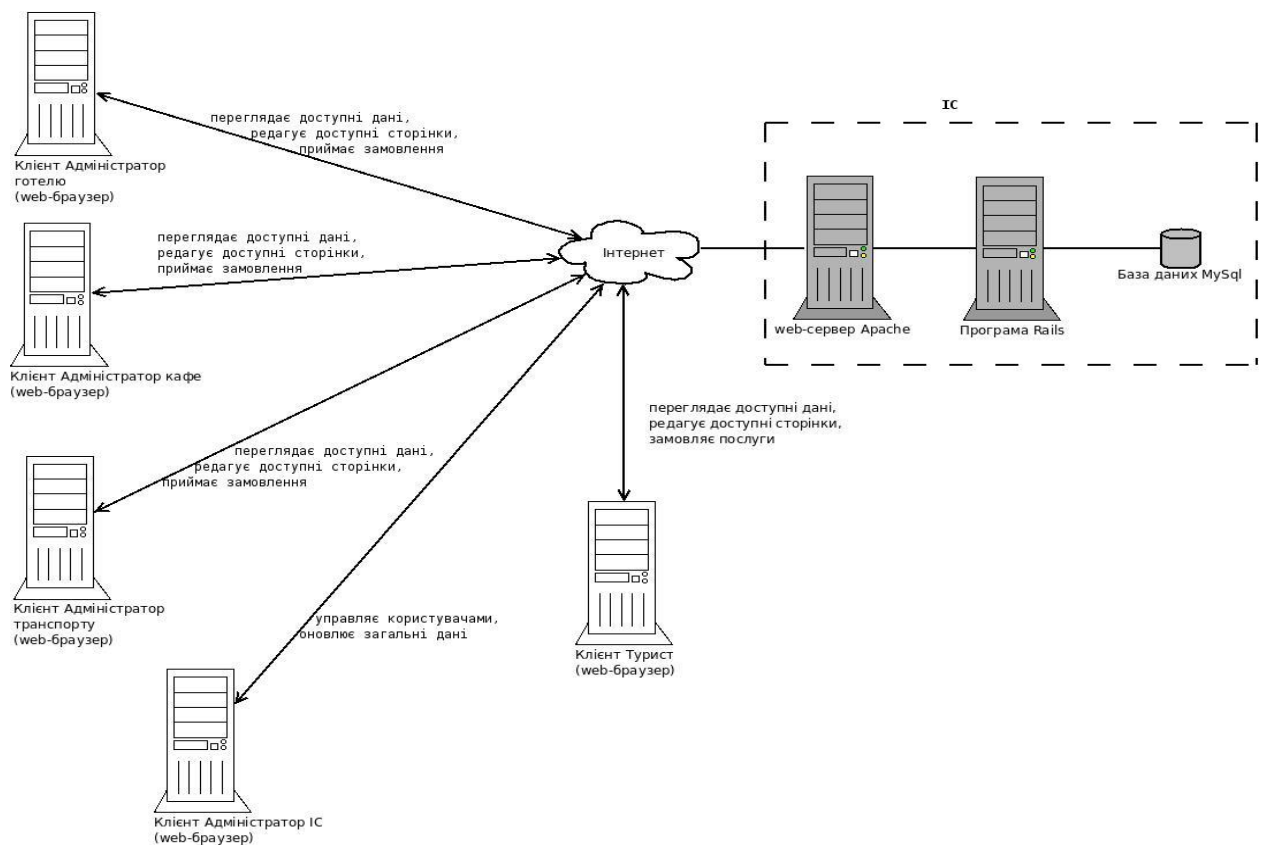


Рисунок 1.6- Варіант 2 концепції створення інформаційної системи

Системний аналіз користувацьких сценаріїв розглядає вебзастосунок як комплексну систему, у якій користувач, інтерфейс, функціональні модулі та контекст використання взаємодіють між собою. Концепція передбачає:

Виявлення ключових користувацьких сценаріїв та моделювання їх у вигляді User Flow та CJM.

Аналіз поведінки користувачів за допомогою аналітики, юзабіліті-тестувань та виявлення «вузьких місць».

Оцінку ефективності інтерфейсу через метрики UX (час виконання, помилки, успішність дій) [8].

Оптимізацію UX шляхом переробки навігації, зменшення когнітивного навантаження, покращення доступності та підвищення інтуїтивності інтерфейсу.

Створення покращеного прототипу та перевірку ефективності змін через тестування та A/B експерименти.

Як результат інтерфейс стає простішим, зручнішим, логічним, швидким та орієнтованим на реальні потреби користувача.

1.2.2 Опис і оцінка переваг і недоліків розроблених альтернативних варіантів створення інформаційної системи

Недоліки варіанту 1 створення інформаційної системи:

Витрати часу на переговори адміністраторів інформаційної системи з адміністраторами підприємств. Низька швидкість обробки заявки адміністратором.

З розвитком інформаційної системи витрати коштів на велику кількість додаткових адміністраторів.

Адміністратор інформаційної системи не знає всіх особливостей і деталей об'єкту. Він може принести неясність, помилки при переговорах. Це може зіпсувати репутацію закладу.

Переваги варіанту 1 створення інформаційної системи:

Економія часу розробника інформаційної системи на створення інтерфейсів

для кожного адміністратора.

Переваги варіанту 2 створення інформаційної системи:

Швидкість обробки заявок від користувачів інформаційної системи.

Відсутність втрати клієнтів через перевантаженість адміністратора прийому заявок.

Інтерфейси для адміністраторів однотипні, тому витрати на їх розробку незначні.

Зменшення ймовірності помилок при виборі закладу відпочинку, замовленні через те, що адміністратор інформаційної системи не знає всіх особливостей і тонкостей.

Недоліки створення інформаційної системи:

Недоліком створення нової інформаційної системи є неминучі витрати часу, а отже і коштів на проектування і розробку програмного продукту. Ми маємо ризики не завершити програму вчасно. У найгіршому випадку це веде за собою ризик не завершити її взагалі. В результаті ми не отримаємо вкладені гроші, і очікуваний прибуток.

Переваги:

Ми маємо можливість створення саме такого програмного продукту, який максимально відповідає потребам нашої предметної області. Інформаційна система створена спеціально для даного міста зможе найбільш вигідно і яскраво висвітлити особливості місцевості. Ми можемо надати послуги великій кількості категорій користувачів.

1.2.3 Порівняльний аналіз вимог користувача до ІС і варіантів концепції ІС на предмет задоволення потреб користувача

Порівняння вимог користувача до ІС і варіантів концепції ІС на предмет задоволення потреб користувача занесемо у таблицю 1.1

Таблиця 1.1- Порівняння вимог користувача до ІС і варіантів концепції ІС на предмет задоволення потреб користувача

№	Вимоги користувача до ІС	Варіант 1	Варіант 2
---	--------------------------	-----------	-----------

		концепції	концепції
1	дізнатись інформацію про місця проживання	+	+
2	дізнатись інформацію про транспорт	+	+
3	дізнатись інформацію про місця харчування	+	+
4	zareєструватись постачальникам послуг	+	+
5	постачальникам послуг викласти інформацію про вільні місця	-	+
6	бронювати послуги (проживання, транспорт, харчування);	+	+
7	обмінюватись відгуками	-	+
8	дізнатись історію міста, зручну карту, фото, статті	+	+
9	переглядати інформацію про погоду	+	+
10	портативна, легковісна, не навантажена рекламою, зручний дизайн	+	+
11	неупереджена інформація	-	+

1.2.4 Очікувані результати і ефективність реалізації обраного варіанту концепції ІС

Основним очікуваним результатом реалізації обраного варіанта концепції є отримання ІС, яка буде відповідати вимогам, поставленим у пункті 1.1.

Оберемо найбільш ефективний варіант реалізації інформаційної системи.

Для цього створимо порівняльну таблицю варіантів реалізації програми.

Таблиця 1.2-Порівняння варіантів реалізації програми

Функція, варіанти			Переваги	Недоліки
розробка бази даних	open source	MyS ql	прозорі, надійні, відтестовані, безпечні, розповсюджена	невеликі набори даних
		Postg re SQL	швидка, великі набори даних	нерозповсюджена
	пропріетарні (Oracle)		обробляють великі бази даних	дорогі, непрозорі щодо помилок і безпечності
написання програми	системи Jumla, Mambo,		Швидка розробка cms, зручна для керування контентом	Додаткові витрати. Програми не

	Drupall		захищена від sql - injection.
	Agile framework ROR	Вигідна з точки зору економії, так як в організації вже є передплачений Ruby hosting. Зручний, так як генерує код, скорочує час розробки. Захищена від sql — injection.	додаток працює менш швидко
приєднання карти	Flash	красива	довго завантаж-ся, не портативна
	векторна графіка	легка, швидко завантажується	менш гарна
	Ajax	зручна, інформативна, гарна, легка у навігації.	-

Найбільш ефективною буде наступна реалізація концепції ІС.

Написання програмного продукту сучасною об'єктно-орієнтованою інтерпретованою мовою програмування Ruby з використанням Agile framework Ruby on Rails. Також ефективність ROR полягає в архітектурі MVC (Model-View-Controller). Модель відповідає за підтримку стану додатку. Іноді цей стан являється короткочасним, що триває лише лише під час декількох взаємодій з користувачем. А іноді стан є постійним і зберігається за межами додатку, найчастіше і бази даних. Модель- це більше, ніж просто дані, в ній прописані всі бізнес-правила, що застосовуються до даних. Вид (View) відповідає за формування інтерфейсу користувача, який звичайно оснований на даних моделі. Допускається існування множини видів(views), що мають доступ до одних даних моделі, з різними цілями.

Контролери організують роботу додатку. Вони сприймають події зовнішнього світу (введення даних користувачем), взаємодіють з моделлю і відображають відповідні види для користувача. На рисунку 1.7 MVC показана в абстрактному представленні.

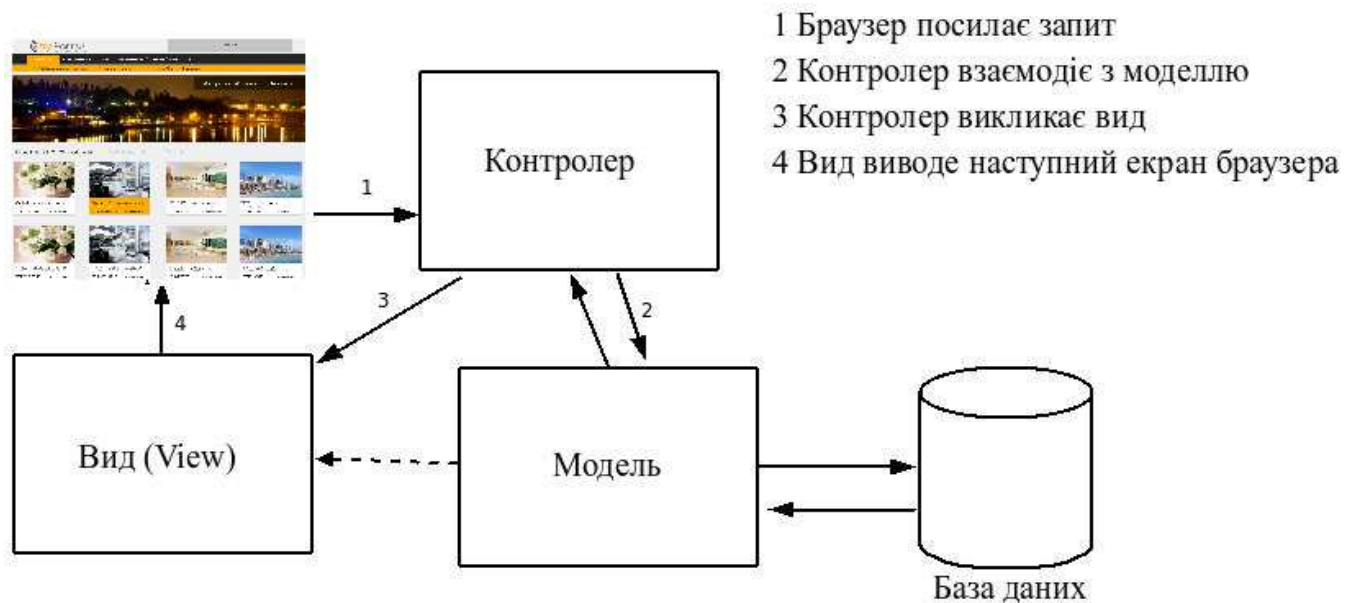


Рисунок 1.7 - Абстрактне представлення MVC

Існують 2 варіанти СУБД: безплатні(open source) і пропріетарні (закриті комерційні). Перевагою безплатних СУБД є не тільки економія коштів на розробку проекту інформаційної системи, а й те що безплатні програмні продукти завдяки своїй відкритості є більш прозорими, надійними щодо помилок, відтестованими, безпечними. Хоча безумовно СУБД Oracle(пропріетарна) є надійною, прекрасно обробляє величезні бази даних(в Oracle 10 g одна таблиця може мати розмір до 128 Тб, розмір бази даних до одного екзобайт) . Дана предметна область не вимагає великої кількості даних і всієї множини функцій, що надає Oracle. Отже не раціонально купувати такий дорогий і потужний продукт як Oracle. Саме тому ми будемо обирати найбільш ефективний серед open source продуктів.

Лідерами серед безплатних СУБД на сьогоднішній день є MySql і PostgreSQL.

PostgreSQL є швидшим в обробці даних ніж MySql. В PostgreSQL розмір таблиці бази даних до 32 Гб, а в MySql всього до 4 Гб. PostgreSQL підтримує наслідування.

Але все ж таки на даній предметній області не має дуже великої кількості інформації. База даних ІС прогнозовано буде розміром менше 1Гб. Тому немає великої необхідності в цих можливостях PostgreSQL. Наслідування засобами

СУБД також не потрібно, так як в Rails використовується бібліотека ORM(object relationship mapping). Де наслідування реалізується поліморфними зв'язками і наслідуванням однієї таблиці.

СУБД MySql є більш розповсюдженою ніж PostgreSQL. В результаті чого я, як розробник ІС добре знаю інсталяцію, налаштування MySql, роботу з нею.

Найбільш ефективним буде використання таких open source продуктів, з якими у мене є досвід роботи :

- Ajax технологія, бібліотека jQuery для приєднання карти Google Maps;
- Web- сервер Apache, що є надійним, гнучким в конфігурації, дозволяє модифікувати повідомлення про помилки;
- система контролю версій git.

По технічному завданню і проекту інформаційної системи, що будуть розроблені у пунктах 1.3 і 2 відповідно, напишемо код програми.

1.2.5 Обґрунтування вибору оптимального варіанту концепції і опис запропонованої інформаційної системи

З таблиці 1.4 видно, що варіант 2 концепції інформаційної системи задовольняє всі пункти вимог користувача до ІС.

Зваживши всі недоліки і переваги варіантів концепції, здатність їх задовольняти вимоги користувача, результати ФВА зробимо висновок.

За оптимальний варіант концепції інформаційної системи обираємо варіант 2- створення розподіленої інформаційної системи з програмними інтерфейсами для кожного типу користувачів.

Технічні засоби є якісними, якщо :

- a) середній % працездатності системи за місяць не менше 99%;
- b) втрата пакетів за місяць не більше 0,2%;
- c) обробка до 100 заявок за хвилину;
- d) технічні засоби відповідають технічному завданню(пункт 1.3).

1.3 Математичне забезпечення розробки

1. Позначення та початкові поняття

- Нехай $U = \{u_1, \dots, u_N\}$ — множина користувачів (або класів користувачів).
- Нехай $S = \{s_1, \dots, s_M\}$ — множина станів/екранів інтерфейсу.
- Користувацький сценарій c — послідовність станів $(s_{i_1}, s_{i_2}, \dots, s_{i_k})$.
- Час виконання кроку $t(s_a \rightarrow s_b)$.
- Імовірність переходу між станами $P_{ab} = P(s_{t+1} = s_b \mid s_t = s_a)$.
- Успішність сценарію для користувача u : $I_c(u) = \begin{cases} 1, & \text{успіх} \\ 0, & \text{неуспіх} \end{cases}$.

2. Модель сценаріїв — марковський підхід

Сценарії моделюються як дискретний марковський ланцюг зі стінами/термінальними станами (успіх/відмова).

Матриця переходів:

$$\mathbf{P} = [P_{ab}]_{M \times M}, \quad \sum_{b=1}^M P_{ab} = 1 \quad \forall a.$$

Статистично важлива величина — ймовірність досягнення успішного термінального стану s_{succ} з початкового стану s_i :

$$\pi_i = P(\text{hit } s_{\text{succ}} \mid s_0 = s_i).$$

π можна знайти як розв'язок системи лінійних рівнянь (для поглинальних ланцюгів).

Середній час до поглинання (time-to-success) з початкового стану s_i :

$$\tau_i = \mathbb{E}[\text{time to absorption} \mid s_0 = s_i].$$

3. Основні UX-метрики (формули)

- Task Success Rate (TSR) для сценарію c :

$$\text{TSR}_c = \frac{1}{N} \sum_{u \in U} I_c(u).$$

- Average Time on Task (ToT):

$$\text{ToT}_c = \frac{1}{N_c} \sum_{u \in U_c} T_c(u),$$

де $T_c(u)$ — час виконання сценарію, U_c — користувачі, які пробували сценарій.

- Error Rate (ER):

$$\text{ER}_c = \frac{\text{кількість помилок у } c}{\text{загальна кількість спроб } c}.$$

- Conversion Rate (CR) (для бізнес-цілей):

$$\text{CR} = \frac{\text{кількість користувачів, що завершили ціль}}{\text{кількість відвідувачів}}.$$

- Efficiency (E) — відношення успішності до часу:

$$E_c = \frac{\text{TSR}_c}{\text{ToT}_c}.$$

- Composite UX score (напр., W-складова) — зважена сума метрик:

$$\text{UX}_c = w_1 \cdot \text{TSR}_c - w_2 \cdot \text{ToT}_c - w_3 \cdot \text{ER}_c,$$

де w_i — ваги, $w_i \geq 0$.

4. Завдання оптимізації (формалізація)

Мета: покращити UX, тобто максимізувати UX_c або мінімізувати час та помилки при фіксованому порозі успішності.

Приклад математичної постановки (одноцільова):

$$\min_{\theta \in \Theta} J(\theta) = \alpha \cdot \text{ToT}_c(\theta) + \beta \cdot \text{ER}_c(\theta) - \gamma \cdot \text{TSR}_c(\theta),$$

де θ — вектор параметрів інтерфейсу (містить наприклад: структура меню, кількість кроків, положення елементів), α, β, γ — налаштовані ваги.

Або мультиоб'єктив:

$$\max_{\theta \in \Theta} (\text{TSR}_c(\theta), -\text{ToT}_c(\theta))$$

і шукаємо Парето-оптимальні рішення.

Методи рішення: евристики (генетичні алгоритми), градієнтні/байєсові оптимізації (якщо є диференційована модель), еволюційні стратегії або A/B-експерименти (чисто емпіричний пошук).

5. Статистична перевірка ефективності (A/B-тести)

5.1 Різниця пропорцій (наприклад, CR або TSR)

Різниця $\Delta = \hat{p}_A - \hat{p}_B$.

Стандартна помилка:

$$SE = \sqrt{\frac{\hat{p}_A(1 - \hat{p}_A)}{n_A} + \frac{\hat{p}_B(1 - \hat{p}_B)}{n_B}}.$$

Z-статистика:

$$Z = \frac{\Delta}{SE}.$$

Для 95% довірчого інтервалу: $\Delta \pm 1.96 \cdot SE$.

5.2 Середній час (t-тест)

При нормальному розподілі:

$$t = \frac{\bar{T}_A - \bar{T}_B}{\sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}},$$

де s_A^2, s_B^2 — дисперсії часів.

5.3 Розмір вибірки (приблизно, для різниці пропорцій)

Щоб виявити різницю d при рівні значущості α і потужності $1 - \beta$:

$$n \approx \frac{(z_{1-\alpha/2} \sqrt{2\bar{p}(1-\bar{p})} + z_{1-\beta} \sqrt{p_1(1-p_1) + p_2(1-p_2)})^2}{(p_1 - p_2)^2},$$

де $\bar{p} = (p_1 + p_2)/2$. Для грубої оцінки при малих відмінностях використовують спрощення.

