

**Faculty of Engineering Department of  
Informatics Engineering-Software and  
Information Systems Engineering**



## **Documentation For Organizing Trips And Travel**

A senior 1 project report \_submitted to complete the requirements for obtaining a bachelor's degree on informatics engineering

### **Prepared By**

Aya Ahmad Dawood

Sara Hussam Ghraizi

Sedra Almuntha Mazen Layla

### **Supervised By**

Eng. Anas Abdulaziz



الجامعة السورية الخاصة  
SYRIAN PRIVATE UNIVERSITY

كلية الهندسة  
قسم الهندسة المعلوماتية  
هندسة البرمجيات ونظم المعلومات

## Documentation For Organizing Trips And Travel

تقرير مشروع تخرج تم تقديمها لاستكمال متطلبات الحصول على درجة البكالوريوس في هندسة المعلومات.

إعداد:

سارة حسام غريزي

أية احمد داود

سدرة المنتهى مازن ليلي

إشراف:

المهندس أنس عبد العزيز

## **ABSTRACT**

This project presents a smart web-based platform designed to enhance the travel planning experience for users by organizing and centralizing tourism services within a single system.

Traditional travel planning methods often rely on multiple disconnected platforms, resulting in fragmented information, poor coordination, and inefficient trip management. To address these limitations, the project integrates destination exploration, booking services, and structured trip planning into one unified environment.

The platform enables users to register securely, explore destinations and hotels, search flight options through external APIs, and manage transportation and bookings. By combining manual and AI-assisted trip planning with secure payment and notification features, the system provides a comprehensive solution for organizing travel plans. Additional features include booking history management, favorites lists, and administrative tools for managing destinations, hotels, drivers, and services.

## الملخص

يقدم هذا المشروع منصة إلكترونية ذكية تهدف إلى تحسين تجربة تخطيط الرحلات من خلال تنظيم وتوحيد الخدمات السياحية ضمن نظام واحد. تعتمد طرق تخطيط السفر التقليدية على منصات متعددة وغير مترابطة، مما يؤدي إلى تشتت المعلومات وضعف التنسيق وصعوبة إدارة الرحلات. لمعالجة هذه المشكلات، يدمج المشروع استكشاف الوجهات، خدمات الحجز، وتخطيط الرحلات ضمن بيئة موحدة.

تتيح المنصة للمستخدمين التسجيل بشكل آمن، استكشاف الوجهات والفنادق، البحث عن الرحلات الجوية عبر واجهات برمجية خارجية، وإدارة خدمات النقل والجوزات. ومن خلال الجمع بين التخطيط اليدوي والتخطيط الذكي المدعوم بالذكاء الاصطناعي مع الدفع الآمن ونظام الإشعارات، يوفر النظام حلًا متكاملًا لتنظيم الرحلات. كما تشمل الميزات الإضافية إدارة سجل الجوزات، قوائم المفضلة، وأدوات إدارية لإدارة الوجهات والفنادق والسائلين والخدمات.

# TABLE OF CONTENTS

## Contents

ABSTRACT .....	3
الملخص .....	4
TABLE OF CONTENTS .....	5
CHAPTER 1 - INTRODUCTION .....	14
1.1 Introduction: .....	15
1.2 Problem Statement: .....	15
1.3 Proposed System: .....	16
1.4 Project Objective: .....	16
1.5 Report Organization:.....	17
1.6 Summary .....	18
Chapter 2 - Fundamental Concepts and Literature Review .....	19
2.1 Introduction: .....	20
2.2 Fundamental Concepts: .....	20
2.2.1 Smart Tourism Systems: .....	20
2.2.2 Integrated Service Platforms: .....	20
2.2.3 Transportation and Vehicle Management: .....	21
2.2.4 Hotel and Destination Management: .....	21
2.2.5 Trip Planning (Manual and AI-Assisted): .....	21
2.2.6 Online Booking and Payment: .....	22
2.2.7 Notification System: .....	22
2.3: Literature Review: Comparative Analysis of Existing Platforms:.....	22
2.3.1 Flight Network: .....	22
2.3.2 TripAdvisor: .....	23
2.3.3 Booking.com: .....	24
2.3.4 Expedia: .....	24
2.4 Gap Analysis and Innovation: .....	25
2.5 Comparative Table: Feature-by-Feature Analysis:.....	25
2.6 Summary: .....	27
Chapter 3 - Project Management .....	28
3.1.Introduction:.....	29
3.2. Project Management Documents: .....	29
3.2.1 Project Charter: .....	29
3.2.2 Statement of Work (SOW): .....	31

3.2.3 Risk Management:.....	34
3.2.4 Gantt Chart:.....	35
3.2.5 Version Control and Configuration Management: .....	36
3.3. Initial System Study:.....	41
3.3.1 Introduction:.....	41
3.3.2 High-Level Analysis: .....	41
3.3.3 Development Process: .....	44
3.3.4 High-Level System Design:.....	49
3.4 Initial RTM: .....	51
3.5 Initial Test Cases: .....	53
Chapter 4 - System Analysis.....	55
4.1 Introduction:.....	56
4.2 Purpose: .....	56
4.3 Project Scope:.....	56
4.4 Requirements Elicitation:.....	57
4.5 Requirements Table:.....	58
4.5.1 Functional Requirements: .....	58
4.5.2 Non-Functional Requirements: .....	62
4.6 Analysis .....	63
4.6.1 Requirements Modeling:(use case).....	64
Chapter 5 - System Design .....	190
5.1 Introduction:.....	191
5.2 MVC Components: .....	191
5.2.1 View (V): .....	191
5.2.2 Controller (C):.....	191
5.2.3 Model (M): .....	191
5.3 External Services Integration: .....	192
5.4 Database: .....	192
5.5 Class Diagram:.....	193
5.5 ERD Diagram:.....	194
Chapter 6 - Implementation and Testing .....	195
6.1 Introduction:.....	196
6.2 Technologies Used: .....	196
6.2.1 Laravel Framework:.....	196
6.2.2 HTML, CSS, and JavaScript: .....	196
6.2.3 Tailwind CSS:.....	196

6.2.4 PayPal Payment Gateway: .....	196
6.2.5 OpenStreetMap and Geocoding Service:.....	196
6.2.6 OpenWeather API:.....	197
6.2.7 Amadeus API for Flight Searching: .....	197
6.2.8 Groq API for AI Trip Planning:.....	197
6.2.9 MySQL Database:.....	197
6.3 Implementation Overview: .....	197
6.4 Testing Strategy:.....	197
6.5 System Interfaces:.....	198
3.5 Test Cases: .....	236
3.6 Table – RTM V.4 (Requirements Traceability Matrix) .....	242
3.7 Future Work and Conclusion: .....	246
3.7.1 Future Work: .....	246
3.8 Conclusion: .....	247

# Figures:

Figure <sup>1</sup> Gantt Chart .....	35
Figure 2 GitHub Repository for the Travel-Planner -Platform Project .....	37
Figure 3 Git tags used to mark versions v1.0 .....	40
Figure <sup>4</sup> High level usecase .....	43
Figure <sup>5</sup> High level modules diagram.....	51
Figure <sup>6</sup> Traveler UseCase.....	64
Figure <sup>7</sup> Admin Usecase .....	65
Figure <sup>8</sup> Driver UseCase.....	66
Figure <sup>9</sup> Activity Diagram (Register).....	68
Figure <sup>10</sup> Sequence Diagram( Register).....	69
Figure <sup>11</sup> Activity Diagram(Sub-Use Case ) .....	70
Figure <sup>12</sup> Sequence Diagram(Sub-Use Case) .....	71
Figure <sup>13</sup> Activity Diagram( Login).....	72
Figure <sup>14</sup> Sequence Diagram (Login).....	73
Figure <sup>15</sup> Activity Diagram (Logout).....	74
Figure <sup>16</sup> Sequence Diagram(Logout) .....	75
Figure <sup>17</sup> Activity Diagram (Edit Profile) .....	76
Figure <sup>18</sup> Sequence Diagram (Edit Profile ) .....	77
Figure <sup>19</sup> Activity Diagram(View Destinations) .....	78
Figure <sup>20</sup> Sequence Diagram(View Destinations) .....	79
Figure <sup>21</sup> Activity Diagram (View Destination Details) .....	80
Figure <sup>22</sup> Sequence Diagram(View Destination Details .....	80
Figure <sup>23</sup> Activity Diagram (Search Destination).....	81
Figure <sup>24</sup> Sequence Diagram(Search Destination).....	82
Figure <sup>25</sup> Activity Diagram (Add Destination) .....	84
Figure <sup>26</sup> Sequence Diagram(Add Destination) .....	85
Figure <sup>27</sup> Activity Diagram(Edit Destination ) .....	87
Figure <sup>28</sup> Sequence Diagram(Edit Destination) .....	88
Figure <sup>29</sup> Activity Diagram (Delete Destination) .....	89
Figure <sup>30</sup> Sequence Diagram(Delete Destination) .....	89
Figure <sup>31</sup> Activity Diagram(View Hotels) .....	90
Figure <sup>32</sup> Sequence Diagram(View Hotels) .....	91
Figure <sup>33</sup> Activity Diagram (View Hotel Details) .....	92
Figure <sup>34</sup> Sequence Diagram (View Hotel Details).....	92
Figure <sup>35</sup> Activity Diagram (Search Hotels).....	93
Figure <sup>36</sup> Sequence Diagram (Search Hotels).....	94
Figure <sup>37</sup> Activity Diagram(Add Hotel) .....	96
Figure 38 Sequence Diagram (Add Hotel).....	97
Figure <sup>39</sup> Activity Diagram (Edit Hotel) .....	99
Figure <sup>40</sup> Sequence Diagram(Edit Hotel).....	100
Figure <sup>41</sup> Activity Diagram (Delete Hotel).....	101
Figure <sup>42</sup> Sequence Diagram (Delete Hotel) .....	102
Figure <sup>43</sup> Activity Diagram(Book Hotel) .....	104
Figure <sup>44</sup> Sequence Diagram(Book Hotel) .....	105
Figure <sup>45</sup> Activity Diagram(Pay).....	107

Figure <sup>41</sup> Sequence Diagram(Pay).....	108
Figure <sup>44</sup> Activity Diagram(Flight search).....	109
Figure <sup>48</sup> Sequence Diagram(Flight search).....	110
Figure 49Activity Diagram(Add Activity) .....	112
Figure <sup>50</sup> Sequence Diagram(Add Activity) .....	113
Figure <sup>51</sup> Activity Diagram(Edit Activity) .....	115
Figure <sup>52</sup> Sequence Diagram(Edit Activity) .....	116
Figure 53 Activity Diagram(Delete Activity) .....	117
Figure 54 Sequence Diagram(Delete Activity) .....	118
Figure 55 Activity Diagram(View Activities).....	119
Figure 56 Sequence Diagram (View Activities) .....	119
Figure 57 Activity Diagram (View Activity Details) .....	120
Figure 58 Sequence Diagram (View Activity Details) .....	120
Figure 59 activity digram (search activites) .....	122
Figure <sup>60</sup> Sequence Diagram(Search Activities) .....	123
Figure 61 activity digram (add transport service) .....	125
Figure <sup>62</sup> Sequence Diagram( Add transport service).....	126
Figure 63 avctivity digram (edit transport service) .....	128
Figure 64 Sequence Diagram (Edit transport service) .....	129
Figure <sup>65</sup> activity digram (delete transport service) .....	131
Figure <sup>66</sup> Sequence Diagram (Delete transport service:) .....	132
Figure 67 activity diagram (view transport service).....	133
Figure 68 Sequence Diagram (View transport services) .....	134
Figure 69 activity diagram (show linked vehicles) .....	135
Figure 70 Sequence Diagram(View Transport Linked Vehicles) .....	136
Figure <sup>71</sup> Activity Diagram (Add Vehicle).....	137
Figure <sup>72</sup> Sequence Diagram(Add Vehicle) .....	138
Figure <sup>73</sup> Activity Diagram(Edit Vehicle) .....	140
Figure <sup>74</sup> Sequence Diagram(Edit Vehicle) .....	141
Figure <sup>75</sup> Activity Diagram(Delete Vehicle) .....	143
Figure <sup>76</sup> Sequence Diagram(Delete Vehicle) .....	144
Figure 77 Activity Diagrma (Search Available Vehicles) .....	146
Figure <sup>78</sup> Sequence Diagram(Search Available Vehicles) .....	147
Figure 79 Activity Diagram (Book Vehicle) .....	149
Figure <sup>80</sup> Sequence Diagram(Book Vehicle).....	150
Figure <sup>81</sup> Activity Diagram (Add to Favorites) .....	151
Figure <sup>82</sup> Sequence Diagram (Add to Favorites) .....	152
Figure 83 Activity Diagram(Remove From Favorites).....	153
Figure <sup>84</sup> Sequence Diagram(Remove From Favorites) .....	153
Figure 85 Activity Diagram(View Favorites).....	154
Figure <sup>86</sup> Sequence Diagram(View Favorites) .....	155
Figure 87 Activity Diagram(View Traveler's Reservations) .....	156
Figure <sup>88</sup> Sequence Diagram( View Traveler's Reservations) .....	157
Figure 89 Activity Diagram(View System Reservations) .....	158
Figure <sup>90</sup> Sequence Diagram (View System Reservations) .....	159
Figure <sup>91</sup> Activity Diagram (View Drivers).....	160
Figure <sup>92</sup> Sequence Diagram(View Drivers).....	161
Figure <sup>93</sup> Activity Diagram (Accept/Reject Driver:) .....	163

Figure <sup>9</sup> Sequence Diagram-Accept/Reject Driver .....	164
Figure <sup>9</sup> Activity Diagram (Delete Driver) .....	165
Figure <sup>11</sup> Sequence Diagram(Delete Driver) .....	166
Figure <sup>11</sup> Activity Diagram (View Driver Bookings) .....	167
Figure <sup>11</sup> Sequence Diagram(View Driver Bookings) .....	168
Figure <sup>11</sup> Activity Diagram (Search Drivers) .....	169
Figure <sup>11</sup> Sequence Diagram(Search Drivers) .....	170
Figure <sup>11</sup> Activity Diagram(View Assigned Bookings) .....	171
Figure <sup>11</sup> Sequence Diagram(View Assigned Bookings) .....	172
Figure <sup>11</sup> Activity Diagram(Update Reservation Status).....	173
Figure <sup>11</sup> Sequence Diagram (Update Reservation Status).....	174
Figure 105 Activity Diagram (Receive Notifications) .....	176
Figure 106 Sequence Diagram(Receive Notifications).....	177
Figure 107 Activity Diagram(View Trip Plan).....	178
Figure 108 sequence daiagram view trip plan.....	179
Figure 109 Activity Diagram(View Trip Plan Details) .....	180
Figure 110 sequence diagram View Trips Plan details.....	181
Figure 111Activity Diagram(Delete Trip Plan).....	182
Figure 112 sequence diagram Delete Trips Plan .....	183
Figure 113 Activity Diagram(Create manual trip plan) .....	185
Figure 114 sequence diagram Create Manual Trip Plan .....	186
Figure 115 Acyivity Diagram (Create Ai Plan) .....	188
Figure 116 SEQUANCE DIAGRAM Create AI Trip Plan .....	189
Figure 117 Class Diagram.....	193
Figure 118 ERD Diagram.....	194
Figure 119login (interface).....	198
Figure 120 select role page.....	198
Figure 121 register traveler page .....	199
Figure 122 register driver step1 .....	199
Figure 123 Register Driver – step2 .....	200
Figure 124 Main Dashboard.....	200
Figure 125 profile management .....	201
Figure 126 profile management .....	201
Figure <sup>111</sup> Traveler Interfaces-destinations .....	202
Figure 128 Destination Details interface .....	203
Figure 129 Traveller interfaces Hotels.....	204
Figure 130 Hotel Details interface .....	206
Figure 131 book hotel interface .....	207
Figure 132 flights interface .....	207
Figure 133 flights results interface .....	208
Figure 134 activities interface .....	208
Figure 135 Activity Details interface .....	209
Figure 136 Transport Services interface .....	209
Figure 137 Order Vehicle interface .....	210
Figure 138 Vehicles Results interface .....	211
Figure 139 Book Vehicle interface .....	212
Figure 140 Notifications interface .....	212
Figure 141 Favorites interface.....	213

Figure 142 reservation interface.....	214
Figure 143 Trips interface .....	214
Figure 144 Trip Details interface.....	215
Figure 145 create Trip interface .....	215
Figure 146 create ai Trip interface.....	216
Figure 147 Create Manual Trip – Step1 -interface.....	216
Figure 148 Create Manual Trip – Step2 interface .....	217
Figure 149 Create Manual Trip – Step3 interface.....	218
Figure 150 Create Manual Trip – Step4 interface.....	219
Figure 151 Create Manual Trip – Step5 interface.....	219
Figure 152 Destinations interface.....	220
Figure 153 Create Destination interface .....	220
Figure 154 Edit Destination interface.....	221
Figure 155 Hotels interface .....	222
Figure 156Create Hotel interface .....	223
Figure 157 Edit Hotel interface.....	224
Figure 158 activites interface.....	225
Figure 159 Add Activity interface .....	226
Figure 160 Edit Activity interface .....	227
Figure 161Transport Services interface.....	228
Figure 162 Create Transport Service interface.....	228
Figure 163 Edit Transport Service interface .....	229
Figure 164 Vehicles interface.....	229
Figure 165 Create Vehicle interface.....	230
Figure 166 Edit Vehicle interface .....	231
Figure 167 Reservations interface .....	232
Figure 168 Hotel Reservations interface .....	232
Figure 169 Drivers interface .....	233
Figure 170 Driver's Reservations interface .....	233
Figure 171 Pending Reservations: interface .....	234
Figure 172 Completed Reservations interface.....	234
Figure 173 Notifications interface .....	235

# Tables:

Table 1 Gap Analysis and Innovation .....	25
Table 2 Comparative Table: Feature-by-Feature Analysis .....	27
Table 3 Roles and responsibilities .....	30
Table 4 Risk Management.....	34
Table 5 System Actors.....	42
Table 6 Initial RTM.....	52
Table 7 Initial Test Cases .....	54
Table 8 Functional Requirements.....	62
Table 9 Non-Functional Requirements .....	62
Table 10 Register .....	67
Table 11 Sub-Use Case: Submit for Admin Approval.....	70
Table 12 Login .....	71
Table 13 Logout.....	73
Table 14 Edit Profile .....	75
Table 15 View Destinations .....	78
Table 16 View Destination Details .....	79
Table 17 Search Destination.....	81
Table 18 View Hotels.....	90
Table 19 View Hotel Details .....	91
Table 20 Search Hotels .....	93
Table 21 Reserve Hotels.....	103
Table 22 Pay .....	106
Table 23 Flight Search.....	108
Table 24 Add Activity.....	111
Table 25 Edit Activity.....	114
Table 26 Delete Activity.....	117
Table 27 View Activities .....	118
Table 28 View Activity Details .....	120
Table 29 Search & Filter Activities .....	121
Table 30 Add transport service.....	124
Table 31 Edit transport service.....	127
Table 32 Delete transport service.....	130
Table 33 View transport services.....	133
Table 34 View Transport Linked Vehicles.....	135
Table 35 Add Vehicle.....	136
Table 36 Edit Vehicle.....	139
Table 37 Delete Vehicle .....	142
Table 38 Search Available Vehicles .....	145
Table 39 Reserve Vehicle.....	148
Table 40 Add to Favorites.....	151
Table 41 Remove From Favorites .....	152
Table 42 View Favorites .....	154
Table 43 View Traveler's Reservations .....	155
Table 44 View System Reservations .....	158
Table 45 View Drivers .....	160
Table 46 Accept/Reject Driver:.....	162

Table 47 Delete Driver .....	165
Table 48 View Driver Bookings .....	167
Table 49 Search & Filter Drivers: .....	169
Table 50 View Assigned Bookings.....	171
Table 51 Update Reservation Status.....	173
Table 52 View Trips plans .....	178
Table 53 View Trip Plan Details.....	180
Table 54 Delete Trip Plan .....	182
Table 55 Create Manual Trip Plan .....	184
Table 56 Create AI Trip Plan .....	187
Table 57 Test Cases .....	241
Table <sup>o^</sup> RTM (Requirements Traceability Matrix).....	245

# **CHAPTER 1 - INTRODUCTION**

## **1.1 Introduction:**

In this chapter, we introduce the Smart Tourism and Travel Management System, a digital platform designed to simplify and enhance the travel experience for users by integrating multiple tourism-related services into a single application.

The system enables users to search for destinations, book hotels, explore flight options, request transportation services, and create personalized travel plans—all in one place. Additionally, it provides administrative tools that allow system managers to maintain, update, and monitor various travel-related data such as destinations, hotels, vehicles, and drivers.

The platform aims to make tourism more efficient, accessible, and interactive for both travelers and administrators by leveraging technology to connect users with essential travel services in an organized and user-friendly environment.

## **1.2 Problem Statement:**

The tourism industry faces several challenges related to fragmented information and limited integration among travel services. Users often need to switch between multiple platforms to plan a single trip—one for flights, another for hotels, and yet another for transportation—making the process time-consuming and confusing.

Moreover, travelers lack a unified way to organize their trips, receive reminders, or explore destination activities interactively. Administrators, on the other hand, often struggle to manage large datasets of destinations, hotels, and transportation services efficiently, leading to data inconsistencies and poor customer experience.

To address these challenges, there is a need for a centralized, intelligent tourism management platform that streamlines the entire travel process—covering destination browsing, hotel bookings, real time flights options, transportation management, and personalized trip planning—while providing a smooth administrative experience for system managers.

### **1.3 Proposed System:**

The proposed Smart Tourism and Travel Management System is envisioned as a comprehensive digital platform that integrates all key tourism services into a single, cohesive environment.

Users can register and log into the system, browse and filter destinations, view detailed hotel and flight information, make bookings, and complete secure online payments. The system also supports personalized features such as favorites lists, AI-assisted trip planning, and automated notifications for booking reminders.

Administrators can efficiently manage all travel-related entities, including destinations, hotels, activities, transportation services, vehicles, and drivers. They can also oversee bookings and ensure system data remains accurate and up-to-date.

By combining both user and administrative functionalities, the system ensures a seamless interaction between travelers and service providers, promoting efficiency and satisfaction on both sides.

### **1.4 Project Objective:**

The primary objective of the Smart Tourism and Travel Management System is to develop a comprehensive and user-friendly platform that centralizes the tourism experience for both travelers and administrators.

#### **The key goals include:**

- Providing secure user and admin account management (registration, login, and profile editing).
  - Enabling users to search for and explore destinations, hotels, activities, transport and flight options easily.
  - Offering hotel booking and secure online payment functionalities.
  - Allowing administrators to manage destinations, hotels, transportation, and activities effectively.
- 
- Supporting transportation management, including vehicle and driver assignments.

- Implementing a favorites list feature for users to save destinations and hotels.
- Introducing Manual and AI-based trip planning to assist users in building personalized itineraries.
- Delivering a notification system to remind users of upcoming bookings and inform drivers about new reservations.
- Ensuring a modern, responsive, and user-friendly interface suitable for desktop devices.

## **1.5 Report Organization:**

**The remainder of this report is organized as follows:**

Chapter 1: Introduction

Chapter 2: Literature Review and Background

Chapter 3: Project Management and Feasibility Study

Chapter 4: System Analysis

Chapter 5: System Design

Chapter 6: System Implementation

Chapter 7: System Testing and Evaluation

Chapter 8: Conclusion and Future Work

## **1.6 Summary**

This chapter introduced the motivation, problem definition, and goals of the Smart Tourism and Travel Management System. It highlighted the issues faced by travelers and administrators in managing fragmented tourism data and proposed an integrated solution to unify these services within one intelligent system.

The next chapters will delve into the technical, analytical, and managerial aspects of the system, providing a detailed overview of its design, implementation, and evaluation.

## **Chapter 2 - Fundamental Concepts and Literature Review**

## **2.1 Introduction:**

In the modern era of digital transformation, tourism and travel management have evolved from traditional manual systems into sophisticated digital platforms that provide convenience, personalization, and accessibility to global users. The Smart Tourism and Travel Management System stands at the intersection of technology and hospitality, aiming to streamline the travel experience through an integrated environment that combines destinations, hotels, flights, transportation, and trip planning in one platform.

This chapter explores the fundamental concepts underpinning the project and provides a literature review of existing tourism systems and related technologies. It highlights the technological advancements, identifies limitations in current solutions, and defines the conceptual framework that guides the development of this system.

## **2.2 Fundamental Concepts:**

### **2.2.1 Smart Tourism Systems:**

Smart tourism focuses on using modern technology and intelligent systems to improve travel experiences and simplify the management of tourism resources. These systems connect travelers with essential services such as destinations, accommodations, transportation, and trip planning, allowing users to make decisions more easily and efficiently.

The Smart Tourism and Travel Management System applies these principles by offering a unified platform where users can search for destinations, book hotels, explore transportation options, and organize complete travel plans in one place. The main goal is to create an ecosystem that links travelers, service providers, and administrators through a single, well-structured system.

### **2.2.2 Integrated Service Platforms:**

Modern travel applications often specialize in a single service—such as hotel booking, flight search, or transport management. While effective, this fragmentation forces users to rely on multiple platforms to complete their travel plans.

An integrated platform, by contrast, consolidates all related services within one environment, reducing complexity and enhancing the user experience. The proposed system integrates destination management, hotel reservations, flight

search via API, and transportation services under a single interface, ensuring smooth interaction between the different modules.

This approach enhances operational efficiency, improves data consistency, and provides users with a cohesive and intuitive experience.

### **2.2.3 Transportation and Vehicle Management:**

Transportation is a critical component of the tourism ecosystem. Efficient vehicle and driver management ensures reliability, safety, and customer satisfaction. The proposed system includes dedicated modules for transportation services, allowing administrators to add, modify, and delete transport services, manage vehicle data, and assign drivers to vehicles.

For users, the system provides features to browse, filter, and request transportation services, creating a bridge between service providers and travelers. Notifications are also integrated to inform both drivers and users about upcoming or newly booked rides, improving coordination and punctuality.

### **2.2.4 Hotel and Destination Management:**

Hotels and destinations are the backbone of any tourism platform. The system provides administrators with tools to add, edit, and delete destinations and hotels, as well as attach related information such as photos, and activities.

For users, the platform offers advanced search and filtering options, enabling them to explore destinations by name, or location, and to view detailed hotel information including ..... This structure allows for personalized exploration and simplifies the decision-making process.

### **2.2.5 Trip Planning (Manual and AI-Assisted):**

Trip planning is one of the core features of the system, designed to help users organize their journeys according to their preferences, time, and budget. The platform provides two main modes of planning: manual and AI-assisted.

In manual planning, users can create their travel itineraries step by step by selecting destinations, hotels, user's flight information, and activities of their choice. This mode offers full flexibility and control for travelers who prefer to design their trips independently.

On the other hand, AI-assisted trip planning leverages Artificial Intelligence to generate optimized travel plans automatically. \*\*\*\*\*The system analyzes

user preferences, budget, available destinations, and travel duration to suggest efficient and personalized itineraries.

Together, these two modes—manual and AI-assisted—bridge the gap between customized and automated planning, offering users both creative freedom and intelligent recommendations for an enhanced travel experience.

### **2.2.6 Online Booking and Payment:**

The system enables secure online booking and electronic payments for hotel reservations and transportation. It ensures data privacy, confirmation emails, and an integrated record of all user transactions for transparency and convenience.

### **2.2.7 Notification System:**

A built-in notification mechanism enhances communication between users and service providers. Users receive reminders about upcoming bookings, while drivers are alerted to new transport requests—ensuring timely coordination and reliability.

## **2.3: Literature Review: Comparative Analysis of Existing Platforms:**

Travel and trip planning platforms vary in terms of the services they offer, user experience design, and the level of automation provided in itinerary creation. The following is a comparative analysis of well-known platforms and how our proposed system addresses their limitations.

### **2.3.1 Flight Network:**

*Strengths:*

- Provides a wide range of flight options from multiple airlines.
- User-friendly search filters (price, date, destination, travelers).
- Allows travelers to compare flights easily.

*Weaknesses:*

- Focused mainly on flight booking, offering limited trip planning integration.
- Does not provide activity planning or local transportation management.
- No personalized travel itineraries.

*Similarities:*

- Both systems allow users to search and compare flights based on travel dates, destination, and number of passengers.

*Our Contribution:*

We expand beyond flight comparison to include **destination exploration, hotel booking, activity scheduling, and transportation services** within a unified platform.

### **2.3.2 TripAdvisor:**

*Strengths:*

- Massive database of user reviews and travel experiences.
- Provides hotel, restaurant, and activity recommendations.
- Strong reputation-based filtering and rating system.

*Weaknesses:*

- Does not offer integrated trip-planning workflows.
- Transportation and local driver services are not managed internally.
- The platform relies heavily on third-party providers.

*Similarities:*

- Both systems allow users to browse **activities and destinations** and view detailed descriptions.

*Our Contribution:*

We provide an integrated system where:

- Activities are **managed internally by administrators**, not only reviewed.
- Transportation and drivers are **bookable directly within the platform**.
- Users receive **AI-assisted trip planning suggestions** based on their preferences.

### **2.3.3 Booking.com:**

*Strengths:*

- Extensive hotel database with competitive pricing.
- Reliable booking system with secure payment integration.
- Provides user ratings and filters based on property features.

*Weaknesses:*

- Does not include activity planning.
- No transportation or local driver assignment module.
- No AI-based customized trip itinerary.

*Similarities:*

- Both platforms allow hotel browsing, filtering, and booking.

*Our Contribution:*

We connect hotel bookings with **destination activities, transportation, and custom itineraries** to create an end-to-end trip experience.

### **2.3.4 Expedia:**

*Strengths:*

- Offers bundled travel packages (flight + hotel + car rental).
- Wide availability of international booking services.

*Weaknesses:*

- Car rentals are outsourced; does not manage **drivers or vehicle assignments**.
- Does not provide detailed destination-specific activity planning.

*Similarities:*

- Both systems support multi-service travel management.

*Our Contribution:*

Our system supports **internal transportation fleet management**, including vehicles, drivers, pricing, availability, and assignment.

## 2.4 Gap Analysis and Innovation:

Limitation in Existing Platforms	Our System's Solution
<b>Lack of integrated activity management</b>	Built-in activity creation, categorization, and scheduling
<b>No vehicle and driver management</b>	Full transportation module with cars, drivers, and availability control
<b>No AI-assisted trip planning workflows</b>	AI engine generates optimized itineraries based on budget, dates, and preferences
<b>Scattered booking systems</b>	Unified booking and electronic payment gateway

Table 1 Gap Analysis and Innovation

## 2.5 Comparative Table: Feature-by-Feature Analysis:

Feature / Platform	Our System	Booking.com	TripAdvisor	Flight Network	Expedia
<b>Destination Exploration</b>	✓	X (primarily hotels)	✓	X	✓ (limited suggestions)
<b>Hotel Browsing &amp; Filtering</b>	✓	✓	✓	X	✓
<b>Direct Hotel Booking</b>	✓	✓	X (external redirects)	X	✓
<b>Flight Search &amp; Comparison</b>	✓ (via external API)	✓	X (indirect)	✓	✓
<b>Transportation Service Management (Vehicles &amp; Drivers)</b>	✓	X	X	X	X (car rentals only, external)

<b>In-App Transportation Booking</b>	✓	X	X	X	X
<b>Activity Management</b>	✓ (admin-curated)	X	✓ (reviews only)	X	✓ (limited integration)
<b>User Reviews &amp; Ratings</b>	X	✓	✓ (large community base)	X	✓
<b>Favorites / Wishlist</b>	✓	✓	✓	X	✓
<b>Manual Trip Planning (Custom Itinerary)</b>	✓	X	X	X	X
<b>AI-Assisted Trip Planning</b>	✓	X	X	X	X
<b>Secure Online Payment</b>	✓	✓	X (no direct payment)	✓	✓
<b>Centralized All-in-One Travel Workflow</b>	✓	X	X	X	X
<b>Internal Driver Assignment System</b>	✓	X	X	X	X
<b>Transportation Notifications (User ↔ Driver)</b>	✓	X	X	X	X
<b>Multi-Language Support</b>	X	✓	✓	✓	✓
<b>Currency Conversion Support</b>	X	✓	✓	✓	✓
<b>Mobile App Availability</b>	X	✓	✓	✓	✓
<b>Price Alerts / Deal Tracking</b>	X	✓	X	✓	✓

<b>Loyalty &amp; Rewards Program</b>	X	✓	X	X	✓
<b>Map-Based Search &amp; Navigation</b>	X	✓	✓	X	✓
<b>Social Community Discussions</b>	X	X	✓	X	X

Table 2 Comparative Table: Feature-by-Feature Analysis

## 2.6 Summary:

The reviewed platforms contribute valuable features such as hotel booking, flight comparison, and user reviews. However, **none provide a complete, centralized solution** that integrates:

- Destination browsing
- Activity scheduling
- Manual & AI-assisted trip planning
- Transportation & driver management
- Secure booking & payment
- Notification and coordination tools

Our system fills these gaps by offering **a unified tourism management platform** serving both users and administrators, enhancing travel planning efficiency, personalization, and convenience.

## **Chapter 3 - Project Management**

### **3.1.Introduction:**

In this chapter, we explore the project management phase, which plays a critical role in ensuring the success of the Smart Tourism and Travel Management System. This phase focuses on defining the project objectives, organizing resources, and estimating the time and effort required for development.

The project is divided into manageable tasks, each assigned to specific team members to ensure systematic progress and effective coordination throughout all stages of analysis, design, implementation, and testing.

### **3.2. Project Management Documents:**

#### **3.2.1 Project Charter:**

Project Title: Smart Tourism and Travel Management System

Project Start Date: October,18, 2025

Project Finish Date:January ,28, 2026

Project Manager: Eng. Anas Abdulaziz

#### **Project Objectives:**

The objective of the Smart Tourism and Travel Management System is to develop an integrated platform that simplifies the travel process for users and administrators by combining all tourism-related services in one system.

#### **The system 's main objectives are to:**

- Provide a unified platform for exploring destinations, hotels, flights, and activities.
- Enable hotel and transport booking with secure online payment.
- Allow users to create and manage manual and AI-assisted trip plans.
- Provide administrators with full control over system data (destinations, hotels, vehicles, drivers and activities).
- Integrate notification systems to keep users and drivers updated.
- Ensure efficient user authentication and role-based access management.

## **Approach:**

1. Define the overall scope and objectives of the tourism system.
2. Gather and analyze both functional and non-functional requirements.
3. Perform detailed system analysis to identify actors and system interactions.
4. Design system architecture and database structure.
5. Implement the system incrementally using an Agile approach (sprints).
6. Test each module after completion to ensure quality and functionality.
7. Document all development stages and maintain version control.

## **Roles and responsibilities:**

Name	Role	Responsibility
Eng. Anas Abdulaziz	Project Manager and Supervisor	<b>Project supervising and monitoring</b>
Aya Dawood	Software Engineer (Database & Backend)	<b>Database design and management, defining entities and relationships, database optimization</b>
Sara Ghraziz	Software Engineer (Analysis & Modeling)	<b>System analysis, requirements analysis, and system modeling (use case diagrams, class diagrams)</b>
Sedra Layla	Software Engineer (Frontend & Project Management)	<b>User interface design, usability and layout consistency, project coordination and progress tracking</b>
Aya Dawood, Sara Ghraziz, Sedra Layla	Software Engineer (Backend – Shared Responsibility)	<b>Backend development, implementation of controllers, models, services, and system integration</b>

Table 3 Roles and responsibilities

### **3.2.2 Statement of Work (SOW):**

---

#### **1. Project Title:** Organizing Trips And Travel

#### **2. Introduction:**

This Statement of Work (SOW) outlines the objectives, scope, deliverables, requirements, and assumptions related to the development of the Smart Tourism and Travel Management System. The system is designed to provide an all-in-one platform that enhances the travel experience for users by integrating destinations, hotel booking, flight search, and transportation management, supported by both manual and AI-based trip planning.

---

#### **3. Purpose:**

The purpose of this project is to create a centralized digital platform that simplifies travel management for users and administrators. The system aims to connect all tourism services—destinations, hotels, flights, and transport—while enabling smooth booking, secure payment, and intelligent trip planning within one unified environment.

---

#### **4. Scope:**

The Smart Tourism and Travel Management System will provide the following key modules and functionalities:

User Management: Account creation, login, profile editing, and logout.

Destination Management: Browsing, searching, and viewing destination details.

Hotel Management: Browsing, filtering, and booking hotels with detailed information.

Flight Search Integration: Retrieving available flight data via external APIs.

Transportation Services: Managing vehicles, drivers, and booking requests.

Trip Planning: Supporting both manual and AI-assisted trip creation.

**Notifications:** Alerts for users and drivers regarding bookings and reminders.

**Payments:** Secure online payment gateway integration.

This scope covers all main functional and administrative features and defined in the system's requirements.

---

## **5. Project Goals:**

- G1: Develop a user-friendly and responsive tourism management platform.
  - G2: Integrate hotel, flight, and transport services into one unified system.
  - G3: Enable users to plan trips manually or using AI recommendations.
  - G4: Provide administrators with tools for efficient management of data and services.
  - G5: Ensure secure booking, payment, and notification processes.
- 

## **6. Deliverables:**

- D1: Project Plan
  - D2: Software Requirements Specification (SRS)
  - D3: Software Design Document (SDD)
  - D4: Functional Prototype and Web Application
  - D5: Final Project Report and Presentation
- 

## **7. Technical Requirements:**

**Frontend:** Tailwind CSS and JavaScript

**Backend:** php laravel framework

**Database:** MySql

**API Integration:** Amadeus(flights search), Open weather API, Groq (AI trip planning), OpenStreetMap and Geocoding Service.

**Payment Gateway:** Secure online payment system (PayPal)

## **8. Assumptions:**

All team members and supervisors are available during the development timeline.

Reliable internet connection and hosting environment are provided.

All necessary hardware and software resources required for development and deployment are provided.

---

## **9. Project Resources:**

- **Eng. Anas Abdulaziz:** Project Manager & Supervisor
  - **Aya Dawood:** Backend and Frontend developer
  - **Sedra Layla:** Backend and Frontend developer
  - **Sara Gharizi:** Backend and Frontend developer
- 

## **10. Project Approach:**

The Smart Tourism and Travel Management System follows an Agile Scrum methodology to ensure flexibility, continuous improvement, and iterative delivery throughout the development process.

The project approach includes the following main steps:

Define the project scope and objectives clearly based on stakeholder needs.

Break down the project into sprints, with each sprint focusing on specific modules (e.g., user management, hotel booking, transportation, AI trip planning).

Conduct regular sprint reviews to gather feedback and ensure alignment with project goals.

Implement continuous testing after each sprint to validate functionality and ensure high quality.

Maintain comprehensive documentation for each development phase, including design, implementation, and testing reports.

Collaborate closely between frontend and backend teams to achieve a unified and seamless system design.

---

## 11. Schedule:

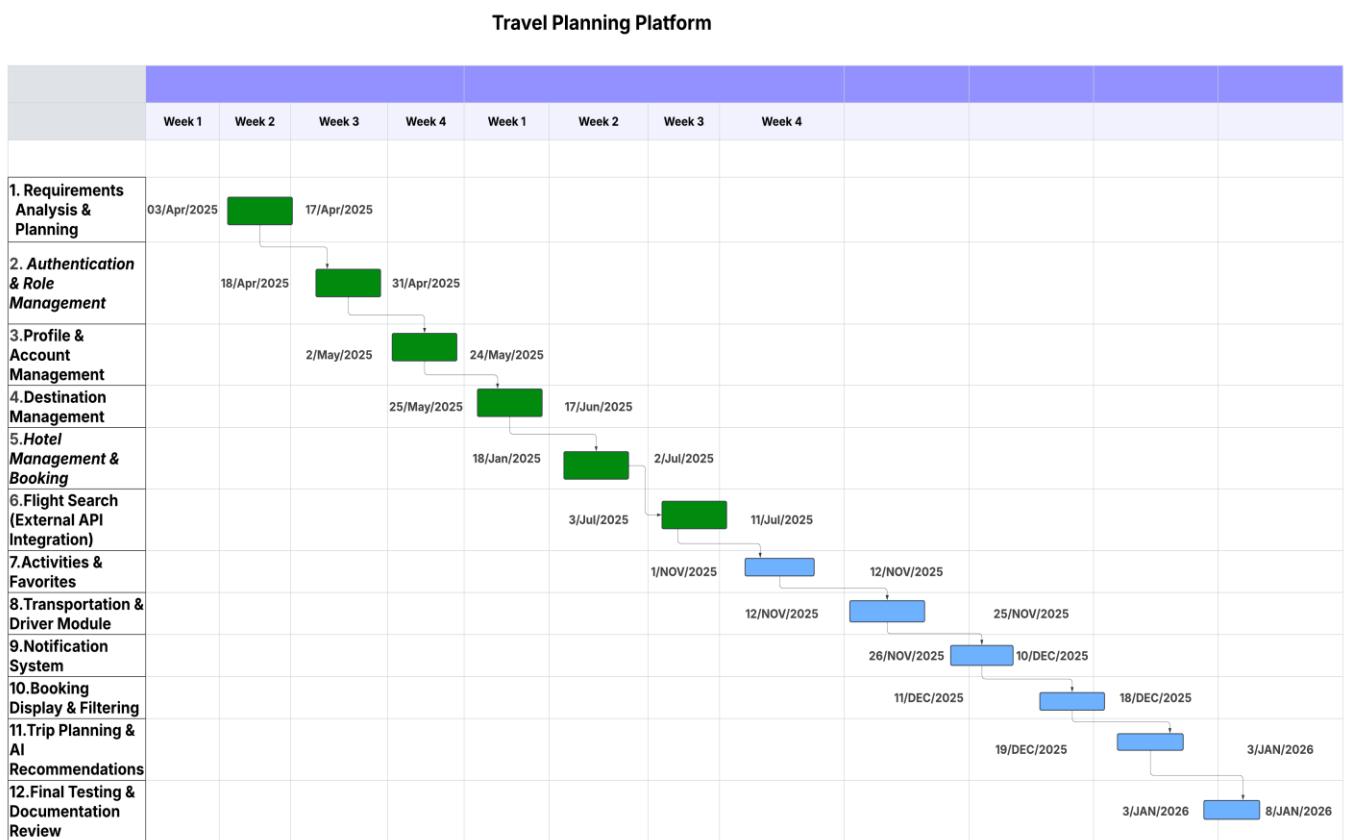
- **Project Start :** October 18, 2025
  - **Project Finish :** January 25, 2026
  - **First Seminar :** November 15, 2025
  - **Second Seminar :** November 27, 2025
  - **Technical Interview :** January 10, 2026
  - **Final Seminar :** January 31, 2026
- 

### 3.2.3 Risk Management:

Risk ID	Risk Title	Description	State	Impact	Mitigation Plan	Likelihood	Reported Date	Tracking Frequency
TRP_RSK_1001	External API Dependency	The system relies on external APIs for flight. Any API downtime may disrupt search and booking features.	Open	High	Implement caching and graceful fallback messages when API is unavailable.	Medium	14/11/2025	Weekly
TRP_RSK_1002	Payment Security Risks	Online payment processing may introduce risks related to secure data handling and user trust.	Open	High	Integrate trusted payment gateways, use SSL encryption, and ensure secure storage of sensitive data.	Low	14/11/2025	Weekly
TRP_RSK_1004	Data Scalability Issues	As the system grows with more destinations and hotels, database queries may become slower.	Open	High	Optimize database indexing, apply pagination, and use performance monitoring tools.	Medium	14/11/2025	Weekly

Table 4 Risk Management

### 3.2.4 Gantt Chart:



Gantt Chart | Figure

### **3.2.5 Version Control and Configuration Management:**

#### ***3.2.5.1 Project Description:***

The Smart Tourism and Travel Management System is a web-based platform developed using the Laravel framework.

The main objective of the project is to provide an integrated system for travel planning and tourism management, including destination exploration, hotel booking, trip planning, cost calculation, and secure payment processing within a single platform.

The system is designed to support modular growth and future extensibility, allowing additional services and features to be integrated without major architectural changes.

It follows a structured backend-driven approach, where the business logic is handled through Laravel controllers, services, and models, while the frontend is implemented using Laravel Blade templates combined with Tailwind CSS and JavaScript.

The project emphasizes maintainability, scalability, and clear separation of concerns, making it suitable for incremental development and future enhancements.

#### ***3.2.5.2 Git Repository Structure:***

Version control for the project is managed using Git and hosted on GitHub. All source code for the system is maintained in a single centralized repository that contains the backend logic, frontend views, configuration files, and database-related components.

Project Repository:

- Repository Name: Travel-planner-platform
- Repository URL: <https://github.com/sedraLa/Travel-planner-platform>
- Purpose: Contains the complete source code of the Smart Tourism and Travel Management System, including backend logic, frontend templates, and configuration files.

The repository follows the standard Laravel project structure, which includes:

- Controllers for handling application logic

- Models for database interaction
- Blade views for frontend rendering
- Routes for managing HTTP requests
- Configuration and environment setup files

This unified repository structure simplifies version control, ensures consistency across system components, and facilitates easier tracking of changes throughout the project lifecycle.

The screenshot shows the GitHub repository page for 'Travel-planner-platform'. At the top, there's a navigation bar with links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the repository name 'Travel-planner-platform' is displayed along with its status as 'Public'. There are buttons for Watch (1), Fork (0), and Star (0). The main content area shows a list of commits. The first commit is from 'ayad2003' merging the 'main' branch. Other commits include 'initial commit', 'implement groq trip planning service and controller and te...', 'edit hotel', 'move hotel from branch to branch', 'edit hotel', and 'implement show all trips and show trip details'. On the right side, there's an 'About' section with a detailed description of the project: 'A smart web platform for planning trips and vacations. Users can explore tourist destinations, check real-time flight options via API, and book hotels directly with online payment support. Admins can manage destinations and hotel data through a dedicated dashboard.' Below the description are links for Readme, Activity, Stars (0), Watching (1), Forks (0), and a Report repository button.

Figure 2 GitHub Repository for the Travel-Planner -Platform Project

### **3.2.5.3 Branching and Merging Strategy:**

A simplified and consistent branching strategy was adopted to support incremental development and maintain system stability throughout the project.

The following branches are used in the repository:

- production: Contains the stable and production-ready versions of the system.
- development: Used for ongoing development, feature integration, and testing.

- feature/\*: Temporary branches created for implementing specific features or modules, such as destination management, hotel booking, or trip planning.

Branch Flow:

- Feature branches are created from the development branch.
- Each feature is implemented and tested independently.
- After completion, feature branches are merged back into the development branch.

This strategy ensures that new features and system changes do not affect the stable production version until they are fully implemented and validated. It also allows controlled integration of functionalities and clear separation between development and release-ready code.

#### ***3.2.5.4 Team Members and Responsibilities:***

The project was developed by a team of three members: **Aya, Sara, and Sedra**. All team members collaboratively participated in the overall system development process, including backend implementation, feature integration, testing, and version control management.

The responsibilities within the team were distributed as follows:

- **Aya – Database Design and Management:**  
Responsible for designing and managing the database structure, including defining entities, relationships, and constraints, as well as ensuring data consistency and integrity. Aya contributed to database migrations and supported backend development through database structure refinement.
- **Sara – System Analysis and Modeling:**  
Responsible for system analysis and documentation, including requirements analysis and system modeling. This involved preparing diagrams such as use case diagrams, class diagrams, and other analytical models that describe the system structure and behavior.
- **Sedra – Frontend Design and Project Management:**  
Responsible for designing the user interfaces of the system, focusing on layout, usability, and visual consistency across the platform. Sedra also handled project management activities, including task coordination, progress tracking, and maintaining overall project organization.

- **Backend Development (Shared Responsibility):**

All team members participated in backend development using the Laravel framework. This included implementing controllers, models, services, and integrating core system modules such as destination management, hotel booking, trip planning, cost calculation, and payment handling.

All team members contributed to version control activities using Git and GitHub. Branching, committing, merging, and repository management were handled collaboratively to ensure organized development, clear change tracking, and a maintainable project structure.

#### ***3.2.5.5 Development Workflow:***

The development workflow follows an incremental and task-based approach to ensure system stability and organized progress throughout the project lifecycle.

The workflow can be summarized as follows:

1. A new feature or development task is identified.
2. A feature branch is created from the development branch.
3. The feature is implemented and tested locally.
4. Changes are committed with clear and descriptive commit messages.
5. The feature branch is merged into the development branch.

This workflow ensures that each feature is developed independently, tested before integration, and only stable versions are merged to the development branch.

#### ***3.2.5.6 Tags and Versioning:***

Git tags are used to mark important project milestones and stable development versions of the system.

Tagging provides a clear reference for specific system states and supports tracking the evolution of the project across its academic phases.

At the current stage of the project, no GitHub Releases have been created. Versioning has been limited to tagging key academic milestones rather than publishing formal releases.

### **v1.0-junior – Junior Project Version**

Version v1.0-junior represents the completed and stable version of the Smart

Tourism and Travel Management System submitted as the junior project. This tag marks the completion of the core functionalities required for academic evaluation and reflects a functional and integrated system state.

This version includes:

- Destination exploration and management
- Hotel booking functionality
- Flights searching
- Authentication system
- Integrated frontend interfaces
- Backend implementation using the Laravel framework
- Stable database structure

#### Future Versioning – Senior Project Phase

A new Git tag will be created to represent the upcoming Senior Project 1 phase. This future tag will mark the extended and improved version of the system currently under development, reflecting enhancements, refinements, and additional features beyond the junior project scope.

This planned tagging approach ensures clear distinction between academic project phases and supports organized tracking of system progress.

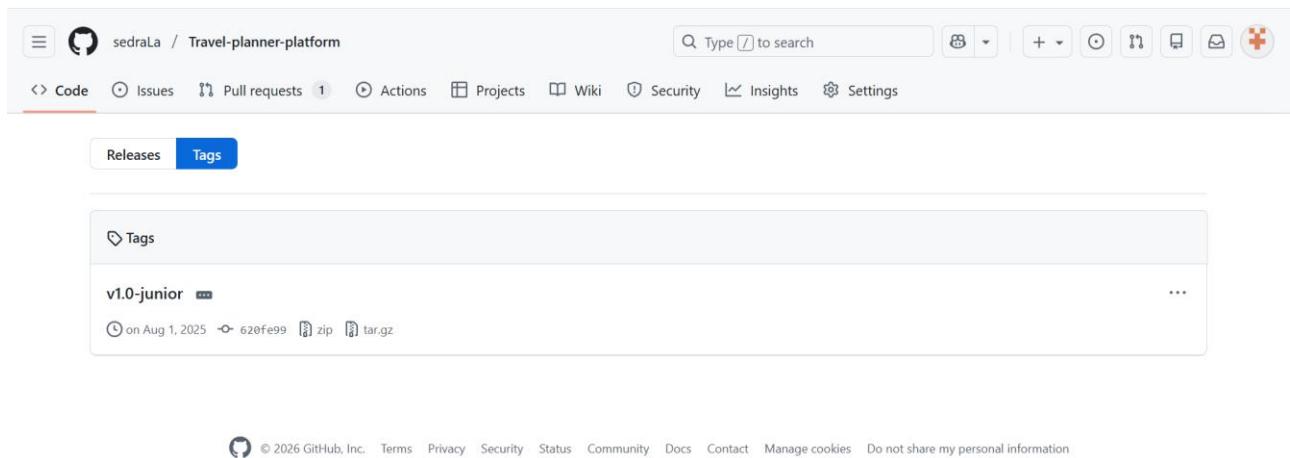


Figure 3 Git tags used to mark versions v1.0

### **3.2.5.7 Summary:**

In conclusion, effective project management and version control practices play a critical role in the successful development of the Smart Tourism and Travel Management System.

The use of Git and GitHub enabled organized development, clear tracking of changes, controlled integration of features, and reliable release management. Overall, these practices contributed to delivering a stable, maintainable, and extensible software system that meets the project objectives.

## **3.3. Initial System Study:**

### **3.3.1 Introduction:**

This chapter presents the initial analysis and design of the Smart Travel Planning Platform, focusing on how the system architecture, functional requirements, and development methodology align with user needs and technical efficiency.

The platform provides an interactive environment for users to explore destinations, search and book hotels, check available flights, manage transportation services, and plan trips in an organized and intelligent way.

This section also provides an overview of the main actors, system decomposition, and UML diagrams that illustrate the high-level structure of the system.

### **3.3.2 High-Level Analysis:**

The Smart Travel Planning Platform is designed as a comprehensive web-based application that simplifies travel planning by integrating hotel booking, flight search, and destination exploration within one system.

The system follows the MVC architecture implemented using Laravel Framework for the backend and Blade + TailwindCSS + JavaScript for the frontend.

### **It provides:**

A seamless experience for travelers to plan, manage, and book their trips.

An administrative dashboard for managing destinations, hotels, activities, drivers, vehicles, and transport services.

### **3.3.2.1 Actors and Interactions Overview:**

This section presents an overview of the main system actors and their roles within the Smart Travel Planning Platform.

Each actor interacts with the system through different functionalities that align with their goals and permissions

#### **Primary Actors:**

Actor	Description	Goals
User	A registered traveler who uses the platform to explore destinations, plan trips, and make bookings.	Search and explore destinations, flights and activities, book hotels and transportation, create trip itineraries, manage profile.
Driver	A transportation service provider assigned by the system to handle user ride requests.	Receive trip requests, update ride status.
Admin	System administrator responsible for managing system content and service resources.	Manage destinations, hotels, activities, drivers, pricing, monitor bookings, oversee platform operations.

Table 5 System Actors

### **3.3.2.2 Use Case Diagram:**

use-case diagrams model the behavior of a system and help to capture the interactions between system actors and the requirements.

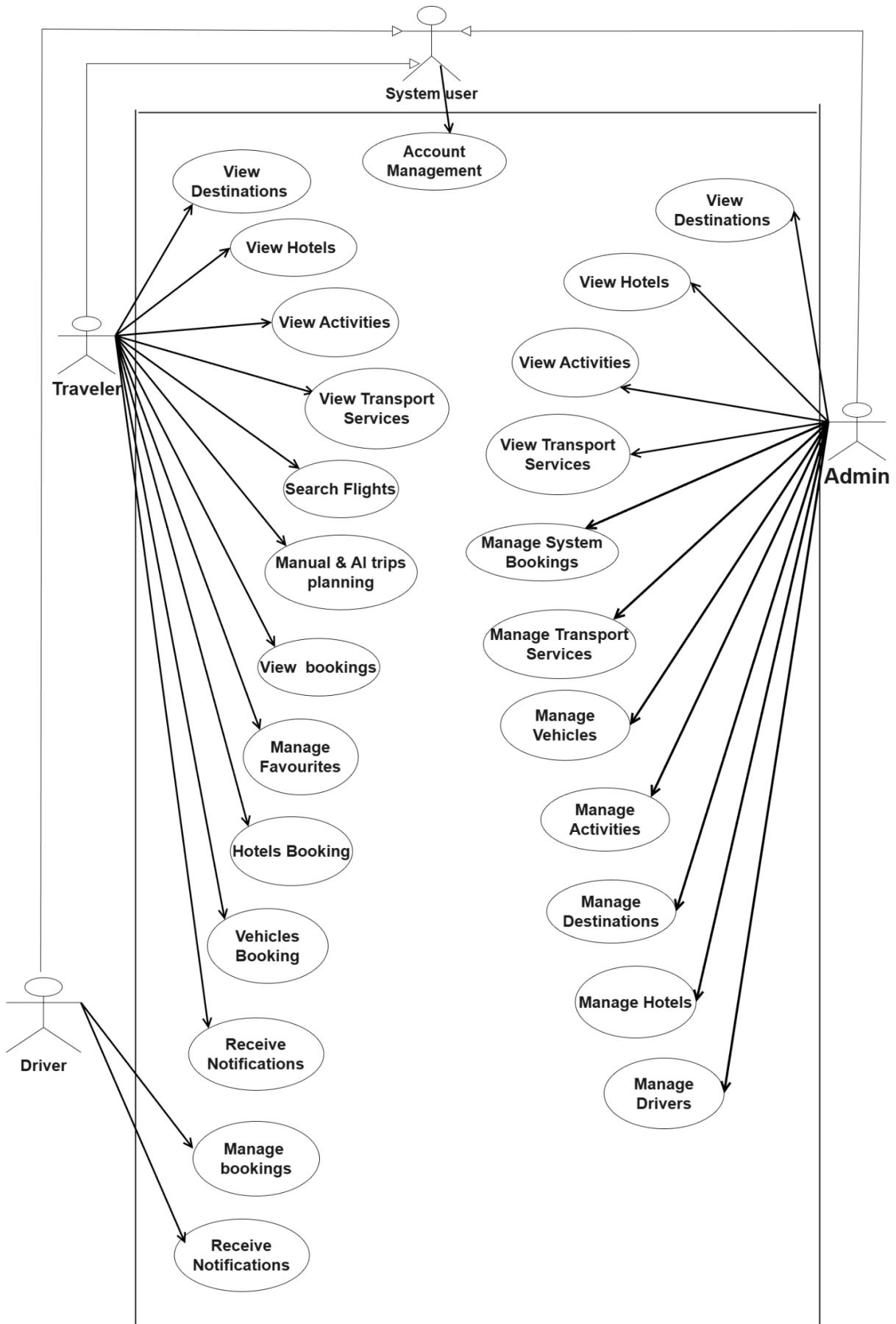


Figure 4: High level usecase

### **3.3.3 Development Process:**

#### **3.3.3.1 Sprint 0 – Preplanning & Initialization**

**Objective:** Establish the foundation for planning, requirements, and architecture.

##### **Activities:**

1. Define project objectives and scope.
2. Identify all system actors and main use cases.
3. Choose technologies (Laravel, Blade, Tailwind, MySQL).
4. Design preliminary database and architecture diagrams.
5. Prepare requirement specification and project plan.

##### **Deliverables:**

- *Project Scope & Technical Documents.*
  - *High-Level Architecture Diagram.*
  - *Initial Timeline & Role Assignments.*
- 

#### **3.3.3.2 Sprint 1 – Authentication & Role Management**

**Objective:** Implement secure login, registration, and role-based access for Admin, User, and Driver.

##### **Activities:**

1. Database schema for roles and users.
2. Registration, login, logout functionalities.
3. Testing and documentation.

##### **Deliverables:**

- *Authentication Module Release.*
  - *Sprint Documentation.*
-

### **3.3.3.3 Sprint 2 – Profile & Account Management**

**Objective:** Allow users, admins, and drivers to manage and update their personal information.

#### **Activities:**

1. Profile view and edit pages.
2. Password update and validation.
3. Testing and documentation.

#### **Deliverables:**

- *Profile Management Module.*
  - *Sprint Documentation.*
- 

### **3.3.3.4 Sprint 3 – Destination Management**

**Objective:** Develop destination creation, exploration, and search functionalities.

#### **Activities:**

1. Admin CRUD for destinations.
2. User interface for exploring and viewing destinations.
3. Search and filter implementation.
4. Testing and documentation.

#### **Deliverables:**

- *Destination Module Release.*
  - *Sprint Documentation.*
- 

### **3.3.3.5 Sprint 4 – Hotel Management & Booking**

**Objective:** Enable hotel management and booking process for users.

#### **Activities:**

1. Admin CRUD for hotels.
2. User interface for exploring and viewing hotels.

3. Search and filter implementation
4. Implement booking process and payment handling.
5. Testing and validation.

***Deliverables:***

- *Hotel Booking Module.*
  - *Sprint Documentation.*
- 

***3.3.3.6 Sprint 5 – Flight Search (External API Integration)***

**Objective:** Integrate external API to allow users to search and compare flights only (no in-app booking).

***Activities:***

1. Analyze and connect to flight API.
2. Implement search (destination, date, passengers).
3. Display flight results.
4. Testing and documentation.

