

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

КВАЛІФІКАЦІЙНА РОБОТА

Тема: Веб-застосунок «Туристична агенція»

Ступінь вищої освіти – бакалавр
Спеціальність - 122 «Комп'ютерні науки»
Освітня програма «Комп'ютерні науки»

ПОЯСНЮВАЛЬНА ЗАПИСКА

Виконав: студент 4 курсу
групи КН-17/А
Стальмаков Д.Ф.
Керівник: Ст. викладач
Ігнатова Л.Б.

м. Київ – 2021 рік

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
«УНІВЕРСИТЕТ ЕКОНОМІКИ ТА ПРАВА «КРОК»

ЗАТВЕРДЖУЮ:

завідувач кафедри комп'ютерних наук

_____ О.С. Тимчук

«__» _____ 2021 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
Стальмакова Данила Федоровича

Тема роботи	Веб-застосунок «Туристична агенція»
Номер та дата наказу про затвердження теми	№29-1 від 29 березня 2021 р.
Коротка постановка завдання	Проектування веб-застосунку з базою даних для роботи туристичної агенції.
Посилання на джерела інформації (не більше п'яти найменувань, які рекомендує науковий керівник)	Герберт Шилдт. Java 7: The Complete Reference. Патрик Ноутон «Java 2. Наиболее полное руководство» Київ, 2007 г. Кей Хорстманн «Java 2.Тонкости программирования. Том 2» Київ, 2007
Вимоги до кваліфікаційної роботи	Кваліфікаційна робота має містити теоретичне, системотехнічне або експериментальне дослідження за темою роботи, яку слід розглядати як складне спеціалізоване завдання або практичну проблему в галузі комп'ютерних наук, яка характеризується комплексністю та невизначеністю умов і потребує застосування теорій і методів інформаційних технологій.
Термін представлення результатів перевірки рукопису на плагіат	до 24 травня 2021 р.
Термін представлення роботи до попереднього захисту	до 27 травня 2021 р.
Термін представлення роботи до захисту	до 2 червня 2021 р.

Дата видачі завдання 28 грудня 2020 р.

Керівник _____

Л.Б. Ігнатова

Здобувач освітнього ступеня бакалавра _____

Д.Ф. Стальмаков

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи, до	Примітка про виконання
Підготовчий етап			
1	Вибір напрямку дослідження	1.12.2020 р.	Виконано
2	Формування теми та призначення керівника	17.12.2020 р.	Виконано
3	Затвердження теми кваліфікаційної роботи	24.12.2020 р.	Виконано
4	Затвердження завдання на кваліфікаційну роботу	28.12.2020 р.	Виконано
Основний етап			
5	Розробка концепції кваліфікаційної роботи	18.02.2021 р.	Виконано
6	Підбір та вивчення джерел інформації з напрямку дослідження	4.03.2021 р.	Виконано
7	Затвердження розширеної постановки завдання	11.03.2021 р.	Виконано
8	Підготовка та подання керівнику розділу 1 кваліфікаційної роботи	25.03.2021 р.	Виконано
9	Проектування інформаційної системи	1.04.2021 р.	Виконано
10	Підготовка та подання керівнику розділу 2 кваліфікаційної роботи	15.04.2021 р.	Виконано
11	Реалізація інформаційної системи	6.05.2021 р.	Виконано
12	Підготовка та подання керівнику розділу 3 кваліфікаційної роботи	10.05.2021 р.	Виконано
13	Підготовка та подання керівнику першого варіанту всієї кваліфікаційної роботи	14.05.2021 р.	Виконано
14	Доопрацювання кваліфікаційної роботи з урахуванням зауважень керівника	17.05.2021 р.	Виконано
15	Представлення керівнику доопрацьованого варіанту кваліфікаційної роботи	20.05.2021 р.	Виконано
Завершальний етап			
16	Представлення рукопису для перевірки на плагіат	24.05.2021 р.	Виконано
17	Попередній захист кваліфікаційної роботи	27.05.2021 р.	Виконано
18	Підготовка супровідних документів	1.06.2021 р.	Виконано
19	Підготовка доповіді та презентації	1.06.2021 р.	Виконано
20	Захист кваліфікаційної роботи	Згідно з наказом по університету	Виконано

Керівник _____

Л.Б. Ігнатова

Здобувач освітнього ступеня бакалавра _____

Д.Ф. Стальмаков

Стальмаков Д.Ф. Веб-застосунок «Туристична агенція»

Пояснювальна записка кваліфікаційної роботи за спеціальністю 122 – Комп'ютерні науки (освітня програма - Комп'ютерні науки) СО Бакалавр. – ВНЗ «Університет економіки та права «КРОК», Навчально-науковий інститут інформаційних та комунікаційних технологій, кафедра комп'ютерних наук, Київ, 2021.

Програмний веб-застосунок туристичної агенції розроблений для виконання задач зв'язаних з доступом адміністратора(туроператора) до веб-ресурсу управління бронюванням авіаквитків та кімнат у готелях.

Ключові слова: веб-застосунок, програмний продукт, туристична агенція, розробка, база-даних.

Табл. 2. Рис. 34. Бібліограф.: 20 найм.

Stalmakov D. Web-application "Travel agency".

Project explanatory note by specialty 122 – Computer science. – «KROK» University, Educational and Scientific Institute of information and communication technologies, Department of Computer Science, Kyiv, 2021.

Web-application for Travel-agency for administrator(tour-operator) to manage tasks of flight bookings and hotel bookings.

Keywords: web-application, software, travel agency, development, database.

Tabl. 2. Fig. 34. Bibliography: 20 Items.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ ...	10
1.1 Проблеми розвитку інформаційних технологій в галузі туризму	10
1.2 Аналіз існуючих інформаційних технологій	11
1.3 Висновки по розділу	14
РОЗДІЛ 2. ПРОЕКТУВАННЯ ЗАСТОСУНКУ	15
2.1 Аналіз варіантів використання	15
2.2 Моделювання процесів.....	17
2.3 Моделювання даних	21
2.4 Проектування інтерфейсу	23
2.5 Проектування структури	24
2.6 Висновки по розділу	29
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЗАСТОСУНКУ «ТУРИСТИЧНОЇ АГЕНЦІЇ»	30
3.1 Особливості реалізації застосунку	30
3.2 Конструювання застосунку.....	31
3.3 Документування застосунку	34
3.4 Адміністрування застосунку.....	42
3.5 Тестування застосунку	45
3.6 Підготовка застосунку до оформлення права інтелектуальної власності відповідно до національного законодавства	50
3.7 Висновки по розділу	51
ВИСНОВКИ.....	52
СПИСОК ПОСИЛАНЬ	53
ДОДАТОК А. ЕКРАНИ ЗАСТОСУНКУ ТУРИСТИЧНОЇ АГЕНЦІЇ	56
ДОДАТОК Б. ФРАГМЕНТИ ЛІСТИНГУ	57
ДОДАТОК В. АНАЛІЗ КОДУ ЗАСТОСУНКУ «ТУРИСТИЧНОЇ АГЕНЦІЇ» НА ЗАПОЗИЧЕННЯ	61
ДОДАТОК Г. ЗАЯВА ДЛЯ ОФОРМЛЕННЯ ПРАВА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ НА ЗАСТОСУНОК ТУРИСТИЧНОЇ АГЕНЦІЇ.....	64

ВСТУП

Актуальність теми роботи. Бурхливий ріст і розвиток сучасних технологій потребують нових рішень для вирішення задач туристичних агенцій, які потребують більшої швидкості та мобільності у роботі підприємства. Сталі програмні продукти на ринку туризму не мають належного вирішення цих питань. Системи управління даними розростаються все більшим об'ємом даних які значно сповільнюють роботу відповідного застосунку. Системи управління даними користуються все більшим попитом і для малих підприємств, які не мають запасного часу на вирішення питань зв'язаних з повільним програмним забезпеченням.

Програмне управління даними прискорює роботу з менеджментом даних клієнтів. Автоматизація вирішення сервісних проблем забезпечує зручність доступу та перегляду веб-ресурсів дозволяють користувачу швидко і без проблем знайти потрібний тур та забронювати квитки та номери готелю.

На сучасному туристичному ринку відбувається активний розвиток з позитивними тенденціями. Так, Всесвітня туристична організація опублікувала данні, згідно яких, кількість міжнародних туристичних прибуттів у 2019 році досягла близько 1,5 мільярда, що на 4%, або на 54 мільйони перевищує показники 2018 року, та є десятим поспіль роком з позитивними тенденціями. Розширення ринку у різних галузях та міжнародних контактів потребують вирішення питань у багатьох сферах економіки, і в головному, в галузі туризму і активного відпочинку, особливо для країн економіка яких зав'язана на туризмі. Світовий досвід вказує на те, що глобалізаційні процеси в багатьох країнах стали об'єктивним чинником прогресу в галузі інформаційних технологій, котрі використовуються суб'єктами туризму.

Інформаційні технології сучасності дозволяють застосовувати всі переваги мережі інтернет та комунікаційних зв'язків для використання у тому числі й у сфері туризму які забезпечують високу якість автоматизації

отримання сервісних послуг: швидке і зручне бронювання квитків на транспорт, бронювання номерів у готелі, прокат автомобілів та мобільний транспорт, обмін валют, замовлення квитків на вистави та спортивні заходи тощо. Найбільшими глобальними розподільчими системами на міжнародному ринку туристичних послуг є AMADEUS, Worldspan, Galileo, Sabre. Що займає більше 90% ринку, 10% займають регіональні системи бронювання і управління та системи, які знаходяться у стадії злиття з однією з вищеперерахованих.

Сучасні інформаційні технології відкривають безліч можливостей. Так є проблема для менших туристичних фірм, як нестача власних можливостей програмного забезпечення щодо тонкощів маніпулювання даних. Тому для вирішення таких проблем, використовується індивідуальний план щодо розробки програмного забезпечення під конкретний випадок. Програмні продукти з індивідуальним підходом, забезпечують вирішення всіх необхідних задач, є мобільними та дешевшими за усталені продукти на ринку інформаційних технологій.

Мета і завдання дослідження. Розробка системи управління туристичним бізнесом: авторизація, інформаційна складова та загальне управління туристичним бізнес-процесом. Мета проекту полягає у створенні веб-застосунку та відповідної бази даних для туристичної агенції. Програмний продукт є інструментом для ілюстрованої демонстрації доступних турів та менеджментом бронювання авіаквитків і номерів в готелях. Для досягнення мети сформульовані такі задачі:

- Провести аналіз існуючих веб-застосунків відповідної галузі, виокремити необхідний функціонал для забезпечення високоефективної роботи застосунку що розробляється.
- Розробити сучасний привабливий дизайн веб-застосунку.
- Підібрати оптимальний набір необхідних бібліотек і фреймворків для реалізації розробки веб-застосунку.
- Розробити структуру бази даних, створити відповідні таблиці.

- Розробити серверну частину щодо маніпуляції з даними в базі даних.
- Розробити підсистеми авторизації, обслуговування клієнта, адміністратора/менеджера.

Об'єкт дослідження.

Об'єктом дослідження є процеси аналізу, проектування, розробки та супроводження програмного забезпечення туристичної агенції.

Предмет дослідження.

Предметом дослідження є процеси аналізу, проектування та реалізації веб-застосунку «Туристична агенція».

Методи дослідження.

Для вирішення поставлених задач були використані технології та методи проектування, сучасні технології та платформи програмування, методи об'єктно-орієнтованого проектування та програмування та методи реалізації архітектури клієнт-сервер.

Практична цінність дослідження.

В процесі експлуатації веб-застосунку, користувач зможе показувати необхідні тури та маніпулювати даними клієнтів в базі даних.

Структура та обсяг роботи.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, списку посилань з 20 найменувань та додатків. Робота містить 34 рисунків, 2 таблиць. Загальний обсяг роботи складає 60 сторінок, основний зміст викладено на 50 сторінках.

У вступі розглянуто проблеми розвитку інформаційних технологій у сферах туризму, сформульовані мета і задачі проекту, практичне значення отриманих результатів.

У першому розділі виконано огляд і аналіз сучасного стану інформаційних технологій у сферах туризму. Проаналізовані існуючі

програмні рішення та їх недоліки. Сформульовано постановку задачі даного проекту.

У другому розділі розроблено моделі та діаграми логіки програми та візуально продемонстровано на структурному та архітектурних рівнях можливість користувачів взаємодіяти зі застосунком.

У третьому розділі описано конструювання та реалізацію застосунку. Описані використані технології клієнтської, серверної частини, бази даних та інформація щодо розробки програмного продукту. Веб-застосунок має клієнт-серверну архітектуру. Програмна реалізація системи представлена у вигляді набору класів, які мають чітку взаємодію.

У додатках приведено лістинги та екрани програми, які представлені фрагментарно.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Проблеми розвитку інформаційних технологій в галузі туризму

Сьогоденний розвиток програмних рішень та впровадження інформаційних технологій в щоденне користування кожної сфери туризму, дозволяють впровадити зміни інформаційних технологій туристичної галузі з такої, яка направлена на вирішення задач об'ємних та сформованих на ринку компаній на індивідуальні туристичні фірми.

Десктопне програмне забезпечення та веб-застосунки існують вже з моменту появи мережі інтернет, і потребують своєчасного оновлення. Головною задачею інформаційних засобів та веб-застосунків у галузі туризму та відпочинку є надання користувачам інструментарію для взаємодії на разі потреби один з одним, обмінюватися інформацією, проводити спільні операції та керувати масивами даних використовуючи сучасні методи.

Головними чинниками у розвитку інформаційних технологій у сфері туризму є вперш за все фактори мобільності отримання інформації щодо продуктів ринку туризму. Індивідуальний підхід, як до туроператорів, так і до клієнтів туристичних фірм та організацій

Розробнику потрібно передбачити внесення в програмне забезпечення даних, які будуть у подальшому наразі потреби корегуватися розробниками програмного забезпечення для поновлення даних у відповідності з оновленням плану туристичної фірми та зможуть забезпечити чітку та зрозумілу альтернативу стандартним та застарілим носіям даних. Як наприклад:

- назви турів;
- данні для авторизації;
- картинки;
- інша важлива інформація.

1.2 Аналіз існуючих інформаційних технологій

Сфера туризму як і всі інші сфери поступово переходить до використання можливостей інформаційних технологій. Розглянемо декілька прикладів сайтів з можливістю знаходити та бронювати тури, номери в готелях та авіаквитки. Наприклад:

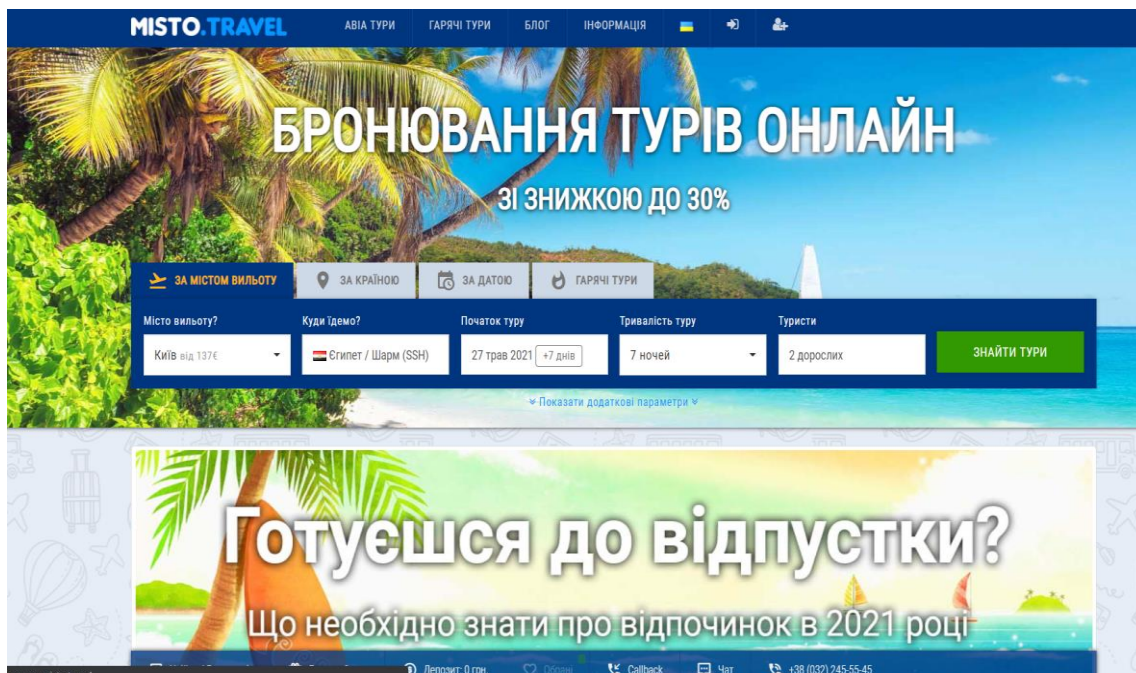


Рисунок 1.1 – вигляд інтернет-сайту «Mistotravel»

