

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра комп'ютерної інженерії

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: «Розробка програмного забезпечення для довідково-інформаційної системи туристичного агенства»

Виконав: студент 5 курсу, групи ППЗ-51
спеціальності

121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

Агапов А.О.

(прізвище та ініціали)

Керівник Негоденко О.В. _____

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормконтроль _____

(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально-науковий інститут Телекомунікацій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки - 121 – Інженерія програмного забезпечення"

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

О.В. Негоденко

“ ” 2022 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Агапову Андрію Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка програмного забезпечення для довідково-інформаційної системи туристичного агенства»

Керівник роботи зав.кафедри ПЗ Негоденко О.В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “18” лютого 2022 року № .

2. Строк подання студентом роботи 03.06.2022.

3. Вхідні дані до роботи:

3.1. Положення побудови довідково-інформаційної системи

3.2. Методи побудови довідково-інформаційних систем

3.3. Існуючі довідково-інформаційні системи для туристичних агентств

3.4. Науково-технічна література

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

4.1. АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРАГЕНСТВА

4.2. РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРАГЕНСТВА

4.3. ЯКІСТЬ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.4. ВИСНОВКИ

5. Перелік графічного матеріалу

1. _____

2. _____

3. _____

4. _____

6. Дата видачі завдання 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	-	-
2	Дослідження інсуючих довідково-інформаційних систем для туристичних агентств	-	-
3	Проектування довідково-інформаційної системи	-	-
4	Підготовка розділу 1	-	До 04.04
5	Підготовка розділу 2	3 04.04	До 25.04
6	Підготовка розділу 3	3 25.04	До 06.05
7	Висновки та презентація	3 06.05	До 13.05
8	Перевірка роботи на антиплагіат + Передзахист	3 16.05	До 01.06
9	Захист роботи	3 02.06	До 26.06
10	Випуск	30.06	

Студент _____ Агапов А.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Негоденко О.В.
(підпис) (прізвище та ініціал)

РЕФЕРАТ

Дипломна робота виконана на 101 сторінці машинописного тексту, містить 44 рисунки, ___ таблиць, _ додатки та список використаних джерел з ___ найменувань.

Ключеві слова: Туризм, Туроператори, Туристична Агенція, Споживач, Клієнт, Web додаток

Об'єкт дослідження - підвищення ефективності та якості туристичних послуг для споживачів, зменшення часу обслуговування клієнтів туроператорами за допомогою програмного забезпечення довідково-інформаційної системи.

Предмет дослідження - інформаційне забезпечення туристичної діяльності.

Мета роботи – розробка програми довідково-інформаційної системи для туристичних агентств, заснованої на автоматизації процесу надання туристичних послуг споживачу туроператором у режиму real time.

Методи дослідження – метод теорії інформації, обробка та аналіз інформації.

У відповідності з поставленою метою для вирішення технічної задачі в роботі вирішено такі завдання:

- Аналіз доступних та зручних способів передачі інформації про тури від туристичних агенцій;
- Аналіз технічних засобів для розробки довідково-інформаційної системи та вибір оптимальних зручних рішень спрощення надання інформації про тури і рекомендації споживачу;
- На основі результатів аналізу досліджень розроблена довідково-інформаційна система.

Проведення розробленої довідково-інформаційної системи підвищить якість обслуговування споживачів у режимі реального часу та скоротить час обслуговування клієнтів туроператором (менеджером).

В роботі виконано аналіз існуючих застосунків для операційної системи Wsndows.

Розроблено логіку довідково-інформаційної системи та зручність користування для туроператорів (менеджерів) Туристичних Агенцій.

Галузь використання – завдяки довідково-інформаційною системи туроператор може за короткий час провести підбір тура на запит клієнта в режимі real time.

Результатом роботи є побудована довідково-інформаційна система для роботи туристичного агентства, а саме ситсеми, яка була написана мовою JavaScript та у якості бази даних використана MongoDB.

Розроблене програмне забезпечення може функціонувати під управлінням операційних систем Linux або Windows 10 та вище.

ЗМІСТ

ВСТУП	6
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРАГЕНСТВА.....	8
1.1 Аналіз організаційної структури та інформаційного забезпечення туристичної діяльності	8
1.2 Аналіз сучасних програмних продуктів автоматизації діяльності туристичного агенства.....	13
1.3 Постановка задачі з розробки програмного забезпечення довідково-інформаційної системи турагенства	20
2 РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРАГЕНСТВА.....	22
2.1 Ескізний проект програмного забезпечення	22
2.1.1 Розробка діаграм використання програмного забезпечення інформаційно-пошукової системи	22
2.1.2 Проектування інтерфейсу програмного забезпечення довідково-інформаційно системи.....	30
2.2 Технічний проект програмного забезпечення.....	33
2.2.1 Проектування моделі даних програмного забезпечення	33
2.2.2 Статична модель програмного забезпечення довідково-інформаційно системи	37
2.2.3 Динамічна модель програмного забезпечення довідково-інформаційно системи	39
2.2.4 Логічні моделі структур даних програмного забезпечення	43
2.3 Робочий проект програмного забезпечення	46
2.3.1 Вибір засобів розв'язання поставленої задачі	46
2.3.2 Фізична модель даних програмного забезпечення довідково-інформаційно системи	47
2.3.3 Тестування класів програмного забезпечення довідково-інформаційно системи.....	51
2.3.4 Діаграма компонентів програмного забезпечення довідково-інформаційно системи	58
2.3.5 Кодування.....	59
2.3.6 Випробування програмного забезпечення довідково-інформаційної системи	63
3 ЯКІСТЬ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	51
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	70
ДОДАТКИ.....	72
Додаток А.....	72
Додаток Б.....	79
Додаток В.....	83
Додаток Г	87

ВСТУП

Актуальність дослідження. Поява туристичних агентств, спрощення системи віз зробило туризм самостійною галуззю бізнесу. По даним статистики у 2019 році туризм перевищив показники на 23% у порівнянні з 2018 роком, кількість туристів з України склало 57,4 млн чоловік. Через необхідність підбору турів у режимі реального часу, для вирішення підвищення ефективності та якості туристичних послуг споживачам виникає потреба в використанні інформаційних технологій. Кожному туристичному агентству потрібен дешевий, простий та зручний спосіб передачі інформації про власні пропозиції та про саму туристичну агенцію. Необхідно розширити поїздки клієнтів і збільшити продажі, а також просувати свою продукцію на ринку туристичних послуг. У зв'язку з цим важливою є розробка програмного забезпечення для автоматизованих онлайн-систем продажу туристичних продуктів та інформації, яка надається кінцевим користувачам за допомогою веб-сайтів туристичних агентств, у тому числі розробка науково обґрунтованих інформаційних моделей та баз даних з даної тематики.

Мета роботи. Виходячи з актуальності дослідження метою даної роботи є розробка програмної довідково-інформаційної системи для туристичних агентств для автоматизації процесу надання туристичних послуг споживачу у режимі реального часу туроператором (менеджером), завдяки чому підвищиться ефективність та якість туристичних послуг для споживачів та зменшиться час обслуговування клієнтів. Програмне забезпечення буде реалізовано як веб-додаток. Для виконання поставленої мети слід виконати наступні завдання:

- провести аналіз предметної області та сформулювати завдання на розробку програмного забезпечення для довідково-інформаційних систем туристичного агентства;
- Розробити вимоги до програмного забезпечення, у вигляді документа «Технічне завдання», проаналізувати технічні засоби, що використовуються для довідково-інформаційних систем;

- виконувати всі етапи проектування програмного забезпечення (ескізи, прийоми, роботи) та його впровадження;
- при необхідності провести тестування програмного забезпечення та відповідну оптимізацію;
- розробити необхідну програмну документацію;
- Розглянути основні питання охорони праці та запропонувати заходи щодо забезпечення належних умов праці при використанні персональних комп'ютерів.

Об'єкт дослідження - підвищення ефективності та якості туристичних послуг для споживачів, зменшення часу обслуговування клієнтів туроператорами за допомогою програмного забезпечення довідково-інформаційної системи.

Предмет дослідження - інформаційне забезпечення туристичної діяльності.

Мета роботи – розробка програмної довідково-інформаційної системи для туристичних агентств, заснованої на автоматизації процесу надання туристичних послуг споживачу туроператором у режимі реального часу.

Методи дослідження – метод теорії інформації, обробка та аналіз інформації.

Практичне значення одержаних результатів. Додатково-інформаційна система допоможе туроператорам (менеджерам) у режимі реального часу, за короткий проміжок часу зробити підбір туру та надати рекомендації на запит клієнта, що підвищить ефективність та якість обслуговування споживача у галузі туризму.

Особистий внесок. Вдосконалено довідково-інформаційну систему для підвищення якості обслуговування споживача в галузі туризму.

Результати роботи. Матеріали дипломної роботи можуть сприяти підвищенню ефективності обслуговування та обробки більшої кількості заявок туристів у режимі реального часу туроператорами Туристичних Агенцій та сприяти росту валового доходу Туристичного Агенства.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ З РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРАГЕНСТВА

1.1 Аналіз організаційної структури та інформаційного забезпечення туристичної діяльності

Туризм включає організацію поїздок та надання необхідних послуг для задоволення різноманітних потреб туристів через систему ринкових відносин (тобто шляхом купівлі-продажу, з метою отримання прибутку чи іншої вигоди).

Основним органом туристичного ринку є юридичні та фізичні особи як виробники та покупці туристичної продукції.

Суб'єктами, які займаються та пропонують туристичну діяльність, є:

- Туроператори (Туристичні оператори) - створені відповідно до законодавства України, виключною діяльністю яких є організація та надання, створення туристичних продуктів, продаж і надання туристичних послуг, а також надання посередницької діяльності та супутніх послуг для певних видів діяльності. Мають мати «Ліцензію» на туроператорську діяльність;
- Туристичне агентство (Турагентство) - Суб'єкт, який реалізує туристичні продукти туроператорів та туристичні послуги інших суб'єктів туристичної діяльності, реалізує конкретні та супутні послуги та має належним чином ліцензію на діяльність туристичного агентства.

Туроператори (Підприємства, які спеціалізуються на плануванні, організації турів (подорожей) та продажу турів за допомогою турагентств (провідних організацій, утворених туроператорами для здійснення туроператорів. Більшість прибутків отримують від комісійних, отриманих від турів)) (Індивід. або групова подорож), що складається з комплексних послуг (транспортування, проживання в готелі, екскурсії, трансфери, харчування), являє собою пакет окремих складових поїздки чи подорожі, включаючи проїзд, проживання, трансфери, оплату екскурсійних визначних пам'яток тощо.

Багато людей продають свої послуги через брошури, розміщені на спеціальних прилавках у туристичних агентствах. Деякі люди здійснюють прямі продажі, рекламуючи на сторінках газет і журналів, використовуючи пряму розсилку та продаючи тури таким метод називається «зі сторінки».

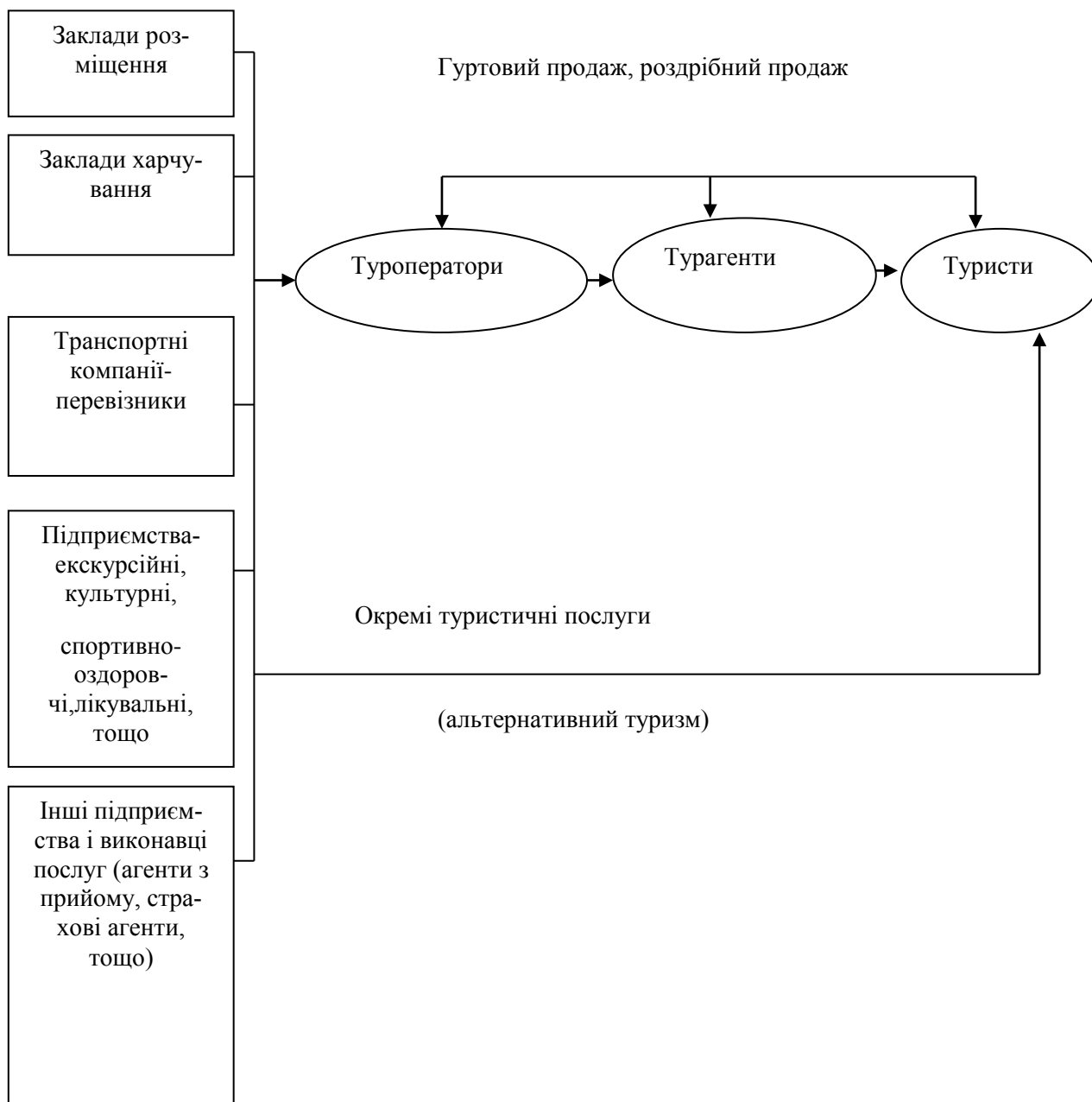


Рисунок 1.1 – Схема формування і продажу комплексної туристичної послуги – тура

Комп'ютерні інформаційно-пошукові системи призначені для задоволення потреб туристичних агентств і їх пріоритети визначаються таким чином:

1) Забезпечити постачальникам туристичних послуг (авіатранспорт, оренда автомобілів, готелі, туристичні агентства, залізниці тощо) надійну та ефективну глобальну мережу продажів, надаючи їм доступ до найбільшої у світі клієнтської бази та комерційних авіаційних бюро, що належать туристичним агентствам. ;

2) Надати абонентам (турагентам) потужні інструменти управління та маркетингу, щоб зробити їх конкурентоспроможними та прибутковими, доступ до місцевих та міжнародних постачальників, управління бізнесом та фінансовий облік компанії;

3) надати авіакомпаніям можливість використовувати систему у своїх комерційних офісах, щоб авіакомпанії та турагенти могли працювати через ту саму систему;

4) Надавати глобальні послуги через технічні зв'язки з іншими системами.

Процес діджиталізації у сфері надання туристичних послуг є необхідним кроком у подальшій діяльності та конкурентоспроможності, оскільки відповідність до вимог клієнта, надійність, ефективність, можливість швидкої обробки та передачі інформації створюють значну перевагу в цій сфері. Для турагентств найбільш нагальним стає питання повної автоматизації робочого процесу від бронювання до розрахунку з клієнтами та готелями. Основним видом діяльності туристичних фірм і туроператорів є надання інформації майбутнім клієнтам. Основна ринкова роль таких організацій полягає в тому, щоб відрізнити постачальників послуг від клієнтів (туристів), іноді за часом і географічно. Лише 48% клієнтів знають, куди вони хочуть подорожувати, 35% мають туманне уявлення про це, а 17% взагалі не мають уявлення.

Туристичні агенти надають інформацію потенційним покупцям щодо туристичних районів, розкладу транспорту, варіантів проживання, поточні ціни, можуть допомогти скласти приблизну вартість поїздки. Безпосередній контакт з клієнтами, бесіди та консультації, а також підбір і перевірка рекомендацій займають багато часу, займаючи половину часу.

Другою функцією туристичного агентства є продаж туристичних послуг, які надають перевізники, заклади громадського харчування та інші туристичні орга-

нізації. Туристичні агентства часто виступають посередниками, продаючи квитки на різні види транспорту, бронюючи готелі та інші бази. Крім того, вони можуть орендувати автомобілі, бронювати тури, оформляти проїзні документи, страхувати туристів, обмінювати валюту, видавати дисконтні картки, продавати туристичну літературу та сувеніри тощо. Крім індивідуальних туристичних послуг, туристичні агенції продають також повні комплексні пакети – тури.

Комп'ютерні інформаційно-пошукові системи дозволяють туристичним агентствам створювати спеціальні туристичні пакети (комплекси подорожей), які охоплюють основні елементи відпочинку: проїзд, проживання, екскурсії, розваги та, звісно, спеціальне обладнання для заходу все це входить до цих планів.

Туристичний пакет – це комплексне рішення обов'язкових (базових) послуг, які надаються за індивідуальним або груповим планом під час подорожжі. Вони мають ряд характеристик і широко продаються.

Звичайно, всі операції з бронювання здійснюються турфірмою через контакти з певними структурами, задіяними на місцях.

На рисунку 1.2 показана структура туристичного пакету, який об'єднав мінімальний, обов'язковий, набір послуг.

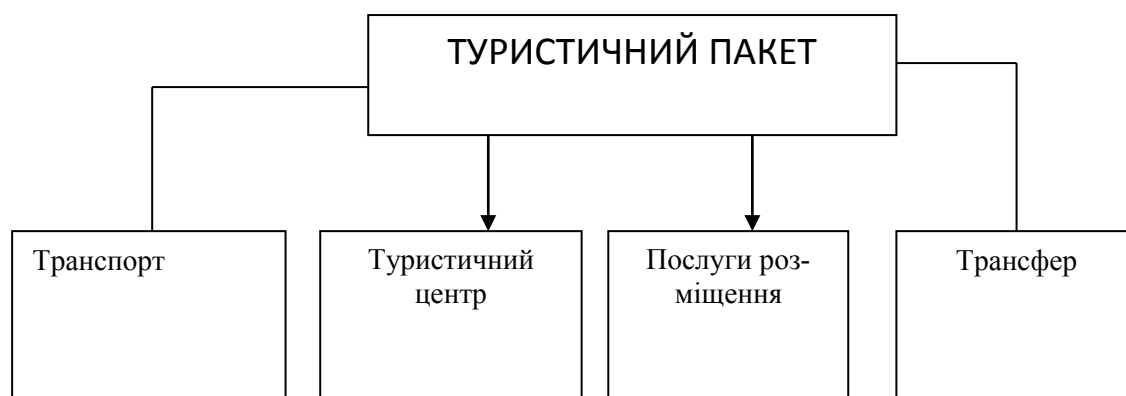


Рисунок 1.2 – Структура туристичного пакету

Туристичний продукт — є комплексом надання матеріальних, у вигляді споживчих, і нематеріальних, у вигляді послуг, які необхідні для задоволення потреб подорожуючих, під час їхньої мандрівки.

Туристичний центр – місцевість, де локалізовані туристичні послуги. Являє собою локацію перебування туристів, яка може містити в собі природні, культурно-історичні, екологічні, етнічні, соціально-демографічні, інфраструктурні рекреаційні можливості.

Транспорт – засіб для пересування. Одна з функцій якого можливість транспортувати подорожуючого до готелю, туристичного центру, аеропорту.

Послуги з розміщення – це конкретні готелі чи інші засоби розміщення, які надаються туристам у туристичних центрах протягом періоду подорожі.

Загальна структура туристичних продуктів є частиною туристичного пакету, як показано на рисунку 1.3.