6. Модель поведінки зважених кліків (heatmap / attention score)

Позначимо c_i — кількість кліків у зоні i , v_i — видимість/важливість зони (0..1). Тоді індекс уваги:

$$A_i = \frac{c_i}{\sum_j c_j} \cdot v_i.$$

Нормалізований heatmap-score: $A'_i = \frac{A_i}{\max_j A_j}$.

7. Модель ймовірності відтоку/відмови

Логістична модель для ймовірності відмови на кроці k :

$$P(\text{drop}_k \mid x) = \sigma(\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p),$$

де $\sigma(z) = \frac{1}{1 + e^{-z}}$, x — вхідні фактори (затримка, кількість полів у формі, тощо). Оцінювання параметрів — методом максимальної правдоподібності.

8. Метрики доступності (Accessibility)

Процент відповідності вимогам:

$$A11Y = \frac{\text{кількість виконаних WCAG-критеріїв}}{\text{загальна кількість релевантних критеріїв}} \times 100\%.$$

9. Приклад застосування: цільова функція оптимізації для сторінки реєстрації

Нехай θ — кількість полів k , положення кнопки (код позначимо b), та розмір СТА s . Можлива дискретна цільова функція:

$$J(\theta) = 0.6 \cdot \text{ToT}(\theta) + 0.3 \cdot \text{ER}(\theta) - 0.1 \cdot \text{TSR}(\theta).$$

Мінімізуємо $J(\theta)$ по допустимому набору θ (наприклад, $k \in \{3, 4, 5\}$, $b \in \{\text{top, mid, bottom}\}$, $s \in \{\text{small, medium, large}\}$) за допомогою A/B тестів або симуляцій марковської моделі.

10. Рекомендації щодо впровадження математичного забезпечення

1. Побудувати марковську модель сценаріїв для ключових user flows та оцінити π_i , τ_i .
2. Збирати ToT, TSR, ER на кожному етапі; формувати composite UX score.
3. Проводити A/B-тести з чіткими статистичними критеріями (вказувати α , β , розмір вибірки).
4. Оптимізувати інтерфейс як мінімізацію цільової функції $J(\theta)$ (або як multi-objective пошук Парето-рішення).
5. Валідувати моделі логістичною регресією (dropout) та перевіряти точність прогнозів на валідаційній вибірці.

1.4 Постановка задачі

На основі проведеного аналізу предметної області та визначених проблем у користувацьких сценаріях вебзастосунків формується комплекс завдань, спрямованих на підвищення ефективності, зручності та інтуїтивності взаємодії користувачів із системою. Постановка задачі передбачає визначення цілей дослідження, обмежень, критеріїв оцінювання та необхідних методів оптимізації UX.

Основною задачею дослідження є розроблення системного підходу до аналізу та оптимізації користувацьких сценаріїв вебзастосунку, що забезпечить покращення UX відповідно до очікувань і поведінкових моделей користувачів.

Для досягнення цієї мети необхідно вирішити такі підзадачі:

Визначити та формалізувати ключові користувацькі сценарії вебзастосунку, встановити їхню структуру, послідовність дій та очікувані результати.

Побудувати моделі взаємодії користувачів із системою (User Flow, CJM, UML-діаграми діяльності) з урахуванням контексту використання та можливих відхилень.

Провести збір та аналіз UX-даних, включаючи результати юзабіліті-тестувань, аналітики поведінки (heatmaps, session recordings), метрик взаємодії та проблемних точок у сценаріях.

Виявити вузькі місця та недоліки інтерфейсу, що знижують ефективність або швидкість виконання користувачами цільових дій.

Розробити методику оптимізації UX, що включає рекомендації щодо переробки навігаційних структур, інтерфейсних рішень, логіки сценаріїв та мінімізації когнітивного навантаження.

Створити прототип або макет оптимізованого інтерфейсу з урахуванням досліджених сценаріїв, принципів доступності та сучасних UX-практик.

Оцінити ефективність запропонованих змін шляхом повторного тестування та порівняння до- та після-метрик (час виконання дій, кількість помилок, показник успішності сценарію, рівень задоволеності користувача).

Висновки до розділу 1

Реалізація поставленої задачі дозволить встановити системний зв'язок між бізнес-цілями вебзастосунку, потребами користувачів та структурою інтерфейсу, що в підсумку забезпечить створення більш ефективного, інтуїтивного та орієнтованого на користувача цифрового продукту.

У першому розділі було проведено системний аналіз предметної області, що дозволив сформувати цілісне уявлення про роль UX у сучасних вебзастосунках та визначити ключові фактори, які впливають на ефективність користувацької взаємодії. Розглянуто основні поняття, підходи та методи, що застосовуються під час аналізу користувацьких сценаріїв, включно з поведінковими моделями, інструментами UX-аналітики та принципами побудови інтерфейсів, орієнтованих на користувача.

Проведений огляд показав, що суттєвими проблемами більшості вебзастосунків є надмірна складність інтерфейсу, перевантаженість навігаційних структур, відсутність цілісності сценаріїв, недостатня адаптивність та невідповідність очікуванням користувачів. Це обґрунтовує актуальність системного аналізу як підходу, що дозволяє розглядати вебзастосунок як комплексну соціотехнічну систему, у якій взаємодія користувачів є критично важливим елементом.

На основі виявлених проблем сформульовано постановку задачі дослідження, що передбачає моделювання користувацьких сценаріїв, аналіз UX-даних, виявлення критичних точок взаємодії та розробку рішень з оптимізації інтерфейсу. Визначені підзадачі забезпечують структурований підхід до вдосконалення UX та дають можливість провести об'єктивну оцінку ефективності запропонованих змін.

Результати першого розділу створюють концептуальну та методологічну основу для подальших етапів дослідження, спрямованих на розробку практичних рекомендацій і прототипу оптимізованого інтерфейсу вебзастосунку.

РОЗДІЛ 2. РОЗРОБКА ПРОЕКТНИХ РІШЕНЬ ПО СИСТЕМНОМУ АНАЛІЗУ КОРИСТУВАЦЬКИХ СЦЕНАРІЇВ ТА ОПТИМІЗАЦІЯ UX ВЕБЗАСТОСУНКІВ ІНФОРМАЦІЙНОГО ПОРТАЛУ МІСТА

2.1 Загальносистемні рішення

2.1.1 Схема організаційної структури

В організаційній структурі інформаційної системи присутні:

- a) турист;
- b) адміністратор ІС ;
- c) адміністратор проживання ;
- d) адміністратор транспорту ;
- e) адміністратор харчування.

2.1.2 Основні функції і зв'язки між окремими службовими особами і їх підпорядкованість

Основна функції туриста полягає у користуванні функціями ІС для вибору даного місця свого відпочинку:

- переглядати інформацію про місця проживання, ціни, рівень комфорту;
- переглядати інформацію про транспорт, ціни, розклад маршрутів, зміни;
- переглядати інформацію про місця харчування, ціни, меню;
- переглядати загальну інформацію (історичні статті міста, зручну карту, фото, статті, новини, погоду);
- реєструватись;
- бронювати послуги (проживання, транспорт, харчування);
- залишати відгуки.

Основна функція адміністратора ІС організовувати і підтримувати безвідмовне функціонування системи:

- підтверджувати або відміняти реєстрацію;
- контролювати відповідність взаємодії користувачів правилам ІС(адміністратор ІС вказує їх в гілці форуму);
- видаляти користувачів-порушників правил і їхні сторінки;

редагувати деяку загальну інформацію(історичні статті міста, фото, статті, новини).

В функції адміністратора ІС не входять такі задачі, як: установлювати Ruby- інтерпретатор, ROR -фреймворк; установлювати і налагоджувати сервер баз даних ;робити резервне копіювання бази даних. Ці задачі виконує хостинг (hosting).

Основні функція адміністраторів готельного, транспортного, комплексу харчування інформувати туриста про свої послуги:

- реєструватись;
- створювати сторінки свого закладу(готельного, транспортного, комплексу харчування);
- редагувати сторінки свого закладу;
- надсилати звіт про вільні номери;
- надсилати лист з підтвердженням замовлення;
- приймати дзвінки від користувачів.

Члени організаційної структури організації взаємодіють між собою, не підпорядковуються один одному. Всі з готельного, транспортного, комплексу харчування знаходяться на одному рівні організаційної ієрархії.

2.1.3 Схема функціональної структури

Інформаційна система міста має такі підсистеми:

- довідково-інформаційна підсистема проживання;
- довідково-інформаційна підсистема харчування;
- довідково-інформаційна підсистема транспорту;
- довідково-інформаційна підсистема загальна;
- бронювання проживання;
- бронювання харчування;
- бронювання транспорту;
- реєстрація;
- форум.

Сукупність дій, що виконуються при реалізації функцій тільки людиною представлено в таблиці 2.1.

Таблиця 2.1 - Сукупність дій, що виконуються при реалізації функцій тільки людиною

№	Назва функції	Сукупність дій людини
1	отримання інформації про місця проживання	перегляд інформації
2	отримання інформації про транспорт	
3	отримання інформації про місця харчування	
4	реєстрація	внесення даних в форму, натиснення “ОК”
5	авторизація	внесення даних в форму, натиснення “ОК”
	замовлення (бронювання) проживання	авторизація
		внесення всіх необхідних даних щодо замовлення
		здійснення запиту на замовлення
5	замовлення (бронювання) транспорту	авторизація
		внесення всіх необхідних даних щодо замовлення
		здійснення запиту на замовлення
6	замовлення (бронювання) харчування	авторизація
		внесення всіх необхідних даних щодо замовлення
		здійснення запиту на замовлення
7	підтвердження замовлення (бронювання)	посилка листа з текстом підтвердження на електронну пошту замовника
7	обмін відгуками	посилка повідомлення
		перегляд
8	отримання інформації про карту місцевості, місця розваг, історію	перегляд
9	уточнення інформації	телефонує адміністратору закладу, що надає послуги(готелю, кафе, транспортного комплексу)

Сукупність дій, що виконуються при реалізації функцій тільки технічними засобами представлено в таблиці 2.2.

Таблиця 2.2 - Сукупність дій, що виконуються при реалізації автоматизованих функцій тільки технічними засобами

№	назва автоматизованої функції (АФ)	сукупність дій програми
1	отримання інформації про місце проживання	показ інформації
2	отримання інформації про транспорт	
3	отримання інформації про місце харчування	
4	замовлення(бронювання) проживання	обробка функції авторизації
		збереження заявки на замовлення
		збереження замовлення
5	замовлення(бронювання) транспорту	обробка функції авторизації
		збереження заявки на замовлення
		збереження замовлення
6	замовлення(бронювання) харчування	обробка функції авторизації
		збереження заявки на замовлення
		збереження замовлення
7	обмін відгуками	показ інформації, надання інтерфейсу для запису даних, збереження інформації
8	отримання інформації про карту місцевості, місця розваг, історію	показ інформації

Представимо графічно схему частини функціональної структури, враховуючи те, що успіх роботи ІС залежить від того, наскільки вона задовольняє вимоги користувача до ІС на предмет задоволення потреб користувача .

Згідно виявленим у пункті 1.2.3 вимогам користувача виявимо варіанти використання ІС користувачем.

Використаємо UML діаграму Джекобсона- діаграму варіантів використання(Use Cases).

Кожен варіант використання є потенційною вимогою до системи, і поки вона не виявлена не можна переходити до планування її реалізації. Варіанти використання є зовнішнім представленням системи.

На цій діаграмі актори представляють роль, котру грає користувач по відношенню до системи. Актори пов'язані з варіантами використання. Актор може виконувати декілька варіантів використання; в свою чергу, у варіанта використання може бути декілька акторів, які його виконують.

Актори не обов'язково люди, хоча на діаграмі варіантів використання вони зображаються у вигляді стилізованих людських фігур. Окремий актор може бути зовнішньою системою, якій необхідна інформація від системи, що розробляється.

Окрім зв'язків між акторами і варіантами використання на діаграмі можуть бути представлені відносини між варіантами використання. Зв'язок включення (include) має місце, коли є певний фрагмент поведінки системи, котрий повторюється більш ніж в одному варіанті використання, і не доречно копіювати його опис в кожному варіанті використання.

Представимо діаграму варіантів використання на рисунку 2.1.

Враховуючи розроблені варіанти використання (Use Case) (рисунок 2.1) інформаційної системи, опишемо вхідні і вихідні дані(таблиця 2.3).

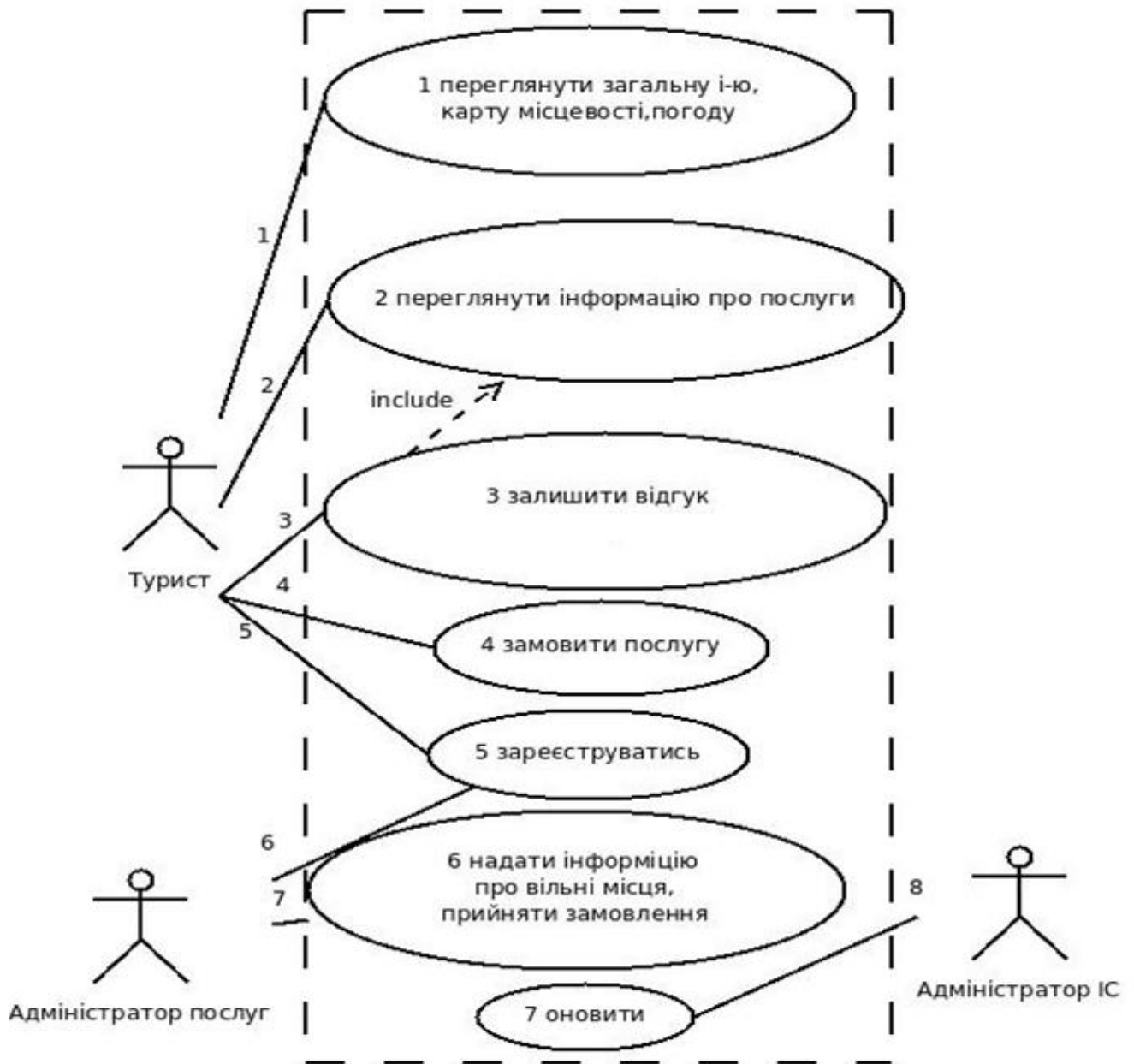


Рисунок 2.1- Діаграма варіантів використання ІС

Враховуючи розроблені варіанти використання (Use Case) (рисунок 2.1) інформаційної системи, опишемо вхідні і вихідні дані(таблиця 2.3).

Побудуємо декомпозицію варіанту використання 2- «Переглянути інформацію про послуги» діаграми використання (рисунок 2.1) і визначимо вхідні і вихідні потоки даних отриманих варіантів використання.

Таблиця 2.3 - Вхідні і вихідні потоки даних

№	Вхідні дані	Вихідні дані
---	-------------	--------------

поток		
1	Фото на сторінках ІС, фото помітки на карті визначних для користувача місць	Фото цікавих красивих місць історії, культури, розваг, історичні статті, карта з зазначеними цікавими місцями, найбільшої необхідності (лікарні, аптеки, і т.ін.), погода
2	Повідомлення з враженнями про заклад	Список комплексів відпочинку, проживання, харчування, транспорту, рівень послуг кожного з них, режим прийому готелів, розклад роботи і меню кафе, розклад маршрутів транспортного, ціни, список нових додаткових послуг
3	Повідомлення з враженнями про підприємства в місті в цілому, про рівень послуг, ціни, цікаві місця історії, культури, розваг. Дані з аутентифікації	Повідомлення інших користувачів з враженнями підприємства в місті в цілому, про рівень послуг, ціни, цікаві місця історії, культури, розваг. Логін, аватар автора іншого повідомлення.
4	Заявка на резервацію(замовлення) послуги	Звіт про вільні місця. Підтвердження резервації (замовлення)
5	Дані про особу користувача	Лист на електронну пошту з підтвердження реєстрації в системі
6	Дані про особу адміністратора. Дані про готель, кафе або транспортний комплекс	Лист на електронну пошту з підтвердження реєстрації в системі
7	Рівень послуг, режим прийому готелів, розклад роботи і меню кафе, розклад маршрутів транспортного, ціни, список нових додаткових послуг.Звіт про	Заявка на резервацію(замовлення) послуги

	вільні місця.Підтвердженн я резервації (замовлення)	
8	Лист з підтвердженням або відміною реєстрації адміністраторів, правила форуму, статті, новини, фото	Заявка на реєстрацію

Декомпозиція варіанту використання 2- «Переглянути інформацію про послуги» представлена на рисунку 2.2.