Джерело: Розроблено автором

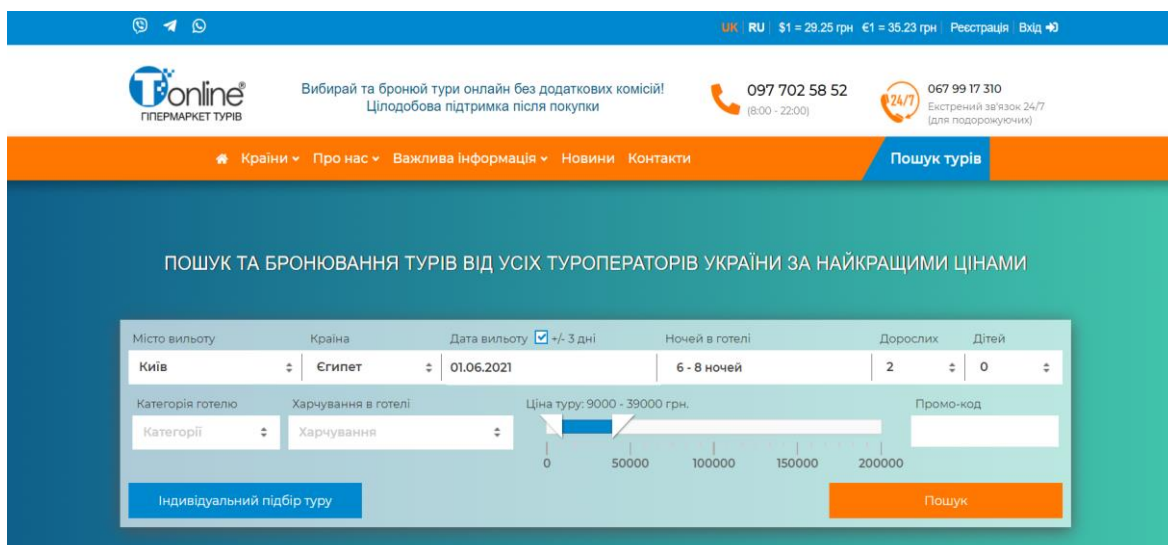


Рисунок 1.2 – вигляд інтернет-сайту «Tour-online»

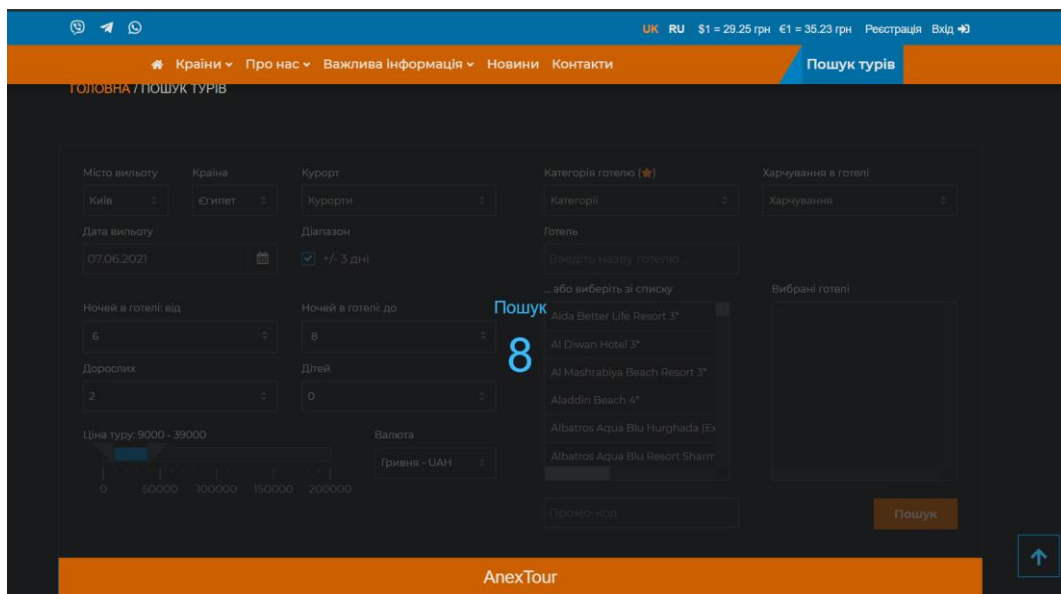


Рисунок 1.3 – вивантаження на сайті «Tour-online» даних з бази даних готелів

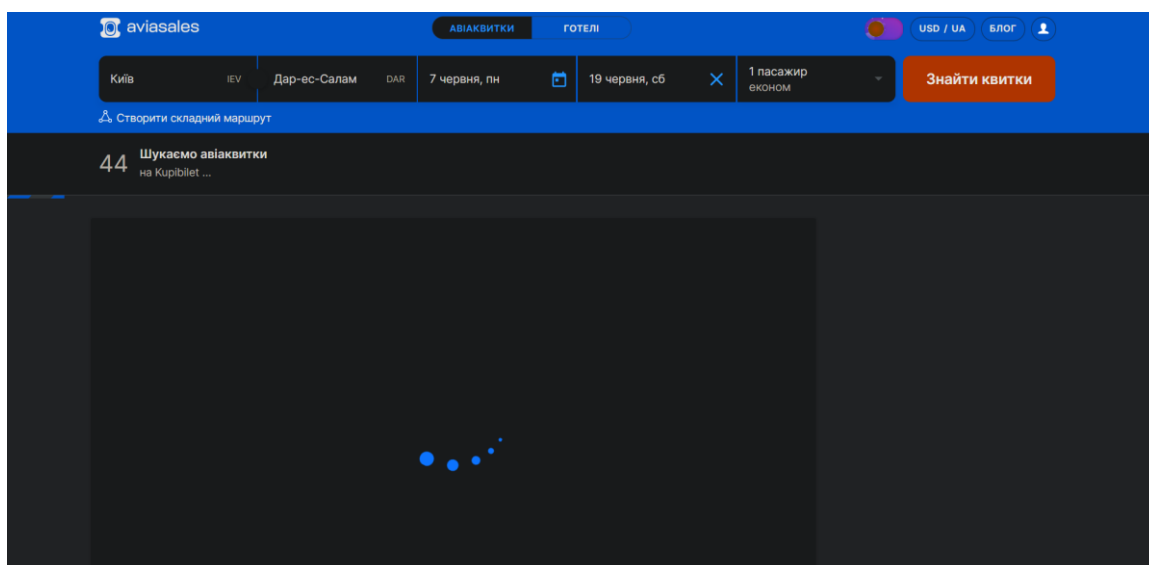


Рисунок 1.4 – вивантаження на сайті «Aviasales» даних з бази даних авіаквитків

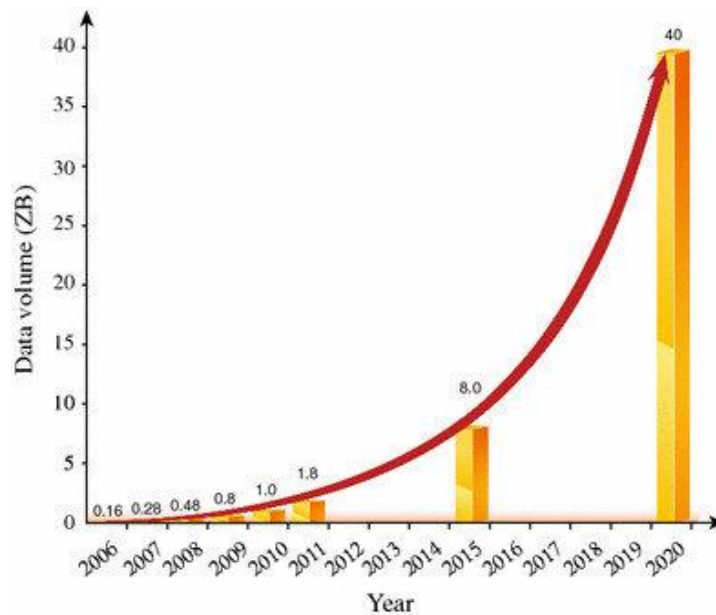


Рисунок 1.5 – глобальна тенденція росту об'єму даних

Популярні і давно сталі програмні продукти мають занадто велику кількість даних, тому як ми бачимо на Рис.1.3-1.4 – вивантаження даних готелів та авіаквитків з бази даних займають що найменше 8 секунд, які згодом розростуться та які будуть потребувати ще більшої кількості ресурсів для підтримки застосунку і бази даних. Туристичним організаціям які тільки вийшли на ринок може не вистачити ресурсів для управління подібними програмними продуктами якщо в них обмежені стартові можливості. Інша проблема з якою можуть зіткнутися туристичні організації – замала швидкість розгортання програмного забезпечення і його експлуатація.

Вже існуючі рішення програмних конструкторів, такі як «Wordpress» чи «Joomla», потребують від користувачів більшої кваліфікації для розробки і обслуговування програмних продуктів ніж використання готового веб-застосунку.

Створення програмних продуктів з використанням сучасних фреймворків та бібліотек гарантують подальшу підтримку розробників цих технологій і можливості для вирішення сучасних проблем.

Майже всі існуючі веб-ресурси мають frontend інтерфейс з інтерактивними сторінками які написані на мовах програмування та

технологіях: HTML, CSS, JS та фреймворки, та з відповідним backend мовою програмування, наприклад такі як: Java, C#, Python, Node.js, Scala, Kotlin, Ruby, PHP.

1.3 Висновки по розділу

Аналіз міжнародного туризму та сучасних, вже готових для використання рішень показав низьку мобільність та великий об'єм даних які не потрібні типовому туристичному оператору який тільки вийшов на ринок надання послуг у сфері туризму. Сучасні сталі на ринку компанії мають великий обсяг даних які не дозволяють швидко та зручно встановлювати та експлуатувати програмний продукт.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ЗАСТОСУНКУ

2.1 Аналіз варіантів використання

Функціональні вимоги:

- реєстрація адміністратора. Реєстрація адміністратора застосунку має здійснюватися через серверний код, за для безпеки даних;
- авторизація адміністратора застосунку;
- додавання необхідних турів та їх інформацію на сайт;
- бронювання авіаквитків та додавання їх у БД;
- редагування інформації про авіаквитки у БД;
- видалення авіаквитків з БД;
- бронювання номерів у готелях в БД;
- редагування інформації про бронювання номерів у готелях в БД;
- видалення даних про бронювання номерів в готелях з БД;
- безпечне використання через фільтрацію.

Нефункціональні вимоги:

- простота в оформленні;
- зручне переміщення між сторінками застосунку;
- зрозумілий зовнішній вигляд;
- простота в оформленні;
- візуальний мінімалізм;
- зрозумілий дизайн;
- тільки необхідний інструментарій.

Проектування роботи програми у випадку прямої взаємодії клієнта з туроператором та демонстрації останнім пропозиції туристичної агенції, показано, як приклад на діаграмі варіантів використання (рис. 2.2)

Діаграма прецедентів (рис. 2.1.) дозволяє побачити головні можливості адміністратора і користувача у взаємодії з програмним забезпеченням.

Користувач має право взаємодіяти з наявними туристичної агенції турами та не маючи доступу створювати облікові записи які будуть у подальшому використовуватися для управління бронювання.

Адміністратор в свою чергу може взаємодіяти з турами, має можливість для авторизації, додавання та редагування даних в базі даних, взаємодіяти з обліковими записами та також створювати облікові записи.

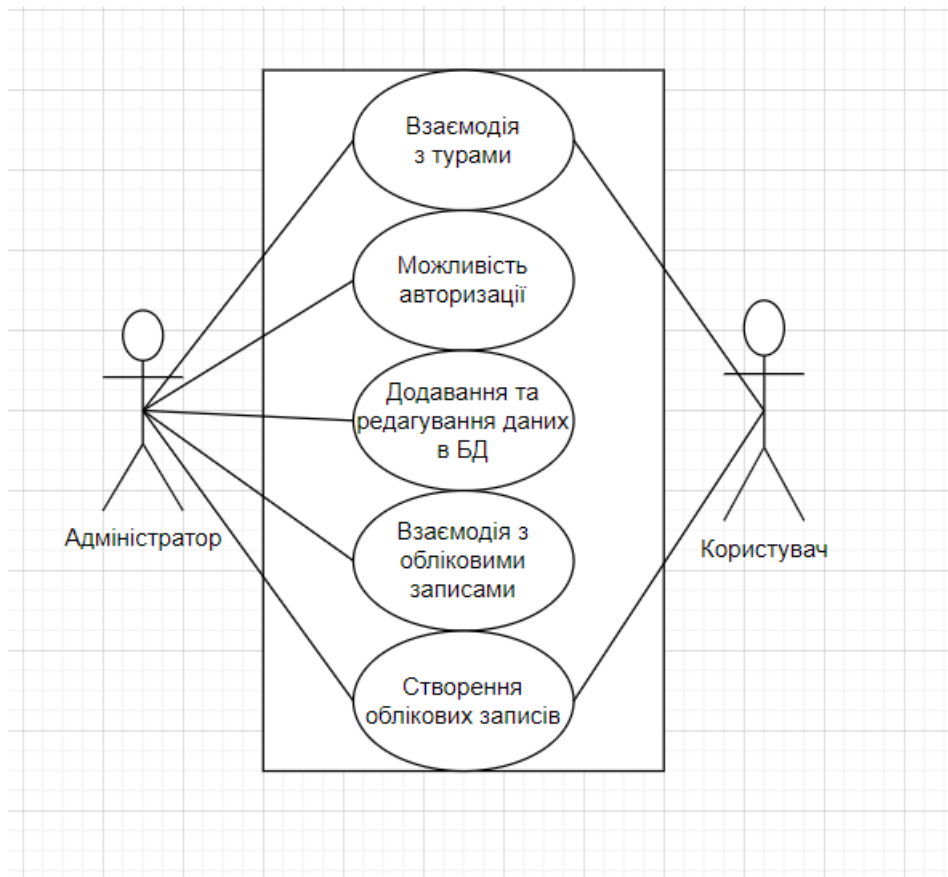


Рисунок 2.1 – Діаграма прецедентів web-застосунку у випадку взаємодії з клієнтом

Джерело: Розроблено автором

2.2 Моделювання процесів

В даному web-застосунку є декілька можливостей взаємодії користувачів з програмним засобом:

- взаємодія туроператора з клієнтом на пряму і показ останньому пропозицій;
- взаємодія адміністратора з програмним забезпеченням і базою даних;

Щоб продемонструвати роботу застосунку зі взаємодією з клієнтом на пряму та подальшу роботу адміністратора щодо бронювання номерів готелів та авіаквитків, можемо подивитися діаграмі діяльності (activity diagram) (рис. 2.2). На якій ми можемо побачити запуск застосунку користувачем і отримання інформації від клієнта про побажання стосовно турів. Наступним кроком адміністратор надає клієнту інформацію та показує тур. Після чого адміністратор приймає данні клієнта і виконує авторизацію для подальшого бронювання номерів готелів та авіаквитків.