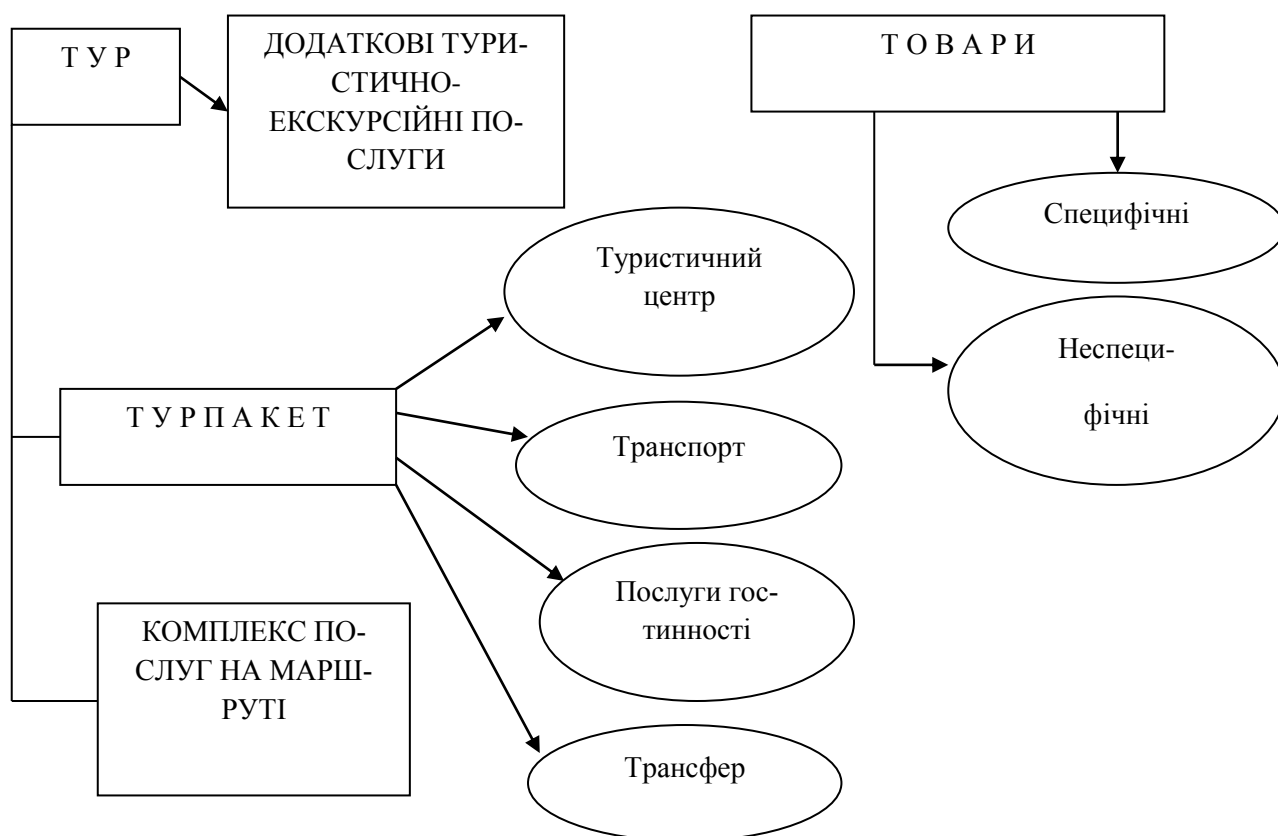


Рисунок 1.3 – Інтегральна структура туристичного продукту

Обладнання та засоби зв'язку залежать від типу підключення системи, обраного турфірмою:

- підключення за допомогою VPN на сервер з програмою;

- встановлення додатку на мобільний пристрій з можливістю авторизації через номер телефону

- сервер телефонії для обробки дзвінків;

- підключення до LAN мережі для роботи через сервер

- підключення мережі інтернет для роботи з WEB інтерфейсом додатку.

Функція агентства для клієнтів, які здійснюють бронювання самообслуговування через Інтернет, полягає в тому, щоб перевірити правильність бронювання, оптимізувати обраний маршрут, оформити та доставити квиток.

Отже, комп'ютерна інформаційно-пошукова система туристичних послуг виконує такі функції:

- дозволяє здійснювати пошук оптимальних цінових варіантів туристичних послуг;

- дозволяє працювати з різними тарифами, включаючи конфіденційність;

- дозволити туристичним агентствам використовувати автоматизовані процедури звітності;

- дозволяє збирати різноманітні статистичні данні;

- забезпечити користувачеві доступність розгалуженої оперативної інформації в довідковій системі;

- дозволяє здійснювати контроль над всіма операціями між клієнтами та постачальниками для взаєморозрахунків;

- Дозволяє створити систему бронювання туристичних послуг для корпоративних та приватних клієнтів на сайті турагентства.

В результаті відвідувачі сайту мають можливість самостійно вибирати маршрути, поїздки, перельоти та готелі та подальші резервування. У частині системи, яка відповідає за бронювання він ,ознайомлений з правилами бронювання, заповнює необхідні інформаційні поля та відправляє замовлення в систему за допомогою електронної пошти з використанням SMTP протоколу.

Після вищевказаних дій автоматизована система оброблює запит на замовлення і надсилає попереднє підтвердження про бронювання, з реквізитами на внесення передоплати у вигляді 35% від вартості туру, тим часом клієнт заноситься в

"лист очікування". Отримавши підтвердження з відповідними гарантіями: передоплати з боку клієнта та гарантійного листа з боку турагента, який свідчить про отримання передоплати. Клієнт отримує від агентства остаточне підтвердження броні, а заявка в системі отримує статус підтверджено.

1.2 Аналіз сучасних програмних продуктів автоматизації діяльності туристичного агентства

Зазвичай автоматизація роботи турфірм здійснюється на основі стандартних офісних процедур, а компанія створює власний сайт в Інтернеті. Найпростішою можливістю Інтернету в туристичній індустрії є поширення інформації про компанії та їх послуги на різних дошках оголошень і форумах. Навіть веб-сайт, який найчастіше використовується як вступна брошура, може дозволити компанії залучити до 20% своїх клієнтів.

У залежності від кількості та типу туристичних агентств постає питання "Чи потрібно використовувати спеціальне програмне забезпечення для автоматизації всієї офісної роботи?", оскільки програмне забезпечення дуже різниться для потреб туристичної агенції та потреб туристичного агента. Тому для компаній, які впроваджують невеликий обсяг послуг, достатньо можливостей Microsoft Office (Excel, Word, Outlook тощо).

Типовий програмний комплекс - є інструментом підвищення ефективності туристичної компанії, що відповідає вимогам глобального управління подорожами. Вони універсальні: незалежно від характеру туристичної діяльності, ними можуть користуватися як невеликі туристичні фірми, так і потужні турфірми. Програма має інтуїтивно зрозумілий інтерфейс, не вимагає спеціальної комп'ютерної підготовки, проста у використанні запобігає помилковій роботі користувачів.

Типовий програмний комплекс автоматизує діловодство туристичного підприємства, а саме:

- Надання опису окремих пакетних послуг, їх зміст із застосуванням алгоритму ціноутворення, моніторингу квот, підтримання договірних та агентських договорів з підрядчиками;

-Демонстрація послуг та пакетів які продаються, підготовка реклами та оцінка її ефективності.

За допомогою типових програмних пакетів змінилася технологія взаємодії турагентств і туристичних операторів: встановлення зв'язків з віддаленими робочими місцями (відправлення листів, доступ до єдиної бази пакетних пропозицій).

У кожен типовий пакет можна встановити такі шаблони:

1. Підготовка туристичних послуг на продаж: створення інформаційних каталогів (таблиці у базі даних з країнами, готелями, авіакомпаніями), створення, упорядкування, встановлення та редагування прайсу, дисконтні програми, календарного циклу, створення описових матеріалів;

2. Створення туризму (формування туристичних продуктів індивідуального та групового споживача, туристичний розрахунок), розробка шаблону стандартного туризму;

3. Фінансові блоки та бухгалтерський облік: аналіз компаній (список найбільш активних турагентств, графіки завантаження готелів, рейтинги популярності, напрямки), взаємовідносини з партнерами та розрахунки, бухгалтерський облік;

4. Розподіл прав і ролей користувачів (дозволи на проведення операцій в системі).

Розглянемо найбільш популярні у туристичній галузі програмні комплекси

Програмний комплекс „Мастер-Тур” Призначений для максимальної автоматизації офісної роботи туристичних компаній, акцентуючись на технології бек-офіс, які:

- Формувати пакети турпродуктів, розраховувати очікуваний прибуток та комісійну суму подорожжі, завершувати групову та індивідуальну співпрацю, прискорювати та спростити процес формування туристів

- Роздруковувати повний комплект необхідних документів (путевих листів, путівок, анкет посольства, фінансових звіти тощо) Помилки у співпраці з клієнтами;

- квота на будь-які туристичні послуги, запобігти перепродажу;

- Стежити за ходом оплати подорожей та контролювати поїздки; - Оцінити поточне фінансове становище компанії.

Основний зміст додатку – «подорожі» та «сервіс».

Формально «Тур» — це пакет послуг, що продається клієнтам. Зазвичай - це переліт, трансфери, проживання, декілька екскурсій, візи, страховка.

Комплекс містить в собі два модулі: «туристичний» та «фінансовий», які представлені сукупністю автоматизованих робочих місць.

Туристичний модуль “Мастер-Тур” включає такі функції:

- формування пакету туристичного продукту;
- продаж туристичного пакету;
- робота з підрядниками;
- робота в мережі Internet;
- адміністратор.

Створення туристичного продукту проходить в чотирьох режимах:

1. Режим “Довідник”: дає можливість підготувати туристичні послуги на продаж;
2. Режим “Квоти на послуги”: дає можливість поставити знижку на пропозиції, як приклад на деякі авіа перельоти. Визначаються та редагуються ціни;
3. Режим “Дата та завантаження турів”: дає можливість завантажити графіки поїздок, сезонних періодів;
4. Режим “Конструктор турів”: дає можливість сконструювати типові туристичні пакети, з яких складається тур на продаж.

Фінансовий модуль «Мастер-Тур» містить в собі наступні компоненти:

- Модуль «Каса»;
- Модуль «Розрахунки з агентами»;
- Модуль «Директор»;
- Модуль «Розрахунки з партнерами»;
- Модуль «Фінансовий стан»;
- Модуль «Затрати по заїзду».

Програмний комплекс “Мастер-Web” турфірми, які співпрацюють з «Мастер Тур», мають можливість відкривати інтернет-магазин (WEB-сервіс) для продажу своїх турів та квитків у режимі реального часу. Оператору достатньо визначити тури та сегменти для продажу, призначивши права доступу різним категоріям споживачів і клієнтів. Вся інформація взята з БД (база даних) «Майстер Тур». Клієнти, підключені до сервера мережі оператора, можуть в режимі real-time:

- переглядати інформацію про тури, квитки, спеціальні пропозиції, планувати подорожі як групові так і індивідуальні;
- Витрати на обчислення;
- Замовити тур.

Туристичні агентства використовують програмне забезпечення для бронювання готелів, квитків та турів у режимі реального часу через Інтернет, так і бронювання в офісі туристичного агентства.

Туристична інформаційна система „ТІС” – Це інтернет-система бронювання туристичних послуг, основою якої є база даних туристичних послуг різних операторів. Загальний алгоритм використання ТІС:

1. Користувач описує свою пропозицію (турпакет), додають фотографії, анімацію, відео та аудіо кліпи. Дана функція може впроваджуватись через Інтернет API або спеціальну програму «ТІС Навігатор».

2. Публікація опису туристичних пакетів ТІС та визначення квот на обслуговування.

3. Турпакети аналізуються агентом. Досліджено зразок встановлених структур туристичних пакетів (готелі, сезони, тарифи, квоти).

4. Оформіть замовлення та розрахуйте вартість виходячи з послуги.

Усі замовлення зберігаються на серверах TIS. Основні характеристики TIS:

- публікації;
- Оперативна інформація щодо змін в наявних турах;
- пошук потрібного турпакета в каталозі;
- створення та розрахунок ціни заявки на туриста або групу туристів;

- робота в режимі real-time з квотами та станом заявок, з архівами вхідних та вихідних заявок.

Програма “ТІС Навігатор” Призначений для автоматизації каталогів подорожей в офісі та в Інтернеті. Завдяки ньому у агентів є можливість з максимальною швидкістю та якістю провести аналіз складу й умов турпакетів від різних операторів, класифікувати та профільтрувати за різними критеріями, шукати прийнятні позиції, проводити процедури реєстрації заявок клієнтів, додавання прайс-листів, створення і редакція каталогів, розміщення рекламних матеріалів.

Характеристики «ТІС Навігатор»:

- Адаптивна структура створення каталогів;
- Можливість додавання нотаток до пакетів послуг;
- розрахунок ліцензії;
- друковані рекламні описи та ціни;
- опис турпакета;
- інформація про готельні послуги;
- Класифікація та групування туристичних пакетів;
- реєстрація заявок на отримання ліцензії;
- Експорт у файли Excel, .TIS, "ТІС Навігатор".

Основні особливості роботи:

1. Усі файли зберігаються в спеціальних окремих папках - Коментарі: папка туристичних пакетів, папка клієнтської програми, папка довідника країн, готелі, примітки телефону.

2. В основному огляді «Туристичний пакет» опис пакета зберігається. У вікні огляду можна виконувати різні операції: читати деталі, сортувати, вибирати пакети за різними атрибутами, редагувати, друкувати описи пакетів, експортувати в Excel, вводити нові пакети.

3. Інформація про турпакет надається у вигляді прайс-листа (готель, сезон, терміни, тарифи). Інформація про декілька готелів буде надана у вигляді зведених цін, що одночасно зрозуміло та економить час на навігацію по каталогу.

4. Надайте інформацію про готель (опис, обслуговування, обслуговування) та загальну туристичну інформацію (текстовий опис подорожі, умови реєстрації, інформацію про транспорт) під час перевірки ціни.

5. Розрахунок вартості туру на групу дуже простий, і вартість туру буде автоматично змінюватися відповідно до обраного конкретного готелю, дати та сезону.

6. Реєстрація заявок та ліцензій здійснюється за інформацією про обрану туристичну групу та готель, а довідник клієнтів автоматично поповнюється при реєстрації нових клієнтів.

7. Заповнена заявка зберігається в папці перегляду заявки. При необхідності експортуйте документ до формату .xls, .xlsx та Excelподібних.

Часто використовуваний додаток **“TurWin”** від компанії **“Арісофт”**. Програму можна використовувати для запуску на локальному комп'ютері та розгортання на сервері з підключенням до локальної мережі. Використання програми розпочинається з створення або заповнення наступних таблиць в базі даних:

- таблиця "Маршрутів":
- таблиця "Готелів":
- таблиця "Рейсів":
- таблиця "Валют":
- таблиця даних, які необхідні для самостійного формування та розрахунку поїздки чи туристичного маршруту.

Спрощення роботи з покупцями відбувається завдяки заповненню стандартизованих шаблонів, які містять наступні поняття: тури, побажання, клієнти, прибуття тощо. Заповнивши такий шаблон, виходить пропозиція з максимумальною вигодою та друкована анкета замовника. Інформація про покупця автоматично відображається в таблицях страхування, візи, бронювання у базі даних. У програмі можна зручно вести статистику конвертуючі звіти в Excel, Word. В програмі є багатифункціональний фінансовий модуль, який дозволяє відстежувати внесення сплати за акти виконаних робіт і вести підприємчу звітність. Доступ до конфіденційної інформації в програмі обмежений.

Слід зазначити, що всі перелічені програмні продукти є комерційними та досить ресурсоемними. Крім того мають закритий вихідний код, що не дозволяє їх адаптувати під особливості роботи компанії.

Таким чином актуальним є розробка довідково-інформаційної системи, яка б повністю задовольняла всім вимогам туристичного агентства.

1.3 Постановка задачі з розробки програмного забезпечення довідково-інформаційної системи турагентства

Аналіз показує, що оскільки існуючі програмні продукти не можуть повністю задовольнити всі вимоги туристичних агентств, необхідно та доцільно розробити довідкову інформаційну систему туристичних послуг. Підвищення ефективності та якості туристичних послуг споживачам шляхом використання інформаційних технологій. Створений веб-сайт буде зберігати та постійно оновлювати інформацію та створювати для цього відповідну базу даних, що дозволить відвідувачам сайту, потенційним клієнтам туристичних агенцій, мати доступ до найсвіжішої та найточнішої інформації в будь-який момент, не виходячи з дому. Взаємодія користувача з програмою відбувається через інтерфейс, що дозволяє переглядати, вибирати та замовляти певні параметри туристичних послуг.

Взаємодія користувача (адміністратора) з програмою відбувається через інтерфейс, який дозволяє створювати, редагувати, видаляти та переглядати повний перелік туристичних послуг, включаючи замовлення. Програмне забезпечення має виконувати такі функції:

- 1) Зареєструватися в системі;
- 2) Ідентифікація;
- 3) Відображати, вводити та редагувати доступну інформацію про тур та вартість проїзду;
- 4) Відображення, введення та виправлення інформації про клієнта;
- 5) Відображати та редагувати інформацію про бронювання турів, надавати замовнику копію договору;

б) Обробка замовлення

Для успішної і коректної роботи програмного забезпечення необхідний персональний комп'ютер з наступною сукупністю апаратних і програмних засобів:

Мінімальні системні вимоги

- процесор Intel Pentium G4400, AMD ATHLON 220 GE;
- 8 GB оперативної пам'яті;
- Мережевий адаптер з підключенням 1 Gbit;
- 120 GB SSD;
- ОС Microsoft Windows 10/ Server 2016 або вище, ОС Linux.

Рекомендовані системні вимоги

- процесор Intel Core I3 6100, AMD Ryzen 1600;
- 16 GB оперативної пам'яті з підтримкою ECC;
- Мережевий адаптер з підключенням 2,5 Gbit;
- 240 GB SSD;
- ОС Microsoft Windows 10/ Server 2016 або вище, ОС Linux.

Технічне завдання на розробку програмного забезпечення довідково-інформаційної системи туристичного агенства наведено в додатку А.

2 РОЗРОБКА ПРОЕКТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДОВІДКОВО-ІНФОРМАЦІЙНОЇ СИСТЕМИ ТУРАГЕНСТВА

2.1 Ескізний проєкт програмного забезпечення

У підрозділі «ескізний проєкт» представляються результати зовнішнього проектування програмного забезпечення. Розробляються загальна технологія розв’язання задачі з використанням обчислювальної техніки і необхідна для цього інформаційна модель.

2.1.1 Розробка діаграм використання програмного забезпечення інформаційно-пошукової системи

Базові вимоги до програмного продукту, які описані у ТЗ (додаток А), представлені у вигляді діаграми варіантів використання рисунок 2.1.

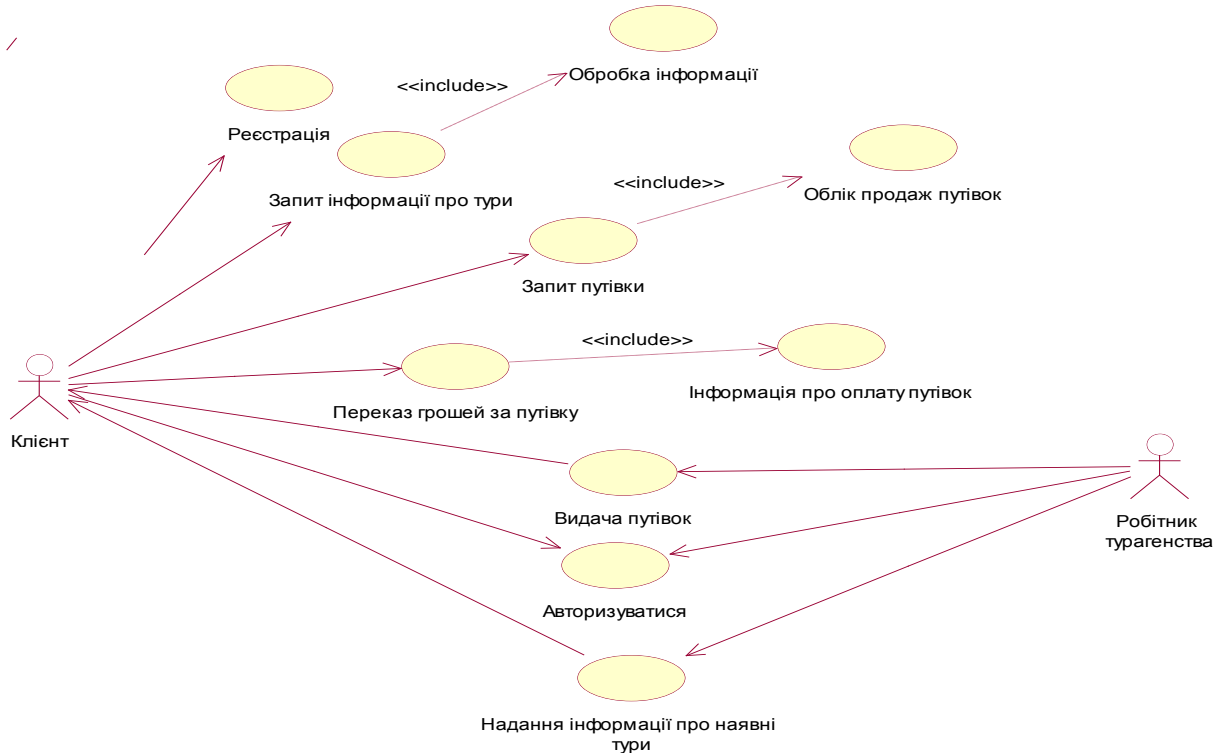


Рисунок 2.1 – Діаграма варіантів використання програмного забезпечення довідково-інформаційної системи туристичного агенства

Для більш зрозумілого сприйняття діаграми варіантів використання наведеної на рисунку 2.1 необхідно привести специфікації кожного варіанту використання.