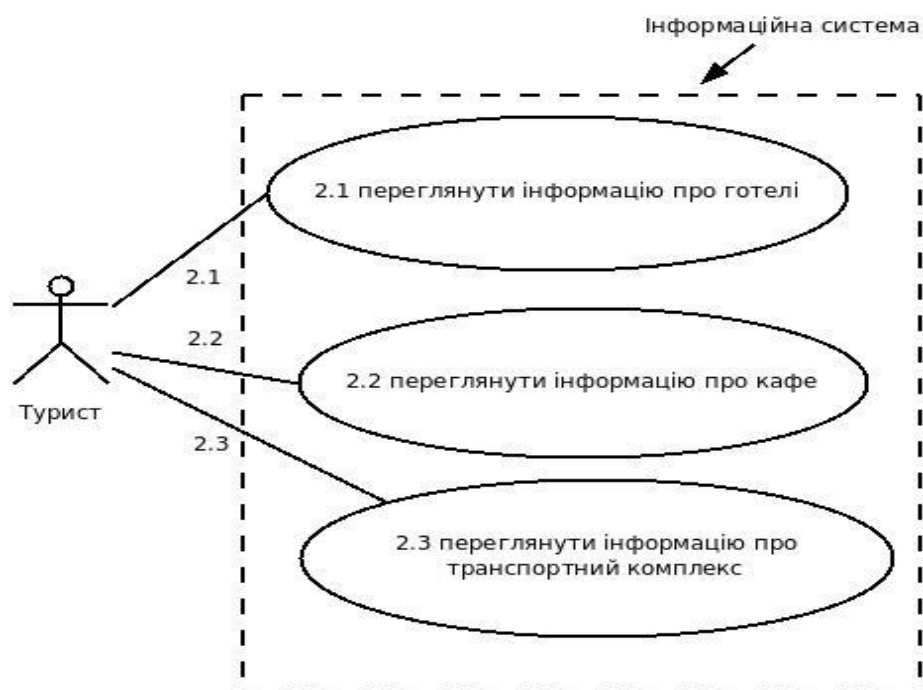


Рисунок 2.2 -Декомпозиція варіанту використання 2- «Переглянути інформацію про послуги»

Вхідні і вихідні потоки даних декомпозиції варіантів використання представлено в таблиці 2.4, 2.5 відповідно

Таблиці 2.4 — Декомпозиція вихідних потоків даних варіанту використання «Переглянути інформацію про послуги»

№ потоку	Вихідні дані
2.1 переглянути інформацію	назва готелю
	ціна номеру

про готель	опис рівня послуг, розкладу прийому
	фото
	адреса
	електронна адреса

Таблиця 2.5 — Декомпозиція вхідних потоків даних варіанту використання «Переглянути інформацію про послуги»

№ потоку	Вихідні дані
2.2 переглянути інформацію про кафе	номер телефону
	назва кафе
	адреса
	опис рівня послуг, розкладу прийому
	фото
	електронна адреса
	номер телефону
	текстове пояснення про сервіс, кухню, режим роботи
	меню
2.3 переглянути інформацію про транспорт	назва транспортного комплексу
	адреса
	розкладу маршрутів
	фото транспорту
	електронна адреса
	номер телефону
	тип транспорту
	опис транспорту

Побудуємо декомпозицію варіанту використання 4- «Замовити послугу» діаграми використання (рисунок 2.1) і визначимо вхідні і вихідні потоки даних отриманих варіантів використання.

Декомпозиція варіанту використання 2- «Замовити послугу» представлена на рисунку 2.3.

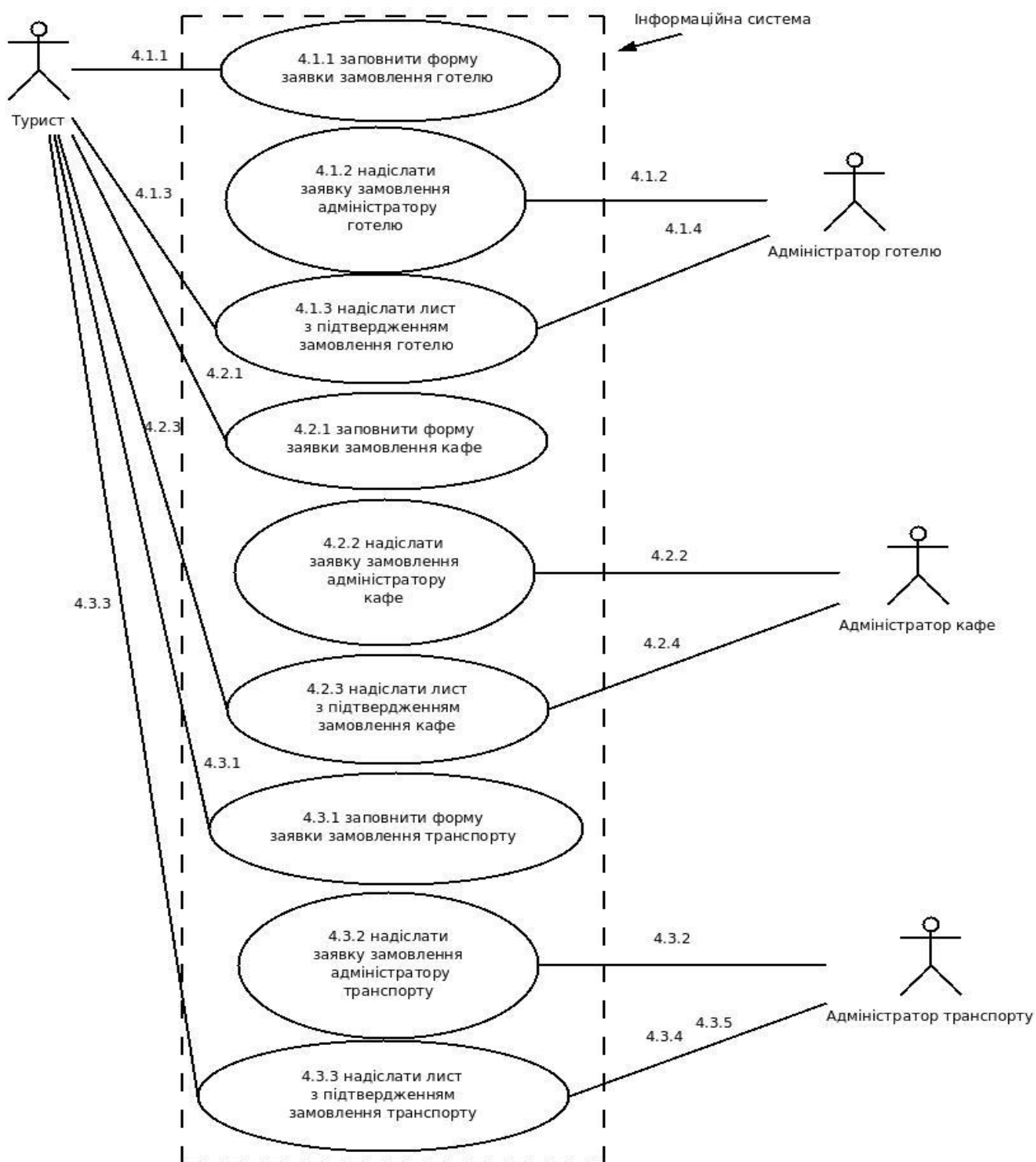


Рисунок 2. 3 - Декомпозиція варіанту використання 2- «Замовити послугу»

Вхідні і вихідні потоки даних декомпозиції варіанту використання 2- «Замовити послугу» для готелю представлено в таблицях 2.5, 2.6. Вхідні і вихідні потоки даних декомпозиції варіанту використання 2- «Замовити послугу» для кафе представлено в таблиці 2.7, 2.8. Вхідні і вихідні потоки даних декомпозиції варіанту використання 2- «Замовити послугу» для транспорту представлено в

таблицях 2.9, 2.10 відповідно.

Таблиця 2.5 — Декомпозиція вхідних потоків даних варіанту використання «Замовити послугу» для готелю

№ потоку	Вхідні дані
4.1.1	Кількість кімнат
	Кількість гостей
	Дата заїзду
	Дата виїзду
	Дані туриста з варіанту використання «Зареєструватись»
4.1.4	Статус «відхилити заявку» або «відхилити заявку» і текстове повідомлення з причиною відмови або «прийняти заявку»
	Дата заїзду
	Дата виїзду

Таблиця 2.6 — Декомпозиція вихідних потоків даних варіанту використання «Замовити послугу» для готелю

№ потоку	Вихідні дані
4.1.1	Список вільних готельних номерів
	Текстовий опис вільних номерів
4.1.2	Кількість кімнат
	Кількість гостей
	Дата заїзду
	Дата виїзду
	id заявки
	Дані туриста з варіанту використання «Зареєструватись»
4.1.3	Текстове повідомлення «Замовлення не прийнято» або «Замовлення не

	прийнято» і текстове повідомлення з причиною відмови» або «Замовлення прийнято»
--	---

Таблиця 2.7 — Декомпозиція вхідних потоків даних варіанту використання «Замовити послугу» для кафе

№ потоку	Вхідні дані
4.2.1	Кількість гостей
	Меню
	Дата
	Дані з варіанту використання «Зареєструватись»
4.2.4	Статус «відхилити заявку» або «відхилити заявку столик зайнято» або текстове повідомлення з причиною відмови або «прийняти заявку»
	Дата

Таблиця 2.8 — Декомпозиція вихідних потоків даних варіанту використання «Замовити послугу» для кафе

№ потоку	Вихідні дані
4.2.1	Список вільних столиків
	Текстовий опис столиків
4.2.2	Кількість гостей
	Дата
	id заявки
	Дані туриста з варіанту використання «Зареєструватись»
4.2.3	Текстове повідомлення «Замовлення не прийнято» або «Замовлення не прийнято» і текстове повідомлення з причиною відмови» або «Замовлення прийнято»

Таблиця 2.9 — Декомпозиція вхідних потоків даних варіанту використання

«Замовити послугу» для транспорту

№ потоку	Вхідні дані
4.3.1	Кількість пасажирів
	Вид транспорту
	Дата приїзду
	Пункт призначення
	Дані з варіанту використання «Зареєструватись»
4.3.4	статус «відхилити заявку» або «відхилити заявку» і текстове повідомлення з причиною відмови або «прийняти заявку»
	Дата

Таблиця 2.10 — Декомпозиція вихідних потоків даних варіанту використання «Замовити послугу» для транспорту

№ потоку	Вихідні дані
4.3.1	Список вільних місць
4.3.2	Кількість пасажирів
	Дата
	id заявки
	Дані туриста з варіанту використання «Зареєструватись»
4.3.3	Текстове повідомлення «Замовлення не прийнято» або «Замовлення не прийнято» причина відмови» або «Замовлення прийнято»

2.1.4 Опис постановки задачі

Згідно варіантів використання ІС, виділяю об'єкти, класи об'єктів, що існують на предметній області і необхідні при використанні інформаційної системи.

Для визначення класів, опису типів об'єктів системи і статистичних відношень між ними, використовуємо діаграму класів UML. Діаграму будуємо з точки зору

специфікації. Асоціації представляють собою відповідальності класів. На діаграмі зазначу необхідну кількість атрибутів і методів на даному рівні проектування. Далі, у розділі 2.6 -Рішення з програмного забезпечення, розгляну і зображу класи з точки зору реалізації і детальніше перелічу всі методи і атрибути моделей.

Як вже було вище сказано (пункт 1.2.5 Очікувані результати і ефективність реалізації обраного варіанту концепції ІС. Орієнтований план реалізації обраного варіанту концепції ІС) реалізується програма архітектурою MVC (Модель- Вид- Контролер). В даному розділі розглядаю моделі. Діаграми контролерів із описом всіх їх конкретних методів і видів побудую в розділі 2.6. Модель зберігає в собі дані. Має функції обробки цих даних. Модель має функції перевірки валідності даних. Містить в собі дані про зв'язки (асоціації, композиції) між класами моделей, наслідування. Адміністратори готелів, кафе, транспорту, туристи поведуть себе однаково при реєстрації, авторизації. Вводять однакові набори даних при реєстрації (представлені в таблиці 2.16). Тому введемо клас користувач, підтипами якого будуть адміністратори, туристи. Винесемо властивості, що повторюються у класів Готель, Кафе, Транспортний комплекс (указані в пункті 2.3.5) в абстрактний клас Сервіс, для збереження цих даних.

Так як діаграма є великою, кількістю зв'язків, що неминуче пересікаються, то для більшої наглядності я зображаю зв'язок наслідування між класом- предком і класом дочірнім товстою лінією, а інші лінії тонкими.

Представимо діаграму об'єктів на момент реєстрації готельного номеру на рисунку 2.4, на момент замовлення готельного номеру на рисунку 2.5.

Представимо діаграму класів моделей архітектури MVC на рисунку 2.6.



Рисунок 2.4- Діаграма об'єктів на момент реєстрації готельного номеру

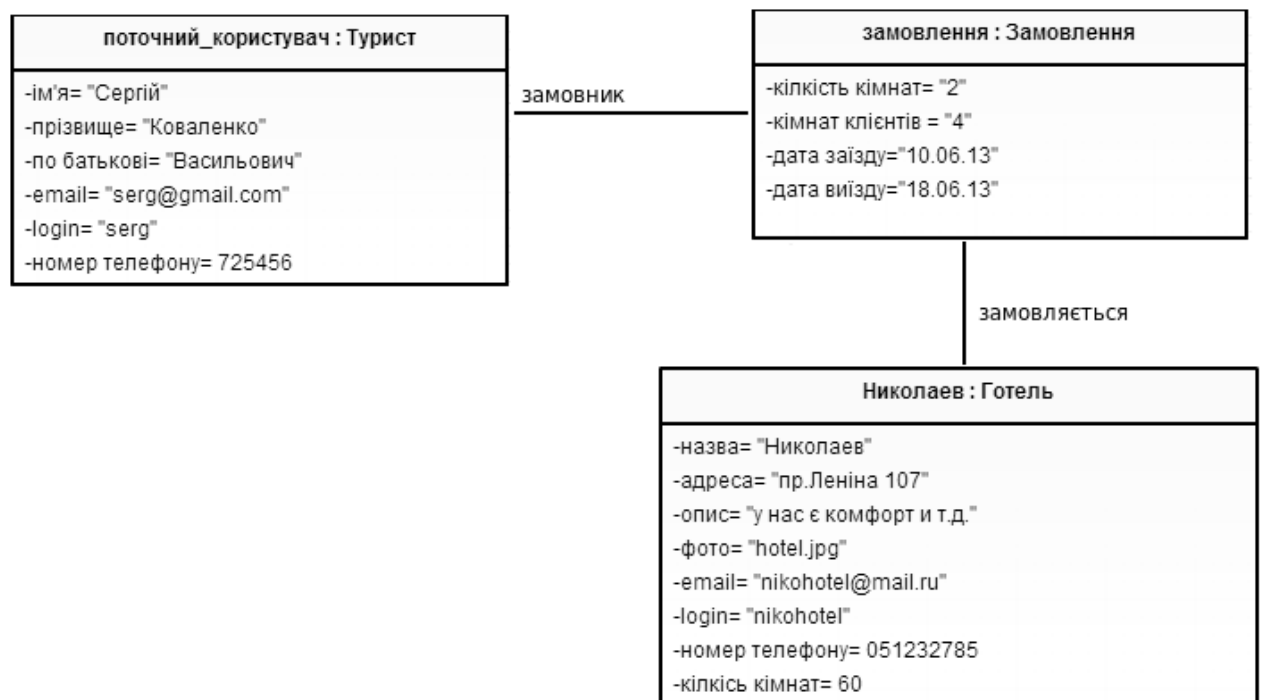


Рисунок 2.5- Діаграма об'єктів на момент замовлення готельного номеру

2.1.5 Загальний опис системи

Вид діяльності, для якої призначена система:

Інформаційна система призначена для користування нею туристів. Ціллю створення ІС є підвищення рівня послуг та прибутку рідного міста.

Перелік функцій, що реалізуються системою вказано і описано в пункті 2.1.2.

Опис підсистем вказано в пункті 2.1.2. Представимо роботу системи діаграмами станів UML. Розробимо для класу ReservationRoom (Замовлення готельного номеру) діаграму станів, представлену на рисунку 2.7.

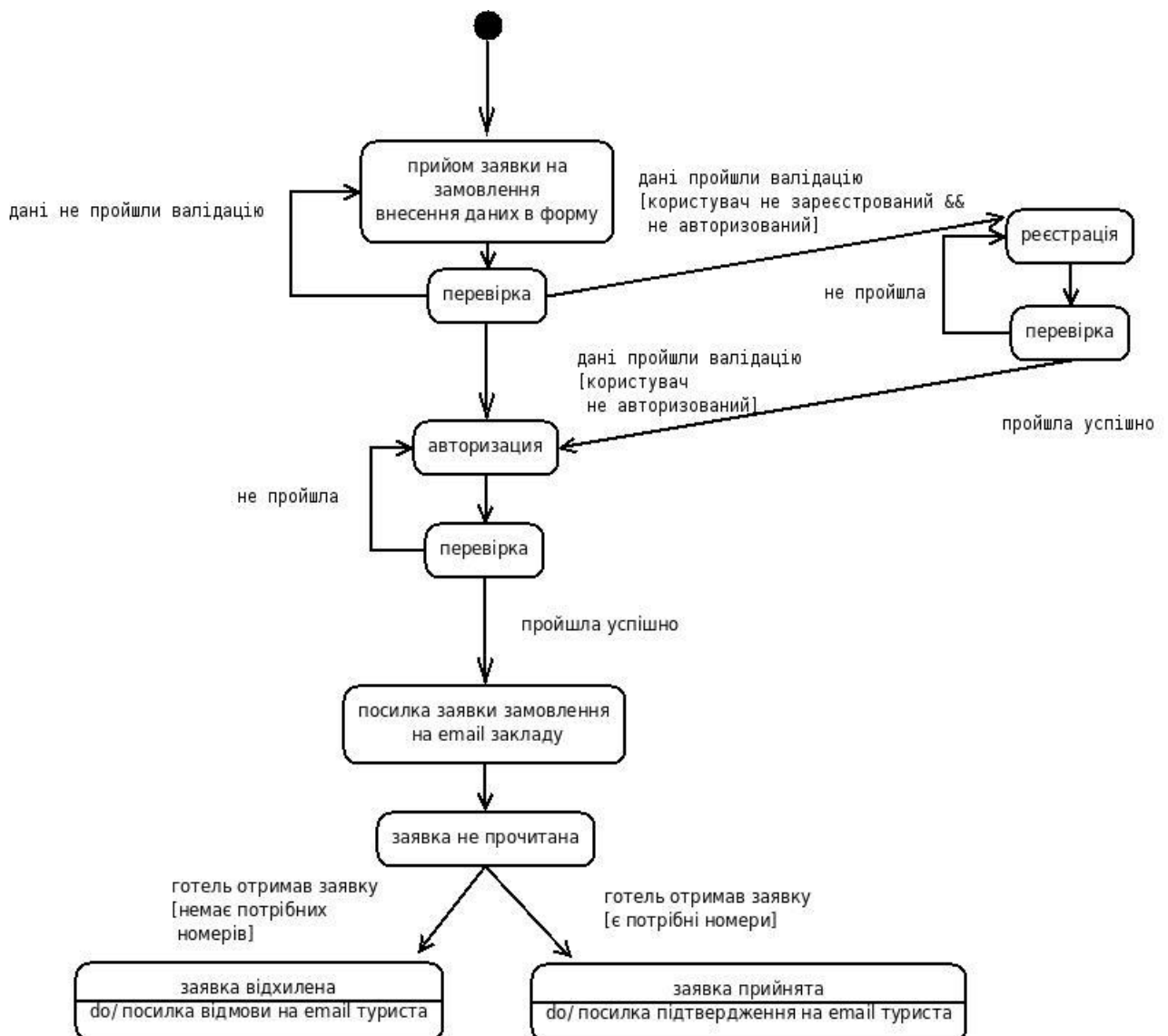


Рисунок 2.7- Діаграма станів класу ReservationRoom

2.2 Рішення з організаційного забезпечення

2.2.1 Опис організаційної структури

Опис організаційної структури і функцій підрозділів наведено в пункті 2.1.1. Опишемо організаційну структуру докладніше в таблиці 2.11 .

Таблиця 2.11 - Опис організаційної структури і функцій підрозділів

№	найменування посадової особи	основні функції
1	турист	користуватись пунктами меню, переглядати доступні дані, редагувати доступні сторінки, здійснювати замовлення послуг, залишати відгуки
2	адміністратор ІС	підтверджувати або відмінити реєстрацію користувачів, контролювати відповідність взаємодіє користувачів правилам ІС(адміністратор ІС вказує їх в гілці форуму), видаляти порушників правил ІС, редагувати спільну інформацію(статті, новини, фото)
3	адміністратор готельного комплексу	редагувати сторінку готелю, на якому зареєстрований, інші доступні сторінки, приймати замовлення
4	адміністратор комплексу харчування	редагувати сторінку кафе, на якому зареєстрований, інші доступні сторінки, приймати замовлення
5	адміністратор транспортного комплексу	редагувати сторінку транспортного комплексу , на якому зареєстрований, інші доступні сторінки , приймати замовлення

2.3Склад інформаційного забезпечення

Враховуючи вимоги до інформаційної системи, розроблену концепцію, технічне завдання і розроблені варіанти використання (Use Case) інформаційної системи, визначимо склад інформаційного забезпечення.

Опишемо сутності, їх атрибути , присутні в інформаційному забезпеченні.

Опишемо сутності текстом, а атрибути у вигляді таблиці атрибута з описом для кожної сутності.

Сутність Турист- користувач, який уповноважений працювати з програмою. Для кожного користувача встановлене розмежування повноважень,

тобто встановлені права доступу на виконання функцій програми.