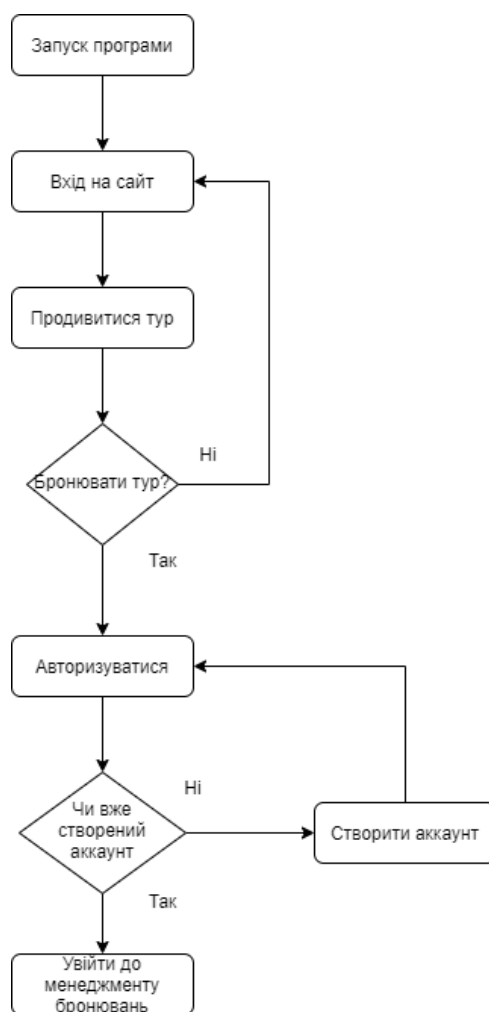


Рисунок 2.2 – діаграма початкової діяльності web-застосунку у випадку взаємодії з клієнтом

Джерело: Розроблено автором

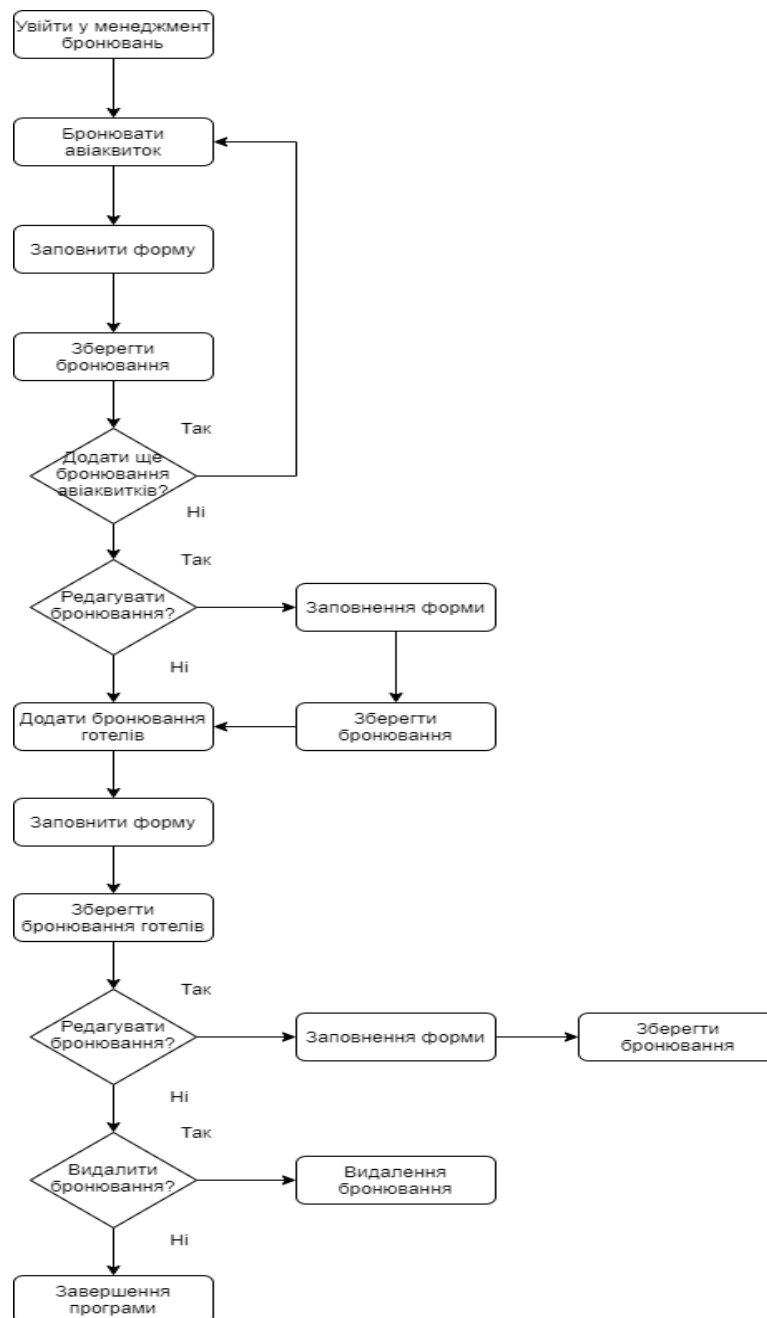


Рисунок 2.2 – діаграма діяльності застосунку щодо управління бронюванням

Джерело: Розроблено автором

Після успішного входу на сторінки управління бронюванням, користувач бронює авіаквиток та відповідно заповнює форму бронювання і натискає кнопку «Зберегти», після чого може додати ще бронювання авіаквитку чи редагувати вже існуючі бронювання.

Якщо користувач погоджується, то заповнює форму для редагування необхідного бронювання та зберігає бронювання. Після чого користувач може забронювати номери готелю відповідно до авіаквитка. Користувач переходить на сторінку бронювання номерів готелів і бронює використовуючи відповідну форму. В подальшому користувач може або редагувати або видалити бронювання.

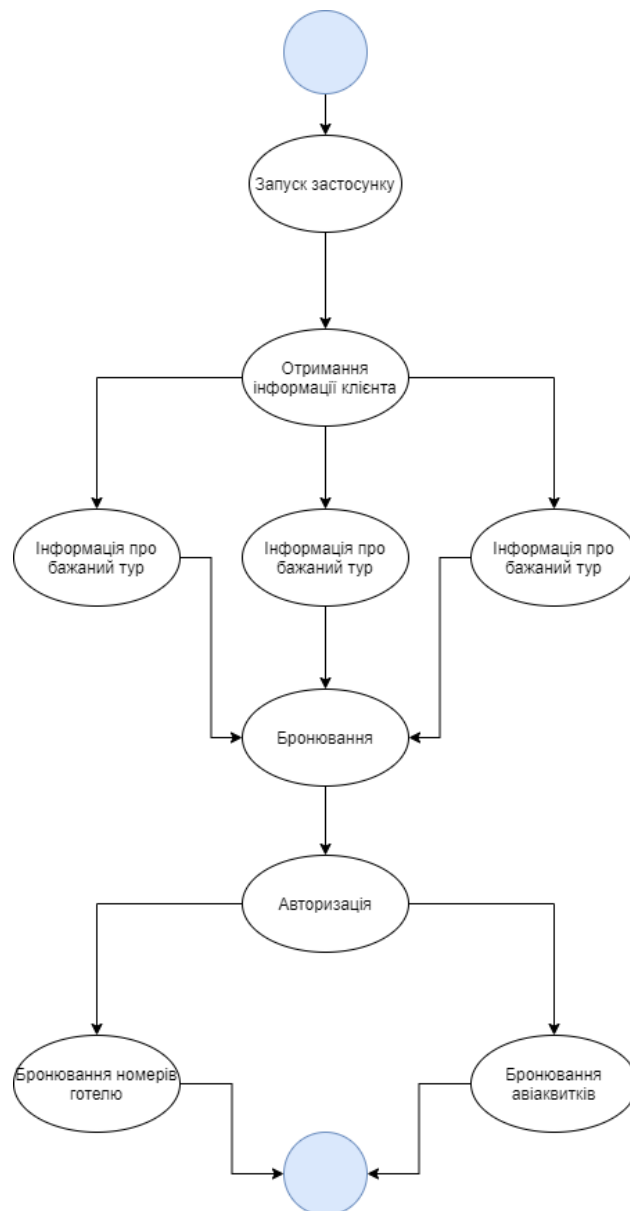


Рисунок 2.2 – діаграма діяльності web-застосунку у випадку взаємодії з клієнтом

Джерело: Розроблено автором

На діаграмі у випадку взаємодії з клієнтом показані загальні кроки щодо роботи застосунку у випадку взаємодії з клієнтом.

Клієнт отримає інформацію від туристичного оператора в залежності від бажання щодо подорожі. Адміністратор в свою чергу демонструє ті варіанти які є в наявності та після чого заходить на сторінки управління менеджментом та здійснює бронювання.

Після успішного бронювання, адміністратор може повернутися до демонстрації турів клієнту та повторити операцію знову.

2.3 Моделювання даних

Моделювання даних необхідно для розуміння діяльності застосунку в цілому. Виконання логіки веб-застосунку можливо побачити на діаграмі взаємодії(рис. 2.3.).

Робота користувача може мати наступну послідовність дій, а саме:

- отримує доступ та ініціює можливість взаємодії з ПЗ;
- користувач потрапляє на стартову сторінку;
- користувач переходить на сторінку туристичних пропозицій;
- користувач передивляється тур;
- є можливість переходу на сторінку авторизації;
- користувач отримує необхідні результати;

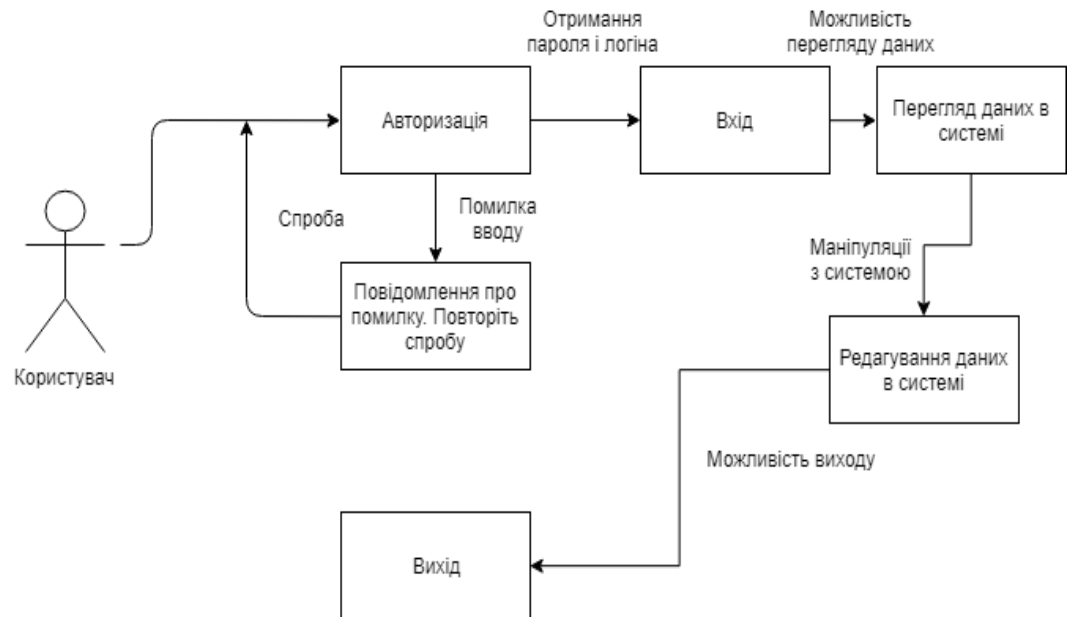


Рисунок 2.3 – діаграма взаємодії користувача з програмою

Джерело: Розроблено автором

Адміністратор має повний набір інструментарію щодо роботи з базою даних та взаємодією.

Робота користувача може мати наступну послідовність дій:

- адміністратор виконує запуск програмного забезпечення на сервері;
- адміністратор заходить на стартову сторінку застосунку;
- влаштовується взаємодія з турами на головній сторінці;
- влаштовується реєстрація у системі адміністратора за необхідністю;
- якщо адміністратор вже є аккаунт у базі даних, влаштовується авторизація в системі;
- програма додає, видаляє чи редагує бронювання авіаквитків за запитом адміністратора;
- програма додає, видаляє чи редагує бронювання номерів в готелі за запитом адміністратора;

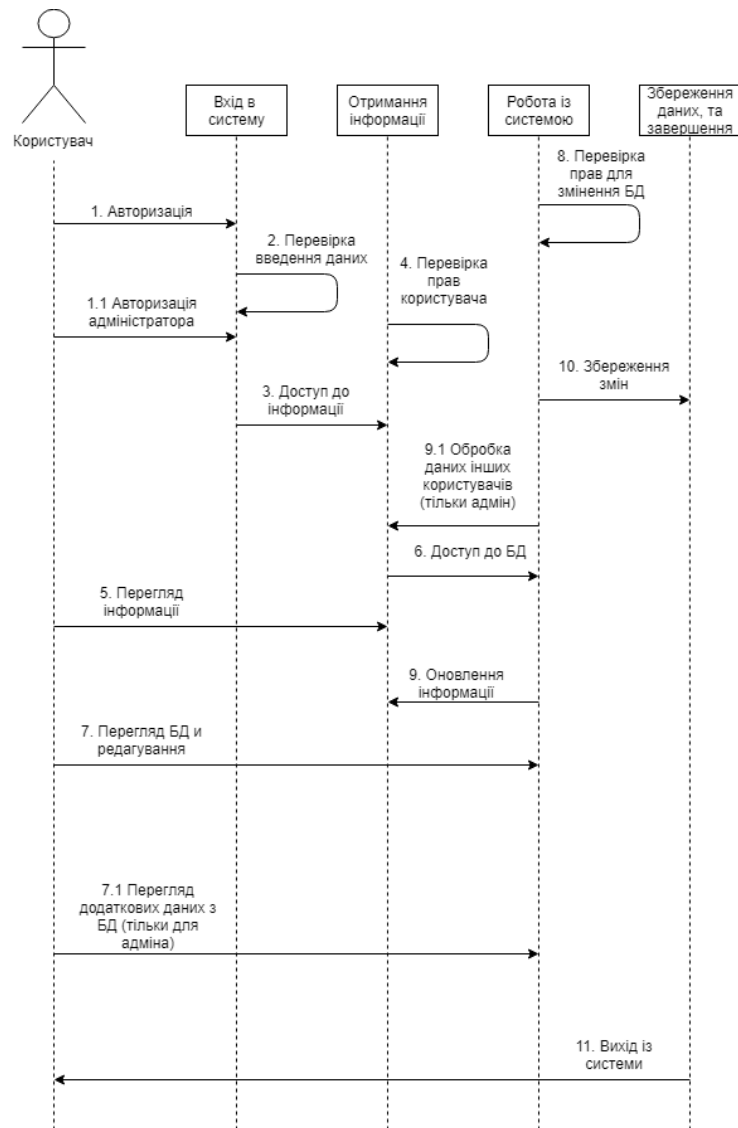


Рисунок 2.4 – діаграма взаємодії адміністратора з програмою
Джерело: Розроблено автором

2.4 Проектування інтерфейсу

При проектуванні логіки взаємодії користувача з програмним застосунком, приділяють увагу розробці вайрфрейму, який дозволяє розставити акценти різних елементів інтерфейсу та використовуючи макет розробити клієнтські сторінки.

Образ дизайну низької точності потрібен для надання структурної схеми сторінок, які будуть допомагати розробникам клієнтських сторінок у приблизному розташуванні кнопок, картинок, тощо.



Рисунок 2.5 – wireframe головної сторінки

Джерело: Розроблено автором

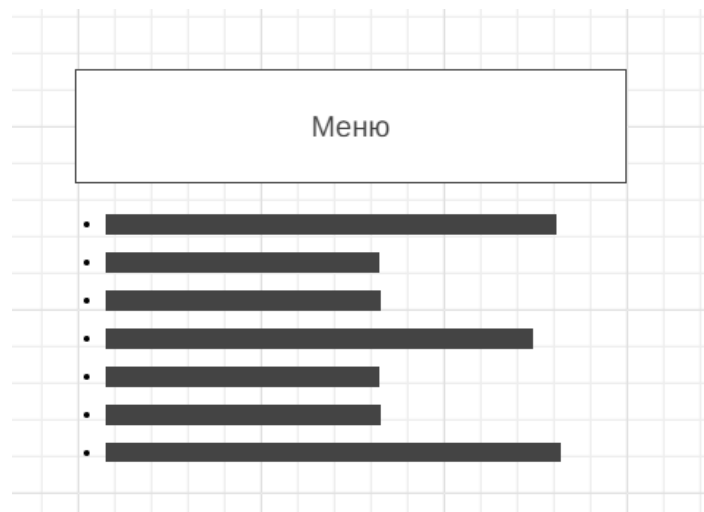


Рисунок 2.6 – wireframe сторінок взаємодії з БД

Джерело: Розроблено автором

2.5 Проектування структури

Однією з найважливіших завдань при проектуванні логіки програми є розробка діаграм навігації по застосунку (рис. 2.7). Діаграма дозволяє прослідити етапи взаємодії контролера та його логіки.