Специфікація варіанта використання "Аутентифікація":

- a) Найменування варіанту використання "Аутентифікація".
- b) Призначення: для надання певному користувачу прав для замовлення турів, перегляду інформації про тури та інше.
- c) Учасники: активний суб'єкт "Клієнт".
- d) Попередня дія: будь яка дія в системі.
- e) Стартова дія: ініціюється активним "Клієнтом".
- f) Потoki подій: в будь-який момент роботи програми "Touristservices", на лівій панелі перед покупцем знаходиться вікно авторизації. Він повинен ввести свій логін і пароль, та підтвердити початок процесу аутентифікації. Клієнт також може не вводити данні, а продовжувати роботу з програмою до необхідності аутентифікації.
- g) Результати: При вдалому завершенні авторизації покупця, йому надається можливість замовляти тур, переглядати інформацію про тури. При невдалій авторизації клієнту повідомляється про помилку, з детальним її описом.
- h) Умови після закінчення використання: надання відповідних прав або повідомлення про помилку.
- i) Особливості: визначено декілька типів помилок - невірний логін, невірний пароль, не задані вхідні дані.

Діаграма діяльності для варіанта використання "Аутентифікація": представлена на рисунку 2.2.

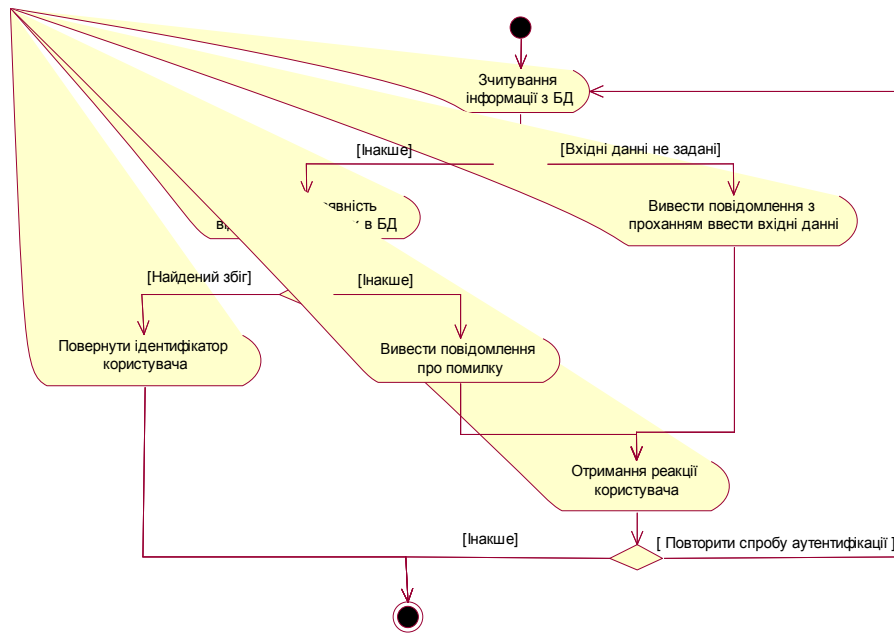


Рисунок 2.2 – Діаграма діяльності для варіанта використання "Аутентифікація"

Специфікація варіанта використання "Реєстрація клієнта":

- a) Найменування варіанта використання "Реєстрація клієнта".
- b) Призначення: створення нового клієнта.
- c) Учасники: активний суб'єкт "Клієнт".
- d) Попередня дія: будь-яка дія в системі.
- e) Стартова дія: ініціюється активним суб'єктом "Клієнт".
- f) Потоки подій: в будь-який момент роботи програми "Touristservices", не аутентифікований клієнт може перейти по посиланню, розташованому в лівій частині програми в блоці аутентифікації на сторінку реєстрації клієнта, заповнити форму та зареєструвати нового покупця.
 - g) Результати: створення нового покупця.
 - h) Умови після закінчення використання: здійснення аутентифікації щойно створеного покупця.
 - i) Особливості: аутентифікований покупець не має змоги зареєструвати нового покупця.

Діаграма діяльності для варіанта використання "Реєстрація клієнта" представлена на рисунку 2.3.

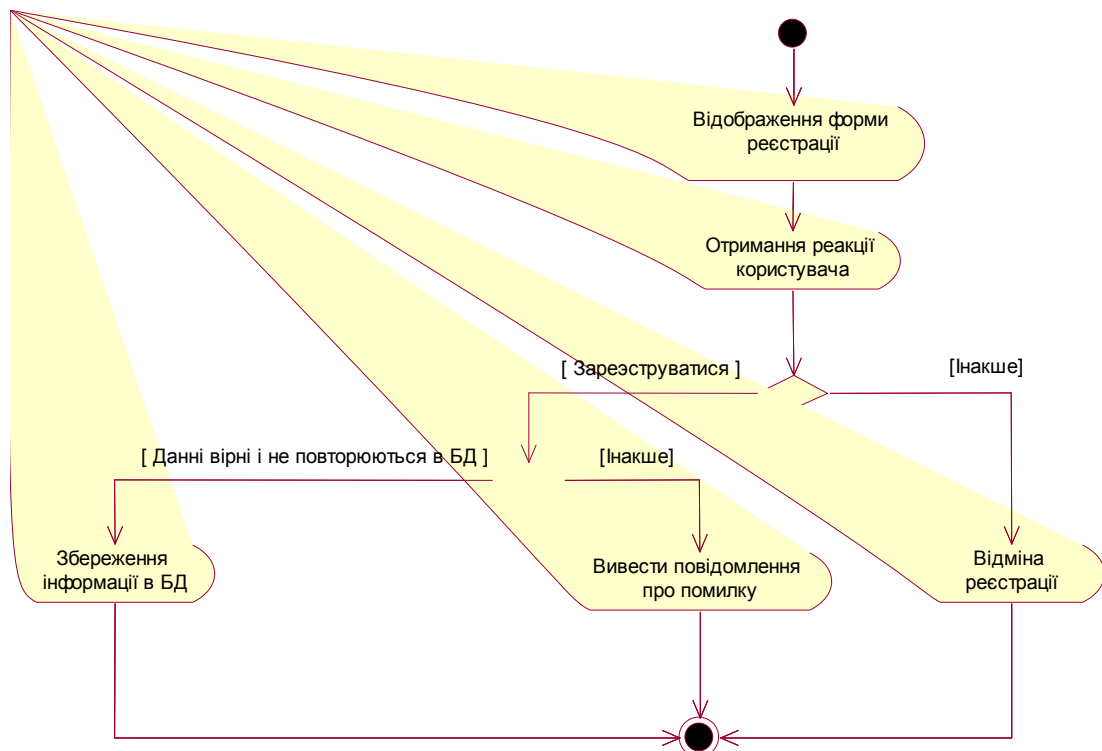


Рисунок 2.3 – Діаграма діяльності для варіанта використання "Реєстрація клієнта"

Специфікація варіанта використання "Запит інформації про тури":

- a) Найменування варіанта використання "Запит інформації про тури".
- b) Призначення: для пошуку та відображення списку знайдених турів.
- c) Учасники: активний суб'єкт "Клієнт".
- d) Попередня дія: будь-яка дія в системі.
- e) Стартова дія: ініціюється активним суб'єктом "Клієнта".
- f) Потоки подій: в будь-який момент роботи програми "Touristservices", клієнт має змогу здійснювати пошук турів, заповнивши форму певними параметрами та отримати результат пошуку, у вигляді, списку турів, або повідомлення про відсутність результату пошуку.
 - g) Результати: клієнт отримує на сторінці результат пошуку (списком турів).
 - h) Умови після закінчення використання: відсутні.
 - i) Особливості: відсутні.

Діаграма діяльності для варіанта використання "Запит інформації про тури" представлена на рисунку 2.4.

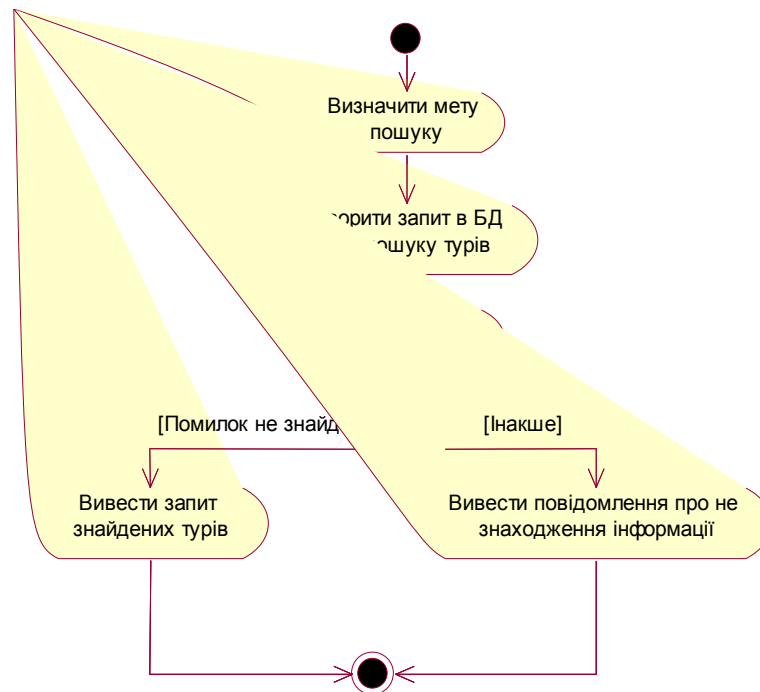


Рисунок 2.4 – Діаграма діяльності для варіанта використання "Запит інформації про тури"

Специфікація варіанта використання "Запит путівки":

- a) Найменування варіанта використання "Запит путівки".
- b) Призначення: для замовлення обраного туру.
- c) Учасники: активний суб'єкт "Клієнт".
- d) Попередня дія: будь-яка дія в системі.
- e) Стартова дія: ініціюється активним суб'єктом "Клієнтом".
- f) Потоки подій: користувач, знаходячись на сторінці "Інформація про тури", має змогу, замовити тур, натиснувши на кнопку "Оформить заявку", після чого буде перенаправлений на сторінку замовлення, заповнену приватними даними покупця та натиснути кнопку "Відправити заявку", для підтвердження замовлення.
- g) Результати: створення нового замовлення в БД.
- h) Умови після закінчення використання: відсутні.
- i) Особливості: відсутні.

Діаграма діяльності для варіанта використання "Запит путівки" наведено на рисунку 2.5.

Специфікація варіанта використання "Переказ коштів":

- a) Найменування варіанта використання "Переказ коштів".
- b) Призначення: для оплати обраного туру.
- c) Учасники: активний суб'єкт "Клієнт".
- d) Попередня дія: "Запит путівки".
- e) Стартова дія: ініціюється активним суб'єктом "Клієнтом".
- f) Потоки подій: користувач оформляє форму переказу, якщо інформація надійшла про переказ коштів, то клієнт отримує форму видачі путівки.
- g) Результати: отримати форму видачі путівки.
- h) Умови після закінчення використання: відсутні.
- i) Особливості: відсутні.

Діаграма діяльності для варіанта використання "Переказ коштів" представлена на рисунку 2.6.

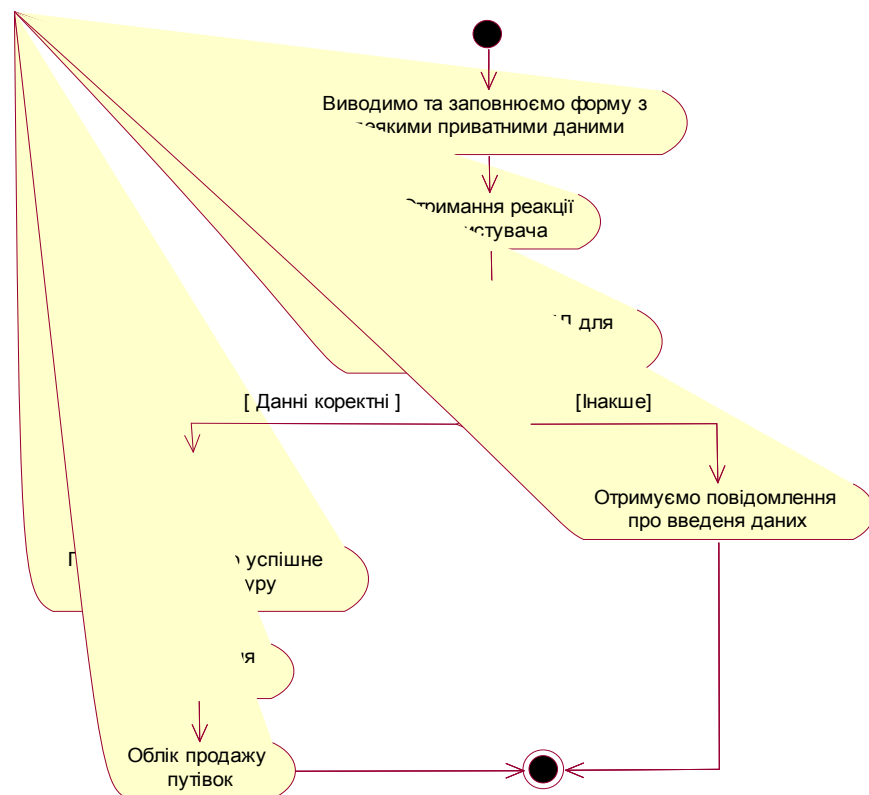


Рисунок 2.5 – Діаграма діяльності для варіанта використання "Запит путівки"

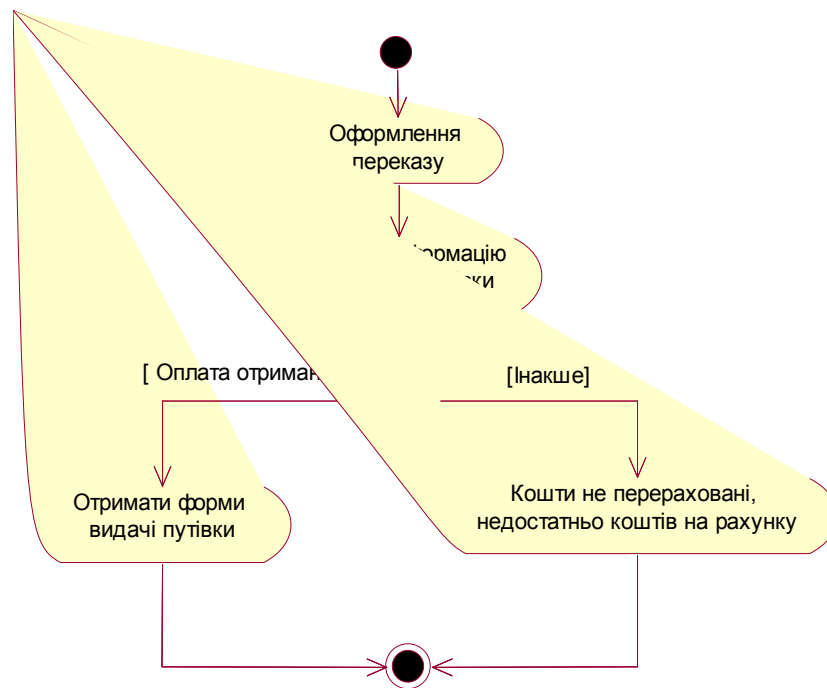


Рисунок 2.6 – Діаграма діяльності для варіанта використання "Переказ коштів"

Специфікація варіанта використання "Видача путівки":

- a) Найменування варіанта використання "Видача путівки".
- b) Призначення: для отримання документів на путівку.
- c) Учасники: активний суб'єкт "Робітник турагенства".
- d) Попередня дія: будь-які дії в системі.
- e) Стартова дія: ініціюється активним суб'єктом "Робітник турагенства".
- f) Потіки подій: користувач відправляє документи на електрону пошту, або залишаються в офісі.
- g) Результати: відправка документів.
- h) Умови після закінчення використання: відсутні.
- i) Особливості: відсутні.

Діаграма діяльності для варіанта використання "Видача путівки" на рисунку 2.7.

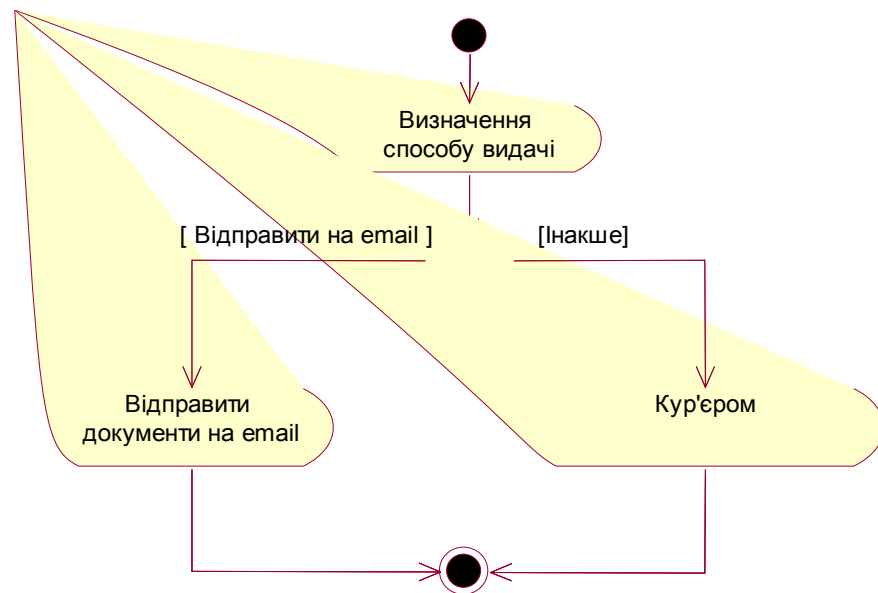


Рисунок 2.7 – Діаграма діяльності для варіанта використання "Видача путівки"

Специфікація варіанта використання "Інформація про наявні тури":

- a) Найменування варіанта використання "Інформація про наявні тури".
- b) Призначення: для отримання інформацію про наявні тури.
- c) Учасники: активний суб'єкт "Робітник турагенства".
- d) Попередня дія: будь-яка яка інша дія в системі.
- e) Стартова дія: ініціюється активним суб'єктом "Робітник турагенства".
- f) Потоки подій: користувач заповнює інформацію про наявні тури.
- g) Результати: отримання списку наявних турів.
- h) Умови після закінчення використання: відсутні.

Діаграма діяльності для варіанта використання "Інформація про наявні тури" на рисунку 2.8.

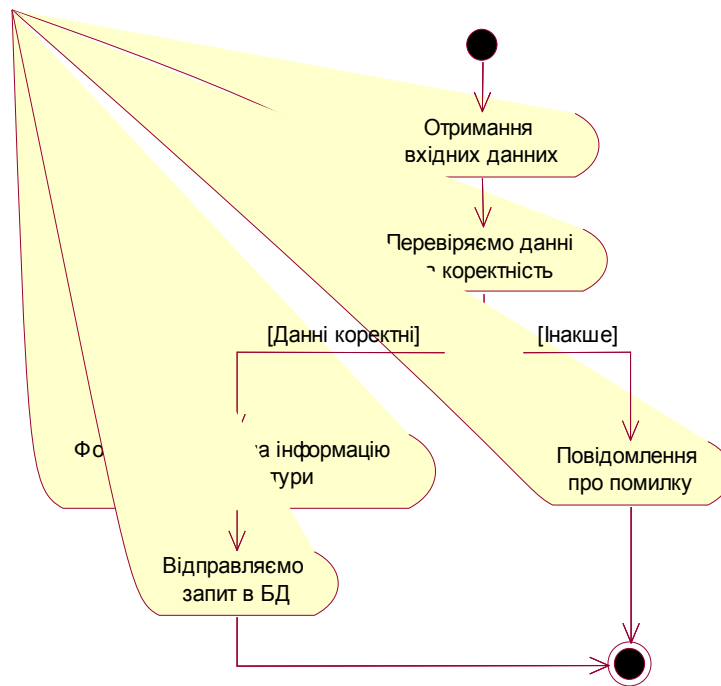


Рисунок 2.8 – Діаграма діяльності для варіанта використання "Інформація про наявні тури"

2.1.2 Проектування інтерфейсу програмного забезпечення довідково-інформаційно системи

На основі описаних вище варіантів використання визначимо інтерфейси користувача. Так, проєкт інтерфейсу для варіанту використання "Авторизуватися" представлений на рисунку 2.9.

Логін
Пароль
<input type="button" value="OK"/> <input type="button" value="Вийти"/>

Рисунок 2.9 – Проєкт інтерфейсу користувача для варіанту використання "Авторизуватися"

Проєкт інтерфейсу користувач для варіанту використання "Реєстрація" зображено на рисунку 2.10.

Будь ласка зареєструйтесь	
Ім'я	<input type="text"/>
Прізвище	<input type="text"/>
Дата народження	<input type="text"/>
Email	<input type="text"/>
Логін	<input type="text"/>
Пароль	<input type="text"/>
Підтвердження пароля	<input type="text"/>
<input type="button" value="OK"/>	<input type="button" value="Вийти"/>

Рисунок 2.10 – Проект інтерфейсу користувача для варіанту використання "Реєстрація"

Проект інтерфейсу користувач для варіанту використання "Запит інформація про тури" зображено на рисунку 2.11.