***Deliverables:***

- *Flight Search Module (API-based).*
  - *Sprint Documentation.*
- 

***3.3.3.7 Sprint 6 – Activities & Favorites***

**Objective:** Add activity management and favorites system for user experience enhancement.

***Activities:***

1. CRUD for admin to manage activities.
2. User-side display for activities under destinations.
3. Implement favorites system (add/remove/view).
4. Testing and documentation.

**Deliverables:**

- *Activities & Favorites Module.*
  - *Sprint Documentation.*
- 

**3.3.3.8 Sprint 7 – Transportation & Driver Module**

**Objective:** Manage transportation services, driver assignments, and trip notifications.

**Activities:**

1. CRUD for transports, vehicles and drivers.
2. Booking system for rides.
3. Driver assignment and status tracking.
4. Testing and documentation.

**Deliverables:**

- Transportation Module.
  - Sprint Documentation.
- 

**3.3.3.9 Sprint 8 – Notification System:**

**Objective:** Develop a real-time notification system for user-driver communication.

**Activities:**

1. Implement event-based notifications for trip status updates.
2. Enable user-driver communication via alerts.
3. Integration with booking and transport modules.
4. Testing and documentation.

**Deliverables:**

5. Notification System Module.
  6. Sprint Documentation.
-

### **3.3.3.10 Sprint 9 – Booking Display & Filtering:**

**Objective:** Allow users and admins to view, manage, and filter bookings.

#### **Activities:**

1. Create user interface for viewing current and past bookings.
2. Implement filters (by date, status).
3. Admin panel for monitoring bookings.
4. Testing and validation.

#### **Deliverables:**

1. Booking Management Module.
  2. Sprint Documentation.
- 

### **3.3.3.11 Sprint 10 – Trip Planning & AI Recommendations**

**Objective:** Implement AI-based recommendation and manual trip planning.

#### **Activities:**

1. Build itinerary builder for user-planned trips.
2. Integrate AI module for smart suggestions (destinations, hotels, activities).
3. Testing and optimization.
4. Documentation of results.

#### **Deliverables:**

- AI Trip Planner Module.
  - Sprint Documentation.
- 

### **3.3.3.12 Sprint 11 – Final Testing & Documentation Review**

**Objective:** Ensure full system stability and finalize documentation.

#### **Activities:**

1. Conduct system integration and acceptance testing.
2. Fix remaining bugs and improve performance.
3. Prepare deployment and final report.

**Deliverables:**

- *Final System Release.*
  - *Project Closure Documentation*
- 

### **3.3.4 High-Level System Design:**

#### **3.3.4.1 High-Level System Architecture:**

The system follows the **MVC (Model–View–Controller)** architectural pattern provided by **Laravel**, which organizes the project into three main layers that separate the user interface, business logic, and data handling.

- **Model:**  
Represents the data structure and defines relationships between entities such as users, destinations, hotels, bookings, and activities.  
It communicates directly with the **MySQL** database using **Eloquent ORM**, which simplifies queries and data persistence.
- **View:**  
Handles the presentation logic and what the user sees in the browser.  
It is implemented using **Laravel Blade templates**, styled with **Tailwind CSS**, and enhanced with **JavaScript** for interactivity and responsiveness.
- **Controller:**  
Acts as the intermediary between the Model and the View.  
It processes incoming requests, applies business logic, retrieves necessary data from the Model, and passes it to the View for rendering.  
Controllers are also responsible for authentication, booking workflows, and communication with APIs such as flight search.

This MVC structure ensures **clean separation of concerns**, **maintainability**, and **modularity**, allowing each part of the system to be developed and tested independently.

#### **3.3.4.2 High-Level System Decomposition:**

The system consists of several interconnected modules, each mapped to controllers and models within the MVC structure:

##### **1. User & Profile Management Module**

Handles user registration, authentication, and profile editing.  
Includes role-based access for users, admins, and drivers.

## **2. Destination & Hotel Module**

Manages CRUD operations for destinations and hotels, enabling users to explore and filter travel options.

## **3. Booking & Transportation Module**

Implements hotel and vehicle booking functionalities, driver assignment, and payment handling.

## **4. Notification Module**

Sends alerts and updates to users and drivers for booking confirmations, cancellations, and trip updates.

## **5. Activities & Favorites Module**

Allows users to view activities available in destinations and mark their favorite destinations or hotels.

## **6. Flight Search Module (External API)**

Integrates with an external flight API to allow users to search and compare flights without booking.

## **7. Trip Planning & AI Recommendation Module**

Provides itinerary creation and AI-powered travel suggestions based on user preferences and history.

Each of these modules interacts through the **MVC structure**, ensuring smooth coordination between frontend and backend components.

### 3.3.4.3 High-Level Modules Diagram:

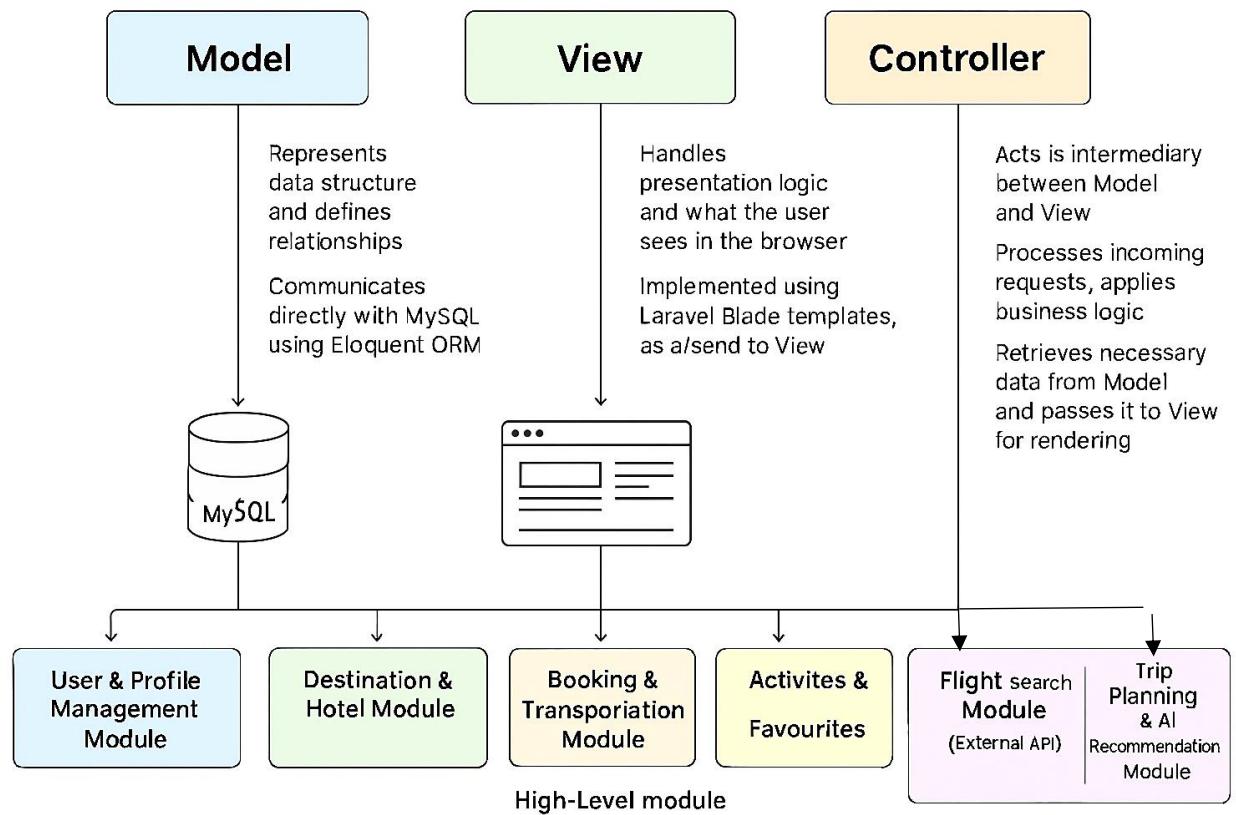


Figure ⑤ High level modules diagram

## 3.4 Initial RTM:

ID	Title	Analysis Section	Design Section	Code Integration	Test Unit Test
STP-FR-01	The system should allow users to register using email and password.				
STP-FR-02	The system should allow users to login and logout securely.				
STP-FR-03	The system should allow users to view and update their profile.				
STP-FR-04	The system should allow admins to manage destinations (add, edit, delete).				
STP-FR-05	The system should allow admins to manage hotels (add, edit, delete).				
STP-FR-06	The system should allow users to explore destinations and view their details.				

<b>STP-FR-07</b>	The system should allow users to view hotels and their details.				
<b>STP-FR-08</b>	The system should allow users to filter and search hotels and destinations.				
<b>STP-FR-9</b>	The system should allow users to book hotels and pay securely online.				
<b>STP-FR-10</b>	The system should allow users to view their current and previous bookings.				
<b>STP-FR-11</b>	The system should allow admins to view and filter all bookings.				
<b>STP-FR-12</b>	The system should allow integration with external API for flight search.				
<b>STP-FR-13</b>	The system should allow users to search flights (without booking).				
<b>STP-FR-14</b>	The system should allow admin to manage transportation (transport services ,vehicles, drivers).				
<b>STP-FR-15</b>	The system should allow users to request vehicles.				
<b>STP-FR-16</b>	The system should allow drivers to receive vehicles requests.				
<b>STP-FR-17</b>	The system should provide real-time notifications between driver and user.				
<b>STP-FR-18</b>	The system should allow admin to manage activities for each destination(add,edit,delete).				
<b>STP-FR-19</b>	The system should allow users to view and search for activities.				
<b>STP-FR-20</b>	The system should allow users to add and remove favorite destinations or hotels.				
<b>STP-FR-21</b>	The system should allow users to manually build their trip itinerary.				
<b>STP-FR-22</b>	The system should provide AI-based trip recommendations.				
<b>STP-FR-23</b>	The system should ensure secure online payments.				

Table 6 Initial RTM

### 3.5 Initial Test Cases:

Test Case ID	Test Name	Purpose
STP-TC-01	Register New User with Valid Data	Verify that new users can register successfully using a valid email and password.
STP-TC-02	Login with Valid Credentials	Verify that users can login successfully using correct credentials.
STP-TC-03	Login with Invalid Credentials	Verify that the system rejects invalid login attempts.
STP-TC-04	Logout User Session	Verify that logged-in users can logout successfully.
STP-TC-05	View User Profile	Verify that the system retrieves and displays user profile data correctly.
STP-TC-06	Update User Profile	Verify that users can update their personal information successfully.
STP-TC-07	Admin Manage Destinations	Verify that admin can add, edit, and delete destinations.
STP-TC-08	Admin Manage Hotels	Verify that admin can add, edit, and delete hotels.
STP-TC-09	Explore Destinations	Verify that users can view all available destinations and open destination details.
STP-TC-10	View Hotels	Verify that users can browse hotels and view hotel details.
STP-TC-11	Filter Hotels and Destinations	Verify that users can filter and search hotels or destinations based on criteria.
STP-TC-12	Book Hotel Successfully	Verify that users can complete a hotel booking and payment process.
STP-TC-13	View User Bookings	Verify that users can view their current and previous bookings.
STP-TC-14	Admin View Bookings	Verify that admin can view and filter all bookings in the system.
STP-TC-15	Retrieve Flights via API	Verify that flight data is fetched correctly from the external API.
STP-TC-16	Search Flights (No Booking)	Verify that users can search and compare flights without booking.
STP-TC-17	Admin Manage Transportation	Verify that admin can manage transport services, vehicles, and drivers.
STP-TC-18	Request Vehicle	Verify that users can request transportation services.
STP-TC-19	Driver Receive Request	Verify that drivers can view vehicle requests.

<b>STP-TC-20</b>	Real-time Notifications	Verify that notifications are delivered in real-time between user and driver.
<b>STP-TC-21</b>	Admin Manage Activities	Verify that admin can add, edit, and delete activities for destinations.
<b>STP-TC-22</b>	View and Search Activities	Verify that users can view and search activities.
<b>STP-TC-23</b>	Add Favorite Destination/Hotel	Verify that users can add destinations or hotels to favorites.
<b>STP-TC-24</b>	Remove Favorite Destination/Hotel	Verify that users can remove favorites successfully.
<b>STP-TC-25</b>	Build Trip Itinerary	Verify that users can manually build and organize their trip itinerary.
<b>STP-TC-26</b>	AI Trip Recommendation	Verify that the system generates accurate AI-based trip recommendations.
<b>STP-TC-27</b>	Secure Payment Processing	Verify that all payment transactions are encrypted and processed securely.

*Table 7 Initial Test Cases*

## **Chapter 4 - System Analysis**

## **4.1 Introduction:**

This chapter presents a detailed analysis of the **Smart Trip Planner System**. It defines the system's main objectives, core functionalities, and technical requirements by examining user needs in the context of modern travel planning. The chapter highlights the functional and non-functional requirements, identifies the main actors, and illustrates their interactions with the system. This analysis serves as the foundation for designing and implementing an intelligent, user-friendly travel planning platform that simplifies the entire trip experience — from discovery to booking and management.

## **4.2 Purpose:**

The purpose of this project is to design and implement a **smart travel planning platform** that allows users to easily explore destinations, activities and hotels, search real time flights, plan their itineraries, and manage their bookings all in one place. The system also aims to provide **AI-based trip recommendations** to assist users in making personalized and cost-effective travel decisions. Furthermore, it offers **admin functionalities** to manage destinations, hotels, activities, and transportation efficiently.

## **4.3 Project Scope:**

The **Smart Trip Planner System** covers the full travel management process, integrating multiple services into a unified platform that connects users, administrators, and drivers.

### **Users can:**

- Register, log in, and manage their accounts securely.
- Search and explore destinations, hotels and activities.
- Manage their favorite list.
- Book hotels and request transport services.
- Build personalized itineraries manually or receive AI-based trip recommendations.
- Make secure online payments and view booking history.

### **Administrators can:**

- Manage destinations, hotels, activities, and transportation (vehicles, drivers, and services).
- Review and approve or reject driver registration requests.
- Monitor and filter all user and driver bookings.

### **Drivers can:**

- Submit a registration request to join the system.
- Receive an approval or rejection email based on the admin's decision.
- Log in only after approval and access their personal dashboard.
- View all assigned trip requests, including both pending and completed bookings.
- Update the status of assigned trips as they are completed.

Additionally, the system integrates with **external APIs** to retrieve flight search results and employs **real-time notifications** to improve communication between users and drivers.

This scope ensures that all key travel elements—planning, booking, and transportation management—are centralized, intelligent, and easy to use for all involved actors.

### **4.4 Requirements Elicitation:**

During the analysis phase, several existing travel planning systems and booking platforms were examined to identify their **strengths and weaknesses**.

The findings were used to formulate the best possible set of requirements for a comprehensive and efficient platform.

The main goal was to combine convenience, automation, and personalization, providing an experience that is both **intuitive for users** and **practical for administrators**.

Through user feedback, case studies, and feasibility analysis, the final set of requirements was designed to ensure:

- Simplified user interactions.
- Secure and reliable booking operations.
- Smooth integration with external services.

- Personalized AI-based trip recommendations.
- Real-time communication between different user roles.

## 4.5 Requirements Table:

### 4.5.1 Functional Requirements:

Req_ID	Req.Name	Requirement Title	Actor	Category
FR-SU-01	Login	The system should allow the user to login using their credentials.	System User	Account Management
FR-SU-02	Logout	The system should allow the user to logout.	System User	Account Management
FR-SU-03	Edit Profile	The system should allow the user to update personal information such as email and password.	System User	Account Management
FR-USER-01	Register	The system should allow the user to register using their information.	Traveller/Driver	Account Management
FR-USER-02	View Destinations	The system should allow the user to browse the list of available tourist destinations.	Traveller/Admin	Destinations
FR-USER-03	View Destination Details	The system should allow the user to view full details of a selected tourist destination.	Traveller/Admin	Destinations
FR-USER-04	Search Destination	The system should allow the user to search for a tourist destination by name or location.	Traveller/Admin	Destinations
FR-USER-05	View Hotels	The system should allow the user to browse the list of available hotels.	Traveller/Admin	Hotels
FR-USER-06	View Hotel Details	The system should allow the user to view details of a selected hotel.	Traveller/Admin	Hotels
FR-USER-07	Search Hotels	The system should allow the user to search for a hotel by name, location, or rating.	Traveller/Admin	Hotels

<b>FR-USER-08</b>	View Transport Services	The system should allow the user to browse the list of available transport services.	Traveller/Admin	Transport
<b>FR-USER-09</b>	View Activities	The system should allow the user to browse the list of available activities.	Traveller/Admin	Activities
<b>FR-USER-10</b>	View Activity Details	The system should allow the user to view details of a selected activity.	Traveller/Admin	Activities
<b>FR-USER-10</b>	Search Activities	The system should allow the user to search for an activity by name or destination and filter them by multiple criteria.	Traveller/Admin	Activities
<b>FR-USER-11</b>	Search Bookings	The system should allow the travellers to search and filter their own bookings, and for admins to search and filter all system bookings.	Traveller/Admin	Booking Management
<b>FR-TRV-01</b>	Book Hotel	The system should allow the user to book hotels using an in-app booking form.	Traveller	Hotels
<b>FR-TRV-02</b>	Search Vehicles	The system should allow the user to search for available vehicles according to their required date, time, and transport service.	Traveller	Vehicles
<b>FR-TRV-03</b>	Book Vehicle	The system should allow the user to reserve the required vehicle.	Traveller	Vehicles
<b>FR-TRV-04</b>	Receive Notifications	The system should notify the traveller when a new booking is created.	Traveller	Notifications
<b>FR-TRV-05</b>	Pay	The system should allow the user to complete hotel booking and transport reservation payments via a secure payment gateway.	Traveller	Payment
<b>FR-TRV-06</b>	Search Flights	The system should allow the user to search for flights and view results via an external API.	Traveller	Flight Searching
<b>FR-TRV-07</b>	Add To Favorites	The system should allow the user to add hotels destinations, and activities to a favorites list.	Traveller	Favorites
<b>FR-TRV-08</b>		The system should allow the user to remove hotels	Traveller	Favorites

	Remove From Favorites	destinations, and activities from their favorites list.		
<b>FR-TRV-09</b>	View Favorites	The system should allow the user to view their favorites list.	Traveller	Favorites
<b>FR-TRV-10</b>	Create Manual Trip Plan	The system should allow the user to create travel plans manually.	Traveller	Travel Planning
<b>FR-TRV-11</b>	Create AI Trip Plan	The system should allow the user to create travel plans using AI.	Traveller	Travel Planning
<b>FR-TRV-12</b>	View Trip Plan details	The system should allow the user to view the details of their travel plans.	Traveller	Travel Planning
<b>FR-TRV-13</b>	Delete Trip Plan	The system should allow the user to delete their travel plans.	Traveller	Travel Planning
<b>FR-TRV-14</b>	View Trips Plans	The system should allow the user to view all of their travel plans.	Traveller	Travel Planning
<b>FR-TRV-15</b>	View Bookings	The system should allow the travellers to view their own bookings of hotels and transports.	Traveller	Booking Management
<b>FR-DR-01</b>	View Driver's Assigned Bookings	The system should allow drivers to view bookings assigned to them.	Driver	Booking Management
<b>FR-DR-02</b>	Update Booking Status	The system should allow drivers to update bookings status (completed, cancelled).	Driver	Booking Management
<b>FR-DR-03</b>	Receive Notifications	The system should notify drivers when a new booking is created.	Driver	Notifications
<b>FR-ADMIN-01</b>	Add Destination	The system should allow the admin to add tourist destinations.	Admin	Destinations Management
<b>FR-ADMIN-02</b>	Edit Destination	The system should allow the admin to edit tourist destinations.	Admin	Destinations Management
<b>FR-ADMIN-03</b>	Delete Destination	The system should allow the admin to delete tourist destinations.	Admin	Destinations Management

<b>FR-ADMIN-04</b>	Add Hotel	The system should allow the admin to add hotels.	Admin	Hotel Management
<b>FR-ADMIN-05</b>	Edit Hotel	The system should allow the admin to edit hotels.	Admin	Hotel Management
<b>FR-ADMIN-06</b>	Delete Hotel	The system should allow the admin to delete hotels.	Admin	Hotel Management
<b>FR-ADMIN-07</b>	View System Bookings	The system should allow the admin to view all system bookings of users and drivers.	Admin	Booking Management
<b>FR-ADMIN-08</b>	Add Transport Service	The system should allow the admin to add transportation services.	Admin	Transportation
<b>FR-ADMIN-09</b>	Edit Transport Service	The system should allow the admin to edit transportation services.	Admin	Transportation
<b>FR-ADMIN-10</b>	Delete Transport Service	The system should allow the admin to delete transportation services.	Admin	Transportation
<b>FR-ADMIN-11</b>	View Transport Linked Vehicles	The system should allow the admin to view a list of all available vehicles for a specific transport service.	Admin	Vehicles
<b>FR-ADMIN-12</b>	Add Vehicle	The system should allow the admin to add vehicles and assign them to services.	Admin	Vehicles
<b>FR-ADMIN-13</b>	Edit Vehicle	The system should allow the admin to edit vehicles.	Admin	Vehicles
<b>FR-ADMIN-14</b>	Delete Vehicle	The system should allow the admin to delete vehicles.	Admin	Vehicles
<b>FR-ADMIN-15</b>	Accept/Reject Drivers	The system should allow the admin to approve/reject drivers and send them an email.	Admin	Driver Management
<b>FR-ADMIN-16</b>	View Drivers	The system should allow the admin to view all drivers with their details.	Admin	Driver Management
<b>FR-ADMIN-17</b>	Search Drivers	The system should allow the admin to search for drivers and filter them.	Admin	Driver Management
<b>FR-ADMIN-18</b>	Delete Driver	The system should allow the admin to delete a driver.	Admin	Driver Management

<b>FR-ADMIN-19</b>	Add Activity	The system should allow the admin to add activities for each destination.	Admin	Activities
<b>FR-ADMIN-20</b>	Edit Activity	The system should allow the admin to edit activities.	Admin	Activities
<b>FR-ADMIN-21</b>	Delete Activity	The system should allow the admin to delete activities.	Admin	Activities

*Table 8 Functional Requirements*

#### **4.5.2 Non-Functional Requirements:**

Req_ID	Requirement Title	Category	Type
<b>NFR-01</b>	The system should handle up to 1000 concurrent users without performance degradation.	Scalability	Non-Functional
<b>NFR-02</b>	The system should be extensible to allow adding new features, services, modules, and API integrations.	Extensibility	Non-Functional
<b>NFR-03</b>	The system should use secure token-based authentication with password hashing.	Security	Non-Functional
<b>NFR-04</b>	The system should maintain at least 90% uptime and handle up to 1200 users simultaneously without crashing.	Availability	Non-Functional
<b>NFR-05</b>	The system should load pages in less than 3 seconds for standard user operations.	Performance	Non-Functional
<b>NFR-06</b>	The system should store data in a reliable database.	Reliability	Non-Functional
<b>NFR-07</b>	The system should support multiple browsers (Chrome, Firefox, Edge, Safari) for all users.	Usability	Non-Functional
<b>NFR-08</b>	The system should encrypt sensitive data in transit using HTTPS.	Security	Non-Functional
<b>NFR-09</b>	The system should send notifications and emails reliably for bookings, approvals, and alerts.	Reliability	Non-Functional

*Table 9 Non-Functional Requirements*

## 4.6 Analysis

In this section, the analytical study of the **Smart Trip Planner System** is presented. The analysis focuses on identifying, organizing, and modeling the main functional requirements of the system using **Unified Modeling Language (UML)** diagrams such as **use case diagrams, activity diagrams, and sequence diagrams**. These models illustrate the system's structure and dynamic behavior, including the interactions between users, administrators, and drivers.

The analysis aims to ensure a clear understanding of how each functional module—such as **user and admin account management, destination and hotel management, booking and payment processing, transportation services, trip planning with AI recommendations, and notification management**—works together to achieve the system's objectives.

This structured analytical approach provides a comprehensive view of the system's functionality, guaranteeing alignment between user needs, functional requirements, and software design. It also serves as a foundation for accurate implementation, testing, and future maintenance of the system.

## 4.6.1 Requirements Modeling:(use case)

### 4.6.1.1 Traveler Use Case:

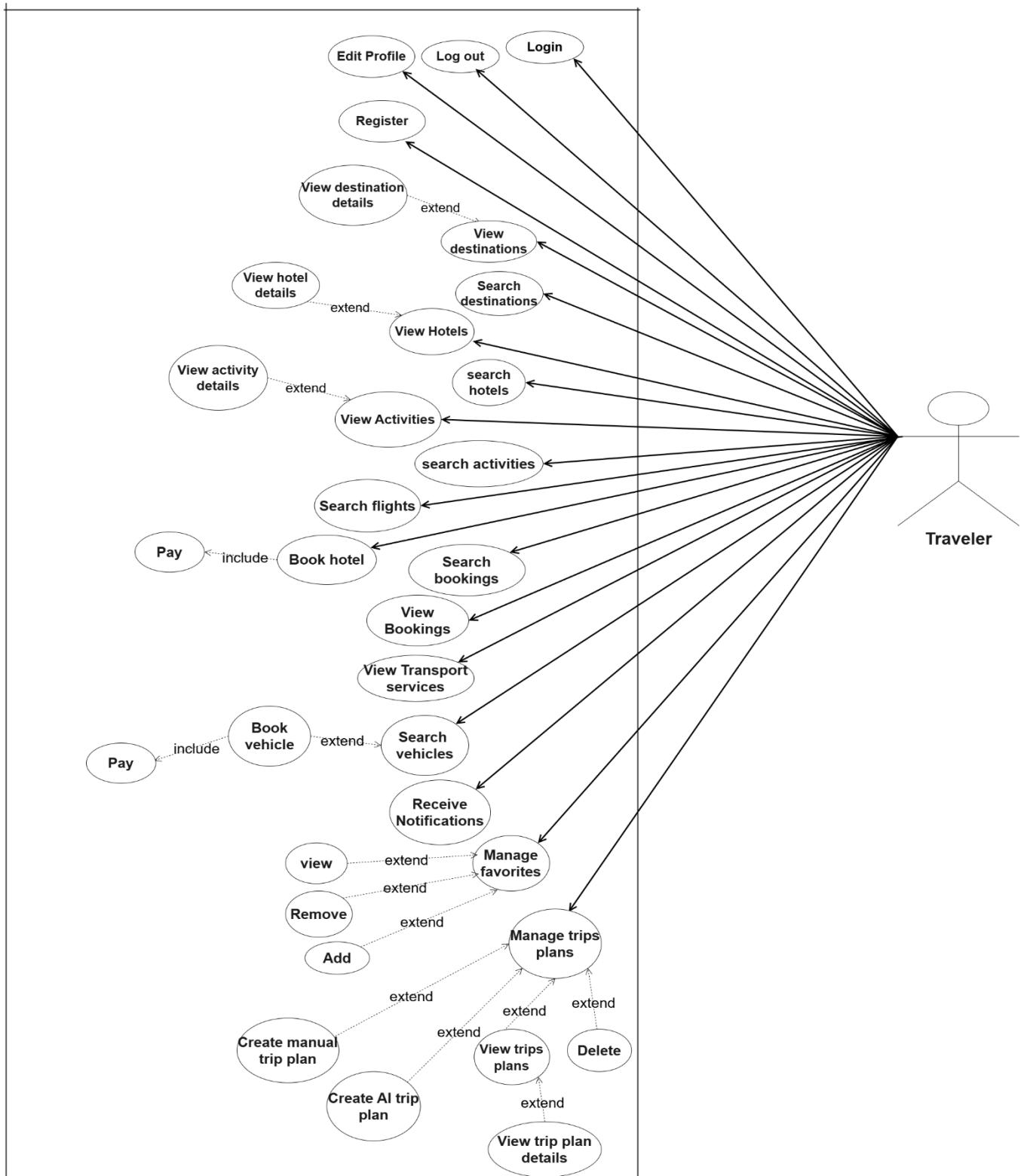


Figure 7 Traveler UseCase

#### 4.6.1.2 Admin UseCase:

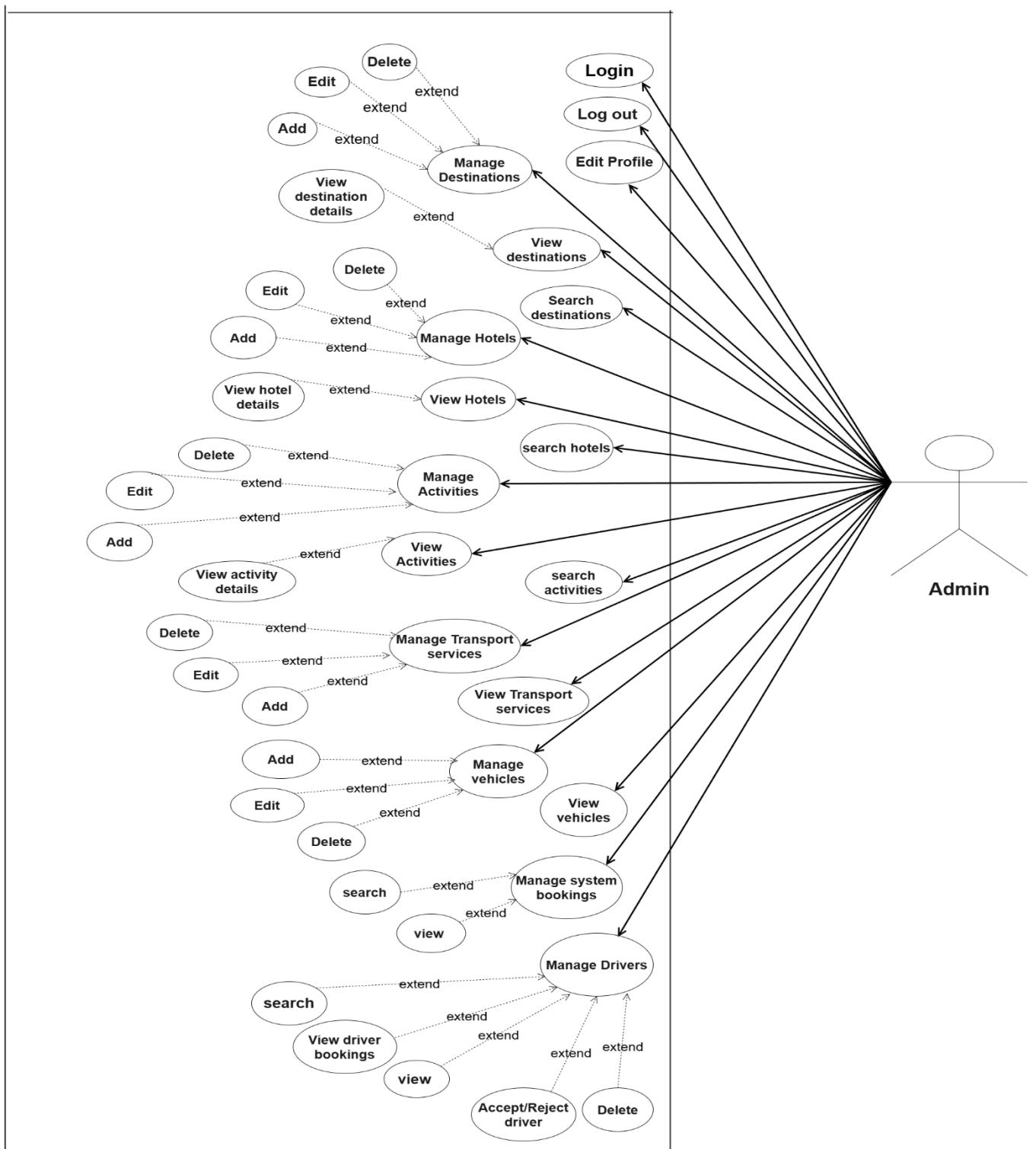


Figure V Admin Usecase

#### 4.6.1.3 Driver Use Case:

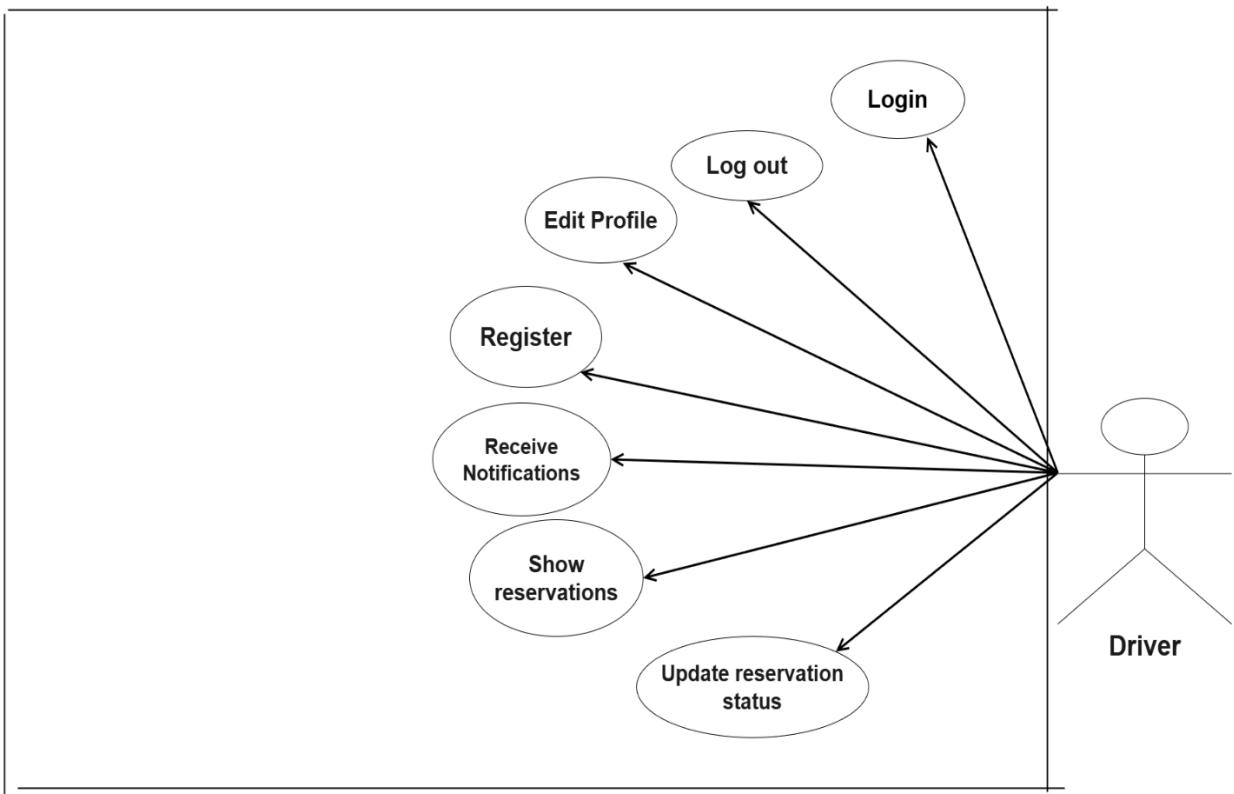


Figure 4 Driver UseCase

## ● Register:

<b>ID</b>	<b>UC-01</b>
<b>Use Case</b>	<b>Register</b>
<b>Actor</b>	<b>Traveller / Driver</b>
<b>Pre Condition</b>	The user is not registered or logged in yet.
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>- Traveller account is activated immediately.</li> <li>- Driver account is submitted for admin approval.</li> </ul>
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks <b>Create New Account</b>.</li> <li>2. The system prompts the user to select a <b>Role</b>: Traveller or Driver.</li> <li>3. The user selects a role.</li> <li>4. The system shows a form based on the selected role.</li> <li>5. The user fills in the required information.</li> <li>6. The user clicks on <b>Register</b>.</li> <li>7. The system validates the entered information: <ul style="list-style-type: none"> <li>- If the selected role is Traveller: <ul style="list-style-type: none"> <li>. The system creates the account immediately and redirects the user to the Dashboard.</li> </ul> </li> <li>- If the selected role is driver: <ul style="list-style-type: none"> <li>. The system triggers the <b>sub-use case: Submit for Admin Approval</b>.</li> </ul> </li> </ul> </li> </ol>
<b>Alternative Scenario</b>	<ul style="list-style-type: none"> <li>- Step 7: if required fields are missing or invalid, an error message is displayed and returns to Step 4.</li> <li>- Step 7: If the email already exists, the system displays an error message and returns to Step 4.</li> </ul>

Table 10 Register

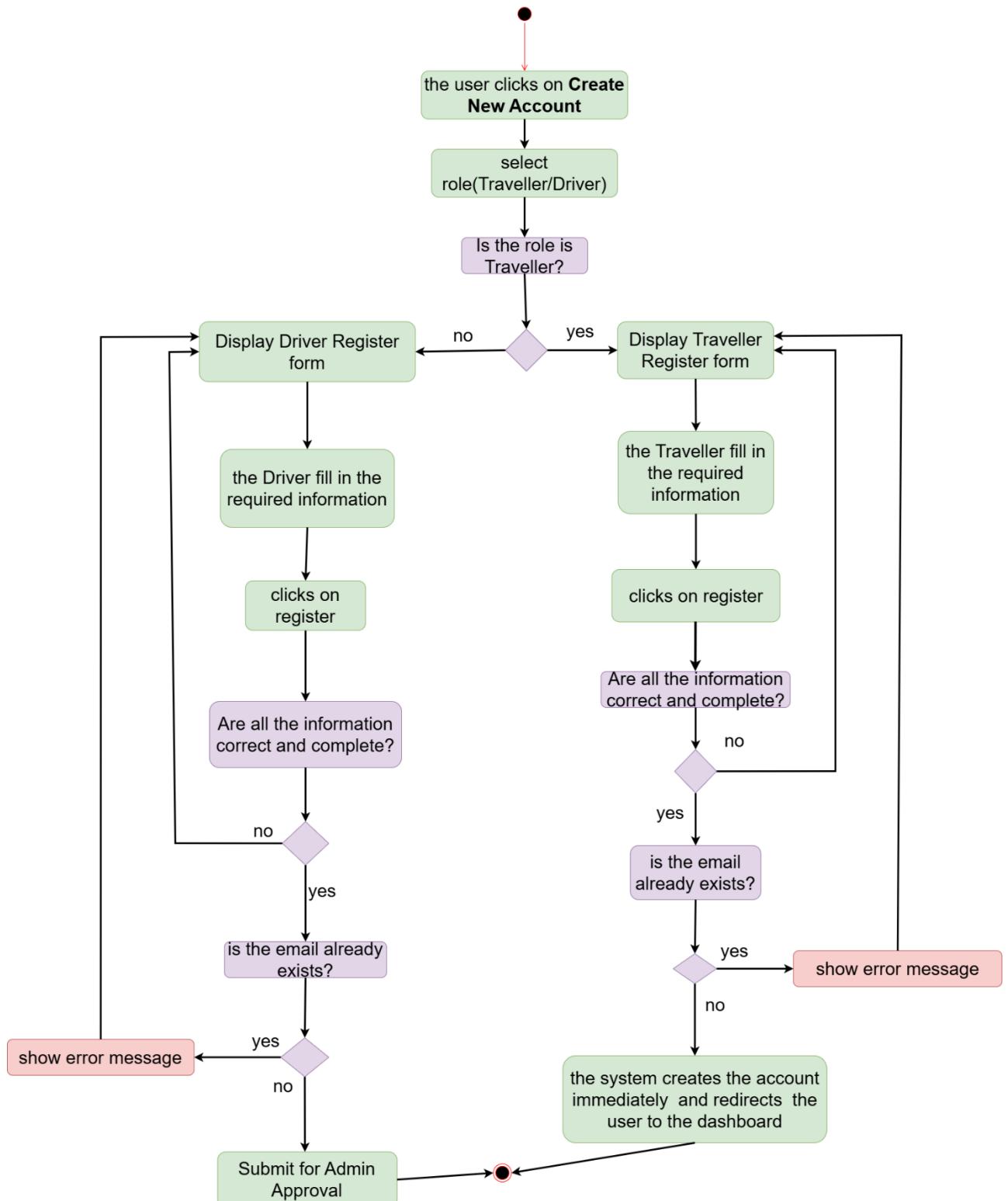


Figure 1 Activity Diagram (Register)

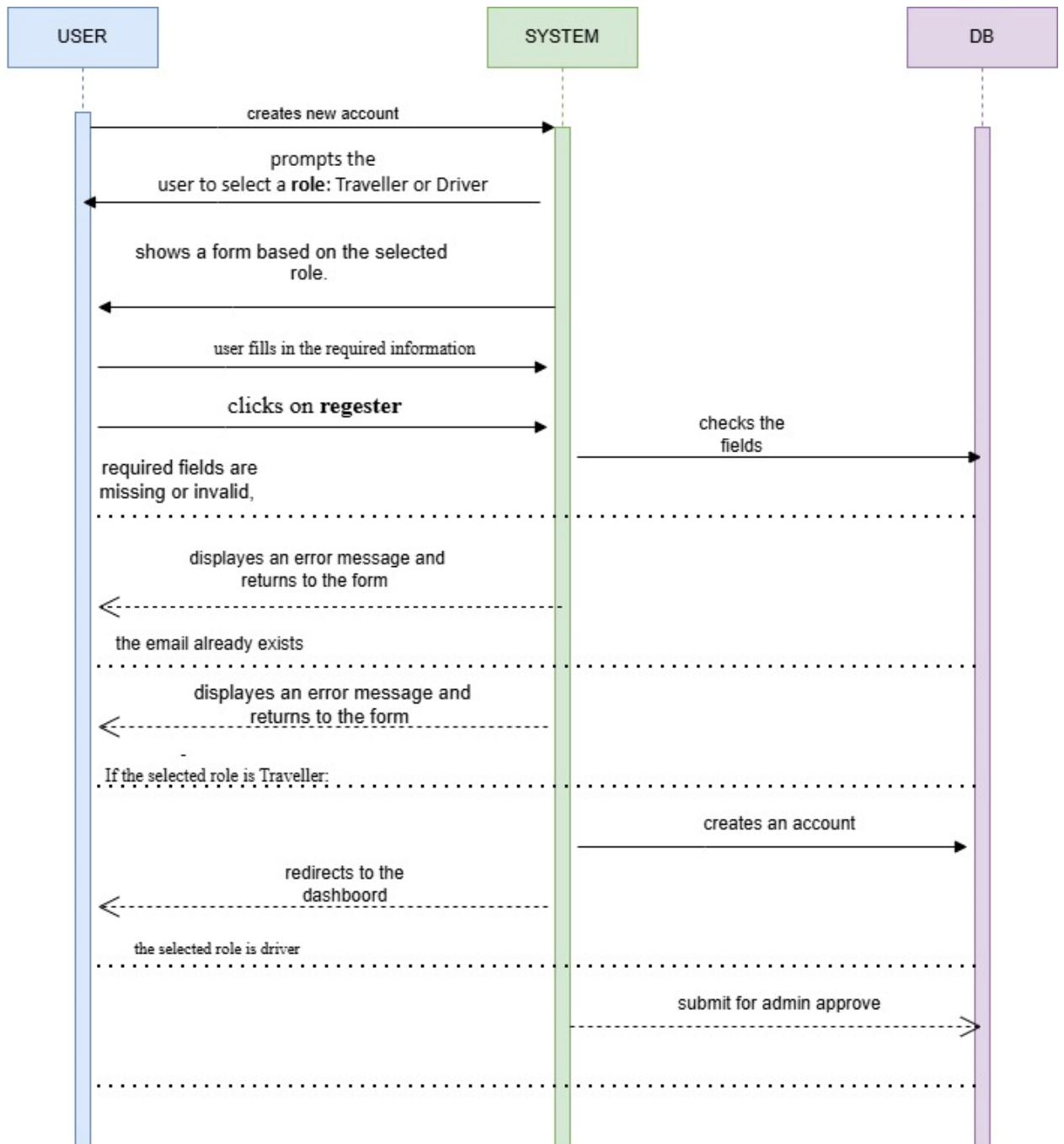


Figure 1 • Sequence Diagram( Register)

## ● Sub-Use Case: Submit for Admin Approval:

<b>ID</b>	UC-02Sub
<b>Use Case</b>	<b>Submit For Admin Approval</b>
<b>Actor</b>	<b>Driver</b>
<b>Pre Condition</b>	The driver has completed the registration form.
<b>Post Condition</b>	The driver account is sent for admin approval.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The system sends the driver's registration request to the Admin for approval.</li> <li>2. The system displays the message: "Your request has been sent to review, please check your email".</li> </ol>
<b>Alternative Scenario</b>	

Table 11 Sub-Use Case: Submit for Admin Approval

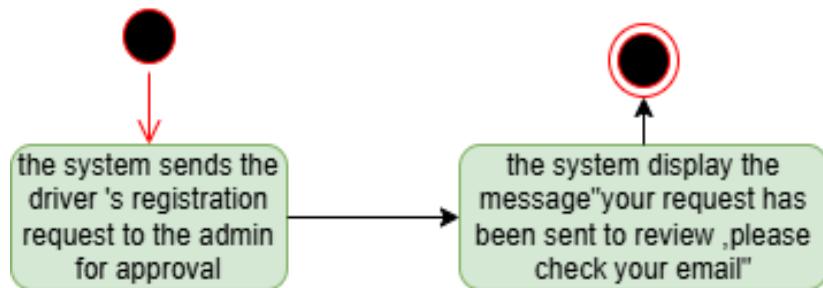


Figure 11 Activity Diagram(Sub-Use Case )

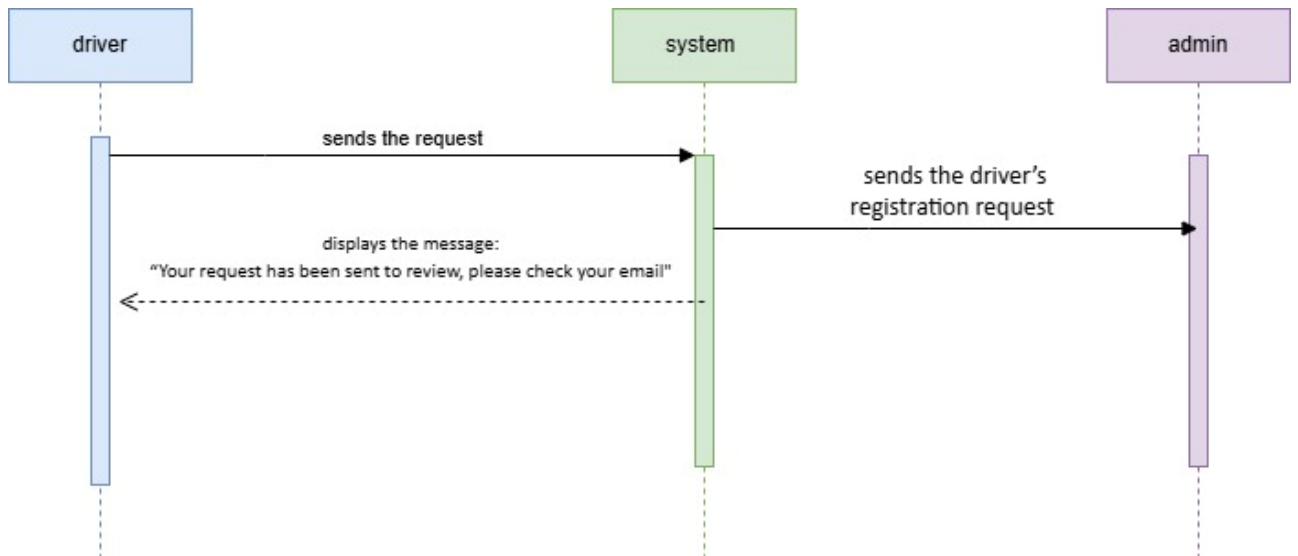


Figure 17 Sequence Diagram(Sub-Use Case)

## ● Login:

ID	UC-03
Use Case	Login
Actor	Traveller / Driver / Admin
Pre Condition	The user is registered.
Post Condition	The user successfully logs into the system.
Main Scenario	<p>1. The user enters email and password on the login page.</p> <p>2. The user clicks the <b>Login</b> button.</p> <p>3. The system validates the credentials.</p> <p>4. The system checks the role and account status:</p> <ul style="list-style-type: none"> <li>- <b>Traveller/Admin:</b> Credentials valid → system logs in the user and redirects to Dashboard.</li> <li>- <b>Driver:</b> Credentials valid and account approved → system logs in the user and redirects to Dashboard.</li> </ul>
Alternative Scenario	<p><b>Step 4 :</b></p> <ul style="list-style-type: none"> <li>- If credentials not valid → system displays message: "These credentials do not match our records".</li> </ul> <p><b>Step 4 (Driver):</b></p> <ul style="list-style-type: none"> <li>- If account still pending → system displays message: "Your account is under review. Please check your email".</li> <li>- If account is rejected → system displays message: "These credentials do not match our records".</li> </ul>

Table 12 Login

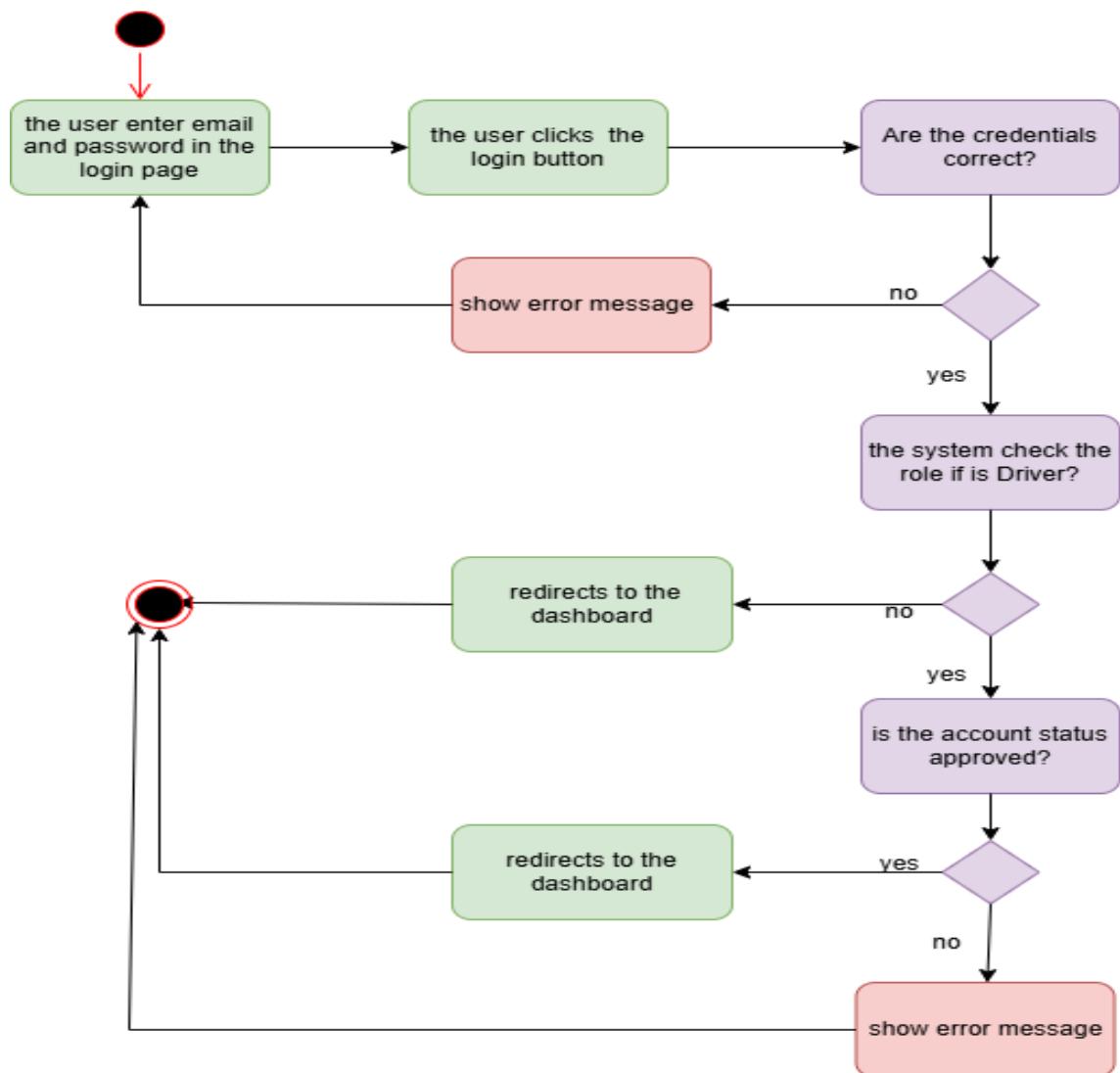


Figure 11 Activity Diagram( Login)

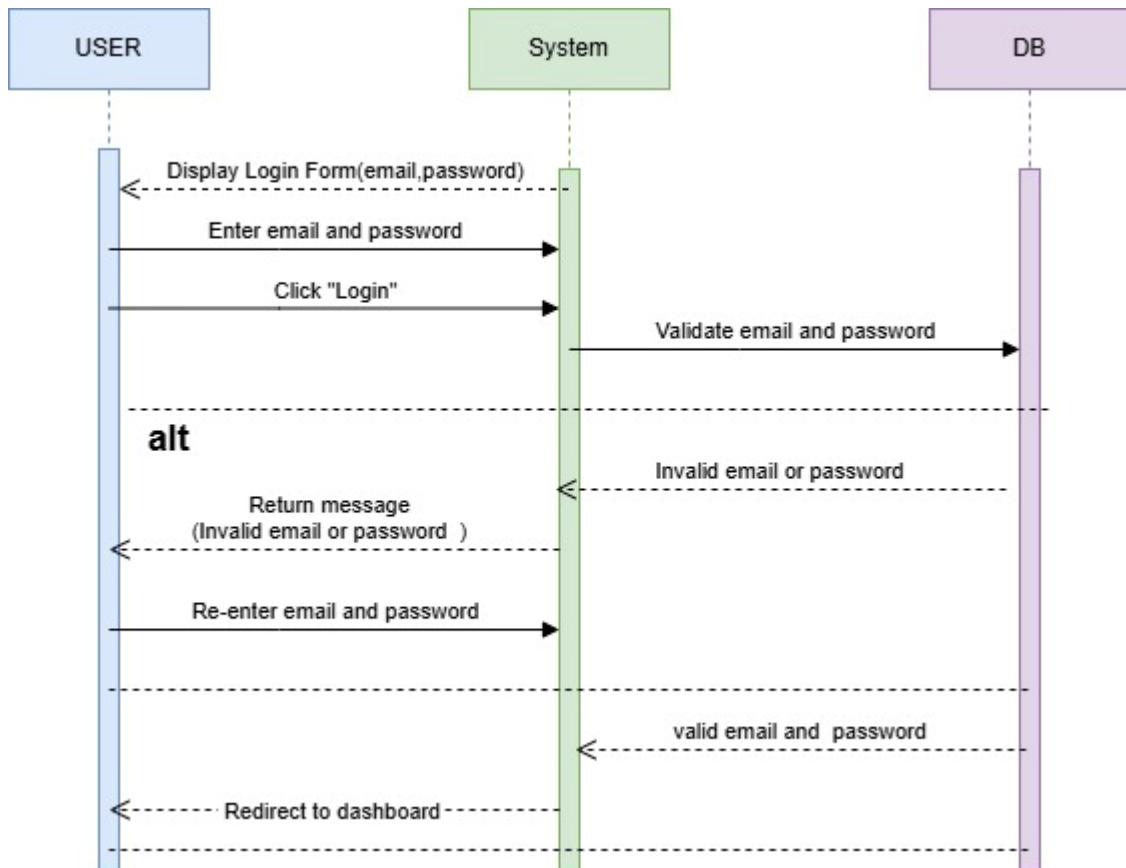


Figure 14 Sequence Diagram (Login)

## ● Logout:

ID	UC-04
Use Case	<b>Logout</b>
Actor	<b>System User</b>
Pre Condition	The user is already logged into the system
Post Condition	The user is logged out and redirected to the login page.
Main Scenario	<ol style="list-style-type: none"> <li>1. The user clicks on their username.</li> <li>2. The user clicks the <b>Logout</b> button.</li> <li>3. The system redirects the user to the login page.</li> </ol>
Alternative Scenario	

Table 13 Logout

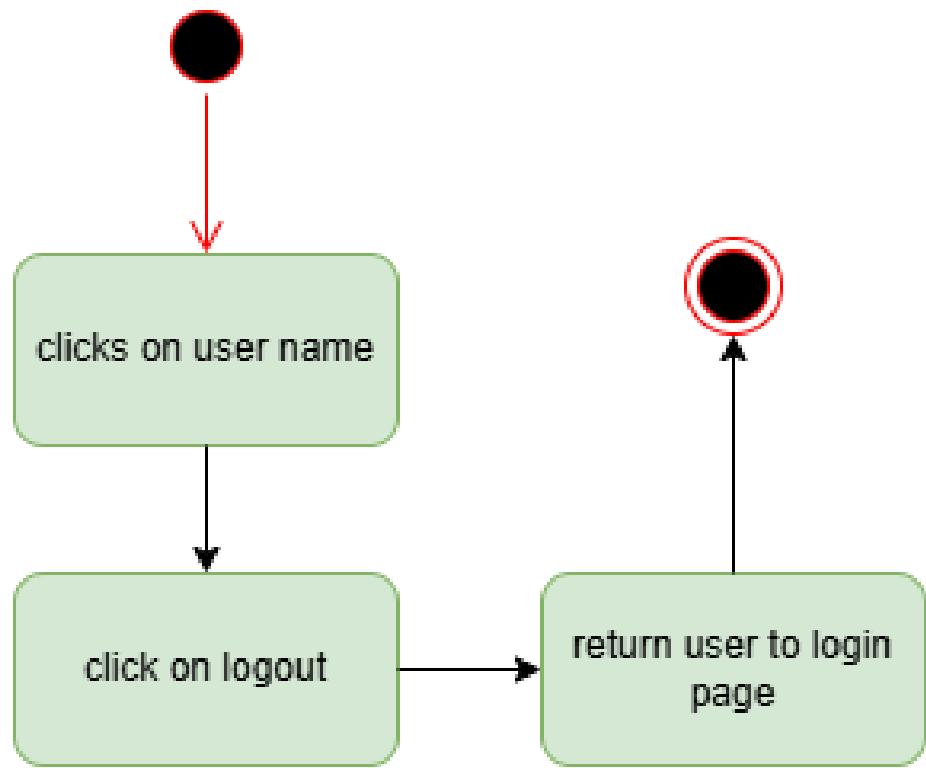


Figure 1◦ Activity Diagram (Logout)

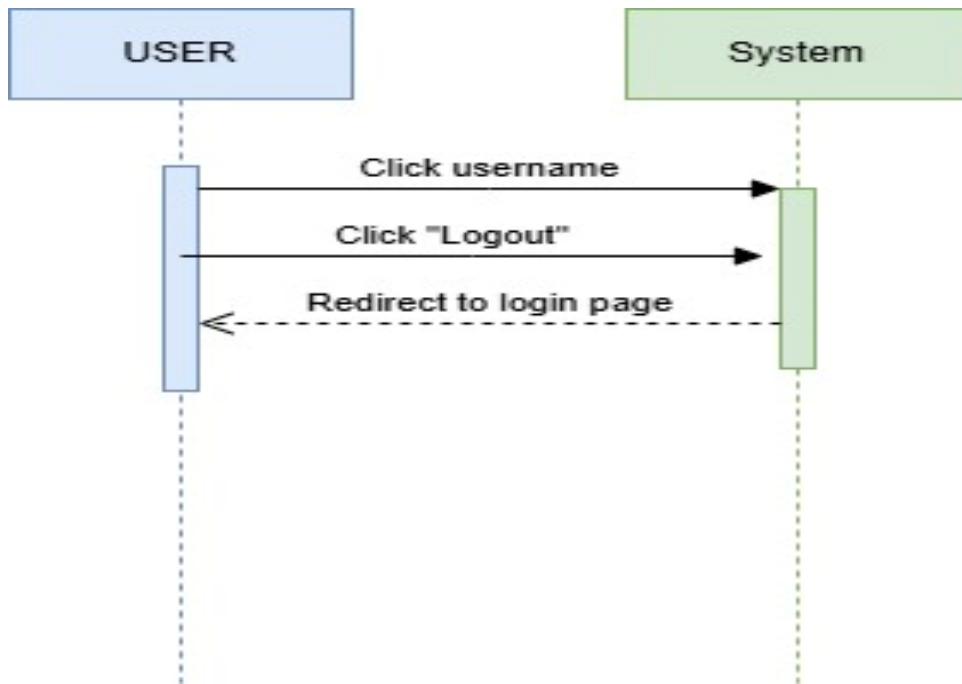


Figure 17 Sequence Diagram(Logout)

## ● Edit Profile:

<b>ID</b>	UC-05
<b>Use Case</b>	<b>Edit Profile</b>
<b>Actor</b>	<b>System User</b>
<b>Pre Condition</b>	The user is already logged into the system
<b>Post Condition</b>	The user's personal information is updated
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks their username.</li> <li>2. The system displays the <b>Profile</b> button.</li> <li>3. The user clicks <b>Profile</b>.</li> <li>4. The system displays a form to edit (email, first name, last name, country, phone number, new password, and current password).</li> <li>5. The user enters their new information.</li> <li>6. The user clicks <b>Save</b>.</li> <li>7. The system checks that the email is not already used and verifies the old password.</li> <li>8. The system saves the updated information and displays a success message.</li> </ol>
<b>Alternative Scenario</b>	Step 7: If the email already exists or the old password is incorrect, an error message is displayed.