Сутність Адміністратор ІС - фізична особа, яка контролює, регулює роботу ІС. Не реєструється через web-інтерфейс. Інші права доступу вказані в таблиці 7-Права доступу користувачів до функцій ІС.

Сутність Адміністратор Готельного комплексу - фізична особа, яка через web-інтерфейс реєструється, реєструє готель, в якому працює, редагує сторінки готелю, на якому зареєстрований. Інші права доступу вказані в таблиці 7-Права доступу користувачів до функцій ІС.

Сутність Адміністратор Транспортного комплексу - фізична особа, яка через web-інтерфейс реєструється, реєструє, транспортний комплекс, в якому працює, редагує сторінки готелю, на якому зареєстрований. Інші права доступу вказані в таблиці 7-Права доступу користувачів до функцій ІС.

У сутностей Турист, Адміністратор ІС, Адміністратор Готельного комплексу, Адміністратор Транспортного комплексу, Адміністратор Кафе аналогічні атрибути в логічній моделі. Тому опис спільних атрибутів цих сутностей занесемо в одну таблицю 2.11.

Опис атрибутів сутності Готель, Готельний номер, Замовлення, Харчування, Меню занесемо в таблиці 2.12, 2.13, 2.14, 2.15, 2.16 відповідно.

Таблиця 2.11-Опис атрибутів сутностей Турист, Адміністратор ІС, Адміністратор Готельного комплексу, Адміністратор Транспортного комплексу, Адміністратор Кафе

№	атрибут	Опис
1	Прізвище	Паспортні дані
2	Ім'я	Паспортні дані
3	По батькові	Паспортні дані
4	Номер телефону	робочий номер телефону
5	email	Адреса електронної пошти
6	Логін	Алфавітно- цифровий набір символів, що ідентифікує

		користувача і разом із паролем використовується для надання йому дозволу на з'єднання з системою та визначення його прав доступу до ресурсів. Логін унікальний в межах даної системи.
7	Пароль	секретне слово або набір символів, для підтвердження особи
8	Аватар	Довільне зображення у, яке асоціюється з обліковим записом користувача. Картинка або фото. Фото не обов'язково цього користувача

Сутність Готель- це може бути готель, пансіонат, санаторій, приватний будинок чи квартира, де є спальні місця і зручності для клієнта.

Таблиця 2.12-Опис атрибутів сутності Готель

№	атрибут	Опис
1	Назва	Унікальна назва в межах міста. Зазначається це готель, чи пансіонат, санаторій, приватний будинок. Може бути назва типу «Приватний будинок вулиця № »
2	Адреса	Місто, вулиця, номер
3	Кількість номерів	Кількість всіх номерів готелю
4	Опис	Текстове пояснення про умови в готелі, його особливості, наявність розважальних програм
5	Адресу електронної пошти.	Адреса електронної пошти готелю
6	Фото	Зображення, яке обов'язково асоціюється з даним готелем

Сутність Готельний номер-це номер в готелі, пансіонаті, кімната в квартирі.

має такі **атрибути**

Таблиця 2.13-Опис атрибутів сутності Готельний номер

№	атрибут	Опис
1	Номер	-
2	Кількість місць в номері	кількість спальних місць
3	Ціна одного спального місця	ціна одного односпального ліжка

4	Ціна цілого номеру	ціна цілого номеру
5	Фото	Зображення, яке обов'язково асоціюється з даним готельним номером

Сутність Заовлення готельного номеру- Завчасна (найчастіше не пізніше однієї доби) письмова заявка на аренду місця проживання. Здійснити заовлення може тільки зареєстрований користувач.

Таблиця 2.14- Опис атрибутів сутності Заовлення готельного номеру

№	атрибут	Опис
1	Кількість кімнат	-
2	Кількість гостей	-
3	Дата заїзду	Дата не раніше 12 години дня
4	Дата виїзду	Дата не раніше 12 години дня

Сутність Харчування — додаткова послуга до проживання в готелі, санаторії, пансіонаті, або приватному будинку. Необов'язкова послуга, так як туристу може не підходити харчування даного готелю по стану здоров'я, по вподобанням.

Таблиця 2.15 - Опис атрибутів сутності Харчування

№	атрибут	Опис
1	Опис	Текстовий опис
2	Графік	Розклад прийомів їжі по годинам

Сутність Меню- певний вид меню, в залежності від призначення(«Лікувальне», «Дієтичне», «Весільне», «Дитяче святкове»)

Таблиця 2.16- Опис атрибутів сутності Меню

№	атрибут	Опис
1	Назва меню	Назва меню в залежності від призначення(«Лікувальне», «Дієтичне», «Весільне», «Дитяче святкове»)
2	Назва страви	-

3	Склад	перелік інгредієнтів в страві
4	Ціна страви	ціна однієї порції
6	Вага страви	вага однієї порції

Сутність Транспортний комплекс -організація або приватна особа, що надає послуги перевезення пасажирів, багажу

Опис атрибутів сутності Транспортний комплекс, Транспорт, Замовлення транспорту занесемо в таблицю 2.17, 2.18, 2.19 відповідно.

Таблиця 2.17 - Опис атрибутів сутності Транспортний комплекс

№	атрибут	Опис
1	Назва	Унікальна назва в межах міста. Зазначається: це транспортний комплекс, автопарк чи приватний підприємець. Може бути назва «Приватний підприємець ПІБ»
2	Адреса	Місто, вулиця, номер
3	Кількість машин	Кількість всіх машин
4	Опис	Текстове пояснення про види транспорту
5	Адресу електронної пошти.	Адреса електронної пошти Транспортного комплексу
6	Фото	Зображення, яке обов'язково асоціюється з даним транспортним комплексом

Сутність Транспорт — одиниця транспортного засобу(автомобіль, катер, автобус, мікроавтобус)

Таблиця 2.18 - Опис атрибутів сутності Транспорт

№	атрибут	Опис
1	Номер	-
2	Назва	Назва марки
3	Кількість місць	кількість місць для сидіння

4	Ціна 1-го квитка	
5	Ціна всієї машини	Може відрізнятись від суми ціни всіх місць окремо
6	Тип	легковий автомобіль, пікап, мікроавтобус
7	Опис	скільки вміщується багажу
8	Фото	Зображення, яке обов'язково асоціюється з даним готельним номером

Сутність Замовлення транспорту -завчасна (найчастіше не пізніше ніж пів години) письмова заявка на резервування місця в транспортному засобі. Здійснити замовлення може тільки зареєстрований користувач.

Таблиця 2.19 - Опис атрибутів сутності Замовлення транспорту

№	атрибут	Опис
1	Кількість пасажирів	
2	Вид транспорту	легковий автомобіль, мікроавтобус, автобус, катер, човен, моторний човен
3	Дата приїзду	дата (РМД), час(ГХ)
4	Пункт призначення	текстовий опис, зазначення точки на карті ІС

Опис атрибутів сутності Кафе в таблицю 2.20 .

Сутність Кафе — місце харчування, з місцями для сидіння. Це може бути кафе, столова, чи бар-ресторан.

Таблиця 2.20 - Опис атрибутів сутності Кафе

№	атрибут	Опис
1	Назва	Унікальна назва в межах міста. Зазначається: це кафе, столова, чи бар-ресторан.
2	Адреса	Місто, вулиця, номер
3	Кількість столів	Кількість всіх столів
4	Опис	Текстове пояснення про сервіс, кухню, режим

		роботи
5	Адресу електронної пошти.	Адреса електронної пошти
6	Фото	Зображення, яке обов'язково асоціюється з даним кафе

2.3.1 Організація інформаційного забезпечення

В якості носія даних ІС вибрано хостинг (hosting). Хостинг- це послуга, що надає дисковий простір для розміщення фізичної інформації на сервері. Програма Rails запущена на хостінгу і база даних зберігається на хостінгу .

Хостинг надає:

- 95% часу безперервної роботи системи. Сервер баз даних MySQL; єдину базу даних розміром до 300 Мб, ір адресу сервера даних 192.168.100.1;
- віртуальний порт на web сервері Apache, зовнішній ір *.*.*.* (приховано для безпеки), з прив'язкою до нього домена oschakov.mk.ua;
- середовище виконання Rails, з працюючим http- проху mongrel, ір адреса 192.168.100.2, порт 10080.

Mongrel це швидкий однопоточний HTTP сервер для Ruby Rails. Mongrel дозволяє запускати Rails додатки без використання сторонніх HTTP сервера, такого наприклад як Apache. Mongrel призначений для хостингу Ruby веб-додатків, mongrel використовує HTTP замість FastCGI або SGI. Ми використовуємо його, і тому що саме він попередньо установлений на хостингу.

З web-сервером Apache програми роблять швидше, стабільніше, паралельно. Тому додатково використовується web сервер Apache.

На хостингу web сервер Apache, Rails програма, база даних MySQL можуть розміщатись як на одній машині, так і кожний компонент на різних машинах. Безпечніше, звичайно розміщувати компоненти на різних машинах. Якщо злоумисниками буде взломано web сервер Apache, програма і ,важливіше, база даних залишаться непошкодженими.

Графічна схема фізичної структури ІС в мережі представлено на рисунку пункт 2.4

В даній дипломній роботі є єдина база даних MySQL. Переваги MySQL описані в пункті 1.2.5.

Найменування бази даних « resort_development ». База даних призначена для збереження, редагування, видалення даних про туристів, їх замовлення, готелі, кафе, ресторани, транспорт, та інші заклади, що можуть залучатись до роботи системи.

База даних дозволяє швидко передавати текстові, графічні повідомлення, без збоїв передавати звіти про замовлення .

2.3.2 Опис організації інформаційної бази

Опишемо логічну і фізичну структури бази даних.

Враховуючи вимоги до інформаційної системи, розроблену концепцію, технічне завдання, розробимо концептуальну модель даних ІС курорту.

Побудуємо ER- схему нотації Питера Пин-Шэн Чена, представлену на рисунку 2.8.

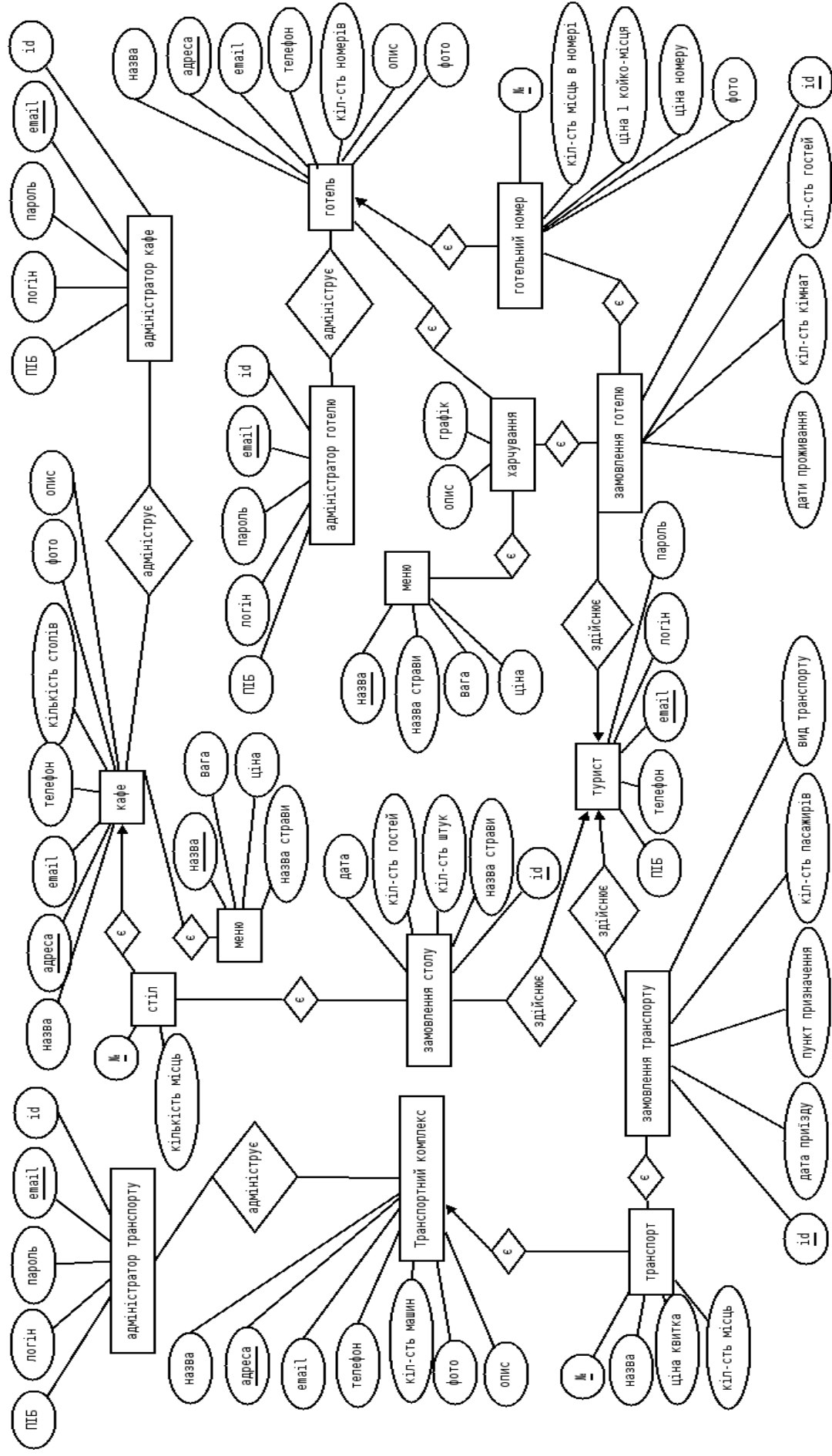


Рисунок 2.8 - Концептуальна модель даних ІС

Враховуючи концептуальну модель даних, розробимо логічну модель даних ІС курорту. Кожній з сутностей на ERD відповідатиме таблиця бази даних. Треба визначити структуру цих таблиць зі зв'язками між ними. Треба визначити типи атрибутів(записів) в таблицях. І що дуже важливо, треба визначити ключові атрибути таблиць- первинні, зовнішні, альтернативні. Для цього побудуємо ER-схему нотації Баркера, представлену на рисунку 2.9.

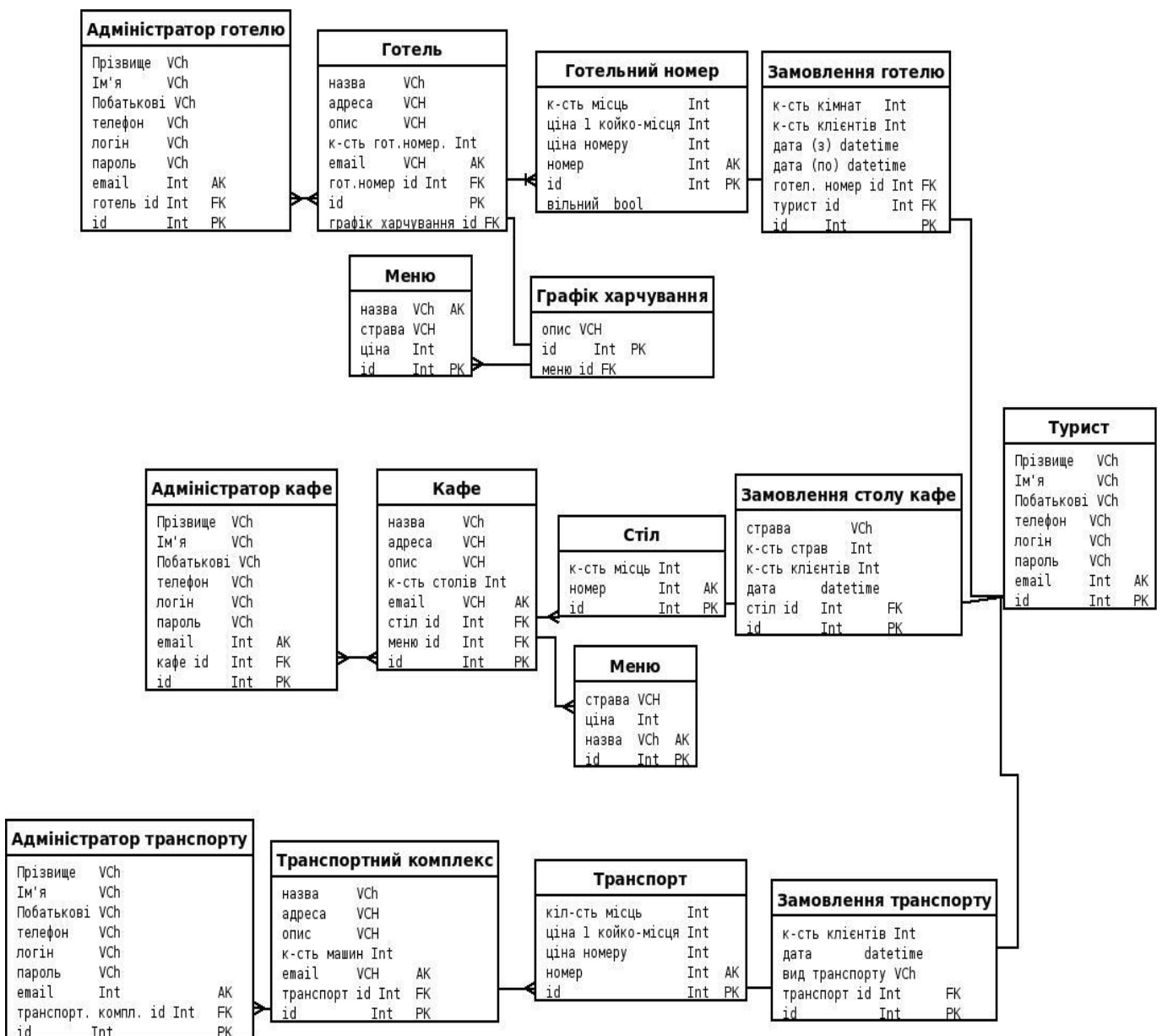


Рисунок 2.9 — Логічна модель даних ER- схема нотації Баркера

Реляційні бази даних дозволяють встановлювати зв'язки між таблицями: наприклад, рядок в таблиці замовлення зв'язується з рядком в таблиці записів послуг. Ці відносини визначаються статично. Але саме ця обставина викликає незручності.

Обійти його можна різними вивертами програмування, але нам, на щастя, цього робити не доведеться. В Rails є механізм для відображення реляційної моделі в більш складній, об'єктно-орієнтованій моделі наслідуванням з використанням однієї таблиці.[3] З рисунку 2.3 видно, що ІС може взаємодіяти з людьми, що виступають в різних ролях: із адміністраторами готелів, кафе, транспорту, туристами. У всіх цих ролей є спільні властивості: прізвище, ім'я, по батькові, номер телефону, email, логін, пароль, аватар. Адміністратори готелів, кафе, транспорту, туристи поводять себе однаково при реєстрації, аворизації. Вводять однакові набори даних при реєстрації (представлені в таблиці 2.22).

Таблиця 2.22— Потоки даних реєстрації Адміністратори готелів, кафе, транспорту, туристів

Вхідні дані реєстрації			
Адміністратор готелю	Адміністратор транспорту	Адміністратор кафе	Турист
Прізвище	Прізвище	Прізвище	Прізвище
Ім'я	Ім'я	Ім'я	Ім'я
По батькові	По батькові	По батькові	По батькові
Номер телефону	Номер телефону	Номер телефону	Номер телефону
email	email	email	email
Логін	Логін	Логін	Логін
Пароль	Пароль	Пароль	Пароль
Аватар	Аватар	Аватар	Аватар

Вводять однакові набори даних при авторизації (представлені в таблиці 2.23).

Таблиця 2.23 — Потоки даних авторизації Адміністратори готелів, кафе, транспорту, туристів

Вхідні дані авторизації			
Авторизація Адміністратора готелю	Авторизація Адміністратора кафе	Авторизація Адміністратора транспорту	Авторизація Туриста
Логін	Логін	Логін	Логін
Пароль	Пароль	Пароль	Пароль

Все це можна змоделювати, сказавши, що класи Адміністратор готелю, Адміністратор кафе, Адміністратор транспорту, Турист являються підкласами класу Користувач.