Проект інтерфейсу користувач для варіанту використання "Запит путівки" зображено на рисунку 2.12.

Проект інтерфейсу користувач для варіанту використання "Переказ коштів" зображено на рисунку 2.13.

Місто відбуття	<input type="text"/>	Тип туру	<input type="text"/>
Країна прибуття	<input type="text"/>	Курорт	<input type="text"/>
Дата заїзду	<input type="text"/>	Тур	<input type="text"/>
Кількість ночей	<input type="text"/>	Дорослих осіб	<input type="text"/>
Тип харчування	<input type="text"/>	Дітей	<input type="text"/>
Готелі по туру	<input type="text"/>	Максимальна ціна	<input type="text"/>
			<input type="button" value="Пошук"/>

Рисунок 2.11 – Проект інтерфейсу користувача для варіанту використання "Запит інформації про тур"

Оформлення Заявки	
НАЗВА ТУРУ	
Дата заїзду:	
Тур:	
Готель:	
Курорт:	
Харчування:	
Тривалість:	
Ціна:	
У рахунок включено:	
Додаткова оплата:	
Вкажіть ваші персональні дані:	
Ім'я:	
Телефон та e-mail:	
Примітки і додаткові відомості:	

Рисунок 2.12 – Проект інтерфейсу користувача для варіанту використання "Запит путівки"

Номер заказу:
Вид розрахунку:
1) Оплатити за допомогою карти
Номер карти
Таємний код
2) Розрахунок готівкою
Країна
Почтовий індекс
Місто
Вулиця
Отримувач

Рисунок 2.13 – Проект інтерфейсу користувача для варіанту використання "Переказ коштів"

Проект інтерфейсу користувач для варіанту використання "Інформація про наявні тури" зображено на рисунку 2.14.

Назва туру
Країна:
Місто відбуття:
Дата заїзду:
Маршрут:
Про тур:
Варіанти туру:
Програма дозвілля:
У вартість включено:
1) Розміщення в готелях рівня "економ", 3 * або 4 *, відповідно до замовлення: Експерсії з гідом:
2) Харчування:
3) Всі трансфери:
4) Послуги представника:
5) Медичний страховий поліс;
6) Авіапереліт:
Доплати (обов'язкові)
1) Консульський збір ;
2) Страхівка від невіїзду ;
3) Навушники для прослуховування експерсій ;
Паливний збір .
Не включено у вартість туру:

Рисунок 2.14 – Проєкт інтерфейсу користувача для варіанту використання "Інформація про наявні тури"

2.2 Технічний проєкт програмного забезпечення

У даному підрозділі програмне забезпечення у якості системи, яка утворюється з елементів (модулів). Ці модулі досить прості для реалізації. Визначаються зв'язки між модулями які класифікуються за даними і управлінням, що забезпечує функціонування всієї системи відповідно до вимог. Нижче наведені "статичні" та "динамічні" моделі програмного забезпечення.

2.2.1 Проєктування моделі даних програмного забезпечення

На підставі технічного завдання, з'являється необхідність створення бази даних. Визначимо логічні зв'язки між сутностями та розробимо схему даних, у

вигляді концептуальної моделі даних, яка представлена за допомогою діаграми класів, та зображена на рисунку 2.15.

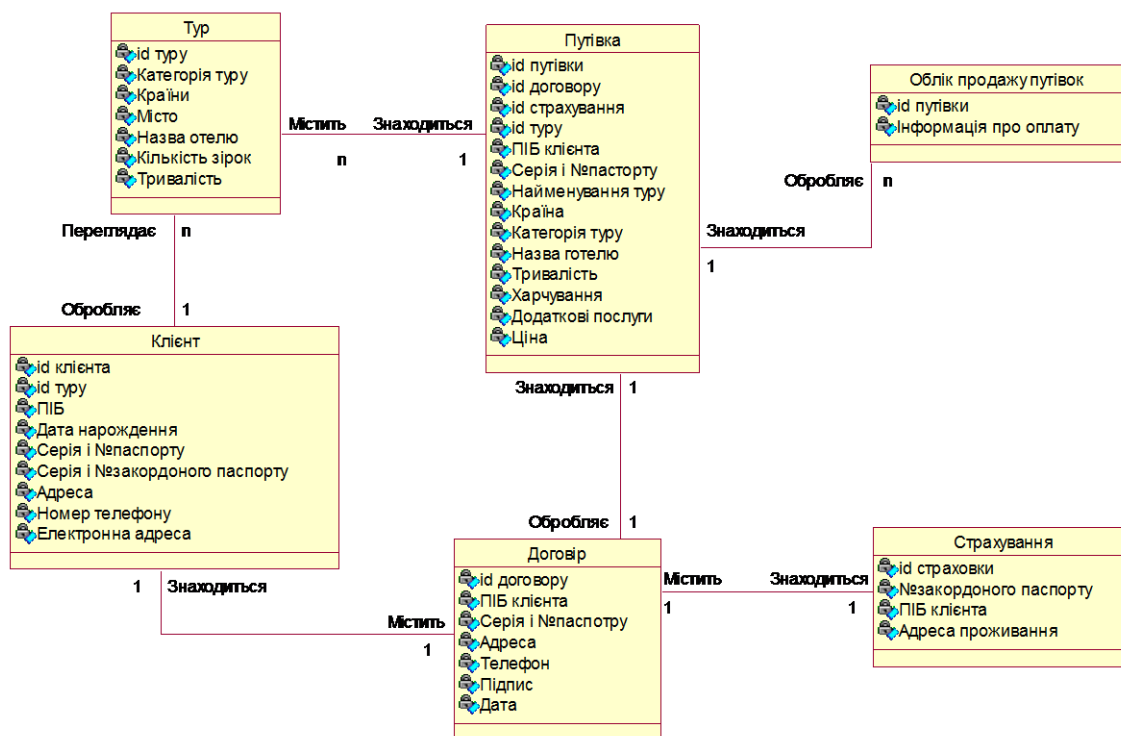


Рисунок 2.15 – Діаграма класів для концептуальної моделі даних ПЗ довідково-інформаційно системи туристичного агенства

Концепт моделі являє собою сукупність об'єктів та їх зв'язків, не вказуючи, як вони фізично зберігаються. Отже, концепт насправді є "моделлю" поля зору. Розробляючи концептуальну модель максимум зусиль розробників зосереджуються насамперед на структуруванні інформації та виявленні їхні взаємозв'язки, незалежно від ефективності обробки та реалізації. Розробляючи концептуальну модель, розробник спирається на аналізуванні завдань, яке вирішує підприємство. Концептуальні моделі включають описи об'єктів, що представляють інтерес, та їх взаємозв'язків у цій області перегляду та визначаються як результат аналізу даних.

Побудуємо діаграму аналітичної моделі програмного забезпечення, що приймає веб-замовлення, що показує об'єкти, які використовує програмне забезпечення. Об'єкт - це екземпляр класу. Цей клас визначає набір об'єктів, які мають

спільні властивості (властивості), поведінку (функції), семантику та зв'язки з іншими об'єктами.

Класам відповідають певні стереотипи, що дозволяють створювати нові різновиди елементів модельованого програмного забезпечення. У моделі аналізу, до таких стереотипів відносяться:

1) entity (сутність). Клас сутність моделює структуру даних і поведінку, що відрізняються стабільним характером. Клас подібного типу відображає якості сутності реального миру або застосовується для виконання внутрішніх функцій програмного забезпечення. Класи сутності зазвичай не залежать від оточення, тобто не чутливі до того, яким чином зовнішні обставини впливають на роботу програмного забезпечення. У багатьох випадках такі класи вдається з успіхом застосовувати в декількох різних програмних продуктах.

2) boundary (межа). Класи меж обслуговують процеси взаємодії між програмним забезпеченням і його оточенням, забезпечуючи інтерфейси для користувачів і сторонніх програмних продуктів. Такі класи утворюють частину програмного забезпечення, яка безпосередньо "спілкується" із зовнішнім світом.

3) control (управління). Класи управління знаходять застосування при реалізації характеристик поведінки програмного забезпечення, властивих одному або декільком варіантам використання, і координують події, що виникають у міру функціонування ПЗ в рамках цих варіантів. Клас управління можна сприймати як абстракцію, що відображає динаміку варіанту використання програмного забезпечення. Класи управління зазвичай тісно "прив'язані" до особливостей конкретного програмного продукту.

Діаграма моделі аналізу для програмного забезпечення отримання замовлень через мережу Internet зображена на рисунку 2.16.

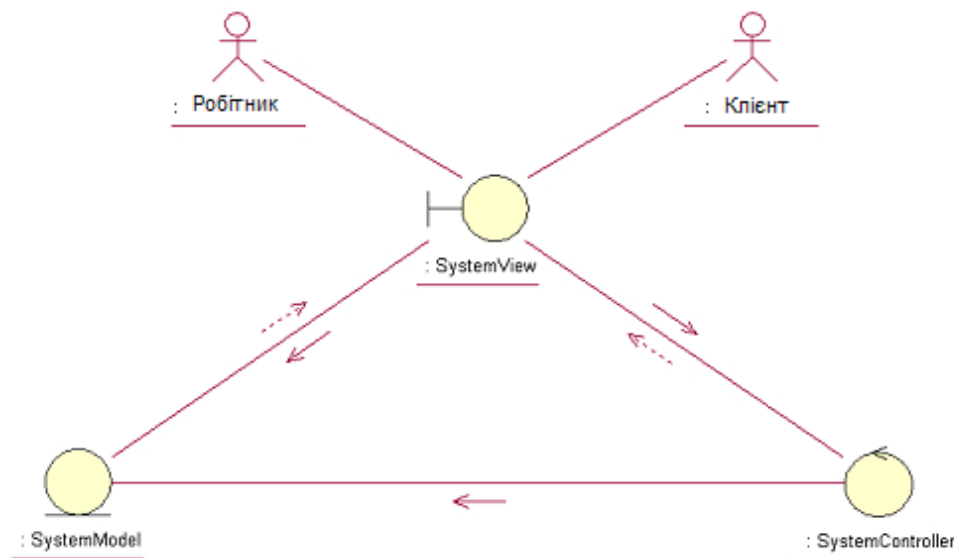


Рисунок 2.16 – Діаграма моделі аналізу для ПЗ довідково-інформаційно системи туристичного агенства

Для побудови програмного продукту була вибрана методологія MVC. Це архітектурна модель ПЗ, де модель даних програмного забезпечення, користувацький інтерфейс та логіка керування містять в собі три окремі компоненти, тому зміни в одному компоненті мінімально впливають на інші.

Шаблони MVC дозволяють розділити дані, перегляди та обробку дій користувача на три окремі компоненти:

- 1) Модель. "Моделі" надають дані (зазвичай для Views) і відповідають на запити (зазвичай від контролерів), змінюючи їх стан.
- 2) Перегляд. Відповідає за відображення інформації (інтерфейс користувача).
- 3) Поведінка (Контролер). Інтерпретувати дані, введені користувачем, і повідомляти моделі та ідеї про необхідність відповідної реакції.

Важливо відзначити, що як сприйняття, так і поведінка залежать від моделі. Однак модель не покладається на уяву чи поведінку. Це одна з головних переваг даної галузі. Воно дозволяє будувати модель незалежно від візуального уявлення, а також створювати декілька різних уявлень для однієї моделі.

2.2.2 Статична модель програмного забезпечення довідково-інформаційно системи

Розроблену вище аналітичну модель діаграми необхідно вивести на технічний рівень. Використовується у вигляді статичної моделі – діаграми класів програмного забезпечення.

Діаграми класів використовуються для представлення статичних програмних моделей і, зокрема, можуть відображати різні відносини між різними сутностями (такими як об'єкти та підсистеми) у предметній області, а також їхню внутрішню структуру та типи зв'язків.

Ця діаграма не містить інформації про терміни виконання програмних операцій. Він складається з багатьох елементів, які разом відображають декларативні знання предметної області. Ці знання інтерпретуються як фундаментальні поняття мови UML, такі як класи та їх зв'язки.

На рисунку 2.17 показана схема класів програмного забезпечення системи пошуку інформації туристичної агенції.

Для більш детального сприйняття ієрархії класів приведемо специфікацію головних класів:

1) Main – головний клас програми, який відповідає за з'єднання з базою даних, містить в собі методи для отримання даних з БД. Зв'язує всі елементи програми та забезпечує функціонування.

2) Tour – клас відповідає за взаємодію частини програмного забезпечення (каталогу турів та адміністрування каталогу турів) з базою даних, зокрема, приймає параметри та визиває SQL запити, через клас Main, з цими параметрами, після обробки БД, повертає результат.

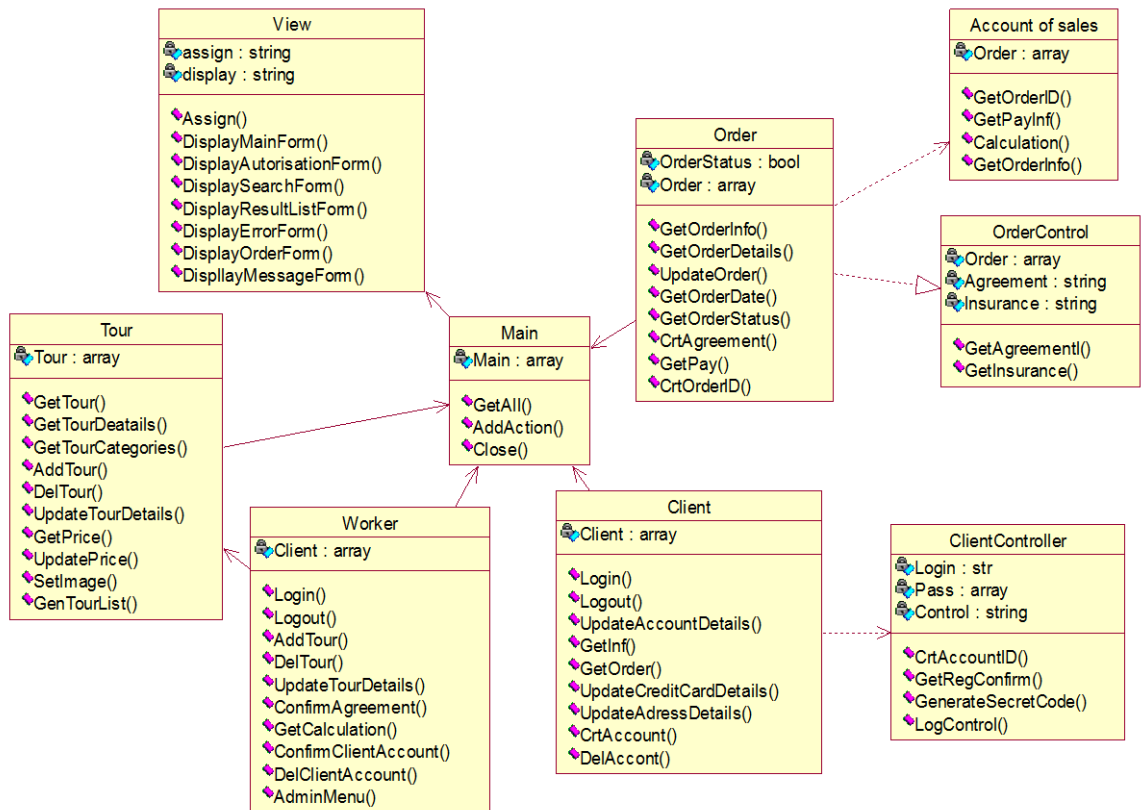


Рисунок 2.17 – Діаграма класів для ПЗ довідково-інформаційно системи туристичного агенства

3) Client – клас відповідає за взаємодію частини програмного забезпечення (Аутентифікація користувача, реєстрація покупця, редагування інформації покупця, тобто всіма операціями в системі над користувачами) з базою даних, зокрема, приймає параметри та визиває SQL запити, через клас Main, з цими параметрами, після обробки БД, повертає результат.

4) Worker – клас відповідає за взаємодію частини програмного забезпечення (Аутентифікація адміністратора, редагування інформації сайту, тобто всіма операціями в системі) з базою даних, зокрема, приймає параметри та визиває SQL запити, через клас Main, з цими параметрами, після обробки БД, повертає результат.

5) Order – клас відповідає за взаємодію частини програмного забезпечення (Замовлення туру (путівки), адміністрування замовлень) з базою даних, зокрема, приймає параметри та визиває SQL запити, через клас Main, з цими параметрами, після обробки БД, повертає результат.

6) Account of sales – цей клас обробляє всю вхідну інформацію що відноситься до каталогу Order(перевіряє наявність та коректність вхідних даних, а також інформацію про оплату послуг), після виконання всіх необхідних операцій з БД, формує результати, та зберігає їх в атрибутах.

7) ClientController – цей клас обробляє всю вхідну інформацію що відноситься до каталогу Client(перевіряє наявність та коректність даних авторизації та реєстрації), після виконання всіх необхідних операцій з БД, формує результати, та зберігає їх в атрибутах.

8) View – спеціальний клас, який забезпечує взаємодію між класами контролерами та представленням (являється HTML, XML документами), тобто зчитує данні з атрибутів класів контролерів та надає можливість виводу їх на екран в представленні.

9) OrderControl – цей клас обробляє вхідну інформацію що відноситься до каталогу Order (створює лист договору та страхування), формує результати.

2.2.3 Динамічна модель програмного забезпечення довідково-інформаційно системи

Реалізацією варіантів використання є динамічна модель, яку можна представити у вигляді діаграм послідовностей, наведемо діаграми для деяких варіантів використання. Діаграму послідовностей, яка є реалізацією варіанту використання "Аутифікація (клієнта)" наведено на рисунку 2.18.

Діаграму послідовностей, яка є реалізацією варіанту використання "Запит інформації про тури" наведено на рисунку 2.19.

Діаграму послідовностей для варіанту використання "Запит путівки (замовлення)" наведено на рисунку 2.20.

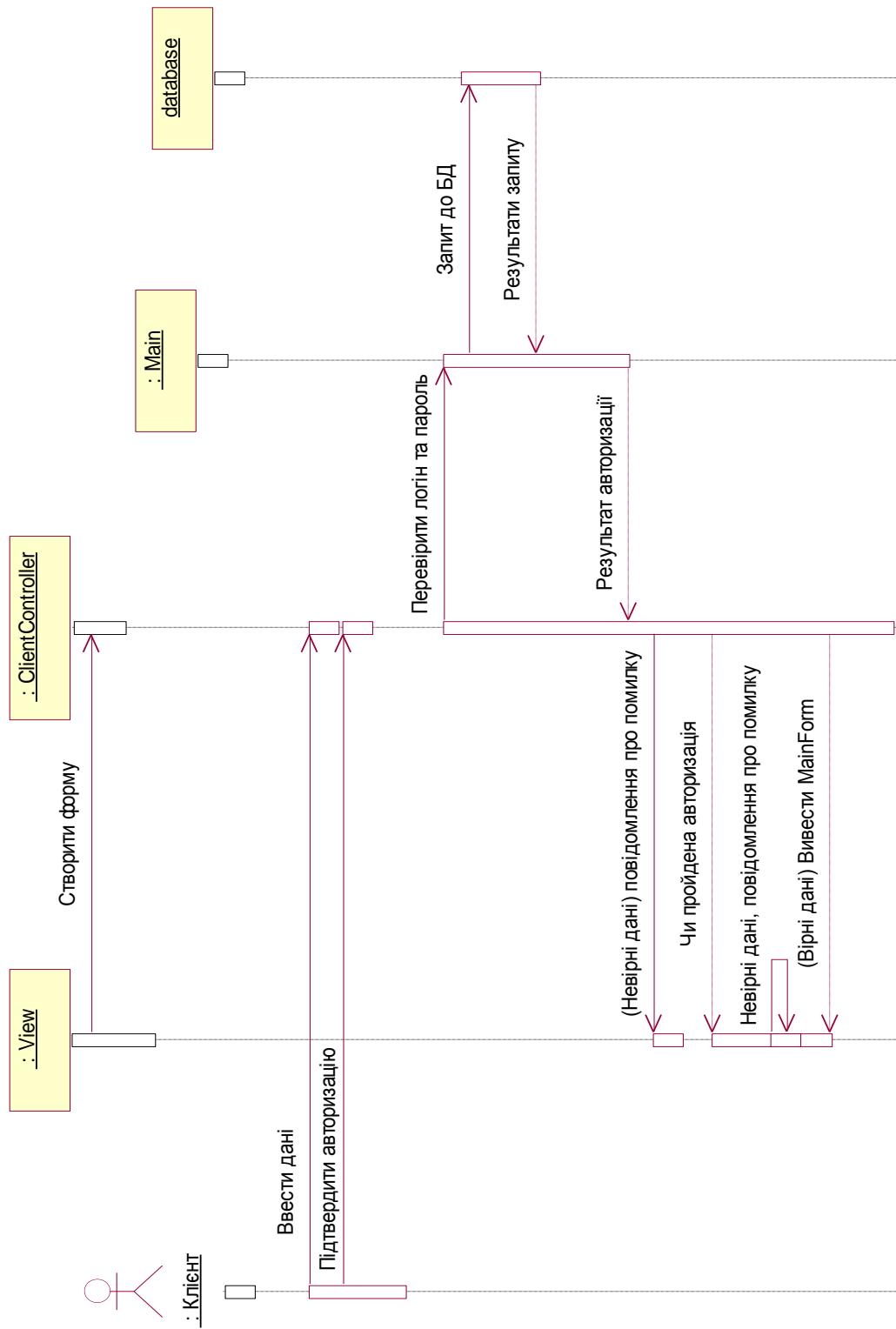


Рисунок 2.18 – Діаграма послідовностей для варіанту використання "Аутифікація (клієнта)"