Table 14 Edit Profile

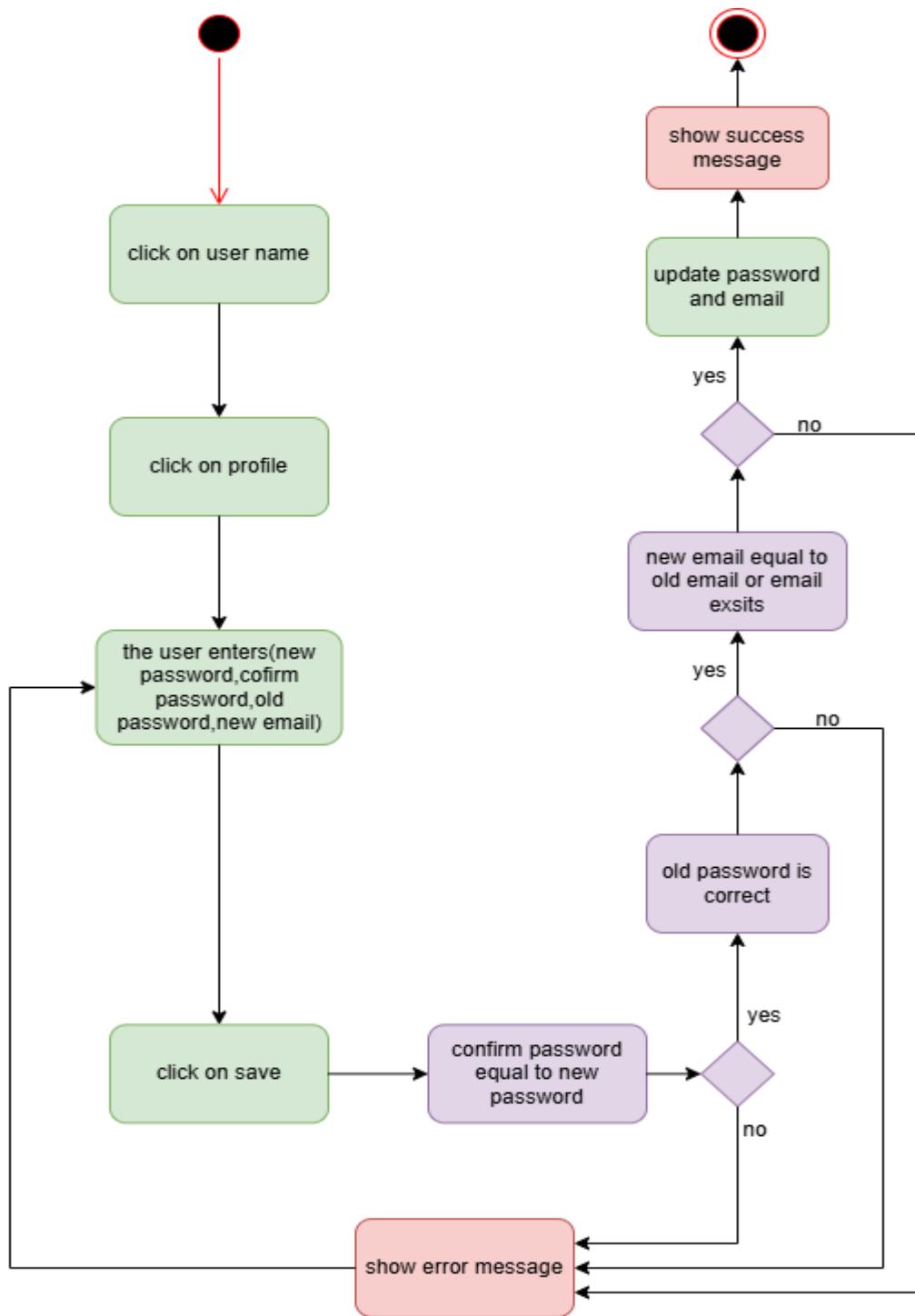


Figure 14 Activity Diagram (Edit Profile)

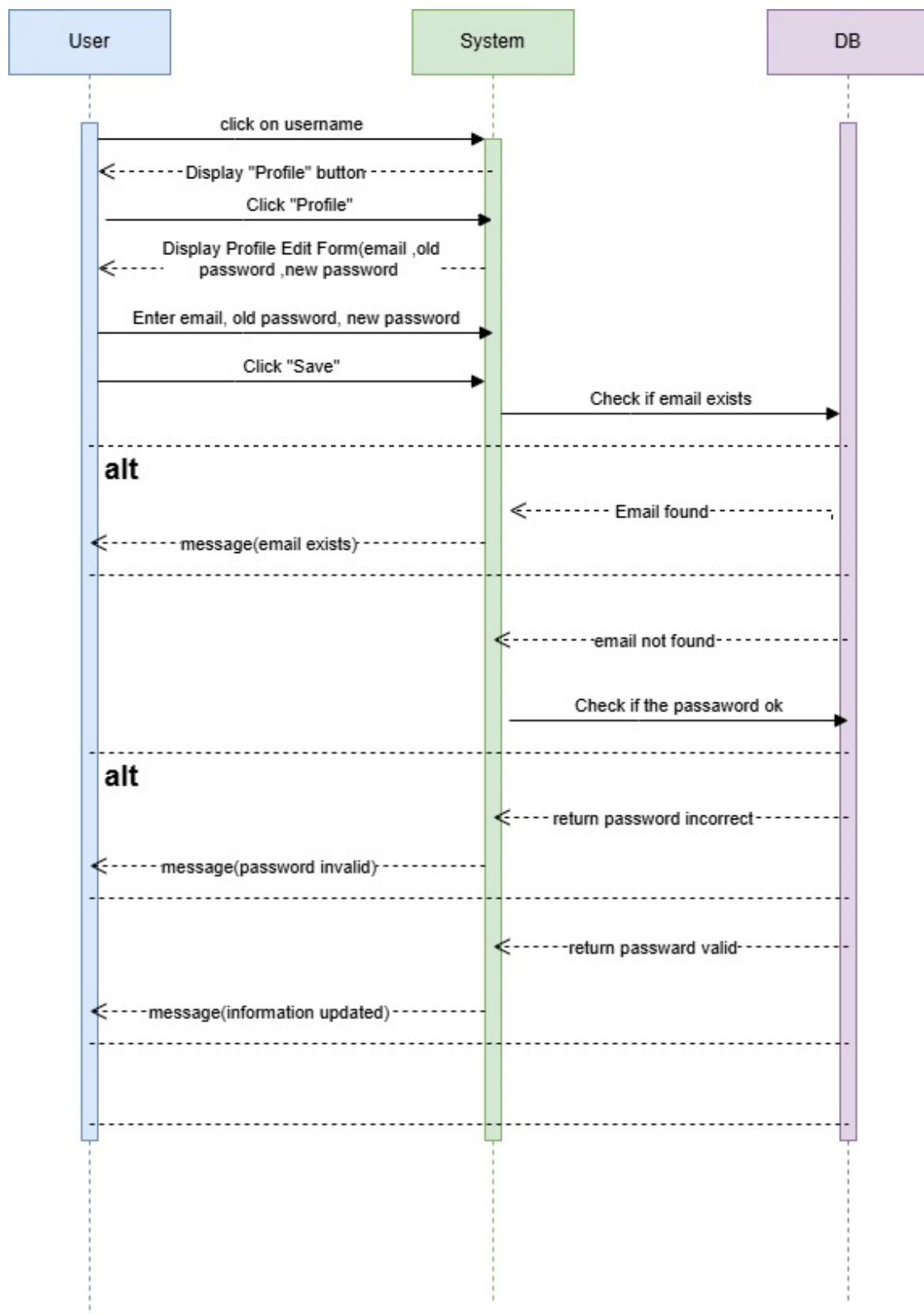


Figure 14 Sequence Diagram (Edit Profile)

## ● View Destinations:

<b>ID</b>	UC-06
<b>Use Case</b>	<b>View Destinations</b>
<b>Actor</b>	<b>User</b>
<b>Pre Condition</b>	The user is already logged into the system.
<b>Post Condition</b>	The user views available destinations
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks the <b>Destinations</b> button.</li> <li>2. The system displays a list of available destinations with image, name, and location.</li> </ol>
<b>Alternative Scenario</b>	

Table 15 View Destinations

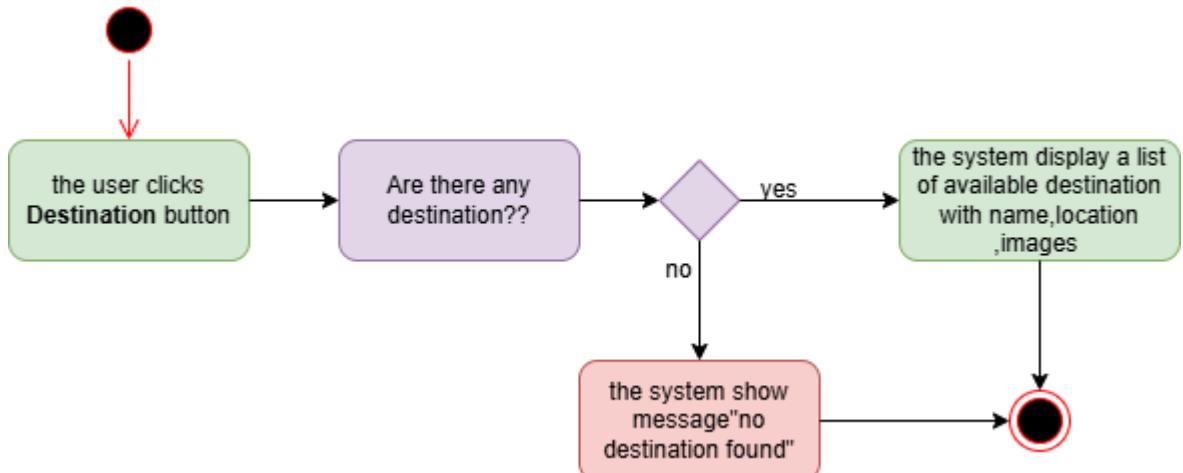


Figure 11 Activity Diagram(View Destinations)

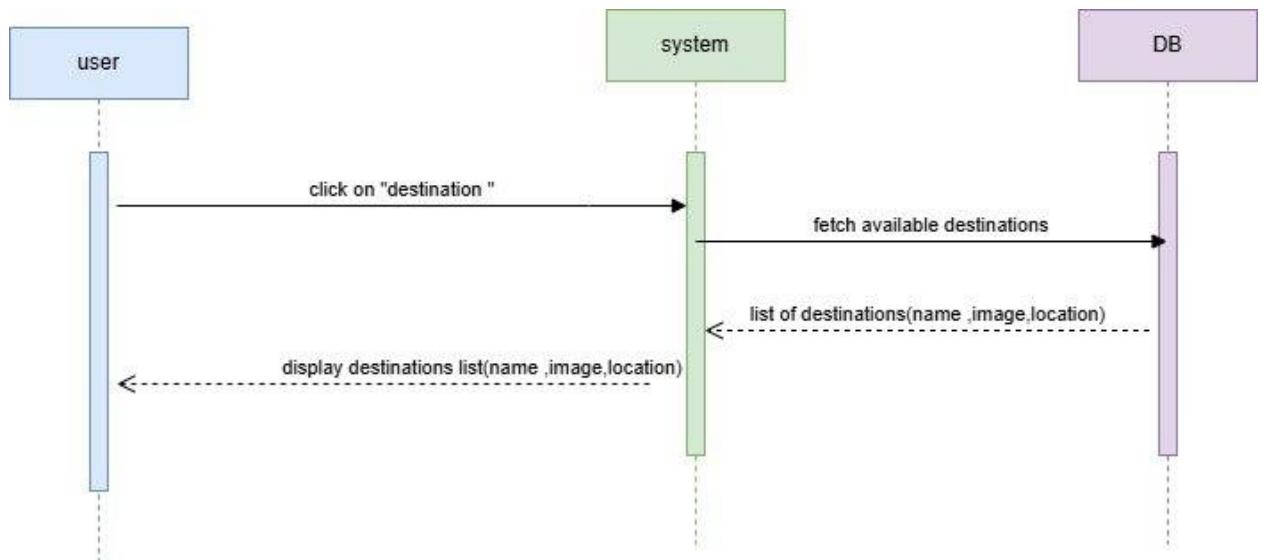


Figure 1 • Sequence Diagram(View Destinations)

## ● View Destination Details:

<b>ID</b>	<b>UC-07</b>
<b>Use Case</b>	<b>View Destination Details</b>
<b>Actor</b>	<b>User</b>
<b>Pre Condition</b>	The user is logged in and has accessed the destinations page
<b>Post Condition</b>	The user views detailed information about a selected destination
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks on a destination.</li> <li>2. The system displays detailed information (name, city, country, description, location details, weather, images, timezone, language, currency, nearest airport, best time to visit, emergency numbers, local tip, highlights,).</li> </ol>
<b>Alternative Scenario</b>	

Table 16 View Destination Details

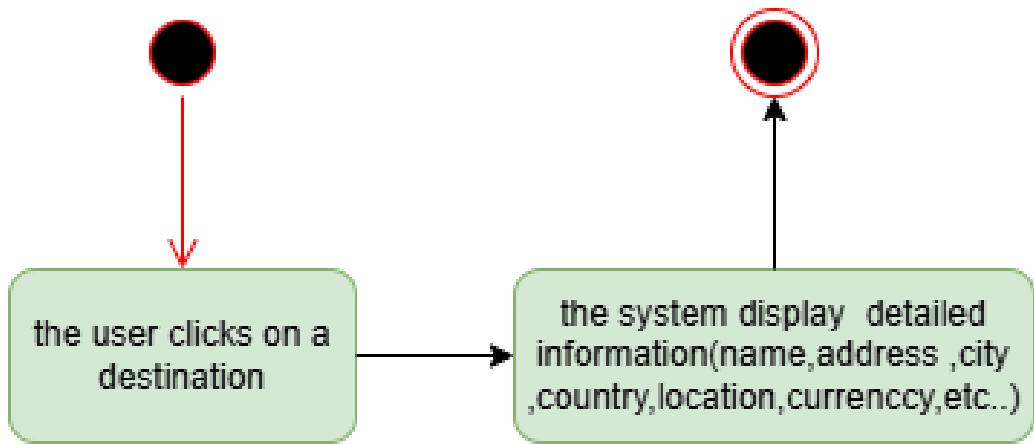


Figure 11 Activity Diagram (View Destination Details)

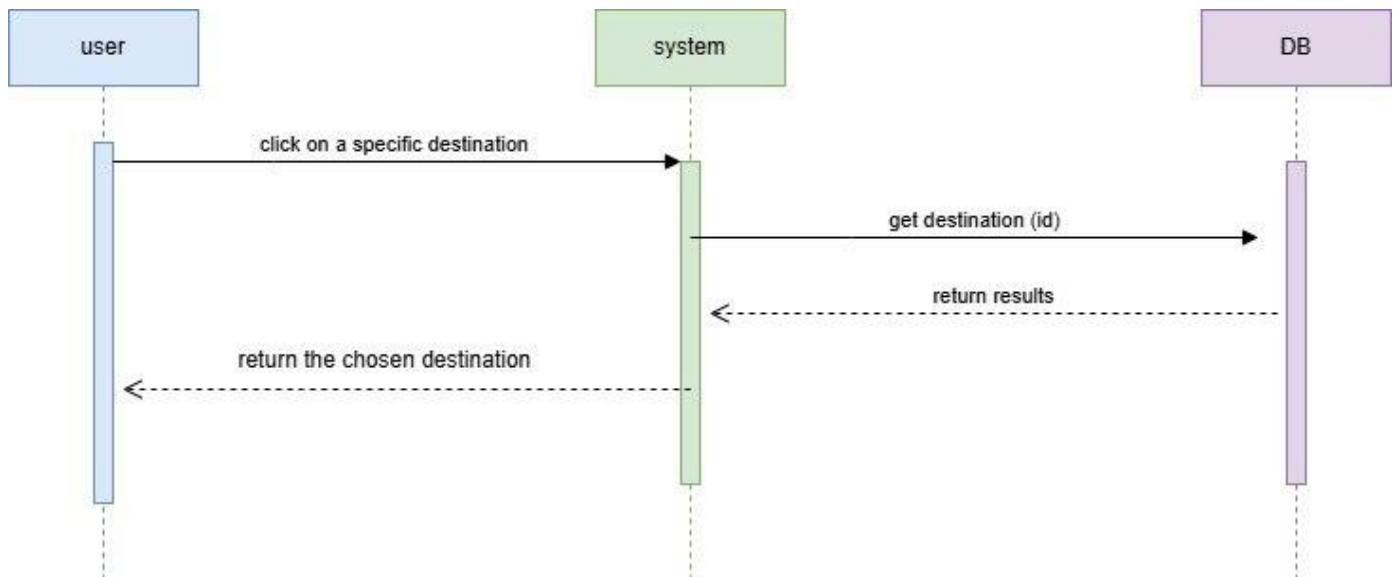


Figure 12 Sequence Diagram(View Destination Details)

## ● Search Destination:

<b>ID</b>	UC-08
<b>Use Case</b>	<b>Search Destination</b>
<b>Actor</b>	<b>User</b>
<b>Pre Condition</b>	The user is logged in and has accessed the destinations page
<b>Post Condition</b>	The user views a filtered list of destinations matching their criteria.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user enters a destination name or location in the search bar.</li> <li>2. The user clicks the <b>Search</b> button.</li> <li>3. The system displays a list of destinations matching the search criteria.</li> </ol>
<b>Alternative Scenario</b>	Step 2: If no results are found, the system shows a message indicating no matches

Table 17 Search Destination

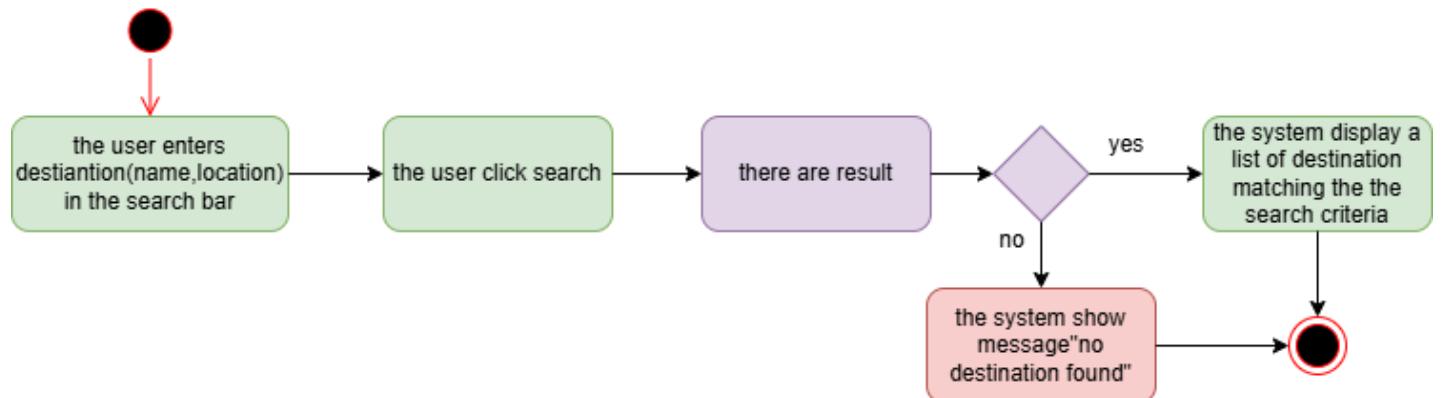


Figure 11 Activity Diagram (Search Destination)

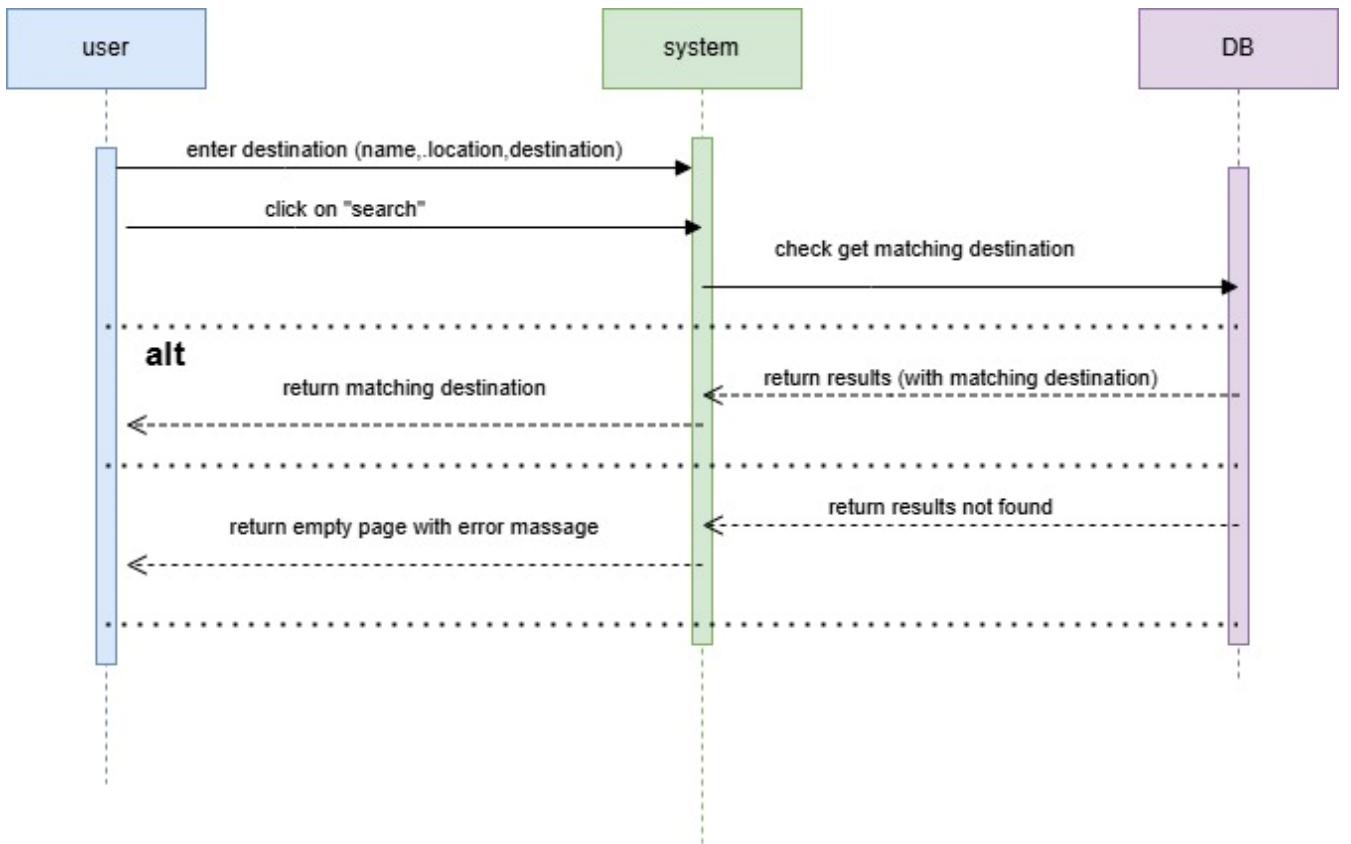


Figure 14 Sequence Diagram(Search Destination)

## ● Add Destination :

<b>ID</b>	<b>UC-09</b>
<b>Use Case</b>	<b>Add Destination</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The Admin is logged into the system The admin is on destination pages
<b>Post Condition</b>	A new destination is added
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks the <b>Add New Destiantion</b> button.</li> <li>2. The system displays a form containing:( Destination Name, City, Country, Description, location_details, IATA Code, Images, timezone, language, currency, nearest_airport, best_time_to_visit, emergency_numbers, local_tip, Highlight)</li> <li>3. The admin enters the information</li> <li>4. The admin click create destination</li> <li>5. The system validate the information</li> <li>6. The system redirect the admin to the destination list page with the success message</li> </ol>
<b>Alternative Scenario</b>	<p>Step5:If information is missing the system display error message</p> <p>Step5: If the destination already exists, the system returns an error message and redirects the admin .back to step 3</p>

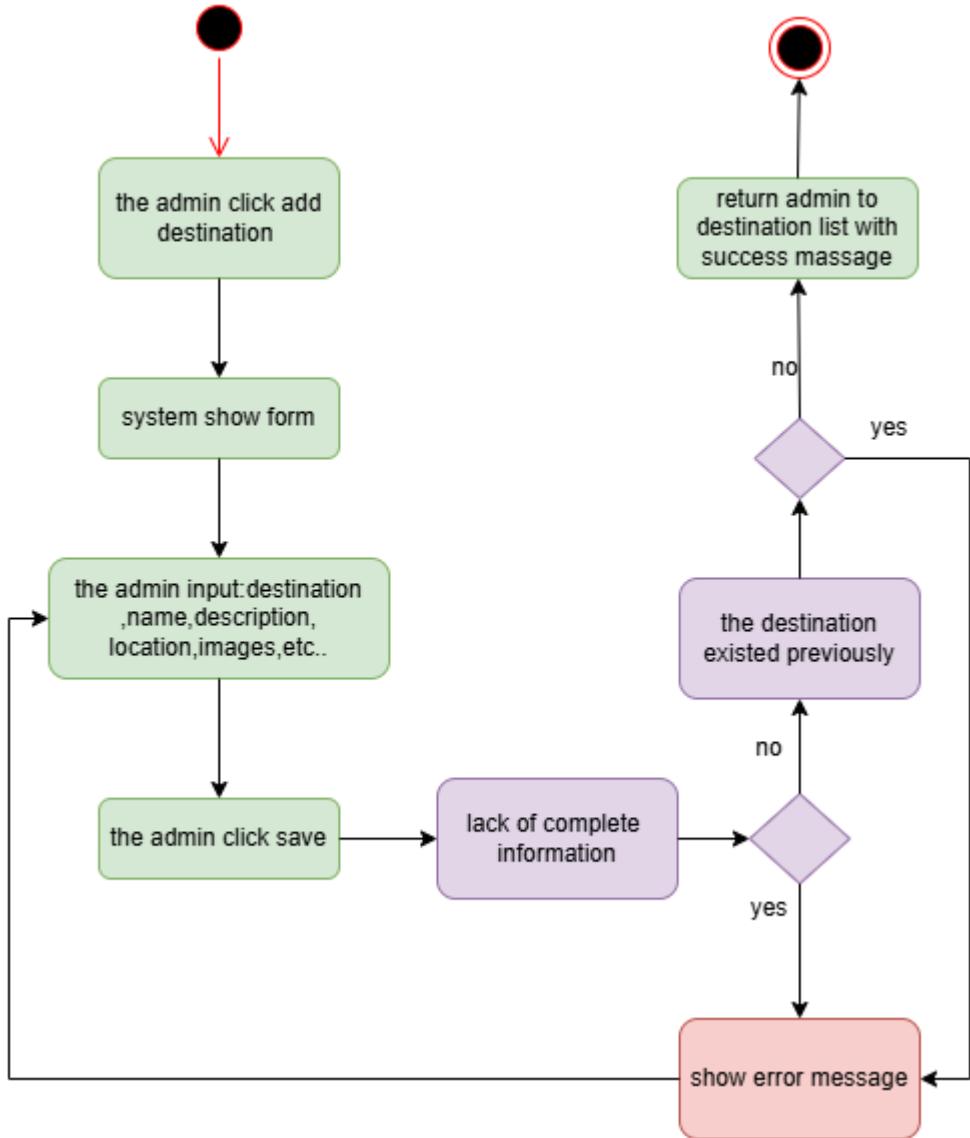


Figure 10 Activity Diagram (Add Destination)

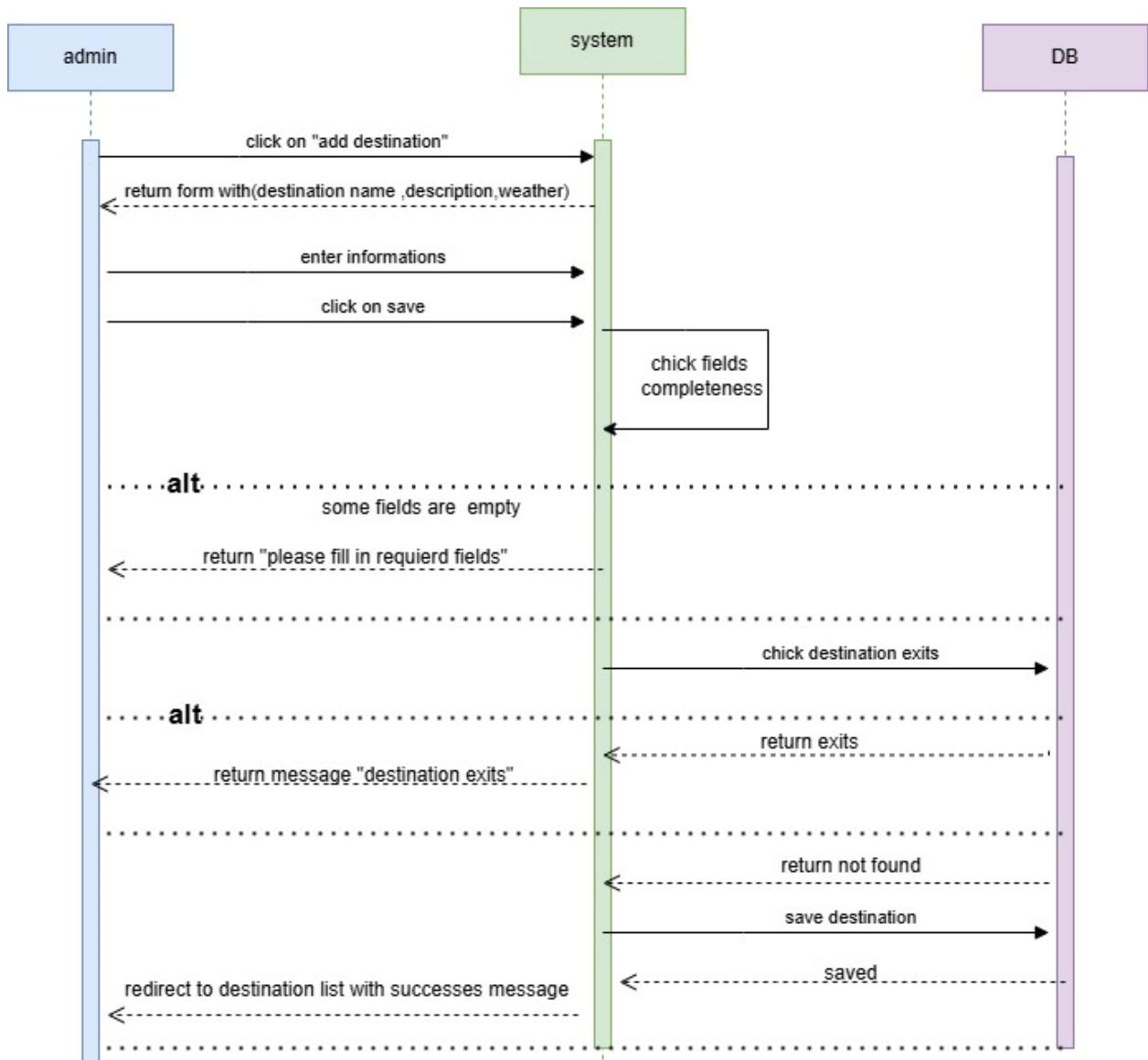


Figure 11 Sequence Diagram(Add Destination)

## ● Edit Destination :

<b>ID</b>	<b>UC-10</b>
<b>Use Case</b>	<b>Edit Destination</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The Admin is logged into the system The admin is on destination pages
<b>Post Condition</b>	Destination information is updated
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks the <b>Edit</b> button for the selected destination.</li> <li>2. The system displays a form with current destination details.</li> <li>3. The admin modifies the information</li> <li>4. The admin click Save</li> <li>5. The system validates the updated information.</li> <li>6. The system updates the destination</li> <li>7. The admin redirect to the destination list page with a success message .</li> </ol>
<b>Alternative Scenario</b>	<p>Step5: If the destination already exists, the system returns an error message and redirects the admin back to step 3.</p> <p>Step5:if information in invalid ,the system returns an error message.</p>

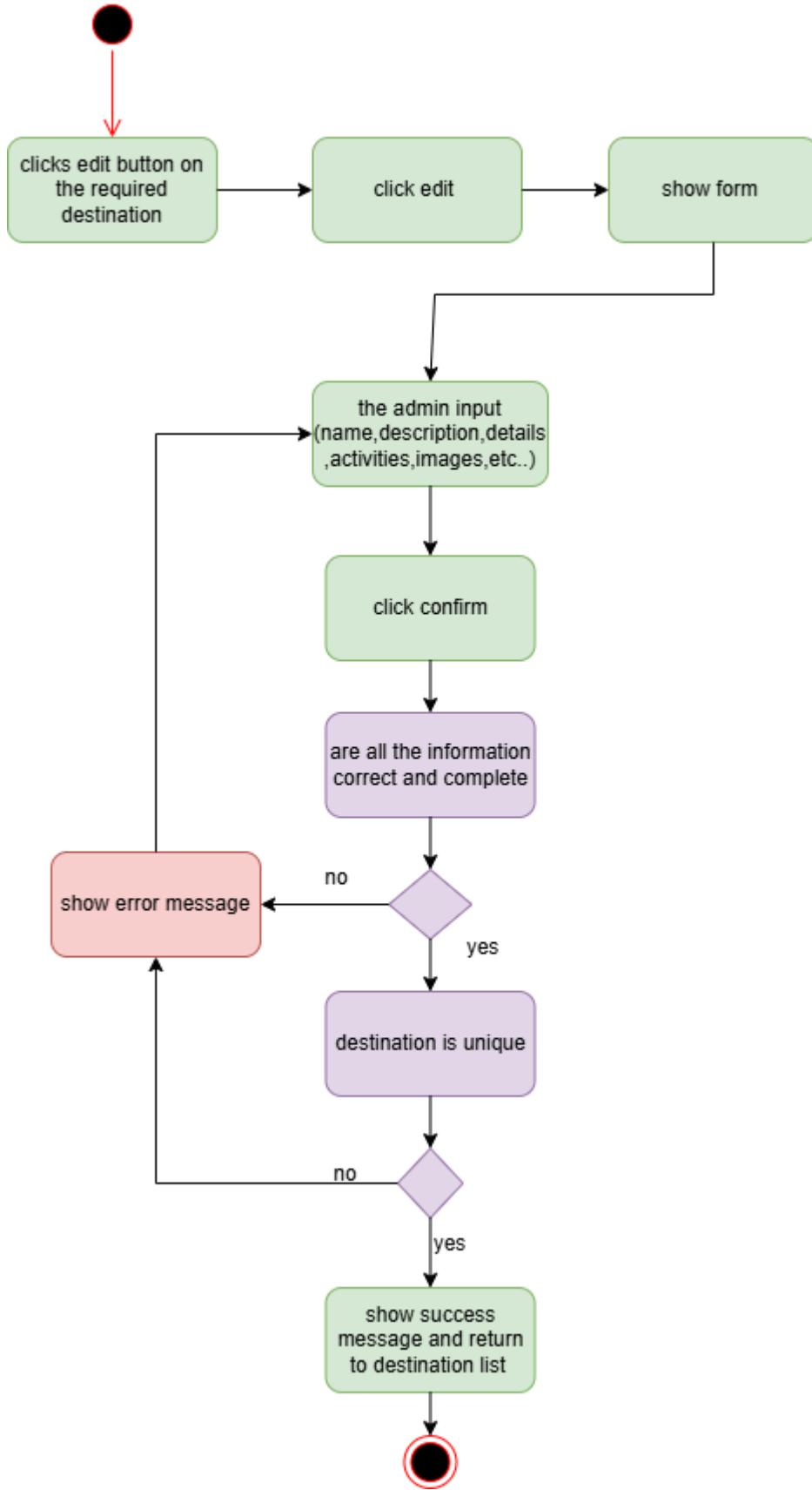


Figure 11 Activity Diagram(Edit Destination )

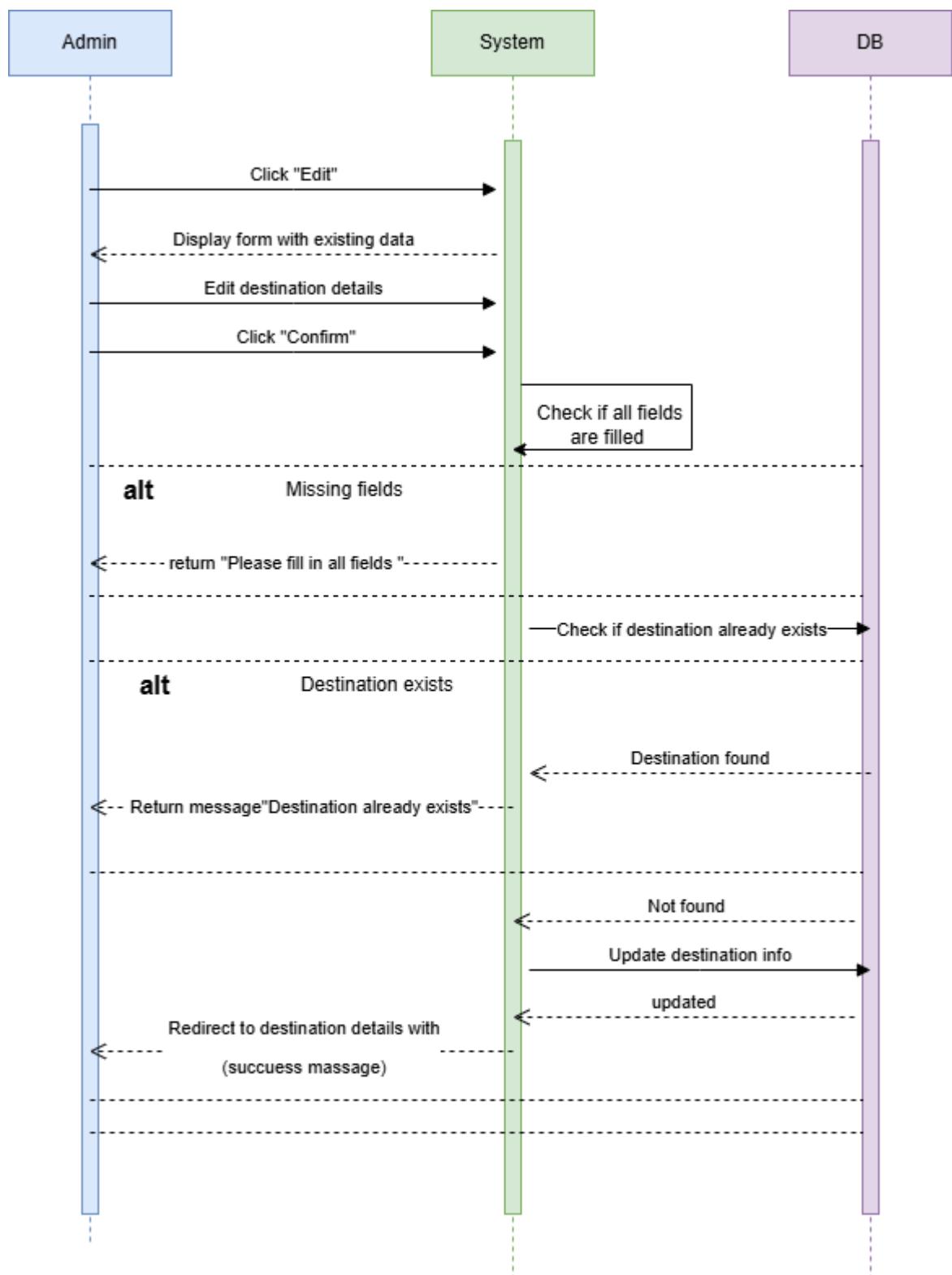


Figure 14 Sequence Diagram(Edit Destination)

## ●Delete Destination:

ID	UC-11
Use Case	Delete Destination
Actor	Admin
Pre Condition	The Admin is logged into the system The admin is on destination pages
Post Condition	Destination is deleted
Main Scenario	<ol style="list-style-type: none"> <li>1. The admin clicks the <b>Delete button</b></li> <li>2. The system delete the destination</li> <li>3. The system display a success message</li> </ol>
Alternative Scenario	

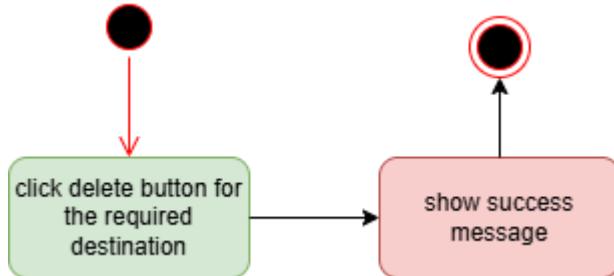


Figure 19 Activity Diagram (Delete Destination)

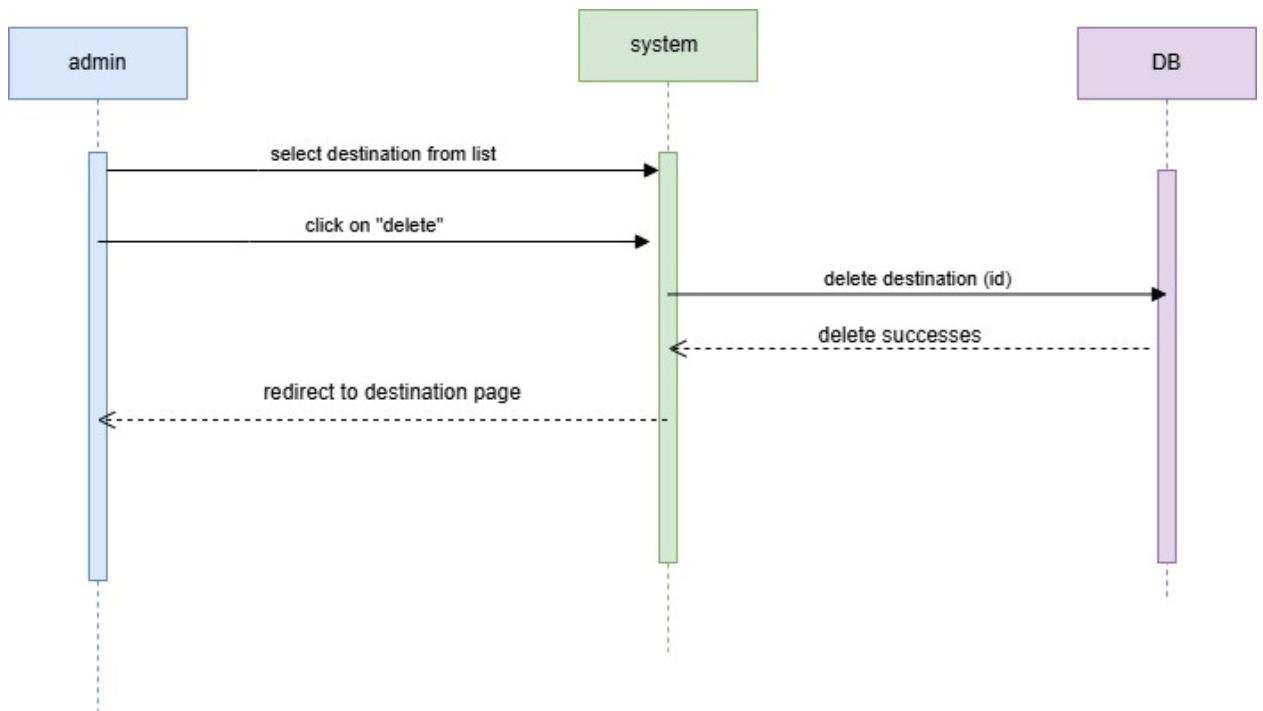


Figure 19 Sequence Diagram(Delete Destination)

## ● View Hotels :

<b>ID</b>	<b>UC-12</b>
<b>Use Case</b>	<b>View Hotels</b>
<b>Actor</b>	<b>User/Admin</b>
<b>Pre Condition</b>	The user is logged into the system
<b>Post Condition</b>	The user views available hotels
<b>Main Scenario</b>	<p>4. The user clicks the <b>Hotels</b> button.</p> <p>5. The system displays a list of available hotels with image, name, and location.</p>
<b>Alternative Scenario</b>	Step 2: If no results are found, the system shows "no hotels found"

Table 18 View Hotels

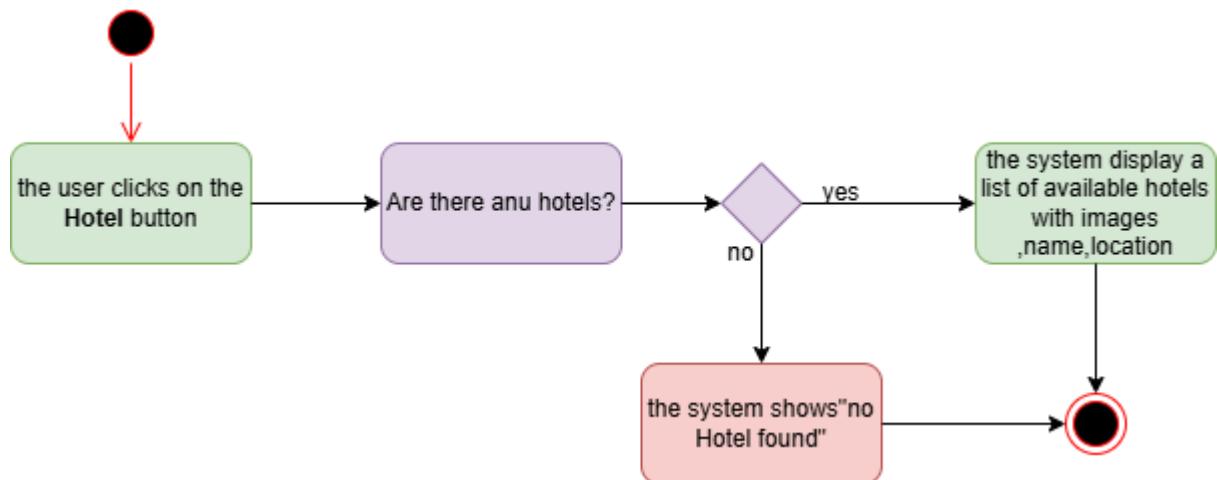


Figure 11 Activity Diagram(View Hotels)

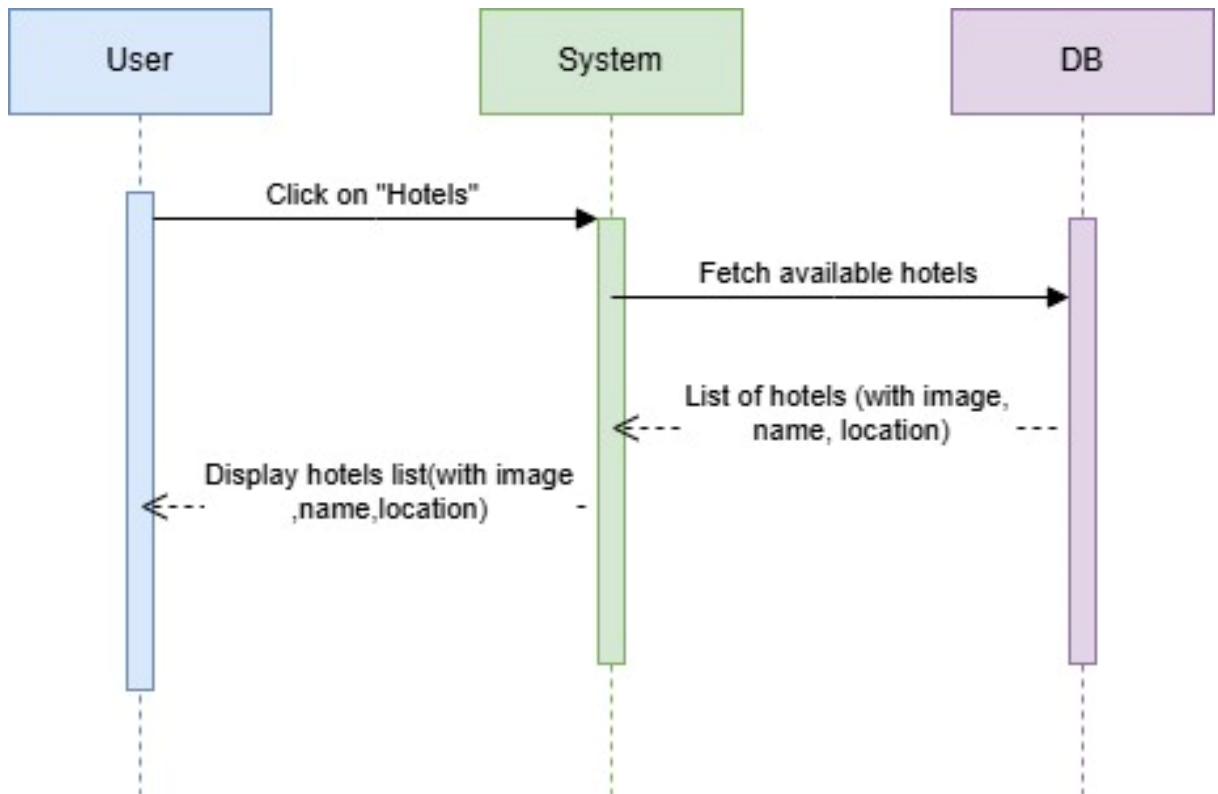


Figure 11 Sequence Diagram(View Hotels)

## ● View Hotel Details:

<b>ID</b>	UC-13
<b>Use Case</b>	<b>View Hotel Details</b>
<b>Actor</b>	<b>User/Admin</b>
<b>Pre Condition</b>	The user is logged in and has accessed the hotels page.
<b>Post Condition</b>	The user views detailed information about a selected hotel.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks on a hotel.</li> <li>2. The system displays detailed information (name, description ,address, city, country, global rating, price per night, total rooms, stars, amenities, pets allowed, check in time, check out time, policies, phone number, email, website, nearby landmarks, images)</li> </ol>
<b>Alternative Scenario</b>	

Table 19 View Hotel Details

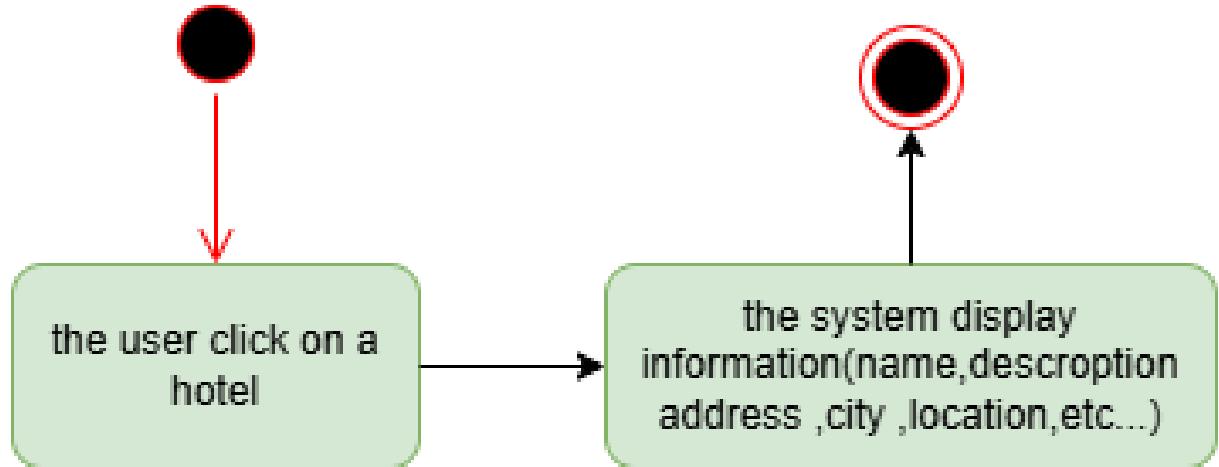


Figure 11 Activity Diagram (View Hotel Details)

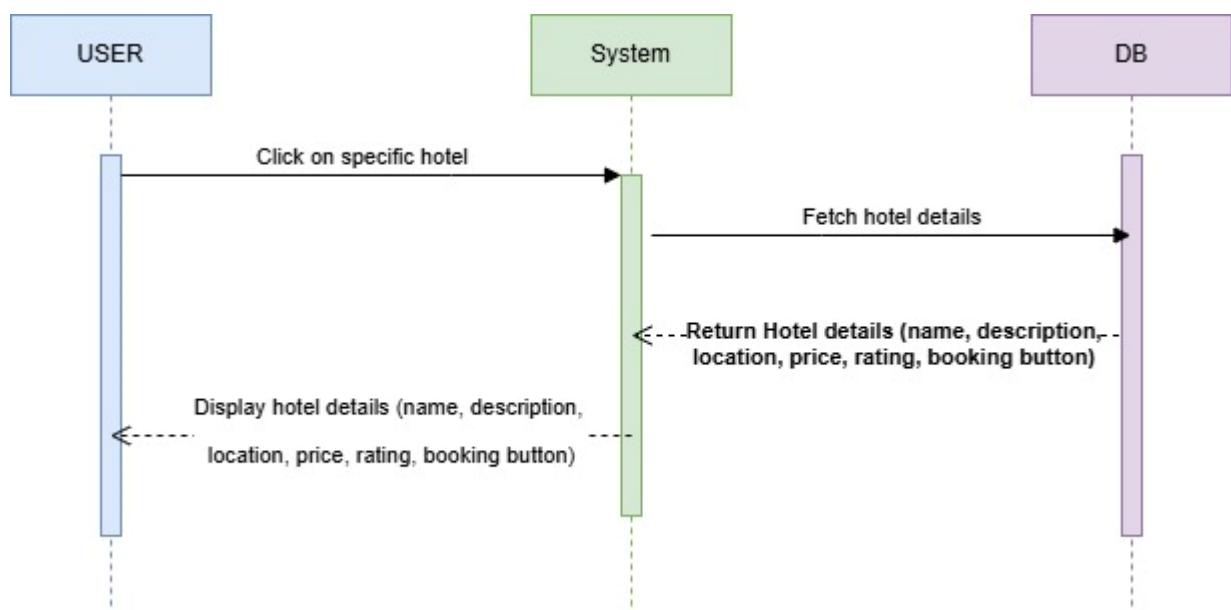


Figure 12 Sequence Diagram (View Hotel Details)

## ● Search Hotels:

<b>ID</b>	UC-14
<b>Use Case</b>	<b>Search Hotels</b>
<b>Actor</b>	<b>User/Admin</b>
<b>Pre Condition</b>	The user is logged in and has accessed the hotels page
<b>Post Condition</b>	The system views a list of hotels matching the search criteria.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user enters a hotel name or location or global rating in the search bar.</li> <li>2. The user clicks the <b>Search</b> button.</li> <li>3. The system displays a list of hotels matching the search criteria.</li> </ol>
<b>Alternative Scenario</b>	Step 2: If no results are found, the system shows a message indicating no matches.