Готель, Кафе, Транспортний комплекс містять однакові дані :

- назва;
- адреса;
- опис;
- фото;
- email;
- номер телефону

Винесемо властивості, що повторюються у класів Готель, Кафе, Транспортний комплекс в абстрактний клас Сервіс, для збереження цих даних.

Діаграму класів з представленням наслідування показано в пункті 2.1.4
Опис постановки задач.

Підкласи успадковують властивості та зобов'язання класів, що породили їх.

У світі реляційних баз даних поняття успадкування відсутнє, відношення головним чином виражаються поняттями зв'язків. Але однотабличне наслідування, описане Мартіном Фаулером (Martin Fowler) у книзі [7] дозволяє зображати всі класи в ієрархії успадкування в одній таблиці бази даних. [8.256] Ця таблиця містить стовпчик для кожної властивості всіх класів ієрархії. Додатково в неї включається стовпчик, який за угодою називається type, що

ідентифікує, який конкретно клас об'єктів представлений кожною конкретним конкретним записом.

Все відображено на фізичній моделі даних на рисунку 2. 10

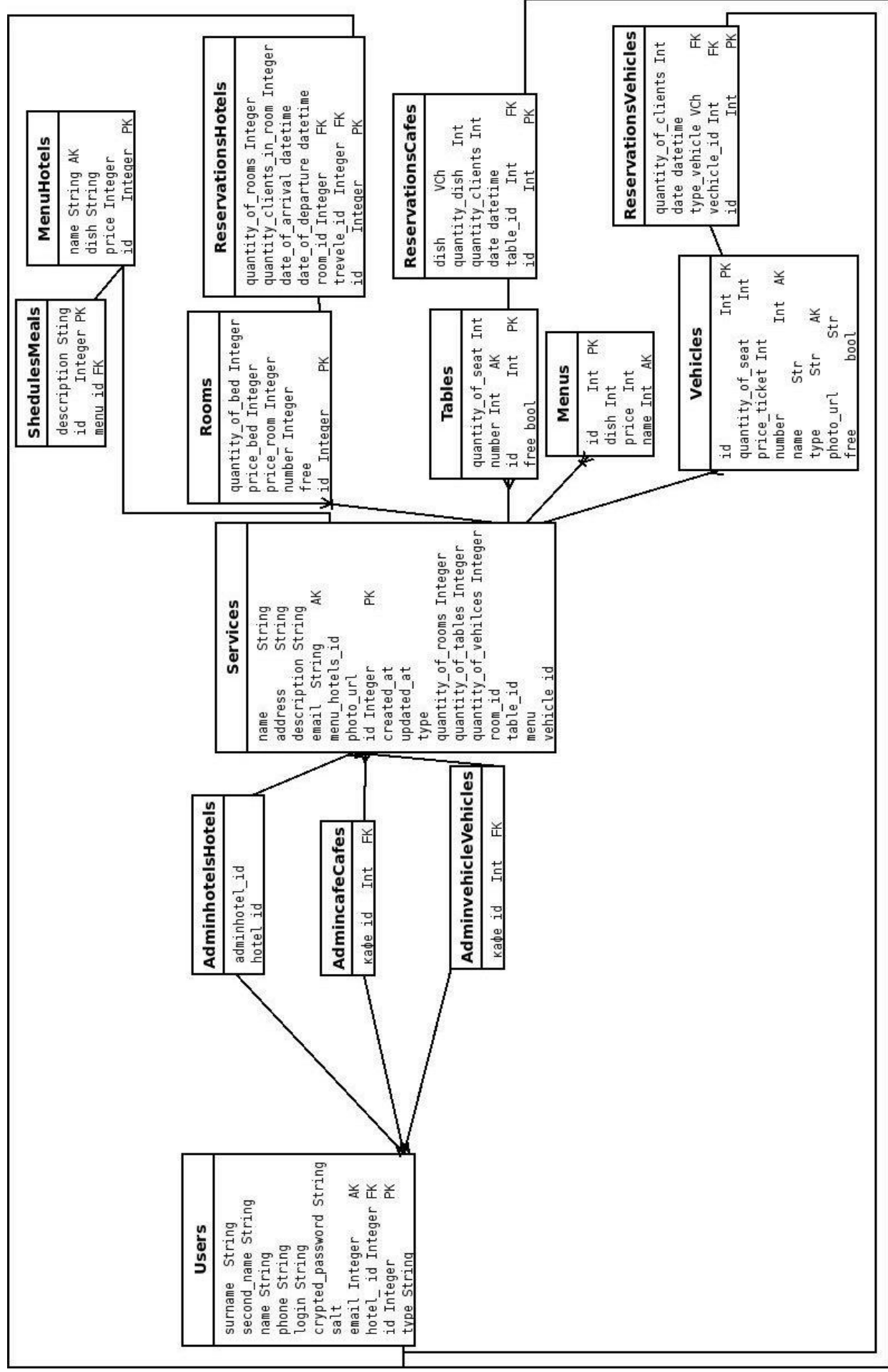


Рисунок 2.5 - Фізична модель даних

Логічна модель розроблена для реляційної моделі даних, а фізична модель даних об'єктно- реляційна. Тому замінено абстрактні назви полів на ті, що будуть в базі даних та вказано розмірність полів. Розробляється база даних MySQL. Імена у логічній і фізичній базах написані різними мовами. Логічна модель розробляється для розуміння її людиною, тому зручніші україномовні позначення. Фізична модель розробляється для розуміння її машиною, тому в базі даних записи англійською мовою. Фізична модель бази даних представлена в таблиці 2.24.

Таблиця 2.24- Фізична модель ІС

Імя фізичної моделі	Поля	Тип	Формат	Ключ в ЛМ
Hotels	name	varchar	255 bytes	
	address	varchar	255 bytes	
	email	varchar	255 bytes	PK
	phone	varchar	255 bytes	
	quantityOfRooms	integer	32 bytes	
	room_id	varchar	255 bytes	FK
	discription	text	64 kbytes	
	created_at	datetime	iso 8601	
	meal_id	varchar	255 bytes	FK
Rooms	number	integer	32 bytes	PK
	priceBed	integer	32 bytes	
	priceIndivisibleRoom	integer	32 bytes	
	quantityOfBed	integer	32 bytes	
	photo_url	varchar	255 bytes	
	created_at	datetime	iso 8601	
	updated_at	datetime	iso 8601	
ReservationHotels	dateOfArrival	date	iso 8601	
	dateOfDeparture	date	iso 8601	
	quantityOfRooms	integer	32bytes	
	QuantityClientsInRoom	integer	32bytes	
	id	integer	32bytes	PK

	shedule	varchar	255 bytes	FK
	treveler_id	varchar	255 bytes	FK
	room_id	varchar	255 bytes	FK
Meals	shedule	varchar	255 bytes	
	discription	text	64 kbytes	

Продовження таблиці 2.24- Фізична модель ІС

Ім'я фізичної моделі	Поля	Тип	Формат поля	Ключ в ЛМ
Cafes	name	varchar	255 bytes	
	address	varchar	255 bytes	
	email	varchar	255 bytes	PK
	phone	varchar	255 bytes	
	quantityOfTable	integer	32 bytes	
	photo_url	varchar	255 bytes	
	discription	text	64 kbytes	
	menu_id	varchar	255 bytes	FK
	table_id	varchar	255 bytes	FK
Tables	number	integer	32 bytes	PK
	quantitySeats	integer	32 bytes	
	created_at	datetime	iso 8601	
	updated_at	datetime	iso 8601	
Menus	name	varchar	255 bytes	PK
	dish	varchar	255 bytes	
	price_dish	integer	32 bytes	
	created_at	datetime	iso 8601	
	updated_at	datetime	iso 8601	
ReservationCafes	time_reservat	date	iso 8601	
	quantity_clients	integer	32 bytes	
	id	integer	32 bytes	PK
	dish	varchar	255 bytes	
	how_many_dish	integer	32 bytes	

	table_id	varchar	255 bytes	FK
	treveler_id	varchar	255 bytes	FK
Trevelers	firstName	varchar	255 bytes	
	secondName	varchar	255 bytes	
	surname	varchar	255 bytes	
	id	varchar	255 bytes	PK
	email	varchar	255 bytes	AK

Продовження таблиці 2.24- Фізична модель ІС

Імя фізичної моделі	Поля	Тип	Формат поля	Ключ в ЛМ
TransportComplexs	name	varchar	255 bytes	
	address	varchar	255 bytes	
	email	varchar	255 bytes	PK
	phone	varchar	255 bytes	
	quantity_of_vehicle	integer	32 bytes	
	photo_url	varchar	255 bytes	
	discription	text	64 kbytes	
	created_at	datetime	iso 8601	
	transport_id	varchar	255 bytes	FK
Transports	number	integer	32 bytes	PK
	price_ticket	integer	32 bytes	
	quantity_of_seat	integer	32 bytes	
	name	varchar	255 bytes	
	photo_url	varchar	255 bytes	

2.4 Рішення з технічного забезпечення

2.4.1 Схеми автоматизації

Представимо схему автоматизації спрощеним зображенням об'єкта на рисунку 2.11.

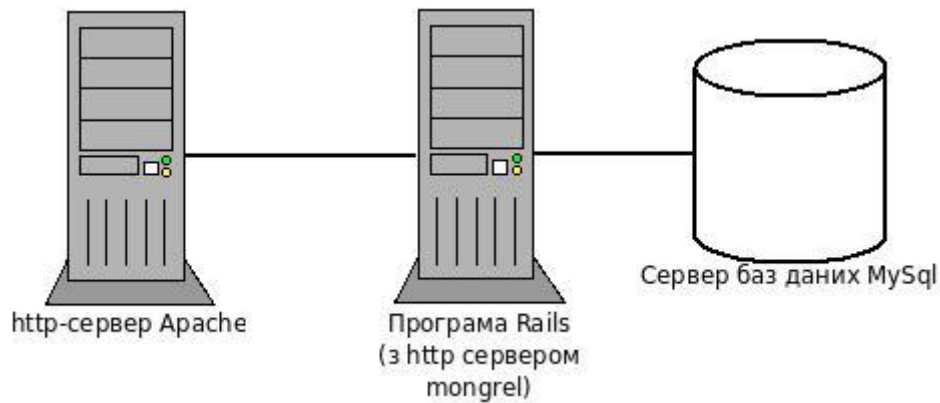


Рисунок 2. 11- Схема фізичного розміщення системи
Зберігається програмний код і запущена програма Rails на орендовній ЕОМ хостингу (hosting). База даних зберігаються на сервері баз даних, який також розміщений на орендовній ЕОМ хостингу.

Технічні засоби, що беруть участь у процесі — це ЕОМ, інтернет- мережа.

На хостингу http- сервер Apache, Rails програма, база даних MySQL розміщатись на різних машинах (рисунок 2.7). Rails програма і база даних MySQL можуть розміщатись на одній ЕОМ окремо від http-сервер Apache (рисунок 2.8).

Всі компоненти можуть розміщатись як на одній машині.

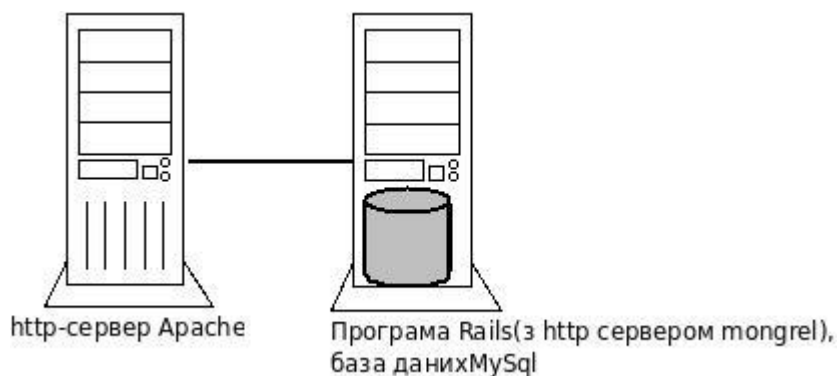


Рисунок 2. 12- Схема фізичного розміщення системи на одній ЕОМ

Безпечніше, звичайно, розміщувати компоненти на різних машинах і найчастіше так і розміщується. Якщо злоумисниками або вірусами буде взломано чи пошкоджено web- сервер Apache, програма і , тим більше, база даних залишаться непошкодженими.

На роботу інформаційної системи не впливає фізичне розміщення її

компонентів. Адміністратор хостінгу установлює компоненти системи на свій розсуд.

Хостинг приховує від всіх фізичне розміщення частин системи в цілях безпеки. Таким чином ми уберігаємо впроваджену ІС від рейдерських атак.

Передплатений хостинг гарантовано забезпечує захист ЕОМ, на якій зберігається програмний код, бази даних від механічних, теплових, електромагнітних та інших впливів. Гарантовано забезпечує захист даних від несанкціонованого доступу до них. Проводить резервне копіювання даних.

Хостинг надає: 95% часу безперервної роботи системи. Сервер баз даних MySQL; єдину базу даних розміром до 300 Мб, ір адресу сервера даних 192.168.100.1(внутрішній не видний користувачам); віртуальний порт на web сервері Apache, зовнішній ір 217.77.222.88 , з прив'язкою до нього домена oschakov.mk.ua; середовище виконання Rails, з працюючим http- проху mongrel, ір адреса 192.168.100.2, порт 10080(внутрішній не видний користувачам).

Mongrel це швидкий однопоточний HTTP сервер для Ruby Rails. Mongrel дозволяє запускати Rails додатки без використання сторонніх HTTP сервера, такого наприклад як Apache. Mongrel призначений для хостингу Ruby веб-додатків, mongrel використовує HTTP замість FastCGI або CGI. Ми використовуємо його, і тому що саме він попередньо установлений на хостингу.

З web-сервером Apache програми роблять швидше, стабільніше, ніж тільки з Mongrel. Web-сервер Apache являється найбільш розповсюдженим промисловим http сервером. Це фактично стандарт web сервера. Web-сервер Apache підтримує паралельну роботу клієнтів (http з'єднань). Apache підтримує кешування статичного наповнення (контенту)- малюнків.

Тому додатково використовується web сервер Apache.

Отже для фізичного розміщення інформаційної системи було обрано орендовану ЕОМ на хостингу з причин:

- 1) гарантований захист даних від механічних, теплових, електромагнітних та інших впливів, від несанкціонованого доступу до них;
- 2) проведення резервного копіювання даних;

- 3) 95% часу безперервної роботи системи;
- 4) надання сервера баз даних MySQL;
- 5) надання єдиної бази даних;
- 6) надання віртуального порту на web- сервері Apache, зовнішнього ip , з прив'язкою до нього домена oshakov.mk.ua;
- 7) надання середовища виконання Rails, з працюючим http- проху mongrel;
- 8) економія електроенергії;
- 9) отримання потужної машини за малі кошти;
- 10) убереження від рейдерських атак.

2.5 Призначення і характеристика

В результаті розробки схеми функціональної структури (пункт 2.1.2), розробимо алгоритм роботи інформаційної системи відповідно до варіантів. По варіанту використання **«Зареєструватись»** розробимо алгоритм **«Реєстрація»**.

Алгоритм **«Реєстрація»** призначений для реєстрації користувача. Після проходження реєстрації користувач має свій обліковий запис в системі. Це дає йому можливість резервувати (бронювати) послуги. Якщо він зареєструвався, як адміністратор, він може створювати нові сторінки свого закладу відпочинку, щоб туристи могли замовляти у нього послуги.

По варіанту використання **«Замовити послугу»** розробимо алгоритм **«Заявка на замовлення готелю»**.

2.5.1 Використана інформація

Вхідні дані варіанту використання «Зареєструватись»:

Прізвище, Ім'я, По батькові, Логін, Пароль, номер телефону (не обов'язково), Адреса електронної пошти.

Вхідні дані варіанту використання «Заявка на замовлення готелю»:

Дата заїзду, Дата виїзду, кількість готельних номерів, кількість жителів.

2.5.2 Алгоритм рішення

Алгоритм «Реєстрація» користувача в системі приведено на рисунку 2. 12.



Рисунку 2.13 - Алгоритм «Заявка на замовлення готелю»



Рисунок 2.13 - Алгоритм «Заявка на замовлення готелю»

2.6 Методи і засоби програмного забезпечення

Програмне забезпечення буде написано сучасною об'єктно-орієнтованою інтерпретованою мовою програмування Ruby з використанням Agile framework Ruby on Rails (ROR). Ефективність ROR полягає в архітектурі MVC (Model-View-Controller).

Модель відповідає за підтримку стану додатку. Іноді цей стан являється короткочасним, що триває лише під час декількох взаємодій з користувачем. А іноді стан є постійним і зберігається за межами додатку, найчастіше у базі даних.

Модель- це більше, ніж просто дані, в ній прописані всі бізнес-правила, що застосовуються до даних.

Вид (View) відповідає за формування інтерфейсу користувача, який звичайно оснований на даних моделі. Допускається існування множини видів(views), що мають доступ до одних даних моделі, з різними цілями.

Контролер організує роботу додатку. Він сприймає події зовнішнього світу (введення даних користувачем), взаємодіє з моделлю і відображає відповідні види для користувача.

Також:

- Контролер відповідає за направлення зовнішніх запитів внутрішнім діям. Він обробляє URL-адреси, орієнтовані на людське сприйняття.
- Контролер керує кешуванням, яке пришвидшує роботу програми в декілька разів.
- Він управляє допоміжними модулями, які розширюють можливості шаблонів уявлень без збільшення обсягу їх коду.
- Контролер управляє сесіями, що дають користувачам відчуття неперервної взаємодії з нашим додатком.

Ruby on Rails використовує **бібліотеку Active Record** (активний запис).

Active Record впритул слідує стандартам ORM (Object relational mapping- об'єктно-реляційне відображення)- моделі:

- **таблиці відображаються у вигляді класів,**
- **рядки - у вигляді об'єктів,**
- **стовпці - у вигляді цих об'єктів**

Покажемо, яким таблицям бази даних відповідають класи програми:

1) Таблиці в базі даних «Vehicles» відповідає в програмі клас Vehicle. Записи цієї таблиці відповідають об'єктам класа, конкретний транспортний засіб являє собою об'єкт класа Vehicle.

2) Таблиці в базі даних «Rooms» відповідає в програмі клас Room. Записи цієї таблиці відповідають об'єктам класа - конкретна кімната являє собою об'єкт класа Room.

3) Таблиці в базі даних «Tables» відповідає в програмі клас Table. Записи цієї таблиці відповідають об'єктам класа, конкретний стіл являє собою об'єкт класа Table.

4) Таблиці в базі даних «ReservationCafes» відповідає в програмі клас ReservationCafe. Записи цієї таблиці відповідають об'єктам класа, конкретне замовлення кафе являє собою об'єкт класа ReservationCafe.

5) Таблиці в базі даних «ReservationTransports» відповідає в програмі клас ReservationTransport. Записи цієї таблиці відповідають об'єктам класа, конкретне замовлення транспорту являє собою об'єкт класа ReservationTransport.

6) Таблиці в базі даних «ReservationHotels» відповідає в програмі клас ReservationHotel. Записи цієї таблиці відповідають об'єктам класа, конкретне замовлення готелю являє собою об'єкт класа ReservationHotel.

7) Таблиці в базі даних «Services» відповідає в програмі абстрактний клас Service. Він не має об'єктів. Записам таблиці «Services» відповідають об'єкти класів TransportComplex, Cafe, Hotel. Ці класи наслідують клас Service.

8) Таблиці в базі даних «TransportComplexes» відповідає в програмі клас TransportComplex. Записи цієї таблиці відповідають об'єктам класа, конкретний транспортний комплекс являє собою об'єкт класа TransportComplex.

9) Таблиці в базі даних «Cafes» відповідає в програмі клас Cafe. Записи цієї таблиці відповідають об'єктам класа, конкретне кафе являє собою об'єкт класа Cafe.

10) Таблиці в базі даних «Hotels» відповідає в програмі клас Hotel. Записи цієї таблиці відповідають об'єктам класа, конкретний готель являє собою об'єкт класа Hotel.