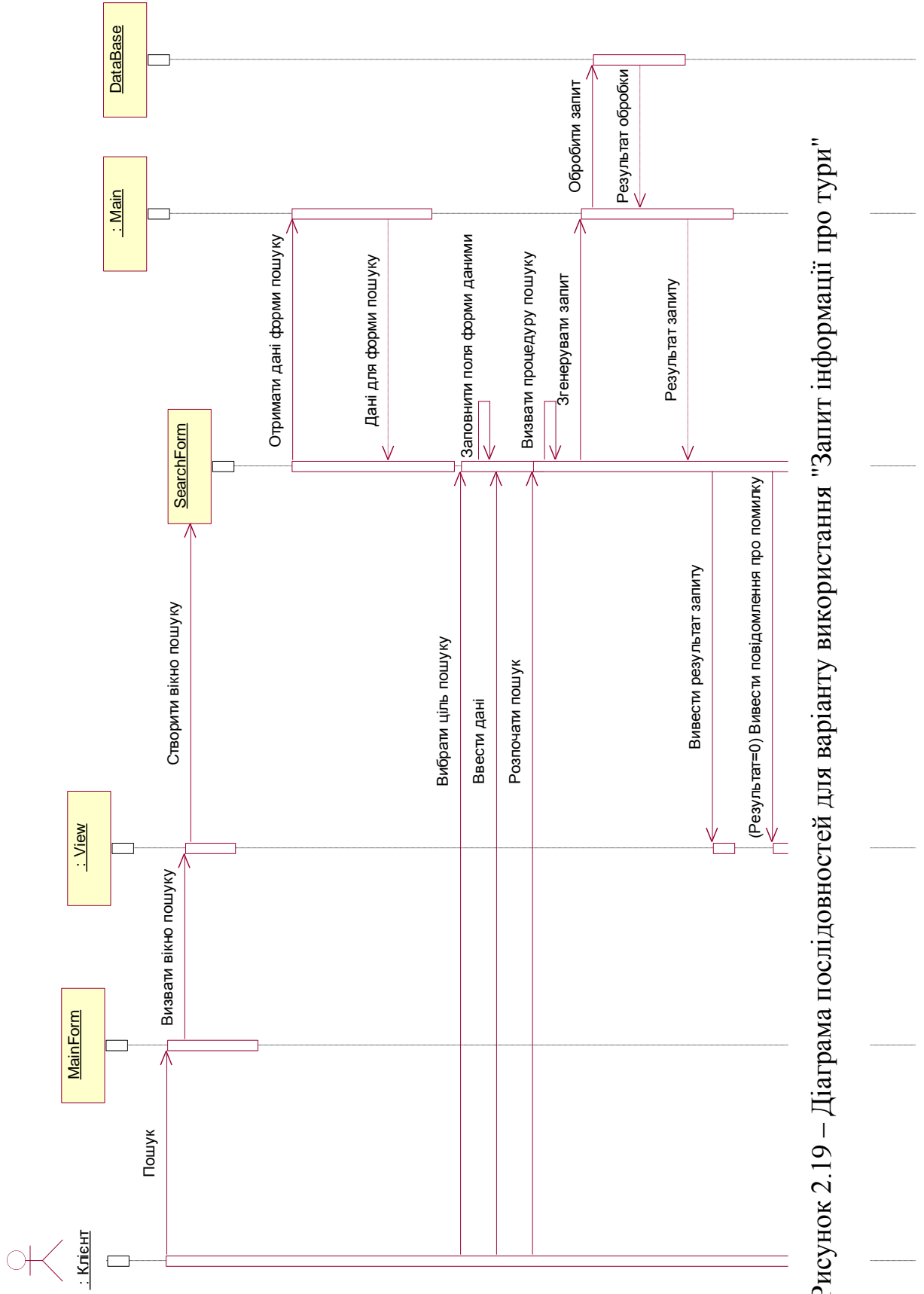


Рисунок 2.19 – Діаграма послідовностей для варіанту використання "Запит інформації про тури"

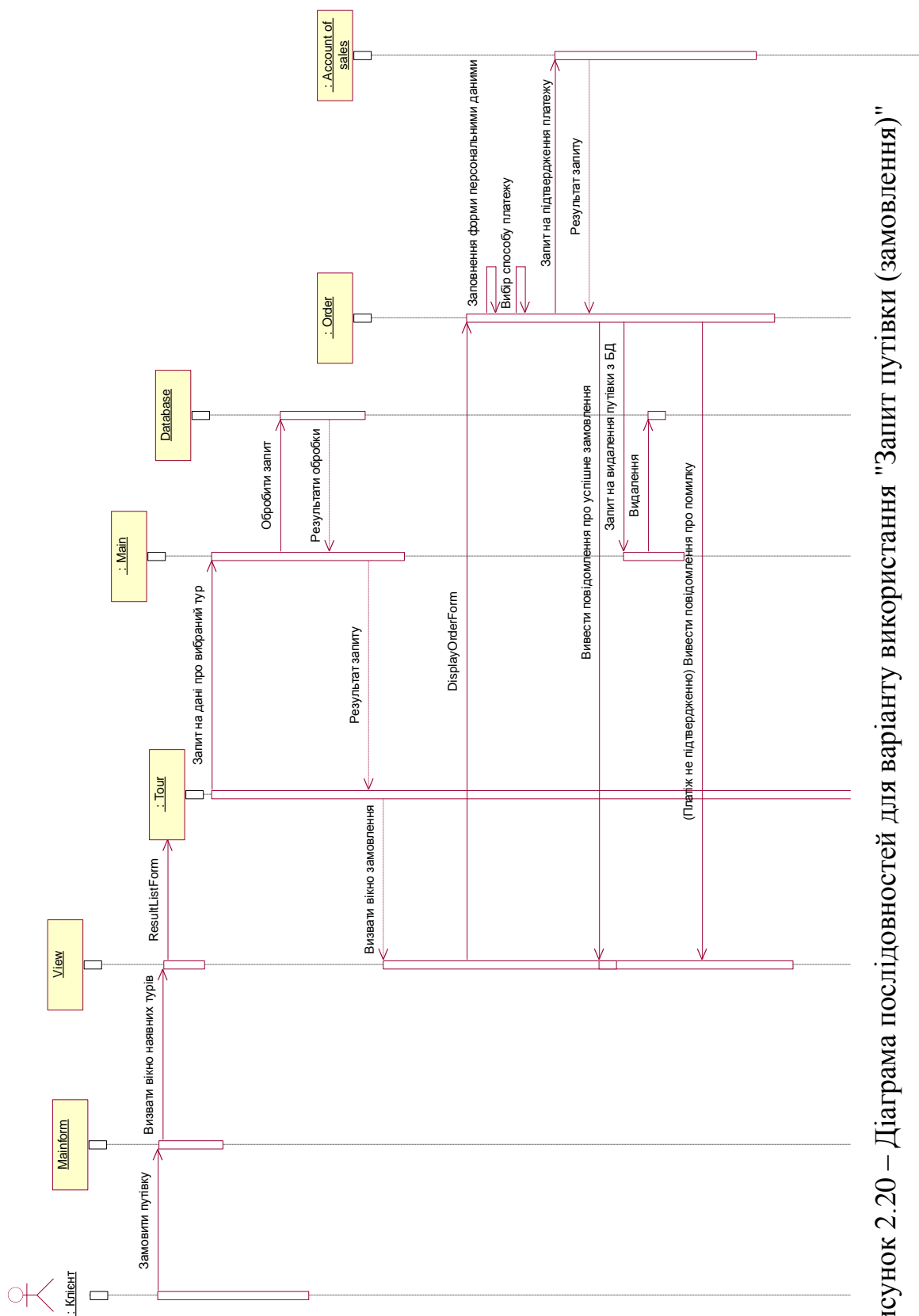


Рисунок 2.20 – Діаграма послідовностей для варіанту використання "Запит пультки (замовлення)"

2.2.4 Логічні моделі структур даних програмного забезпечення

Після визначення структури та змісту даних, які необхідно зберігати в БД побудуємо логічну модель даних інформаційного сховища даних програмного забезпечення.

Логічну модель даних для програмного забезпечення бронювання туристичних послуг в режимі реального часу будуємо на основі концептуальної моделі даних представленої на рисунку 2.21.

Більш детальна інформація по атрибутам сутностей логічної моделі наведена в таблицях 2.1-2.6.

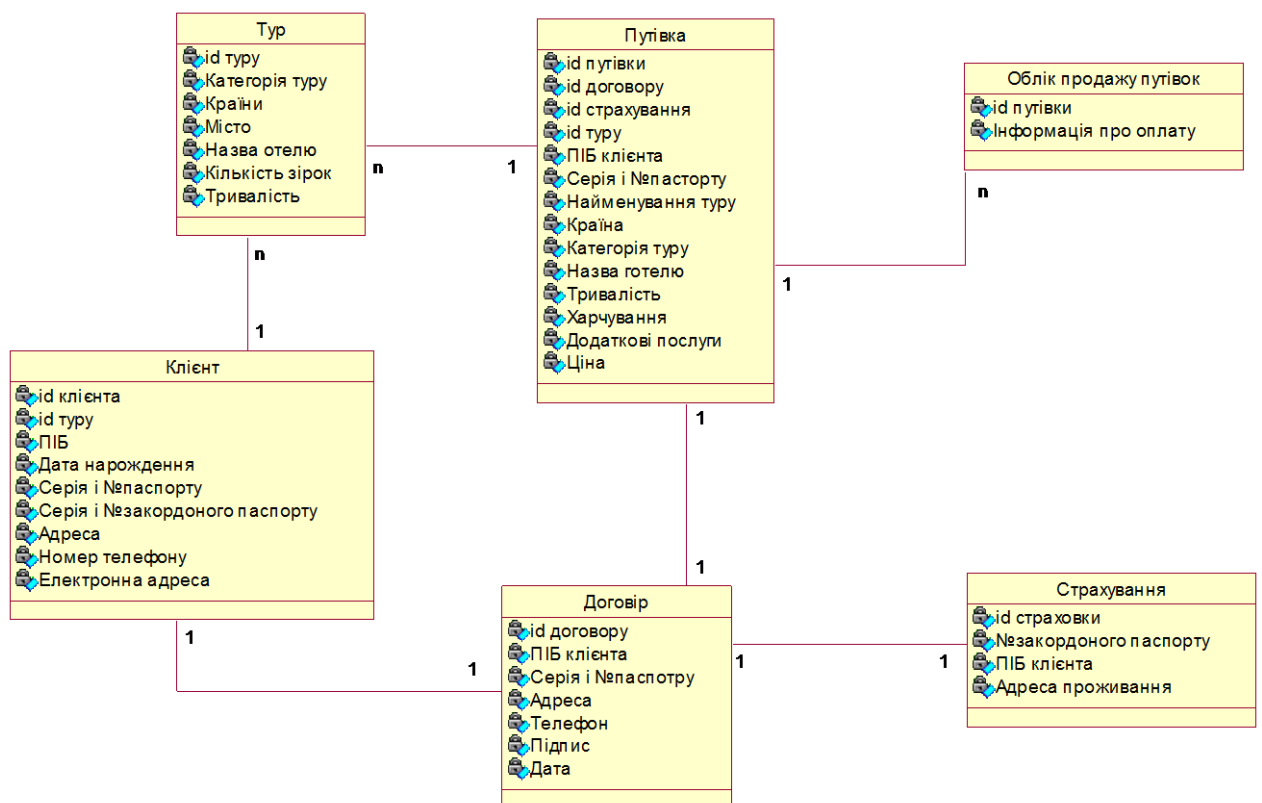


Рисунок 2.21 – Логічна модель даних ПЗ
довідково-інформаційно системи туристичного агенства

Усі атрибути, які мають тип "Лічильник" є первинними ключами. Зв'язані з ними атрибути інших сутностей мають аналогічну назву та тип "Зовн. ключ".

Таблиця 2.1 – Структура даних сутності "Тур"

Атрибут	Тип даних	Опис
Id туру	Лічильник	Ідентифікаційний номер туру
Категорія туру	Текстовий	Категорія туру
Країна	Текстовий	Назва країни
Місто	Текстовий	Назва міста
Назва готелю	Текстовий	Назва готелю
Кількість зірок	Числовий	Кількість зірок готелю
Тривалість	Дата	Дата початку та закінчення туру

Таблиця 2.2 – Структура даних сутності "Путівка"

Атрибут	Тип даних	Опис
Id путівки	Числовий	Ідентифікаційний номер путівки
Id договір	Числовий	Ідентифікаційний номер договору
Id туру	Числовий	Ідентифікаційний номер туру
ПІБ	Текстовий	Прізвище ім'я по батькові туриста
Серія і №паспорту	Текстовий	Серія і №паспорту туриста
Найменування туру	Текстовий	Найменування туру
Країна	Текстовий	Назва країни
Категорії туру	Текстовий	Категорії туру
Назва готелю	Текстовий	Назва готелю
Тривалість	Дата	Тривалість туру
Харчування	Текстовий	Вид харчування
Додаткові послуги	Текстовий	Додаткові послуги
Ціна	Числовий	Вартість путівки

Таблиця 2.3 – Структура даних сутності "Клієнт"

Атрибут	Тип даних	Опис
Id клієнта	Числовий	Ідентифікаційний номер клієнта
Id туру	Числовий	Ідентифікаційний номер туру
ПІБ	Числовий	Прізвище ім'я по батькові клієнта
Дата народження	Текстовий	Дата народження
Серія і №паспорту	Дата	Серія і №паспорту клієнта
Серія і №закордонного паспорту	Текстовий	Серія і №закордонного паспорту
Номер телефону	Числовий	Номер телефону клієнта
Адреса	Текстовий	Адреса клієнта
Електронна адреса	Числовий	Електронна адреса клієнта

Таблиця 2.4 – Структура таблиці "Договір"

Атрибут	Тип даних	Опис
Id договору	Числовий	Ідентифікаційний номер договору
ПІБ	Числовий	Прізвище ім'я по батькові турагента
Серія і №паспорту	Текстовий	Серія і №паспорту туриста
Адреса	Текстовий	Адреса проживання туриста
Телефон	Текстовий	Телефон туриста
Підпис	Числовий	Підпис туриста
Дата	Текстовий	Дата складання договору

Таблиця 2.5 – Структура таблиці "Страхування"

Атрибут	Тип даних	Опис
Id страхування	Числовий	Ідентифікаційний номер страхування
Серія і № закордонного паспорту	Текстовий	Серія і номер закордонного паспорту туриста
ПІБ	Текстовий	Прізвище ім'я по батькові туриста
Адреса проживання	Текстовий	Адреса проживання туриста

Таблиця 2.6 – Структура таблиці "Облік продажу путівок"

Атрибут	Тип даних	Опис
Id путівки	Числовий	Ідентифікаційний номер путівки
Інформація про оплату	Числовий	Інформація про оплату

2.3 Робочий проект програмного забезпечення

2.3.1 Вибір засобів розв'язання поставленої задачі

Для розробки програмного забезпечення інформаційно-пошукової системи туристичного агенства можна використовувати різні мови: Python, JavaScript PHP, JAVA, ASP і інші. Кожна мова має свої переваги і недоліки.

Результат аналізу мов програмування, які теоретично могли б бути використані для розробки програмного продукту, показав, що більшість мов мають стандартний набір бібліотек і компонентів.

В Python, JavaScript, PHP, JAVA, ASP реалізовано повну підтримку об'єктно-орієнтованого програмування. ASP має перевагу в швидкості в порівнянні з Python, JavaScript PHP, JAVA. Такі ж мови програмування як Python, PHP, JavaScript, JAVA мають потужні вбудовані функції для обробки текстової інформації. Але за рахунок того, що JavaScript має широку браузерну підтримку, тому для створення проекту було обрано JavaScript. Під час розробки на vanilla JS, було прийнято рішення використати фреймворк Vue.js. Чому саме Vue? Тому що він досить простий в засвоєнні та не втрачає від цього ефективності. Це один з найсучасніших засобів для веб-розробки, який також зараз має дуже великий попит.

Для бази даних будемо використовувати СУБД MongoDB.

MongoDB – opensource документоорієнтована система, яка використовується для побудови та керування реляційними базами даних. Широко використовується в проектах написаних на JavaScript. Головний плюс, це її безкоштовність! OPENSOURCE У МАСИ!

Кодування програми проводиться на основі статичної (пункт 2.2.2) та динамічної (пункт 2.2.3) моделей технічного проєкту ПЗ бронювання туристичних послуг в режимі реального часу.

Кодування програми буде проводитись на мові програмування JavaScript.

Весь текст програми наведено у додатку Б.

2.3.2 Фізична модель даних програмного забезпечення довідково-інформаційно системи

На основі розробленої логічної моделі програмного забезпечення інформаційно-пошукової системи туристичного агентства побудуємо фізичну модель даних.

На основі таблиць логічної моделі структури даних, зобразимо фізичну модель програми, для чого побудуємо фізичні моделі таблиць, використовуючи типи СУБД MongoDB (рисунок 2.22). Фізичні моделі таблиць даних наведено у таблицях 2.7-2.12.

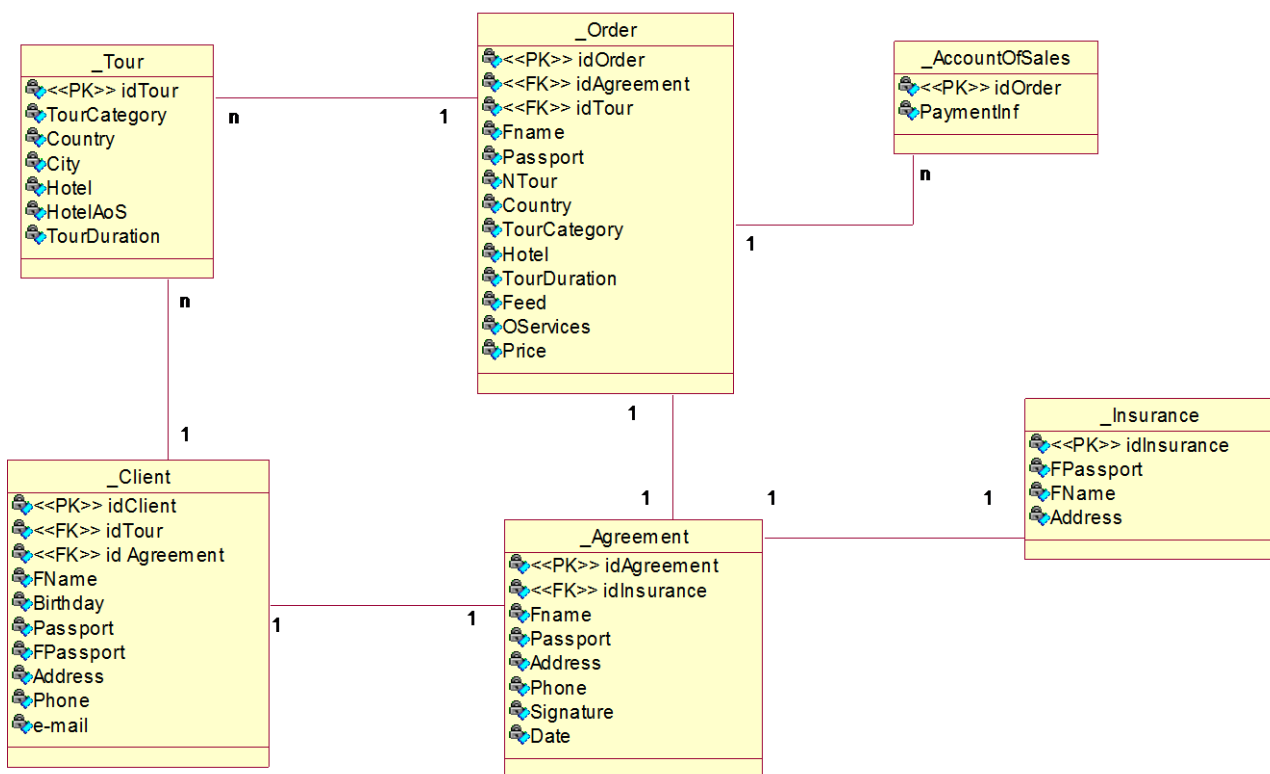


Рисунок 2.22 – Фізична модель даних ПЗ довідково-інформаційно системи туристичного агентства

Таблиця "Tour" призначена для зберігання інформації про тури. Структуру фізичної моделі сутності "Tour" наведено у таблиці 2.7.