Table 20 Search Hotels

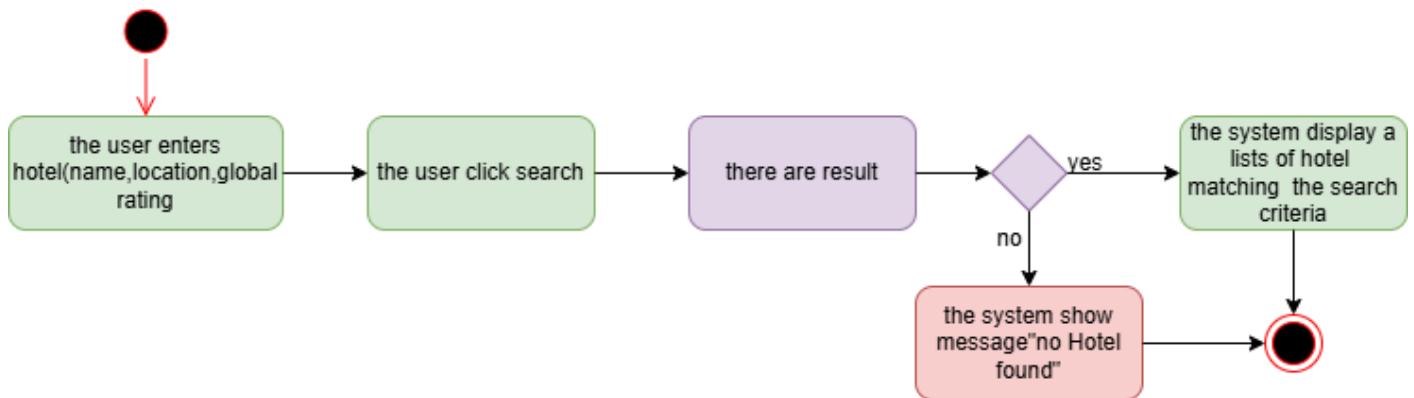


Figure 10 Activity Diagram (Search Hotels)

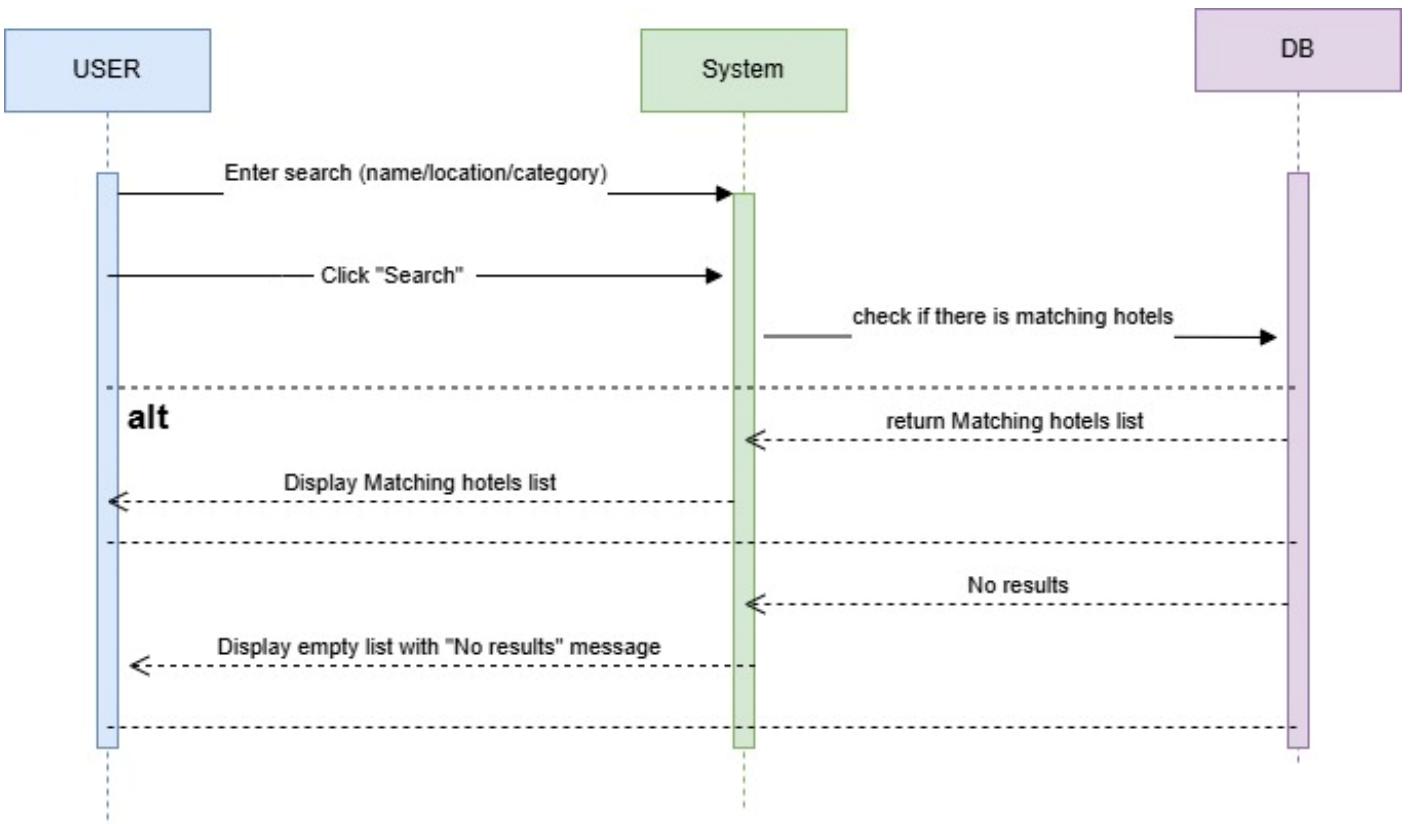


Figure 77 Sequence Diagram (Search Hotels)

## ● Add Hotel:

<b>ID</b>	<b>UC-15</b>
<b>Use Case</b>	<b>Add Hotel</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The Admin is logged into the system The admin is on hotel page
<b>Post Condition</b>	A new hotel is added
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks the <b>Add New Hotel</b> button.</li> <li>2. The system displays a form containing:( Hotel Name, Associated destination, City, Country, Description, address, global_rating, price_per_night, total_rooms, stars, Images, Pets Allowed, check_in_time, check_out_time, policies, phone_number, email, website, nearby_landmarks,Amenities)</li> <li>3. The admin enters the information</li> <li>4. The admin click create Hotel</li> <li>5. Th system validates the information</li> <li>6. The system redirect the admin to the hotel list page with the success message</li> </ol>
<b>Alternative Scenario</b>	<p>Step5:If information is missing the system display error message</p> <p>Step5: If the hotel already exists, the system returns an error message and redirects the admin back to step 3</p>

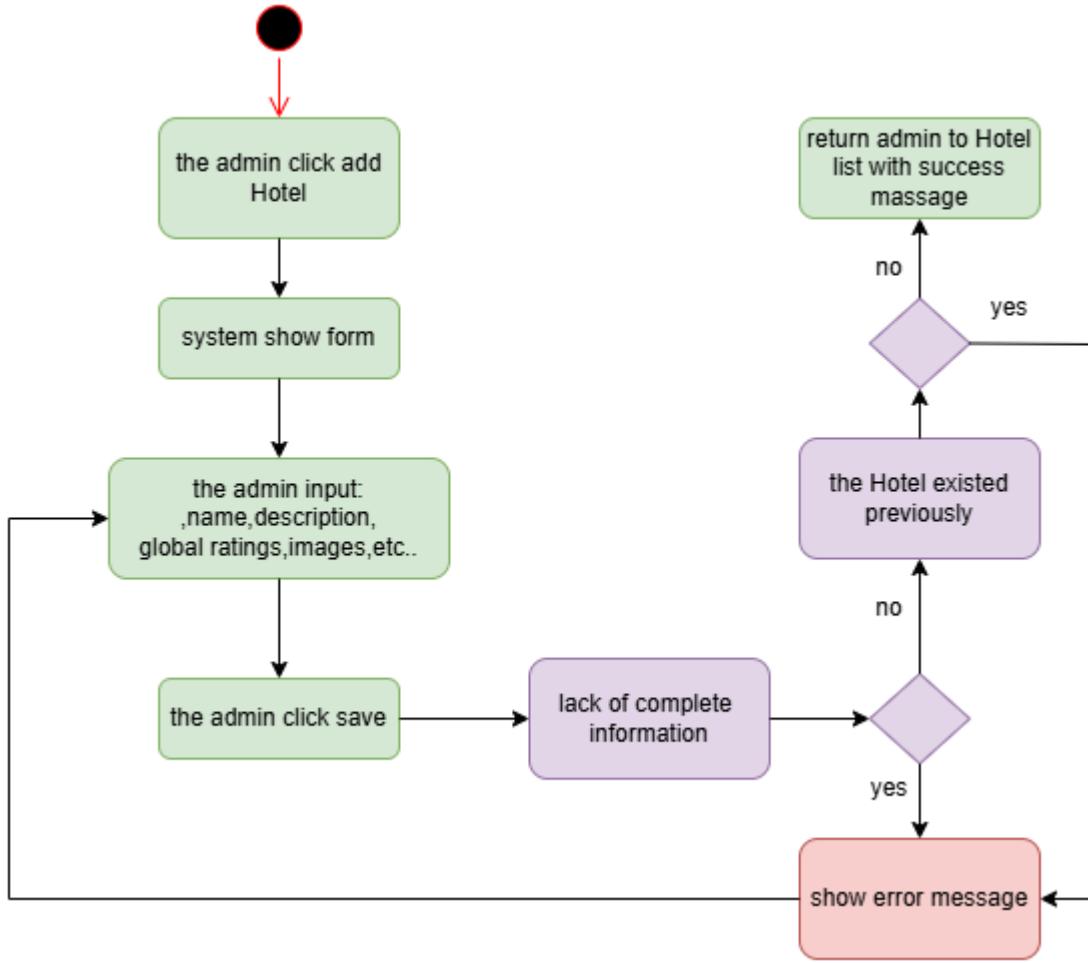


Figure 14 Activity Diagram(Add Hotel)

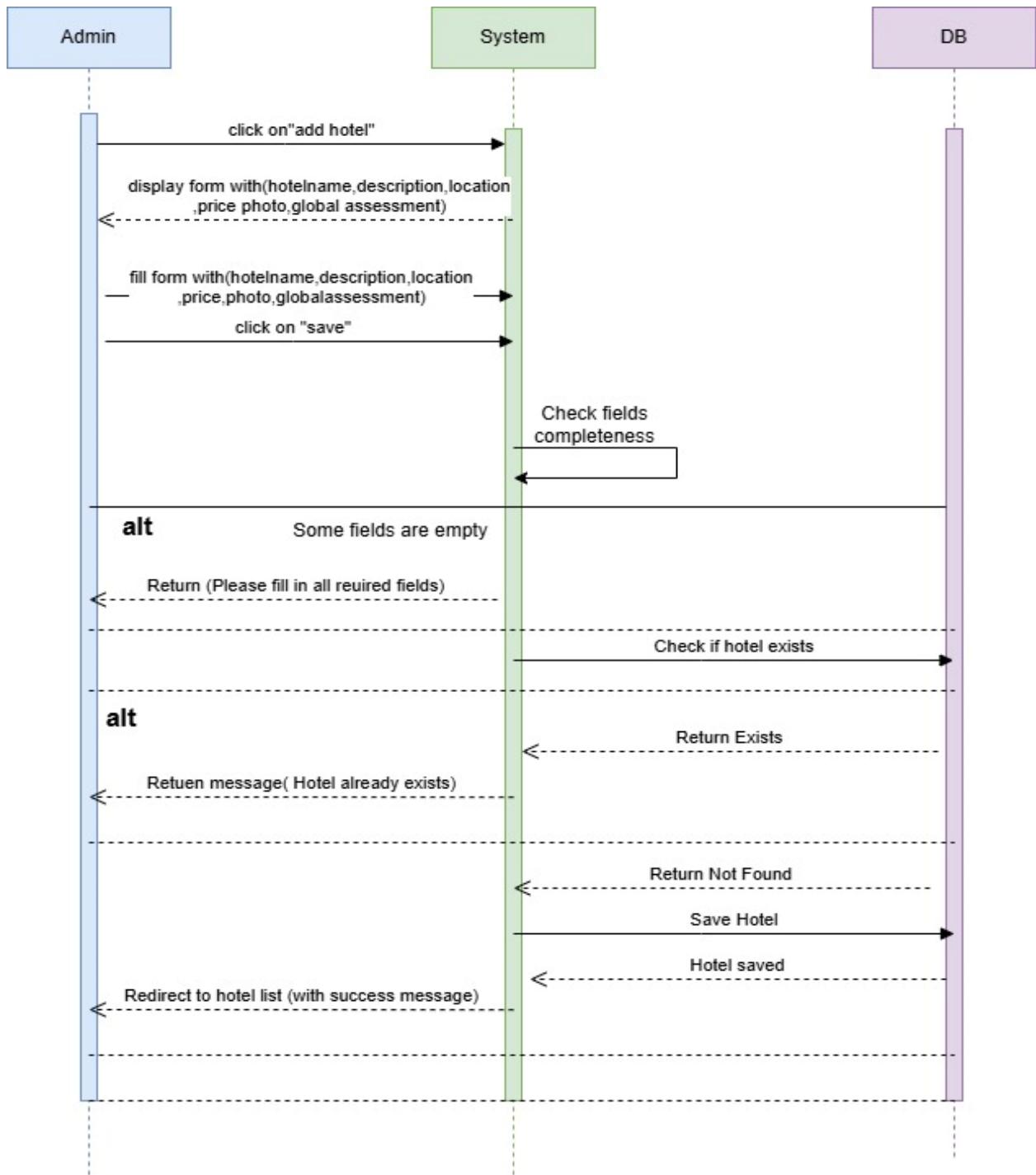


Figure 38 Sequence Diagram (Add Hotel)

## ●Edit Hotel:

<b>ID</b>	<b>UC-16</b>
<b>Use Case</b>	<b>Edit Hotel</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The Admin is logged into the system The admin is on hotel page
<b>Post Condition</b>	Hotel information is updated
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks the <b>Edit</b> button for the selected hotel.</li> <li>2. The system displays a form with current hotel details.</li> <li>3. The admin modifies the information</li> <li>4. The admin click Save</li> <li>5. The system validates the updated information.</li> <li>6. The system updates the hotel</li> <li>7. The admin redirect to the hotel list page with a success message .</li> </ol>
<b>Alternative Scenario</b>	<p>Step5: If the hotel already exists, the system returns an error message and redirects the admin back to step 3.</p> <p>Step5:if information in invalid ,the system returns an error message.</p> <p>Step5: If the hotel has reservations,the system return an error message</p>

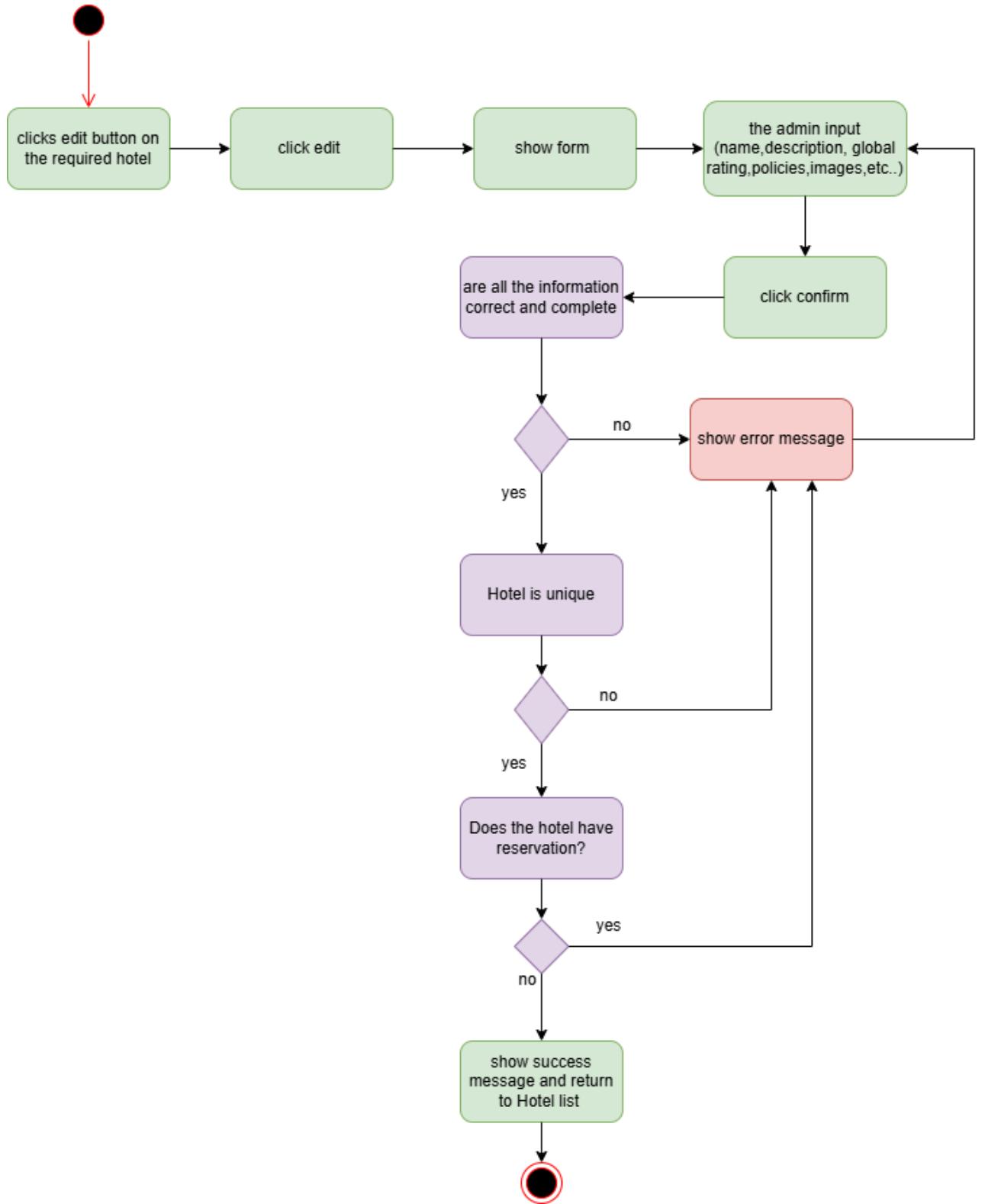


Figure 19 Activity Diagram (Edit Hotel)

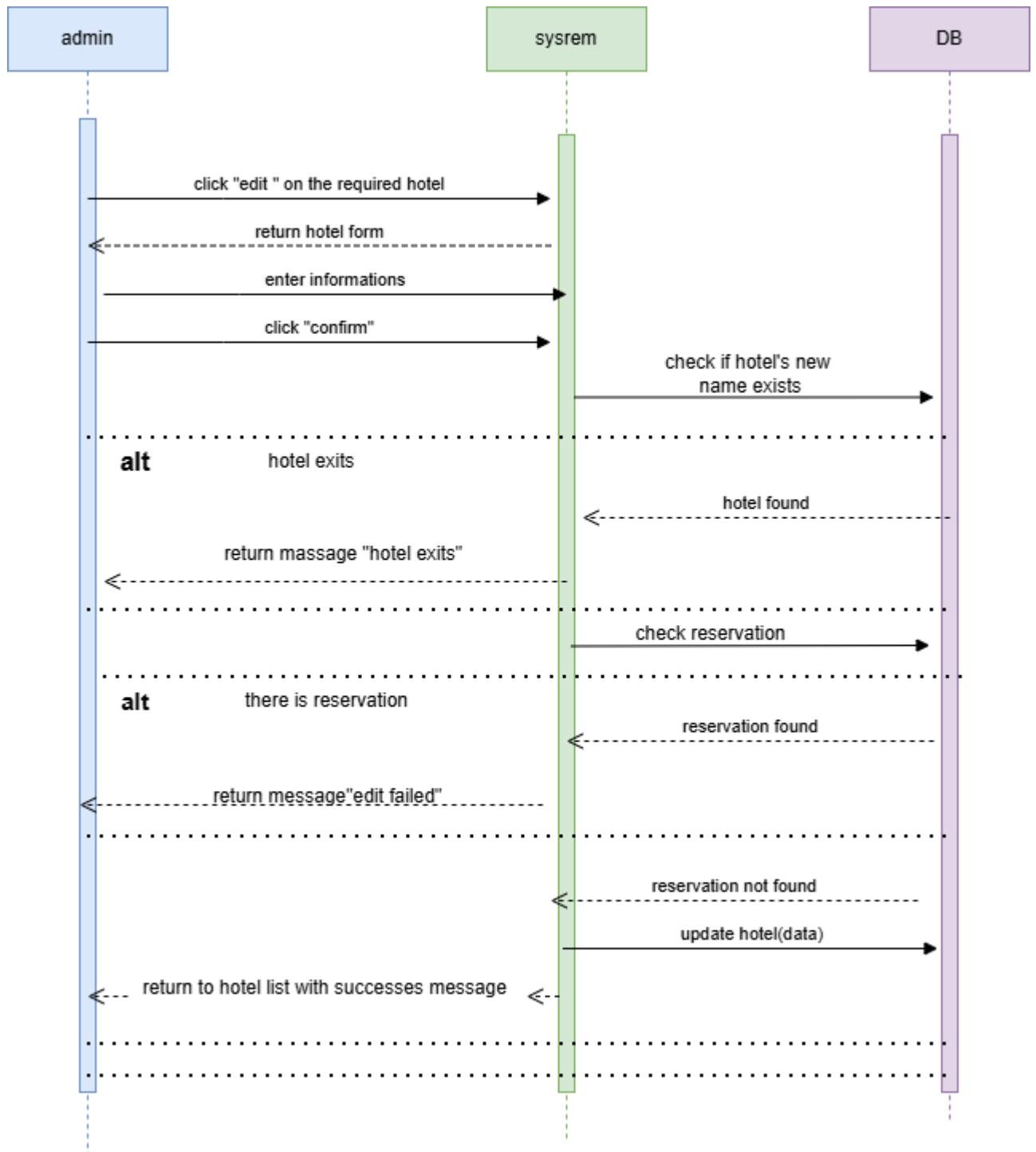


Figure 5 • Sequence Diagram(Edit Hotel)

## ● Delete Hotel:

<b>ID</b>	UC-17
<b>Use Case</b>	<b>Delete Hotel</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The Admin is logged into the system The admin is on hotel pages
<b>Post Condition</b>	Hotel is deleted
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks the <b>Delete button</b></li> <li>2. The system checks if there are coming reservation linked to the Hotel</li> <li>3. The system delete the hotel</li> <li>4. The system display a success message</li> </ol>
<b>Alternative Scenario</b>	If linked coming reservation exist->deletion Is blocked and error message is shown

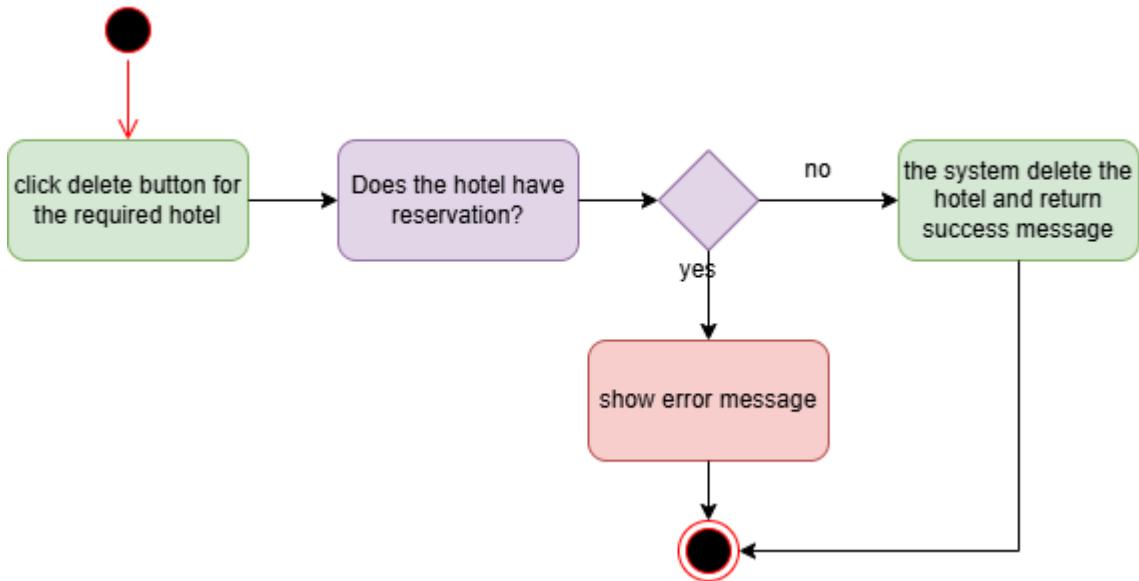


Figure 5.1 Activity Diagram (Delete Hotel)

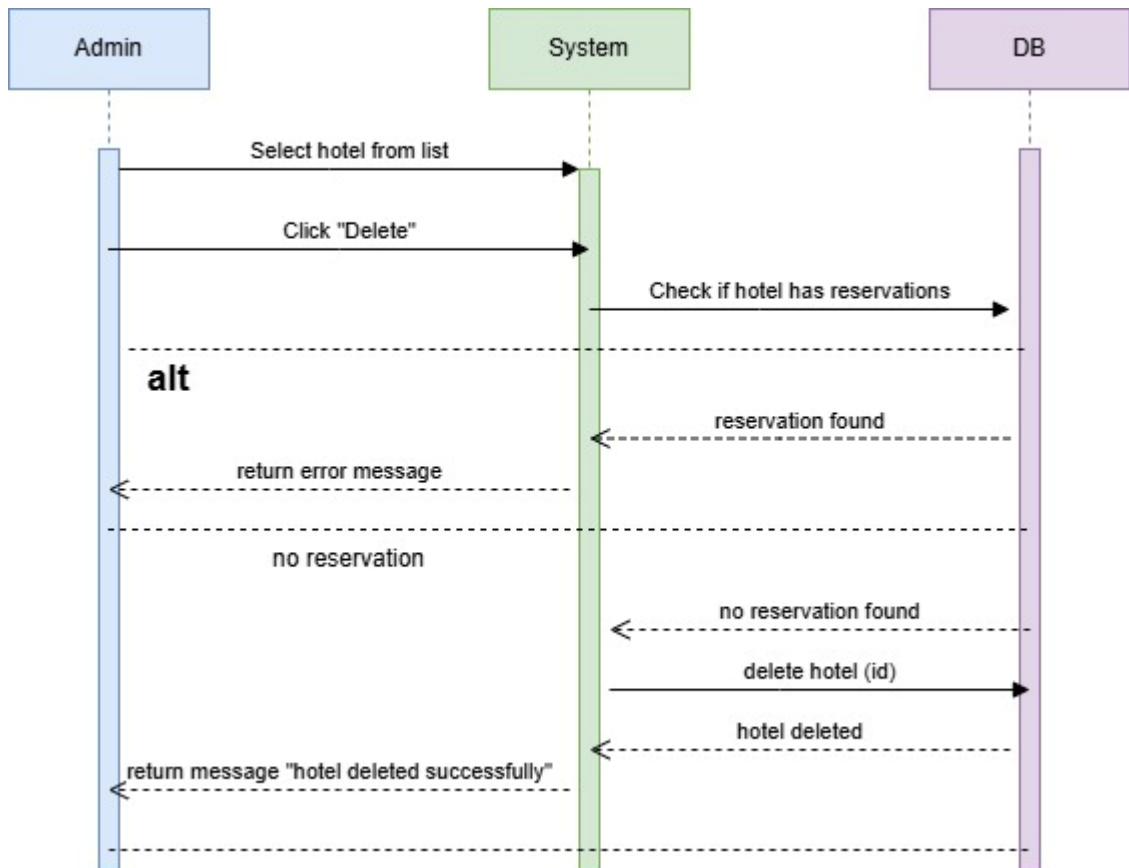


Figure 41 Sequence Diagram (Delete Hotel)

## ● Book Hotel:

<b>ID</b>	<b>UC-18</b>
<b>Use Case</b>	<b>Reserve Hotels</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The user is logged in and is on hotels page.
<b>Post Condition</b>	A hotel booking confirmed.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks the <b>Book</b> button for a specified hotel.</li> <li>2. The system displays a form containing (number of persons, rooms required, check in and check out dates, name, email “??</li> <li>3. The user fills in the information .</li> <li>4. The user clicks confirm.</li> <li>5. The system prompts the user to processed the payment.</li> </ol>
<b>Alternative Scenario</b>	<p>Step 3: if no rooms are available for the selected dates an error message is shown.</p> <p>Step 5: if required fields are missing an error message appears.</p>

Table 21 Reserve Hotels

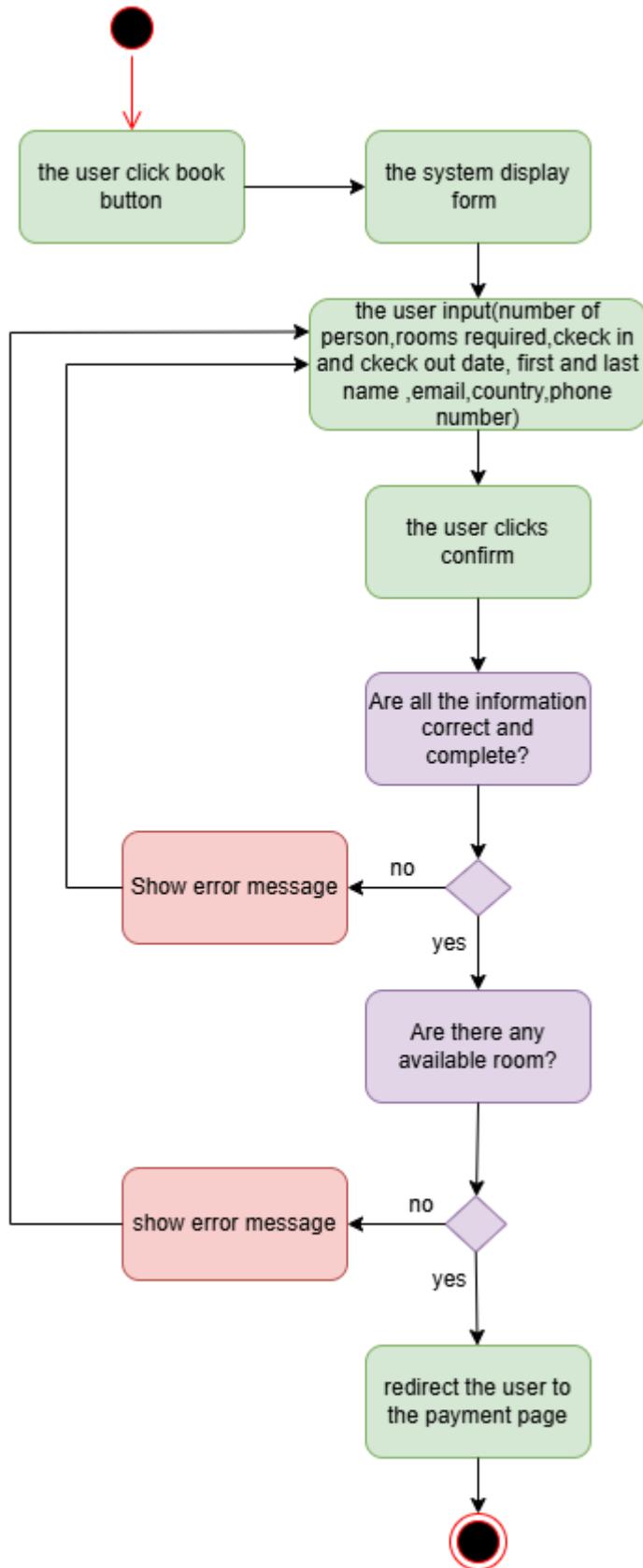


Figure 4 Activity Diagram(Book Hotel)

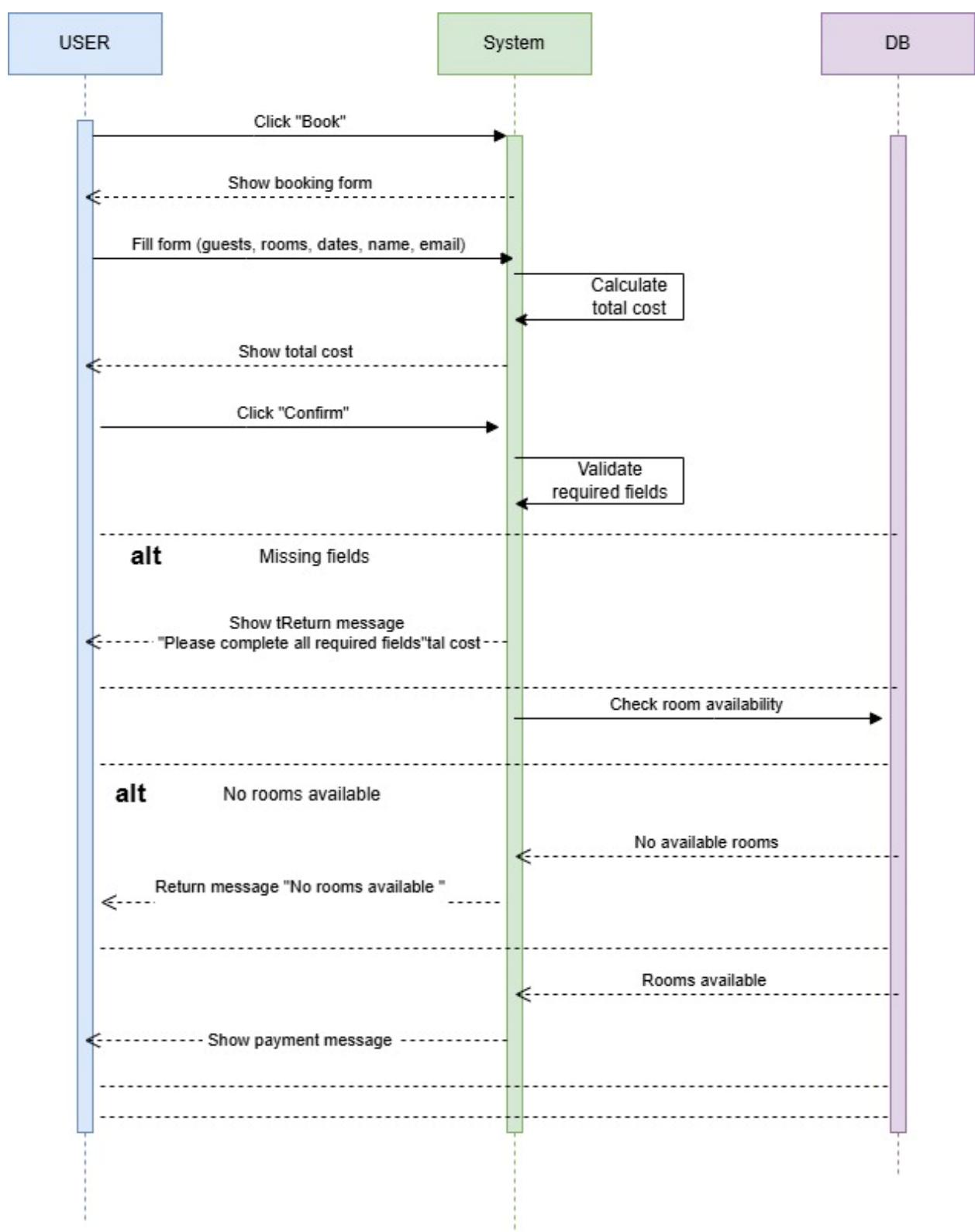


Figure 55 Sequence Diagram(Book Hotel)

● Pay:

<b>ID</b>	<b>UC-19</b>
<b>Use Case</b>	<b>Pay</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The user has confirmed a hotel booking.
<b>Post Condition</b>	Payment is completed and a confirmation email is sent.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks the pay button.</li> <li>2. The system redirect the user to the external payment gateway.</li> <li>3. The user enters the payment details.</li> <li>4. Upon success the system displays a confirmation message and redirect the user back.</li> <li>5. A confirmation email with booking details is sent.</li> </ol>
<b>Alternative Scenario</b>	If payment fails an error message with reason of failure is displayed.

Table 22 Pay

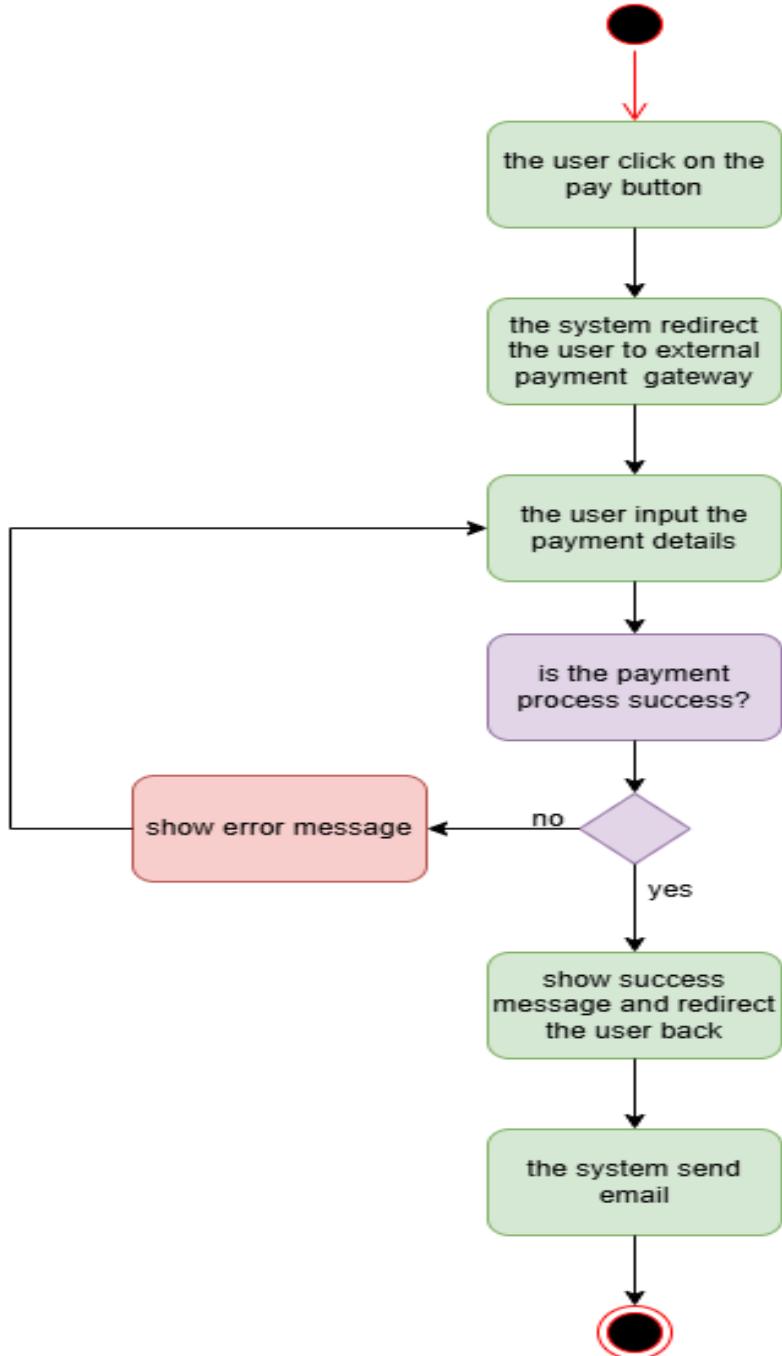


Figure 50 Activity Diagram(Pay)

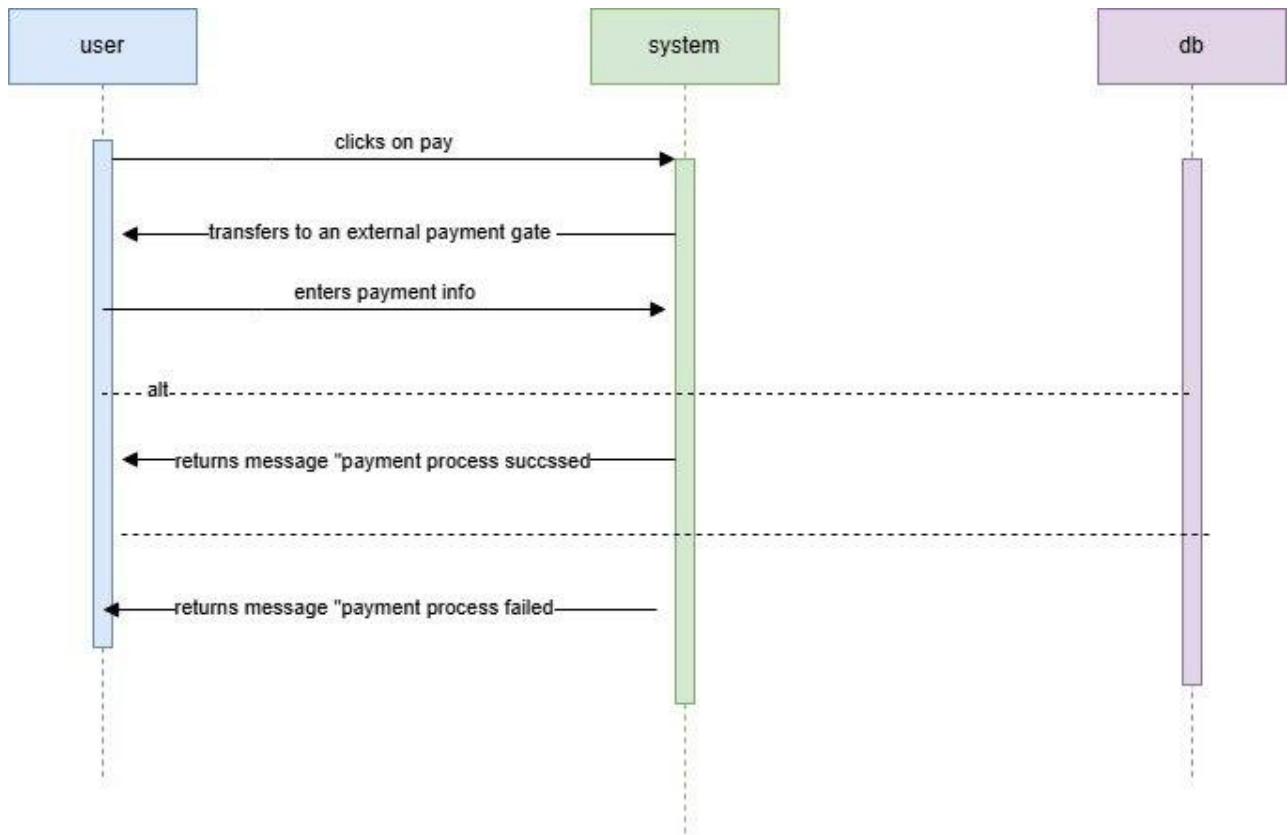


Figure 27 Sequence Diagram(Pay)

## ● Flight search:

<b>ID</b>	<b>UC-20</b>
<b>Use Case</b>	<b>Flight search</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The user is logged into the system.
<b>Post Condition</b>	The user views available flight options
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks flights from the navigation bar.</li> <li>2. The system displays a form with (destination, travel dates, number of passengers and trip type “round-trip or one way”).</li> <li>3. The user enters the details.</li> <li>4. The user clicks search.</li> <li>5. The system retrieves available flights data via an API .</li> <li>6. The system displays the results</li> </ol>
<b>Alternative Scenario</b>	If no flights are found the system shows a message indicating no available results.

Table 23 Flight Search

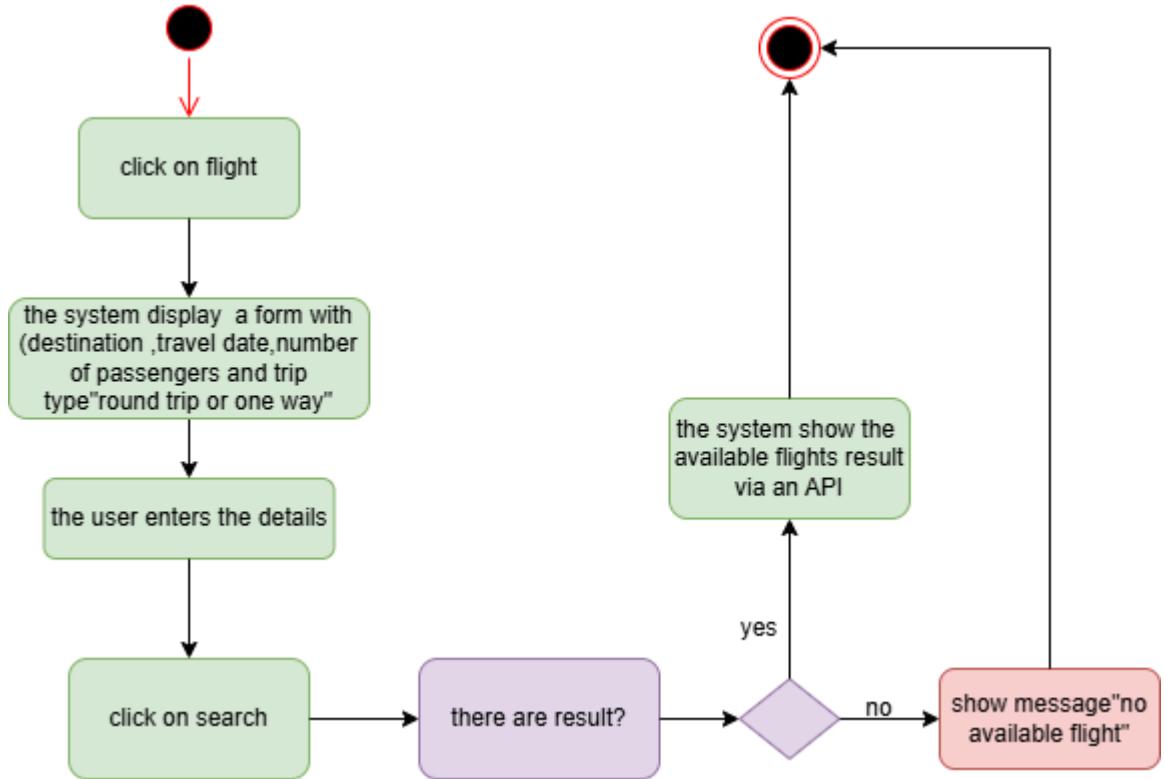


Figure 5 V Activity Diagram(Flight search)

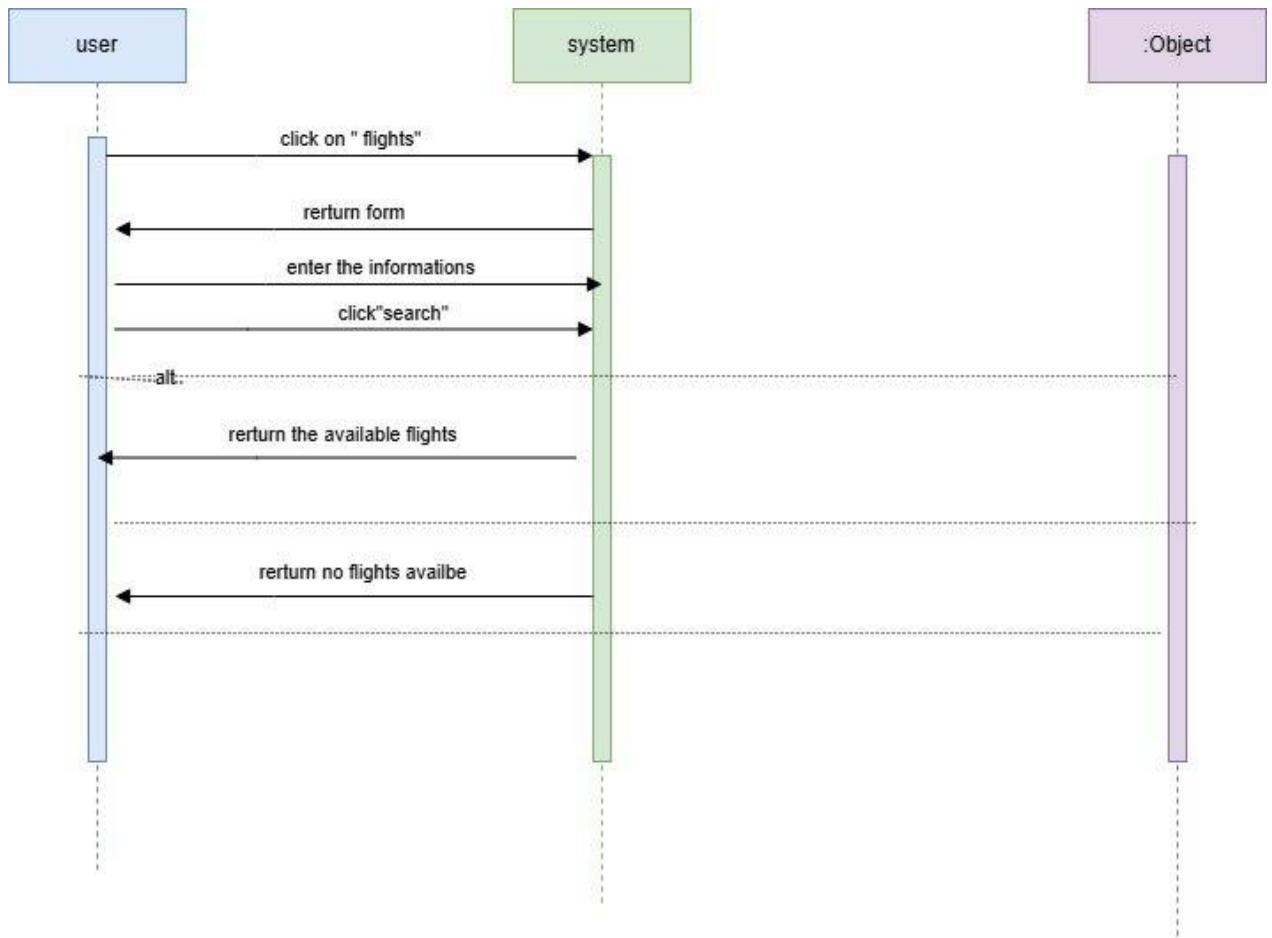


Figure 4 Sequence Diagram(Flight search)

## ● Add Activity:

<b>ID</b>	<b>UC-21</b>
<b>Use Case</b>	<b>Add activity</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The admin is logged into the system.
<b>Post Condition</b>	A new activity is successfully added. The admin on the Activity list.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Add New Activity.</li> <li>2. The system displays a form containing the following fields: (Activity Name, Price, Category, Description, Is Active, Destination, Duration, Difficulty Level, Duration Unit, Amenities, Address, Start Time, End Time, Start Date, End Date, Requires Booking?, Pets Allowed?, Family Friendly?, Availability, Highlights, Guide Name, Activity Image, Guide Language, Contact Number, Requirements).</li> <li>3. The admin enters the required information.</li> <li>4. The admin clicks Save.</li> <li>5. The system validates the entered information.</li> <li>6. The system adds the activity.</li> <li>7. The admin is redirected to activities list page with a success message.</li> </ol>
<b>Alternative Scenario</b>	If information is incomplete or invalid in step 6, the system returns an error message.

Table 24 Add Activity

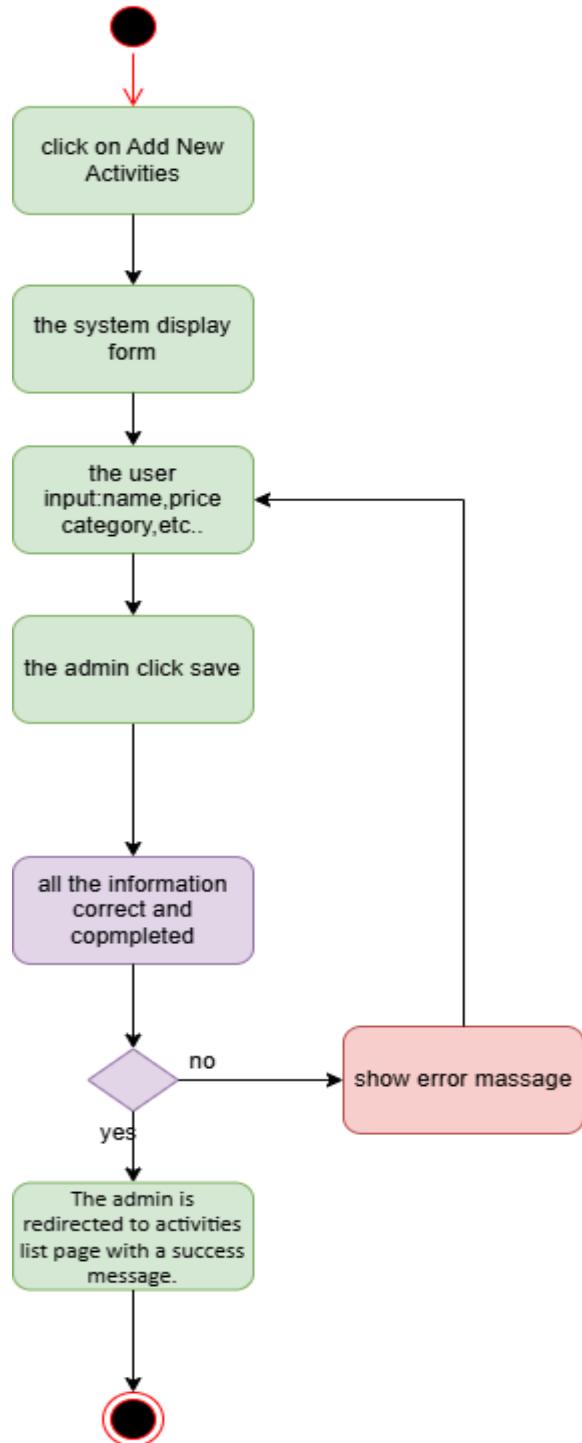


Figure 49Activity Diagram(Add Activity)

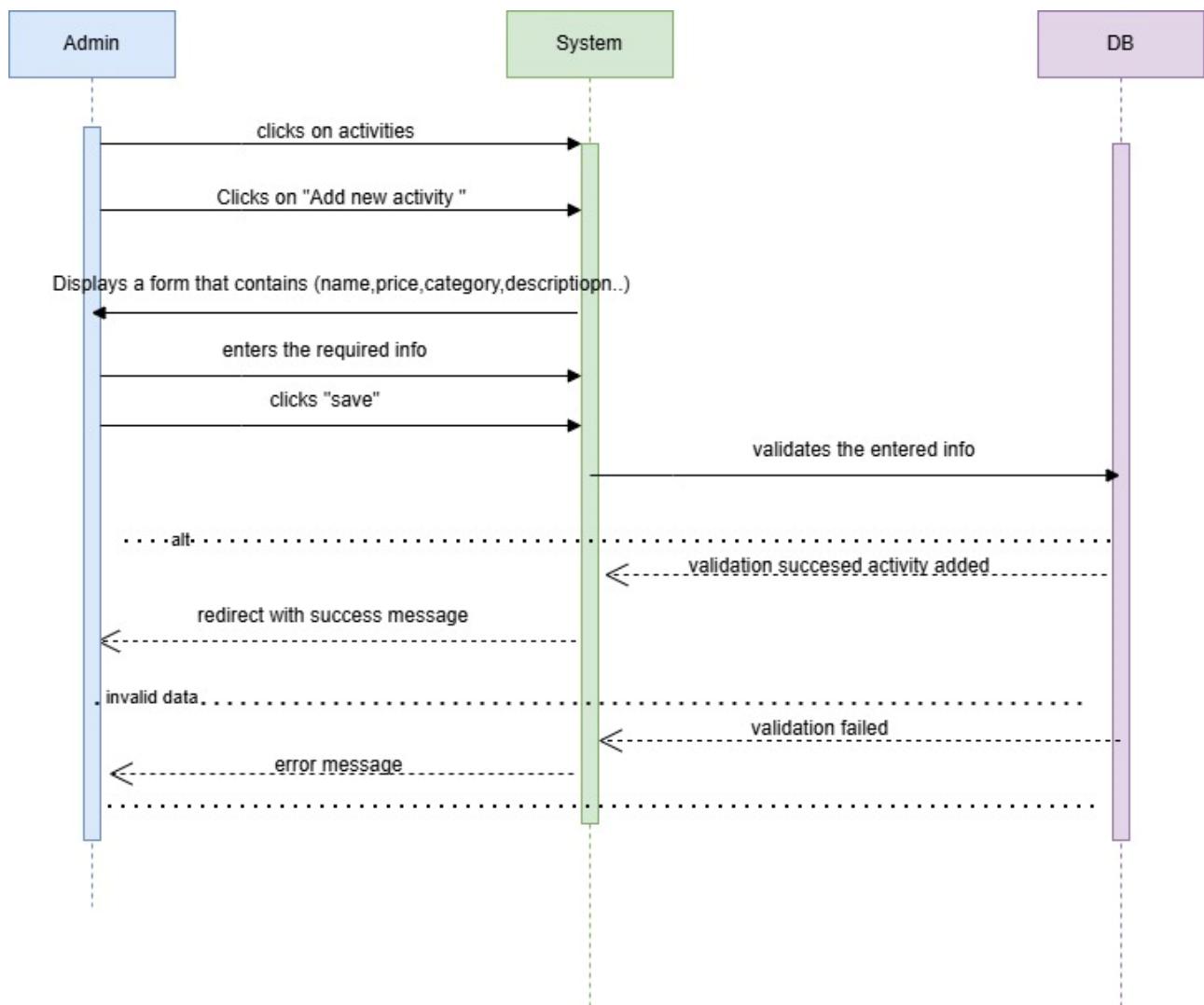


Figure 24 Sequence Diagram(Add Activity)

## ● Edit Activity:

<b>ID</b>	<b>UC-22</b>
<b>Use Case</b>	<b>Edit activity</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The admin is logged into the system.
<b>Post Condition</b>	Activity information is updated. The admin on the Activity list.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Edit for the desired activity.</li> <li>2. The system displays a form containing the current activity details.</li> <li>3. The admin updates the information.</li> <li>4. The admin clicks Save.</li> <li>5. The system validates the updated information.</li> <li>6. The system updates the activity data.</li> <li>7. The admin is redirected to activities list page with a success message.</li> </ol>
<b>Alternative Scenario</b>	If information is incomplete or invalid in step 6, the system returns an error message.

Table 25 Edit Activity

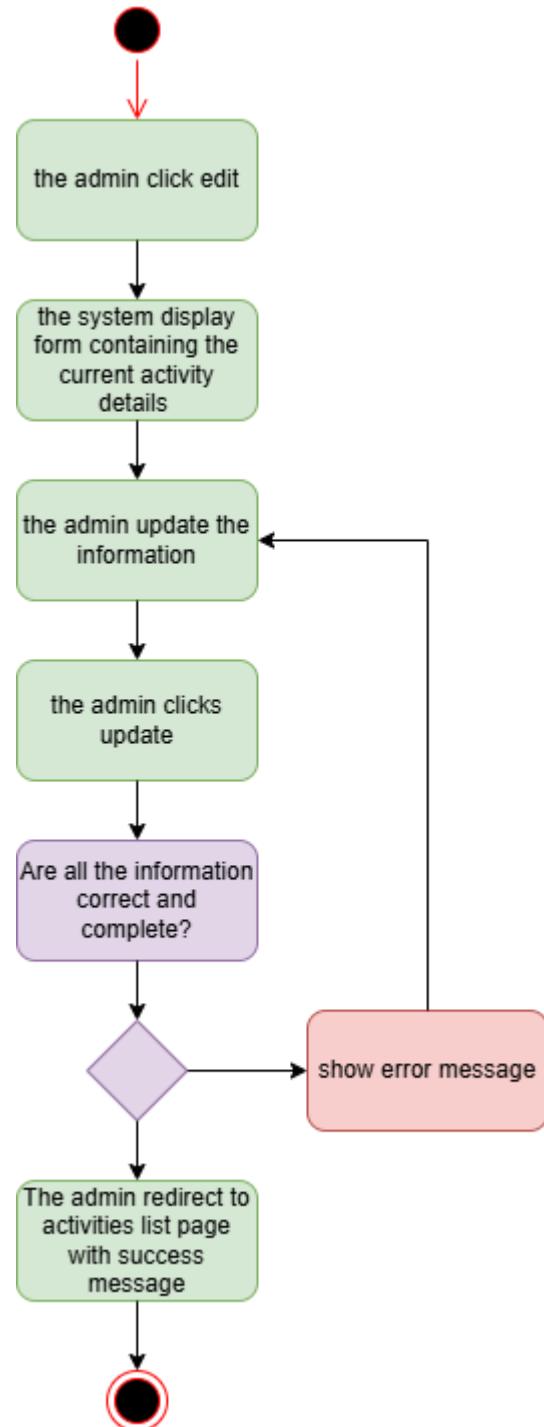


Figure 5.1 Activity Diagram(Edit Activity)

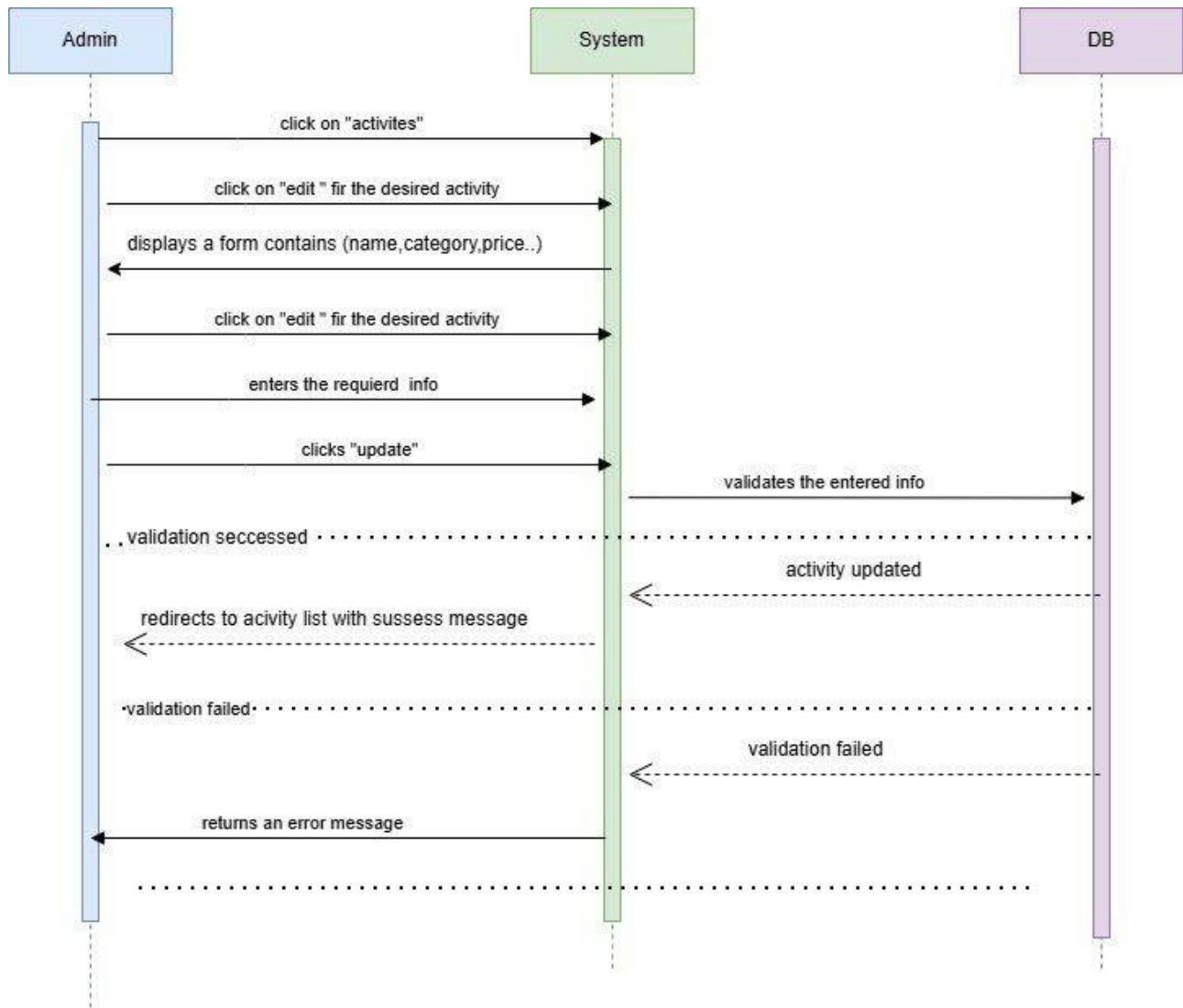


Figure 01 Sequence Diagram(Edit Activity)

## ● Delete Activity:

<b>ID</b>	UC-23
<b>Use Case</b>	Delete activity
<b>Actor</b>	Admin
<b>Pre Condition</b>	The admin is logged into the system.
<b>Post Condition</b>	The selected activity is deleted. The admin on the Activity list.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Delete for the desired activity.</li> <li>2. The system deletes the activity.</li> <li>3. The system displays a success message.</li> </ol>

Table 26 Delete Activity

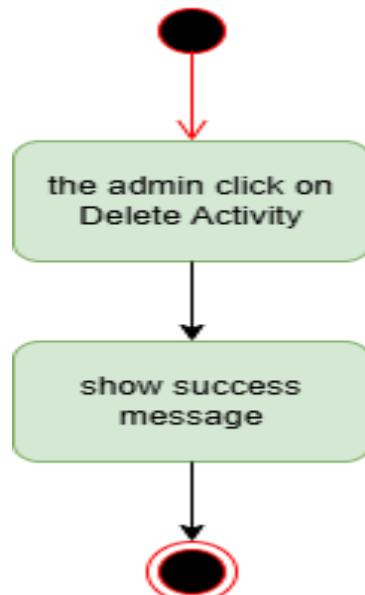


Figure 53 Activity Diagram(Delete Activity)

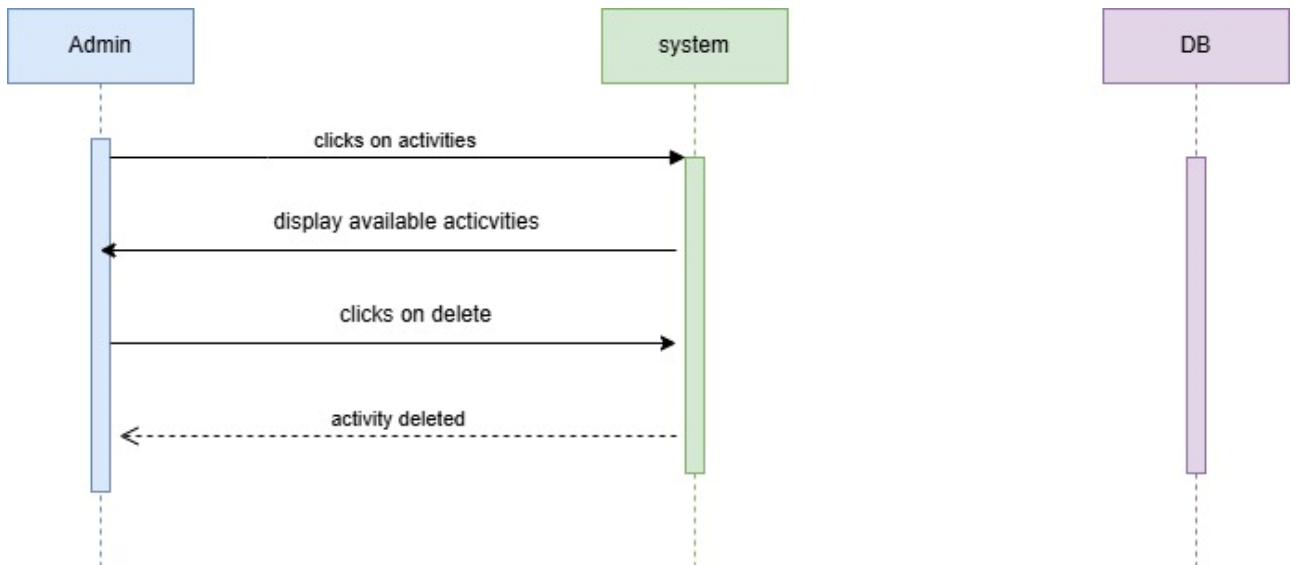


Figure 54 Sequence Diagram(Delete Activity)

## ● View Activities:

<b>ID</b>	<b>UC-24</b>
<b>Use Case</b>	<b>View Activities</b>
<b>Actor</b>	<b>Traveler/Admin</b>
<b>Pre Condition</b>	The actor is logged into the system.
<b>Post Condition</b>	Activities are displayed.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The actor clicks Activities from the navigation bar.</li> <li>2. The system displays available activities.</li> </ol>
<b>Alternative Scenario</b>	If no activities exist, the system displays "No activities found".