11) Таблиці в базі даних «Users» відповідає в програмі клас User. Записи цієї таблиці відповідають об'єктам класа, конкретний користувач інформаційної системи являє собою об'єкт класа User.

12) Таблиці в базі даних «AdministratorHotels» відповідає в програмі клас AdministratorHotel. Записи цієї таблиці відповідають об'єктам класа, конкретний адміністратор готелю являє собою об'єкт класа AdministratorHotel.

13) Таблиці в базі даних «AdministratorCafes» відповідає в програмі клас AdministratorCafe. Записи цієї таблиці відповідають об'єктам класа, конкретний адміністратор кафе являє собою об'єкт класа AdministratorCafe.

14) Таблиці в базі даних «AdministratorTransportComplexs» відповідає в програмі клас AdministratorTransportComplex. Записи цієї таблиці відповідають об'єктам класа, конкретний адміністратор транспортного комплексу являє собою об'єкт класа AdministratorTransportComplex

15) Таблиці в базі даних «Trevelers» відповідає в програмі клас Treveler. Записи цієї таблиці відповідають об'єктам класа, конкретний турист являє собою об'єкт класа Treveler.

2.6.1 Структура програмного забезпечення

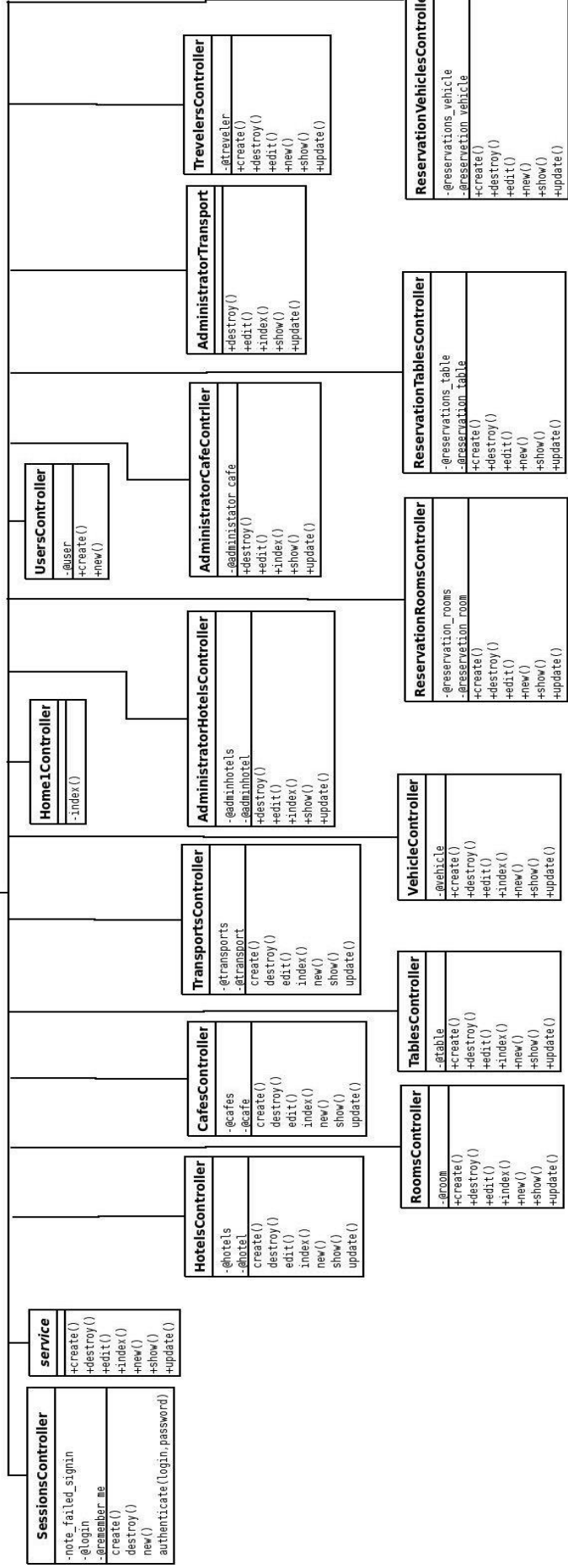
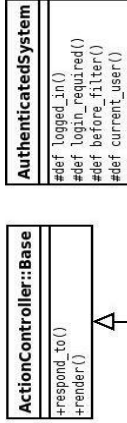
Структура програмного забезпечення складається класів. Класи в свою чергу діляться на класи моделі, класи контролери, класи види (архітектура MVC (Model-View-Controller)). Діаграму класів моделей представлено у підрозділі 2.1.4 на рисунку 2.4 .

Виділимо такі класи- моделі:

- 1) User,
- 2) Treveler,
- 3) AdministratorTransportComplex,
- 4) AdministratorCafe,
- 5) AdministratorHotel,
- 6) Service,
- 7) Hotel,
- 8) Room,
- 9) Cafe,
- 10) Table,
- 11) TransportComplex,
- 12) Vechicle,
- 13) ReservationHotel,

14) ReservationTransport,

15) ReservationCafe.



Всі класи моделі наслідуються від класу ActionRecord.

Класи- моделі мають між собою асоціативні і агрегаційні відношення:

1) Класи TransportComplex і Vehicle зв'язані агрегацією з чисельністю один до багатьох. Один транспортний комплекс містить багато транспортних засобів. Транспортний засіб являється частиною транспортного комплексу.

2) Класи Vehicle і ReservationVehicle зв'язані асоціацією з кратністю багато до багатьох.

3) Класи ReservationVehicle і Treveler зв'язані асоціацією з кратністю багато до одного. Один турист може оформити багато замовлень транспорту. Одне замовлення не може оформити багато туристів.

4) Класи Cafe і Table зв'язані агрегацією з кратністю один до багатьох. Одне кафе містить багато стілів. Стіл являється частиною кафе.

5) Класи Table і ReservationTable зв'язані асоціацією з кратністю багато до багатьох.

6) Класи ReservationTable і Treveler зв'язані асоціацією з кратністю багато до одного. Один турист може оформити багато замовлень столів. Одне замовлення не може оформити багато туристів.

7) Класи Hotel і Room зв'язані агрегацією(композицією) з кратністю один до багатьох. Один готель містить багато готельних номерів. Готельний номер являється частиною готелю і його життєвий цикл співпадає з життєвим циклом готелю.

8) Класи Room і ReservationRoom зв'язані асоціацією з кратністю багато до багатьох.

9) Класи ReservationRoom і Treveler зв'язані асоціацією з кратністю багато до одного. Один турист може оформити багато замовлень кімнат. Одне замовлення не може оформити багато туристів.

ІС може взаємодіяти з людьми, що виступають в різних ролях: із адміністраторами готелів, кафе, транспорту, туристами. У всіх цих ролей є

спільні властивості: прізвище, ім'я, по батькові, номер телефону, email, логін, пароль, аватар. Адміністратори готелів, кафе, транспорту, туристи поводять себе однаково при реєстрації, авторизації. Вводять однакові набори даних при реєстрації (представлені в таблиці 2.16). Вводять однакові набори даних при авторизації (представлені в таблиці 2.17). Все це можна змодельовати, сказавши, що класи `Treveler`, `AdministratorTransportComplex`, `AdministratorCafe`, `AdministratorHotel` являються підкласами класу `User`.

Готель, Кафе, Транспортний комплекс містять однакові дані :

- назва;
- адреса;
- опис;
- фото;
- email;
- номер телефону

Винесемо властивості, що повторюються у класів `TransportComplex`, `Cafe`, `Hotel` в абстрактний клас `Service`, для збереження цих даних.

Отже класи- моделі мають між собою відношення наслідування:

- 1) Класи `TransportComplex`, `Cafe`, `Hotel` є нащадками абстрактного класу `Service`
- 2) Класи `Treveler`, `AdministratorTransportComplex`, `AdministratorCafe`, `AdministratorHotel` є нащадками класу `User`.
- 3) Класи `Service`, `User`, `Vehicle` і `ReservationVehicle`, `Table` і `ReservationTable`, `Room` і `ReservationRoom` є нащадками класу `ActiveRecord`.

Класи контролери:

- 1) `SessionsController`,
- 2) `ServicesController`,
- 3) `Home1Controller`,
- 4) `UsersController`,
- 5) `HotelsController`,

- 6) CafesController,
- 7) TransportsComplexController,
- 8) AdministratorHotelsController,
- 9) AdministratorCafesController,
- 10) AdministratorTransportComplexsController,
- 11) RoomsController
- 12) TablesController,
- 13) VehiclesController,
- 14) ReservationRoomsController,
- 15) ReservationTablesController,
- 16) ReservationVehiclesController,
- 17) ApplicationController,
- 18) ActionController,
- 19) AuthenticateSystemController.

Класи контролери 1-16 у вище приведеному списку наслідуються від класу 17)ApplicationController.

ApplicationController, в свою чергу наслідує поведінку від 18) ActionController і 19) AuthanticateSystemController.

Висновки до розділу 2

У другому розділі було виконано комплексне проектування вебзастосунку з урахуванням принципів системного аналізу користувацьких сценаріїв та сучасних вимог до UX/UI. На основі визначених функціональних потреб, цільових груп користувачів та результатів аналізу сценаріїв взаємодії було сформовано логічну структуру системи, що забезпечує інтуїтивну, послідовну та безперебійну навігацію між основними елементами інтерфейсу.

Було розроблено архітектуру вебзастосунку, включаючи модульну побудову інтерфейсу, інформаційну модель даних та навігаційний каркас, що відповідає ключовим користувацьким потокам. На основі CJM та User Flow сформовано оптимізовані сценарії виконання типових задач, що дозволило мінімізувати кіль-

кість кроків, когнітивне навантаження та ризики помилкових дій користувача.

Створені прототипи інтерфейсу відображають основні принципи UX-дизайну, такі як простота, доступність, візуальна ієрархія, адаптивність та узгодженість елементів. Особливу увагу приділено побудові зрозумілої системи взаємодії, яка забезпечує користувачам швидке знаходження необхідної інформації та зручне виконання цільових операцій.

У межах розділу також проведено попередню експертну оцінку юзабіліті, що дозволила виявити потенційні недоліки та закласти основу для подальшої оптимізації інтерфейсу. Отримані результати підтвердили ефективність застосованого підходу та відповідність проєктних рішень вимогам користувачів та поставленим задачам дослідження.

Таким чином, у розділі було сформовано цілісну концепцію та проєктну модель вебзастосунку, які надалі слугуватимуть фундаментом для розробки, тестування та фінальної оптимізації UX системи.

РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ КОРИСТУВАЦЬКИХ СЦЕНАРІЇВ ТА ОПТИМІЗАЦІЯ UX ВЕБЗАСТОСУНКІВ

В результаті розробки проекту інформаційної системи було розроблено програмне забезпечення, яке надає можливість віддалено:

- дізнатись інформацію про місця проживання, ціни, рівень комфорту;
- дізнатись інформацію про транспорт, ціни, розклад маршрутів, зміни;
- дізнатись інформацію про місця харчування, ціни, меню;
- зареєструватись постачальникам послуг;
- постачальникам послуг викласти інформацію про вільні місця;
- бронювати послуги (проживання, транспорт, харчування);
- відпочиваючим обмінюватись відгуками;
- дізнатись історію міста, переглянути зручну карту, фото, статті, новини місцевості; переглядати інформацію про погоду.

Програмне забезпечення написано сучасною об'єктно-орієнтованою інтерпретованою мовою програмування Ruby з використанням Agile framework Ruby on Rails (ROR). Ефективність ROR полягає в архітектурі MVC (Model-View-Controller).

Зберігається програмний код і запущена програма Rails на орендованій ЕОМ хостингу (hosting). База даних зберігаються на сервері баз даних, який також розміщений на орендованій ЕОМ хостингу.

Технічні засоби, що беруть участь у процесі — це ЕОМ, інтернет- мережа.

На хостингу розміщається http- сервер Apache, Rails програма, база даних MySql.

Висновки до розділу 3

У третьому розділі було реалізовано програмну частину вебзастосунку відповідно до розроблених у попередньому розділі моделей, архітектурних рішень та результатів системного аналізу користувацьких сценаріїв. Створено функціональний прототип, що забезпечує повну підтримку основних сценаріїв взаємодії користувачів, включаючи навігацію, пошук інформації, виконання ключо-

вих операцій та роботу з динамічним контентом.

У процесі розробки використано сучасні вебтехнології та фреймворки, що забезпечують високу продуктивність, масштабованість та адаптивність застосунку. Реалізація інтерфейсу виконана з урахуванням принципів UX-дизайну: забезпечено читабельність, логічну структуру сторінок, оптимальну візуальну ієрархію, доступність для різних категорій користувачів та адаптивну роботу на різних типах пристроїв.

Особливу увагу приділено оптимізації користувацьких шляхів: скорочено кількість взаємодій для виконання типової задачі, підвищено зрозумілість елементів управління, зменшено когнітивне навантаження, забезпечено швидкий зворотний зв'язок системи. Проведено первинне юзабіліті-тестування створеного застосунку, яке дало змогу виявити окремі зони покращення та виконати доопрацювання інтерфейсу.

У ході роботи підтверджено ефективність підходу, заснованого на системному аналізі користувацьких сценаріїв, який дав можливість створити зручний, інтуїтивно зрозумілий та функціонально збалансований вебзастосунок. Реалізоване рішення відповідає вимогам користувачів, технічним критеріям проєкту та стало основою для подальшої експлуатації та вдосконалення системи.

Таким чином, у розділі завершено повний цикл розробки вебзастосунку — від впровадження функціональних модулів до інтеграції UX-оптимізованих інтерфейсних рішень, що забезпечують високий рівень зручності та ефективності роботи кінцевих користувачів.

РОЗДІЛ 4. ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ РОЗРОБКИ

4.1. Мета та принципи техніко-економічного обґрунтування

Техніко-економічне обґрунтування (ТЕО) є важливим етапом у впровадженні інформаційних систем, оскільки дає змогу визначити доцільність, економічну вигідність та ефективність розробки вебзастосунку. Даний розділ містить аналіз технічних передумов, обсягів ресурсів, витрат на реалізацію, очікуваних економічних вигід та терміну окупності проєкту. Метою ТЕО є підтвердження раціональності впровадження системи для підтримки аналізу користувацьких сценаріїв та оптимізації UX.

4.2. Технічні передумови та ресурсне забезпечення

Для розробки вебзастосунку необхідно забезпечити відповідні технічні засоби, інструменти та середовище виконання. До основних ресурсів належать:

4.2.1. Апаратне забезпечення

Персональні комп'ютери розробників (CPU від 4 ядер, RAM 8–16 GB). Сервер або хостинг для розгортання (віртуальний сервер із параметрами 2 vCPU, 4–8 GB RAM, SSD 40–80 GB). Мобільні та десктопні пристрої для тестування. Мережеве обладнання та стабільний інтернет.

4.2.2. Програмне забезпечення

Середовище розробки (VS Code). Фреймворки: React / Vue.js, Node.js. Системи контролю версій (GitHub, GitLab). Інструменти UX-аналізу (Figma, Google Analytics, Hotjar). Серверне ПЗ (Nginx, Docker за потреби). Бази даних (MySQL/PostgreSQL).

4.2.3. Кадрові ресурси

UX/UI дизайнер, фронтенд-розробник, бекенд-розробник, аналітик/системний інженер, тестувальник.

За умови невеликого масштабу проєкт може реалізовуватися командою 2–3 спеціалістів.

4.3. Оцінка витрат на розробку

Витрати на створення вебзастосунку включають заробітну плату виконавців, придбання інструментів, оренду серверів та накладні витрати.

4.3.1. Витрати на оплату праці

Середня тривалість розробки — 10–12 тижнів.

Посада	Кількість годин	Тариф, грн/год	Сума, грн
UX/UI дизайнер	120	350	42 000
Фронтенд-розробник	220	400	88 000
Бекенд-розробник	200	400	80 000
Тестувальник	80	300	24 000
Аналітик/менеджер	60	380	22 800
Разом:	—	—	256 800 грн

4.3.2. Витрати на програмні засоби та інструменти

Більшість інструментів має безкоштовні версії, але можуть знадобитися преміум-функції.

Інструмент	Вартість, грн
Хмарний сервер (3 місяці)	1 800
Figma Pro (місяць × 3)	1 500
Аналітичні сервіси (GA/HJ)	0–500
Разом:	3 300 грн

4.3.3. Накладні витрати

Включають інтернет, комунікації, офісні витрати, обладнання $\approx$ 5 000 грн

4.3.4. Загальна сума витрат

Загальні інвестиції в розробку $\approx$ 265 000 – 270 000 грн

4.4. Економічна доцільність та ефективність впровадження

Впровадження вебзастосунку забезпечує такі вигоди:

4.4.1. Прямі економічні вигоди

Зменшення витрат на підтримку, завдяки оптимізованим користувацьким

сценаріям (до 20–30 %). Зростання продуктивності роботи користувачів. Зменшення часу на виконання ключових операцій на 30–50 %. Можливість монетизації (реклама, преміум-функції, партнерські програми).

4.4.2. Непрямі вигоди

Підвищення якості UX → більша лояльність та утримання користувачів. Покращення іміджу сервісу. Зростання конверсій (реєстрацій, покупок, взаємодій) на 15–40 %. Зменшення навантаження на службу підтримки.

4.5. Аналіз окупності проєкту

При консервативній моделі монетизації реклама/партнерські програми дають 25–40 тис. грн/місяць, преміум-функції можуть давати 15–30 тис. грн/місяць.

Очікуваний місячний дохід: 40–70 тис. грн.

Термін окупності $\approx$ 4–6 місяців після запуску.

При активній аудиторії окупність може скоротитися до 3 місяців.

Висновки до розділу 4

Проведений аналіз показав, що впровадження вебзастосунку, розробленого на основі системного аналізу користувацьких сценаріїв та принципів UX-оптимізації, є технічно реалістичним та економічно доцільним. Проєкт вимагає помірних інвестицій, не потребує дорогого обладнання і може бути реалізований невеликою командою спеціалістів.

Очікувані економічні вигоди, включаючи збільшення продуктивності, покращення взаємодії користувачів та потенційні доходи від монетизації, значно перевищують затрати на розробку. Розрахований термін окупності свідчить про високу ефективність проєкту та його перспективність для подальшого масштабування.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

5.1 Загальні положення

Під охороною праці розуміється система збереження життя й здоров'я працівників у процесі трудової діяльності, що включає правові, соціально-економічні, організаційно-технічні, санітарно-гігієнічні, лікувально-профілактичні, реабілітаційні й інші заходи.

Ці заходи мають своєю ціллю створення умов праці, що відповідають вимогам збереження життя й здоров'я працівників у процесі трудової діяльності. На практиці під охороною праці нерідко мають на увазі техніку безпеки, виробничу санітарію й т.і. Така позиція помилкова. Охорона праці - це система заходів, де техніка безпеки або гігієна праці є її складовими.

Суттєвий вплив на стан організму працівника, його працездатність здійснює мікроклімат (метеорологічні умови) у виробничих приміщеннях, під яким розуміють клімат внутрішнього середовища цих приміщень, що визначається діючою на організм людини сукупністю температури, вологості, руху повітря та теплового випромінювання нагрітих поверхонь.

Метою проведення цієї роботи є детальний розгляд та розрахунки системи освітлення та засобів захисту від електромагнітного випромінювання для певного виробничого приміщення. Також у роботі маємо висвітити питання вимог нормативних документів щодо безпечних та здорових умов праці у виробничих приміщеннях з використанням комп'ютерної техніки та детально розглянути зменшення електромагнітного випромінювання.

4.2 Аналіз шкідливих та небезпечних факторів

Негативний вплив на роботу інформаційної системи інформаційного забезпечення міста, причини виникнення, санітарні норми

У ДСТ 12.0.003-74 "ССБТ. Небезпечні і шкідливі виробничі фактори. Класифікація", приводиться класифікація елементів умов праці, що виступають у ролі небезпечних і шкідливих виробничих факторів. Вони підрозділяються на 4 групи: фізичні, хімічні, біологічні і психофізіологічні.