Таблиця 2.7 – Фізична модель таблиці "Tour"

Атрибут	Тип даних	Тип в БД	PK	FK
Id Tour	Лічильник	Int(10)	*	
TourCategory	Текстовий	Varchar(100)		
Country	Текстовий	Varchar(100)		
City	Текстовий	Varchar(100)		
Hotel	Текстовий	Varchar(100)		
HotelAos	Числовий	Int(5)		
TourDuration	Дата	DateTime		

Таблиця "Agreement" призначена для зберігання інформації договору. Структуру фізичної моделі сутності "Agreement" наведено у таблиці 2.8.

Таблиця 2.8 – Структура таблиці "Agreement"

Атрибут	Тип даних	Тип в БД	PK	FK
idAgreement	Числовий	Int(10)	*	
idInsurance	Числовий	Int(10)		*
Fname	Текстовий	Varchar(100)		
Passport	Текстовий	Varchar(100)		
Address	Текстовий	Varchar(100)		
Phone	Числовий	Int(12)		
Signature	Текстовий	Varchar(100)		
Date	Дата	DateTime		

Таблиця "Client" призначена для зберігання інформації про клієнта. Структуру фізичної моделі сутності "Client" наведено у таблиці 2.9

Таблиця 2.9 – Структура даних сутності "Client"

Атрибут	Тип даних	Тип в БД	PK	FK
idClient	Числовий	Int(10)	*	
idTour	Числовий	Int(10)		*
idAgreement	Числовий	Int(10)		*
Fname	Текстовий	Varchar(100)		
Birthday	Дата	DateTime		
Passport	Текстовий	Varchar(100)		

FPassport	Текстовий	Varchar(100)		
Address	Текстовий	Varchar(100)		
Phone	Числовий	Int(12)		
e-mail	Текстовий	Varchar(100)		

Таблиця "Order" призначена для зберігання інформації про путівку. Структуру фізичної моделі сутності "Order" наведено у таблиці 2.10.

Таблиця 2.10 – Структура даних сутності "Order"

Атрибут	Тип даних	Тип в БД	PK	FK
idOrder	Числовий	Int(10)	*	
idAgreement	Числовий	Int(10)		*
idTour	Числовий	Int(10)		*
Fname	Текстовий	Varchar(100)		
Passport	Текстовий	Varchar(100)		
Ntour	Текстовий	Varchar(100)		
Country	Текстовий	Varchar(100)		
TourCategory	Текстовий	Varchar(100)		
Hotel	Текстовий	Varchar(100)		
TourDuration	Дата	DateTime		
Feed	Текстовий	Varchar(100)		
Oservices	Текстовий	Varchar(1000)		
Price	Числовий	Decimal(6,2)		

Таблиця "Insurance" призначена для зберігання інформації про страхування. Структуру фізичної моделі сутності "Insurance" наведено у таблиці 2.11.

Таблиця 2.11 – Структура таблиці "Insurance"

Атрибут	Тип даних	Тип в БД	PK	FK
idInsurance	Числовий	Int(10)	*	
FPassport	Текстовий	Varchar(100)		
FName	Текстовий	Varchar(100)		
Address	Текстовий	Varchar(100)		

Таблиця "AcconutOfSales" призначена для зберігання інформації обліку продаж путівок. Структуру фізичної моделі сутності "AcconutOfSales" наведено у таблиці 2.12.

Таблиця 2.12 – Структура таблиці "AcconutOfSales"

Атрибут	Тип даних	Тип в БД	PK	FK
idOrder	Числовий	Int(10)	*	
PaymentInf	Числовий	Decimal(10,2)		

3 ЯКІСТЬ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Тестування класів програмного забезпечення довідково-інформаційно системи

Дві стратегії тестування програмного забезпечення стали найбільш популярними.

Перший, відомий як стратегія чорного ящика, — це тестування на основі даних, так зване "тестування вводом-виводом". Використовуючи цю стратегію, додаток вважається "чорним ящиком". Такі тести призначені для визначення, випадків поведінки програми не відповідаючи своїм специфікаціям.

Тестові дані, використні під час перевірки, відповідають специфікації (тобто виключаючи знання її внутрішньої структури). Використовуючи цей метод, для знайдення всіх помилок у програмі є критерієм комплексного тестування введення. Останнього можна досягти, використовуючи всі уявні вхідні набори як тестові набори.

Методика «білого ящика» чи алгоритм тестування, який контролює логіку додатку, дозволяючи дослідити внутрішню структуру програми. В такому варіанті розвитку подій людина отримує дані тесту, аналізуючи логіку програми. Цей підхід характеризується мірою, в якій тест виконує та охоплює логіку програми (вихідний текст). Комплексне тестування принципу "білого ящика" передбачає виконання кожного шляху в програмі. Існує кілька методів тестування, пов'язаних із цією стратегією: охоплення оператора, покриття рішення, умовне покриття, рішення/умовне покриття, комбіноване покриття. Для надійного тестування програмного забезпечення необхідно перевірити логіку кожного окремого процесу, модуля та перевірити їх взаємодію.

Через велику кількість програмного коду та складність програмних модулів підсистеми тестування за методом «білого ящика» займає багато часу, тому про-

грамне забезпечення, розроблене підсистемою, використовує для тестування лише метод «чорного ящика».

Будемо проводити тестування наступних класів програми – Main, View, Tour. Скористаємося методом еквівалентної розбивки правильних і неправильних класів. Цей метод дозволяє провести досить повне тестування програми.

Результати розбивки вхідних і вихідних даних на класи еквівалентності представлені в таблицях 2.13 – 2.15.

Таблиця 2.13 – Класи еквівалентності для класу Main

	Вхідні умови	Правильні класи	Неправильні класи
1	Параметри підключення к БД	Всі параметри підключення задані вірно.	Відсутність одного чи декількох параметрів підключення.
		Є доступ до місця зберігання інформації під час збереження параметрів підключення.	Один чи більше параметрів задано невірно.
			Заборонено в доступі до місця зберігання інформації.
2	Виконання SQL-запитів	Параметри SQL-запиту задані вірно.	Один чи більше параметрів задано невірно.
3	Генерація SQL-запитів	При генерації всі параметри запиту задані вірно та він відповідає стандарту.	Запит, що був сгенерований не відповідає стандарту.

Таблиця 2.14 – Класи еквівалентності для класу View

	Вхідні умови	Правильні класи	Неправильні класи
1	Зображення вікон	При створенні вікна присутня вся необхідна інформація.	Інформації при створенні вікна недостатньо.
2	Реакція на зміну значення полів	Зміна будь-якого поля з даними на коректне значення.	Зміна значення поля на некоректне значення.

Таблиця 2.15 – Класи еквівалентності для класу Tour

	Вхідні умови	Правильні класи	Неправильні класи
1	Параметри в адресному рядку	Всі параметри в адресному рядку задані вірно.	Відсутність одного чи декількох параметрів в адресному рядку.
			Один чи більше параметрів в адресному рядку задано невірно.
2	Параметри, передані через змінне оточення	Всі параметри передані через змінне оточення задані вірно.	Відсутність одного чи декількох параметрів переданих через змінне оточення.
3	Данні отримані в результаті виконання запиту до БД	Масив або рядок з даними.	Пустий рядок

Для виконання великої кількості SQL-запитів була розроблена невеличка програма. Результати тестування SQL-запитів представлені нижче:

```
SQL PROCEDURE = CALL `get_tour_info`(IN inidTour INT)
```

```
BEGIN
```

```
    SELECT tour_id, TourCategory, Country, City, Hotel, HotelAoS, TourDyration
```

```
        FROM tour WHERE tour_id = inTourId;
```

```
END$$
```

is active and works fine

```
SQL PROCEDURE = CALL `main_get_main`(IN inMain ARR)
```

```
BEGIN
```

```
IF arrGetAll = "on" THEN
```

```
PREPARE statement FROM
```

```
    SELECT DisplayMainForm FROM View WHERE DisplayMainForm =
inDisplayMainForm,
```

```

idTour FROM Tour WHERE idTour = arrTour,
idClient FROM Client WHERE idClient = arrClient,
idOrder FROM Order WHERE idOrder = arrOrder,
ELSE
PREPARE statement FROM
(DataBase Error)
END$$
is active and works fine
SQL PROCEDURE = CALL `get_view`(In inView varchar(1000))
BEGIN
IF assign = "on" THEN
PREPARE statement FROM
    SELECT DisplayMainForm FROM View WHERE DisplayMainForm =
inDisplayMainForm,
    ELSE
PREPARE statement FROM
    SELECT DisplayErrorForm FROM View WHERE DisplayErrorForm=
inDisplayErrorForm,
END$$

```

is active and works fine

Як бачимо з результатів, усі запити виконуються без помилок.

Тести, що покривають правильні й неправильні класи еквівалентності представлені в таблицях 2.16-2.18.

Таблиця 2.16 – Тести класів еквівалентності для класу Main

№	Класи	Вхідні дані	Вихідні дані
1	2	3	4
1	1	Всі параметри підключення задані вірно. База даних доступна.	Вдале підключення.
2	1	Всі параметри підключення задані вірно. База даних не доступна.	Вікно з повідомленням про помилку: 500 DataBase Error
3	1	Є доступ до місця зберігання інформації під час збереження параметрів підключення.	Успішне збереження параметрів підключення.
4	1	Відсутність одного чи декількох параметрів підключення.	Вікно з повідомленням про помилку: 500 DataBase Error
5	1	Один чи більше параметрів задано невірно.	Вікно з повідомленням про помилку: Incorrect number of arguments for PROCEDURE
6	1	Заборонено в доступі до місця зберігання інформації.	Вікно з повідомленням про помилку: Access denied for user
7	3	Генерація SQL-запитів	Коректна вибірка даних з БД.
8	3	Генерація SQL-запитів	Вікно з повідомленням про помилку: Incorrect number of arguments for PROCEDURE

Таблиця 2.17 – Тести класів еквівалентності для класу StirView

№	Класи	Вхідні дані	Вихідні дані
1	1	При створенні вікна присутня вся необхідна інформація.	Вдале створення вікна
2	1	Інформації при створенні вікна недостатньо.	Вікно з повідомленням про помилку: 405 Method Not Allowed
3	2	Зміна будь-якого поля з даними на коректне значення.	Зміна інших полів, які залежать від цього поля
4	2	Зміна значення поля на некоректне значення.	Вікно з повідомленням про помилку: The page that you have requested doesn't exist

Таблиця 2.18 – Тести класів еквівалентності для класу Tour

№	Класи	Вхідні дані	Вихідні дані
1	1	При передачі невірною параметру через адресний рядок, покупець буде перенаправлений на сторінку помилки (404)	Сторінка помилки з кодом 404: The page that you have requested doesn't exist
2	2	При створенні замовлення присутній хоча б один тур	Повідомлення про помилку: Your search generated no results
3	3	При створенні замовлення в БД існує інформація про замовника	Повідомлення про помилку: It is not enough to information about a customer.

Після проведеного тестування методом "чорного ящика" можна зробити висновок, що програма здійснює контроль над коректністю введених даних і перевіряє відповідність формату даних вимогам, висунутим у технічному завданні.

3.2 Діаграма компонентів програмного забезпечення довідково-інформаційно системи

Для кращого розуміння компонентного складу модулів програми, необхідно розробити діаграму компонентів, яка представлена на рисунку 2.23.

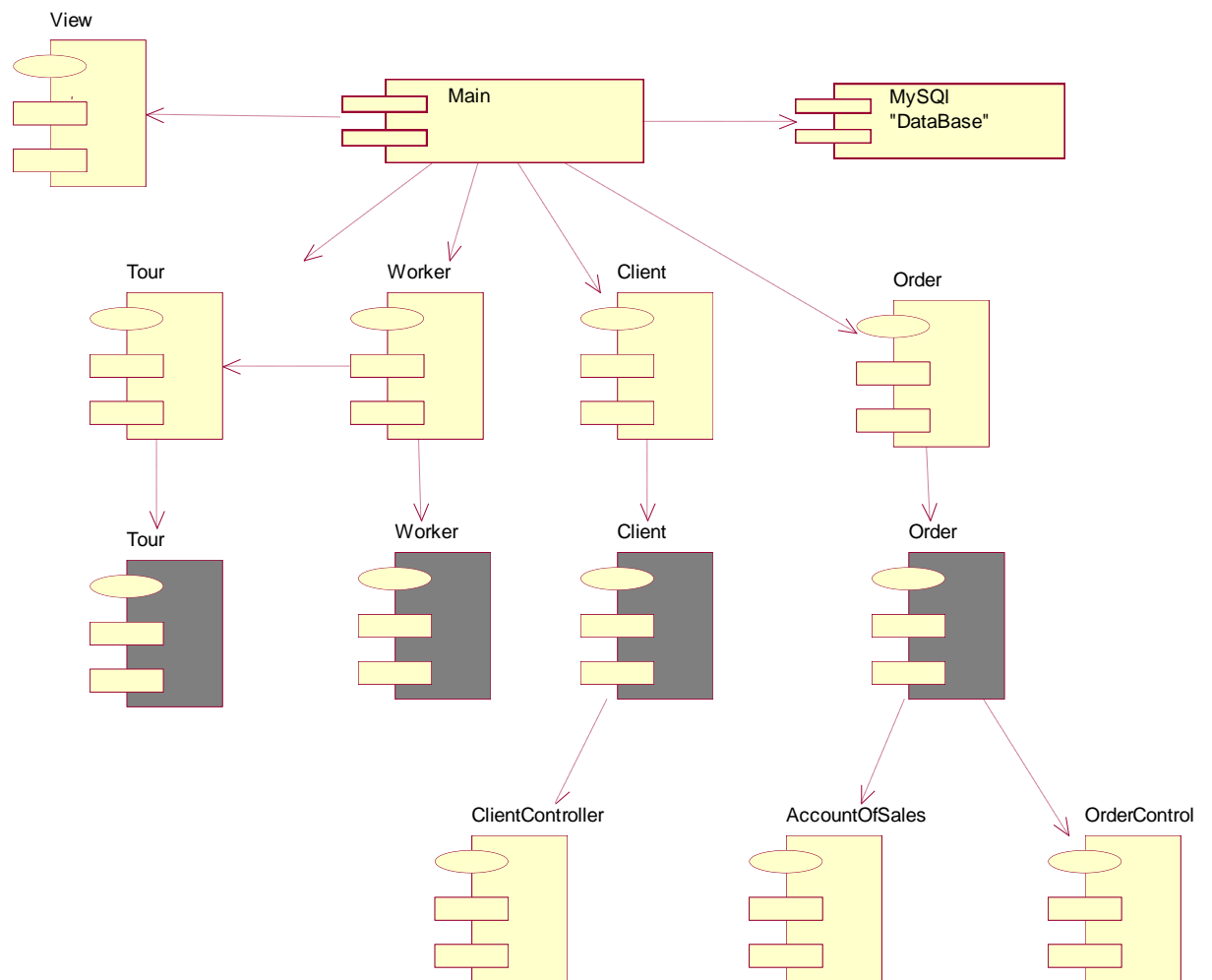


Рисунок 2.23 – Діаграма компонентів для ПЗ
довідково-інформаційно системи туристичного агенства

3.3 Кодування

Кодування програми проводиться на основі специфікації класів програми, представлених у технічному проекті.

```

<template>
  <div class="app">
    <the-header @showRegistration="openRegistrationForm" />
    <the-body
      @form-closed="openReg = false"
      :showRegistration="openReg"/>
    <the-footer />
  </div>
</template>

```

```

<script setup>
import TheHeader from "@/components/layout/TheHeader";
import TheBody from "@/components/layout/TheBody";
import TheFooter from "@/components/layout/TheFooter";
import { ref } from 'vue';
const openReg = ref(false);
function openRegistrationForm(){
openReg.value = true;
}

</script>

<style lang="scss" scoped>
.app{
  display: flex;
  min-height: 100vh;
  flex-direction: column;
}
</style>

```

Текст головного компоненту app. У нього будуть вмонтовані усі інші компоненти.

```

<template>
  <main class="body">
    <transition>
      <registration-form
        class="registration-form"
        v-if="showReg"
        @closeRegistration="closeRegistration"
      />
    </transition>
    <transition>
      <div class="container">
        <div class="body__inner" v-if="tours">
          <div class="body__inner-navigation main-container">
            <search-element
              @letSearch="letSearch"
              class="inner__search-element"> </search-element>
            <filters-row
              v-if="tours"
              @filterByPriceLow="filterByPriceLow"
              @filterByPriceHight="filterByPriceHight"
              @filterByCountryLow="filterByCountryLow"
              @filterByCountryHight="filterByCountryHight"
              @filterByDuraionLow="filterByDuraionLow"
              @filterByDurationHight="filterByDurationHight"
              class="inner__filters-row"> </filters-row>
            </div>
          <div class="main-container">

```

```

<div
  class="body__tours-table"
  :style="{ backgroundImage: `url(/tour-img/${tour.image}.jpg)` }"
  v-for="tour in tours"
  :key="tour.id"
>
  <table-row
    :tour="tour"
    @showForm="
      getTours(tour);
      openForm = true;
    "
  />
</div>
<transition>
  <booking-form v-if="openForm" @quitForm="openForm = false">
    <template v-slot:country>
      <div class="booking-form__characteristic">
        <h3>Країна:</h3>
        {{ bookingFormTour.content.country }}
      </div>
    </template>
    <template v-slot:city>
      <div class="booking-form__characteristic">
        <h3>Місто:</h3>
        {{ bookingFormTour.content.city }}
      </div>
    </template>
    <template v-slot:duration>
      <div class="booking-form__characteristic">
        <h3>Тривалість:</h3>
        {{ bookingFormTour.content.duration }}
      </div>
    </template>
    <template v-slot:price>
      <div class="booking-form__characteristic">
        <h3>Ціна:</h3>
        {{ bookingFormTour.content.price }}
      </div>
    </template>
    <template v-slot:description>
      <div class="booking-form__description">
        {{ bookingFormTour.content.description }}
      </div>
    </template>
    <template v-slot:image>
      <div class="booking-form__body-image"
        style="background-position: center;"
        :style="{ backgroundImage: `url(/tour-
img/${bookingFormTour.image}.jpg)` }" />
    </template>
  </transition>

```



```

        </booking-form>
      </transition>
    </div>
  </div>
</div>
</transition>
</main>
</template>

<script setup>
import { toursApi } from "@api/tours";
import TableRow from "@components/TableRow";
import { onMounted, ref, defineProps, watch, defineEmits } from "vue";
import SearchElement from "@components/UI/SearchElement";
import FiltersRow from "@components/UI/FiltersRow";
import BookingForm from "@components/bookingForm/BookingForm";
import RegistrationForm from "../registrationForm/RegistrationForm.vue";
const props = defineProps({
  showRegistration:{
    type: Boolean,
    default: () => false
  }
})
const emits = defineEmits(['form-closed'])
const tours = ref([]);
let showReg = ref(props.showRegistration)
const bookingFormTour = ref({});
const ALL TOURS_ENDPOINT = "/336b772d-3732-4e39-8dc8-386ed3686e70";
onMounted(async () => {
  const { data } = await toursApi.get(ALL TOURS_ENDPOINT);
  tours.value = data;
  console.log(tours.value);
});

watch(props, ()=>{
  showReg.value = props.showRegistration
})
function letSearch(query){
  tours.value = tours.value.find(item => item.content.country == query);
  console.log(tours.value)
}
function filterByPriceLow(){
  tours.value.sort((a, b)=>{return a.content.price - b.content.price});
}
function filterByPriceHight(){
  console.log(tours.value.sort((a, b)=>{return b.content.price - a.content.price}));
}
function filterByCountryLow(){
  tours.value.sort((a, b)=>{
    if(a.content.country < b.content.country)
      return -1
  }

```

```

    if(a.content.country > b.content.country)
      return 1
    return 0
  });
}
function filterByCountryHight(){
  tours.value.sort((a, b)=>{
    if(a.content.country < b.content.country)
      return 1
    if(a.content.country > b.content.country)
      return -1
    return 0
  });
}
function filterByDuraionLow(){
  tours.value.sort((a, b)=>{return a.content.duration - b.content.duration});
}
function filterByDurationHight(){
  tours.value.sort((a, b)=>{return b.content.duration - a.content.duration});
}
function getTours(tour) {
  bookingFormTour.value = tour;
}
function closeRegistration(){
  showReg.value = false;
  emits('form-closed')
}
let openForm = ref(false);
</script>

```

Це текст body з алгоритмами фільтрацій.

Уся інформація береться з сервісу створення фейкових API <https://mocki.io/fake-json-api>

Цей сервіс імітує видачу даних у форматі JSON від бази даних.

3.5 Випробування програмного забезпечення довідково-інформаційної системи

Програма та методика випробувань наведено в Додатку Г. На основі інструкцій, що наведені в цьому додатку ми проведемо випробування розробленого програмного забезпечення:

- 1) Авторизація.

Виконані встановлені початкові дії (коректні логін, пароль та настройки підключення, виділені користувачі видалені). Результат відповідає очікуваному (перехід до головного вікна програми). Проблем не виникло.

2) Редагування інформації про користувачів.