Table 27 View Activities

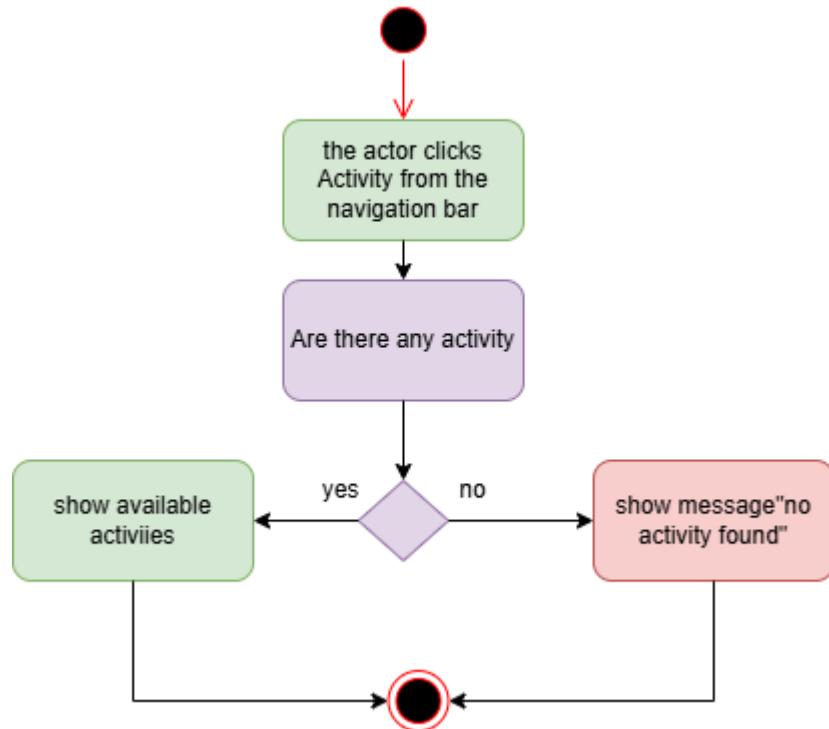


Figure 55 Activity Diagram(View Activities)

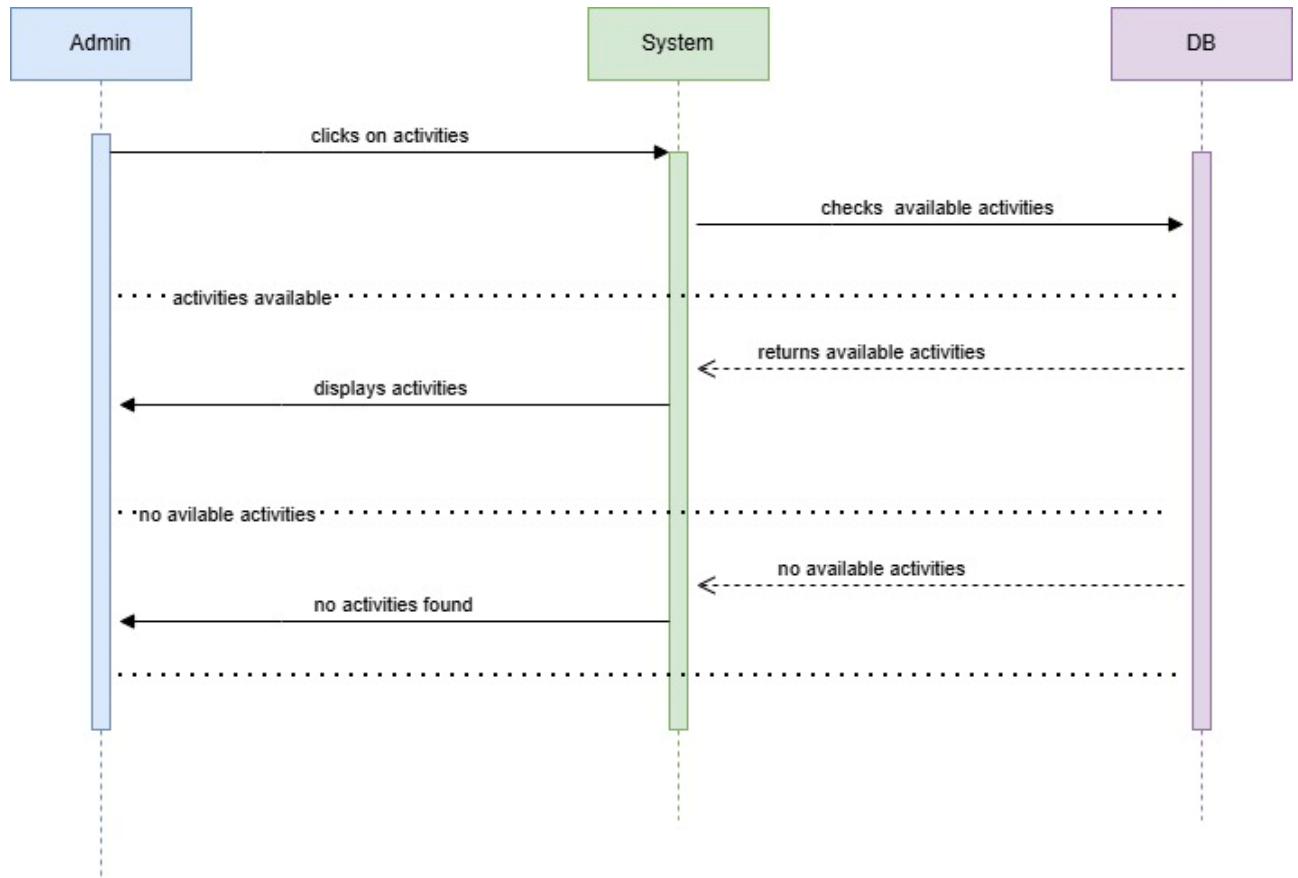


Figure 56 Sequence Diagram (View Activities)

## ● View Activity Details:

<b>ID</b>	UC-25
<b>Use Case</b>	<b>View Activity Details</b>
<b>Actor</b>	<b>Traveler/Admin</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>- The actor is logged into the system.</li> <li>-The actor is on the activities list page.</li> </ul>
<b>Post Condition</b>	.Activity details are displayed
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The actor clicks More Details for the desired activity.</li> <li>2. The system displays a popup showing the activity details.</li> </ol>

Table 28 View Activity Details

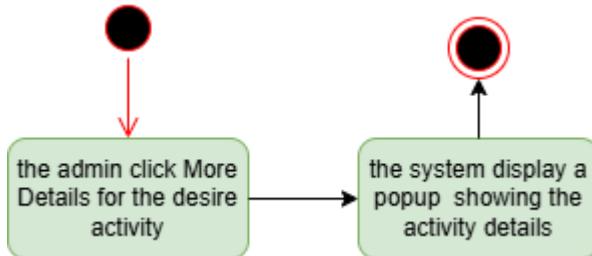


Figure 57 Activity Diagram (View Activity Details)

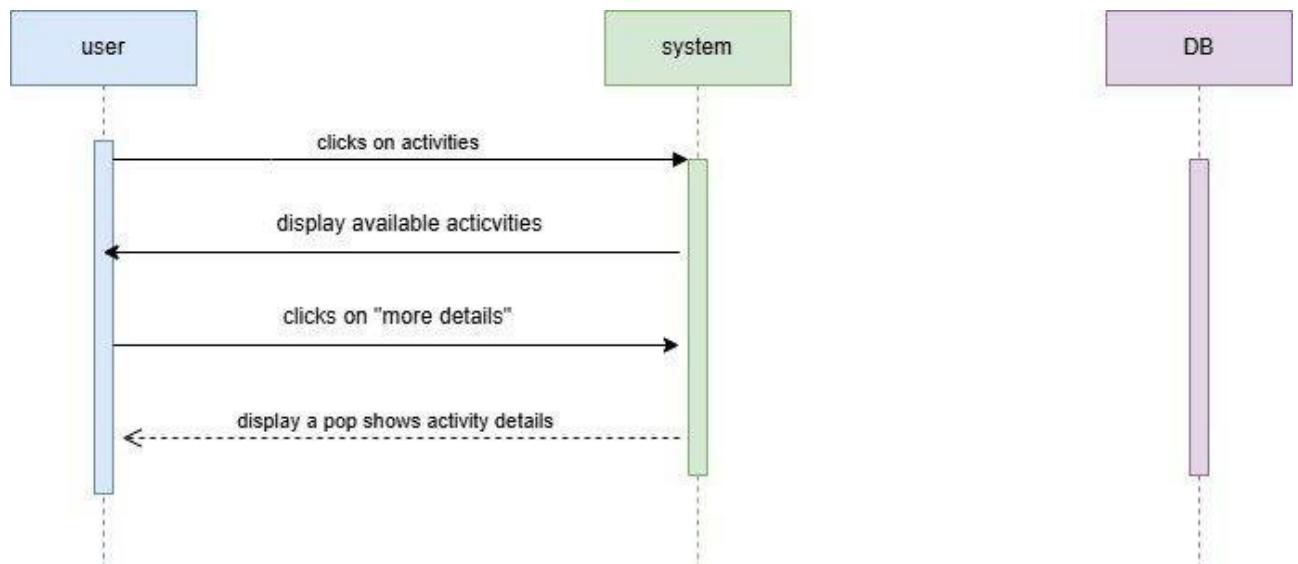


Figure 58 Sequence Diagram (View Activity Details)

## ● Search Activities:

<b>ID</b>	<b>UC-26</b>
<b>Use Case</b>	<b>Search &amp; Filter Activities</b>
<b>Actor</b>	<b>Traveler/Admin</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>- The actor is logged into the system.</li> <li>-The actor is on the activities list page.</li> </ul>
<b>Post Condition</b>	The actor views a list of activities that match the search and filter criteria.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The actor enters a keyword in the <b>Search Bar</b> (Activity Name or Destination).</li> <li>2. The system retrieves activities that match the keyword.</li> <li>3. The actor optionally applies filters from the available filter options: (Availability, Difficulty Level, Guide Language, Requires Booking, Family Friendly, Pets Allowed, Category).</li> <li>4. The system applies the selected filters to the search results.</li> <li>5. The system displays the filtered list of activities.</li> </ol>
<b>Alternative Scenario</b>	If no activities match the search and filter criteria, the system displays " <b>No activities found.</b> "

Table 29 Search & Filter Activities

Yes

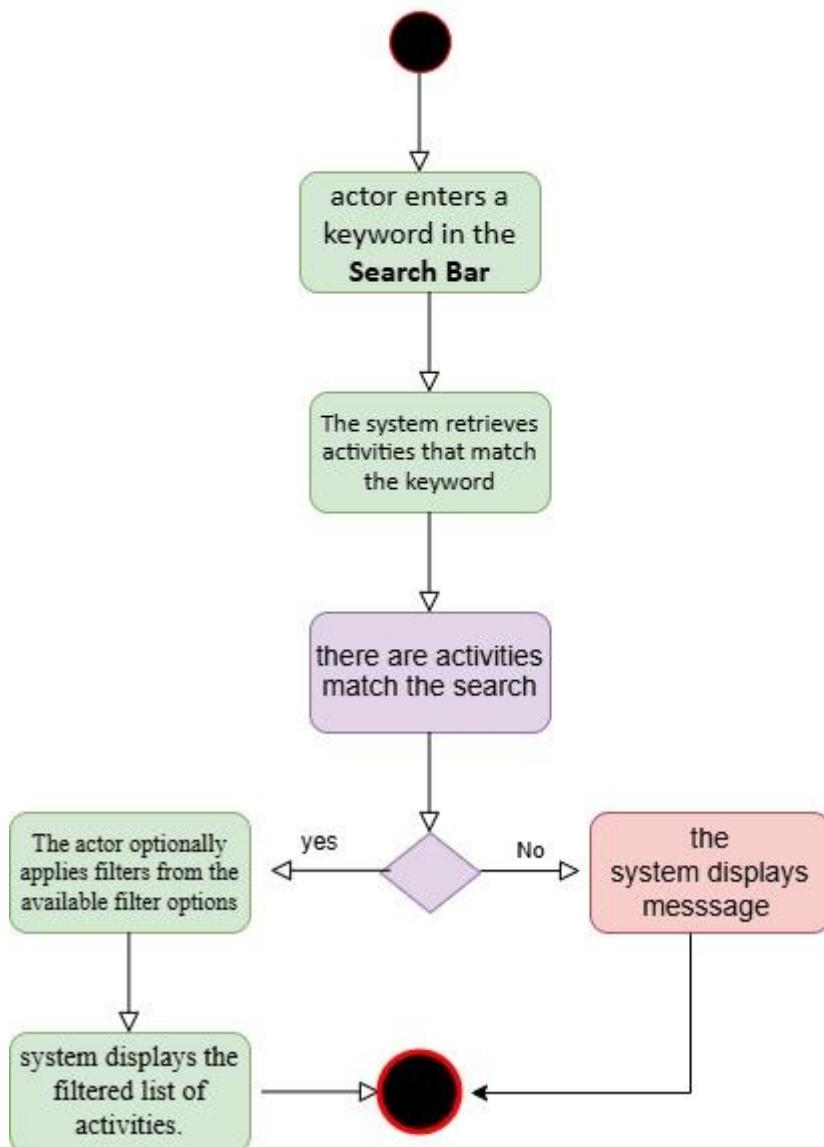


Figure 59 activity diagram (search activites)

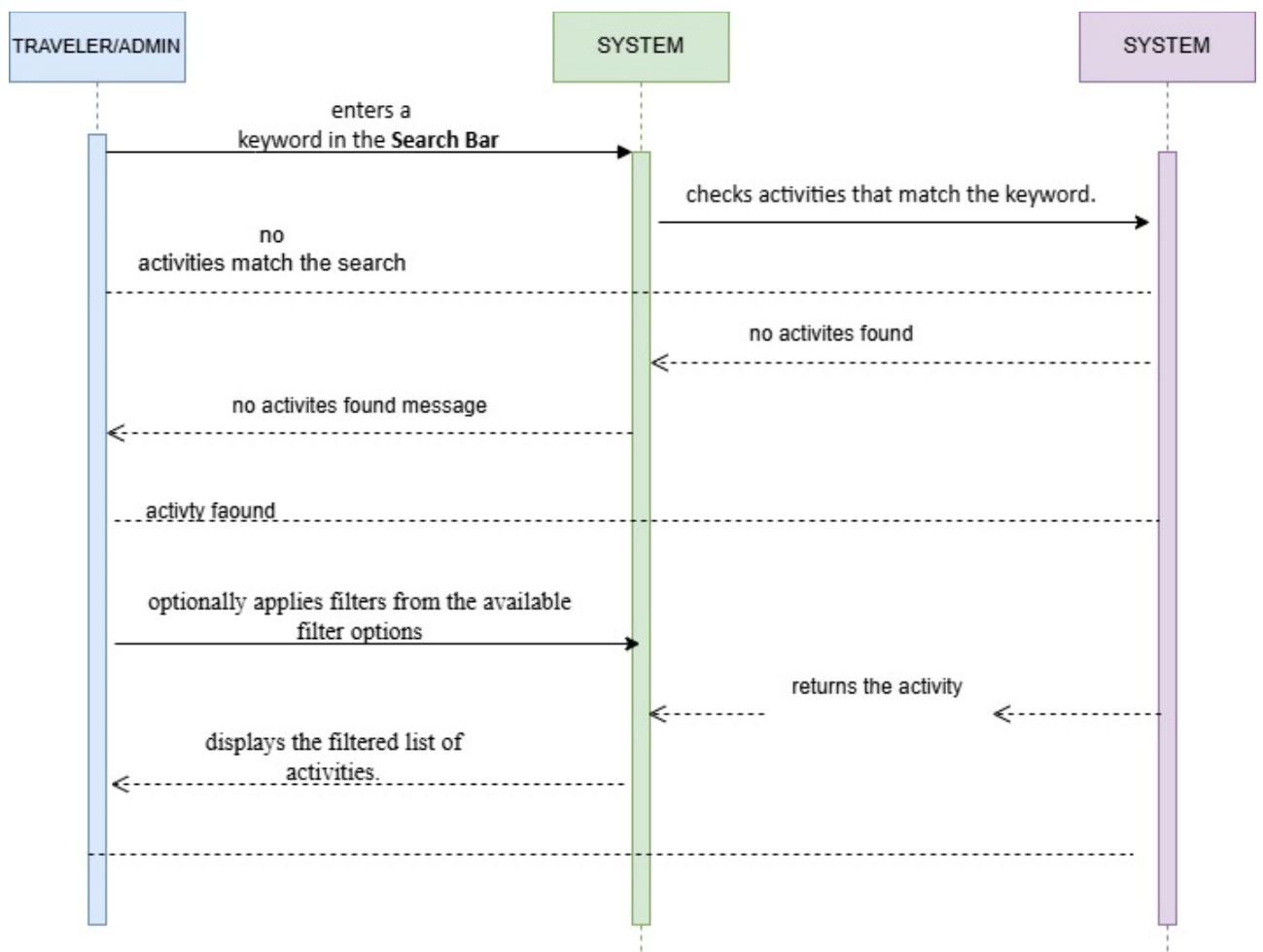


Figure 7. Sequence Diagram(Search Activities)

## ● Add transport service:

<b>ID</b>	<b>UC-27</b>
<b>Use Case</b>	Create transport service:
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The admin is logged into the system. And is on transport services page
<b>Post Condition</b>	A new transport service is added.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Add New Service button.</li> <li>2. The system displays a popup containing: (Service Name, Service Type, Description, Service Image).</li> <li>3. The admin enters the required service information.</li> <li>4. The admin clicks Save.</li> <li>5. The system checks if a service with the same name already exists.</li> <li>6. The system validates the information.</li> <li>7. The system adds the new service.</li> <li>8. The admin is redirected to transport services page with a success message.</li> </ol>
<b>Alternative Scenario</b>	<p>If the service name already exists in step 5 the system displays error message.</p> <p>If information is missing in step 6 the system displays error message.</p>

Table 30 Add transport service

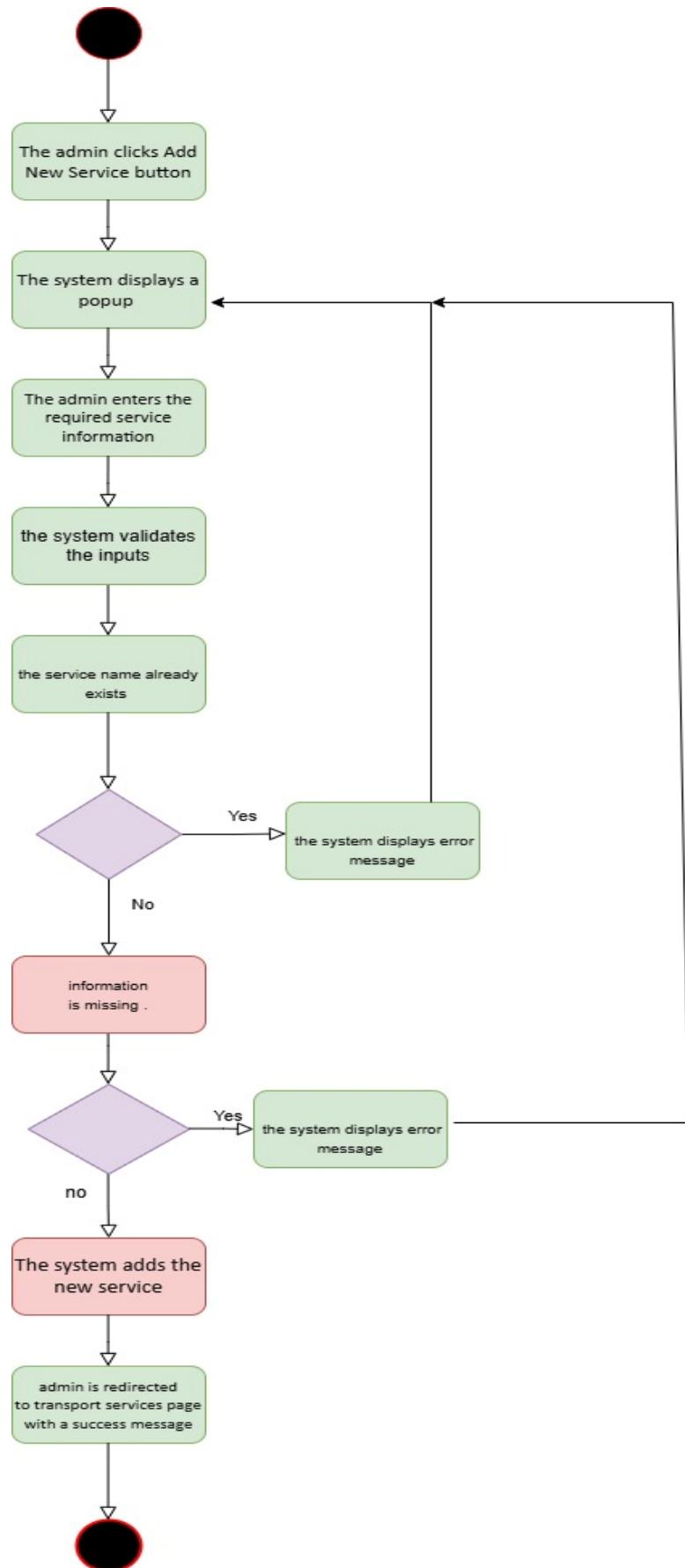


Figure 61 activity diagram (add transport service)

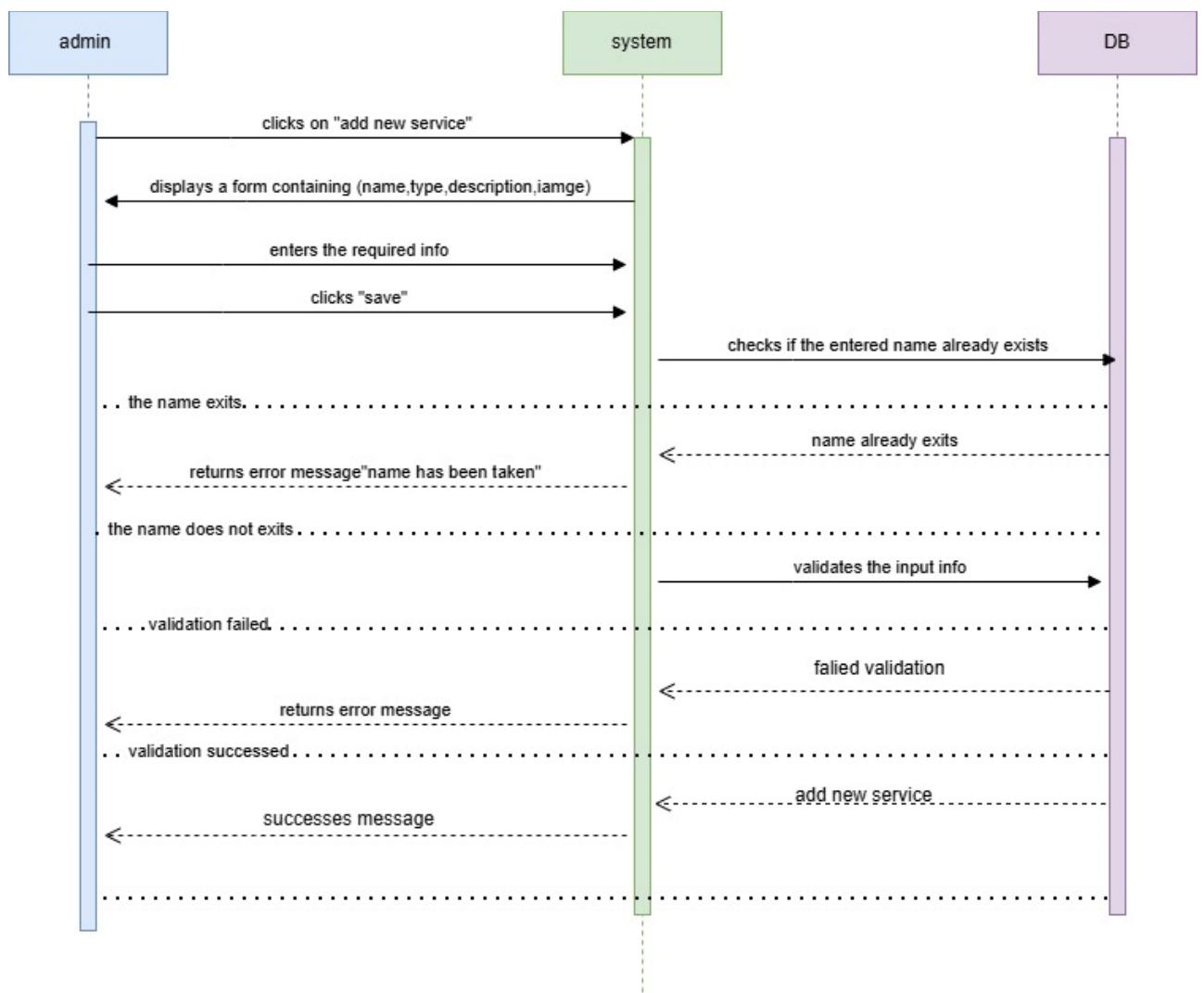


Figure 11 Sequence Diagram( Add transport service)

## ● Edit transport service:

<b>ID</b>	<b>UC-28</b>
<b>Use Case</b>	Edit transport service
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	.Admin is on the Transport Services list page
<b>Post Condition</b>	.Service information is updated
<b>Main Scenario</b>	<p>1. The admin clicks Edit for the selected service.</p> <p>2. The system displays a form with the current service details.</p> <p>3. The admin modifies the information.</p> <p>4. The admin clicks Save.</p> <p>5. The system validates the updated information.</p> <p>6. The system updates the service.</p> <p>7. The admin is redirected to the service list with a success message.</p>
<b>Alternative Scenario</b>	If information is invalid, the system returns an error message

Table 31 Edit transport service

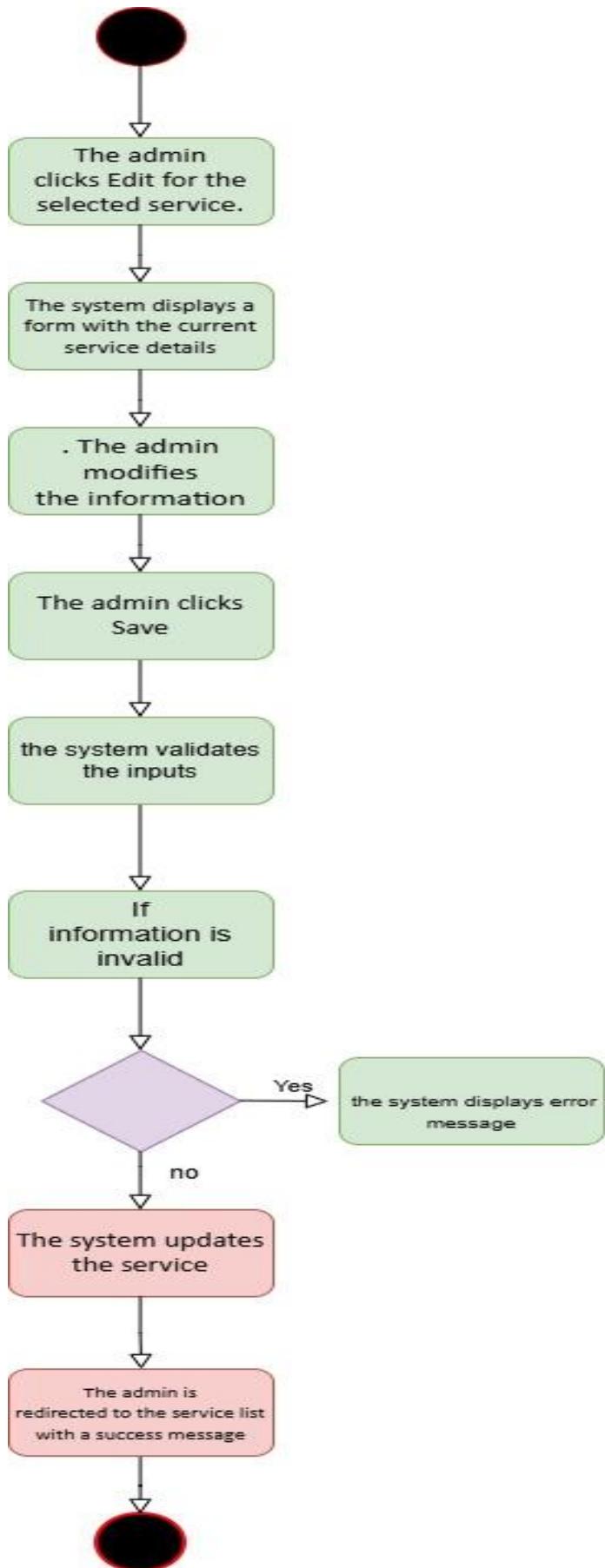


Figure 63 avctivity digram (edit transport service)

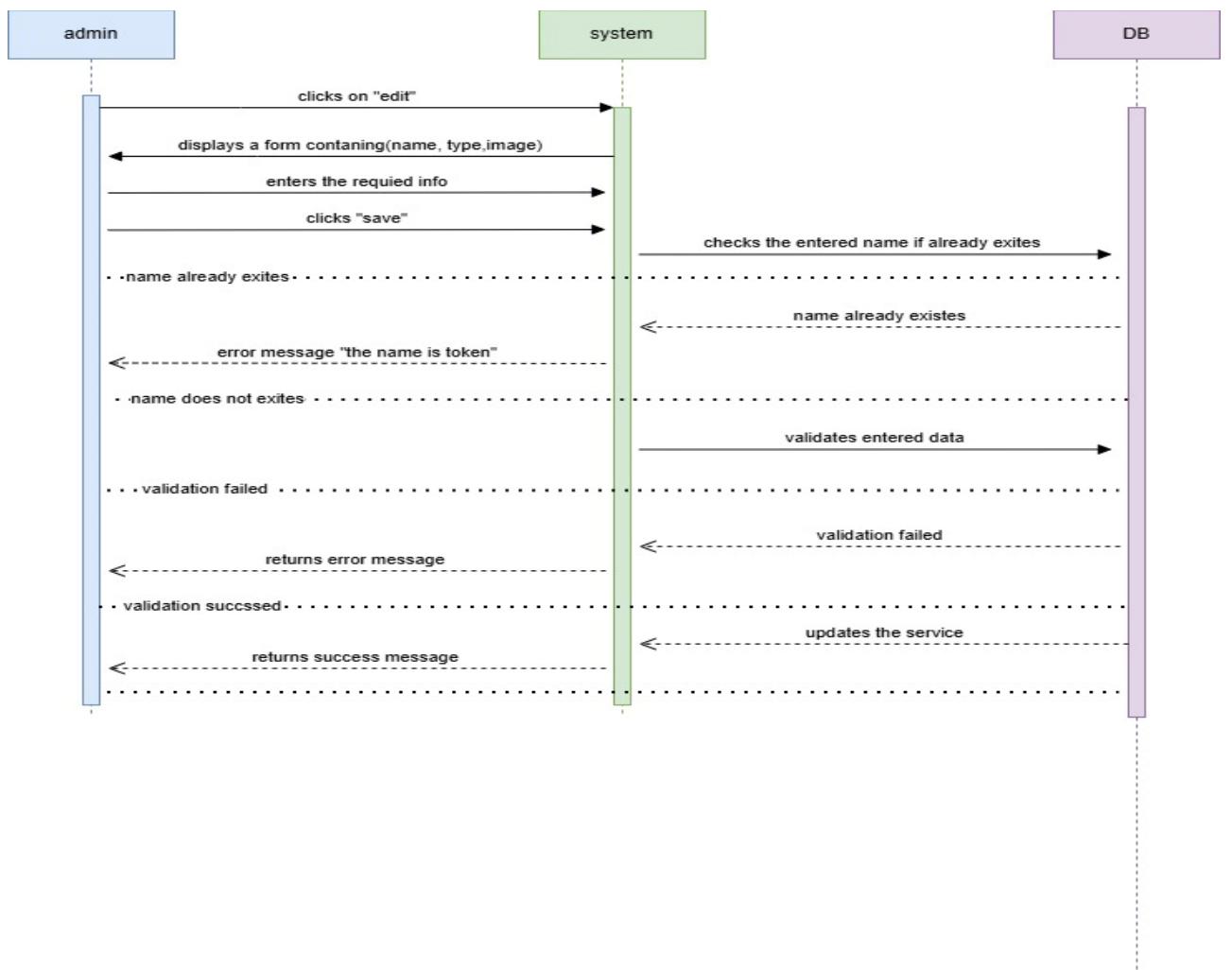


Figure 64 Sequence Diagram (Edit transport service)

## ● Delete transport service:

<b>ID</b>	<b>UC-29</b>
<b>Use Case</b>	Delete transport service
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	Admin is on the Transport Services list page
<b>Post Condition</b>	Transport service is deleted.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Delete.</li> <li>2. The system checks if there are reserved vehicles linked to the service.</li> <li>3. The system deletes the service.</li> <li>4. The system displays a success message.</li> </ol>
<b>Alternative Scenario</b>	If linked reserved vehicles exist → deletion is blocked and an error message is shown.

*Table 32 Delete transport service*

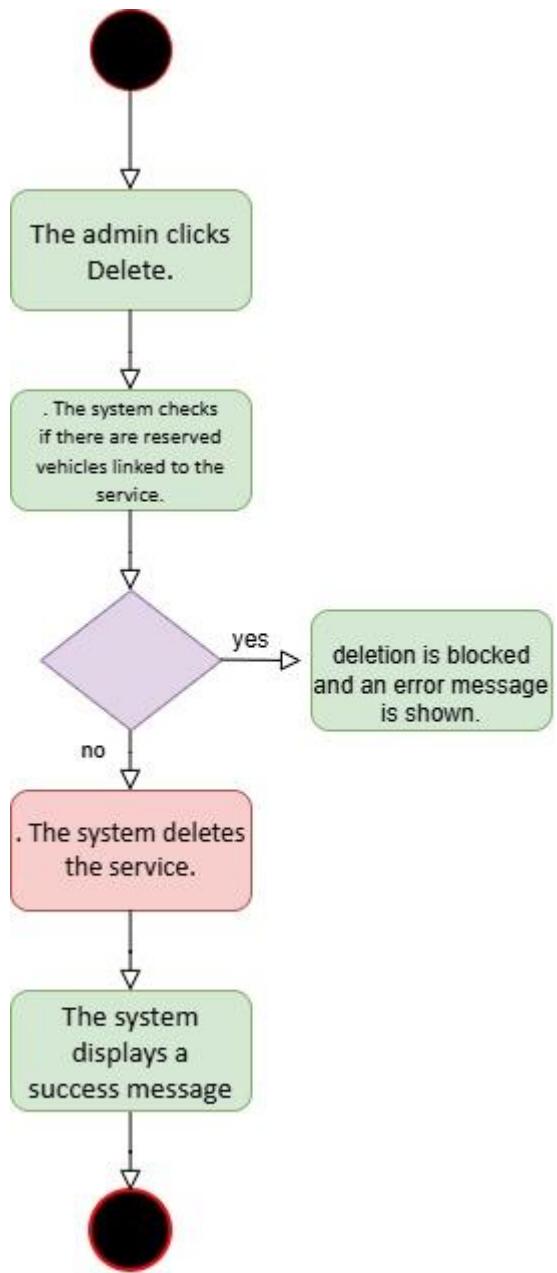


Figure 70 activity diagram (delete transport service)

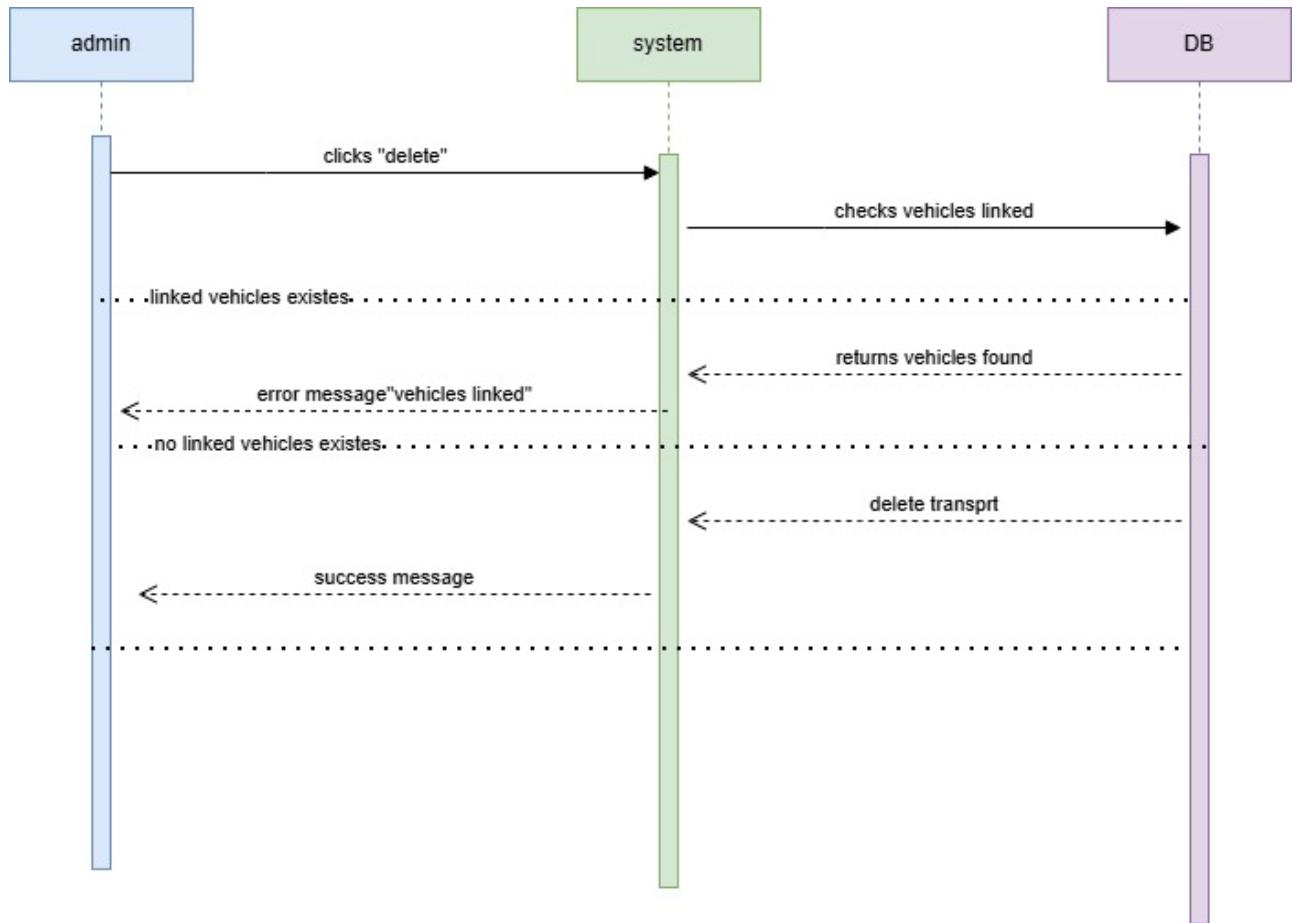


Figure 77 Sequence Diagram (Delete transport service:)

## ● View transport services:

<b>ID</b>	UC-30
<b>Use Case</b>	<b>View transport services</b>
<b>Actor</b>	<b>Traveler/Admin</b>
<b>Pre Condition</b>	The actor is logged into the system.
<b>Post Condition</b>	Transport services are displayed.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The actor clicks Transport from the navigation bar.</li> <li>2. The system displays available transport services.</li> </ol>
<b>Alternative Scenario</b>	If no transport services exist the system displays "No ."transport found

Table 33 View transport services

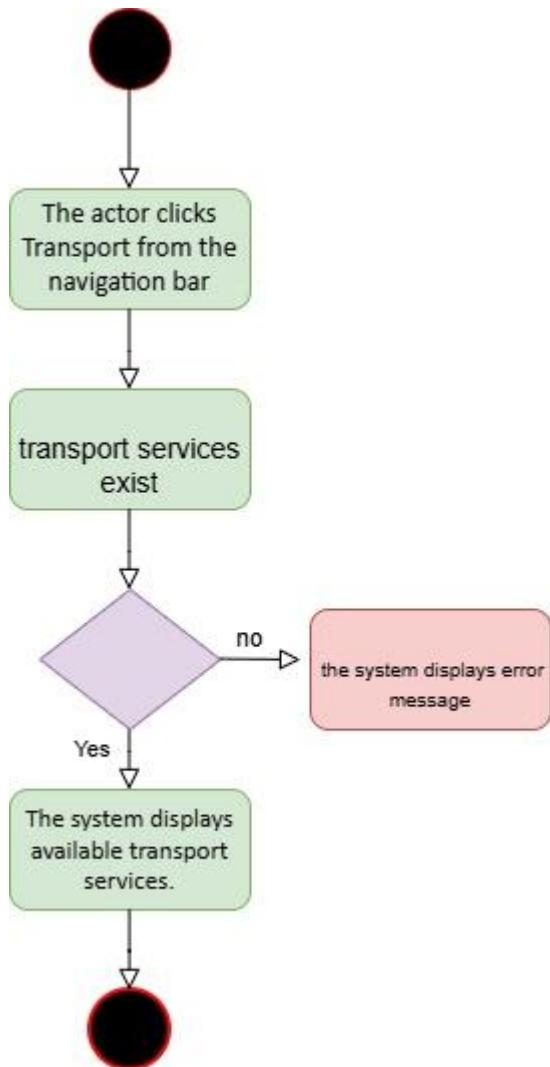


Figure 67 activity diagram (view transport service)

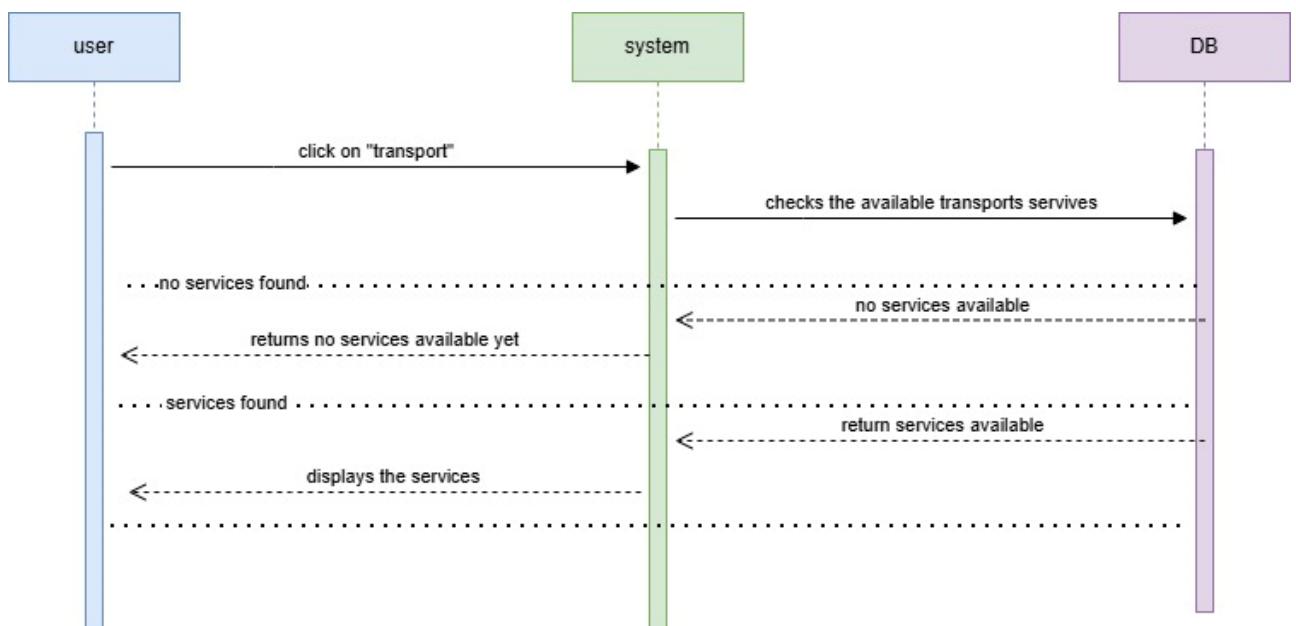


Figure 68 Sequence Diagram (View transport services)

## ● View Transport Linked Vehicles:

<b>ID</b>	<b>UC-31</b>
<b>Use Case</b>	<b>View transport Linked Vehicles</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	Admin is on the Transport list page.
<b>Post Condition</b>	Vehicles for the selected service are displayed.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Show Vehicles for the desired service.</li> <li>2. The system displays all linked vehicles.</li> </ol>
<b>Alternative Scenario</b>	If no vehicles exist the system displays "No vehicles found".

Table 34 View Transport Linked Vehicles

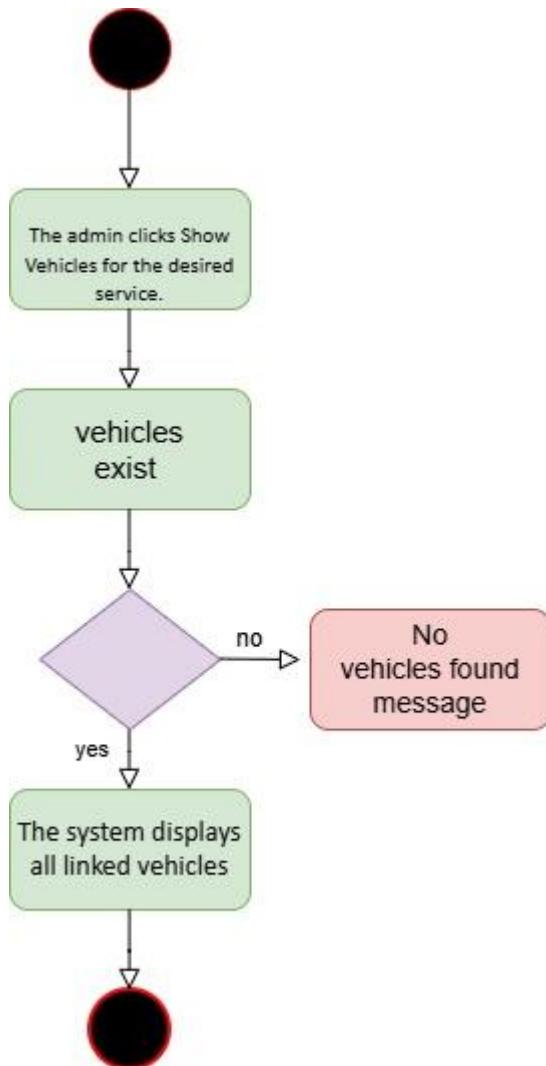


Figure 69 activity diagram (show linked vehicles)

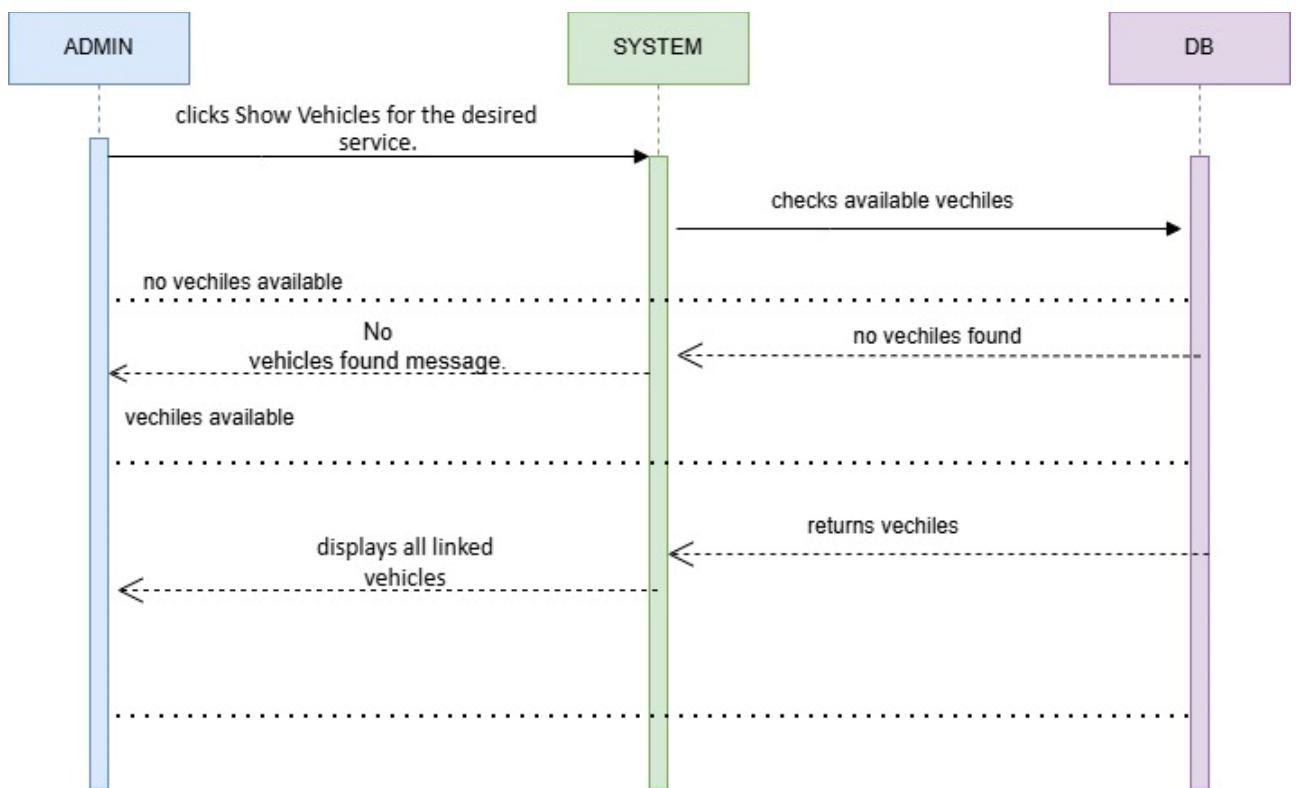


Figure 70 Sequence Diagram(View Transport Linked Vehicles)

## ● Add Vehicle:

<b>ID</b>	<b>UC-32</b>
<b>Use Case</b>	<b>Add Vehicle</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	Admin is on the Transport list page.
<b>Post Condition</b>	A new vehicle is added.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Add Vehicle button.</li> <li>2. The system displays a form containing: (Car Model, Max Passengers, Plate Number, Base Price, Price per KM, Category, Vehicle Image, Driver Name).</li> <li>3. The admin enters the information.</li> <li>4. The admin clicks Save.</li> <li>5. The system validates the information.</li> <li>6. The system redirects the admin to the vehicles list page with a success message.</li> </ol>
<b>Alternative Scenario</b>	If information is missing the system displays error message

Table 35 Add Vehicle

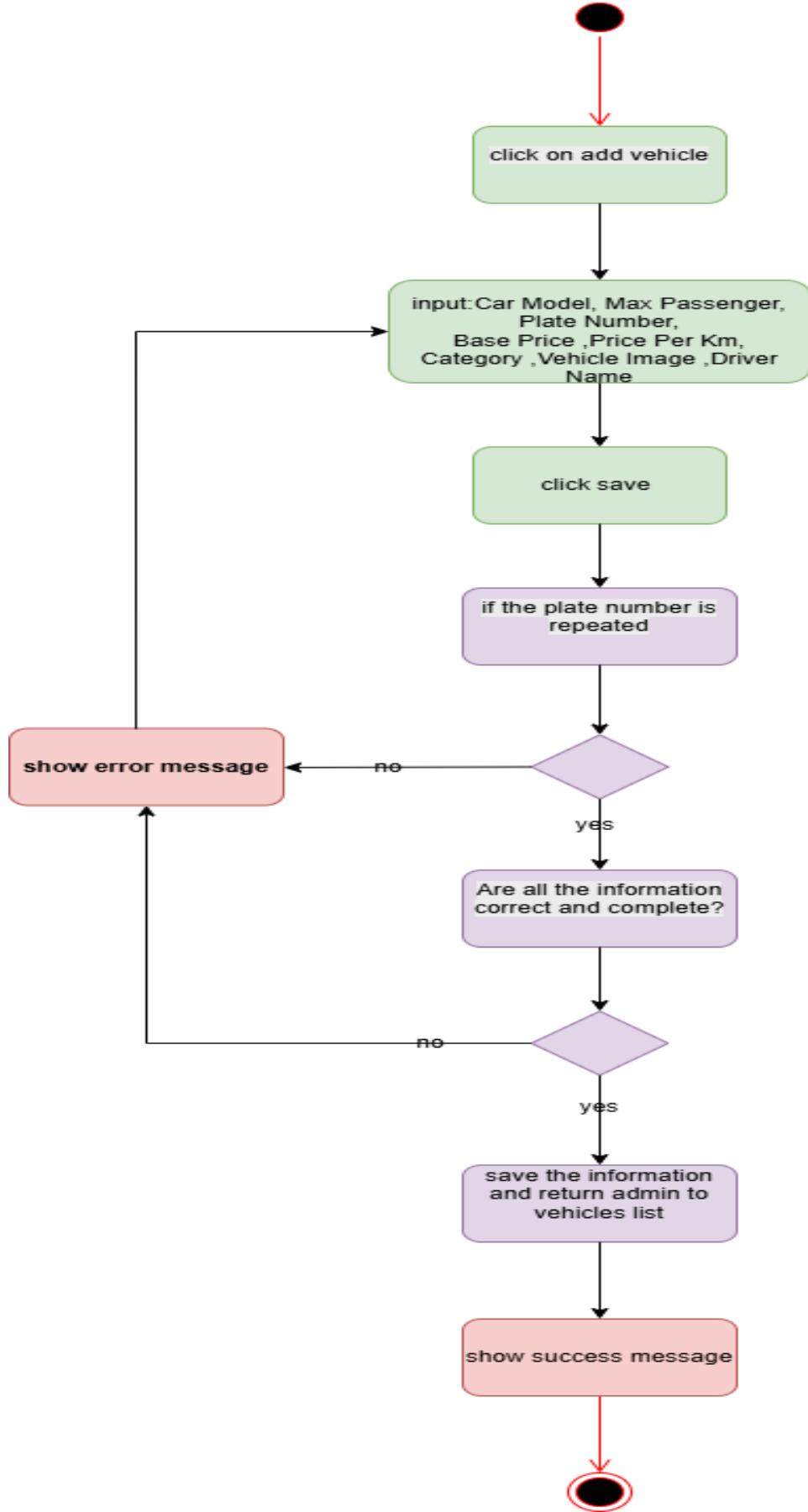


Figure V1 Activity Diagram (Add Vehicle)

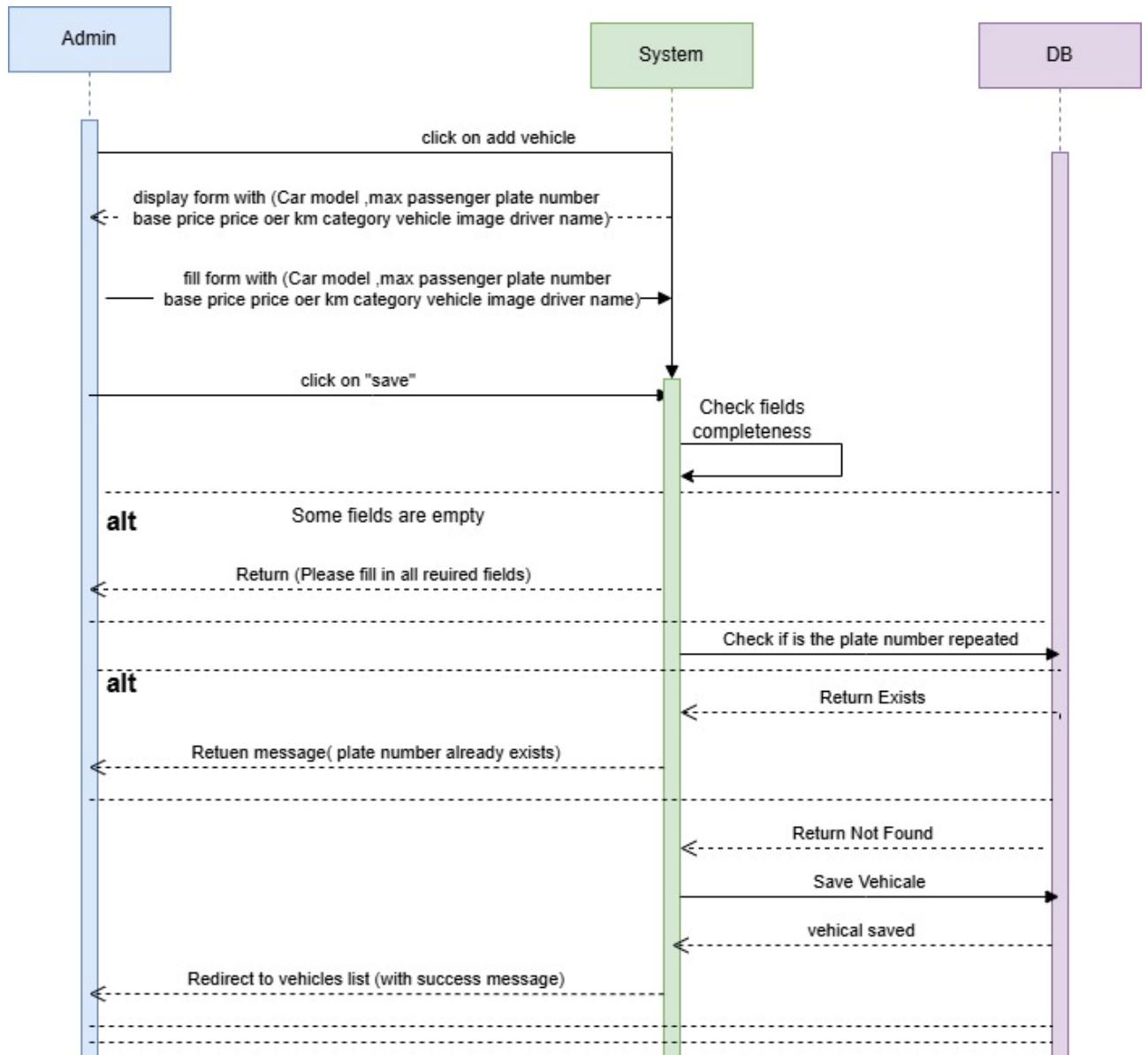


Figure 11 Sequence Diagram(Add Vehicle)

## ● Edit Vehicle:

<b>ID</b>	<b>UC-33</b>
<b>Use Case</b>	Edit Vehicle
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	Admin is on the vehicles list page.
<b>Post Condition</b>	Vehicle information is updated
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Edit for the selected vehicle.</li> <li>2. The system displays a form with the current vehicle details.</li> <li>3. The admin modifies the information.</li> <li>4. The admin clicks Save.</li> <li>5. The system validates the updated information.</li> <li>6. The system updates the Vehicle.</li> <li>7. The admin is redirected to the Vehicle list with a success message.</li> </ol>
<b>Alternative Scenario</b>	If information is invalid, the system returns an error message.