Експлуатація ЕОМ супроводжується виникненням ряду шкідливих і небезпечних виробничих факторів фізичного і психофізіологічного характеру. Працівники, що обслуговують ЕОМ, піддаються під час роботи впливові наступних небезпечних і шкідливих факторів:

фізичних: підвищення рівня шуму, статичної електрики, електромагнітних випромінювань, підвищена напруженість електричних і магнітних полів, недостатня освітленість робочого місця, підвищена яскравість світла;

психофізіологічних: фізичні і нервово-психічні перевантаження (розумова перенапруга, монотонність праці, емоційні перевантаження).

Санітарними нормами і правилами встановлені наступні припустимі значення:

рівень шуму: 50 дБ;

гранично припустима напруженість електростатичного поля на робочому місці при тривалості впливу на людину до 1 години: 60 кВ/м.

Інтенсивність впливу неіонізуючих магнітних полів на обслуговуючий персонал (оператори, програмісти) залежить від потужності джерел випромінювання, режиму роботи, конструктивної особливості випромінюючих пристроїв, технічного стану устаткування, розташування робочого місця й ефективності захисних заходів. Гранично припустима напруженість електричної складової електромагнітного поля складає 10 В/м, магнітної складової 0.3 А/м. Тривалий вплив неіонізуючих електромагнітних випромінювань на організм людини з інтенсивністю перевищуючі гігієнічні нормативи приводить до функціональних змін з боку нервової, серцево-судинної, нейрогуморальної й іншої систем організму.

Фізичні перевантаження найчастіше торкаються зорової системи або кістково-м'язову систему (плечовий суглоб, шия, верхня частина спини). У цілому користувачі ЕОМ не зіштовхуються з якимсь особливим психічним стресом або емоційними проблемами, дослідження показують одиничні випадки дуже напружений, зухвалий стрес, роботи на ЕОМ. Проте, рекомендується вживати заходів до запобігання надмірного дискомфорту і стомлення працюючих на ЕОМ.

У більшості користувачів ЕОМ виникає кістково-м'язовий дискомфорт через статичну напругу м'язів, незручної пози або часто повторюваних одноманітних рухів. імовірність виникнення стійких порушень дуже мала, однак монотонні рухи при роботі на ЕОМ можуть викликати хронічні захворювання.

Важливим фактором надійної роботи засобів ВТ і високої працездатності обслуговуючого персоналу є кліматичні умови. Під кліматичними умовами виробничого середовища відповідно ДО ДЕРЖСТАНДАРТУ 12.1.005-76 розуміють сполучення температури, відносній вологості і швидкості руху повітря. Ці мікрокліматичні фактори як кожний окремо, так і в різних сполученнях впливають на функціональну діяльність людини, його самопочуття і здоров'я і на надійність роботи засобів ВТ.

Температура повітря є одним з основних параметрів, що характеризують тепловий стан мікроклімату. Основними джерелами теплоти в приміщеннях є ЕОМ і допоміжне устаткування, прилади освітлення, що обслуговує персонал. Потрібно враховувати і зовнішні джерела надходження теплоти (теплоту, що надходить через вікна від сонячної радіації і приплив теплоти через непрозорі конструкції, що обгороджують).

На роботу устаткування великий вплив робить відносна вологість повітря. При вологості повітря до 40% підвищується знос магнітних голівок, виходить з ладу ізоляція проводів. При вологості повітря більш 75-80% знижується опір ізоляції, змінюються робочі характеристики елементів ЕОМ, зростає інтенсивність їхніх відмовлень.

Швидкість руху повітря - вектор усередненої швидкості переміщення повітряних потоків під дією різних сил, що спонукують, також впливає на роботу друкувальних пристроїв.

Атмосферний тиск у приміщеннях також нормується. При зниженому тиску повітря погіршується відвід тепла від елементів ЕОМ, знижуються ізоляційні властивості повітря. Перераховане вище обумовлює підвищені вимоги до кліматичних умов середовища в приміщеннях з ЕОМ.

Оптимальні і припустимі норми мікроклімату:

температура повітря 18-25 °С;
відносна вологість повітря 40-70%;
швидкість руху повітря не більш 0.3 м/с;
атмосферний тиск 1013.25 + 266 кпа.

Природне і штучне освітлення в приміщеннях регламентується СНіП 23-05-95 у залежності від характеру зорової роботи, системи і виду освітлення, тла, контрасту об'єкта з тлом. Характеристика зорової роботи визначається найменшим розміром об'єкта розрізнення.

Штучне освітлення нормується кількісними і якісними показниками (показниками засліпленості і дискомфорту, коефіцієнтом пульсації освітленості).

Природне освітлення характеризується тим, що створювана освітленість змінюється в залежності від часу доби, року, метеорологічних умов. Тому як критерій оцінки природного освітлення прийнята відносна величина коефіцієнт природної освітленості.

4.3 Розрахунок системи штучного освітлення

У приміщенні необхідно організувати змішане освітлення, тобто сполучення природного і штучного освітлення. Природне освітлення створюється бічним освітленням через вікно. Штучне освітлення використовується при недостатньому природному освітленні. У даному приміщенні використовується загальне штучне освітлення.

Розрахунок його здійснюється по методу світлового потоку з урахуванням потоку, відбитого від стін і стелі.

Потрібно зробити розрахунок загального освітлення виробничого приміщення. Довжина $A = 3,0$ м, ширина $B = 6,0$ м, висота $H = 3$ м. Висота робочої поверхні $h_p = 1$ м. Для освітлення використовується світильник УПМ-15. Мінімальна освітленість лампи накаливання по нормах $E_{min} = 100$ лк. Коефіцієнт відображення стелі $S_p = 50\%$, стін $S_c = 30\%$, робочої поверхні $S_r = 10\%$. Напруга мережі 220 В. $K_z = 1,3$ та $Z = 1,15$.

Відстань від стелі до робочої поверхні:

$$H_0 = H - h_p = 3 - 1 = 2 \text{ м}$$

Відстань від стелі до світильника:

$$h_c = 0,2 H_0 = 0,2 * 2 = 0,4 \text{ м}$$

Висота підвісу світильника над освітленою поверхнею:

$$h = H_0 - h_c = 2 - 0,4 = 1,6 \text{ м}$$

Висота підвісу світильника над підлогою:

$$H_n = h + h_p = 1,6 + 1 = 2,6 \text{ м}$$

Для досягнення найбільшої рівномірності освітлення приймаємо відношення

$$L / h = 1,5.$$

Відстань між центрами світильників:

$$L = 1,5 h = 1,5 * 1,6 = 2,4 \text{ м}$$

Необхідна кількість світильників:

$$N = S / L^2 = (3 * 6) / (2,4)^2 = 18 / 5,76 = 3,125$$

Приймаємо кількість світильників: 3.

Індекс приміщення:

$$i = (A B) / (h (A + B)) = (3 * 6) / (1,6 (3 + 6)) = 18 / 14,4 = 1,25$$

Коефіцієнт використання світлового потоку $\eta = 0,69$

Світловий потік однієї лампи:

$$\Phi = (E_{\min} S K Z) / (N \eta) = (100 * 18 * 1,3 * 1,15) / (3 * 0,69) = 1300 \text{ лм}$$

По найденному световому потоку из таблицы “Параметры ламп накаливания общего назначения с расчётными напряжениями 130 и 220 В” выбираем лампу накаливания мощностью 300 Вт, имеющую световой поток 2090 лм, наиболее близкий к расчётному.

Фактична освітленість:

$$E_f = E_{\min} \Phi_l / \Phi_p = 100 * 2090 / 1300 = 160,7 \text{ лк}$$

Загальна потужність:

$$P_{\text{общ}} = P_l N = 300 * 3 = 900 \text{ Вт} = 0,9 \text{ кВт}$$

4.4 Розробка заходів щодо зменшення впливу шкідливих та небезпечних факторів (щодо захисту від електромагнітного випромінювання)

Під впливом ЕМП та випромінювань спостерігаються загальна слабкість, підвищена втома, пітливість, сонливість, а також розлад сну, головний біль, біль в ділянці серця. З'являється роздратування, втрата уваги, зростає тривалість мовно-рухової та зоровомоторної реакцій підвищується межа нюхової чутливості. Виникає ряд симптомів, які є свідченням порушення роботи окремих органів — шлунку, печінки селезінки, підшлункової та інших залоз. Пригнічуються харчовий та статевий рефлекс.

Реєструються зміни артеріального тиску, частота серцевого ритму форма електрокардіограми. Це свідчить про порушення діяльності серцево-судинної системи. Фіксуються зміни показників білкового та вуглеводного обміну, збільшується вміст азоту в крові та сечі, знижується концентрація альбуміну та зростає вміст глобуліну, збільшується кількість лейкоцитів, тромбоцитів, виникають й інші зміни складу крові.

Для зменшення впливу ЕМП на персонал та населення, яке знаходиться у зоні дії радіоелектронних засобів, потрібно вжити ряд захисних заходів. До їх числа можуть входити організаційні, інженерно-технічні та лікарсько-профілактичні.

Здійснення організаційних та інженерно-технічних заходів покладено передусім на органи санітарного нагляду. Разом з санітарними лабораторіями підприємств та установ, які використовують джерела електромагнітного випромінювання, вони повинні вживати заходів з гігієнічної оцінки нового будівництва та реконструкції об'єктів, котрі виробляють та використовують радіо засоби, а також нових технологічних процесів та обладнання з використанням ЕМП, проводити поточний санітарний нагляд за об'єктами, які використовують джерела випромінювання, здійснювати організаційно-методичну роботу з підготовки спеціалістів та інженерно-технічний нагляд.

Ще на стадії проектування повинне бути забезпечене таке взаємне розташування випромінюючих та опромінюваних об'єктів, яке б зводило до мінімуму інтенсивність опромінення. Оскільки повністю уникнути опромінення неможливо, потрібно зменшити ймовірність проникнення людей у зони з високою інтен-

сивністю ЕМП, скоротити час перебування під опроміненням. Потужність джерел випромінювання мусить бути мінімально потрібною.

Виключно важливе значення мають інженерно-технічні методи та засоби захисту: колективний (група будинків, район, населений пункт), локальний (окремі будівлі, приміщення) та індивідуальний. Колективний захист спирається на розрахунок поширення радіохвиль в умовах конкретного рельєфу місцевості. Економічно найбільш доцільно використовувати природні екрани — складки місцевості, лісонасадження, нежитлові будівлі. Встановивши антену на горі, можна зменшити інтенсивність поля, яке опромінює населений пункт, у багато разів. Аналогічний результат дає відповідна орієнтація діаграми напрямленості, особливо високо спрямованих антен, наприклад, шляхом збільшення висоти антени. Але висока антена складніша, дорожча, менш стійка. Крім того, ефективність такого захисту зменшується з відстанню.

При захисті від випромінювання екрана повинне враховуватись затухання хвилі при проходженні через екран (наприклад, через лісову смугу). Для екранування можна використовувати рослинність. Спеціальні екрани у вигляді відбивальних і радіопоглинальних щитів дорогі, малоефективні і використовуються дуже рідко.

Локальний захист дуже ефективний і використовується часто. Він базується на використанні радіозахисних матеріалів, які забезпечують високе поглинання енергії випромінювання у матеріалі та віддзеркалення від його поверхні. Для екранування шляхом віддзеркалення використовують металеві листи та сітки з доброю провідністю. Захист приміщень від зовнішніх випромінювань можна здійснити завдяки обклеюванню стін металізованими шпалерами, захисту вікон сітками, металізованими шторами. Опромінення у такому приміщенні зводиться до мінімуму, але віддзеркалене від екранів випромінювання розповсюджується у просторі та потрапляє на інші об'єкти.

До інженерно-технічних засобів захисту також належать конструктивна можливість працювати на зниженій потужності в процесі налагоджування, регулювання та профілактики. Робота на еквівалент налагоджування. Дистанційне

керування.

Радіопоглинальні матеріали можуть використовуватися для захисту навколишнього середовища від ЕМП, яке генерується джерелом, що знаходиться в екранованому об'єкті. Крім того, радіопоглиначами для захисту від віддзеркалення личкуються стіни безлунких камер — приміщень, де випробовуються випромінювальні пристрої. Радіопоглинаючі матеріали використовуються в кінцевих навантаженнях, еквівалентах системах.

Висновки до розділу 5

Охорона праці відіграє важливу роль як соціальний чинник, оскільки якими б вагомими не були трудові здобутки, вони не можуть компенсувати людині втраченого здоров'я, а тим більше життя. Окрім соціального, охорона праці має важливе економічне значення — це й висока продуктивність праці, зниження витрат на оплату лікарняних, компенсацій за важкі та шкідливі умови праці тощо.

Зважаючи на все вищесказане, необхідно суворо дотримуватись всіх норм мікроклімату в виробничому приміщенні. Оптимальними (комфортними) вважаються такі умови праці, за котрих має місце найвища працездатність і гарне самопочуття.

У даній роботі було розраховано показники природного та штучного освітлення для обраного виробничого приміщення та з'ясовано заходи щодо захисту від електромагнітного випромінювання.

РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1. Загальні положення та актуальність екологічної безпеки у сфері ІТ

У сучасних умовах цифрової трансформації інформаційні технології відіграють важливу роль у бізнесі, освіті та суспільних процесах. Хоча діяльність у сфері розробки вебзастосунків безпосередньо не пов'язана з виробництвом чи забрудненням довкілля, вона має опосередкований екологічний вплив, що пов'язаний із використанням енергоресурсів, серверних потужностей, комп'ютерного обладнання, електронних відходів та споживанням електроенергії.

У межах роботи над системним аналізом користувацьких сценаріїв та оптимізацією UX вебзастосунків важливо враховувати екологічні аспекти, пов'язані з енергоефективністю цифрових продуктів, їх оптимізацією та скороченням непотрібного навантаження на інфраструктуру.

6.2. Екологічні аспекти розробки вебзастосунків

Створення та експлуатація вебзастосунків супроводжується такими екологічними факторами:

6.2.1. Енергоспоживання серверів

Хостинг, обчислювальні операції та робота баз даних потребують електроенергії. Сучасні дата-центри використовують системи охолодження, які також збільшують енерговитрати. Оптимізація коду, зниження навантаження та використання CDN сприяють зменшенню ресурсів, необхідних для роботи застосунку.

6.2.2. Енергоспоживання кінцевих користувачів

Важкі сторінки, неоптимізовані зображення та низька швидкість завантаження призводять до збільшення часу роботи пристроїв користувачів, а отже — більшого споживання енергії. UX-оптимізація сприяє скороченню зайвих дій та підвищенню ефективності використання системи.

6.2.3. Електронні відходи

Розробка програмного забезпечення опосередковано спричиняє появу еле-

ктронного брухту, адже потребує оновлення техніки. Важливо забезпечувати тривалий життєвий цикл обладнання та коректну утилізацію.

6.3. Енергоефективність та екологічні принципи у проєктуванні вебзастосунку

У процесі створення вебзастосунку застосовано такі заходи.

1. Оптимізація швидкодії та зменшення навантаження
мінімізація обсягу файлів (CSS, JS, зображення);
використання lazy-loading для медіафайлів;
оптимізація запитів до бази даних;
кешування контенту на стороні клієнта та сервера.
Це зменшує час роботи серверів і енергоспоживання пристроїв користувачів.

2. Легка та чиста архітектура інтерфейсу
Інтуїтивний та оптимізований UX дає змогу скоротити:
кількість взаємодій користувачів із системою,
кількість повторних запитів,
тривалість сесій.

Це позитивно впливає на загальні витрати енергії.

3. Використання екологічних стандартів веброзробки
застосування ресурсоефективних технологій;
використання хостингів із «green energy» політикою (за можливості);
мінімізація дублювання даних та непотрібної обробки.

6.4. Вимоги техніки безпеки та охорони праці при роботі з комп'ютерною технікою

Розробка вебзастосунку виконується у приміщеннях з використанням комп'ютерів, тому дотримання правил безпеки є обов'язковим.

Основні вимоги:

Ергономіка робочого місця
коректне розташування монітора (відстань 50–70 см);
правильна висота стільця та стола;

достатнє освітлення без відблисків.

Мікроклімат

температура 18–24°C;

вологість 40–60%;

регулярне провітрювання.

Режим праці

перерви кожні 45–60 хвилин;

виконання вправ для очей та постава.

Електробезпека

використання справних розеток та кабелів;

заборона роботи з пошкодженими пристроями.

Пожежна безпека

наявність вогнегасника;

відсутність легкозаймистих матеріалів біля обладнання.

6.5. Заходи щодо зменшення негативного впливу IT-інфраструктури на довкілля

Для скорочення екологічного впливу вебзастосунків пропонується використання серверів із високим класом енергоефективності. Застосування контейнеризації (Docker) для оптимального використання ресурсів. Зменшення дублювання даних у БД. Перехід на хмарні платформи, що використовують відновлювані джерела енергії. Відповідальна утилізація та ремонт обладнання. Продовження терміну служби комп'ютерної техніки. Впровадження екологічного менеджменту в IT-проєктах.

Висновки до розділу 6

У ході дослідження встановлено, що розробка вебзастосунку хоча й не має прямого промислового впливу на навколишнє середовище, однак супроводжується низкою екологічних факторів, пов'язаних із використанням енергоресурсів та IT-інфраструктури. Завдяки застосуванню оптимізації UX, ефективних алгоритмів, сучасних фреймворків та ресурсозберігаючих технологій вдається значно

зменшити навантаження на сервери та пристрої користувачів, що позитивно впливає на екологічні показники.

Дотримання вимог техніки безпеки та охорони праці забезпечує безпечні умови роботи для розробників. Реалізовані екологічні заходи доводять, що цифрові продукти можуть бути енергоефективними, безпечними та дружніми до довкілля.

ВИСНОВКИ

У магістерській роботі проведено комплексне дослідження процесів взаємодії користувачів із вебзастосунками на основі системного аналізу, методів UX-дизайну та сучасних інструментів оцінювання ефективності інтерфейсів. Поставлені у роботі цілі досягнуті повністю.

У результаті дослідження:

- виконано системний аналіз предметної області, визначено ключові проблеми користувацької взаємодії та чинники, що впливають на якість UX;
- сформовано модель користувацьких сценаріїв, на основі яких ідентифіковано критичні точки навантаження, когнітивні бар'єри та поведінкові патерни користувачів;
- розроблено та реалізовано оптимізований UX для вебзастосунку з урахуванням принципів юзабіліті, доступності, адаптивності та швидкодії;
- створено прототип і функціональну реалізацію інтерфейсу, що забезпечує зменшення часу виконання цільових дій користувача, підвищення інтуїтивності навігації та зниження кількості помилок;

проведено оцінювання ефективності запропонованих рішень, що продемонструвало покращення ключових метрик UX (task success rate, time-on-task, satisfaction score).

Результати роботи підтверджують, що застосування системного підходу до проєктування користувацького досвіду дозволяє підвищити якість вебзастосунків, зробити їх більш зручними, доступними та ефективними. Запропоновані підходи та розроблені рішення можуть бути використані в подальших наукових дослідженнях, а також у практиці створення сучасних цифрових продуктів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Singh, S. UX Design Process Simplified: From User Research to Usability. — Packt Publishing, 2022. — 320 p. [Електронний ресурс].
2. Dong, Y., Zhu, R., Tian, Q. A Scenario Interaction-centered Conceptual Information Model for UX Design of User-oriented Product-service System // *Procedia CIRP*. — 2019. — Vol. 83. — P. 335–338. [Електронний ресурс].
3. Revi, A.T., et al. A Systematic Analysis of User Experience Dimensions for Interactive Digital Narratives. — Univ. of Southampton, 2020. — (збірник/звіт). [Електронний ресурс].
4. Ажищев, В. Ф., Качурка, Д. І., Леус, С. О., & Морарь, М. І. (2025). Дослідження методів системи динамічного призначення ролей у системах керування доступом підприємства. У матеріалах VI Міжнародної науково-практичної інтернет-конференції «Інформаційні технології: моделі, алгоритми, системи (ITMAS – 2025)» (16–17 листопада 2025 р.). Миколаїв: НУК імені адмірала Макарова С. 245-247
<https://itconf.nuos.edu.ua/2025/publications/investigation-of-methods-for-dynamic-role-assignment-in-enterprise-access-control-systems/>
5. Iqbal, M.W., et al. Usability and optimization of online apps in user's context // *International Journal* (open access). — 2024. — (стаття, аналіз UX, практичні рекомендації). [Електронний ресурс].
6. Rocha, A., et al. (eds.) *Web Information Systems and Technologies — WEBIST 2017. Selected Papers*. — Springer, 2017. — (матеріали конференції WEBIST 2017).
7. Marcus, A., et al. (eds.) *Design, User Experience, and Usability: Theory and Practice — DUXU 2017*. — Springer, 2017. — (збірник наукових праць з теорії й практики UX).
8. Hartson, R., Pyla, P. *The UX Book: Agile UX Design for a Quality User Experience*. — 2nd ed., Morgan Kaufmann, 2019. — 800 p. [Підручник/посібник для практиків UX].
9. Interaction Design Foundation. What Are User Flows? — *Interaction-Design.org*, (оновлюється). — Онлайн-стаття/посібник (практичний опис User Flow та приклади). [Електронний ресурс].