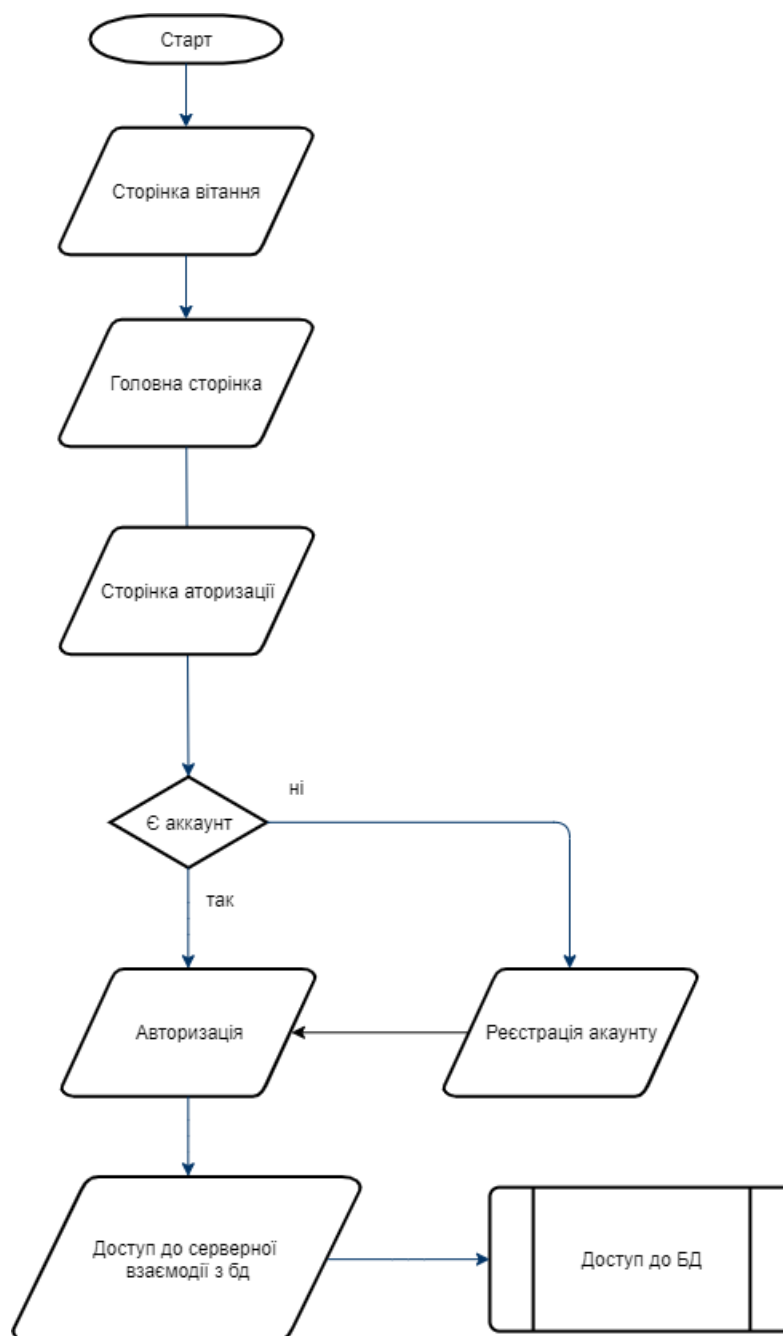


Рисунок 2.7 - Діаграма завдань

Джерело: Розроблено автором

Для розробки веб-застосунку був підібраний оптимальний набір бібліотек та фреймворків для подальшої роботи над програмним забезпеченням.

Для розробки серверної частини програмного продукту був використаний Spring Framework з його модулями які надають широкий спектр застосувань.

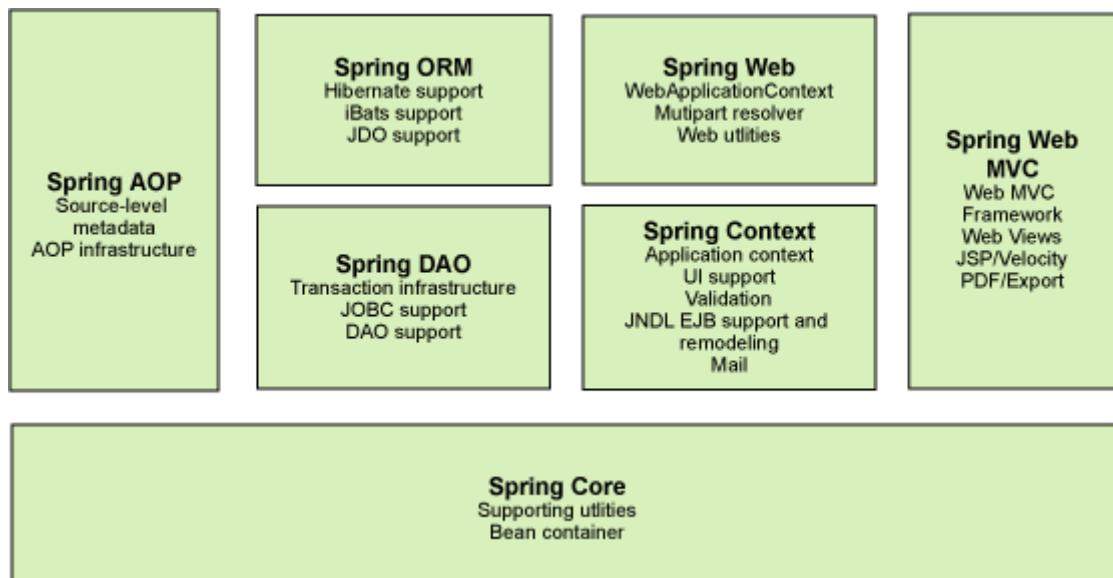


Рисунок 2.8 – Модулі Spring Framework

Джерело: Розроблено автором

Були використані сучасні можливості бібліотек та фреймворків для розробки веб-застосунку. Розробка серверної частини розроблялася за допомогою Hibernate та Spring Framework контролерів які на пряму взаємодіють через Dispatcher Servlet з сервлетами і подальший вивід на сторінки HTML, CSS, JS або вивід JSON об'єкту через REST контролер на сторінки фронтенду де методами JavaScript виконується взаємодія з HTML сторінками.

Вся екосистема застосунку побудована з використанням проекту Spring Boot, який спрощує розробку та налаштування програми, дозволяючи позбутися рутинних кроків конфігурації застосунку. В свою чергу контролери здійснюють маніпуляції з Базою Даних через інтерфейси Spring Framework бібліотеки – CrudRepository які інжекються у контролер завдяки Dependency Injection(рис. 2.9) зі всестороннєю взаємодією з базою даних. Всі сервлети виконують функцію після перенаправлення основного Dispatcher Servlet. (рис. 2.10)

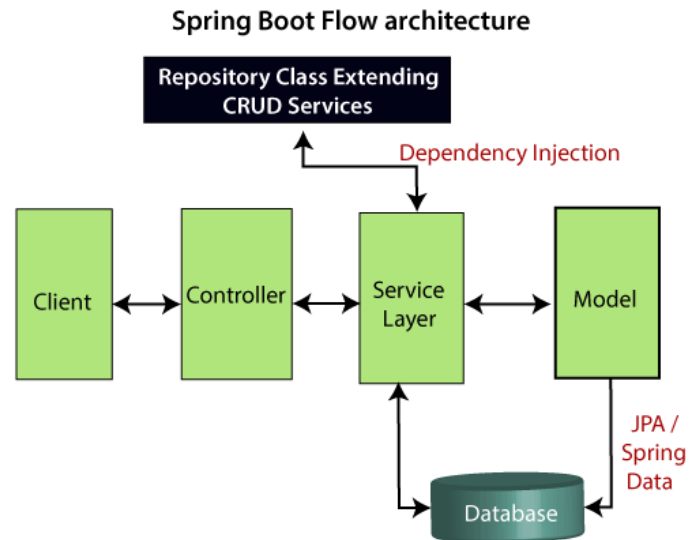


Рисунок 2.9 – Архітектура Spring Boot проекту

Джерело: Розроблено автором

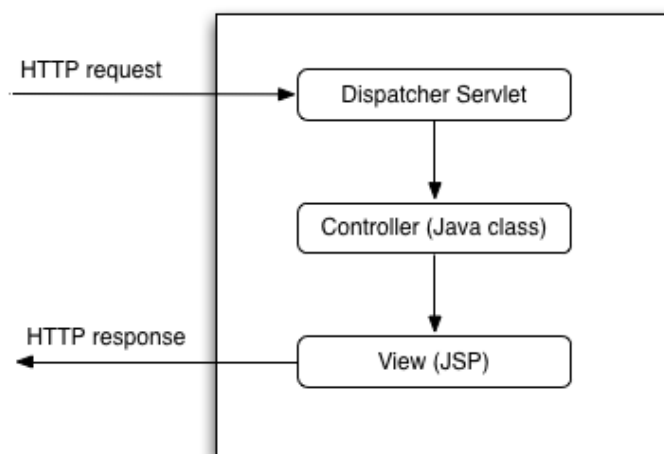


Рисунок 2.10 - Діаграма послідовної виконання запитів

Джерело: Розроблено автором

Використання фреймворків і необхідних бібліотек з грамотним проектуванням та налаштуванням бази даних які були протестовані в дії розробниками, дозволяють скоротити і розробити дуже швидкий код при розгортанні застосунку та при експлуатації. На рисунку 2.11 зображено логування середою розробки швидкодії розвертання і запуску коду на сервері.

```

INFO 65300 --- [ restartedMain] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "{[/authlogin],methods=[POST]}" onto public org.springframework.
INFO 65300 --- [ restartedMain] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "{[/createUser]}" onto public org.springframework.web.servlet.Mc
INFO 65300 --- [ restartedMain] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "{[/login]}" onto public java.lang.String com.stalmakof.diploma.
INFO 65300 --- [ restartedMain] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "{[/error]}" onto public org.springframework.http.ResponseEntity
INFO 65300 --- [ restartedMain] s.w.s.m.a.RequestMappingHandlerMapping : Mapped "{[/error],produces=[text/html]}" onto public org.springframework
INFO 65300 --- [ restartedMain] o.s.w.s.h.SimpleUrlHandlerMapping : Mapped URL path [/webjars/**] onto handler of type [class org.springfra
INFO 65300 --- [ restartedMain] o.s.w.s.h.SimpleUrlHandlerMapping : Mapped URL path [/**] onto handler of type [class org.springframework.w
INFO 65300 --- [ restartedMain] o.s.w.s.h.SimpleUrlHandlerMapping : Mapped URL path [/**/favicon.ico] onto handler of type [class org.sprin
INFO 65300 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
INFO 65300 --- [ restartedMain] o.s.j.e.a.AnnotationMBeanExporter : Registering beans for JMX exposure on startup
INFO 65300 --- [ restartedMain] s.b.c.e.t.TomcatEmbeddedServletContainer : Tomcat started on port(s): 8080 (http)
INFO 65300 --- [ restartedMain] c.s.d.DiplomaApplication : Started DiplomaApplication in 13.914 seconds (JVM running for 19.949)
INFO 65300 --- [nio-8080-exec-1] o.a.c.c.C.[.].[/] : Initializing Spring FrameworkServlet 'dispatcherServlet'
INFO 65300 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization started
INFO 65300 --- [nio-8080-exec-1] o.s.w.s.DispatcherServlet : FrameworkServlet 'dispatcherServlet': initialization completed in 43 ms
INFO 65300 --- [nio-8080-exec-1] c.s.d.c.ViewController : Authenticate User = admin

```

Рисунок 2.11 – Логування запуску серверу

Джерело: Розроблено автором

Servlet API, дозволяє взаємодіяти веб-контейнеру з сервлетами. Веб-контейнер — це один з головних компонентів сервера, який створюється для взаємодії користувачем з сервлетами. Він взаємодіє з описною логікою сервлетів та обробляє вхідні URL запити користувача у сервлет та гарантує, щоб зроблений запит, мав відповідні права доступу.

Мета DispatcherServlet у коді програми є ключовою для управління викликами сервлетів. DispatcherServlet, є керівним сервлетом для обробки запитів від користувачів та знаходить правильну комбінацію методів чи класів та представлень, які виконуються поєднуються і обробляються для формування клієнтської сторінки, з яким взаємодіють користувачі.

Схема роботи застосування у сервлет частині:

- користувач потрапляє на веб-сторінку та надсилає потрібний HTTP запит на сервер;
- сервер отримує запит та відправляє його на контейнер сервлетів;
- контейнер сервлетів аналізує, логіку якого сервлету необхідно; використати, відповідно запиту яку надав запит користувача.
- контейнер сервлетів викликає необхідний сервлет з request і response;
- сервлет виконує дії які були прописані в ньому та взаємодіє за необхідністю з базою даних та з клієнтськими сторінками через об'єкти;

- після завершення логіки сервлету, контейнер сервлетів повертає управління назад на сервер.

2.6 Висновки по розділу

Туроператор може здійснювати діяльність в управлінні та маніпулюванні даними які стосуються бронювань клієнта, так і напряду взаємодіючи з клієнтом демонструючи необхідні тури, використовуючи при цьому лише єдиний програмний продукт – веб-застосунок «туристична агенція». Реалізація застосунку полягає у використанні клієнт-серверної взаємодії та взаємодії серверу з базою даних. Програмний продукт містить бібліотеки та необхідні данні які можуть автоматично завантажуватись за допомогою засобу для автоматизації роботи з програмними проєктами та може розташовуватися та розвертатися на хмарному сервері.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ЗАСТОСУНКУ «ТУРИСТИЧНОЇ АГЕНЦІЇ»

3.1 Особливості реалізації застосунку

Головною задачею проекту є розробка чіткої і надійної взаємодії користувача з логікою програми, тому приділялося багато уваги плануванню та розробці саме архітектури програмного забезпечення, та взаємодії всіх компонентів між одна одним.

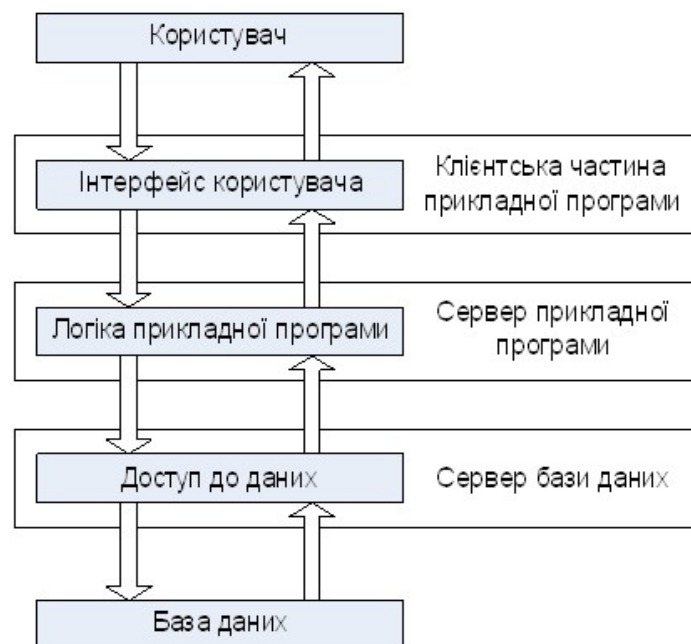


Рисунок 3.1 - Діаграма взаємодії компонентів програмного забезпечення

Джерело: Розроблено автором

Реалізація автоматизації процесів роботи з базою даних для завантаження та вивантаження даних має працювати без нарікань в коді для швидкого реагування бази даних у високо-навантажених умовах взаємодії з нею. База даних повинна бути налаштована правильно, з сучасним функціоналом для зниження можливості несправності.

3.2 Конструювання застосунку

Конструювання веб-застосунку «туристичної агенції» відбувалося з етапів створення серверної частини та бази даних, в якій було створено моделі сутностей(entities) та їх зв'язків, створення класів для взаємодії з базою даних через CRUD репозиторії, Hibernate та REST контролери для роботи з методами та JSON даними. Crud репозиторії у застосунку були використані для взаємодії з даними таблиці «booking», «flight» та таблицею «user». Збірка проекту виконувалась завдяки програмному забезпеченню для роботи зі залежностями проектів – Maven. Де було описано необхідні бібліотеки та їх версії, які необхідно загрузити з maven репозиторію для подальшої компіляції, збірки і запуску проекту.

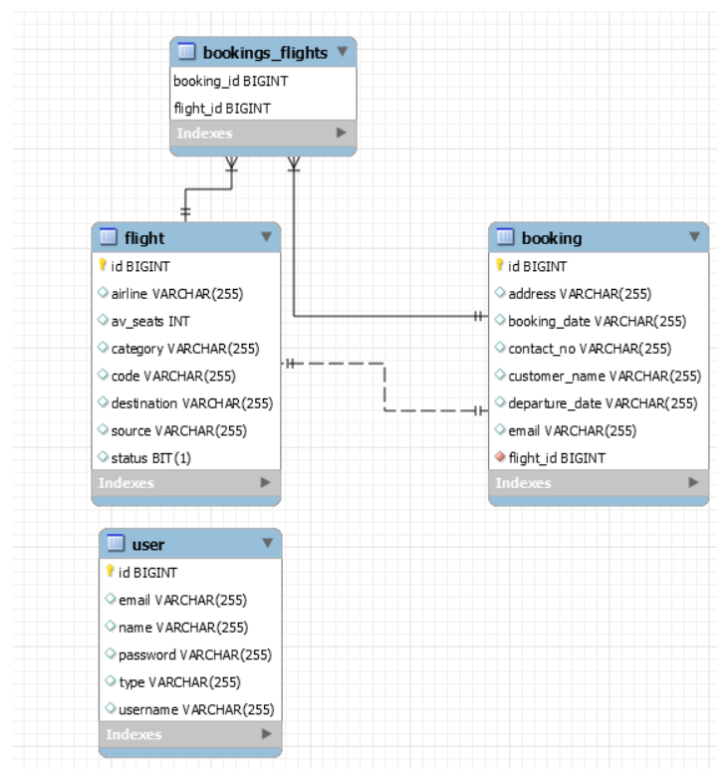


Рисунок 3.2 - Діаграма таблиць в базі даних

Джерело: Розроблено автором

Maven дозволяє використовувати плагіни та відповідні команди для збірки, очищення зібраного проекту, компіляції, інсталяції, верифікації та розгортання проекту.