Table 36 Edit Vehicle

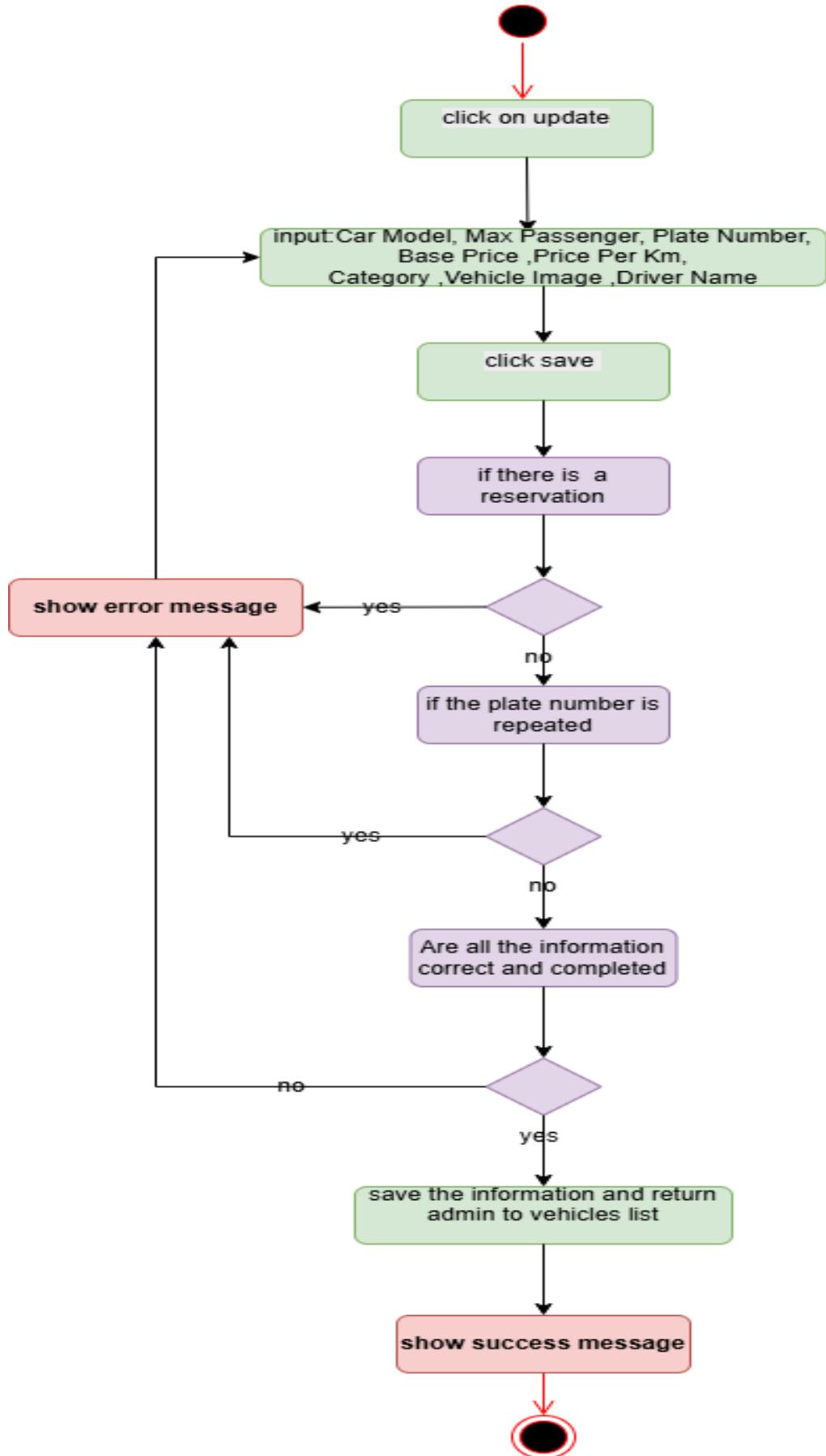


Figure 11 Activity Diagram(Edit Vehicle)

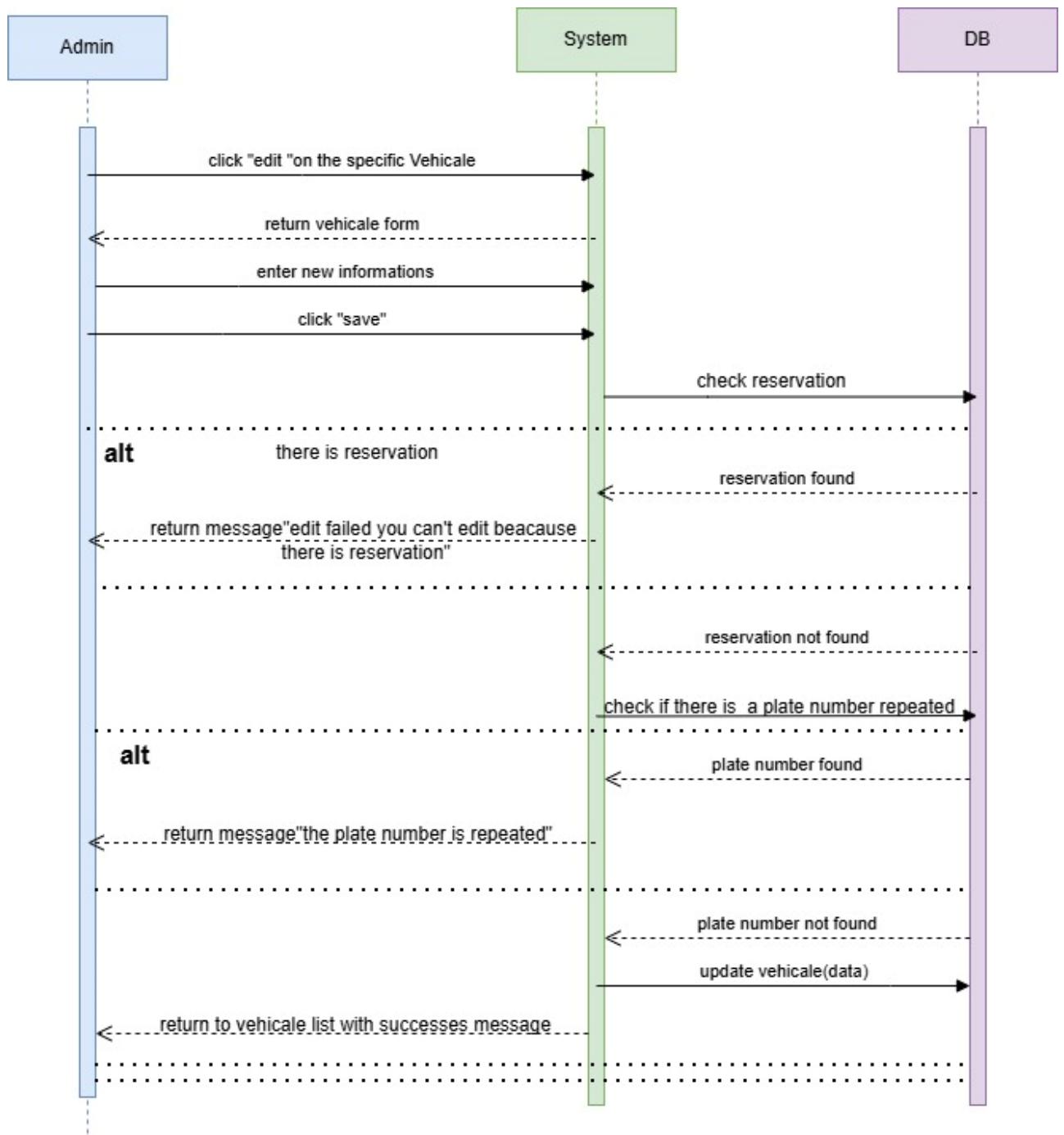


Figure 12 Sequence Diagram(Edit Vehicle)

## ● Delete Vehicle:

<b>ID</b>	<b>UC-34</b>
<b>Use Case</b>	<b>Delete Vehicle</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	Admin is on the vehicles list page.
<b>Post Condition</b>	Vehicle is deleted.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks Delete.</li> <li>2. The system checks if there are coming reservations linked to the vehicle.</li> <li>3. The system deletes the vehicle.</li> <li>4. The system displays a success message.</li> </ol>
<b>Alternative Scenario</b>	If linked coming reservations exist → deletion is blocked and an error message is shown.

*Table 37 Delete Vehicle*

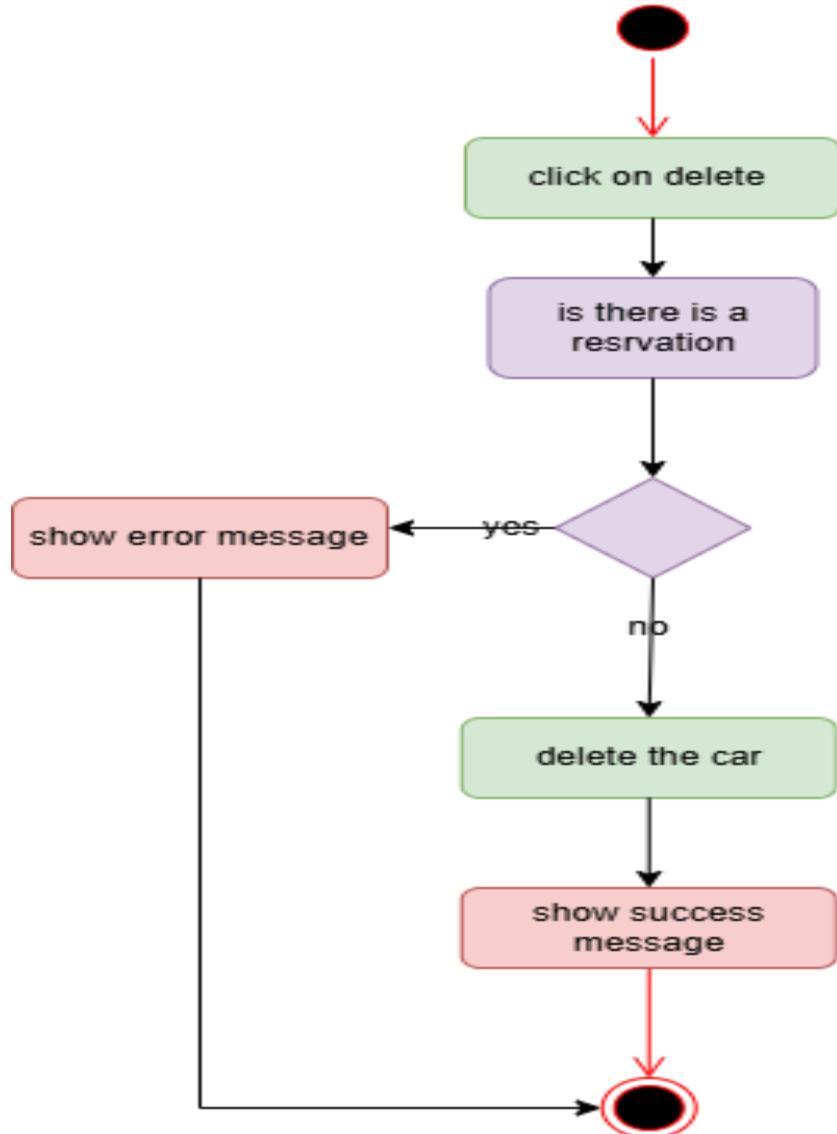


Figure 10 Activity Diagram(Delete Vehicle)

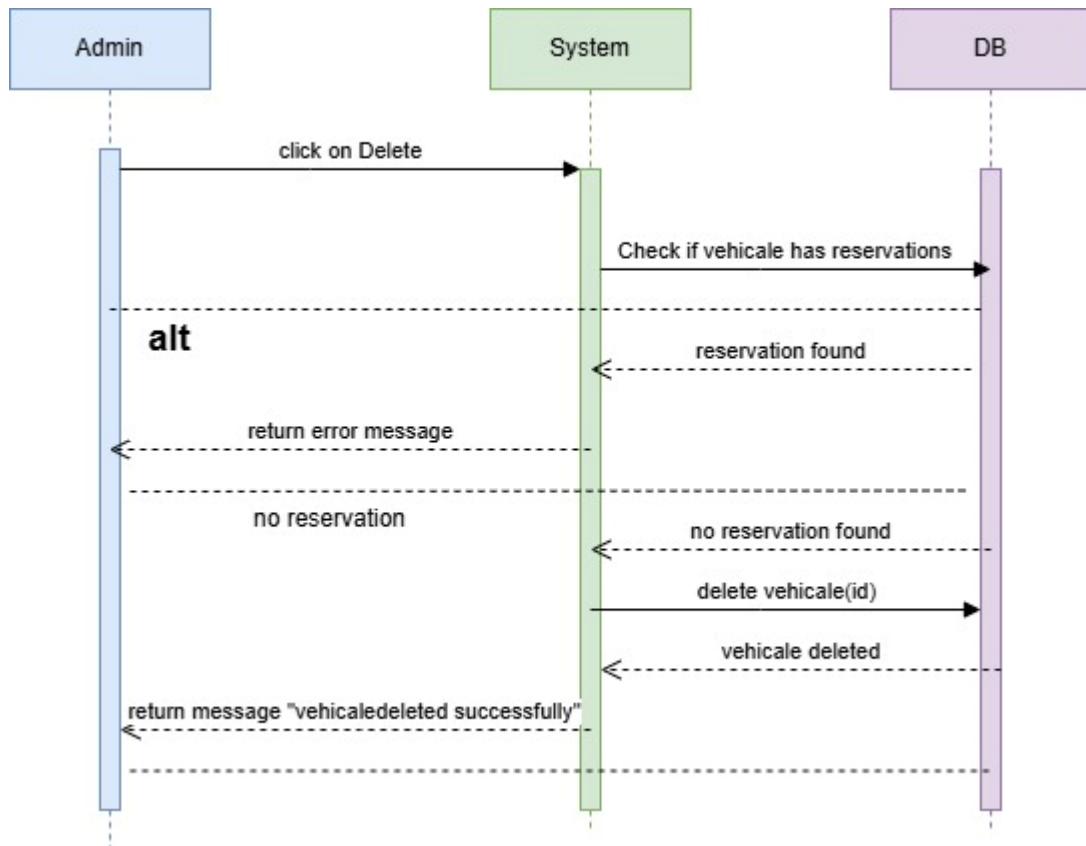


Figure 11 Sequence Diagram(Delete Vehicle)

## ● Search Available Vehicles:

<b>ID</b>	<b>UC-35</b>
<b>Use Case</b>	Search Available Vehicles
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The traveler is logged into the system and is on the Transport Services page.
<b>Post Condition</b>	Available vehicles for the selected transport service are displayed.
<b>Main Scenario</b>	<p>1. The traveler clicks the <b>Order Car</b> button for a selected transport service.</p> <p>2. The system displays a form containing the following fields: (Pickup Location, Drop-off Location, Pickup Date and Time, Number of Passengers).</p> <p>3. The traveler enters the required trip information.</p> <p>4. The traveler clicks <b>Find Available Cars</b>.</p> <p>5. The system validates the entered information.</p> <p>6. The system searches for available vehicles based on the specified information.</p> <p>7. The system calculates the trip details including route, distance, and estimated duration.</p> <p>8. The system displays the <b>Available Vehicles</b> page showing:</p> <ul style="list-style-type: none"> <li>• Trip overview (Pickup Location, Destination, Date and Time, Number of Passengers).</li> <li>• Route map.</li> <li>• Distance and estimated trip duration.</li> <li>• List of available vehicles with details: (Vehicle Name/Model, Category, Maximum Passengers, Plate Number, Driver Name, Driver Phone, Base Price, Price per Kilometer).</li> </ul>
<b>Alternative Scenario</b>	<p>If required information is missing or invalid in step 6, the system displays an error message and returns the traveler to the form.</p> <p>If no available vehicles are found in step 7, the system displays a message indicating <b>“No Available Vehicles Found”</b>.</p>

Table 38 Search Available Vehicles

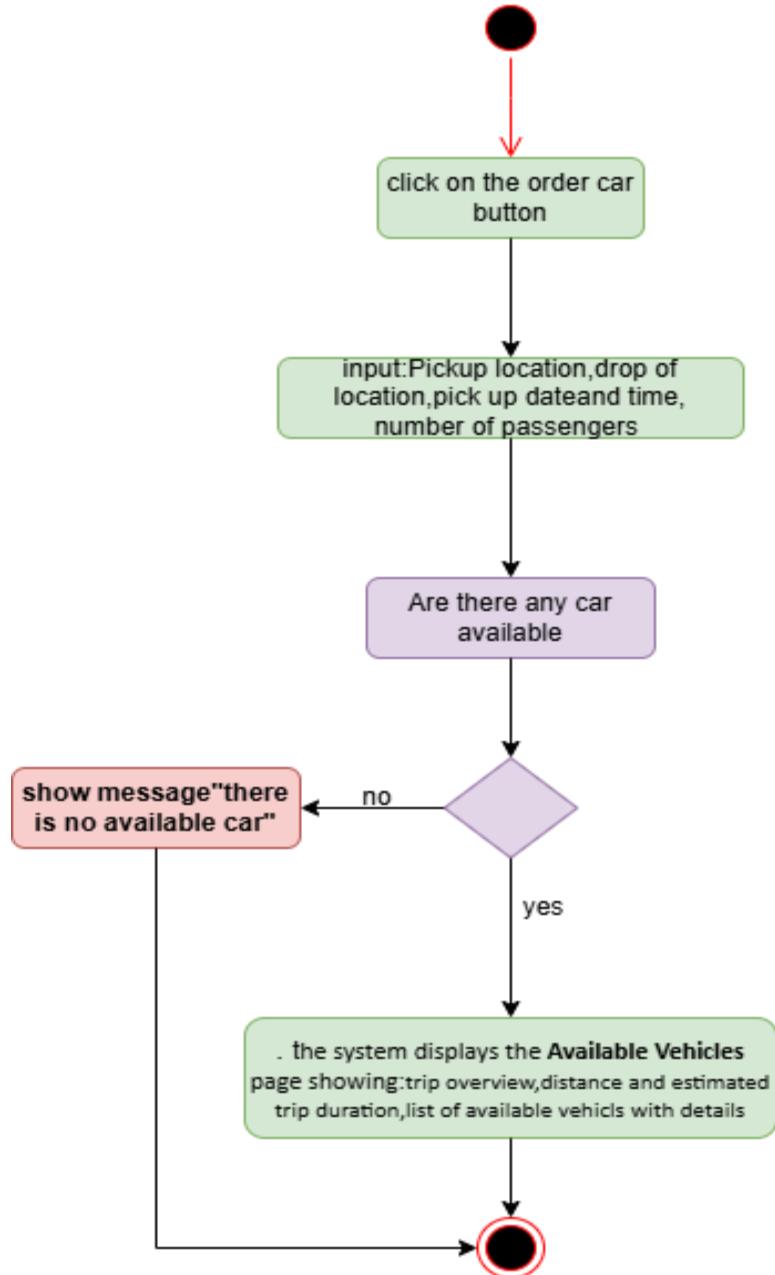


Figure 77 Activity Diagram (Search Available Vehicles)

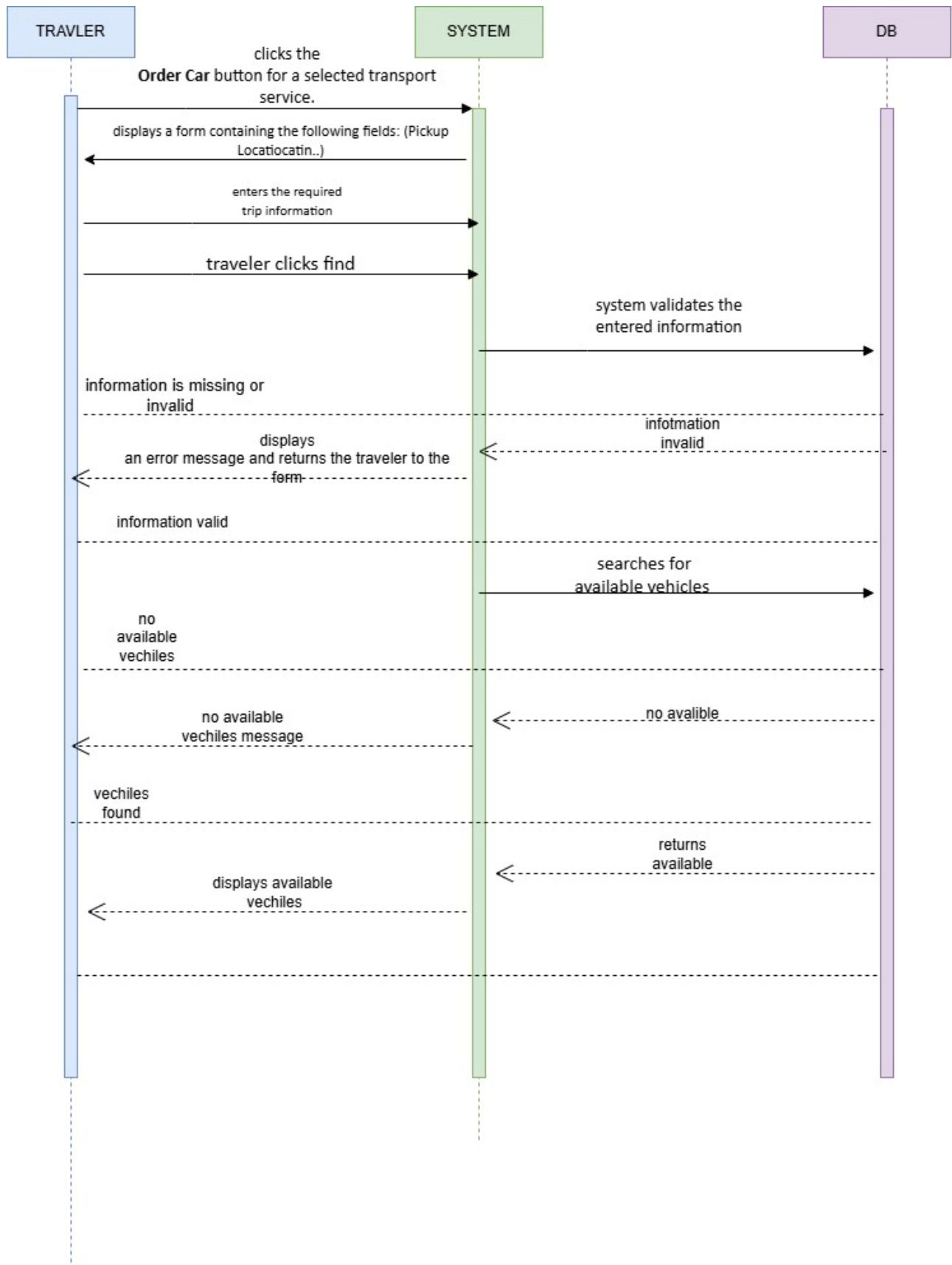


Figure 14 Sequence Diagram(Search Available Vehicles)

## ● Book Vehicle:

<b>ID</b>	<b>UC-36</b>
<b>Use Case</b>	Reserve Vehicle
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The traveler is logged into the system. The traveler is on the <b>Available Vehicles</b> page.
<b>Post Condition</b>	A vehicle reservation is successfully created and paid.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The traveler clicks <b>Reserve</b> for a selected vehicle.</li> <li>2. The system displays a <b>Contact Information</b> form containing calculated Total Price and the following fields: (First Name, Last Name, Phone Number, Email).</li> <li>3. The system automatically fills the form fields using the traveler's stored information.</li> <li>4. The traveler reviews or updates the contact information.</li> <li>5. The traveler clicks <b>Continue to Payment</b>.</li> <li>6. The system validates the entered contact information.</li> <li>7. The system redirects the traveler to the <b>Pay with PayPal</b> payment page.</li> <li>8. The traveler completes the payment process.</li> <li>9. The system creates the vehicle reservation.</li> <li>10. The system redirects the traveler to the Transport Services page with a success message.</li> </ol>
<b>Alternative Scenario</b>	If contact information is missing or invalid, the system displays an error message and returns the traveler to the form. If the payment fails, the system returns the traveler to the contact form and displays a payment error message.

Table 39 Reserve Vehicle

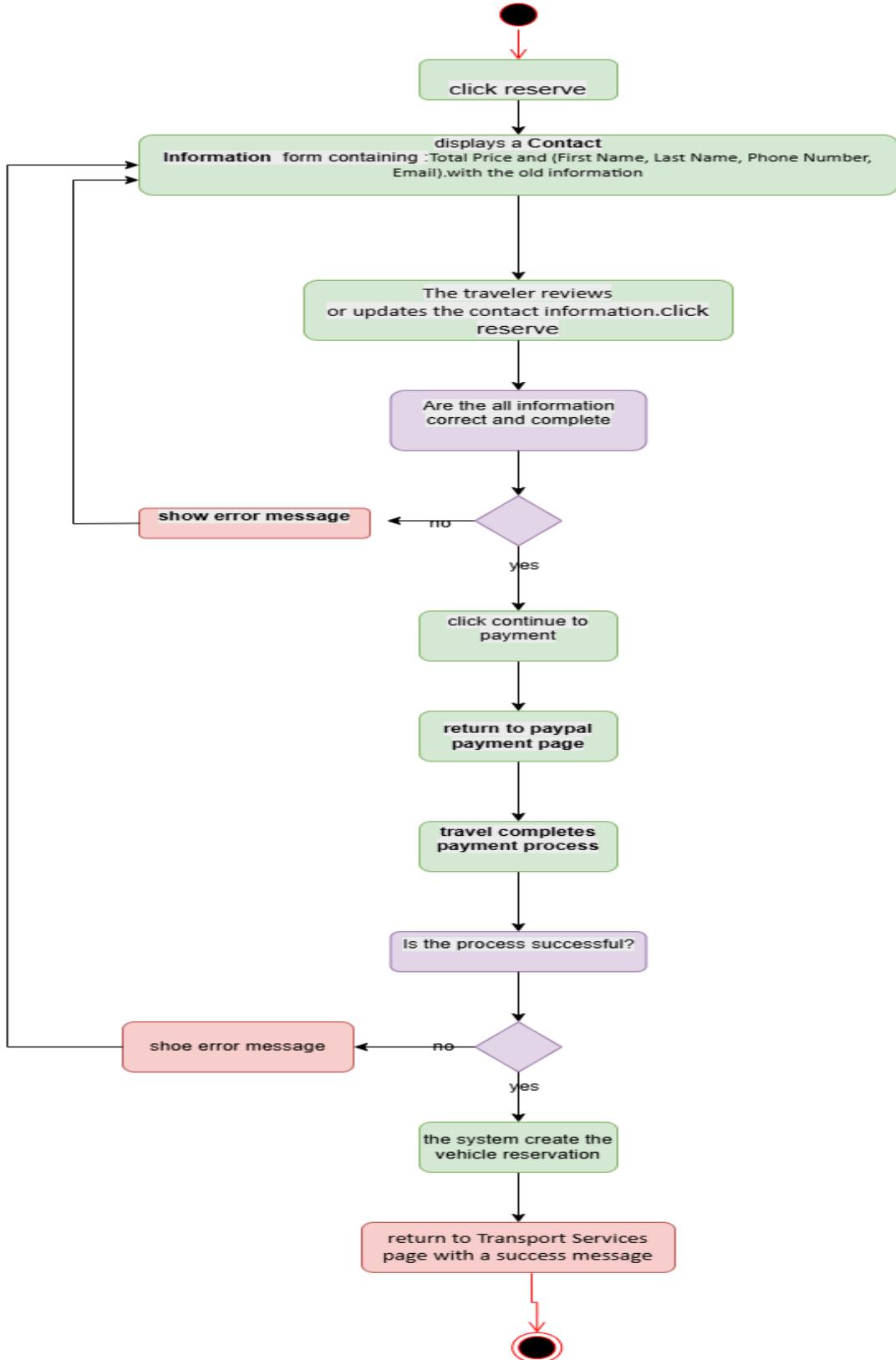


Figure 79 Activity Diagram (Book Vehicle)

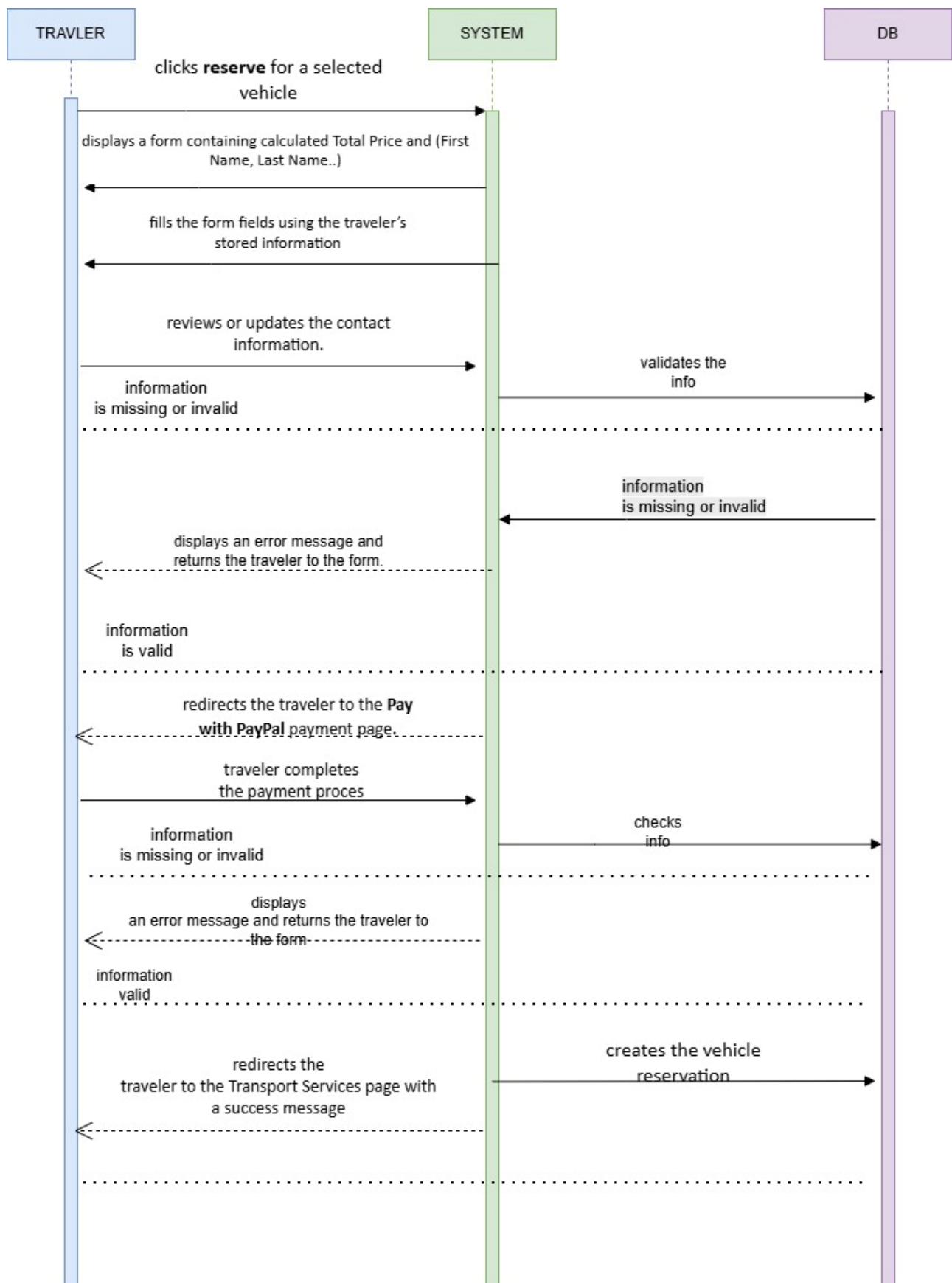


Figure 4. Sequence Diagram(Book Vehicle)

## ● Add to Favorites:

<b>ID</b>	<b>UC-37</b>
<b>Use Case</b>	<b>Add To Favorites</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The traveler is logged into the system and is browsing one of the following pages: (Destinations, Hotels, Activities).
<b>Post Condition</b>	The selected item is added to the traveler's favorites list.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The traveler clicks the <b>favorite (heart)</b> icon for a selected item.</li> <li>2. The system adds the selected item to the traveler's favorites list.</li> <li>3. The system updates the heart icon to indicate the item is marked as favorite (heart turns red).</li> </ol>

Table 40 Add to Favorites

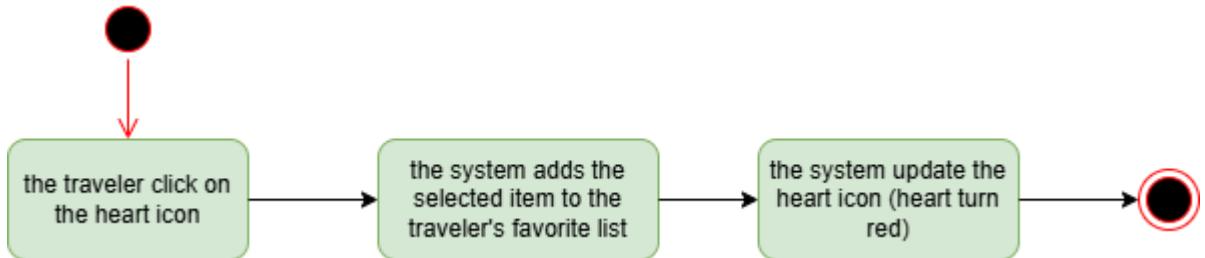


Figure 41 Activity Diagram (Add to Favorites)

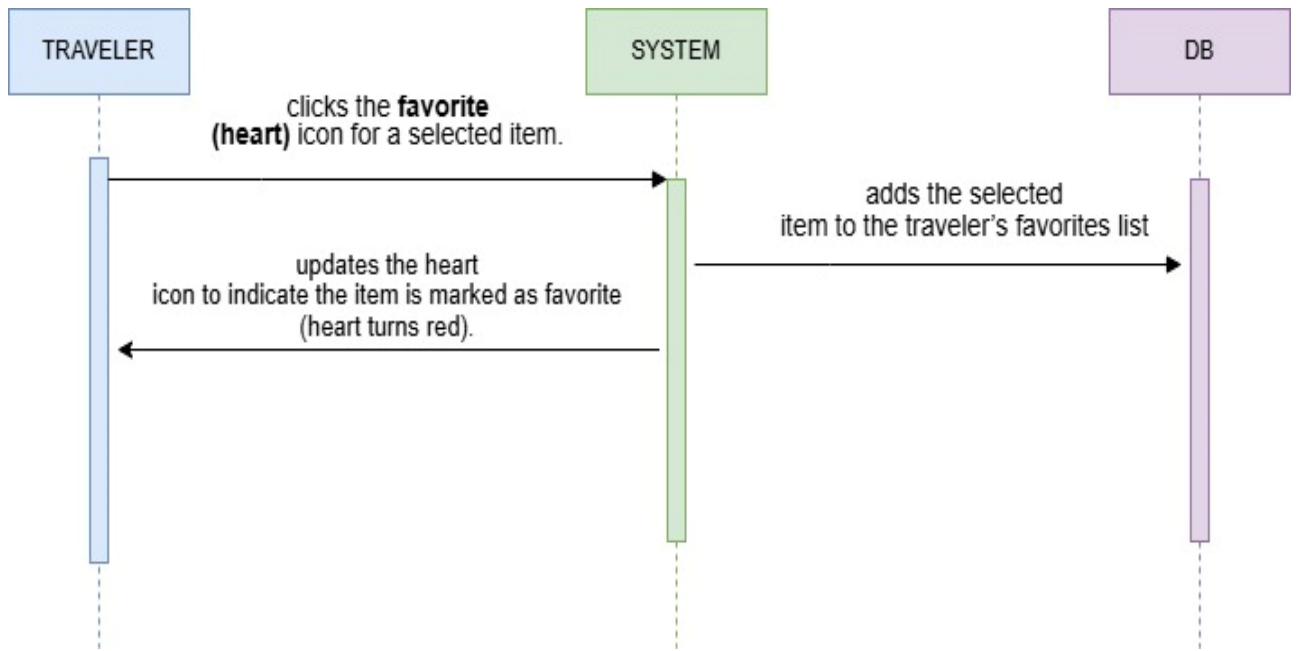


Figure 44 Sequence Diagram (Add to Favorites)

## ● Remove From Favorites:

ID	UC-38
Use Case	Remove From Favorites
Actor	Traveler
Pre Condition	The traveler is logged into the system and is browsing one of the following pages: (Destinations, Hotels, Activities) and the selected item is already marked as favorite.
Post Condition	The selected item is removed from the traveler's favorites list.
Main Scenario	<ol style="list-style-type: none"> <li>1. The traveler clicks the <b>favorite (red heart)</b> icon.</li> <li>2. The system removes the selected item from the traveler's favorites list.</li> <li>3. The system updates the heart icon to indicate the item is no longer a favorite (heart turns white).</li> </ol>

Table 41 Remove From Favorites

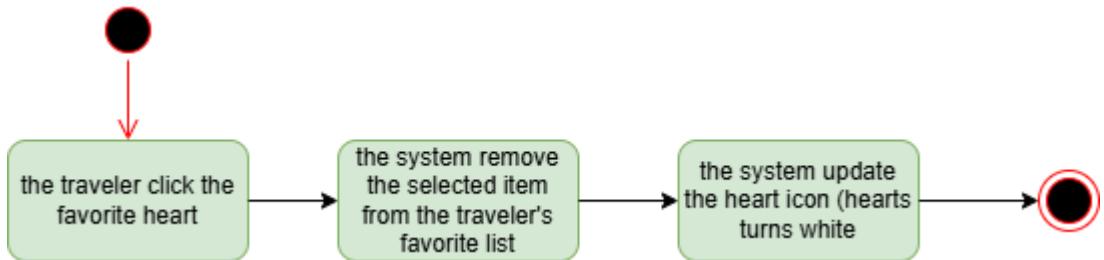


Figure 83 Activity Diagram(Remove From Favorites)

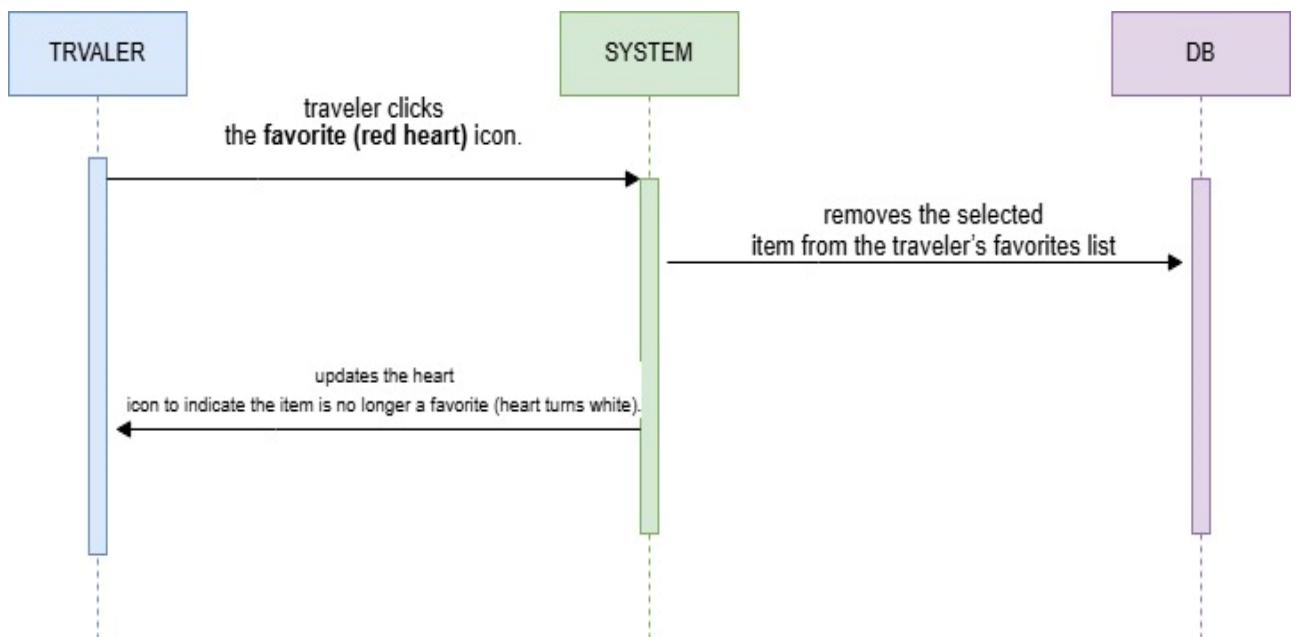


Figure 84 Sequence Diagram(Remove From Favorites)

## ● View Favorites:

<b>ID</b>	UC-39
<b>Use Case</b>	View Favorites
<b>Actor</b>	Traveler
<b>Pre Condition</b>	The traveler is logged into the system.
<b>Post Condition</b>	The traveler views the list of favorite items.
<b>Main Scenario</b>	<p>1. The traveler clicks on their profile name from the navigation bar.</p> <p>2. The traveler clicks <b>Show Favorites</b>.</p> <p>3. The system retrieves the traveler's favorite items.</p> <p>4. The system displays a favorites list containing:</p> <ul style="list-style-type: none"> <li>• Favorite Destinations</li> <li>• Favorite Hotels</li> <li>• Favorite Activities</li> </ul>
<b>Alternative Scenario</b>	If the traveler has no favorite items, the system displays " <b>No favorites found</b> ".

Table 42 View Favorites

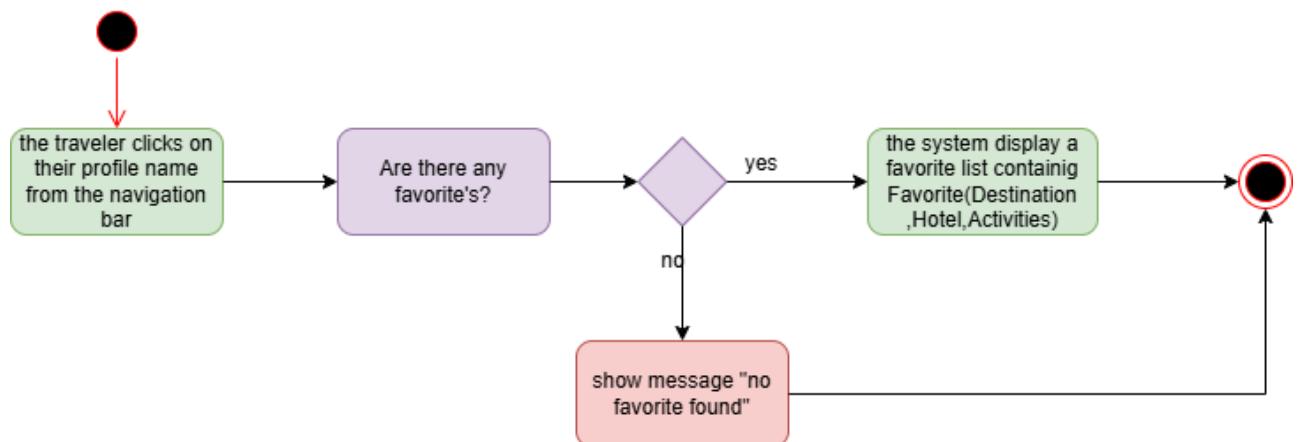


Figure 85 Activity Diagram(View Favorites)

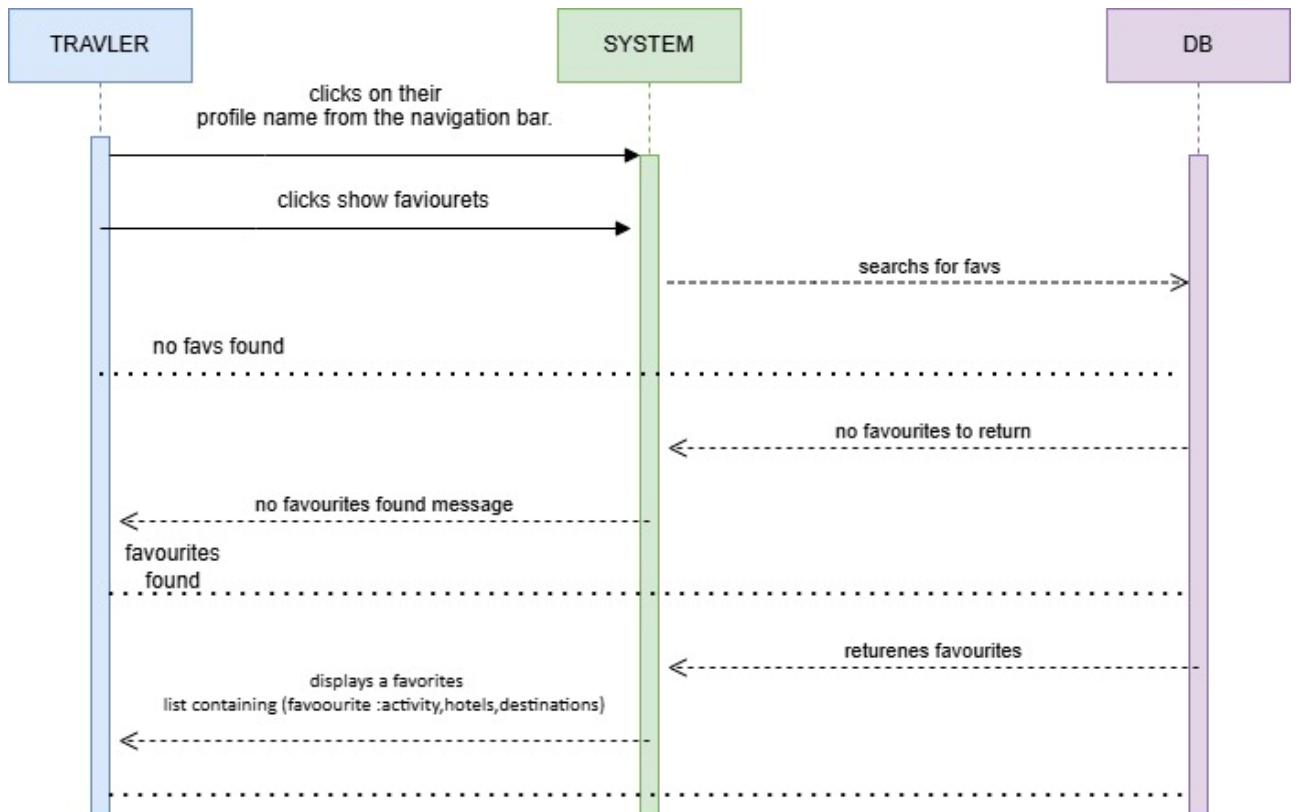


Figure 47 Sequence Diagram(View Favorites)

## ● View Traveler's Reservations:

<b>ID</b>	<b>UC-40</b>
<b>Use Case</b>	View Traveler's Reservations
<b>Actor</b>	Traveler
<b>Pre Condition</b>	The traveler is logged into the system.
<b>Post Condition</b>	The traveler views their reservations.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The traveler clicks on their profile name from the navigation bar.</li> <li>2. The traveler selects one of the following options:           <ul style="list-style-type: none"> <li>• My Hotel Reservations</li> <li>• My Transport Reservations</li> </ul> </li> <li>3. The system retrieves the traveler's reservations based on the selected type.</li> <li>4. The system displays the list of reservations with detailed information for each reservation.</li> </ol>
<b>Alternative Scenario</b>	If no reservations exist for the selected type, the system displays " <b>No reservations found</b> ".

Table 43 View Traveler's Reservations

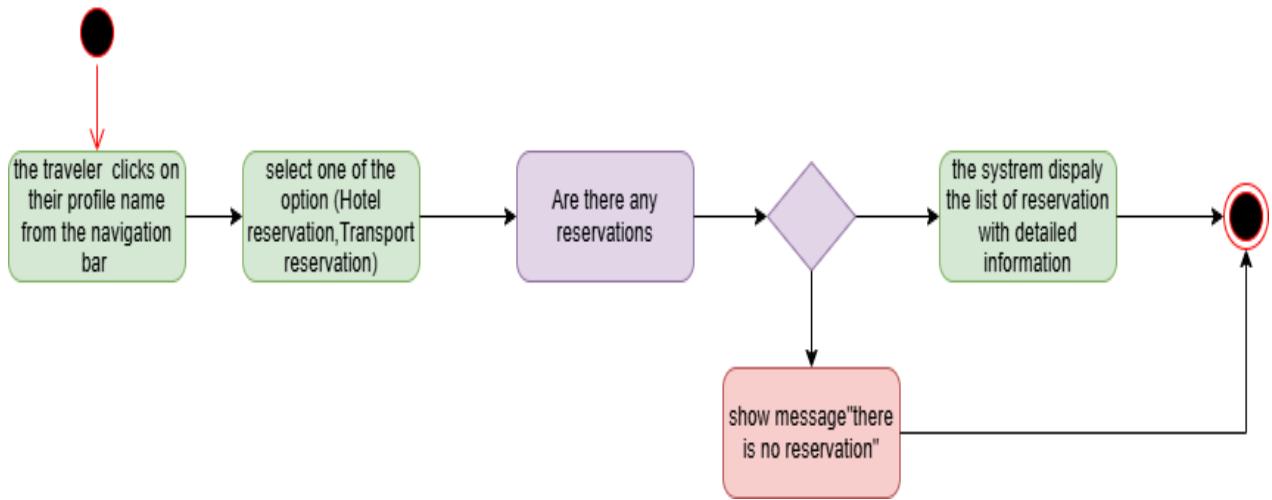
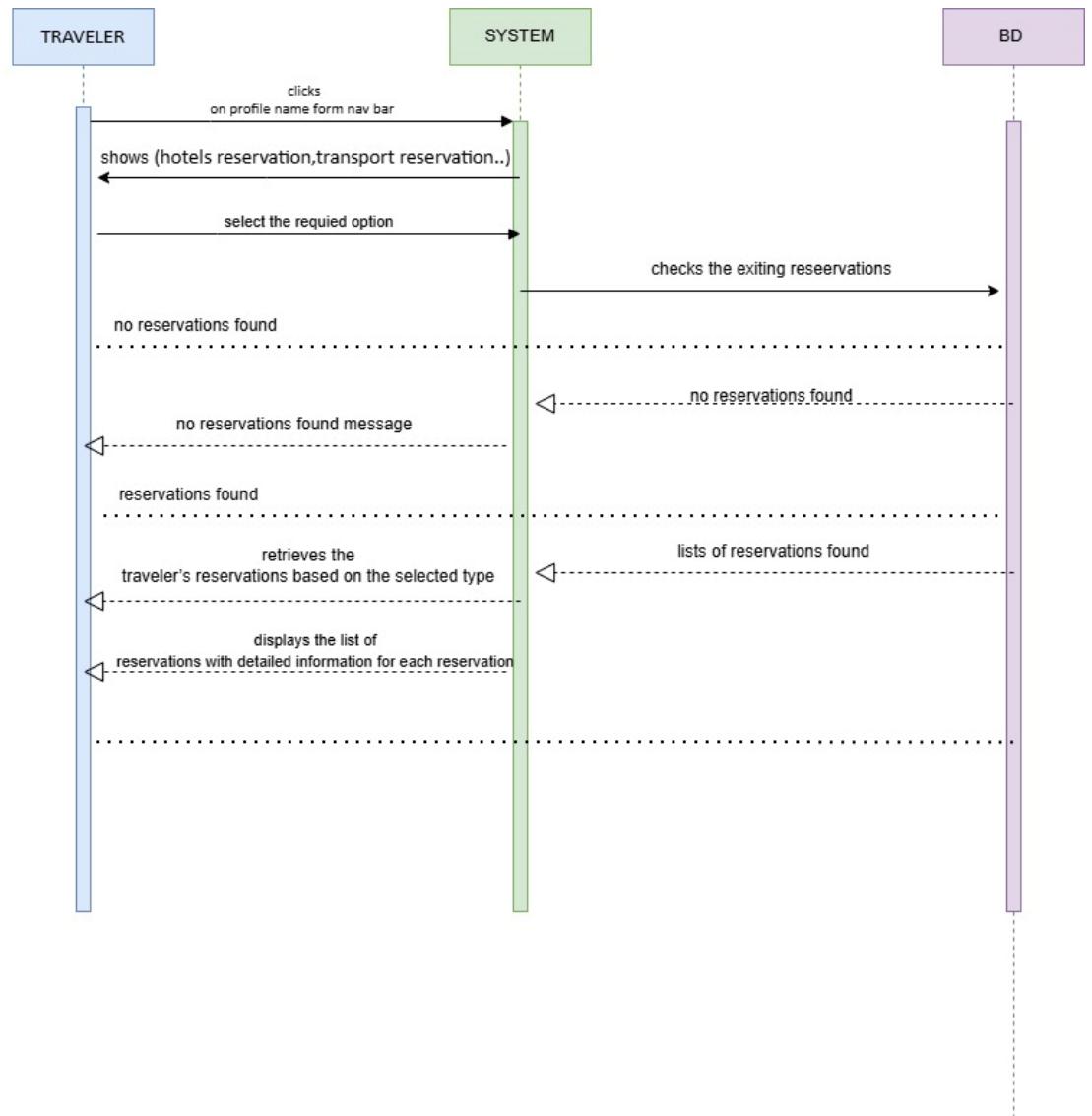


Figure 87 Activity Diagram(View Traveler's Reservations)



*Figure 11 Sequence Diagram( View Traveler's Reservations)*

## ● View System Reservations:

<b>ID</b>	<b>UC-41</b>
<b>Use Case</b>	<b>View System Reservations</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The admin is logged into the system.
<b>Post Condition</b>	The admin views system reservations.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks <b>Reservations</b> from the navigation bar.</li> <li>2. The admin selects one of the following options: <ul style="list-style-type: none"> <li>• Hotel Reservations</li> <li>• Transport Reservations</li> </ul> </li> <li>3. The system retrieves all reservations for the selected category.</li> <li>4. The system displays a list of reservations with their related details.</li> </ol>
<b>Alternative Scenario</b>	If no reservations exist for the selected category, the system displays <b>“No reservations found”</b> .