Виконані встановлені початкові дії (заповнив наступні поля: Прізвище та ім'я, Номер телефону, Логін). Результат відповідає очікуваному (новий клієнт з'явився в списку користувачів). Проблем не виникло.

3) Редагування інформації про розділи.

Виконані встановлені початкові дії (заповнив наступні поля: Назва розділу, Тип розділу, Завантажити зображення, Шаблон). Результат відповідає очікуваному (новий розділ з'явився в списку розподілів, виділені розділи видалені). Проблем не виникло.

Редагування інформації про новини.

Виконані встановлені початкові дії (заповнив поле Назва новини, Відобразити дату). Результат відповідає очікуваному (нова новина з'явилася в списку новин, виділені новини видалені). Проблем не виникло.

Пошук.

Виконані встановлені початкові дії. Результат відповідає очікуваному (результат пошуку відповідає встановленим критеріям). Проблем не виникло.

3.6. Результати апробації та перспективи розвитку проєкту

На підставі постановки завдання та технічного завдання, на попередньому етапі розробки було створено повністю робочу програму інформаційно-пошукової системи туристичного агенства. Програмне забезпечення було розроблене з урахуванням новітніх технологій, які значно зменшують навантаження на сервер, та покращують враження користувача від користування програмою. Програма повністю відповідає вимогам до функціональних характеристик, вимогам до організації вхідних та вихідних даних, вимогам до надійності ПЗ, та інших вимог, зазначених в технічному проєкті.

Програмне забезпечення бронювання туристичних послуг в режимі реального часу може бути завантажено з сайту дистриб'ютора або встановлено представником фірми дистриб'ютора. Розмір програми Для початку роботи з програмою, потрібно розпакувати архів в кореневу папку вашого майбутнього проєкту на веб-сервері. Файл `database.sql` в архіві відповідає за базу даних.

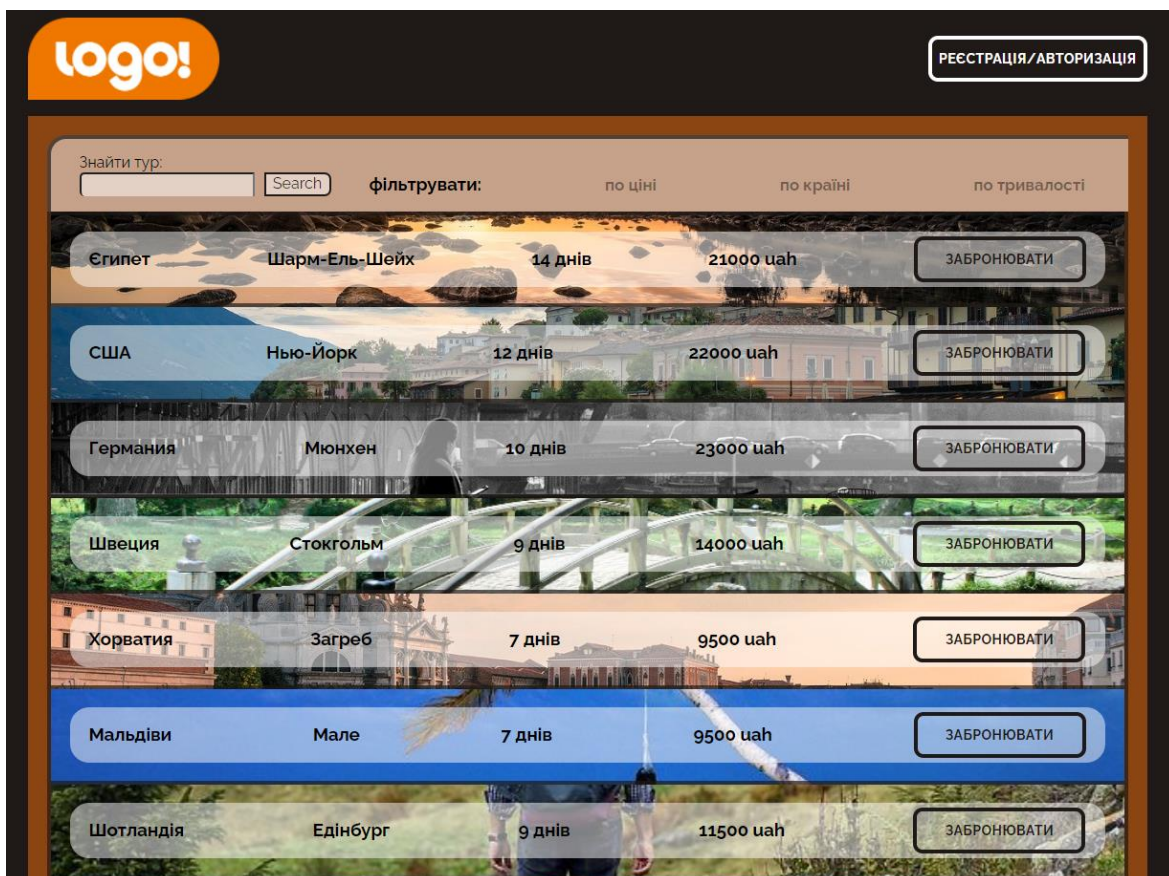
Для роботі програмного забезпечення, потрібні наступні програмні засоби:

- віртуальний сервер Node JS;
- сервер баз даних MongoDB

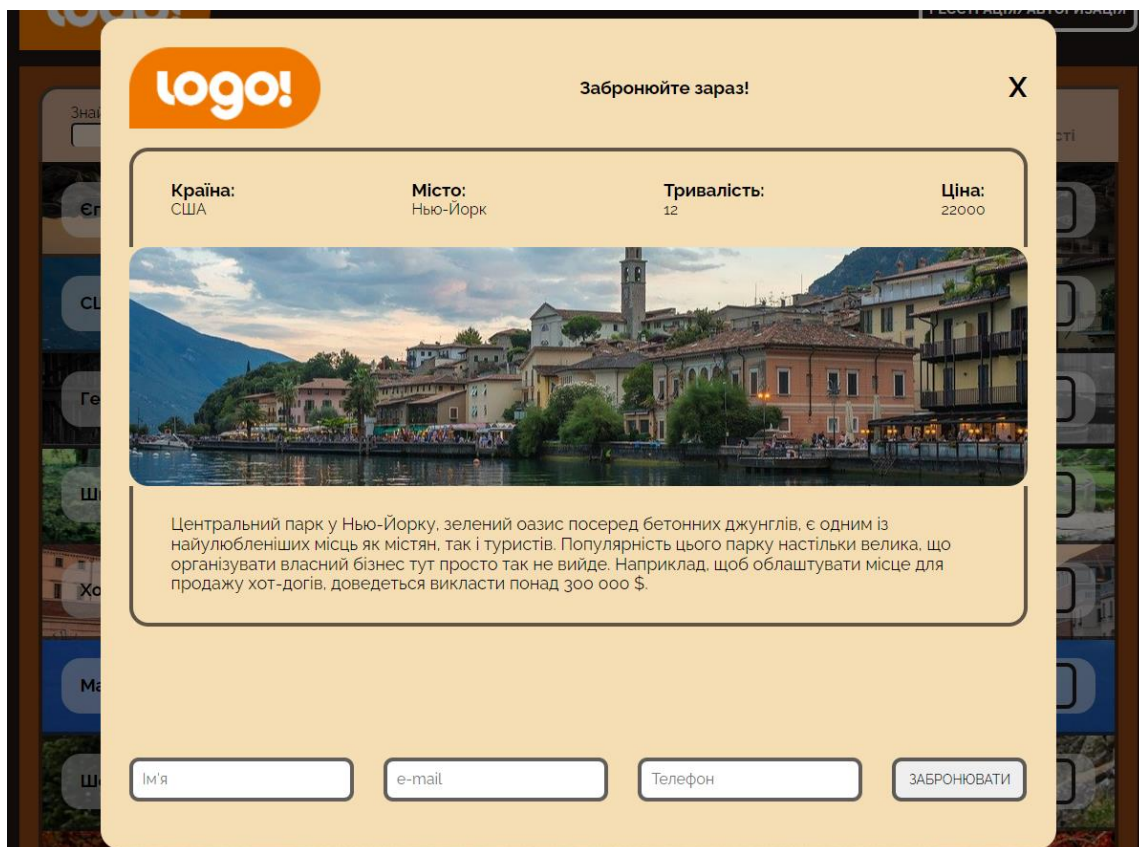
Для користування програмним забезпеченням, потрібен:

- доступ до глобальної мережі Internet, або доступ до видаленого ресурсу на якому встановлене ПЗ;
- web-браузер будь-якого виробника;

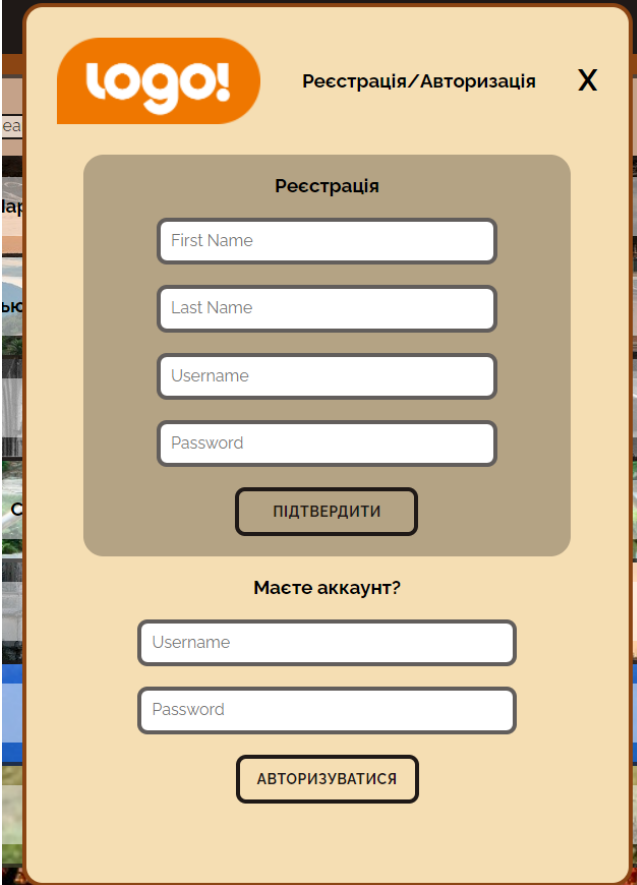
Для візуального представлення програмного забезпечення бронювання туристичних послуг в режимі реального часу, приведемо зображення інтерфейсу покупців та адміністратора на рисунках 3.1 – 3.4.



3.1. Головна сторінка (Дані що беруться з API формують рядки додатку. Їх кількість не обмежена, оформлення генерується автоматично)



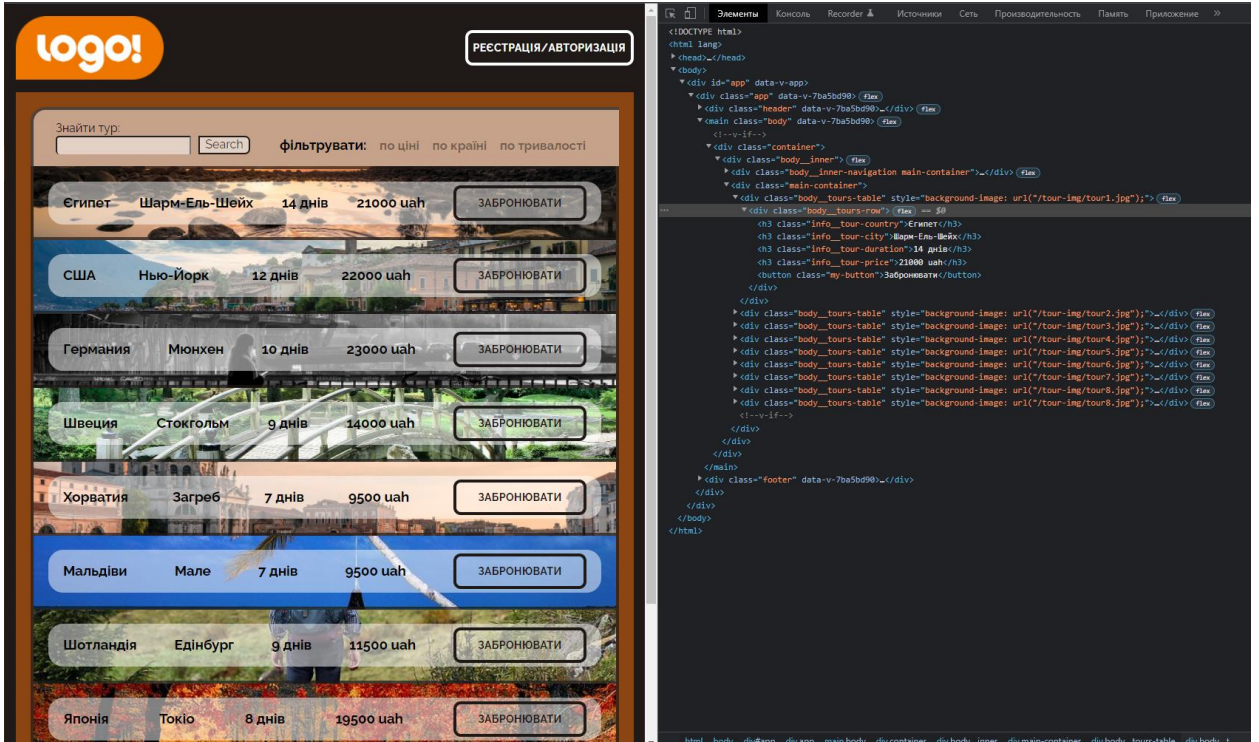
3.2. Після натискання кнопки відкривається модальне вікно з додатковою інформацією та полями для реєстрації.



The image shows a modal window titled "Реєстрація/Авторизація" (Registration/Authorization) with a close button (X) in the top right corner. The window features the "logo!" logo in the top left. The registration section, titled "Реєстрація", includes four input fields: "First Name", "Last Name", "Username", and "Password", followed by a "ПІДТВЕРДИТИ" (CONFIRM) button. Below this is a section titled "Маєте акаунт?" (Do you have an account?) with two input fields: "Username" and "Password", followed by an "АВТОРИЗУВАТИСЯ" (LOGIN) button.

3.3. Форма реєстрації та авторизації.

3.4. Згенерований код HTML з коду Vue, фрагмент



3.5. Фрагмент коду на Vue у компоненті The-Body

```
</div>
<div class="main-container">
  <div
    class="body__tours-table"
    :style="{ backgroundImage: `url(/tour-img/${tour.image}.jpg)` }"
    v-for="tour in tours"
    :key="tour.id"
  >
    <table-row
      :tour="tour"
      @showForm="
        getTours(tour);
        openForm = true;
      "
    />
  </div>
</div>
```

ВИСНОВКИ

В результаті виконання дипломної роботи було досягнуто поставлених цілей і виконано розробку програмного забезпечення довідково-інформаційної системи туристичного агенства, що дозволяє підвищити ефективність та якість надання туристичних послуг споживачу, скоротити час обслуговування клієнтів за рахунок автоматизації процесу надання туристичних послуг в режимі реального часу. Програмне забезпечення реалізоване у вигляді web-додатку та задовольняє вимогам, поставленим в технічному завданні і виконує наступні функції:

- 1) реєстрація в системі;
- 2) ідентифікація;
- 3) відображення, введення та корекцію інформації про тарифи, про наявні тури;
- 4) відображення, введення та корекцію інформації про клієнтів;
- 5) написання та корекцію інформації про замовлення, надання клієнту його примірника договору;
- б) обробка замовлень.

Програмне забезпечення може працювати під управлінням операційної системи Windows.

Тестування та випробування даного програмного продукту показало, що програмне забезпечення успішно справляється з поставленими перед ним задачами щодо бронювання туристичних послуг в режимі реального часу.

Програмне забезпечення розроблено мовою програмування JavaScript з використанням фреймворку Vue.JS та СУБД MongoDB.

Використання програмного забезпечення не потребує додаткового навчання завдяки зручному інтерфейсу.

В роботі розглянути питання з охорони праці та розроблені заходи щодо забезпечення сприятливих умов праці програміста.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- 1 Кифяк, В.Ф. Організація туристичної діяльності в Україні / В.Ф. Кифяк. – Ч.: Зелена Буковина, 2003. – 324 с.
- 2 Роглев, Х.Й. Основи готельного менеджменту / Х.Й. Роглев.– К.: Кондор, 2005. – 408с.
- 3 В. С. Домбровський, І. А. Сорочинська Актуальні проблеми використання комп'ютерних технологій у діяльності туристичних агенцій України [Електронний ресурс] – Режим доступу: <http://www.economy.nauka.com.ua/?op=1&z=2913>.
- 4 Бриггс, С. Маркетинг в туризмі / С. Бриггс. – К.: Знання –Прес, 2005. – 358 с.
- 5 . ГОСТ 19.102-77. Единая система программной документации. Стадии разработки // Единая система программной документации. – М.: Издательство стандартов, 1985. – С. 34-36.
- 6 Методичні вказівки до виконання кваліфікаційної роботи за напрямом підготовки 6.050103 – «Програмна інженерія» / С.Б. Приходько, С.В. Суслов, Л.О. Латанська, І.В. Устенко, Т.В. Пономаренко. – Миколаїв: НУК, 2012. – 50с.
- 7 ГОСТ 19.201-78. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. / М.: Издательство стандартов, 1987. – 2 с.
- 8 Закон України «Про охорону праці», від 14.10.1992р № 2694-ХІІ [Електронний ресурс] – Режим доступу: <http://zakon4.rada.gov.ua/laws/show/2694-12>
- 9 Офіційний гайд по vue.js <https://vuejs.org/tutorial/#step-1> (інтернет ресурс)
- 10 Документація по NodeJS <https://nodejs.org/uk/docs/> (інтернет ресурс)
- 11 Руководство JavaScript <https://developer.mozilla.org/ru/docs/Web/JavaScript/Guide> (інтернет ресурс)
- 12 CSS The Missing Manual David Sawyer McFarland
- 13 Шаблони JavaScript Стоян Стефанов

- 14 Системне та мережеве адміністрування Томас Лимончеллі, Крістіан Хоган і Страта Чейлап Розід 4, Розділ 7
- 15 Офіційний мануал MongoDB <https://www.mongodb.com/docs/manual/>
- 16 Інженерія програмного забезпечення Іан Коммервіл

ДОДАТКИ

Додаток А

Технічне завдання на розробку програмного забезпечення

1 Вступ

Коротке найменування програми: ПЗ «WEB tourist».

Повне найменування програми: програмне забезпечення довідково-інформаційної системи туристичного агенства.

Коротка характеристика галузі застосування: програмне забезпечення довідково-інформаційної системи туристичного агенства застосовується для автоматизації процесу надання туристичних послуг в режимі реального часу, що дозволить підвищити ефективності та якість надання туристичних послуг споживачу, скоротити час обслуговування клієнтів. Програмне забезпечення реалізоване у вигляді web-додатку.

2 Підстави для розробки

Розробка виконується на підставі завдання на дипломну роботу затверджену наказом ректора № ___ - від __.__.20__ р.

3 Призначення розробки

3.1 Функціональне призначення програми

Функціональне призначення розробки: зберігання, редагування, додавання, видалення та відправлення на поштові сервера впорядкованої інформації щодо послуг туристичного агенства.

3.2 Експлуатаційне призначення програми

Експлуатаційне призначення: програму можуть використовувати, як організації та підприємства, так и приватні підприємці, для реалізації туристичних послуг через мережу Internet.

4 Вимоги до програмного виробу

4.1 Вимоги до функціональних характеристик

4.1.1 Вимоги до складу виконуваних функцій

Програма повинна виконувати та мати функцію авторизації для розподілу прав доступу на: адміністратора та клієнтів (відвідувачів сайту тур. агенства).

З правами доступу користувача як адміністратора, програма повинна надати можливість виконувати наступні функції:

- 1) аутентифікація користувача;
- 2) управління населеними пунктами, регіонами та об'єктами (створення, редагування, видалення), список характеристик, meta-дані;
- 3) управління характеристиками кожної категорії;
- 4) управління списком об'єктів;
- 5) управління контентним розділами в необмеженій кількості (допомога на сайті);
- 6) управління новинами на сайті (коротке і повне утримання, зображення);
- 7) редагування будь-якої іншої інформації на сайті;
- 8) зручне редагування meta-інформації для успішного просування сайту;
- 9) редагування блоків тексту;
- 10) управління цінами на об'єкти розміщення;
- 11) управління акціями;

З правами доступу користувача як клієнта:

- 1) аутентифікація користувача;
- 2) реєстрація користувача;
- 3) переглянути повну інформацію по об'єкту: опис, фотографії, ціни;
- 4) ознайомитися з акціями, які проходять на даний момент;
- 5) пошук по об'єктах;
- 6) перехід до оформлення заявки на бронювання після вибору об'єкта;
- 7) сплатити за допомогою кредитної карти або готівкою.

4.1.2 Вимоги до організації вхідних та вихідних даних

Вхідні дані програми: вхідними даними є текстова інформація, що вводиться з клавіатури, яка за допомогою протоколу HTTP передає серверній частині програмного забезпечення.