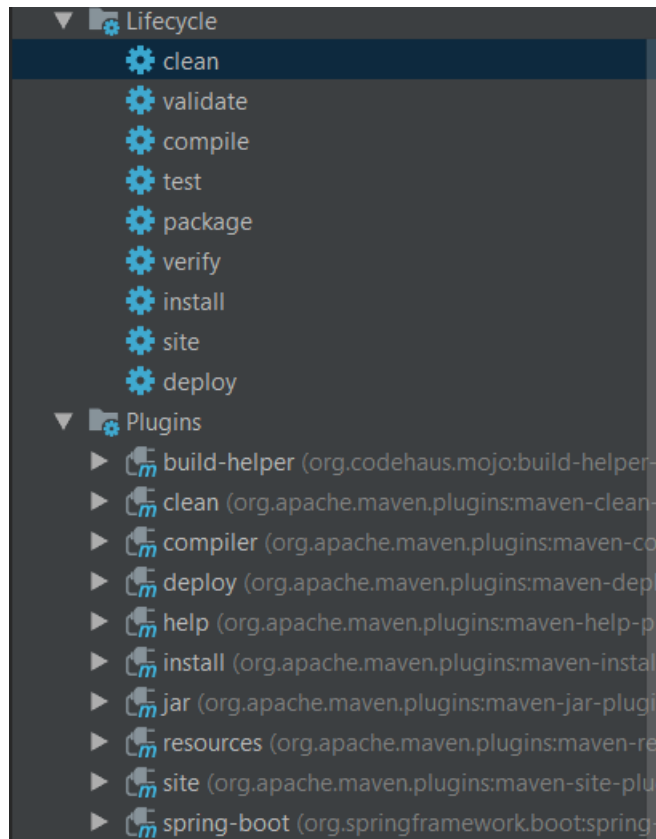


Рисунок 3.3 – Інструментарій збірника Maven

Джерело: Розроблено автором

У файлі `application.properties` містяться дані щодо з'єднання зі сервером бази даних та дані щодо відображення SQL логування у консолі середі розробки.

Клієнтська частина веб-застосунку розташовується у папці «resources» і складається з самих HTML сторінок та їх CSS стилей, бібліотеки Bootstrap, картинок, JavaScript файлів та їх необхідних бібліотек для незначного інтерактиву на сторінках, та AngularJS бібліотеки для передачі даних введених користувачами у формах через створені скрипти, HTTP запитом на контролери серверної частини програмного забезпечення та відображення даних переданих з контролерів на клієнтську частину. AngularJS дозволяє ініціалізувати необхідні данні на JavaScript сторінці та у подальшому за допомогою методів взаємодіяти, редагувати данні або клієнтські сторінки, отримувати данні з клієнтської частини та відправка даних від клієнта на контролери серверу за допомогою JSON формату.

Робота зі змінними у фреймворку здійснюється через \$scopes, які можуть взаємодіяти як з масивами так і поодинокими даними. Данні витягується з форм на клієнтських сторінках.

Веб-застосунок запускається через головний клас програми помічений анотацією «@SpringBootApplication» та виконавши єдиний метод для запуску програмного забезпечення.

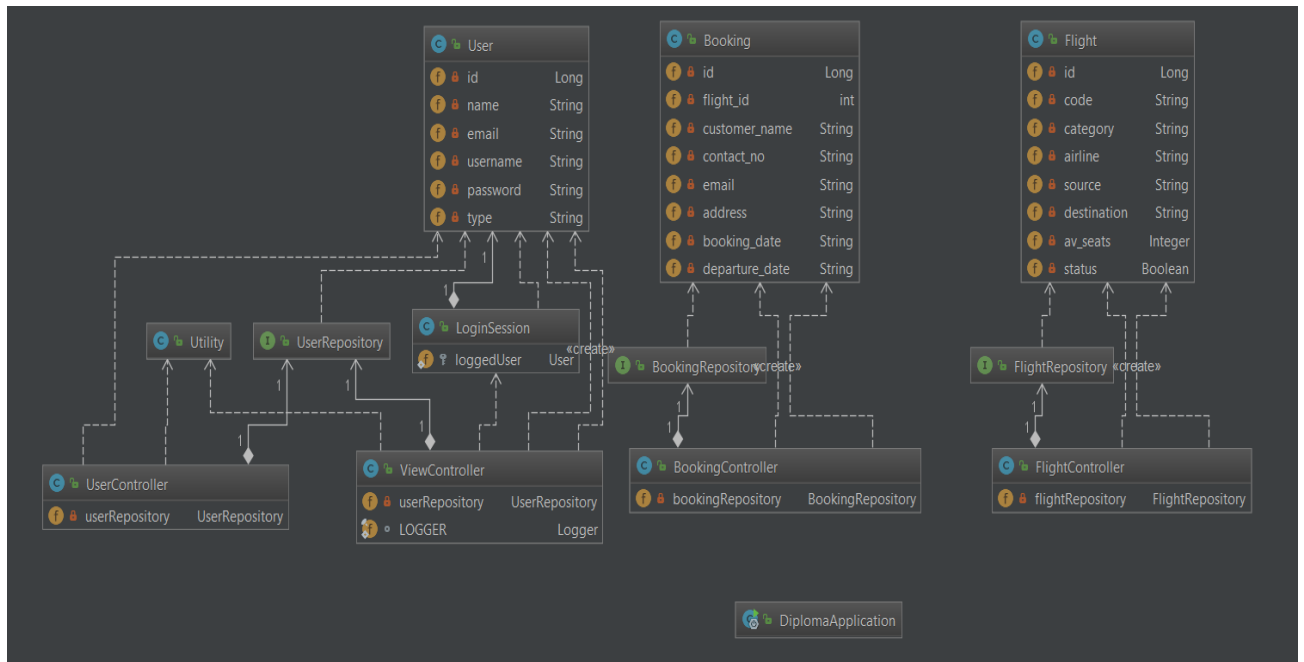


Рисунок 3.3 - Діаграма класів застосунку

Джерело: Розроблено автором

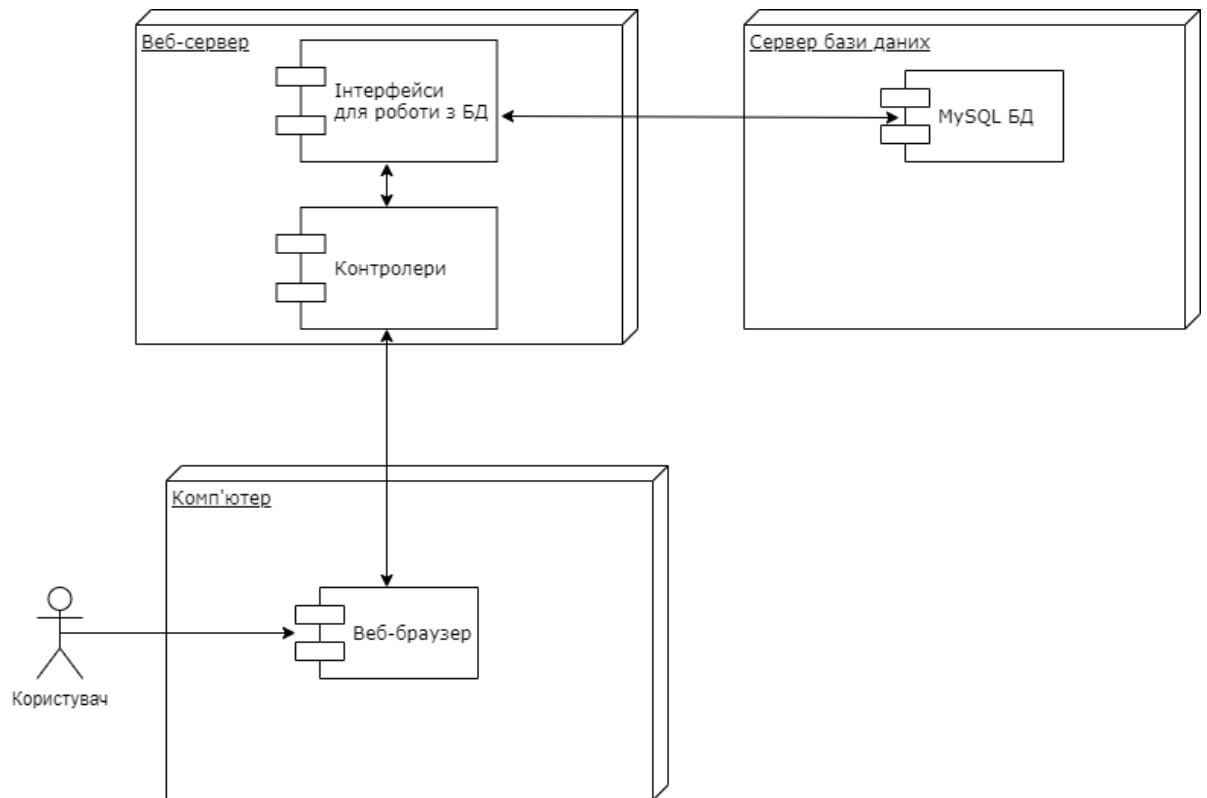


Рисунок 3.4 – Діаграма розгортання
Джерело: Розроблено автором

3.3 Документування застосунку

Інструкція користувача:

1. Запустити сервер на якому розташоване програмне забезпечення
2. Перейти на сторінку - localhost:8080, якщо програмне забезпечення на локальному комп'ютері, або перейти за адресою хостинг серверу

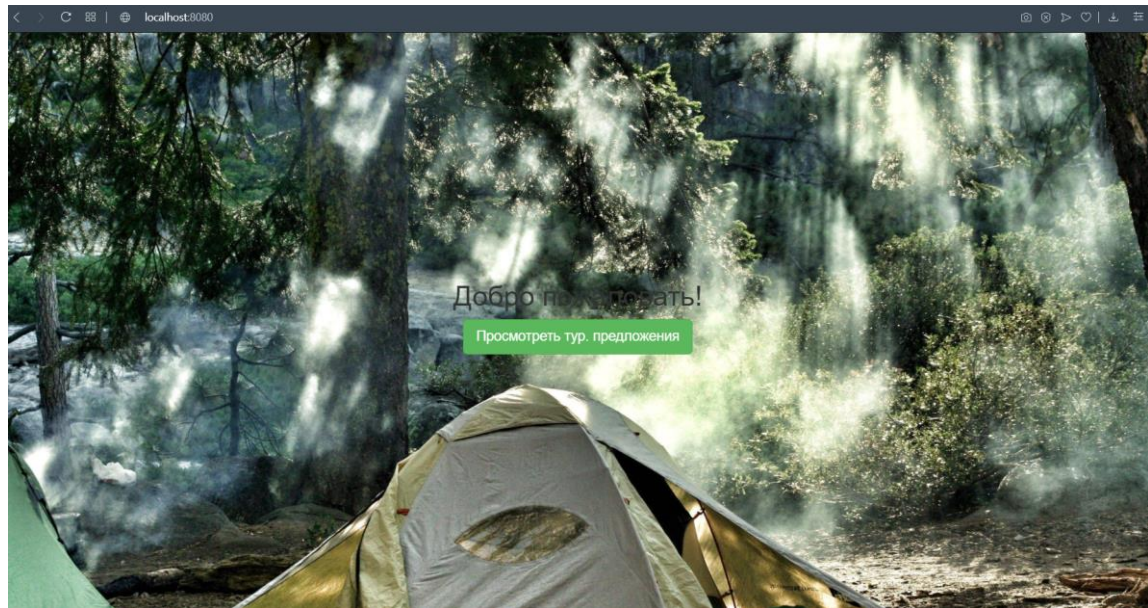


Рисунок 3.5 – Стартова сторінка

Джерело: Розроблено автором

3. Натиснути кнопку «Просмотреть тур. предложения», та перейти на сторінку туристичних пропозицій

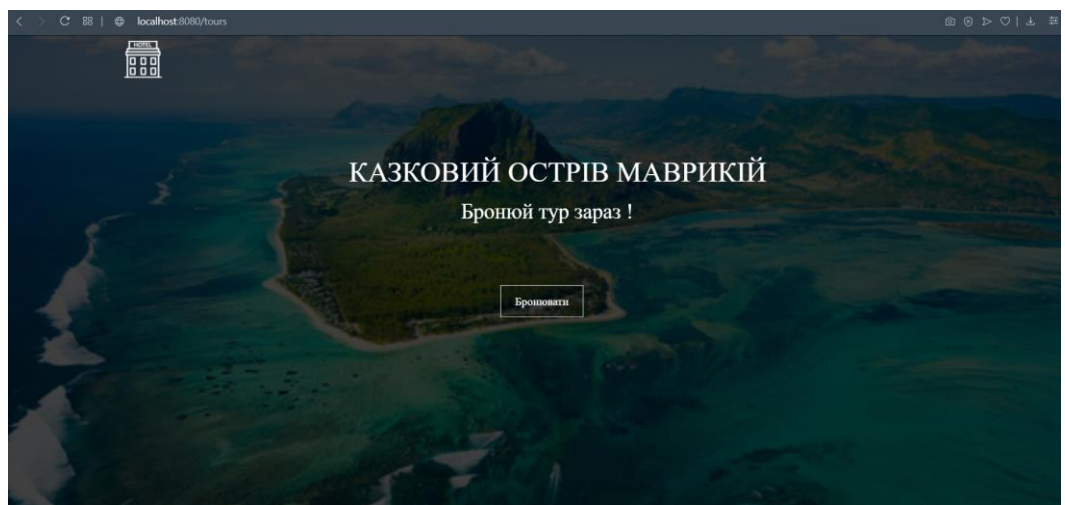


Рисунок 3.6 – Сторінка туристичних пропозицій

Джерело: Розроблено автором



Про Маврикій

У південній частині Індійського океану, недалеко від острова Мадагаскар знаходиться Маврикій – казковий острів з широким смугом білих піщаних пляжів. Коралові рифи оточують його з усіх боків, тому океан тут завжди спокійний.

Маврикій – це цивілізована екзотика з шикарними готелями на побережжі. Маврикій разом з островом Реюньон і островом Родрігес складає групу Маскаренських островів. Тут Ви чудово проведете час, насолоджуючись курортним життям.

Маврикій – невеликий острів, його розміри 65 км на 50 км. Рекомендуємо відвідати Флік-ан-Флак, яке дуже популярне місце для відпочинку місцевих, тут чудові пляжі, а неподалік прекрасний заповідник Казета Берд Парк. Бухта Таварина з її дуже частими потужними хвилями, є ідеальним місцем для серферів на о. Маврикій, відпочинком на якому запам'ятається на довго.



Рисунок 3.7 – Сторінка туристичних пропозицій

Джерело: Розроблено автором



Готель

Готель категорії 5 * розташований в південно-західній частині Маврикія, в 70 км від аеропорту на власному півострові, в 7 км від пляжу біля підніжжя гори Ле Морі. Ідеальне місце для любителів активного відпочинку. Готель славиться своїм прекрасним полем для гольфу (18 лунок). Для зручності гостей по готелю Paradis курсують невеликі електропоїзди. Увечері в готелі грає жива музика, влаштовуються фольклорні танцювальні уявлення, є казино.

Ресторан Blue Martin, що став своєрідним символом готелю, знаходиться прямо біля води, і вечір, проведений при свічках, залишиться в пам'яті надовго. Номерний фонд: 286 номерів і 13 вілл.

Гості готелю можуть повернутися в будь-якому з ресторанів з видом на море або на басейн. У готелі є прекрасний італійський ресторан La Palma і ресторан креольської кухні La Ravanne.



Рисунок 3.8 – Сторінка туристичних пропозицій

Джерело: Розроблено автором

4. Натиснути кнопку «Бронювати» для переміщення до панелі авторизації.

Рисунок 3.9 – Панель авторизації

Джерело: Розроблено автором

5. Якщо адміністратор вже має данні для авторизації в системі – ввести ці данні та натиснути кнопку «Login», якщо адміністратор ще не має даних – натиснути кнопку «Register», для опрацювання внутрішнього коду системи та створення аккаунту, та далі ввести данні і натиснути кнопку «Login»
6. Після авторизації – отримати доступ до менеджменту бронювання авіаквитків

#	Code	Category	Airline	Seats	From	Destination	Actions
1	12125	Business	Turkish Airlines	3	Kyiv	Mauritius	Edit Book Remove
2	54353	Business	Malaysian Airlines	1	Kharkiv	Mauritius	Edit Book Remove
3	43234	Business	Ryanair	1	Kyiv	Stockholm	Edit Book Remove
4	23421	Business	Ryanair	2	Stockholm	Mauritius	Edit Book Remove
5	98723	Business	Ryanair	2	Oslo	Accra	Edit Book Remove
6	34212	Econom	Ukrainian Airlines	2	Accra	Mauritius	Edit Book Remove
7	78934	Econom	Turkish Airlines	1	Istanbul	Jakarta	Edit Book Remove

Рисунок 3.10 – Панель менеджменту бронювання авіаквитків

Джерело: Розроблено автором

7. На сторінці бронювання квитків, користувач може редагувати данні щодо бронювання авіаквитків, для цього він може натиснути кнопку «Edit», після якої користувач потрапляє на сторінку редагування даних, де користувач задає необхідні до бронювання авіаквитка данні.