Table 44 View System Reservations

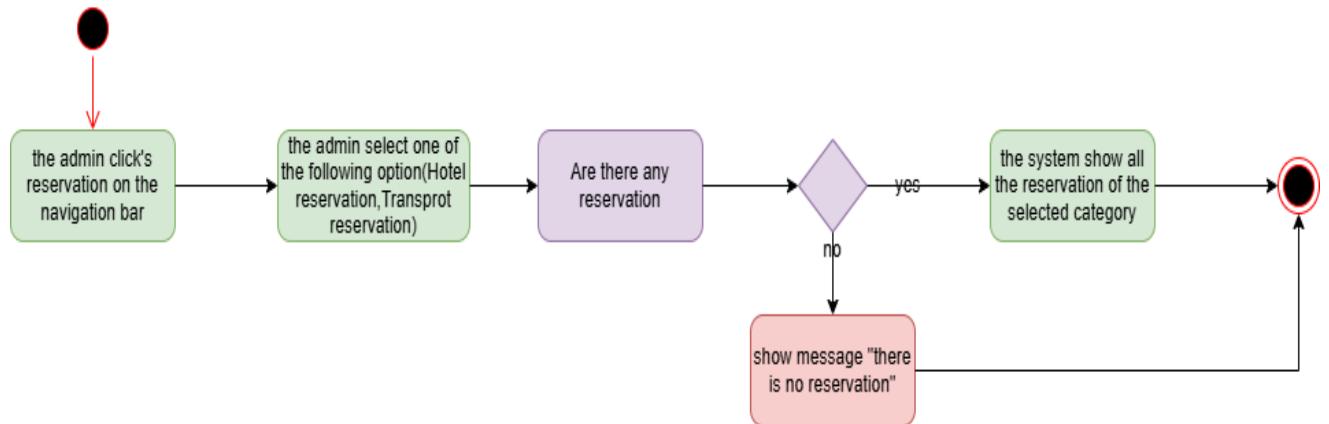


Figure 89 Activity Diagram(View System Reservations)

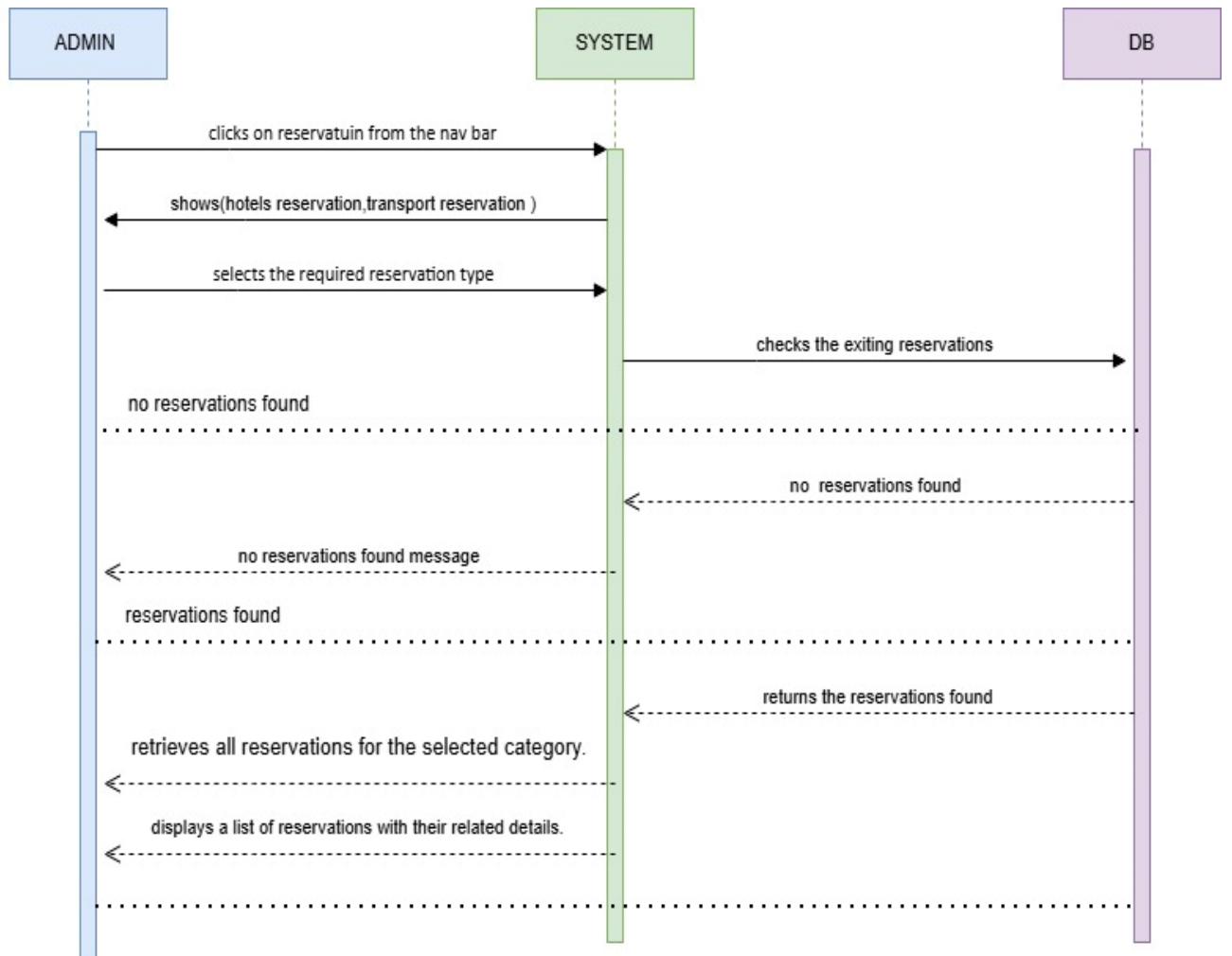


Figure 1 • Sequence Diagram (View System Reservations)

## ● View Drivers:

<b>ID</b>	<b>UC-42</b>
<b>Use Case</b>	<b>View Drivers</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The admin is logged into the system.
<b>Post Condition</b>	The admin views all drivers in the system along with their pending requests, if any.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks <b>Drivers</b> from the navigation bar.</li> <li>2. The system retrieves all drivers and driver requests from the database.</li> <li>3. The system displays a list of drivers with their relevant details and current status.</li> </ol>
<b>Alternative Scenario</b>	If no drivers exist, the system displays “ <b>No drivers found</b> ”.

Table 45 View Drivers

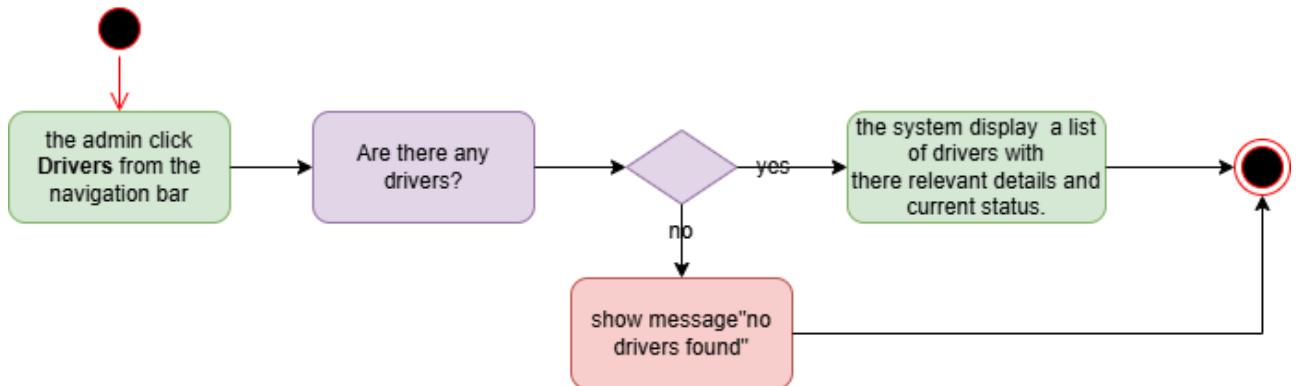


Figure 11 Activity Diagram (View Drivers)

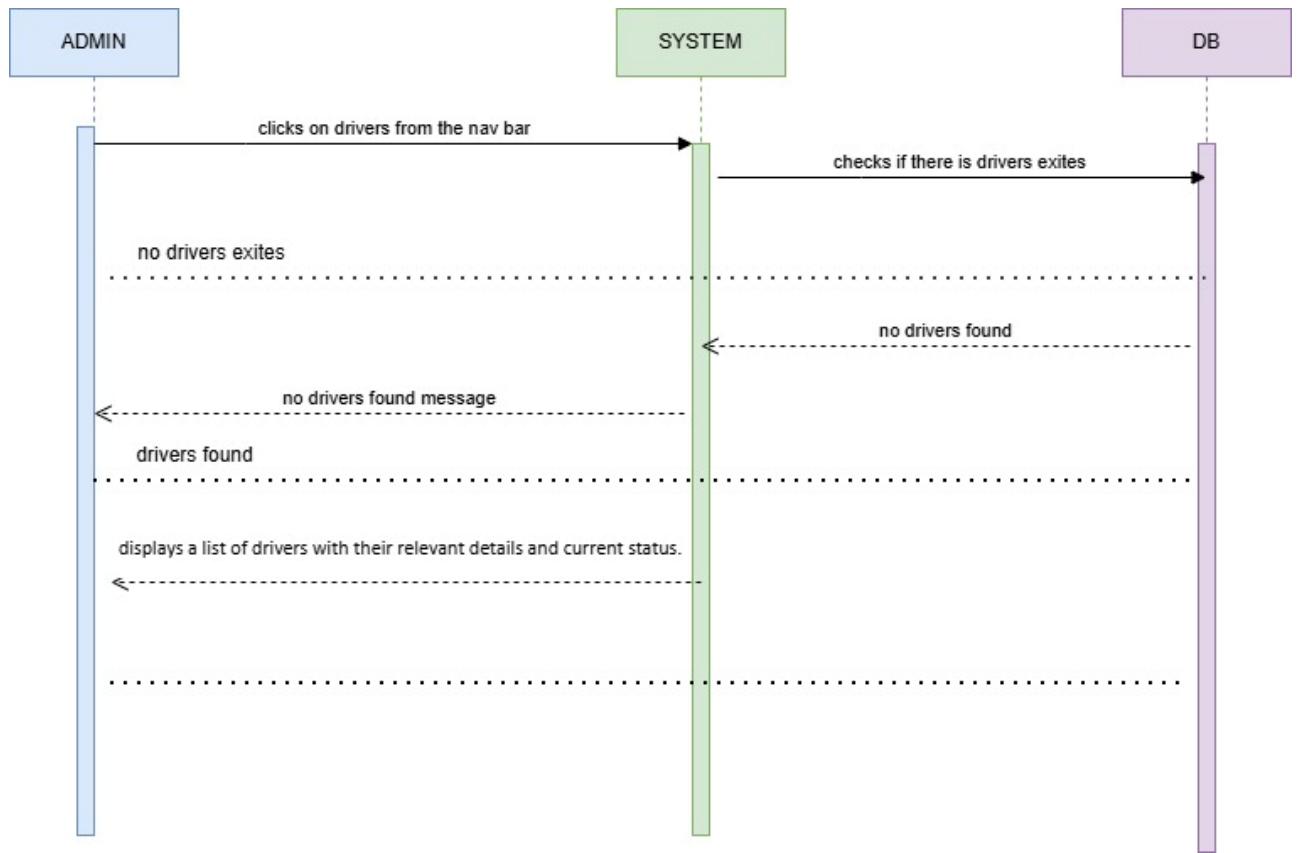


Figure 17 Sequence Diagram(View Drivers)

## ● Accept/Reject Driver:

<b>ID</b>	<b>UC-43</b>
<b>Use Case</b>	Accept/Reject Driver
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	The admin is logged into the system and is on the <b>View Drivers</b> page.
<b>Post Condition</b>	<ul style="list-style-type: none"> <li>- The driver's status is updated to <b>Approved</b> or <b>Rejected</b>.</li> <li>- The system sends a notification email.</li> <li>- If approved, the driver is added to the system with a <b>Date of Hire</b>.</li> <li>- If rejected, the driver is removed from the system.</li> </ul>
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin selects a driver with <b>Pending</b> status.</li> <li>2. The system displays a <b>Select Status</b> dropdown with options: <b>Pending (default)</b>, <b>Approved</b>, <b>Rejected</b>.</li> <li>3. The admin selects either <b>Approved</b> or <b>Rejected</b>.</li> <li>4. The admin clicks <b>Confirm</b>.</li> <li>5. The system updates the driver status accordingly: <ul style="list-style-type: none"> <li>- If <b>Approved</b>: <ul style="list-style-type: none"> <li>• Adds the driver to the system.</li> <li>• Updates the <b>Date of Hire</b>.</li> <li>• Sends an approval email to the driver.</li> </ul> </li> <li>- If <b>Rejected</b>: <ul style="list-style-type: none"> <li>• Removes the driver from the system.</li> <li>• Sends a rejection email to the driver.</li> </ul> </li> </ul> </li> <li>6. The system refreshes the <b>View Drivers</b> list to reflect the updated status with a success message.</li> </ol>
<b>Alternative Scenario</b>	If the admin does not select a status and clicks <b>Confirm</b> , the system displays an error message indicating that a selection must be made.

Table 46 Accept/Reject Driver:

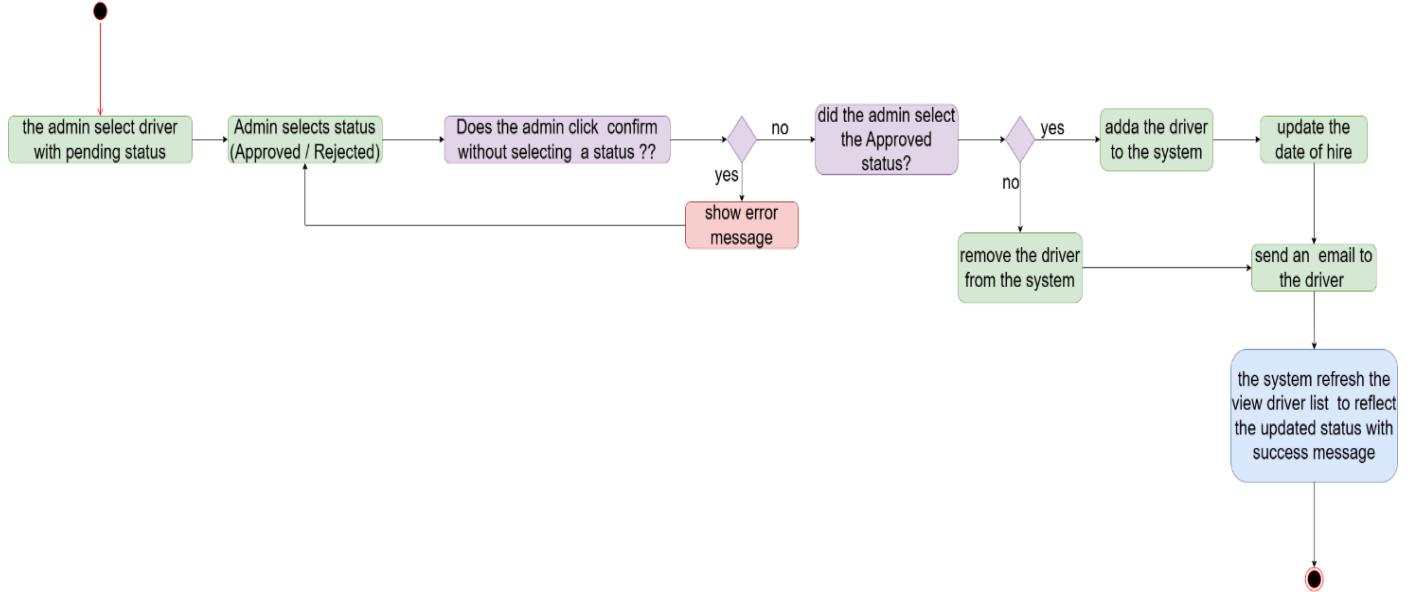
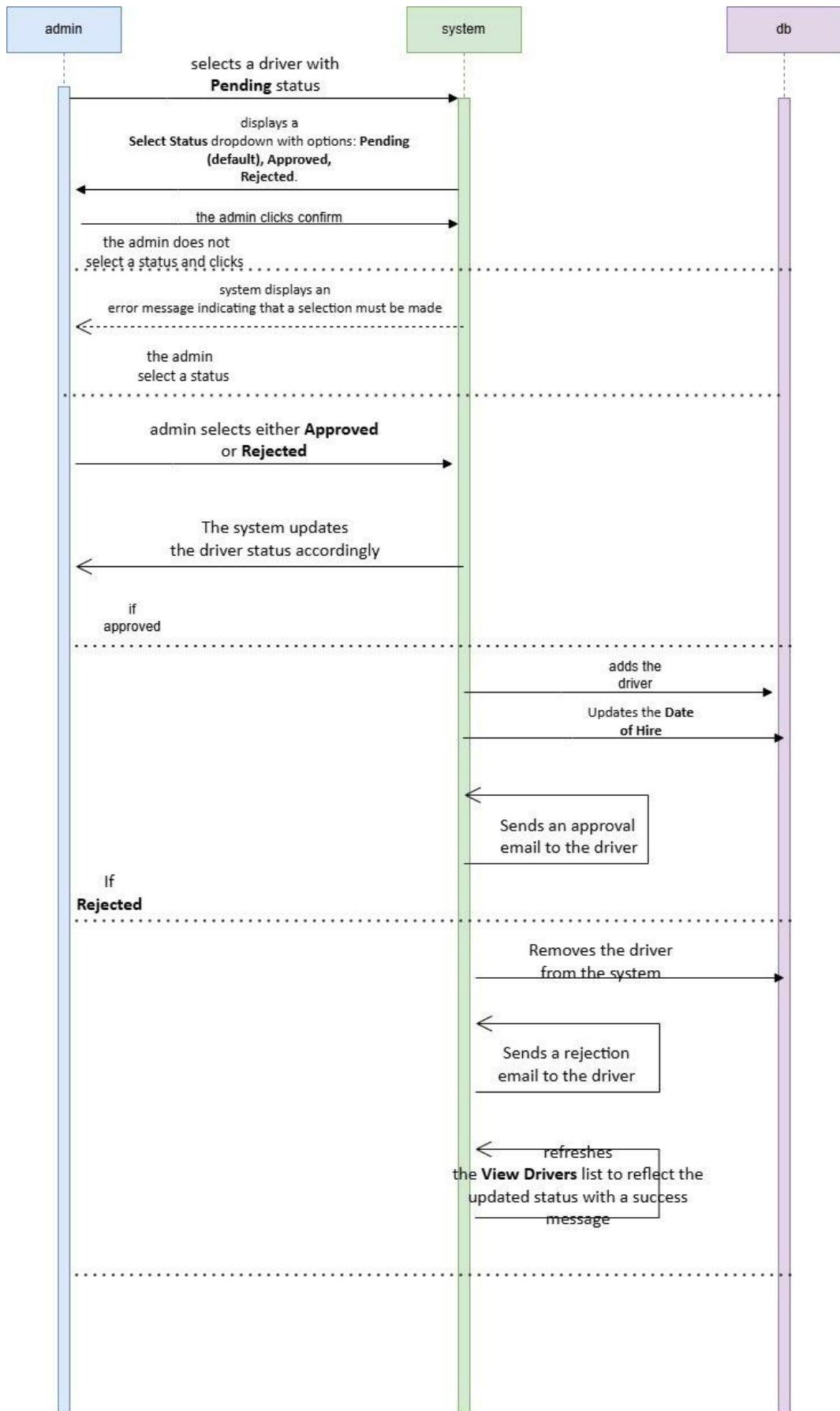


Figure 11 Activity Diagram (Accept/Reject Driver:)



Sequence Diagram-Accept/Reject Driver 12 Figure

## ● Delete Driver:

<b>ID</b>	<b>UC-44</b>
<b>Use Case</b>	Delete Driver
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>- The admin is logged into the system.</li> <li>- The admin is on the <b>View Drivers</b> page.</li> <li>- The driver is approved and exists in the system</li> </ul>
<b>Post Condition</b>	The driver is deleted from the system
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin clicks <b>Delete</b> for a selected driver</li> <li>2. The system checks whether the driver has upcoming (not yet completed) reservations.</li> <li>3. The system deletes the driver from the system.</li> <li>4. The system displays a success message on the Drivers page.</li> </ol>
<b>Alternative Scenario</b>	If the driver has upcoming reservations, the system prevents the deletion and displays an error message.

Table 47 Delete Driver

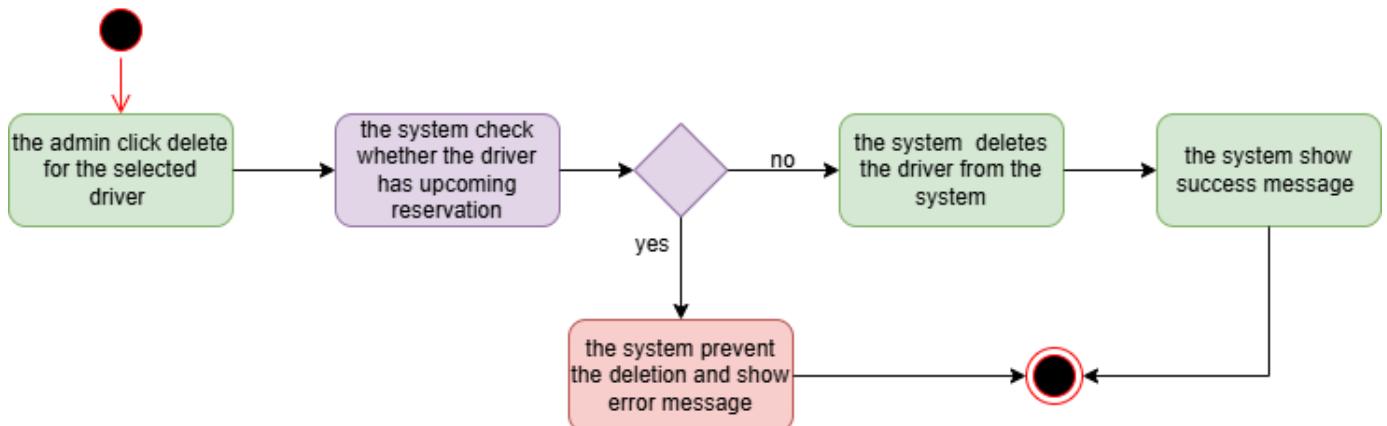


Figure 47 Activity Diagram (Delete Driver)

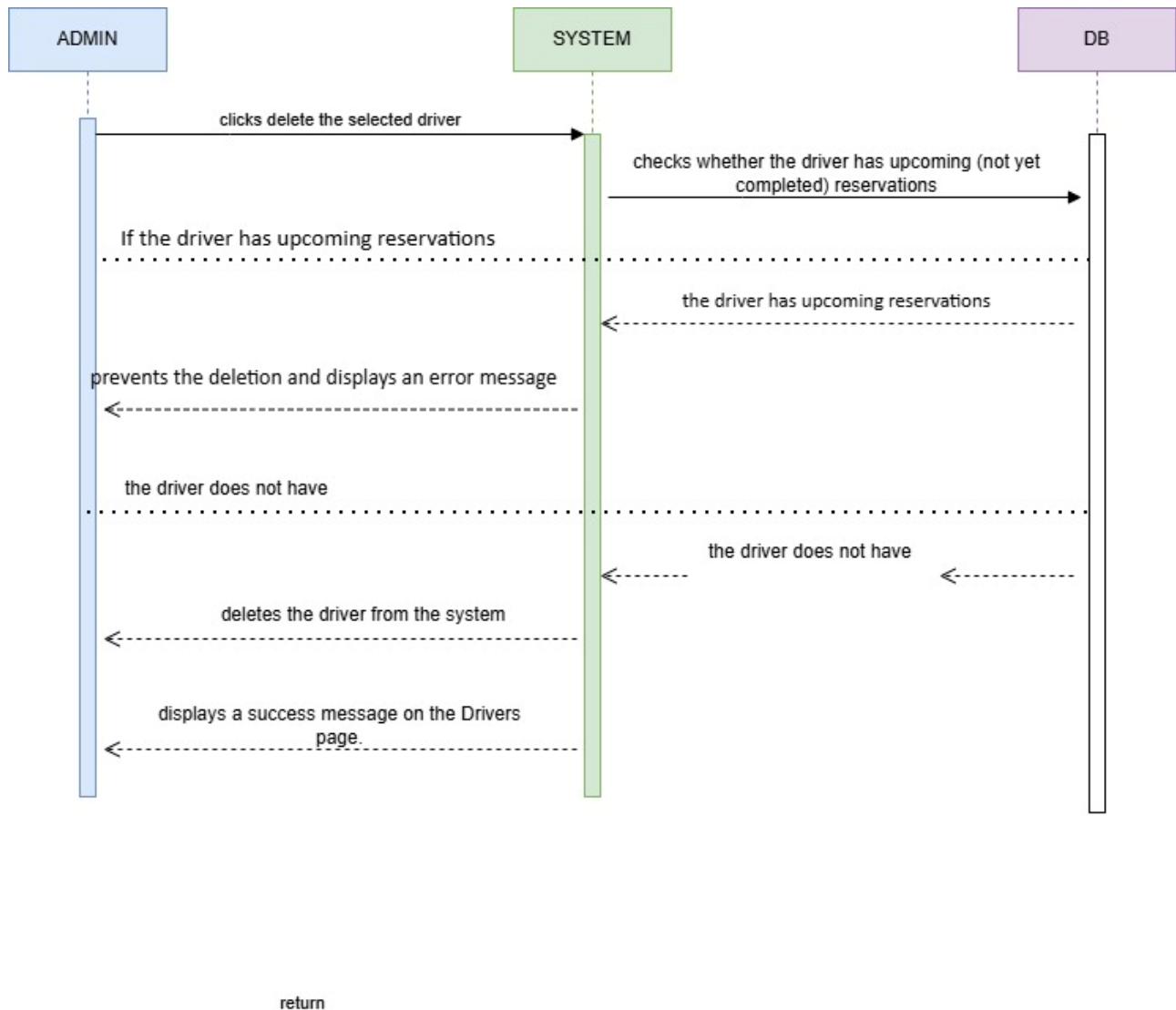


Figure 11 Sequence Diagram(Delete Driver)

## ● View Driver Bookings:

<b>ID</b>	UC-45
<b>Use Case</b>	View Driver Bookings
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>- The admin is logged into the system.</li> <li>- The admin is on the <b>View Drivers</b> page.</li> <li>- The driver is approved.</li> </ul>
<b>Post Condition</b>	The admin views all reservations of the selected driver, categorized by status (Pending or Completed).
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The admin selects a reservation category (<b>Pending</b> or <b>Completed</b>) for a chosen driver..</li> <li>2. The system displays the driver's reservations according to the selected category.</li> <li>3. The system lists each reservation with the following details: (Client Name, Vehicle, Pickup Location, Drop-off Location, Pickup Date and Time, Number of Passengers, Total Price, Reservation Status).</li> </ol>
<b>Alternative Scenario</b>	<p>If the driver has no reservations in the selected category :</p> <ul style="list-style-type: none"> <li>• The system displays a message indicating that no reservations are available.</li> </ul>

Table 48 View Driver Bookings

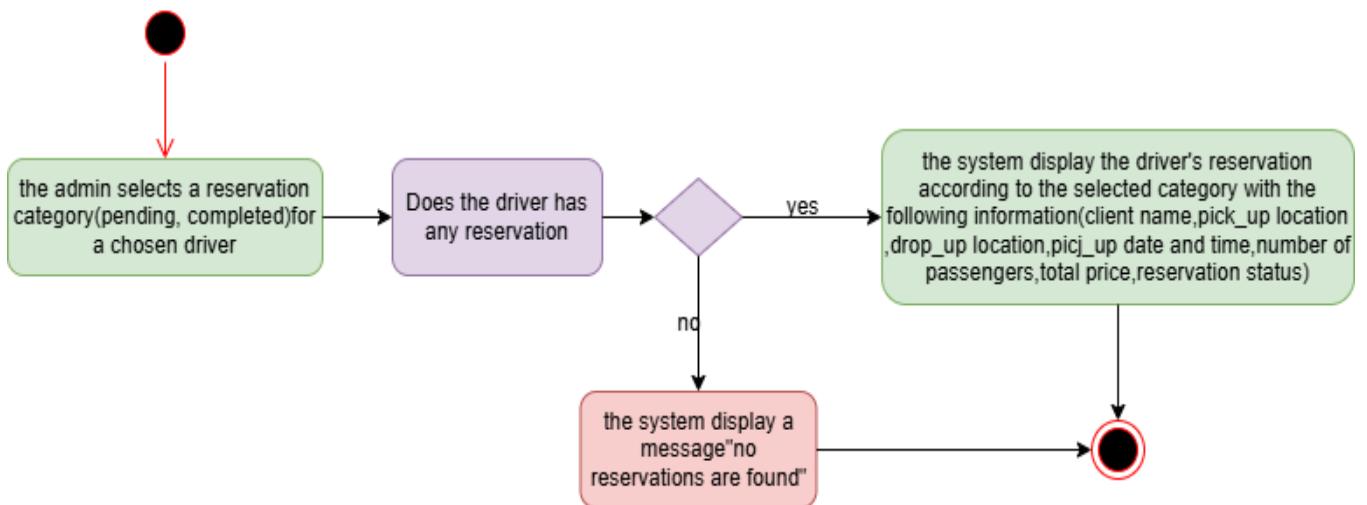


Figure 48 Activity Diagram (View Driver Bookings)

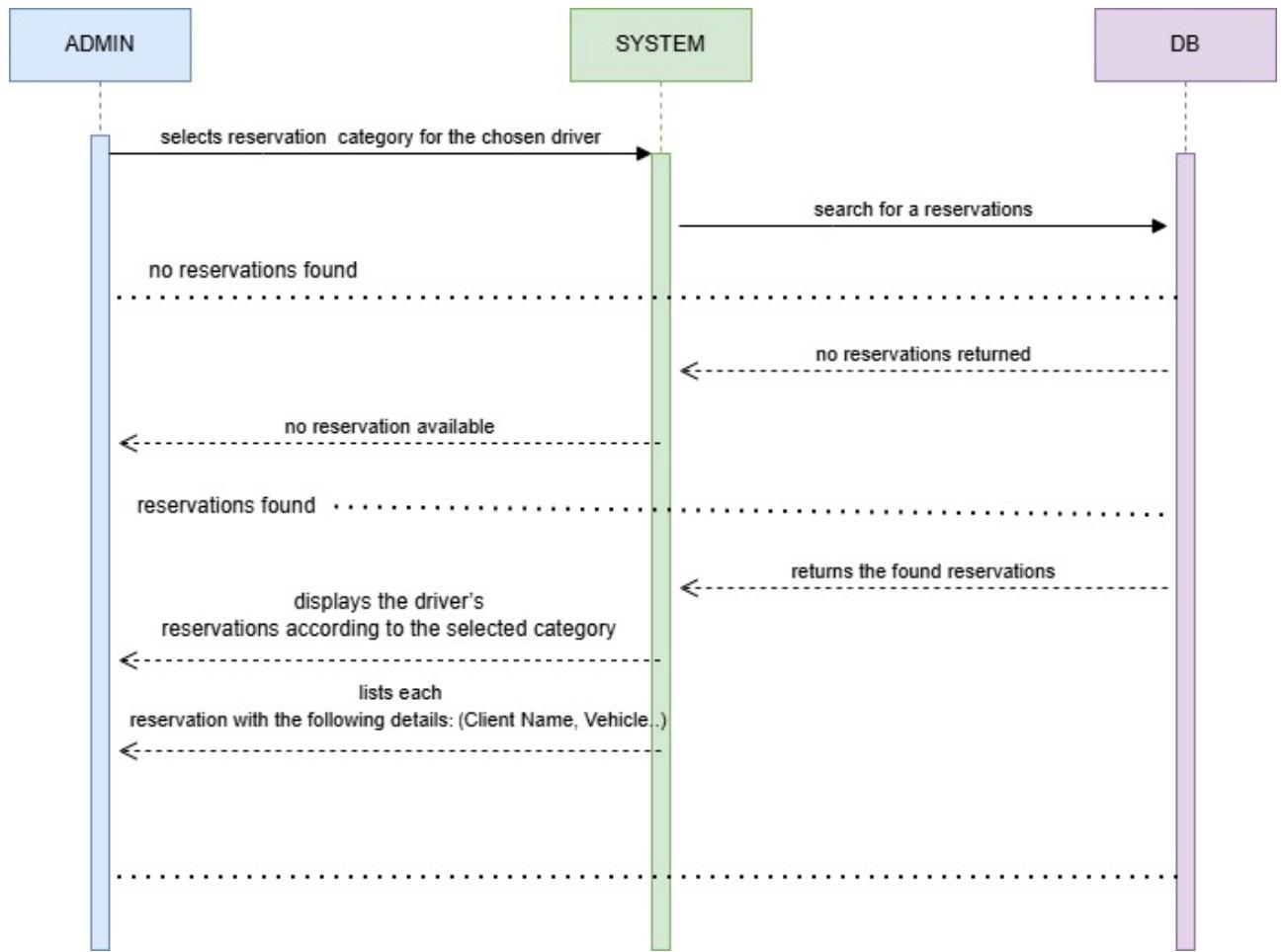


Figure 14 Sequence Diagram(View Driver Bookings)

## ● Search Drivers:

<b>ID</b>	<b>UC-46</b>
<b>Use Case</b>	<b>Search &amp; Filter Drivers</b>
<b>Actor</b>	<b>Admin</b>
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>- The admin is logged into the system.</li> <li>- The admin is on the Drivers page.</li> </ul>
<b>Post Condition</b>	The system displays drivers matching the search and filter criteria.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1.The admin enters a keyword in the <b>Search Bar</b> (Driver First Name, Last Name).</li> <li>2. The admin optionally selects a filter from the available filter options: (Status , License Category, Country)</li> <li>3. The system searches and filters drivers based on the entered keyword and selected criteria.</li> <li>4. The system updates the drivers list to display only the matching results including relevant details: (First Name and Last Name, Email, Contact Number, Address, License Category and image, Status, Date of Hire (if applicable)) with action buttons.</li> </ol>
<b>Alternative Scenario</b>	If no drivers match the search or filter criteria, the system displays <b>“No drivers found.”</b>

Table 49 Search & Filter Drivers:

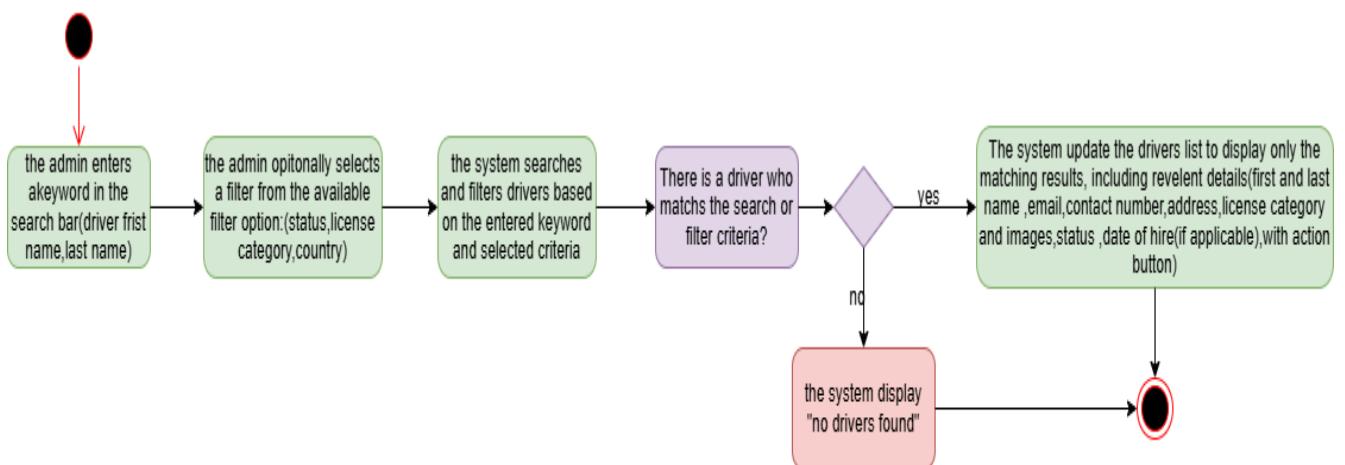


Figure 11 Activity Diagram (Search Drivers)

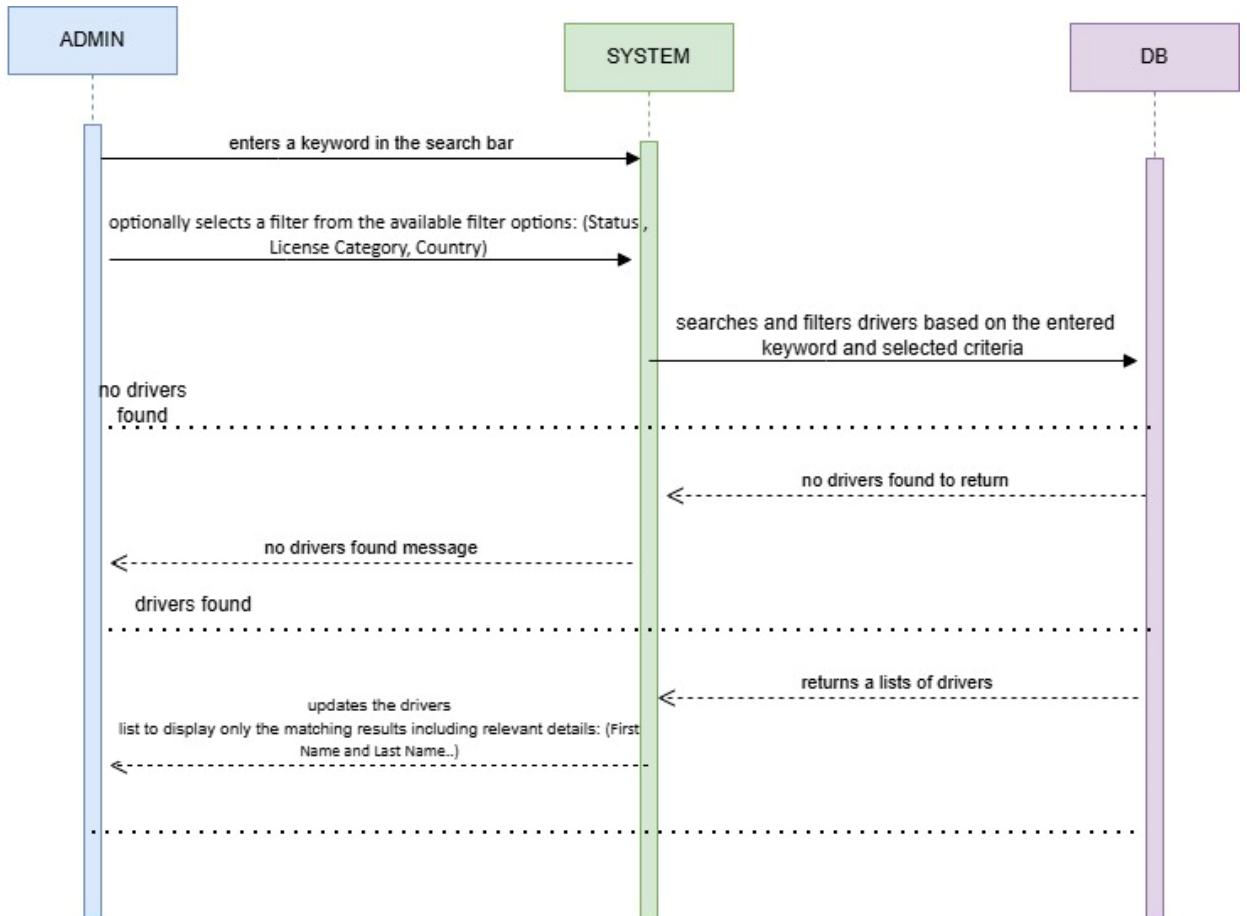


Figure 1 • Sequence Diagram(Search Drivers)

## ● View Assigned Bookings:

<b>ID</b>	<b>UC-47</b>
<b>Use Case</b>	<b>View Assigned Bookings</b>
<b>Actor</b>	<b>Driver</b>
<b>Pre Condition</b>	- The driver is logged into the system
<b>Post Condition</b>	The driver views their assigned bookings categorized by status (Pending and Completed).
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The driver clicks <b>Pending Bookings</b> or <b>Completed Bookings</b> from the navigation bar.</li> <li>2. The system retrieves the driver's bookings based on the selected category.</li> <li>3. The system displays a list of bookings with their relevant details.</li> </ol>
<b>Alternative Scenario</b>	<ul style="list-style-type: none"> <li>• If no bookings exist for the selected category, the system displays "<b>No bookings found.</b>"</li> </ul>

Table 50 View Assigned Bookings

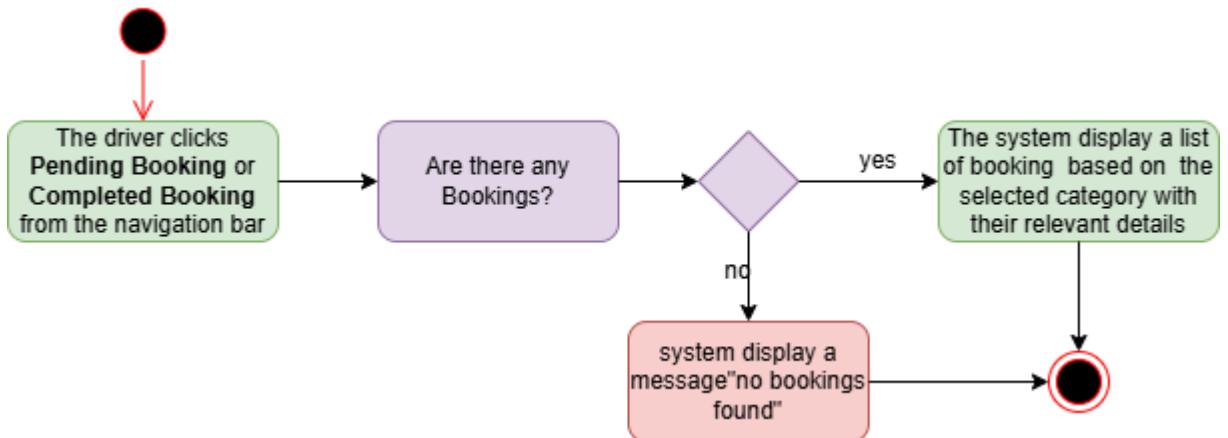


Figure 1·1 Activity Diagram(View Assigned Bookings)

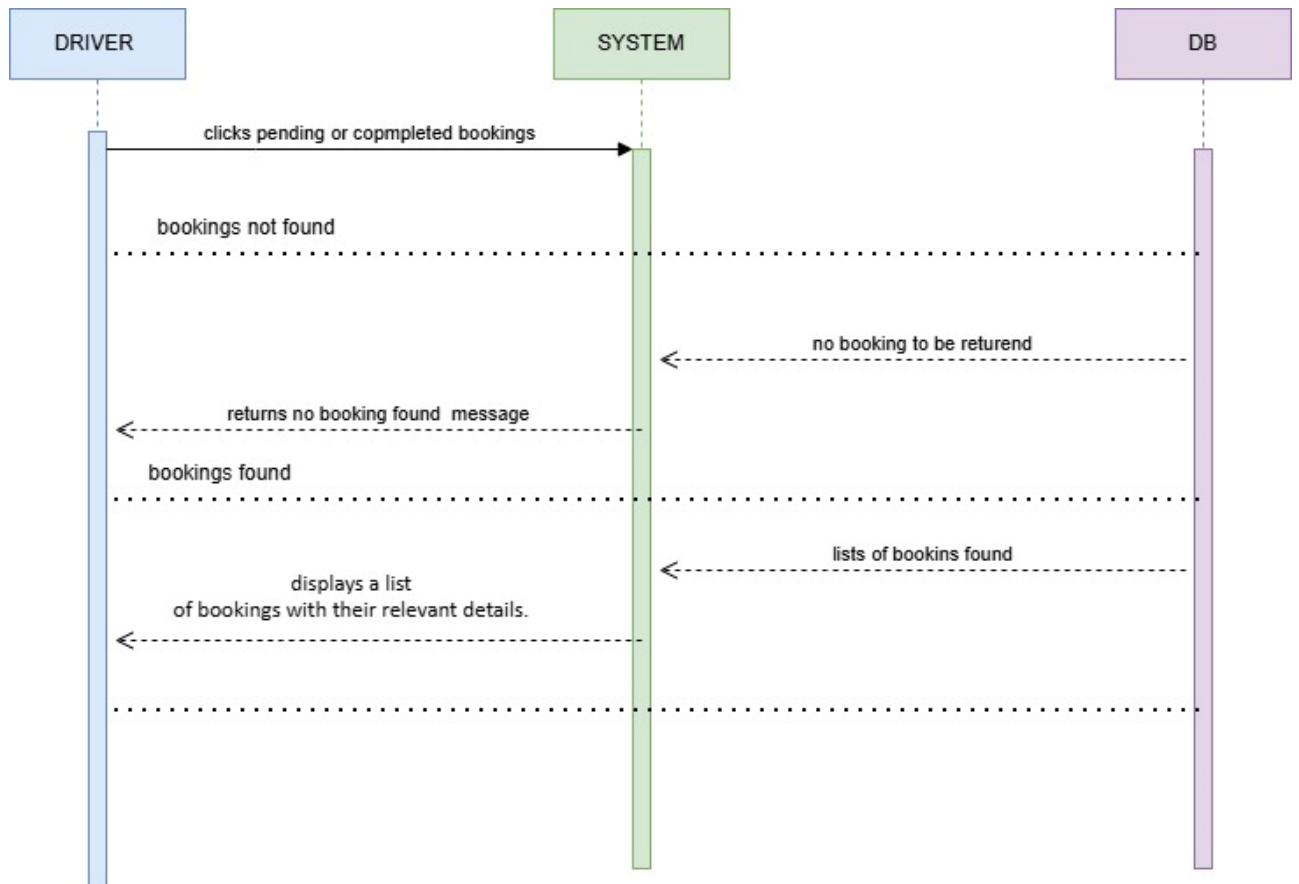


Figure 1.1 Sequence Diagram(View Assigned Bookings)

## ● Update Reservation Status:

<b>ID</b>	UC-48
<b>Use Case</b>	Update Reservation Status
<b>Actor</b>	Driver
<b>Pre Condition</b>	<ul style="list-style-type: none"> <li>- The driver is logged into the system.</li> <li>- The driver is on the Pending Bookings page.</li> </ul>
<b>Post Condition</b>	The reservation status is updated to <b>Completed</b> or <b>Canceled</b> .
<b>Main Scenario</b>	<p>1. The driver chooses an action(<b>Confirm</b> or <b>Cancel</b>) for a specified booking :</p> <ul style="list-style-type: none"> <li>- If <b>Confirm</b>:           <ol style="list-style-type: none"> <li>1. The system updates the reservation status to <b>Completed</b>.</li> <li>2. The system moves the reservation to the Completed Reservations list.</li> <li>3. The system displays a success message.</li> </ol> </li> <li>-if <b>Canceled</b>:           <ol style="list-style-type: none"> <li>1. The system updates the reservation status to <b>Canceled</b>.</li> <li>2. The system removes the reservation from the Pending Bookings list.</li> <li>3. The system displays a cancellation confirmation message.</li> </ol> </li> </ul>

Table 51 Update Reservation Status

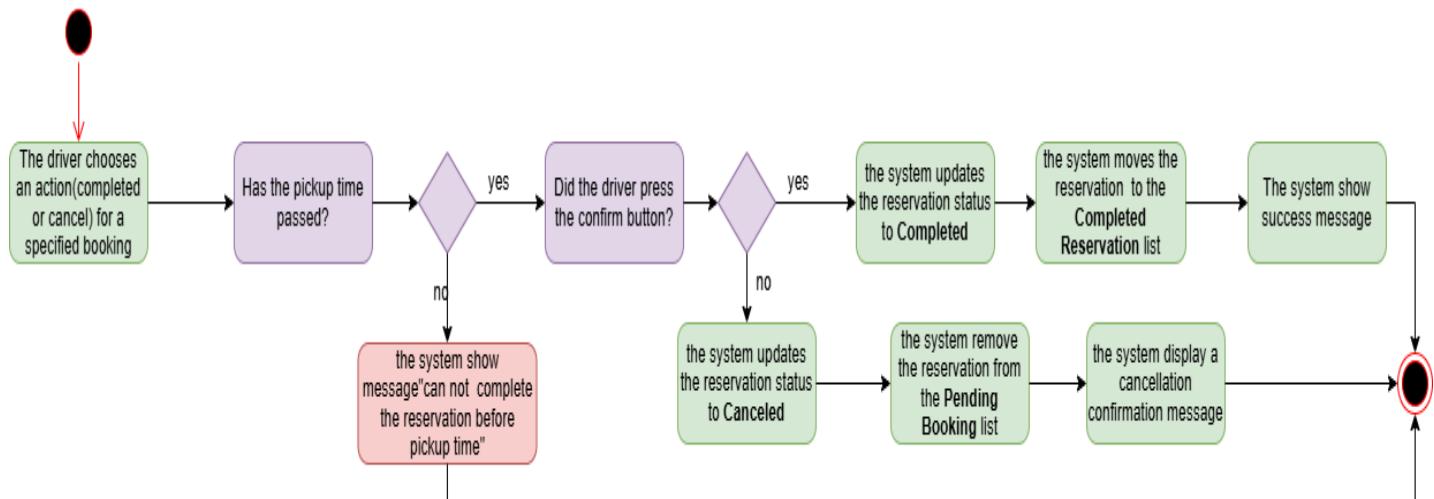


Figure 1.17 Activity Diagram(Update Reservation Status)

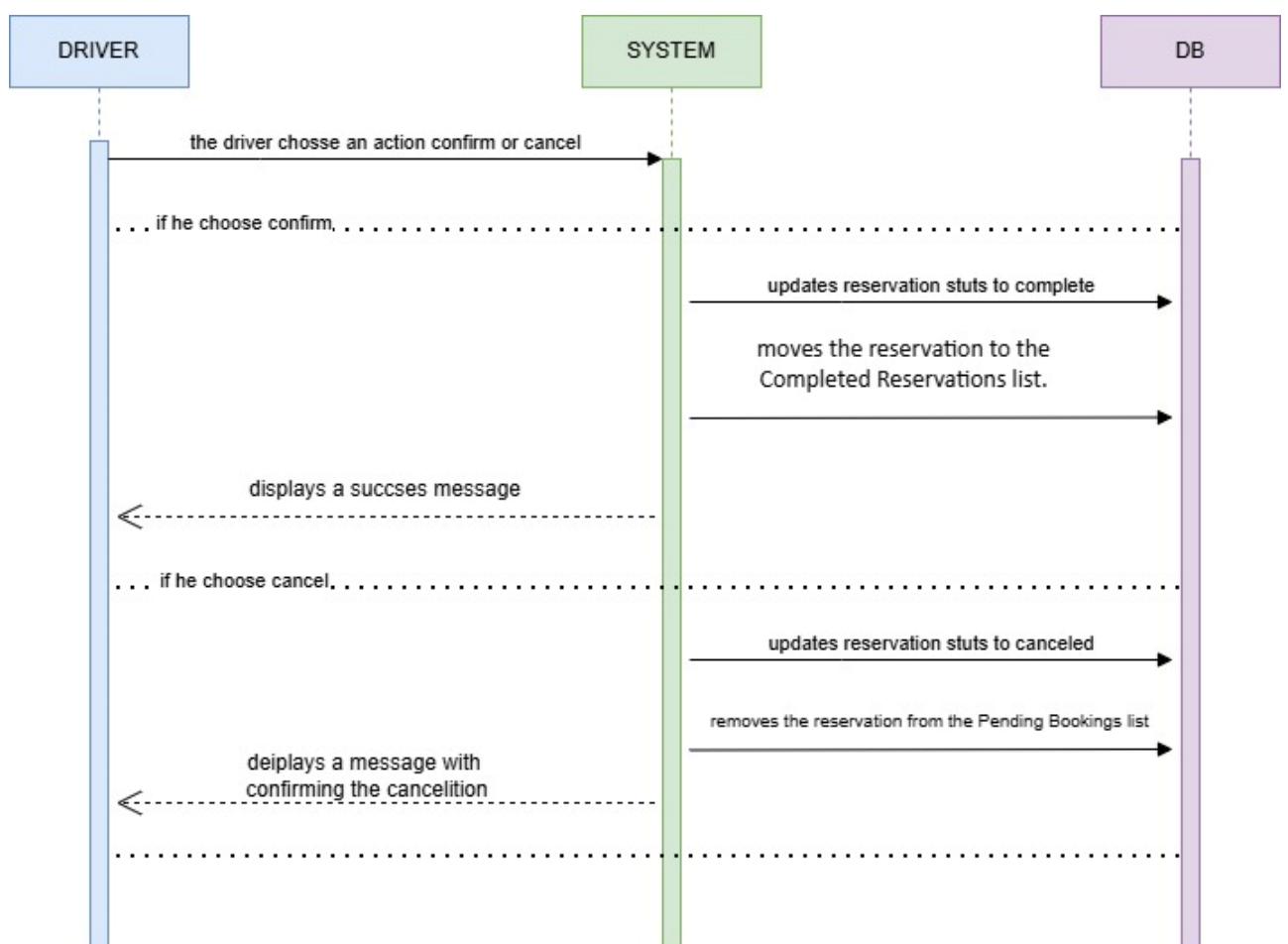


Figure 1.4 Sequence Diagram (Update Reservation Status)

## ● Receive Notifications:

<b>ID</b>	<b>UC-49</b>
<b>Use Case</b>	<b>Receive Notifications</b>
<b>Actor</b>	<b>Traveller/Driver</b>
<b>Pre Condition</b>	The Traveller/Driver is logged into the system A Reservation has been created
<b>Post Condition</b>	View all notification
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The user clicks the <b>Ring icon</b>.</li> <li>2. The system display latest notification and button to display all the notifications</li> <li>3. The user click <b>view All</b></li> <li>4. The system display all notification with details.</li> </ol>
<b>Alternative Scenario</b>	If there is no notification the system display a message “no notification found”

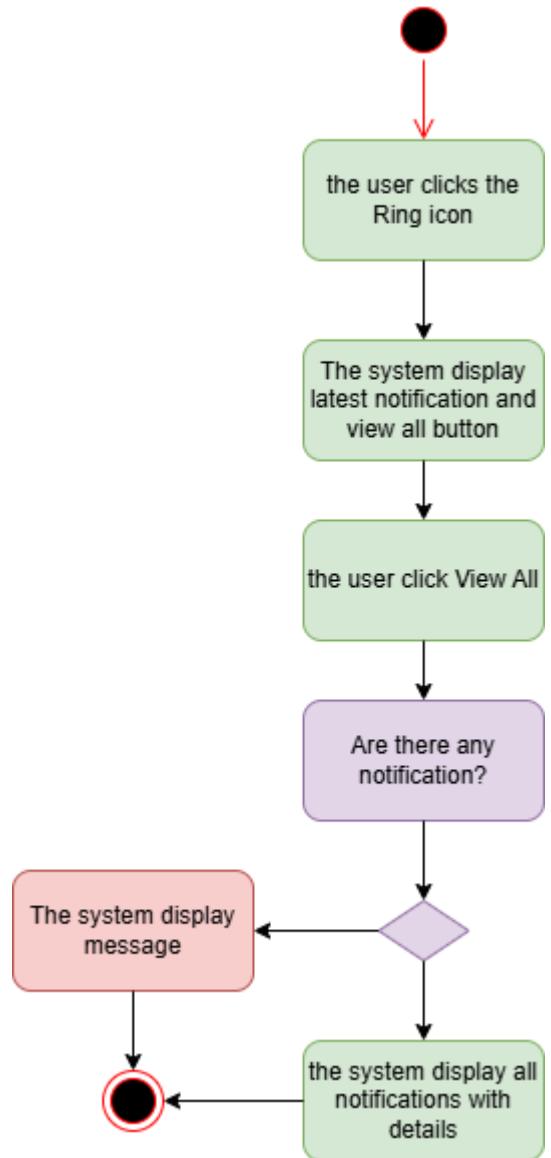


Figure 105 Activity Diagram (Receive Notifications)

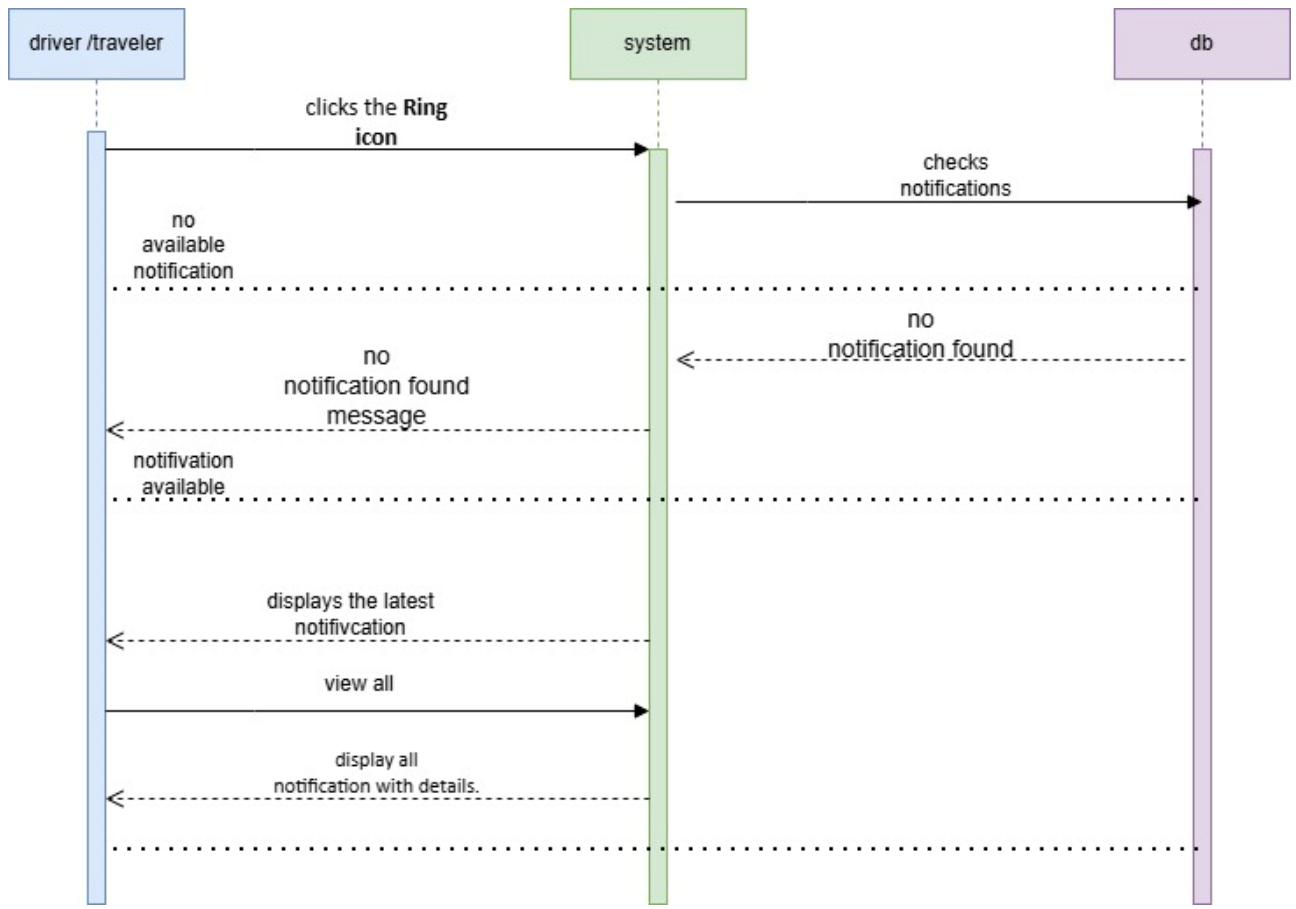


Figure 106 Sequence Diagram(Receive Notifications)

## ● View Trips Plans:

<b>ID</b>	<b>UC-50</b>
<b>Use Case</b>	<b>View Trips Plan</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The user is already logged into the system.
<b>Post Condition</b>	The Traveler views his Plans
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The Traveler clicks the <b>Trips</b> button from the navigation bar.</li> <li>2. The Traveler click <b>my trips</b> button</li> <li>3. The system displays a list of available Trips with dates ,Number of traveler, Description.</li> </ol>
<b>Alternative Scenario</b>	If there is no Trips plans,show message"no Plans found"