10. Evaluation of Web Applications based on UX Parameters / автори. — наукова стаття (2018–2019) — аналіз оцінювання вебзастосунків за UX-параметрами. [Електронний ресурс].
11. Evaluation of user experience: a systematic literature review / автори. — систематичний огляд (опубл. ~2023–2024). — DOI/online. [Електронний ресурс].
12. User Experience (UX) Design and Mobile Applications — науково-практичний огляд, 2019–2024 (PDF). — Аналіз UX-підходів для мобільних та вебзастосунків. [Електронний ресурс].
13. Nielsen Norman Group. User Journeys vs. User Flows — NN/g article, 2023. — Практичні рекомендації щодо використання CJM і User Flow у UX-процесах. [Електронний ресурс].
14. Handley, A. User flow is the new wireframe // UX Collective, 2018. — Стаття про роль user-flows у сучасному дизайні інтерфейсів. [Електронний ресурс].
15. Figma. What is a user flow? Why it's important in UX design — Figma Resource, practical guide (оновлюється). — Інструментальні поради щодо створення user flows. [Електронний ресурс].
16. Speicher, M. Conversion rate & average order value are not UX metrics // UX Design.cc, 2023 — Критичний аналіз метрик UX та рекомендації щодо вимірювання успішності сценаріїв. [Електронний ресурс].

ДОДАТОК А - Технічне завдання

А.1 Загальні відомості

Найменування системи « АРМ адміністратора Web системи інформаційний портал міста». Система розрахована на людей зі всіх куточків країни і світу. що бажають шляхом отримання адекватної, докладної, нової інформації та віддалено замовити послуги. А також програма розрахована на користування нею адміністраторів пунктів постачання послуг.

А.2 Призначення і цілі

Інформаційна система призначена для користування нею жителів та туристів і постачальників послуг.

Головною ціллю створення ІС є підвищення інформативності, а отже рівня послуг та прибутку міста.

Побічною ціллю розробки ІС є написання і захист дипломного проекту.

А.3 Вимоги до інформаційної системи

Вимоги до функціональних характеристик

Дана ІС повинна надавати можливість віддалено:

дізнатись інформацію про місця проживання, ціни, рівень комфорту;

дізнатись інформацію про транспорт, ціни, розклад маршрутів, зміни;

дізнатись інформацію про місця харчування, ціни, меню;

zareєstrуватись постачальникам послуг;

постачальникам послуг викласти інформацію про вільні місця;

бронювати послуги (проживання, транспорт, харчування);

відпочиваючим обмінюватись відгуками;

дізнатись історію міста, переглянути зручну карту, фото, статті, новини місцевості;

переглядати інформацію про погоду.

Вимоги до надійності :

ІС повинна безвідмовно працювати. Допустимим вважається середній % працездатності системи за місяць не менше 99%.

ІС не повинна втрачати дані-середній % втрати пакетів за місяць повинен бути не більше 0,2%.

ІС повинна швидко обробляти інформацію. Допустимим на даному етапі функціонування системи вважається обробка до 100 заявок за хвилину.

А.4 Склад та зміст робіт по створенню системи

Створення ІС складається з визначення вимог, проектування, написання коду програми, проведення тестування. Всі моделі програми повинні на 100% бути покриті тестами білого ящика- Unit test.

Також існує необхідність залучення до взаємодії адміністраторів послуг.

А.5 Порядок контролю і прийому робіт

Контролюється робота випробуваннями програми. Випробування проводяться відповідно до програми і методики випробувань. Приймає роботу керівник дипломного проекту ст.викладач кафедри ІУСТ Беркунський Євгеній Юрієвич. Робота виконується з року. Дата здачі роботи року. У випадку виявлення помилок або невідповідності технічному завданню, протягом семи днів проводиться випробування.

А.6 Вимоги до змісту і складу робіт по підготовці об'єкта автоматизації до вводу системи в дію

Вимоги до змісту і складу робіт по підготовці об'єкта автоматизації до вводу системи в дію представимо у вигляді таблиці А.1.

Таблиця А.1- Вимоги до змісту і складу робіт

	Посадова особа	Робота	Строк здачі
	розробник ІС	визначити вимоги до системи	01.02.1 3
		розробити проектні рішення системи	04.03.1 3

		закодувати програму	30.04.1
			3
		провести тестування	15.05.1
			3
		розробити робочу докумен- тацію	01.06.1
			3
	адміністратор ІС	залучити адміністраторів послуг	10.06.1
			3
		викласти ознайомлювальну інформацію про місто(фото, істо- ричні статті, опис)	15.06.1
			3
	адміністратор го- тельного комплексу	zareestruvatis, викласти докладну інформацію про свою організацію(контактну інформа- цію,режим роботи, ціни, наявність, рівень комфорту і т.ін.)	25.06.1
			3
	адміністратор комплексу харчування	zareestruvatis, викласти докладну інформацію про свою організацію(контактну інформа- цію, режим роботи, меню, ціни, і т.ін.)	30.06.13
	адміністратор транспорту	zareestruvatis, викласти розклад маршрутів,ціни	30.06.13

А.7Вимоги до документації

До програмної документації мають бути включені документи:

опис програми;

інструкція користувачам (адміністраторам у вигляді документу, а туристам та жителям у вигляді web-сторінки в ІС);

програма і методика випробувань.

ДОДАТОК Б - Фрагмент програмного коду

Контролер для адміністратора готелю

```
class AdminhotelsController < ApplicationController
  # GET /adminhotels
  # GET /adminhotels.xml
  def index
    @adminhotels = Adminhotel.find(:all)

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @adminhotels }
    end
  end

  # GET /adminhotels/1
  # GET /adminhotels/1.xml
  def show
    @adminhotel = Adminhotel.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @adminhotel }
    end
  end

  # GET /adminhotels/new
  # GET /adminhotels/new.xml
  def new
    @adminhotel = Adminhotel.new

    respond_to do |format|
      format.html # new.html.erb
      format.xml { render :xml => @adminhotel }
    end
  end

  # GET /adminhotels/1/edit
  def edit
    @adminhotel = Adminhotel.find(params[:id])
  end

  #GET /adminhotels
  #GET /adminhotels.xml
  def to_new_hotel
    format.html { render :action => "to_new_hotel" }
  end

  # POST /adminhotels
  # POST /adminhotels.xml
  def create
    @adminhotel = Adminhotel.new(params[:adminhotel])
  end
end
```

```

respond_to do |format|
  if @adminhotel.save
    flash[:notice] = 'Adminhotel was successfully created.'
    format.html { redirect_to(@adminhotel) }
    format.xml { render :xml => @adminhotel, :status => :created, :location=>
@adminhotel }
  else
    format.html { render :action => "new" }
    format.xml { render :xml => @adminhotel.errors, :status =>
:unprocessable_entity }
  end
end
end

# PUT /adminhotels/1
# PUT /adminhotels/1.xml
def update
  @adminhotel = Adminhotel.find(params[:id])

  respond_to do |format|
    if @adminhotel.update_attributes(params[:adminhotel])
      flash[:notice] = 'Adminhotel was successfully updated.'
      format.html { redirect_to(@adminhotel) }
      format.xml { head :ok }
    else
      format.html { render :action => "555" }
      format.xml { render :xml => @adminhotel.errors, :status =>
:unprocessable_entity }
    end
  end
end

# DELETE /adminhotels/1
# DELETE /adminhotels/1.xml
def destroy
  @adminhotel = Adminhotel.find(params[:id])
  @adminhotel.destroy

  respond_to do |format|
    format.html { redirect_to(adminhotels_url) }
    format.xml { head :ok }
  end
end
end

```

Контролер для туриста

```

class TrevelersController < ApplicationController
  # GET /trevelers
  # GET /trevelers.xml
  def index
    @trevelers = Treveler.find(:all)
  end
end

```

```

    respond_to do |format|
      format.html # index.html.erb
      format.xml { render :xml => @trevelers }
    end
  end

  # GET /trevelers/1
  # GET /trevelers/1.xml
  def show
    @treveler = Treveler.find(params[:id])

    respond_to do |format|
      format.html # show.html.erb
      format.xml { render :xml => @treveler }
    end
  end

  # GET /trevelers/new
  # GET /trevelers/new.xml
  def new
    @user.resource=Treveler.new
    #@treveler = Treveler.new

    respond_to do |format|
      format.html # new.html.erb
      format.xml { render :xml => @treveler }
    end
  end

  # GET /trevelers/1/edit
  def edit
    @treveler = Treveler.find(params[:id])
  end

  # POST /trevelers
  # POST /trevelers.xml
  def create
    @treveler = Treveler.new(params[:treveler])

    respond_to do |format|
      if @treveler.save
        flash[:notice] = 'Treveler was successfully created.'
        format.html { redirect_to(@treveler) }
        format.xml { render :xml => @treveler, :status => :created, :location => @treveler }
      else
        format.html { render :action => "new" }
        format.xml { render :xml => @treveler.errors, :status => :unprocessable_entity }
      end
    end
  end
end

```

```

# PUT /trevelers/1
# PUT /trevelers/1.xml
def update
  @treveler = Treveler.find(params[:id])

  respond_to do |format|
    if @treveler.update_attributes(params[:treveler])
      flash[:notice] = 'Treveler was successfully updated.'
      format.html { redirect_to(@treveler) }
      format.xml { head :ok }
    else
      format.html { render :action => "edit" }
      format.xml { render :xml => @treveler.errors, :status => :unprocessable_entity }
    end
  end
end

# DELETE /trevelers/1
# DELETE /trevelers/1.xml
def destroy
  @treveler = Treveler.find(params[:id])
  @treveler.destroy

  respond_to do |format|
    format.html { redirect_to(trevelers_url) }
    format.xml { head :ok }
  end
end
end

```

Модель для користувача

```

require 'digest/sha1'

class User < ActiveRecord::Base
  include Authentication
  include Authentication::ByPassword
  include Authentication::ByCookieToken

  belongs_to :resource, :polymorphic=> true

  validates_presence_of :login
  validates_length_of :login, :within => 3..40
  validates_uniqueness_of :login
  validates_format_of :login, :with => Authentication.login_regex, :message =>
  Authentication.bad_login_message

  validates_format_of :name, :with => Authentication.name_regex, :message =>
  Authentication.bad_name_message, :allow_nil => true
  validates_length_of :name, :maximum => 100

  validates_presence_of :email

```

```

validates_length_of :email, :within => 6..100 #r@a.wk
validates_uniqueness_of :email
validates_format_of :email, :with => Authentication.email_regex, :message =>
Authentication.bad_email_message

```

```

attr_accessible :login, :email, :name, :password, :password_confirmation

```

```

# Authenticates a user by their login name and unencrypted password. Returns the user or nil.
#
# uff. this is really an authorization, not authentication routine.
# We really need a Dispatch Chain here or something.
# This will also let us return a human error message.
#
def self.authenticate(login, password)
  return nil if login.blank? || password.blank?
  u = find_by_login(login.downcase) # need to get the salt
  u && u.authenticated?(password) ? u : nil
end
def login=(value)
  write_attribute :login, (value ? value.downcase : nil)
end

def email=(value)
  write_attribute :email, (value ? value.downcase : nil)
end
protected
end

```

Вид для туриста

```

<h1>New treveler</h1>

```

```

<% form_for(@user.resource) do |f| %>
  <%= f.error_messages %>

```

```

  <p>
    <label for='firstName'>firstName</label>
    <%= f.text_field :firstName %>
  </p>
  <p>
    <label for ='secondName'>secondName</label>
    <%= f.text_field :secondName %>
  </p>
  <p>
    <label for='surname'>surname</label>
    <%= f.text_field :surname %>
  </p>
  <p>
    <label for='login'>login</label>
    <%=f.text_field :login %>
  </p>
  <p>

```

```
        <label for='name'>name</label>
        <%=f.text_field :name %>
    </p>
    <p>
        <label for='email'>email</label>
        <%=f.text_field :email %>
    </p>
    <p>
        <label for='password'>password</label>
        <%=f.text_field :password %>
    </p>
    <p>
        <label for='password_confirmation'>password_confirmation</label>
        <%=f.text_field :password_confirmation %>
    </p>
    <p>
        <%= f.submit "Create" %>
    </p>
<% end %>

<%= link_to 'Back', trevelers_path %>
```

ДОДАТОК В - Опис системи

В.1 Загальні відомості

Розроблено інформаційну систему- «АРМ адміністратора Web системи інформаційний портал міста».

Програмне забезпечення написано сучасною об'єктно-орієнтованою інтерпретованою мовою програмування Ruby з використанням Agile framework Ruby on Rails (ROR).

Для функціонування інформаційної системи необхідно, щоб було встановлено на сервері: ruby version 1.8.7, framework RAILS VERSION 2.3.3, сервер баз даних MySql 5.0.67; http- сервер Apache 2, http- proxy Mongrel; на клієнті: веб-браузер(Mozilla Firefox, A Web browser, Opera, Opera mini). Завдяки багатоплатформеності Ruby on Rails, цим програмним продуктом продуктом можна користуватися як в операційній системі Linux, так і в Windows.

В.2 Функціональне призначення

Інформаційна система призначена для користування нею жителів та туристів і постачальників послуг.

Функціональне призначення ІС надавати можливість віддалено:

- дізнатись інформацію про місця проживання, ціни, рівень комфорту;
- дізнатись інформацію про транспорт, ціни, розклад маршрутів, зміни;
- дізнатись інформацію про місця харчування, ціни, меню;
- zareєstrуватись постачальникам послуг;
- постачальникам послуг викласти інформацію про вільні місця;
- бронювати послуги (проживання, транспорт, харчування);
- відпочиваючим обмінюватись відгуками;
- дізнатись історію міста, переглянути зручну карту, фото, статті, новини містечовості;
- переглядати інформацію про погоду .

В.3 Опис логічної структури

Схеми алгоритмів ІС представлено у розділі 2.5.3- Алгоритм рішення.

ІС є програмним інтерфейсом. Завдяки використанню Rest (архітектура мережевих протоколів, що забезпечують доступ до інформаційних ресурсів), ІС може інтегруватись з програмами постачальників курортних послуг.

Структура програмного забезпечення складається класів. Класи в свою чергу діляться на класи моделі, класи контролери, класи види (архітектура MVC (Model-View-Controller)).

Класи моделі:

User, Treveler, AdministratorTransportComplex, AdministratorCafe, AdministratorHotel, Service, Hotel, Room, Cafe, Table, TransportCompex, Vechicle, ReservationHotel, ReservationTransport, ReservationCafe.

Класи контролери:

SessionsController, ServicesController, HomeController, UsersController, HotelsController, CafesController, TransportsComplexController, AdministratorHotelsController, AdministratorCafesController, AdministratorTransportComplexsController, RoomsController TablesController, VehiclesController, ReservationRoomsController, ReservationTablesController, ReservationVehiclesController, ApplicationController, ActionController, AuthenticateSystemController.

В. 4 Виклик і завантаження

Клієнти викликають програму віддалено з web - браузерів, установлених на їх комп'ютерах.

ДОДАТОК Г - Інструкція користувача

Галузь застосування – місто, де є готельний, транспортний і комплекс харчування.

Дана ІС надає можливість:

- переглядати фото місцевості, історію міста;
- переглядати карту місцевості;
- переглядати інформацію про погоду;
- переглядати інформацію про місця проживання;
- переглядати інформацію про транспорт;
- переглядати інформацію про місця харчування;
- надавати можливість бронювати послуги(проживання, транспорт, харчування);
- надавати можливість за реєструватися постачальникам послуг;
- надавати можливість відпочиваючим обмінюватись відгуками.

Рівень підготовки користувача може бути початковим.

Користувачеві- туристу потрібно мати елементарні навички користування інтернет-додатками, електронною поштою.

Користувач- адміністратор має вміти користуватись програмами створення звітів, прийому замовлень.

Користувачу не потрібно турбуватись про порядок завантаження програм. Тому що йому не потрібно завантажувати на свій комп'ютер програми.

Завдяки ІС, яка основана на моделі SaaS адміністратор курортного комплексу звільняється від рутинної процедури прийому замовлення. Йому не треба піклуватись за збереження бази даних. База даних замовлень його організації зберігається на хостингу системи. Багато клієнтів віддалено отримують дану інформаційну систему як сервіс. Замовники платять не за володіння ІС як такою, а за її використання.

Користувачу потрібно лише мати на своїй машині клієнт- web браузер.

ДОДАТОК Д -Програма і методика випробувань

Д.1 Об'єкт випробування

Назва інформаційної системи- «АРМ адміністратора Web системи інформаційний портал міста».

Область використання інформаційної системи — місто.

Д.2 Ціль випробувань

Ціль випробувань- перевірка відповідності характеристик інформаційної системи вимогам, поставленим у розділі 1- Формування вимог до інформаційної системи, розробка її концепції, технічне завдання.

Д.3 Вимоги до функціональних характеристик ІС

Дана ІС повинна надавати можливість віддалено:

- дізнатись інформацію про місця проживання, ціни, рівень комфорту;
- дізнатись інформацію про транспорт, ціни, розклад маршрутів, зміни;
- дізнатись інформацію про місця харчування, ціни, меню;
- zareєstrуватись постачальникам послуг;
- постачальникам послуг викласти інформацію про вільні місця;
- бронювати послуги (проживання, транспорт, харчування);
- відпочиваючим обмінюватись відгуками;
- дізнатись історію міста, переглянути зручну карту, фото, статті, новини місцевості;
- переглядати інформацію про погоду.

Д.4 Вимоги до програмної документації

До складу програмної документації, що пред'являється при випробуваннях входять:

- технічне завдання;
- посібник користувачу;
- програма і методика випробувань.

Д.5 Засоби і порядок випробувань

Засіб, на якому має бути проведено випробування- це комп'ютер з такими характеристиками:

процесор сімейства x-86, з більше 500 мільйонів інструкцій за секунду;
512 mB оперативної пам'яті;
100 mB вільної пам'яті на диску;
наявність клавіатури;
наявність монітора.

По кожному варіанту використання визначу, що саме повинні були отримати в результаті випробувань. Перевірю можливість дізнатись інформацію про місця проживання; про транспорт; про місця харчування; зареєструватись постачальникам послуг; постачальникам послуг викласти інформацію про вільні місця; бронювати послуги (проживання, транспорт, харчування); залишати відгуки ; дізнатись історію міста, переглянути зручну карту, фото, статті, новини місцевості; переглядати інформацію про погоду.

Д.6 Методи випробувань

Випробування проводяться в два етапи:

- 1) ввідний;
- 2) випробування.

Д.6.1 Перелік перевірок, що проводяться на 1-му етапі випробувань:

- 1) перевірка комплектності програмної документації;
- 2) перевірка комплектності складу технічних і програмних засобів;

Методики проведення викладено у пункті С.6.3.

Д.6.2 Перелік перевірок, що проводяться на 2-му етапі випробувань:

- а) відповідність технічних характеристик ;
- б) перевірку ступеня виконання вимог функціонального призначення програми.

Методики проведення викладено у пункті Д.6.4.

Д.6.4- Перший етап випробувань

Перевірку комплектності програмної документації проводить керівник дипломного проекту ст.викладач кафедри ІУСТ Беркунський Е.Ю., візуально перевіряючи відповідність програмної документації, приведеної в пункті Д.4- Вимоги до програмної документації.