The screenshot shows a web application for a Travel Agency. A modal window titled 'Edit Flight' is open, allowing a user to edit flight details. The background shows a table of flight bookings with columns for ID, Code, Category, and Action buttons (Book, Remove). The 'Edit Flight' form contains the following fields:

- Flight Code: 23421
- Category: Business
- Airline: Ryanair
- Seats: 2
- From: Stockholm
- Destination: Mauritius

At the bottom right of the form are two buttons: 'Close' and 'Save'.

Рисунок 3.11 – Форма зміни даних бронювання авіаквитків

Джерело: Розроблено автором

8. Користувач натискає кнопку «Save» і успішно зберігає данні.
9. Якщо користувачу потрібно видалити данні, він натискає кнопку «Remove»

10. Після збереження в базу даних інформації щодо бронювання квитків, користувач натискає біля відповідного бронювання кнопку «Book» для бронювання номеру в готелі, після якої користувач потрапляє на наступну форму

Place Booking


Customer Name:

Contact No:

Email:

Address:

Booking Date:

Departure Date:

Close

Save

Рисунок 3.12 – Форма бронювання номерів в готелях

Джерело: Розроблено автором

11. Після заповнення даних у формі, користувач натискає кнопку «Save» для бронювання номеру в готелі, після якої користувач може побачити відповідні данні на сторінці «Bookings»

Travel Agency							
Flights Bookings Users							
BOOKINGS							
#	Flight Index	Name	Email	Contact No	Address	Departure Date	Actions
1	4	Danylo Stalmakov	stalmakov@email.com	+343242342455	Khreshatik St. 29, 78 apt.	2021-05-30T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
2	4	John Doe	johndoe@email.com	+923232323133	York Avenue 25, 32 apt.	2021-05-30T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
3	5	Ivan Ivanov	ivanov@email.com	+382343242315	Centralny Prospect 28, apt.56	2021-06-29T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
4	14	Sergey	jje@je.com	+384653758339	Tv st. 23	2021-05-28T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
5	16	Oleg	kokldf@f.com	+378567856287	Dert St. apt 14.	2021-06-04T21:00:00.000Z	<div>Edit</div> <div>Remove</div>

Рисунок 3.13 – Панель бронювання номерів в готелях

Джерело: Розроблено автором

12. Користувач потрапляє на сторінку заповнення даних у формі, користувач натискає кнопку «Save» для бронювання номеру в готелі, після якої користувач може побачити відповідні данні на сторінці «Bookings» де бачить заброньований номер в готелі.

13. Користувач тепер може редагувати чи видалити бронювання. Для видалення натиснути кнопку «Remove», для редагування бронювання натиснути кнопку «Edit»

Edit Booking

Customer Name:

Danylo Stalmakov

Contact No:

+343242342455

Email:

stalmakov@email.com

Address:

Khreshatik St. 29, 78 apt.

Booking Date:

mm/dd/yyyy

Departure Date:

mm/dd/yyyy

Close

Save

Рисунок 3.14 – Форма редагування номерів в готелях

Джерело: Розроблено автором

14. Користувач після заповнення може чи закрити форму натиснувши кнопку «Close», чи зберегти бронювання, натиснувши кнопку «Save».
15. Відредаговане бронювання відобразиться на сторінці

BOOKINGS

#	Flight Index	Name	Email	Contact No	Address	Departure Date	Actions
1	4	Danylo Stalmakov	stalmakov@email.com	+343242342455	Khreshatik St. 29, 78 apt.	2021-05-30T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
2	4	John Doe	johndoe@email.com	+923232323133	York Avenue 25, 32 apt.	2021-05-30T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
3	5	Ivan Ivanov	ivanov@email.com	+382343242315	Centralny Prospect 28, apt.56	2021-06-29T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
4	14	Sergey	jje@je.com	+384653758339	Tv st. 23	2021-05-28T21:00:00.000Z	<div>Edit</div> <div>Remove</div>
5	16	Oleg	kokldf@f.com	+378567856287	Dert St. apt 14.	2021-06-04T21:00:00.000Z	<div>Edit</div> <div>Remove</div>

Рисунок 3.15 – Відредагване бронювання

Джерело: Розроблено автором

16. Необхідне бронювання, за бажанням можливо видалити натиснувши кнопку «Remove» .

3.4 Адміністрування застосунку

Адміністрування застосунку вимагає від користувачів серверу для розгортання застосунку. Сервер може бути як локальним так і хмарним. При розробці програмного застосунку був використаний сервер – Apache Tomcat v9.0. Spring Framework по стандарту включає в збірку сервер. Для адміністрування та редагування програмного забезпечення розробником потрібно використовуючи середу розробки, наприклад IntelliJ IDEA розгорнути файли Maven форматом та при появі віконця середі спочатку виконати очищення і потім пакування.

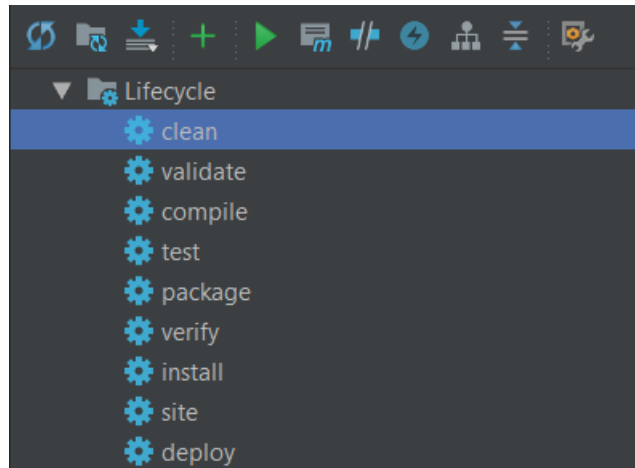


Рисунок 3.11 – Очищення використовуючи життєвий цикл

Джерело: Розроблено автором

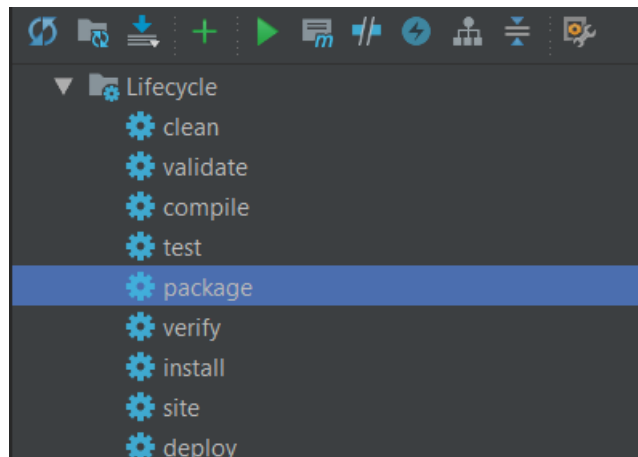


Рисунок 3.11 – Пакування використовуючи життєвий цикл

Джерело: Розроблено автором

Програмний продукт має наступну структуру класів:

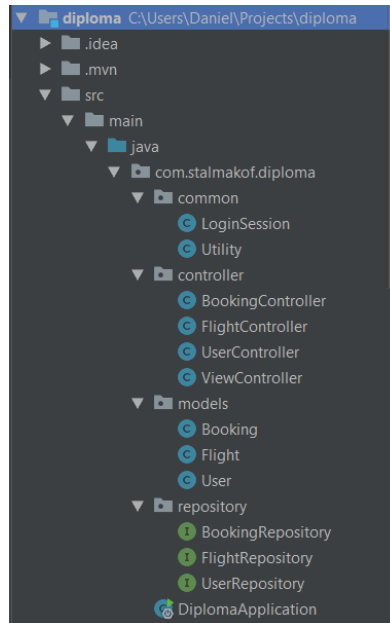


Рисунок 3.11 – Структура класів проекту

Джерело: Розроблено автором

Програмний продукт має наступну структуру ресурсів проекту:

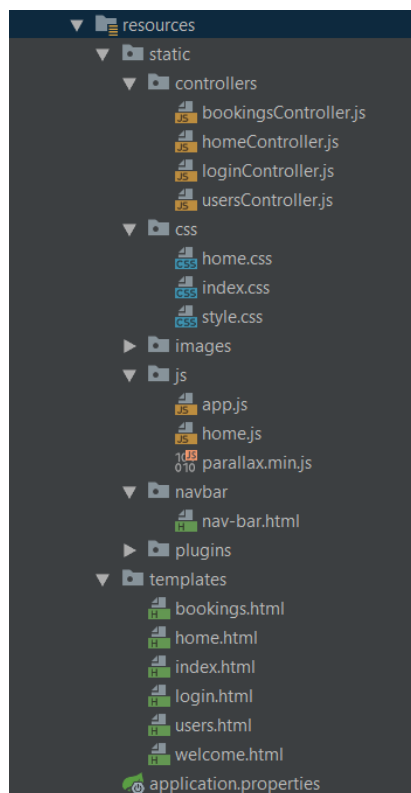


Рисунок 3.12 – Структура ресурсів проекту

Джерело: Розроблено автором

Для роботи програми необхідно мати MySQL сервер встановлений на комп'ютері та бути створені відповідні таблиці в базі даних. Для подальшого адміністрування програмного забезпечення необхідно зайти в клас ViewController.java та в методі createUser(), змінити данні на необхідні, для подальшої реєстрації адміністратора у панелі авторизації. Після пакування файлів виконати запуск веб-застосуку.

3.5 Тестування застосунку

Тестування програмних компонентів дозволяють власникам туристичних агенцій, не втратити важливі часові, людські та матеріальні ресурси на етапі виходу програмного продукту на ринок туристичних послуг. Тому програмуються захисна логіка щодо усунення проблем з продуктивністю та багів. На Рис.3.13-3.16 зображені типові помилки програми при використанні застосунку логіка якого працює некоректно, та запрограмовані способи вирішення цих проблем.

При розробці програмного забезпечення, було використане ручне, мануальне тестування застосунку яке зменшило шанс отримання помилкових і неякісних результатів роботи.