Аутентифікація користувача:

Вхід: текстова інформація введена користувачем в поля форм E-mail: і Password:. Ця інформація повинна відповідати відповідним даними в БД.

Вихід: данні користувача та його ідентифікатор.

Пошук:

Вхід: текстова інформація введена користувачем в поле форми пошуку та можливих параметрів пошуку. Ця інформація перевіряється на наявність відповідності в базі даних.

Вихід: Список відповідних послуг.

Замовлення послуг:

Вхід: інформація від користувача у вигляді тексту набраного з клавіатури та данні послуги, також занесення інформації про замовлення до бази даних та відправка заявки.

Вихід: текстове повідомлення про успішне виконання замовлення, та його статус.

Оплата послуг:

Вхід: користувач може оплатити путівку готівкою або за допомогою картки, інформація надходить у вигляді тексту з клавіатури до бази даних, здійснюється оплата.

Вихід: текстове повідомлення про оплату.

Надання інформацію про наявні тури:

Вхід: робітник турагенства здійснює занесення інформації про наявні тури в базу даних.

Вихід: інформація про тури розміщаються на сайт.

Отримання заявки на купівлю туру:

Вхід: робітник турагенства отримує заявку від покупця.

Вихід: оформлення документів.

Отримати путівку:

Вхід: робітнику турагенства надходить інформація про оплату путівки.

Вихід: відправити документи путівки на електронну адресу або кур'єром.

Отримана інформація з пошуків та переглядів даних виводиться на сторінках браузера. При бажанні вона може бути скопійована та роздрукована на папері.

4.1.3 Вимоги до часових характеристик

Вимоги до часових характеристик програми не висуваються

4.2 Вимоги до надійності

Програмний продукт повинен нормально функціонувати при безперебійній роботі ПК та постійному підключенні до мережі Інтернет. При виникненні збоїв в роботі, відновлення нормальної роботи повинне проводитися після перезавантаження браузера.

Для забезпечення надійності інформації повинна використовуватися СУБД, що забезпечує цілісність транзакцій і несе відповідальність за цілісність інформації.

Система повинна продовжити коректно функціонувати при втраті частини інформації. У випадку неможливості продовження коректної роботи має бути повідомлено про це.

У разі введення користувачем некоректної інформації система повинна повідомити про помилку і надати можливість виправити її.

4.3 Вимоги до умов експлуатації

Необхідний рівень підготовки користувачів: мінімальні навички в користуванні комп'ютером та роботи з мережею Інтернет. Для експлуатації даного програмного забезпечення потрібен сучасний браузер, вихід до мережі Інтернет та всі необхідні вимоги для нормальної роботи ПК. Комп'ютер призначений для роботи в закритому опалювальному приміщенні при наступних умовах навколишнього середовища:

- температура навколишнього повітря від $+10^{\circ}\text{C}$ до $+35^{\circ}\text{C}$;
- атмосферний тиск від 630 до 800 мм ртутного стовпа;
- відносна вологість повітря не більше 80%;
- запиленість повітря не більше $0,75\text{ мг / м}^3$.

4.4 Вимоги до складу та параметрів технічних засобів

Даний програмний продукт потребує від комп'ютеру, на якому він буде встановлений, наступних характеристик, які треба розглядати:

Мінімальні системні вимоги

- процесор Intel Pentium G4400, AMD ATHLON 220 GE;

- 8 GB оперативної пам'яті;
- Мережевий адаптер з підключенням 1 Gbit;
- 120 GB SSD;
- ОС Microsoft Windows 10/ Server 2016 або вище, ОС Linux.

Рекомендовані системні вимоги

- процесор Intel Core I3 6100, AMD Ryzen 1600;
- 16 GB оперативної пам'яті з підтримкою ECC;
- Мережевий адаптер з підключенням 2,5 Gbit;
- 240 GB SSD;
- ОС Microsoft Windows 10/ Server 2016 або вище, ОС Linux.

Для роботи з даним програмним продуктом необхідна наявність сучасного браузер (Mozilla FireFox, Opera, Google Chrome, Internet Explorer) та можливість виходу в мережу Інтернет.

4.5 Вимоги до інформаційної та програмної сумісності

Даний програмний продукт призначений для розгортання на сервері, який має Node JS, СУБД MongoDB.

Мова розробки даної системи: JavaScript.

Технології: JavaScript, Vue JS, HTML, CSS.

4.6 Вимоги до маркування та упакування

Програма не повинна розповсюджуватися. Поширення даного продукту на фізичних носіях не передбачається.

4.7 Вимоги до транспортування та зберігання

Транспортування та зберігання не передбачаються у зв'язку з відсутністю фізичних носіїв.

5 Вимоги до програмної документації

Програмна документація повинна містити:

- технічне завдання;
- інструкція користувача;
- тести програмних модулів;
- програма та методика випробувань.

6 Техніко-економічні показники

Техніко-економічні показники для даного програмного продукту не розраховуються.

7 Стадії та етапи розробки

Стадії та етапи розробки представлені в таблиці А.1.

8 Порядок контролю та приймання

Прийомка та контроль програми повинні проводитися відповідно до узгоджених заздалегідь із замовником програми і методики випробувань.

Кожна стадія розробки повинна бути представлена в зазначені терміни і узгоджена з замовником (викладачем).

Хід проведення приймально-здавальних випробувань проводиться відповідно до програми і методики випробувань і документують за допомогою протоколу проведення випробувань.

На підставі протоколу проведення випробувань виконавець спільно з замовником підписують акт приймання-здачі програмного забезпечення в експлуатацію.

У разі знаходження помилок під час прийому програмного виробу складається акт про знайдені помилки, який підписується представниками замовника і розробника і затверджується керівниками організації - замовника та організації - розробника. Розробник повинен на протязі не більше ніж 2 тижнів виправити зазначені помилки, і оповістити замовника про повторне проведення перевірки.

Таблиця А.1 – Стадії та етапи розробки

Стадії розробки	Етапи робіт	Термін виконання робіт	
		початок етапу	кінець етапу
1. Технічне завдання	1.1 Обґрунтування необхідності розробки програми		
	1.2 Розробка технічного завдання		
	1.3 Затвердження технічного завдання		
2 Ескізний проект	2.1 Розробка ескізного проекту		
	2.2 Затвердження ескізного проекту		
3 Технічний	3.1 Розробка технічного проекту		

проект	3.2 Затвердження технічного проекту		
4 Робочий проект	4.1 Розробка програми		
	4.2 Розробка програмної документації		
	4.3 Випробування програми		

Додаток Б

Тексти програмних модулів

```

<template>
  <div class="app">
    <the-header @showRegistration="openRegistrationForm" />
    <the-body
      @form-closed="openReg = false"
      :showRegistration="openReg"/>
    <the-footer />
  </div>
</template>

<script setup>
import TheHeader from "@components/layout/TheHeader";
import TheBody from "@components/layout/TheBody";
import TheFooter from "@components/layout/TheFooter";
import { ref } from 'vue';
const openReg = ref(false);
function openRegistrationForm(){
openReg.value = true;
}

</script>

<style lang="scss" scoped>
.app{
  display: flex;
  min-height: 100vh;
  flex-direction: column;
}
</style>

```

Текст головного компоненту app. У нього будуть вмонтовані усі інші компоненти.

```

<template>
  <main class="body">
    <transition>
    <registration-form
      class="registration-form"
      v-if="showReg"
      @closeRegistration="closeRegistration"
    />
    </transition>
    <transition>
    <div class="container">
      <div class="body__inner" v-if="tours">
        <div class="body__inner-navigation main-container">

```

```

<search-element
@letSearch="letSearch"
class="inner__search-element"> </search-element>
<filters-row
v-if="tours"
@filterByPriceLow="filterByPriceLow"
@filterByPriceHight="filterByPriceHight"
@filterByCountryLow="filterByCountryLow"
@filterByCountryHight="filterByCountryHight"
@filterByDuraionLow="filterByDuraionLow"
@filterByDurationHight="filterByDurationHight"
class="inner__filters-row"> </filters-row>
</div>
<div class="main-container">
  <div
    class="body__tours-table"
    :style="{ backgroundImage: `url(/tour-img/${tour.image}.jpg)` }"
    v-for="tour in tours"
    :key="tour.id"
  >
    <table-row
      :tour="tour"
      @showForm="
        getTours(tour);
        openForm = true;
      "
    />
  </div>
  <transition>
    <booking-form v-if="openForm" @quitForm="openForm = false">
      <template v-slot:country>
        <div class="booking-form__characteristic">
          <h3>Країна:</h3>
          {{ bookingFormTour.content.country }}
        </div>
      </template>
      <template v-slot:city>
        <div class="booking-form__characteristic">
          <h3>Місто:</h3>
          {{ bookingFormTour.content.city }}
        </div>
      </template>
      <template v-slot:duration>
        <div class="booking-form__characteristic">
          <h3>Тривалість:</h3>
          {{ bookingFormTour.content.duration }}
        </div>
      </template>
      <template v-slot:price>
        <div class="booking-form__characteristic">
          <h3>Ціна:</h3>

```

```

        {{ bookingFormTour.content.price }}
      </div>
    </template>
    <template v-slot:description>
      <div class="booking-form__description">
        {{ bookingFormTour.content.description }}
      </div>
    </template>
    <template v-slot:image>
      <div class="booking-form__body-image"
        style="background-position: center;"
        :style="{ backgroundImage: `url(/tour-
img/${bookingFormTour.image}.jpg)`}" />
    </template>
  </booking-form>
</transition>
</div>
</div>
</div>
</transition>
</main>
</template>

<script setup>
import { toursApi } from "@api/tours";
import TableRow from "@components/TableRow";
import { onMounted, ref, defineProps, watch, defineEmits } from "vue";
import SearchElement from "@components/UI/SearchElement";
import FiltersRow from "@components/UI/FiltersRow";
import BookingForm from "@components/bookingForm/BookingForm";
import RegistrationForm from "../registrationForm/RegistrationForm.vue";
const props = defineProps({
  showRegistration:{
    type: Boolean,
    default: () => false
  }
})
const emits = defineEmits(['form-closed'])
const tours = ref([]);
let showReg = ref(props.showRegistration)
const bookingFormTour = ref({});
const ALL_TOURS_ENDPOINT = "/336b772d-3732-4e39-8dc8-386ed3686e70";
onMounted(async () => {
  const { data } = await toursApi.get(ALL_TOURS_ENDPOINT);
  tours.value = data;
  console.log(tours.value);
});

watch(props, ()=>{
  showReg.value = props.showRegistration
})

```

```

function letSearch(query){
  tours.value = tours.value.find(item => item.content.country == query);
  console.log(tours.value)
}
function filterByPriceLow(){
  tours.value.sort((a, b)=>{return a.content.price - b.content.price});
}
function filterByPriceHight(){
  console.log(tours.value.sort((a, b)=>{return b.content.price - a.content.price}));
}
function filterByCountryLow(){
  tours.value.sort((a, b)=>{
    if(a.content.country < b.content.country)
      return -1
    if(a.content.country > b.content.country)
      return 1
    return 0
  });
}
function filterByCountryHight(){
  tours.value.sort((a, b)=>{
    if(a.content.country < b.content.country)
      return 1
    if(a.content.country > b.content.country)
      return -1
    return 0
  });
}
function filterByDuraionLow(){
  tours.value.sort((a, b)=>{return a.content.duration - b.content.duration});
}
function filterByDurationHight(){
  tours.value.sort((a, b)=>{return b.content.duration - a.content.duration});
}
function getTours(tour) {
  bookingFormTour.value = tour;
}
function closeRegistration(){
  showReg.value = false;
  emits('form-closed')
}
let openForm = ref(false);
</script>

```

Додаток В

Програма та методики випробувань

1. Об'єкт випробувань

Назва об'єкту випробувань: програмне забезпечення довідково-інформаційної системи туристичного агенства.

Програмне забезпечення призначена для надання оперативної інформації щодо послуг туристичних агенства: для перегляду турів та бронювання путівок. Автоматизує проведення завантаження, розміщення, також редагування інформації на web-сервері, що значно скорочує затрати часу та зусиль.

2. Ціль випробувань

Перевірка придатності програми до використання за призначенням, її відповідність заданим вимогам і програмним документам. Перевірка реально досягнутих характеристик програми.

3. Вимоги до програми

3. 1 Вимоги до функціональних характеристик

Програма повинна виконувати наступні функції:

Користувацький відділ:

- аутентифікація користувача;
- реєстрація користувача;
- переглянути повну інформацію по об'єкту: опис, фотографії, ціни;
- ознайомитися з акціями, які проходять на даний момент;
- пошук по об'єктах;
- перехід до оформлення заявки на бронювання після вибору об'єкта;
- сплатити за допомогою кредитної карти або готівкою.

Адміністраторський розділ:

- аутентифікація користувача;
- управління населеними пунктами, регіонами та об'єктами (створення, редагування, видалення), список характеристик, meta-дані;
- управління характеристиками кожної категорії;

- управління списком об'єктів (готелів);
- управління готелями (створення, редагування, копіювання, видалення, блокування), категорія, опис, зображення, характеристики, ціна, meta-дані
- управління тематичними статтями (для інформування користувачів і покращення індексації сайту);
- управління контентним розділами в необмеженій кількості (допомога на сайті);
- управління новинами на сайті (коротке і повне утримання, зображення);
- редагування будь-якої іншої інформації на сайті;
- зручне редагування meta-інформації для успішного просування сайту;
- редагування блоків тексту;
- управління цінами на об'єкти розміщення;

4. Вимоги до програмної документації

У комплект програмної документації повинні входити такі документи:

- програма та методика випробувань;
- інструкція з експлуатації;
- текст програми;

5. Засоби та порядок випробувань

Вимоги до складу і параметрів технічних засобів:

- процесор Intel Pentium G4400, AMD ATHLON 220 GE;
- 8 GB оперативної пам'яті;
- Мережевий адаптер з підключенням 1 Gbit;
- 120 GB SSD;
- ОС Microsoft Windows 10/ Server 2016 або вище, ОС Linux.

Випробування включає в себе наступні етапи:

- 1) Перевірка відповідності вимогам до функціональних характеристик:
 - До складу виконуваних функцій;
 - до організації вхідних та вихідних даних.
- 2) Перевірка відповідності вимогам до надійності.

- 3) Перевірка відповідності вимогам до складу і параметрів технічних засобів.
- 4) Перевірка відповідності вимогам до інформаційної та програмної сумісності.
- 5) Перевірка відповідності вироблених дій зовнішнім ефектам.

6. Методи випробувань

Випробування необхідно проводити стратегією "чорного ящика". Тести з стратегії "чорного ящика" слід складати на основі припущення про помилку.

За рахунок дуже великої кількості функцій, які потрібно випробовувати, представлено тільки декілька результатів випробувань функцій програми. Всі функції програми без сумнівно пройшли випробування.

Необхідно виконати тестування стратегією "чорного ящика" методу CustomerLogin.

Метод CustomerLogin – здійснює перевірку логіна і пароля на відповідність з уже наявними користувачами.

Вхідні дані: логін і пароль користувача.

Тестування слід виконувати за наступною схемою:

- 1) Ввести правильний логін і пароль, система повинна аутентифікувати користувача, повернувши ідентифікатор клієнта;
- 2) Ввести неіснуючий логін і пароль, система повинна видати повідомлення про помилку: "Логін не вірний!";
- 3) Ввести існуючий login, порожній пароль, система повинна видати повідомлення про помилку: "Некоректний пароль!";
- 4) Ввести порожній login і існуючий пароль, система повинна видати повідомлення про помилку: "Логін не вірний!";

В таблиці В.1 наведено очікувані результати тестування функції CustomerLogin стратегією "чорного ящика".

Таблиця В.1 – Результати очікуваного тестування функції CustomerLogin стратегією "чорного ящика"

Вхідні дані	Очікуваний результат роботи процедури	Коментарії	Результати тесту
Введений login и пароль	Вдала аутентифікація	Запис знайдений в БД	Тест пройдено
Неіснуючий login і пароль	Повідомлення про помилку "Логін невірний!"	Запис не знайдений в БД	Тест пройдено
Існуючий login, пустий пароль	Повідомлення про помилку "Пароль невірний!"	В БД не знайдено точної відповідності	Тест пройдено
Пустий login, існуючий пароль	Повідомлення про помилку "Логін невірний!"	В БД не знайдено точної відповідності	Тест пройдено

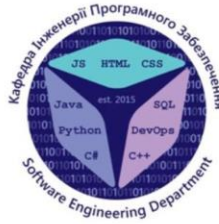
Додаток Г

Презентація



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка програмного забезпечення для довідково-інформаційної системи туристичного агенства

Виконав студент 5 курсу
групи ППЗ-51
Агапов А.О.
Керівник роботи
Негоденко О. В.

Київ – 2022

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- *Об'єкт дослідження* - підвищення ефективності та якості туристичних послуг для споживачів, зменшення часу обслуговування клієнтів туроператорами за допомогою програмного забезпечення довідково-інформаційної системи.
- *Предмет дослідження* - інформаційне забезпечення туристичної діяльності.
- *Мета роботи* – розробка програми довідково-інформаційної системи для туристичних агентств, заснованої на автоматизації процесу надання туристичних послуг споживачу туроператором у режиму real time.

АНАЛОГИ

Інформаційно-довідкова система	Авторизація	Бронювання туру	WEB інтерфейс	Безкоштовність	Автоматизація роботи туристичного агентства
Програмний комплекс „Мастер-Тур”	+	+	-	-	+
Програмний комплекс “Мастер-Web”	+	-	+	-	+
Туристична інформаційна система „TIC”	+	+	+	-	+
Програма “TIC Навігатор”	+	-	+	-	+
додаток “TurWin”	+	+	+	-	+

3

ТЕХНІЧНІ ЗАВДАННЯ

Програмне забезпечення має містити наступні функції:

- 1) Зареєструватися в системі;
- 2) Ідентифікація;
- 3) Відобразити, вводити та редагувати доступну інформацію про тур та вартість проїзду;
- 4) Відображення, введення та виправлення інформації про клієнта;
- 5) Відобразити та редагувати інформацію про бронювання турів, надавати замовнику копію договору;
- 6) Обробка замовлення

4

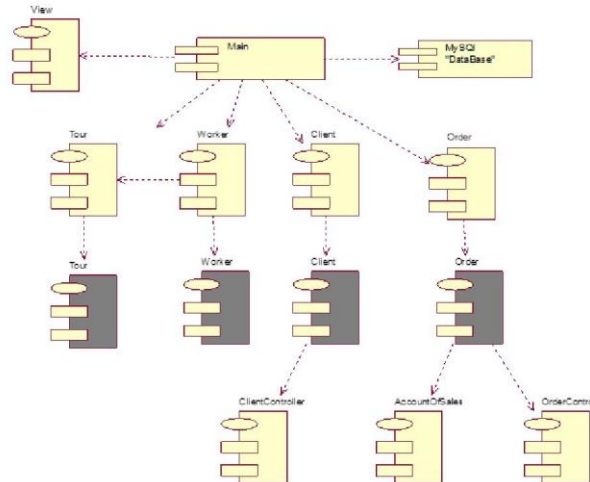
ПРОГРАМНІ ЗАСОБИ РЕАЛІЗАЦІЇ

Для розробки додатку довідково-інформаційної системи було вибрано такі засоби:

- Visual Studio Code для написання коду програми;
- Фреймворк Vue Js.

5

Діаграма компонентів для ПЗ довідково-інформаційно системи туристичного агенства



6

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- 1 Кифяк, В.Ф. Організація туристичної діяльності в Україні / В.Ф. Кифяк. – Ч.: Зелена Буковина, 2003. – 324 с.
- 2 Роглев, Х.Й. Основи готельного менеджменту / Х.Й. Роглев.– К.: Кондор, 2005. – 408с.
- 3 В. С. Домбровський, І. А. Сорочинська Актуальні проблеми вико-ристання комп'ютерних технологій у діяльності туристичних агенцій України [Електронний ресурс] – Режим доступу: <http://www.economy.nauka.com.ua/?op=1&z=2913>.
- 4 Бриггс, С. Маркетинг в туризмі / С. Бриггс. – К.: Знання –Прес, 2005. – 358 с.

7

ВИСНОВКИ

В результаті виконання дипломної роботи було досягнуто поставлених цілей і виконано розробку програмного забезпечення довідково-інформаційної системи туристичного агенства, що дозволяє підвищити ефективність та якість надання туристичних послуг споживачу, скоротити час обслуговування клієнтів за рахунок автоматизації процесу надання туристичних послуг в режимі ре-ального часу. Програмне забезпечення реалізоване у вигляді web-додатку та задовольняє вимогам, поставленим в технічному завданні і виконує наступні функції:

- 1) реєстрація в системі;
- 2) ідентифікація;
- 3) відображення, введення та корекцію інформації про тарифи, про наявні турів;
- 4) відображення, введення та корекцію інформації про клієнтів;
- 5) написання та корекцію інформації про замовлення, надання клієнту його примірника договору;
- 6) обробка замовлень.

Програмне забезпечення розроблено мовою програмування JavaScript з використанням фреймворку Vue.JS та СУБД MongoDB.

8

ДЯКУЮ ЗА УВАГУ!