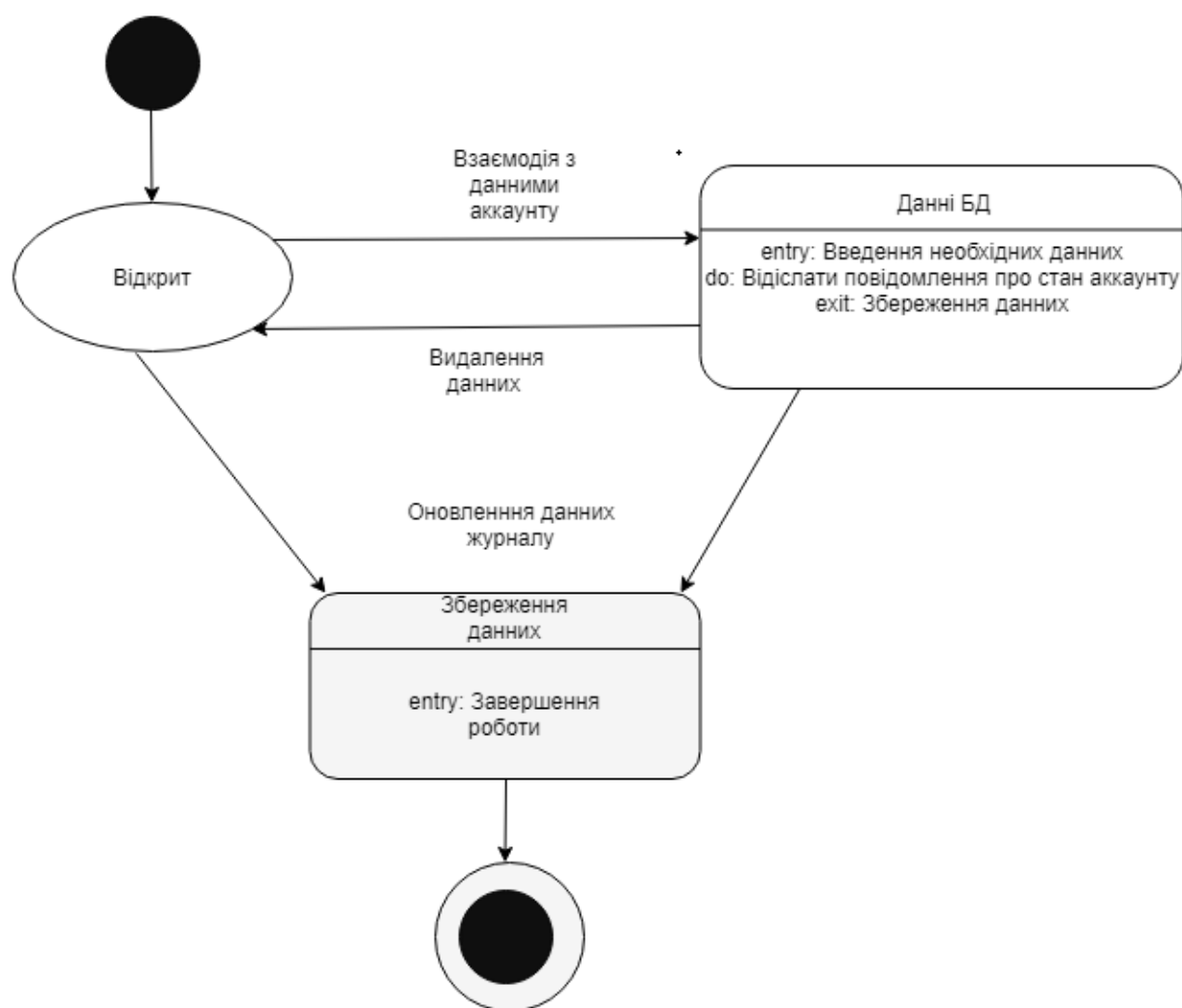


Рисунок 3.13 - Діаграма маніпуляції з програмними даними

Джерело: Розроблено автором

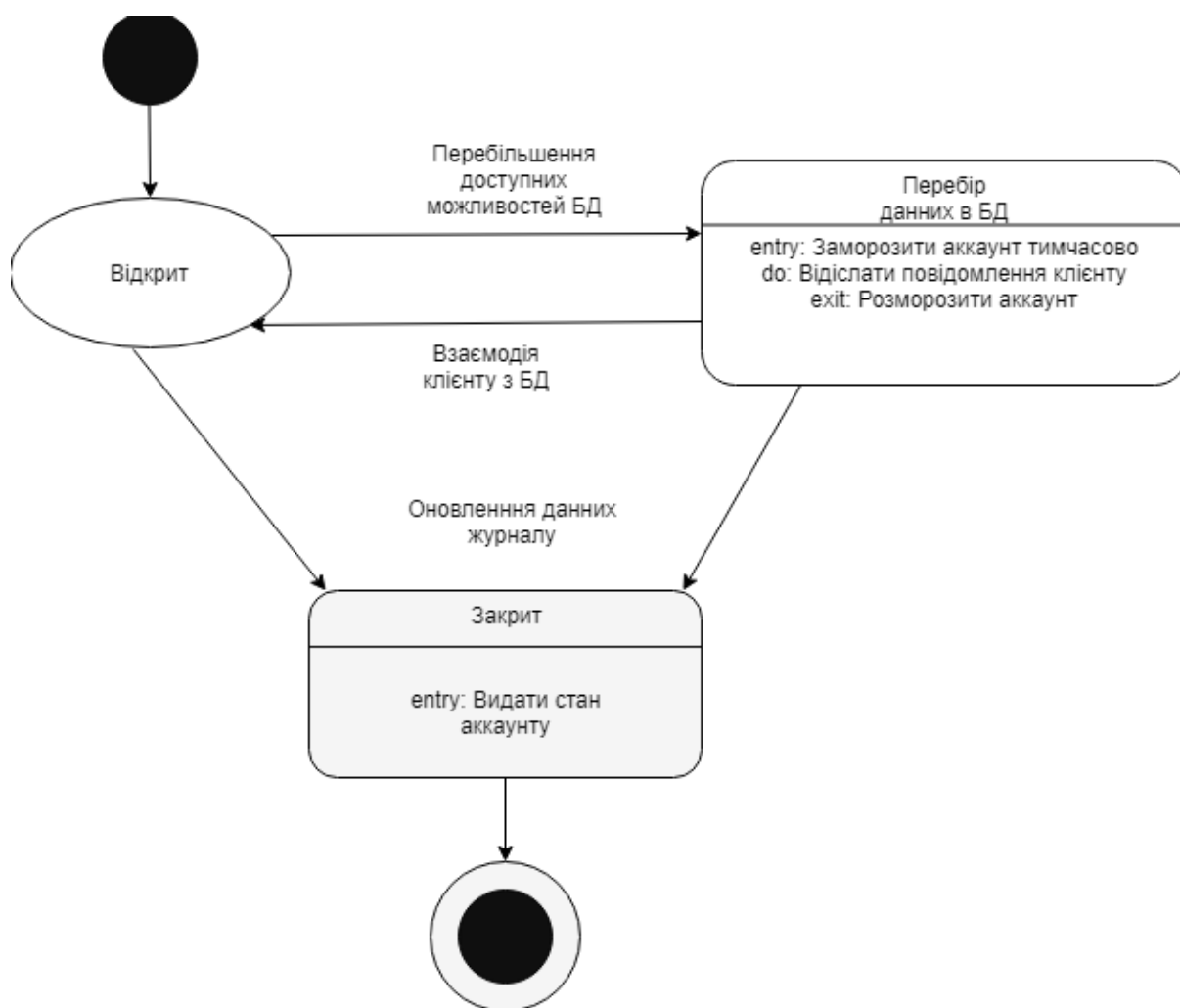


Рисунок 3.14 – Діаграма з зображенням дій програми при перебільшенні даних

Джерело: Розроблено автором

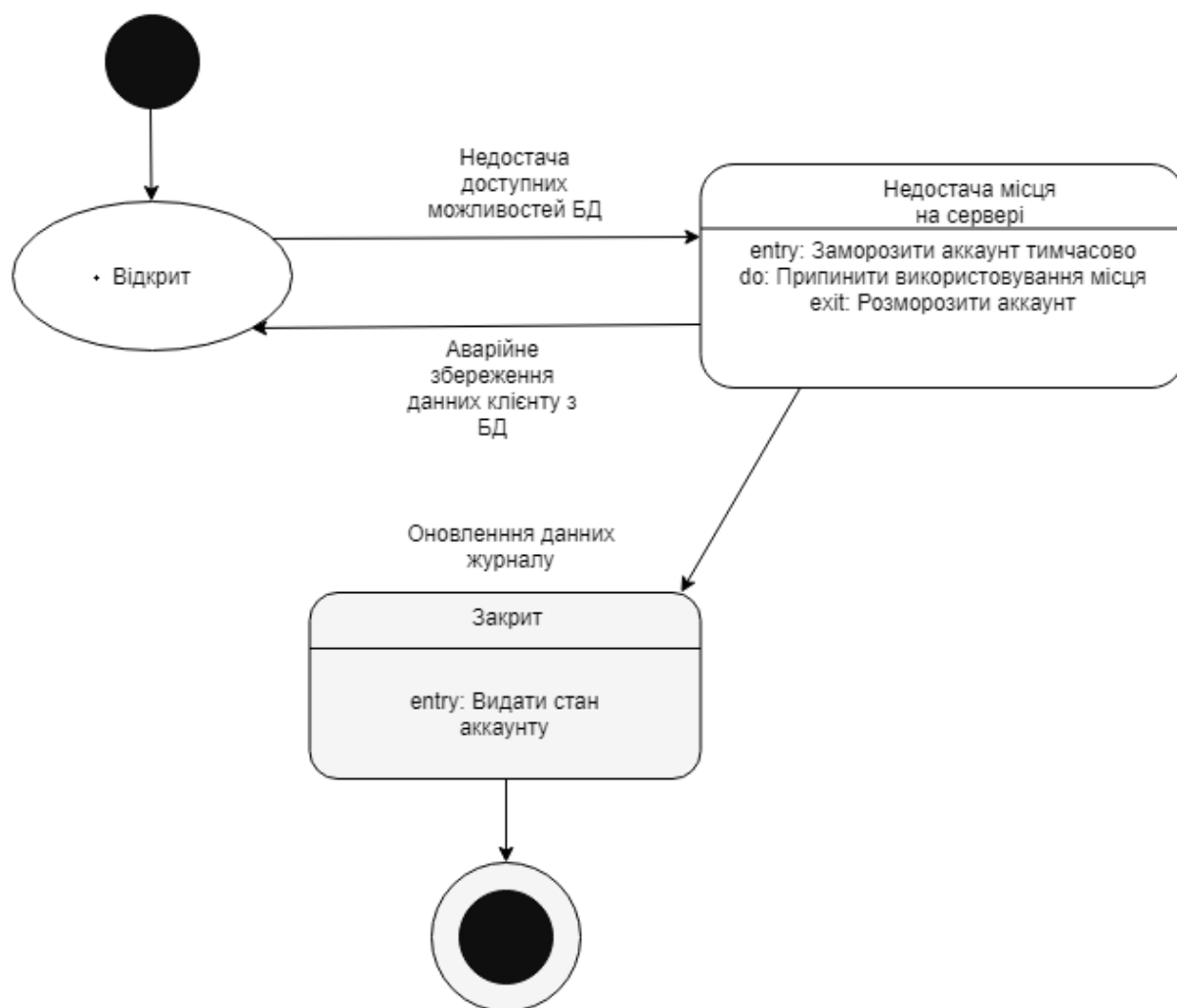


Рисунок 3.15 – Діаграма з зображенням помилки нестачі
місця

Джерело: Розроблено автором

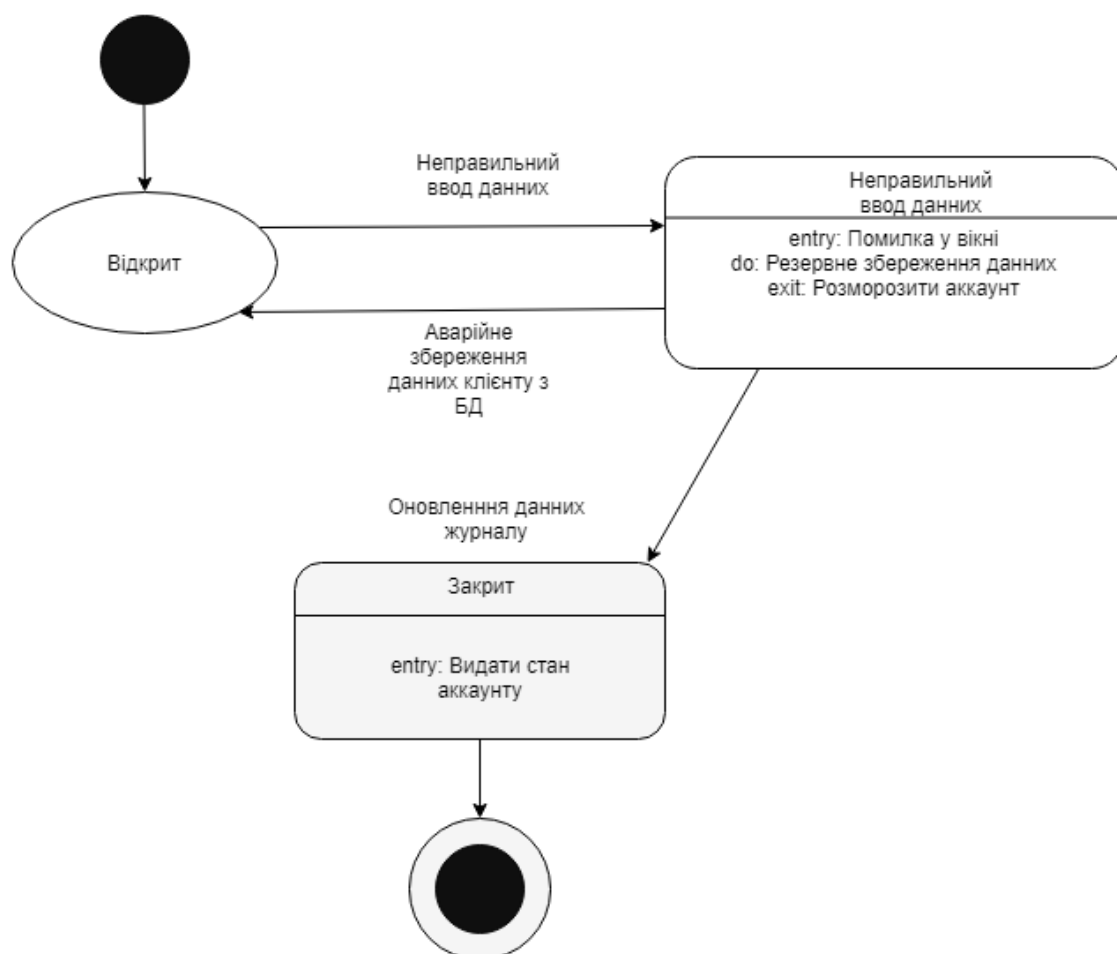


Рисунок 3.16 – Діаграма з зображенням помилки
неправильного вводу даних

Джерело: Розроблено автором

3.6 Підготовка застосунку до оформлення права інтелектуальної власності відповідно до національного законодавства

Інтенсивність розвитку всіх галузей виробництва, а також засобів їх інформаційної підтримки призводять до різкого зростання обсягів інформації, в тому числі представленої в текстовому вигляді. До текстових варіантів відносяться і програмні застосунки які стали стрімко розповсюджуватись на всіх електронних пристроях. Для того, щоб виявити яким чином можна розповсюджувати мій web застосунок із дотриманням прав інтелектуальної власності, він має пройти декілька перевірок, один із них аналіз коду вебзастосунку на запозичення (табл. В.1, додаток В) для виявлення «золотого злитку».

Проведений аналіз дозволяє виділити «золотий злиток» веб-застосунку обсягом 60% (відповідні строки виділено сірим кольором в табл. 3.4).

Для визначення виду вихідної ліцензії на веб-застосунок нами виконано аналіз запозичень фрагментів коду щодо ліцензійних обмежень (табл. В.2, додаток В). Як бачимо, запозичені фрагменти коду веб-застосунку регламентуються ліцензіями Open Source (15%), OFL License (5%), MIT License, the Apache (15%). Ліцензія Open source належить до виду ліцензій «копілефт». MIT – до дозвільних ліцензій з відповідними можливостями/обмеженнями, які узгоджуються між собою. Дозвільні ліцензії не обмежують вибір ліцензії для робіт, похідних від роботи з дозвільною ліцензією. Проте ліцензія Open Source вимагає, аби будь-які похідні роботи також мають випускатись під цією ж ліцензією. Виходячи з цього, вихідна ліцензія для розробленого веб- застосунку – Open Source з наступними ключовими можливостям: Відсутність вимог розкриття коду кінцевого продукту, щодо реєстрації і патентування та відсутність вимог надання кінцевим напрацюванням статусу суспільного надбання. Додаткова інформація про веб-застосунок як об'єкт авторського права для ініціювання процедури реєстрації авторського права на твір приведено у додатку Г.

3.7 Висновки по розділу

Програма видає результат згідно плану проектування програми. Адміністратор має можливість вивантажувати, редагувати та маніпулювати даними згідно плану туристичного закладу. Програмний продукт може бути змінений, доповнений в індивідуальному порядку або вже готовий до використання. Для цілей зміни програмного коду, може бути використана одна з систем контролю версій. Програмний продукт може бути продемонстрований в хмарних сервісах для розвертання програмного забезпечення, наприклад AWS.

ВИСНОВКИ

У цілому даний програмний продукт працює без помітних помилок та всі задачі виконує як потрібно.

Програмне забезпечення може без проблем використовуватися у закладах туризму та інших місцях для показу інформації та з менеджменту бронювання авіаквитків та номерів готелів. Усі програмні компоненти працюють без помилок. Вся інформація о турах на головній сторінці загружається та може редагуватися, данні про бронювання авіаквитків та номерів готелів вивантажуються та видозмінюються. Робота з базою даних ведеться без нарікань. Програма може бути спокійно видозмінена.

Програма була написана на мові програмування Java та за допомогою системи керування баз даних MySQL. При розробці використовувалась інтегрована середа IntelliJ IDEA та сервер Tomcat V9.0.

СПИСОК ПОСИЛАНЬ

1. Кей Хорстманн «Java 2.Тонкости программирования. Том 2» Киев, 2007 г. 34 с.
2. Чистый код: создание, анализ и рефакторинг. Библиотека программиста Мартин Р. 85-99 с.
3. Приемы объектно-ориентированного проектирования Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. 2014. -113 с.
4. Патрик Ноутон «Java 2. Наиболее полное руководство» Киев, 2007 г.320-344с.
5. Tour-Online– [Электронный ресурс] - <https://tour-online.ua>-
Режим доступа: вільний.
6. Герберт Шилдт.Java. Полное руководство. Java SE 7 = Java 7: The Complete Reference. — 8-е изд. — М.: Вильямс, 2012. — 1104 с.
7. Mistotravel– [Электронный ресурс] - <https://misto.travel/index.php?lang=ru>
Режим доступа: вільний.
8. Java Docs– [Электронный ресурс] - <https://docs.oracle.com/en/java/> -
Режим доступа: вільний.
9. Патрик Ноутон «Java 2. Наиболее полное руководство» Киев, 2007 г. 23 с.
10. Грег Б. «Java краткое руководство» Киев, 2001г. 90с.
11. HTML Руководства [Электронный ресурс] MDN Web Docks <https://developer.mozilla.org/ru/docs/Web/HTML> Режим доступа: вільний.
12. MDN web docs JavaScript - JavaScript [Электронный ресурс] MDN Web Docks <https://developer.mozilla.org/ru/docs/Web/HTML> Режим доступа: вільний.
13. jQuery API [Электронный ресурс] <https://api.jquery.com> Режим доступа: вільний.

14. Spring Framework Documentation [Електронний ресурс]
<https://docs.spring.io/spring-framework/docs/current/reference/html/> Режим доступу: вільний.
15. Spring REST Docs [Електронний ресурс] <https://spring.io/projects/spring-restdocs> Режим доступу: вільний.
16. Introduction to Spring REST Docs [Електронний ресурс]
<https://www.baeldung.com/spring-rest-docs> Режим доступу: вільний.
17. Aviasales [Електронний ресурс] <https://www.aviasales.ua> Режим доступу: вільний.
18. IntelliJ Idea JetBrains [Електронний ресурс] <https://www.jetbrains.com/ru-ru/idea/> Режим доступу: вільний.
19. Project Lombok IntelliJ plugin [Електронний ресурс]
<https://projectlombok.org/setup/intellij> Режим доступу: вільний.
20. MySQL [Електронний ресурс] <https://www.mysql.com/why-mysql/white-papers/> Режим доступу: вільний.
21. MySQL Server [Електронний ресурс]
<https://dev.mysql.com/downloads/mysql/> Режим доступу: вільний.
22. Apache Tomcat Docs [Електронний ресурс]
<http://tomcat.apache.org/tomcat-7.0-doc/index.html> Режим доступу: вільний.
23. W3Schools CSS Reference Docs [Електронний ресурс]
<https://www.w3schools.com/cssref/> Режим доступу: вільний.
24. W3Schools HTML Tutorial Reference Docs [Електронний ресурс]
<https://www.w3schools.com/html/default.asp> Режим доступу: вільний.
25. W3Schools SQL Tutorial Reference Docs [Електронний ресурс]
<https://www.w3schools.com/sql/default.asp> Режим доступу: вільний.
26. W3Schools JavaScript Tutorial Reference Docs [Електронний ресурс]
<https://www.w3schools.com/js/default.asp> Режим доступу: вільний.
27. Bootstrap Docs [Електронний ресурс]
<https://getbootstrap.com/docs/5.0/getting-started/introduction/> Режим доступу: вільний.

28. ResearchGate Global growth trend of data volume [Електронний ресурс]
https://www.researchgate.net/figure/Global-growth-trend-of-data-volume-2006-2020-based-on-The-digital-universe-in-2020_fig1_274233315 Режим доступу: вільний.

29. Himanshu's Blog Spring Framework Architecture [Електронний ресурс]
<https://himanshugpt.wordpress.com/2010/07/05/262/> Режим доступу: вільний.

ДОДАТОК А. ЕКРАНИ ЗАСТОСУНКУ ТУРИСТИЧНОЇ АГЕНЦІЇ

Edit Flight

Flight Code:

Category:

Airline:

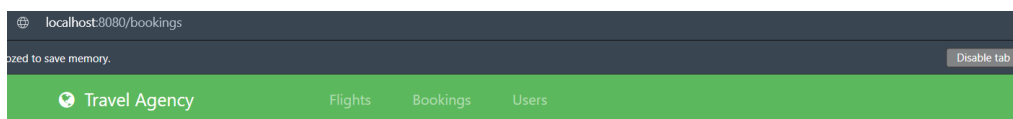
Seats:

From:

Destination:

#	Code	Category
1	12125	Business
2	54353	Business
3	43234	Business
4	23421	Business
5	98723	Business
6	34212	Econom
7	78934	Econom
8	34212	Econom
9	42532	Business

Рисунок А.1 – Форма редагування бронювання авіаквитків

BOOKINGS

Flight #	Index	Name	Email	Contact No	Address	Departure Date	Actions
1	4	Danylo Stalmakov	stalmakov@email.com	+343242342455	Khreshatik St. 29, 78 apt.	2021-05-30T21:00:00.000Z	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
2	4	John Doe	johndoe@email.com	+923232323133	York Avenue 25, 32 apt.	2021-05-30T21:00:00.000Z	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
3	5	Ivan Ivanov	ivanov@email.com	+382343242315	Centralny Prospect 28, apt.56	2021-06-29T21:00:00.000Z	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
4	14	Sergey	jje@je.com	+384653758339	Tv st. 23	2021-05-28T21:00:00.000Z	<input type="button" value="Edit"/> <input type="button" value="Remove"/>
5	16	Oleg	kokltdf@f.com	+378567856287	Dert St. apt 14.	2021-06-04T21:00:00.000Z	<input type="button" value="Edit"/> <input type="button" value="Remove"/>

Рисунок А.2 – Форма бронювання номерів готелей

ДОДАТОК Б. ФРАГМЕНТИ ЛІСТИНГУ

Фрагмент лістингу Б.1 – FlightController

```

@RestController
public class FlightController {

    @Autowired
    private FlightRepository flightRepository;

    @RequestMapping(value = "/api/getFlights", method = RequestMethod.POST)
    public List<Flight> getFlights() {
        return this.getAllFlights();
    }

    @RequestMapping(value = "/api/saveFlight", method = RequestMethod.POST)
    public Flight saveFlight(@RequestBody String data) {
        JSONObject flightData = this.parseData(data);

        Flight flight = new Flight();
        flight.setCode(flightData.get("code").toString());
        flight.setAirline(flightData.get("airline").toString());
        flight.setCategory(flightData.get("category").toString());
        flight.setSource(flightData.get("source").toString());
        flight.setDestination(flightData.get("destination").toString());
        flight.setAv_seats(Integer.parseInt(flightData.get("seats").toString()));
        flight.setStatus(true);

        return this.saveFlightInstance(flight);
    }

    @RequestMapping(value = "/api/editFlight", method = RequestMethod.POST)
    public Flight editFlight(@RequestBody String data) {
        try{
            JSONObject flightData = this.parseData(data);
            Flight flight = this.getFlight(Long.parseLong(flightData.get("flightID").toString()));
            flight.setCategory(flightData.get("new_flight_category").toString());
            flight.setCode(flightData.get("new_flight_code").toString());
            flight.setAirline(flightData.get("new_flight_airline").toString());
            flight.setAv_seats(Integer.parseInt(flightData.get("new_flight_seats").toString()));
            flight.setSource(flightData.get("new_flight_source").toString());
            flight.setDestination(flightData.get("new_flight_destination").toString());
            return this.saveFlightInstance(flight);
        } catch (Exception e) {
            return null;
        }
    }

    @RequestMapping(value = "/api/removeFlight", method = RequestMethod.POST)
    public boolean removeFlight(@RequestBody String data) {
        JSONObject removeData = this.parseData(data);

        Flight flight = new Flight();
        flight.setId(Long.parseLong(removeData.get("flight_id").toString()));
        return this.removeFlight(flight);
    }

    public boolean removeFlight(Flight flight) {
        try {
            flightRepository.delete(flight);

```

```

        return true;
    } catch (TransactionException e) {
        return false;
    }
}

private Flight saveFlightInstance(Flight flight){
    return flightRepository.save(flight);
}

private JSONObject parseData(String data) {
    try {
        JSONParser parser = new JSONParser();
        return (JSONObject) parser.parse(data);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

public Flight getFlight(Long id){
    return flightRepository.findOne(id);
}

public List<Flight> getAllFlights() {
    Iterable<Flight> flights = flightRepository.findAll();
    List<Flight> data = new ArrayList<Flight>();
    for (Flight flight : flights) {
        data.add(flight);
    }
    return data;
}
}

```

Фрагмент лістингу Б.2 – BookingController

```

@RestController
public class BookingController {

    @Autowired
    private BookingRepository bookingRepository;

    @Autowired
    private FlightRepository flightRepository;

    @RequestMapping(value = "/api/saveBooking", method = RequestMethod.POST)
    public Booking saveBooking(@RequestBody String data) {
        JSONObject bookingData = this.parseData(data);

        Booking booking = new Booking();
        booking.setCustomer_name(bookingData.get("customer_name").toString());
        booking.setAddress(bookingData.get("address").toString());
        booking.setContact_no(bookingData.get("contact_no").toString());
        booking.setDeparture_date(bookingData.get("departure_date").toString());
        booking.setBooking_date(bookingData.get("booking_date").toString());
        booking.setEmail(bookingData.get("email").toString());
        booking.setFlight_id(Integer.parseInt(bookingData.get("flight_id").toString()));
    }
}

```

```

        booking.getFlights().add(getFlight(Long.valueOf(booking.getFlight_id())));

        return this.makeBooking(booking);
    }

    @RequestMapping(value = "/api/getBookings", method = RequestMethod.POST)
    public List<Booking> getBookings() {
        return this.getAllBookings();
    }

    @RequestMapping(value = "/api/editBooking", method = RequestMethod.POST)
    public Booking editBooking(@RequestBody String data) {
        try{
            JSONObject bookingData = this.parseData(data);
            Booking booking = this.getBooking(Long.parseLong(bookingData.get("booking_id").toString()));
            booking.setCustomer_name(bookingData.get("customer_name").toString());
            booking.setAddress(bookingData.get("address").toString());
            booking.setContact_no(bookingData.get("contact_no").toString());
            booking.setDeparture_date(bookingData.get("departure_date").toString());
            booking.setBooking_date(bookingData.get("booking_date").toString());
            booking.setEmail(bookingData.get("email").toString());
            booking.getFlights().add(getFlight(Long.valueOf(booking.getFlight_id())));
            return this.makeBooking(booking);
        } catch (Exception e) {
            return null;
        }
    }

    @RequestMapping(value = "/api/removeBooking", method = RequestMethod.POST)
    public boolean removeBooking(@RequestBody String data) {
        JSONObject removeData = this.parseData(data);

        Booking booking = new Booking();
        booking.setId(Long.parseLong(removeData.get("booking_id").toString()));
        return this.removeBooking(booking);
    }

    private JSONObject parseData(String data) {
        try {
            JSONParser parser = new JSONParser();
            return (JSONObject) parser.parse(data);
        } catch (ParseException e) {
            e.printStackTrace();
            return null;
        }
    }

    public Booking getBooking(Long id){
        return bookingRepository.findOne(id);
    }

    public Booking makeBooking(Booking booking) {
        return bookingRepository.save(booking);
    }

    public boolean removeBooking(Booking booking) {
        try {
            bookingRepository.delete(booking);
            return true;
        } catch (TransactionException e) {

```

```
        return false;
    }
}

public List<Booking> getAllBookings() {
    Iterable<Booking> bookings = bookingRepository.findAll();
    List<Booking> data = new ArrayList<Booking>();
    for (Booking booking : bookings) {
        data.add(booking);
    }
    return data;
}

public Flight getFlight(Long id){
    return flightRepository.findOne(id);
}

}
```

ДОДАТОК В. АНАЛІЗ КОДУ ЗАСТОСУНКУ «ТУРИСТИЧНОЇ АГЕНЦІЇ» НА ЗАПОЗИЧЕННЯ

Таблиця В.1

Аналіз коду Застосунку на запозичення

№ з/п	Компонент (назва, тип з діаграми компонентів або класів)	Кількість строк (фізичних/логічних)	Характеристики запозичень	
			джерело	ліцензія
1	Font-awesome	1%	http://fontawesome.io/	SIL OFL 1.1 The MIT License CC BY 3.0 License
2	Bootstrap	3%	https://getbootstrap.com/	The MIT License OpenSource
3	Jquery	2%	URL: https://jquery.org/license/	The MIT License
4	tether	1%	URL: https://github.com/shipshapecode/tether	The MIT License
5	angular	4%	URL: http://angularjs.org/	The MIT License
6	Apache Maven	2%	URL: https://maven.apache.org/ref/3.0/license.html	Apache License 2.0
7	Spring-Framework	4%	URL: https://spring.io/trademarks	Apache License 2.0
8	Java classes	30%		
9	Resources	30%		

Таблиця В.2

Аналіз запозичень у коді Застосунку

№ з/п	Вид ліцензії-донора	% похідного коду	Можливості/обмеження подальшого використання
1	Opensource	8	Обмеження: <ul style="list-style-type: none"> встановлення автора на запозичені частини коду;
2	OFL License	4	<ul style="list-style-type: none"> Put your copyright and Reserved Font Names information at the beginning of the main OFL.txt file in place of the dedicated placeholders (marked with the <> characters). Include this file in your release package. Put your copyright and the OFL text with your chosen Reserved Font Name(s) into your font files (the copyright and license fields).

			<p>A link to the OFL text on the OFL web site is an acceptable (but not recommended) alternative. Also add this information to any other components (build scripts, glyph databases, documentation, test files, etc). Accurate metadata in your font files is beneficial to you as an increasing number of applications are exposing this information to the user. For example, clickable links can bring users back to your website and let them know about other work you have done or services you provide. Depending on the format of your fonts and sources, you can use template human-readable headers or machine-readable metadata. You should also double-check that there is no conflicting metadata in the font itself contradicting the license, such as the fstype bits in the os2 table or fields in the name table.</p> <ul style="list-style-type: none"> • Write an initial FONTLOG.txt for your font and include it in the release package (see Section 6 and Appendix A of the OFL-FAQ for details including a template). • Include the relevant practical documentation on the license by adding the current OFL-FAQ.txt file in your package.
3	CC BY	4	This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use.
4	MIT License	24	<p>Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:</p> <p>The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.</p> <p>THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,</p>

			<p>FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.</p>
5	Apache License 2.0	5	<p>The Apache License is a permissive free software license written by the Apache Software Foundation (ASF).[6] It allows users to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software under the terms of the license, without concern for royalties. The ASF and its projects release their software products under the Apache License. The license is also used by many non-ASF projects.</p>

ДОДАТОК Г. ЗАЯВА ДЛЯ ОФОРМЛЕННЯ ПРАВА ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ НА ЗАСТОСУНОК ТУРИСТИЧНОЇ АГЕНЦІЇ

МІНІСТЕРСТВО РОЗВИТКУ ЕКОНОМІКИ, ТОРГІВЛІ ТА СІЛЬСЬКОГО ГОСПОДАРСТВА УКРАЇНИ (Мінекономіки)

вул. М. Грушевського 12/2, м. Київ, 01008 Тел. (044) 200-47-53, факс (044) 253-63-71
E-mail: meconomy@me.gov.ua, <http://www.me.gov.ua>, код ЄДРПОУ 37508596

ЗАЯВА про реєстрацію авторського права на твір

Службові відмітки:

Підпис начальника відділу
► _____

Номер заявки ▼	Дата подання		
	Число ▲	Місяць ▲	Рік ▲
Номер свідоцтва ▼	Дата реєстрації		
	Число ▲	Місяць ▲	Рік ▲

Прошу зареєструвати авторське право на твір

1. Вид та повна назва твору Комп'ютерна програма – веб-застосунок «Туристична агенція»

Скорочена назва твору (якщо така є)

Попередня чи альтернативна назва твору (якщо така є)

2. Галузь творчої діяльності
Література

Наука, література чи мистецтво

3. До якого об'єкту(ів) авторського права належить твір Комп'ютерна програма

4. Анотація або реферат твору (Публікується в офіційному бюлетені)
Веб-застосунок здійснює менеджмент бронювань

Рекомендований розмір - до 300 друкованих знаків

[illegible]

5. Дата остаточного завершения работы над твором ► Число 25 Місяць травня Рік 2021

6. Відомості про оприлюднення твору (опублікування, сповіщення, виконання, показ тощо)
твір не оприлюднювався _____

7. Відомості про використані твори:

7.1. Відомості про твір(и), відносно якого(их) цей твір є похідним не є похідним

Вказати, на основі якого твору зроблено переклад, адаптацію, аранжування тощо, їх правомірність

7.2. Відомості про твір(и), або частину твору(ів), що включено до твору, права на який реєструються

до твору не включено інші твори або їх частини

Вказати твори інших авторів та правомірність їх включення

8. Відомості про попередню реєстрацію* Ні – ☒; Так – ☐,

реєстрації

та назву реєстру

9. Відомості про автора(ів) твору, зазначеного у п.1 заяви **

9.1. Прізвище, ім'я, по батькові першого автора (псевдонім, за наявності, вказати в дужках) ▼
Стальмаков Данило Федорович

Дата народження ► Число 09 Місяць липня Рік 2000

Повна поштова адреса, телефон Вул. Металістів 35/15 кв. 28 м. Київ,
03057 тел. +380(50)2427010

поштовий індекс

Вулиця, номер будинку, назва населеного пункту, район, область,

Суть авторства, авторський вклад у створення твору автор твору

Цей твір (частину твору) створено:*

Цей твір (частину твору) створено для оприлюднення:*

за договором

☐

під власним ім'ям

☒

у порядку індивідуальної розробки

☒

анонімно

☐

під псевдонімом

☐

* Необхідне позначити "X"

** Якщо авторів декілька, використайте лист подовження бланку заяви

10. Особа, яка подає заявку на реєстрацію (заявник):**10.1. Автор(и), спадкоємець(і) ▼**Стальмаков Данило Федорович

Повне ім'я особи, повна поштова адреса (адреса для листування), телефон

10.2. Довірена особа автора(ів), спадкоємця(ів) ▼

Повне ім'я особи, повна поштова адреса (адреса для листування), телефон

11. Видача свідоцтва (свідоцтв):

- Надіслати за вказаною адресою

Вулиця, номер будинку, назва населеного пункту, район, область, поштовий
індекс

- Видати заявнику

Прізвище, ініціали

12. Перелік документів і матеріалів, що додаються до заяви		Кількість аркушів	Кількість примірників
<input checked="" type="checkbox"/> Примірник твору (форма, в якій представлено твір)	Комп'ютерна програма		1
<input type="checkbox"/> Документ, що підтверджує перехід у спадщину майнового права автора (якщо заявка подається спадкоємцем автора)			
<input type="checkbox"/> Платіжний документ, що підтверджує сплату збору за підготовку до державної реєстрації авторського права на твір			
<input type="checkbox"/> Платіжний документ, що підтверджує сплату збору за оформлення і видачу свідоцтва про державну реєстрацію авторського права на твір			
<input type="checkbox"/> Документ, що підтверджує наявність пільг по сплаті збору			
<input type="checkbox"/> Документ, що свідчить про факт і дату оприлюднення твору (за наявності)			
<input type="checkbox"/> Документ, що підтверджує повноваження довіреної особи (довіреність)			
<input type="checkbox"/> Інші документи, що додаються до заяви			

13. Я, який нижче підписався, підтверджую достатність і достовірність відомостей, вказаних у матеріалах заявки:

Автор(и) Стальмаков Д.Ф.

Прізвище(а), ініціали, підпис(и)

Заявник Стальмаков Д.Ф.

Прізвище, ініціали та підпис особи, яка подає заявку

Дата ► Число ► _____ Місяць ► _____ Рік ► _____

Примітки: _____

Заявку опрацював: _____

Прізвище, ініціали та підпис працівника відділу

Лист подовження до бланку
заяви

9.2. Прізвище, ім'я, по батькові другого автора (псевдонім, за наявності, вказати в дужках) ▼

Дата народження ► Число _____ Місяць _____ Рік _____

Повна поштова адреса, телефон

поштовий індекс _____ Вулиця, номер будинку, назва населеного пункту, район, область,

Суть авторства, авторський вклад у створення твору

Цей твір (частину твору) створено:*

Цей твір (частину твору) створено для оприлюднення:*

		під власним ім'ям	<input type="checkbox"/>
за договором	<input type="checkbox"/>	анонімно	<input type="checkbox"/>
у порядку індивідуальної розробки	<input type="checkbox"/>	під псевдонімом	<input type="checkbox"/>

9.3. Прізвище, ім'я, по батькові третього автора (псевдонім, за наявності вказати в дужках) ▼

Дата народження ► Число _____ Місяць _____ Рік _____

Повна поштова адреса, телефон

поштовий індекс _____ Вулиця, номер будинку, назва населеного пункту, район, область,

Суть авторства, авторський вклад у створення твору

Цей твір (частину твору) створено:*

Цей твір (частину твору) створено для оприлюднення:*

		під власним ім'ям	<input type="checkbox"/>
за договором	<input type="checkbox"/>	анонімно	<input type="checkbox"/>
у порядку індивідуальної розробки	<input type="checkbox"/>	під псевдонімом	<input type="checkbox"/>

* Необхідне позначити "X"