Table 52 View Trips plans

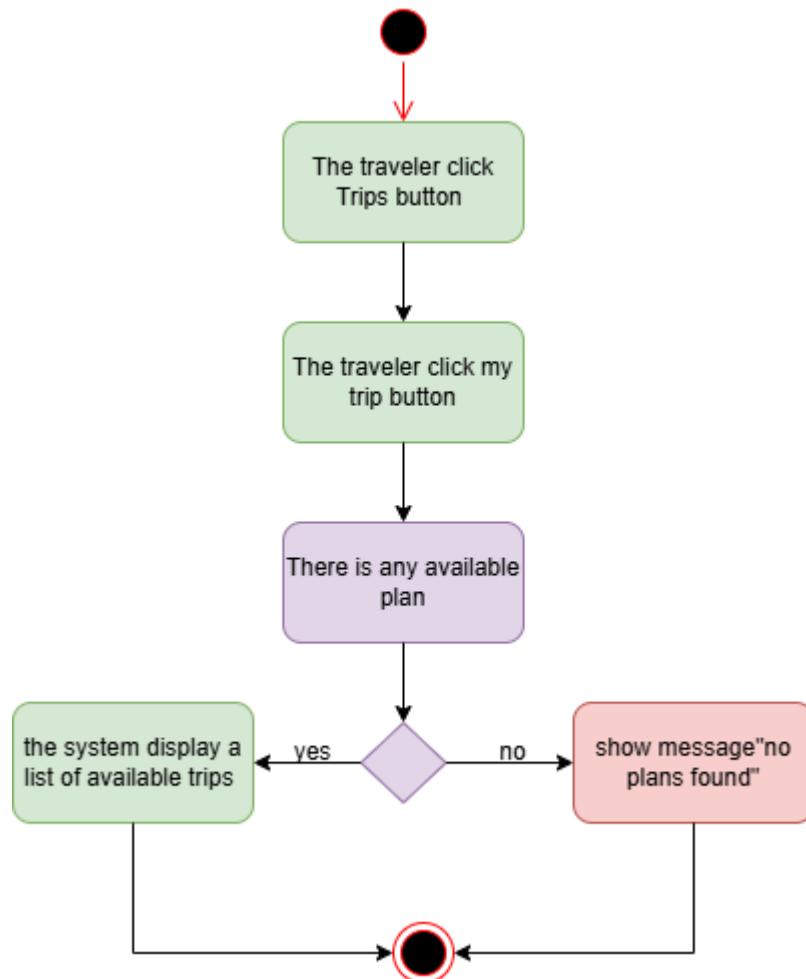


Figure 107 Activity Diagram(View Trip Plan)

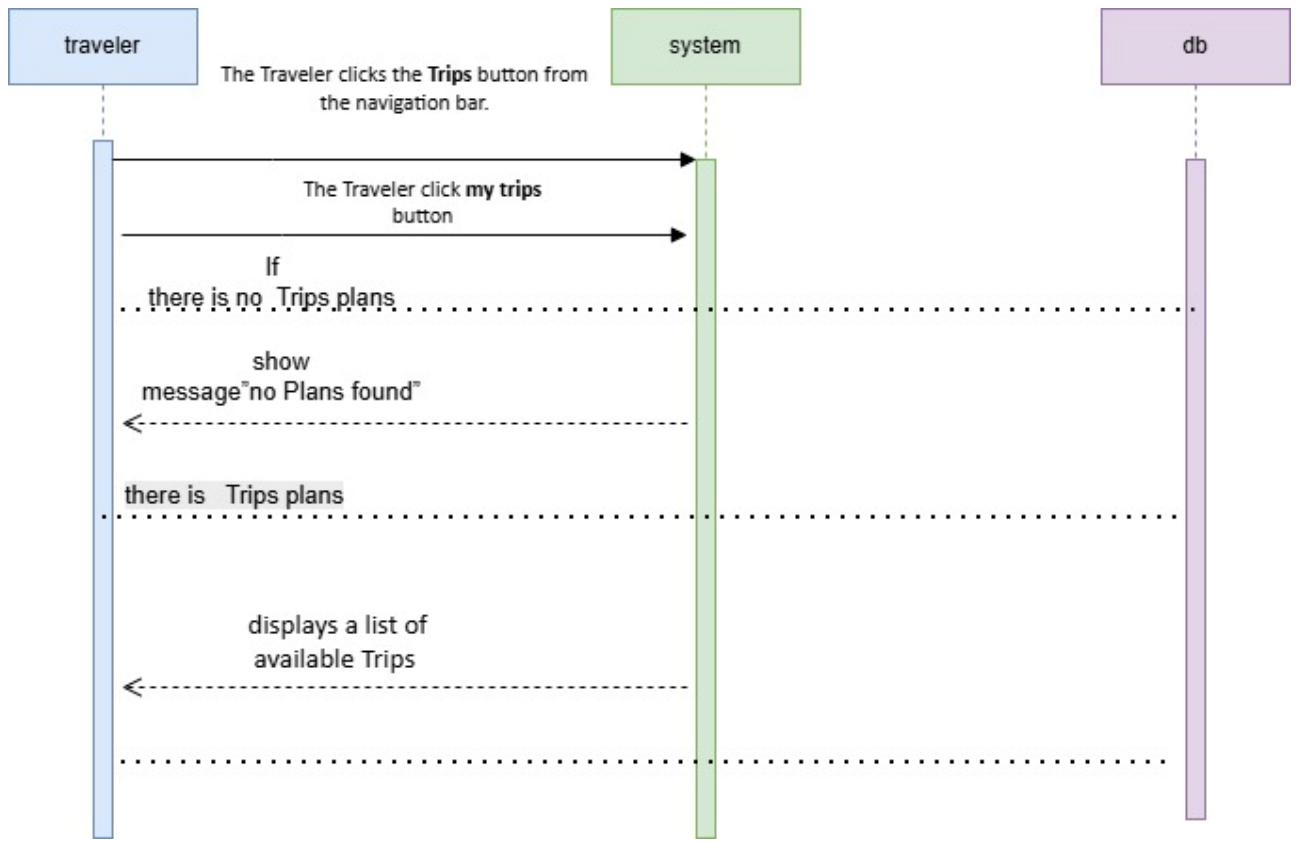


Figure 108 sequence diagram view trip plan

## ●View Trips Plan details:

<b>ID</b>	UC-51
<b>Use Case</b>	<b>View Trips Plan Details</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The user is logged in and has accessed the plans page.
<b>Post Condition</b>	The Traveler views detailed information about a selected plan.
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The Traveler clicks on a View details for the selected plan.</li> <li>2. The system displays detailed information (Duration, Travelers ,Budget, Dates , Airline, ,Flight ,From,To)with button( back to trips,prints Itinerary ))</li> </ol>

Table 53 View Trip Plan Details

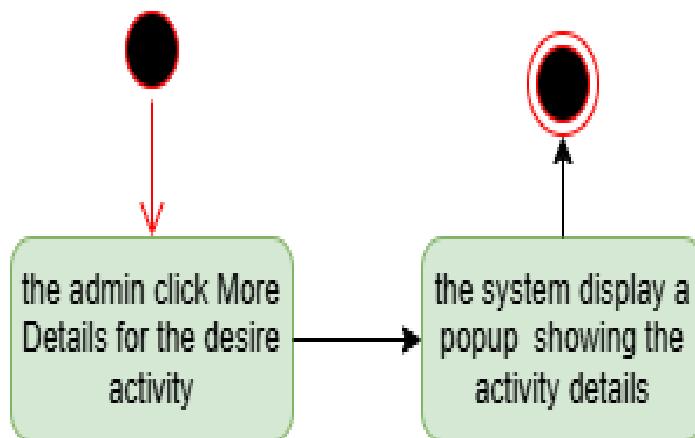


Figure 109 Activity Diagram(View Trip Plan Details)

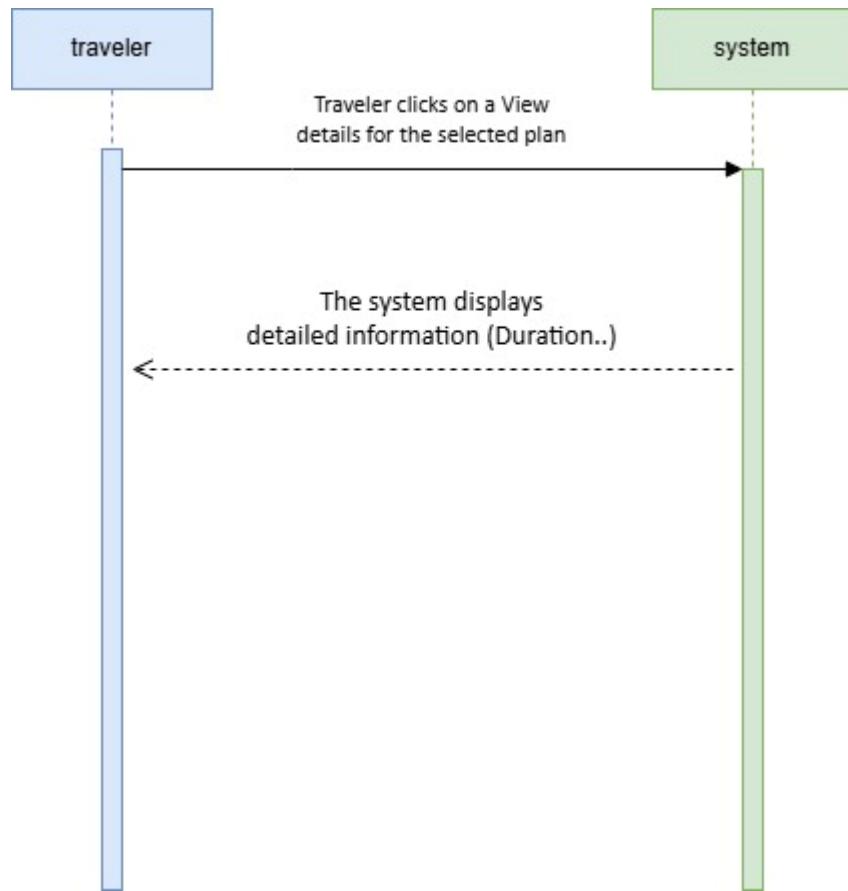


Figure 110 sequence diagram View Trips Plan details

## ● Delete Trips Plan:

<b>ID</b>	UC-52
<b>Use Case</b>	<b>Delete Trip Plan</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The Traveler is logged in and has accessed the plans page.
<b>Post Condition</b>	Trip Plan is deleted
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The traveler clicks the <b>Delete button</b></li> <li>2. The system delete the Plan</li> <li>3. The system display a success message</li> </ol>

Table 54 Delete Trip Plan

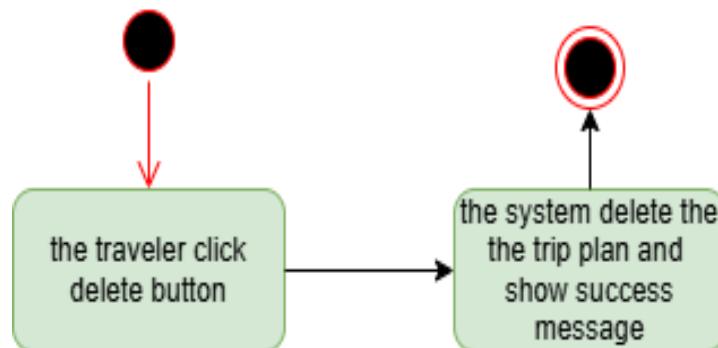


Figure 111Activity Diagram(Delete Trip Plan)

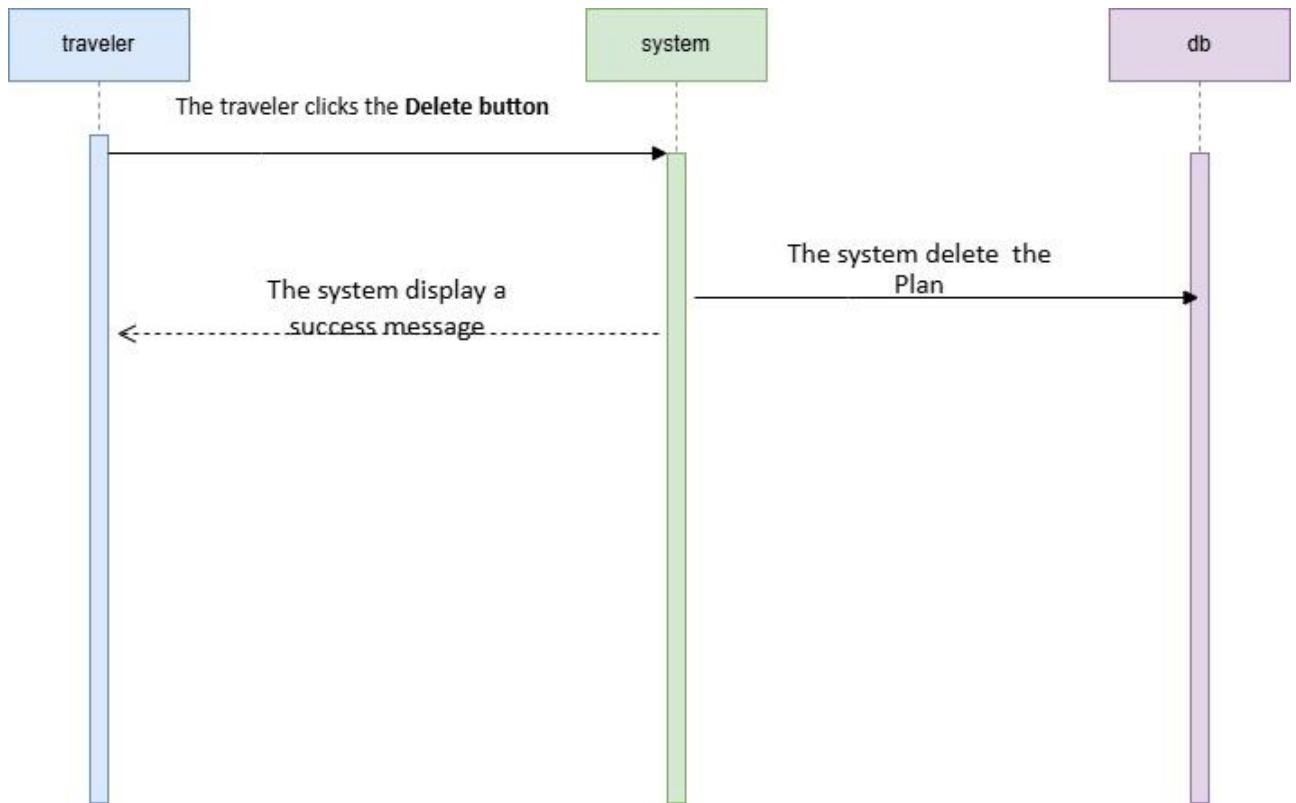


Figure 112 sequence digram Delete Trips Plan

## ● Create Manual Trip Plan:

<b>ID</b>	<b>UC-53</b>
<b>Use Case</b>	<b>Create Manual Trip Plan</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The Traveler is logged into the system
<b>Post Condition</b>	A new Trip Plan is added
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The traveler clicks the <b>Trips</b> button from the navigation bar .</li> <li>2. The traveler click <b>Create Trip</b> button.</li> <li>3. The system display ways to create plans.</li> <li>4. The traveler clicks <b>manual creation</b>.</li> <li>5. The system display multi_steps form including( Basic Trip Information, Destination selection, Hotel Assignment, Activities selection, flight information)</li> <li>6. The traveler completes multi-steps form</li> <li>7. Traveler submits the trip plan</li> <li>8. The System validates inputs</li> <li>9. The system create the trip</li> <li>10. The system redirect to create trip plan page and Show success message</li> </ol>
<b>Alternative Scenario</b>	If information is missing the system display error message .

Table 55 Create Manual Trip Plan

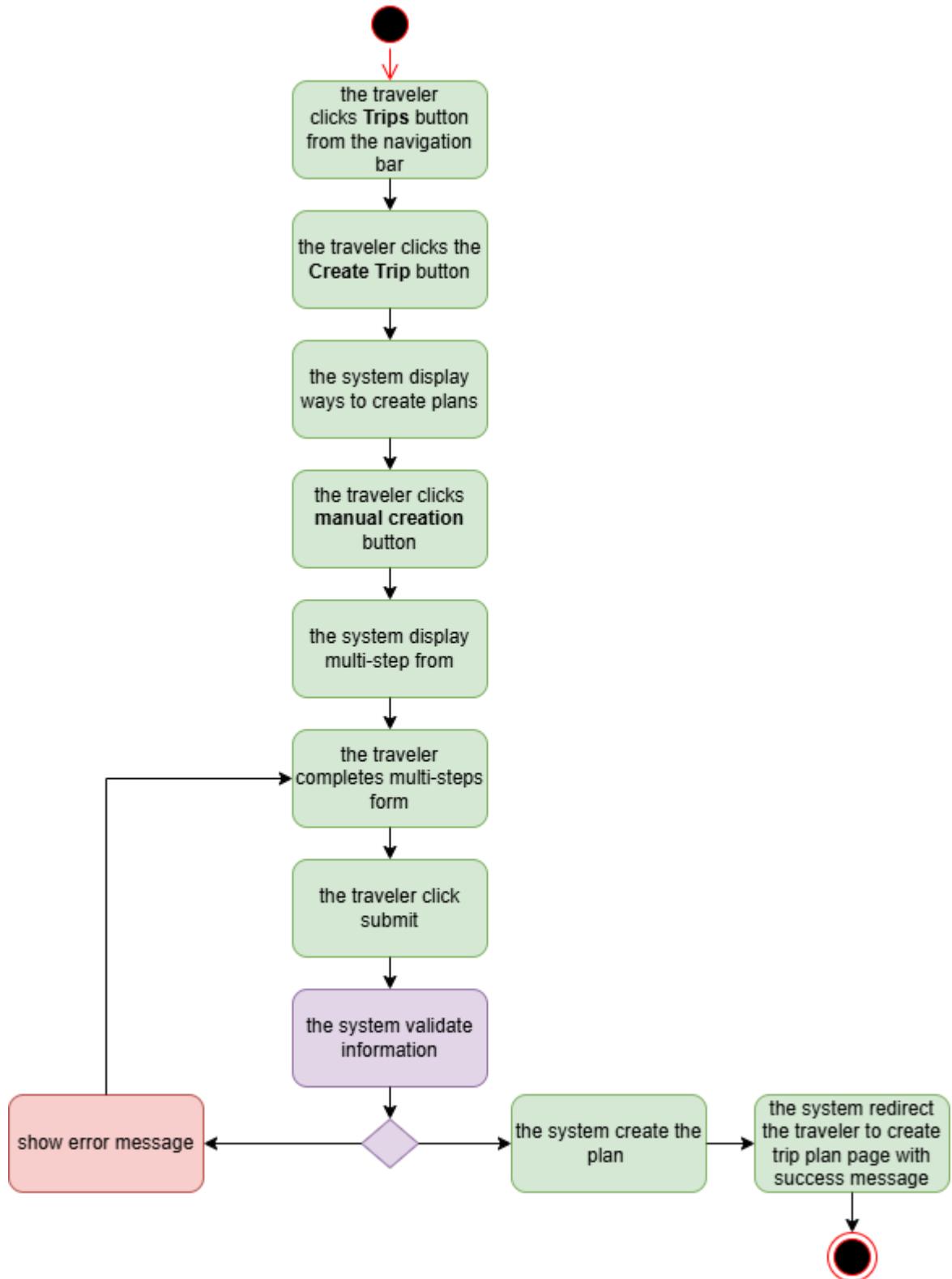


Figure 113 Activity Diagram(Create manual trip plan)

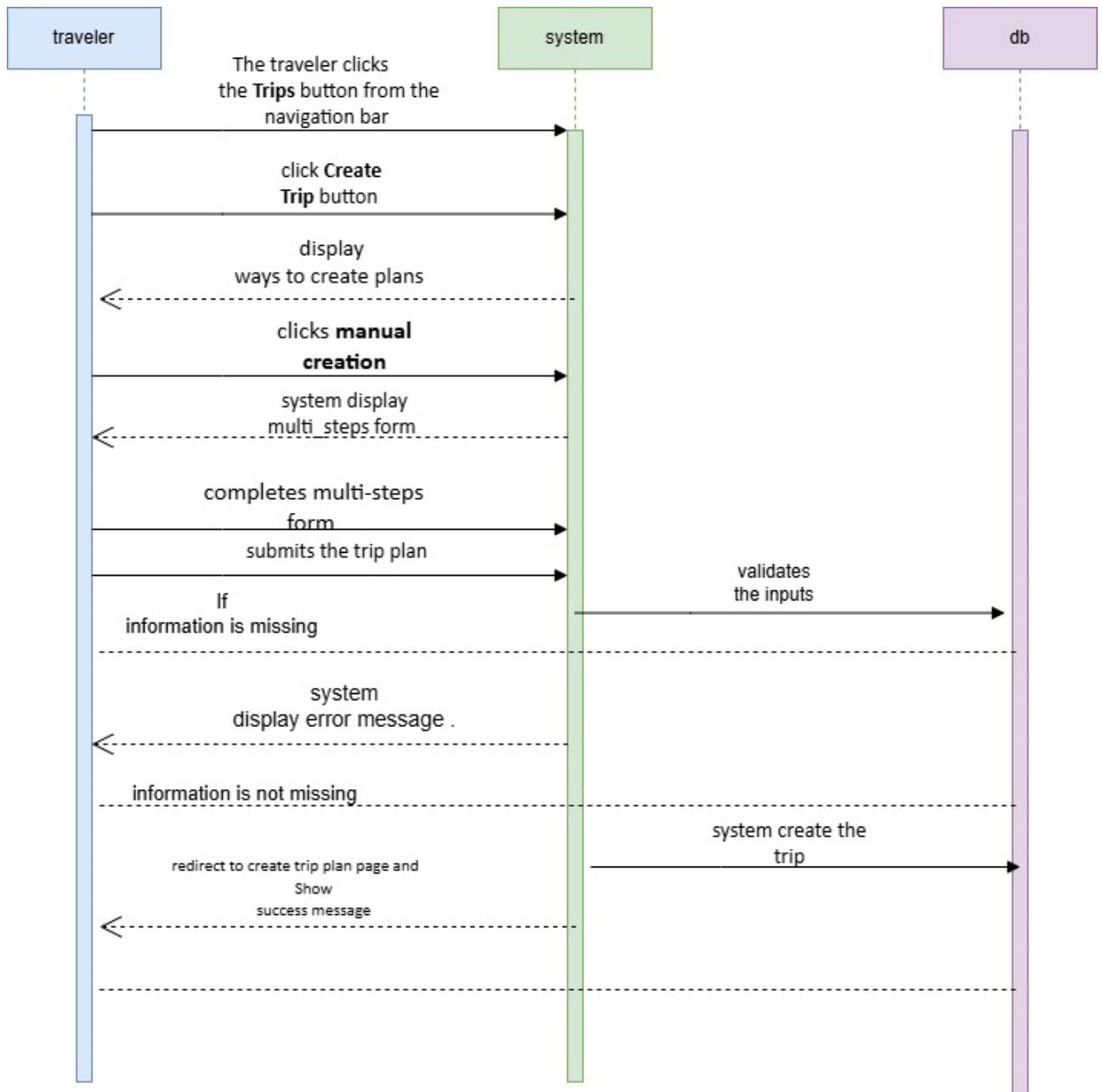


Figure 114 sequence diagram Create Manual Trip Plan

## ● Create AI Trip Plan:

<b>ID</b>	<b>UC-54</b>
<b>Use Case</b>	<b>Create AI Trip Plan</b>
<b>Actor</b>	<b>Traveler</b>
<b>Pre Condition</b>	The Traveler is logged into the system
<b>Post Condition</b>	A new Trip Plan is added
<b>Main Scenario</b>	<ol style="list-style-type: none"> <li>1. The traveler clicks the <b>Trips</b> button from the navigation bar .</li> <li>2. The traveler click <b>Create Trip</b> button.</li> <li>3. The system display ways to create plans.</li> <li>4. The traveler clicks <b>AI creation</b>.</li> <li>5. The system display form containing : (Trip Description ,Number of Travelers ,Duration ,Start Date ,End Date ,Estimated Budget,Language ) The traveler fill the field.</li> <li>6. The traveler click the <b>Generate My Trip</b> button</li> <li>7. The system validate the informaton</li> <li>8. The system sends the trip data to the AI (Groq API) to generate a travel plan.</li> <li>9. The AI returns a generated trip plan based on the provided inputs.</li> <li>10. The system redirect the traveler to the details page with success message</li> </ol>
<b>Alternative Scenario</b>	<p>Step7:If information is missing the system display error message .</p> <p>step:8 If AI fails → system displays error message</p>

Table 56 Create AI Trip Plan

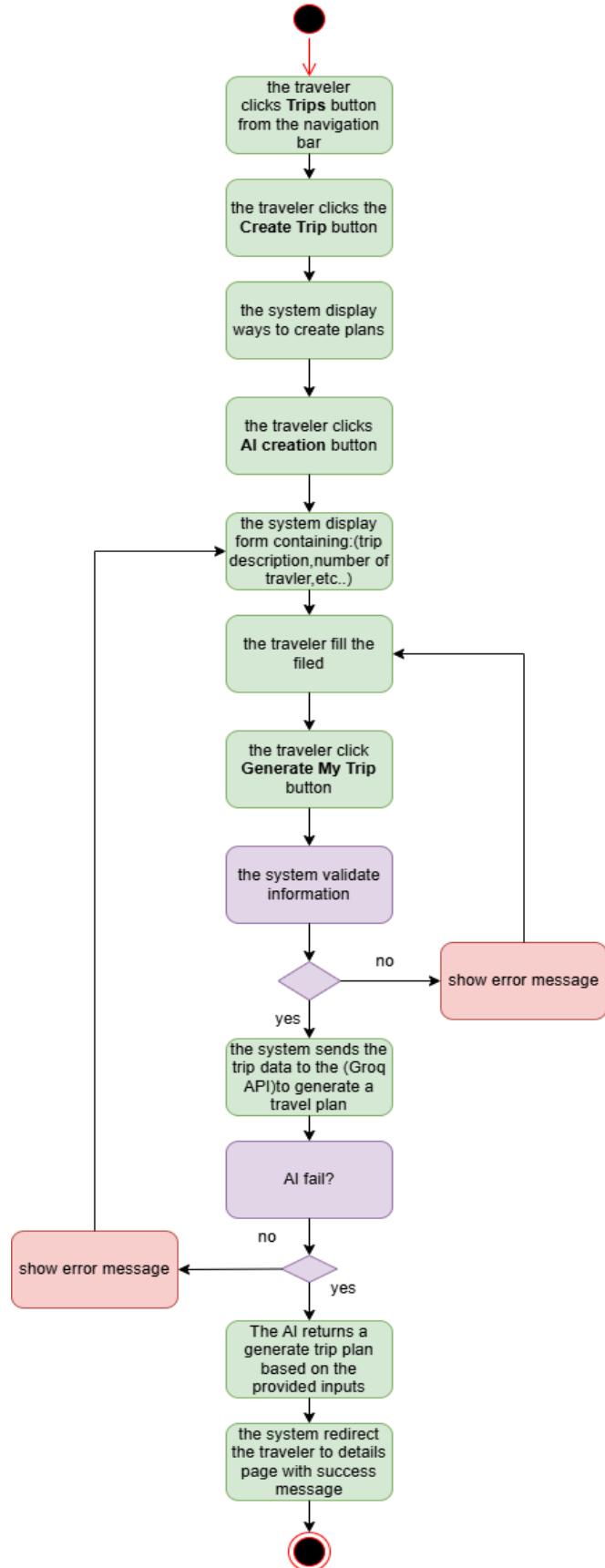


Figure 115 Acivity Diagram (Create Ai Plan)

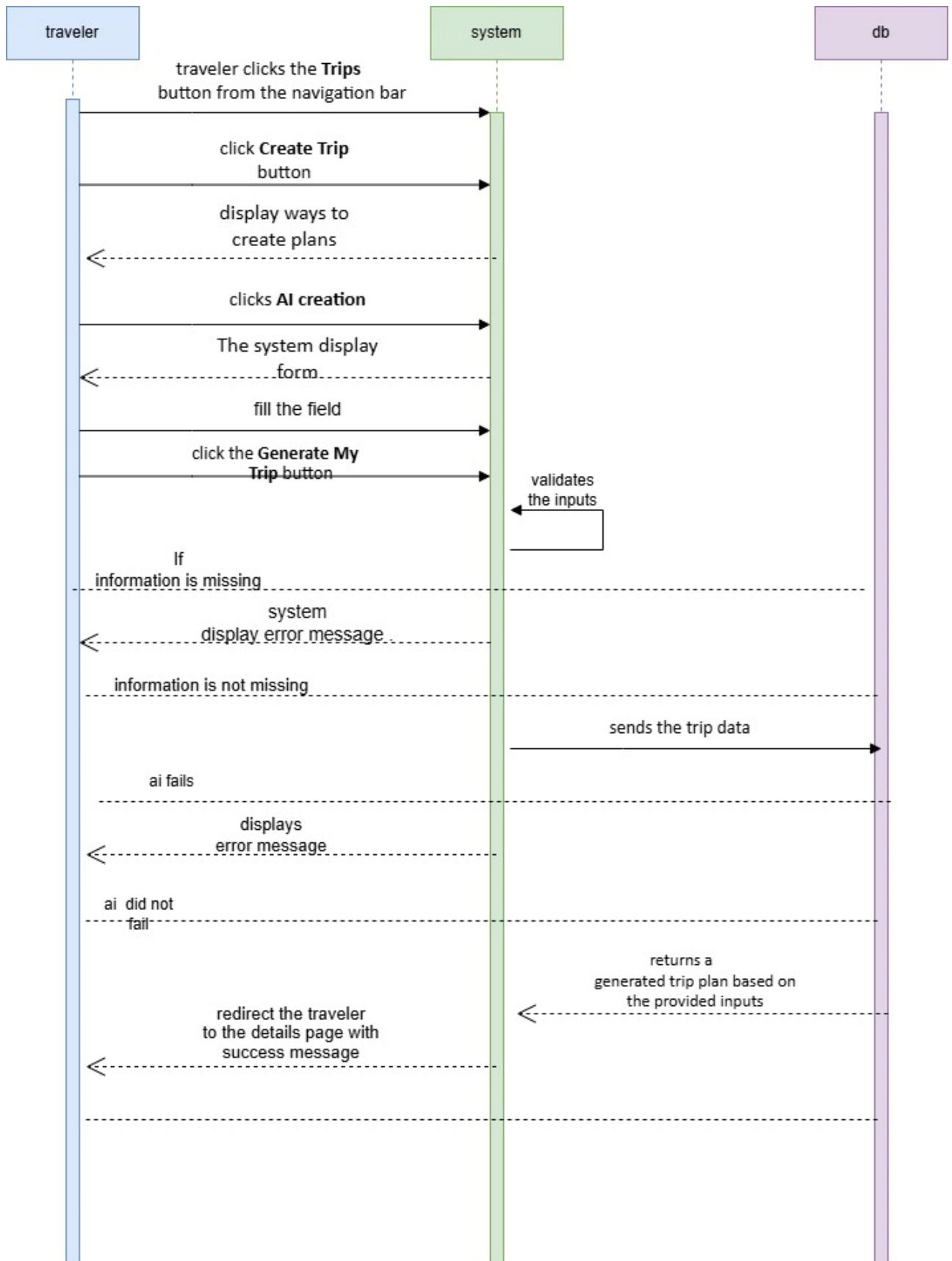


Figure 116 SEQUANCE DIAGRAM Create AI Trip Plan

# **Chapter 5 - System Design**

## **5.1 Introduction:**

This chapter presents the system design of the **Smart Tourism and Travel Planning Platform**. The system follows the **Model–View–Controller (MVC)** architectural pattern implemented using the Laravel framework. This ensures clear separation of concerns, improves maintainability, supports scalability, and simplifies future enhancements.

The MVC design organizes the system into three main parts: Model, View, and Controller, interacting to handle user requests, process data, and present information to users.

## **5.2 MVC Components:**

### **5.2.1 View (V):**

The **View** represents the user-facing part of the system where users interact with the platform through web pages.

**Implementation:** Blade templates, HTML, CSS, JavaScript, Tailwind CSS

#### **Responsibilities:**

- Display tourism destinations, hotels, flights, and weather information
- Collect user input for searches, bookings, and payments
- Render dynamic content received from controllers
- Provide real-time feedback and validation for user actions

### **5.2.2 Controller (C):**

**Controllers** manage the application logic and act as intermediaries between Views and Models.

#### **Responsibilities:**

- Handle HTTP requests and routing
- Validate user input
- Call Models and service classes
- Integrate external APIs such as PayPal, OpenWeather, OpenStreetMap, Amadeus, and Hugging Face
- Return views or JSON responses to the client

### **5.2.3 Model (M):**

The **Model** represents the core business logic and interacts with the database.

#### **Responsibilities:**

- Define entities such as Users, Destinations, Hotels, Bookings, Payments, and Flights
- Manage relationships between entities
- Implement business rules and validation logic
- Process data before saving or retrieving from the database

## **5.3 External Services Integration:**

The system integrates multiple third-party services through dedicated service classes:

- **PayPal Payment Gateway** – secure online payments
- **OpenStreetMap & Geocoding Service** – location mapping and address resolution
- **OpenWeather API** – real-time weather information
- **Amadeus API** – flight search
- **Hugging Face API** – suggesting travel plans and personalized recommendations for travelers

This keeps the MVC structure clean while allowing advanced functionalities without breaking the separation of concerns.

## **5.4 Database:**

The system uses **MySQL**, accessed through Laravel's Eloquent ORM.

### **Responsibilities:**

- Store user accounts, destinations, hotels, bookings, payments, and travel-related data
- Ensure data consistency and integrity
- Support efficient queries and relationships between entities

## 5.5 Class Diagram:

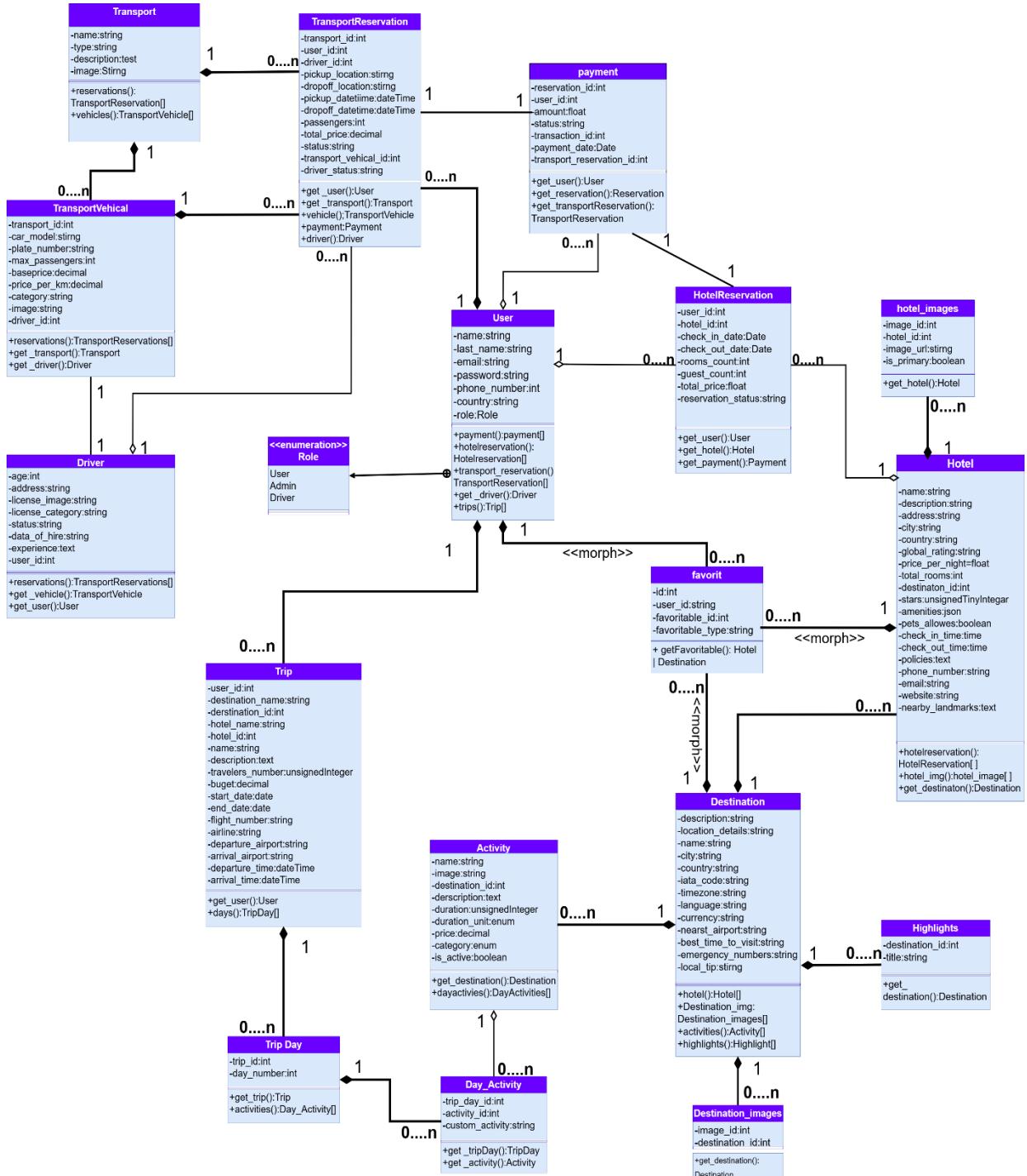


Figure 117 Class Diagram

## 5.5 ERD Diagram:

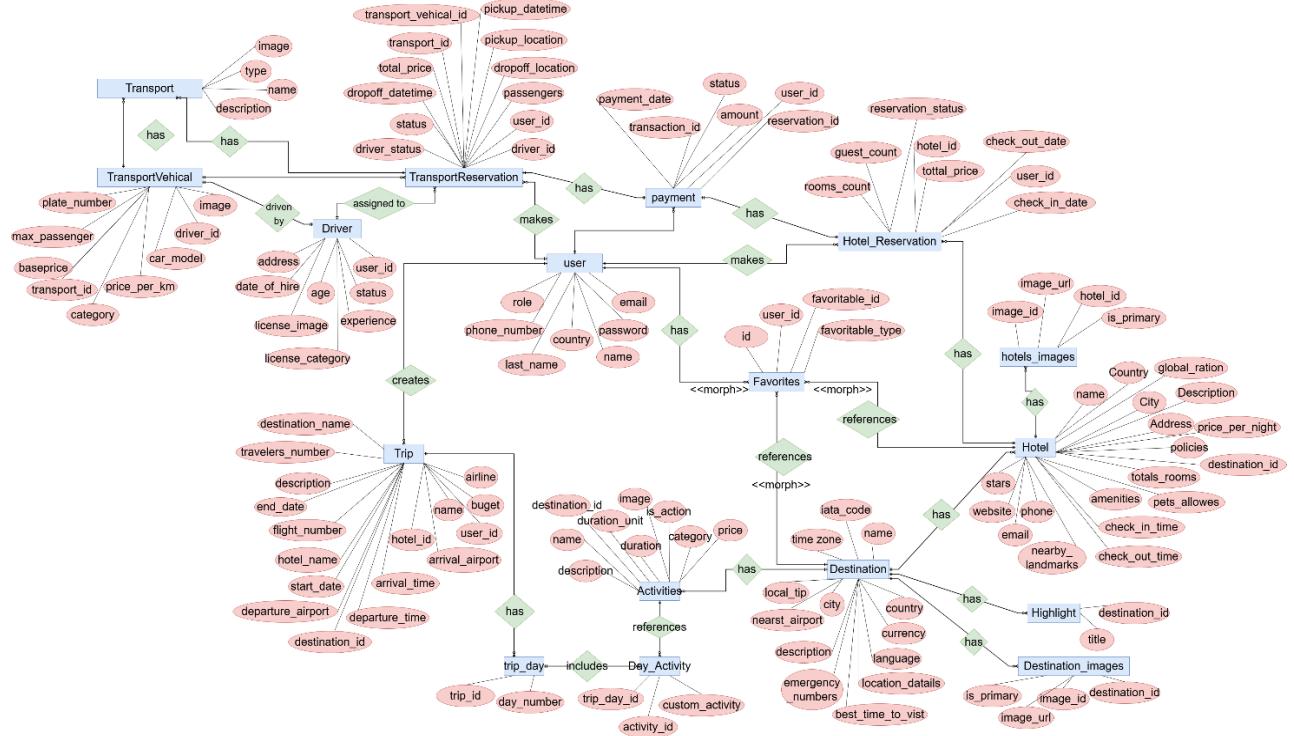


Figure 118 ERD Diagram

# **Chapter 6 - Implementation and Testing**

## **6.1 Introduction:**

This chapter presents the implementation and testing phase of the Smart Travel Planning System. It explains how the functional and non-functional requirements defined in previous chapters were translated into a fully working web application. The chapter focuses on the technologies used, system architecture, module implementation, third-party service integration, and the testing process carried out to ensure system correctness, usability, performance, and security. The objective of this chapter is to demonstrate that the proposed system has been implemented according to the specifications and validated through comprehensive testing.

## **6.2 Technologies Used:**

### **6.2.1 Laravel Framework:**

Laravel is a powerful PHP web application framework that follows the Model-View-Controller (MVC) architectural pattern. It provides a clean and expressive syntax that simplifies common development tasks such as routing, authentication, session management, and database interaction. In this project, Laravel is used as the core backend framework to handle business logic, user authentication, role-based access control, booking management, and API integrations. Laravel's built-in security features, such as CSRF protection and password hashing, contribute significantly to the overall security of the system.

### **6.2.2 HTML, CSS, and JavaScript:**

HTML, CSS, and JavaScript form the foundation of the frontend layer of the system. HTML is used to structure the web pages, CSS is responsible for layout and visual styling, and JavaScript enables dynamic behavior and client-side interactions. These technologies work together to deliver a responsive and interactive user experience, allowing users to browse destinations, search services, submit forms, and interact with the system efficiently.

### **6.2.3 Tailwind CSS:**

Tailwind CSS is a utility-first CSS framework that allows developers to build modern and responsive user interfaces directly within HTML using predefined utility classes. In this project, Tailwind CSS is used to design clean, consistent, and responsive layouts across all system pages. It improves development speed and ensures visual consistency for dashboards, forms, tables, and user interfaces for travellers, drivers, and administrators.

### **6.2.4 PayPal Payment Gateway:**

PayPal is a widely used and secure online payment gateway that enables users to complete transactions safely. The system integrates PayPal to handle hotel booking and transport reservation payments. This integration ensures secure payment processing, protects sensitive financial data, and provides users with a familiar and trusted payment experience. Successful and failed payment scenarios are handled appropriately and reflected in the booking status within the system.

### **6.2.5 OpenStreetMap and Geocoding Service:**

OpenStreetMap is an open-source mapping platform used to display geographical locations on interactive maps. The system integrates OpenStreetMap along with a geocoding service to convert destination addresses and location names into geographic coordinates. This allows users to view

destinations, hotels, and activities on maps, improving navigation and enhancing the overall travel planning experience.

### **6.2.6 OpenWeather API:**

The OpenWeather API is used to retrieve real-time and forecast weather information for selected destinations. This feature allows travellers to view weather conditions while planning their trips, helping them make informed decisions. Weather data is displayed in a user-friendly format and dynamically updated based on the selected location.

### **6.2.7 Amadeus API for Flight Searching:**

The Amadeus API is a global travel technology platform used in this system to search for flights. It allows users to view available flights based on travel criteria such as departure location, destination, and travel dates. The integration provides real-time flight data retrieved from external services and displays it within the system without handling flight bookings directly.

### **6.2.8 Groq API for AI Trip Planning:**

The system integrates the **Groq API** to provide AI-powered travel planning suggestions for travellers. By leveraging large language models accessed through the Groq platform, the system is able to generate travel planning recommendations based on user inputs such as destination, travel dates, interests, and budget.

This functionality enables travellers to receive automated suggestions for trip itineraries, activities, and general travel planning guidance.

The integration enhances the overall user experience by supporting intelligent decision-making and reducing the manual effort required during the trip planning process.

### **6.2.9 MySQL Database:**

MySQL is a relational database management system used to store and manage system data. It stores user accounts, destinations, hotels, vehicles, bookings, payments, favorites, and travel plans. The structured nature of MySQL ensures data integrity, consistency, and efficient querying, making it suitable for managing the complex relationships between system entities.

## **6.3 Implementation Overview:**

The system is implemented following a modular architecture where each major functionality is developed as a separate module. These modules include user account management, destination management, hotel and transport booking, travel planning, payment processing, notifications, and administrative management. Laravel controllers manage application logic, models handle database interaction, and Blade templates combined with Tailwind CSS render the user interface.

## **6.4 Testing Strategy:**

Testing was conducted to verify that each system feature works as expected and meets the specified requirements. Test cases were designed based on the functional requirements and executed manually. The testing process includes authentication testing, booking workflows, payment validation, search and filtering operations, notification delivery, and admin operations. Both valid and invalid scenarios were tested to ensure proper error handling and system robustness.

## 6.5 System Interfaces:

- Login:

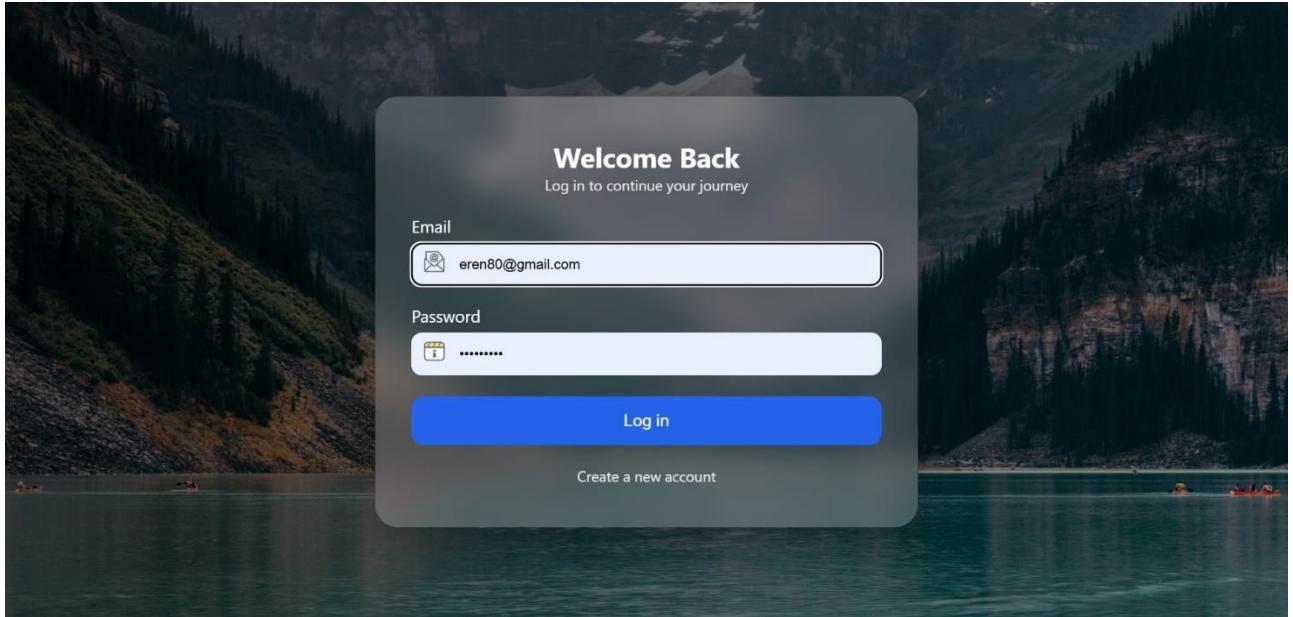


Figure 119login (interface)

-Select Role:

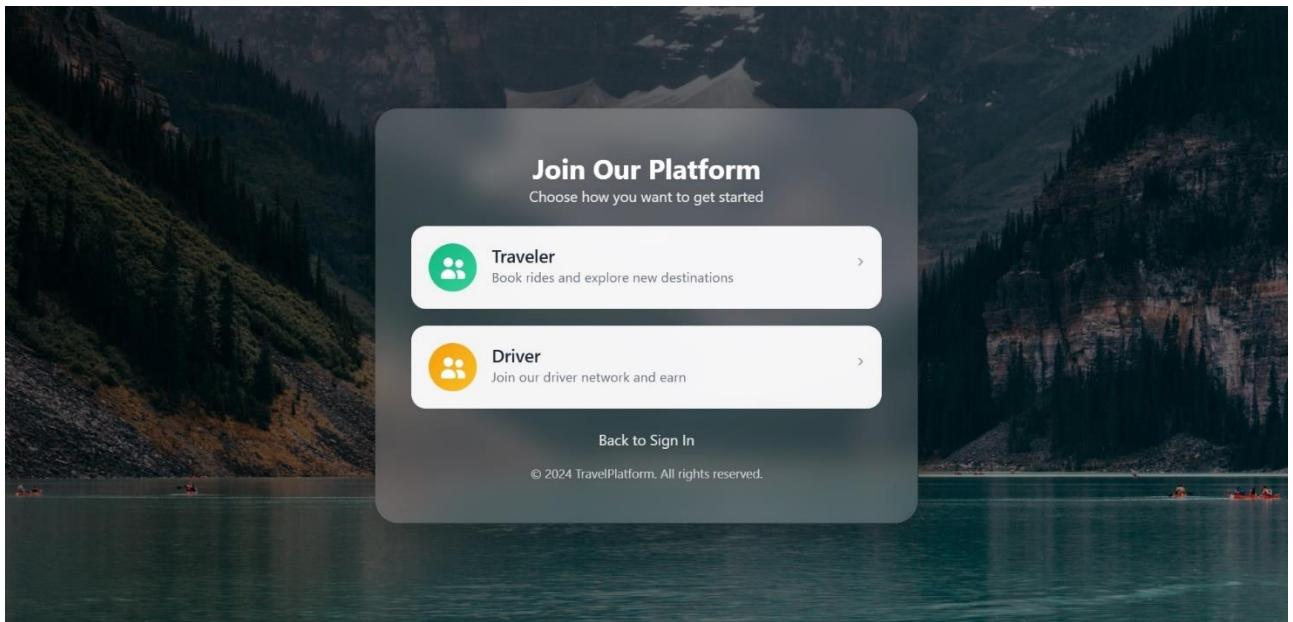


Figure 120 select role page

**-Register Traveler:**

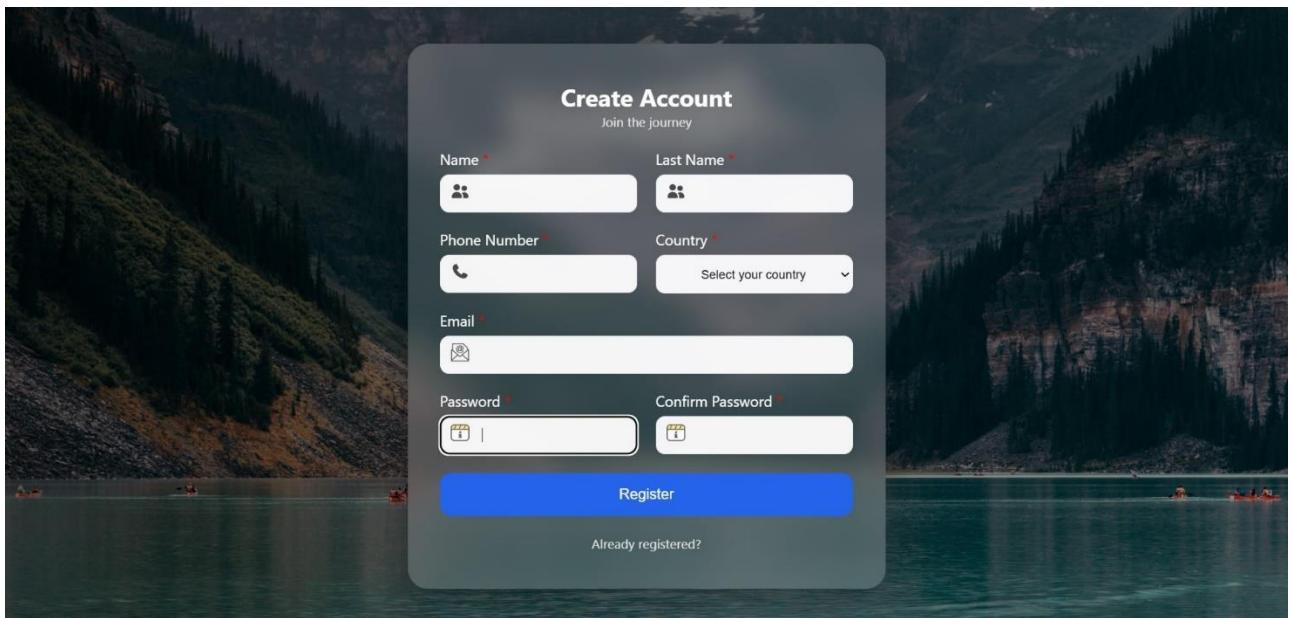


Figure 121 register traveler page

**-Register Driver – step1:**

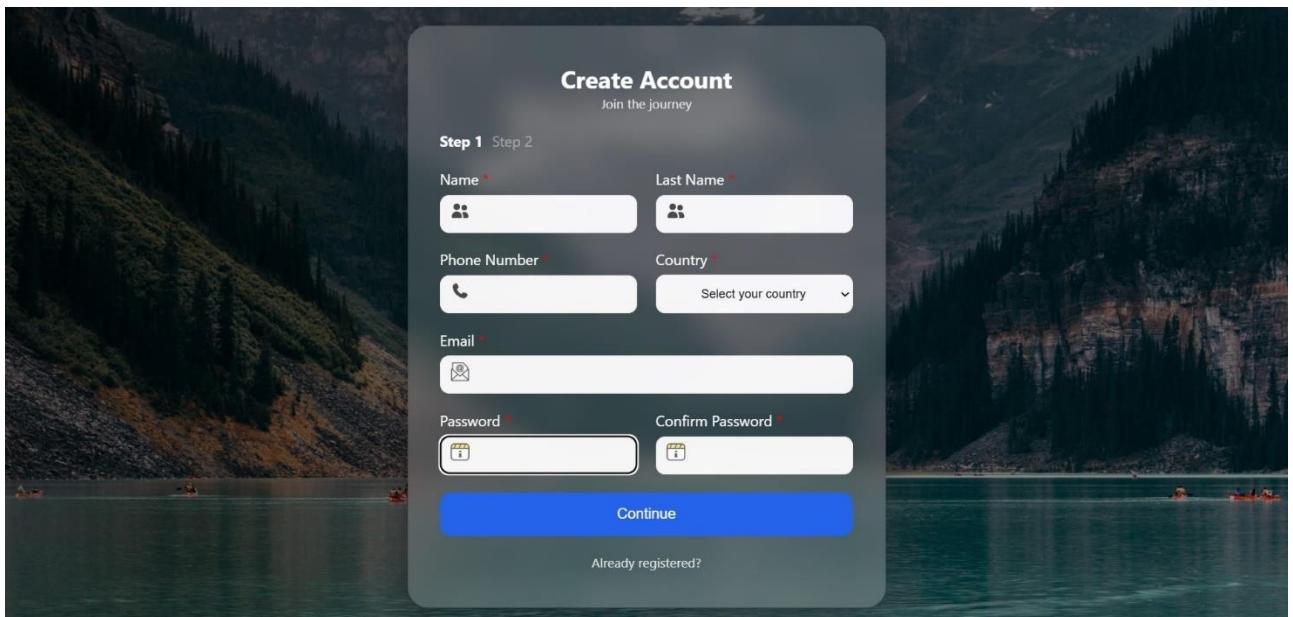


Figure 122 register driver step1

**-Register Driver – step2:**

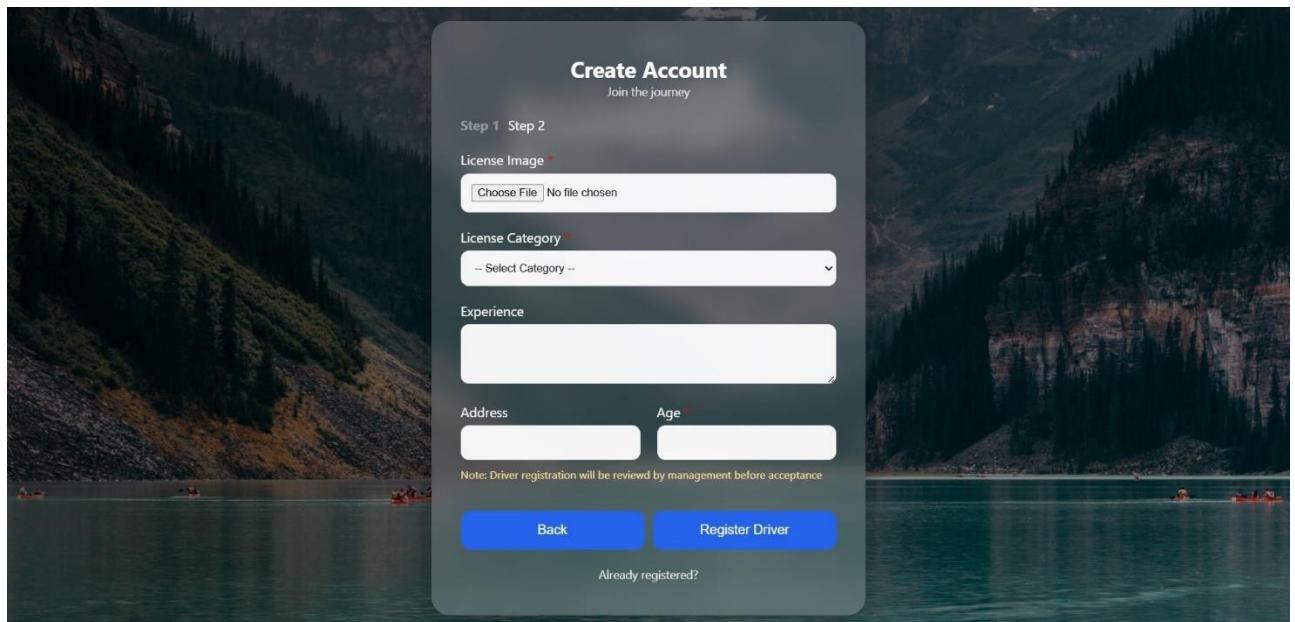


Figure 123 Register Driver – step2

**- Main Dashboard:**

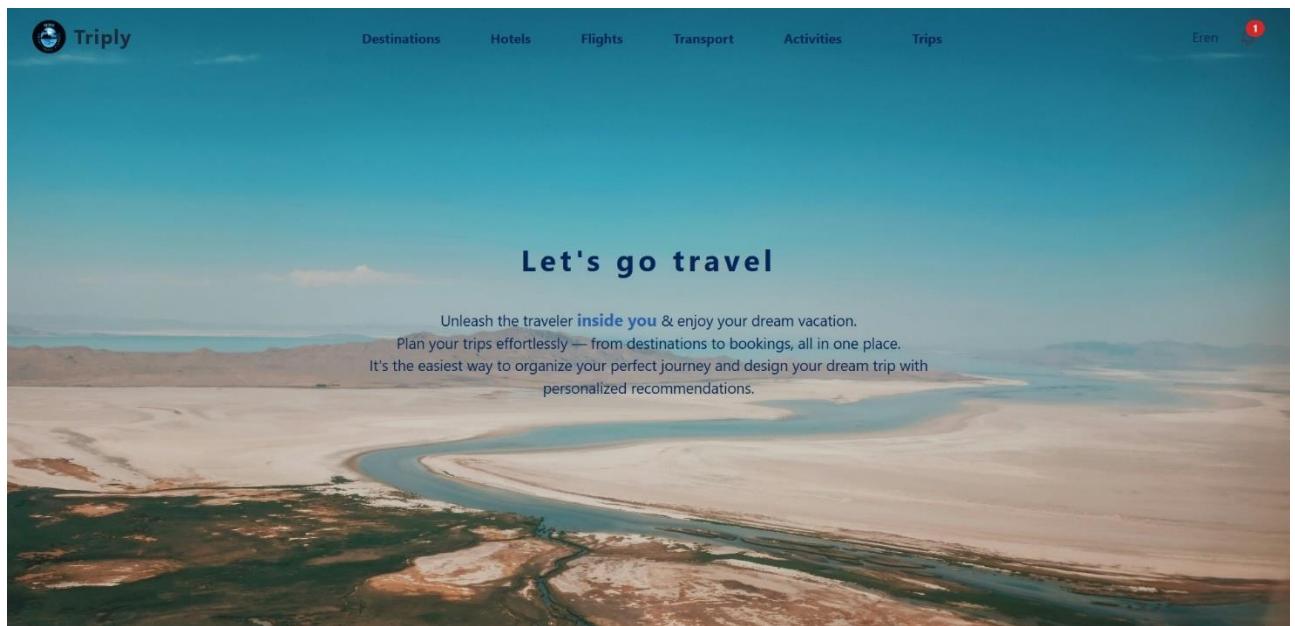
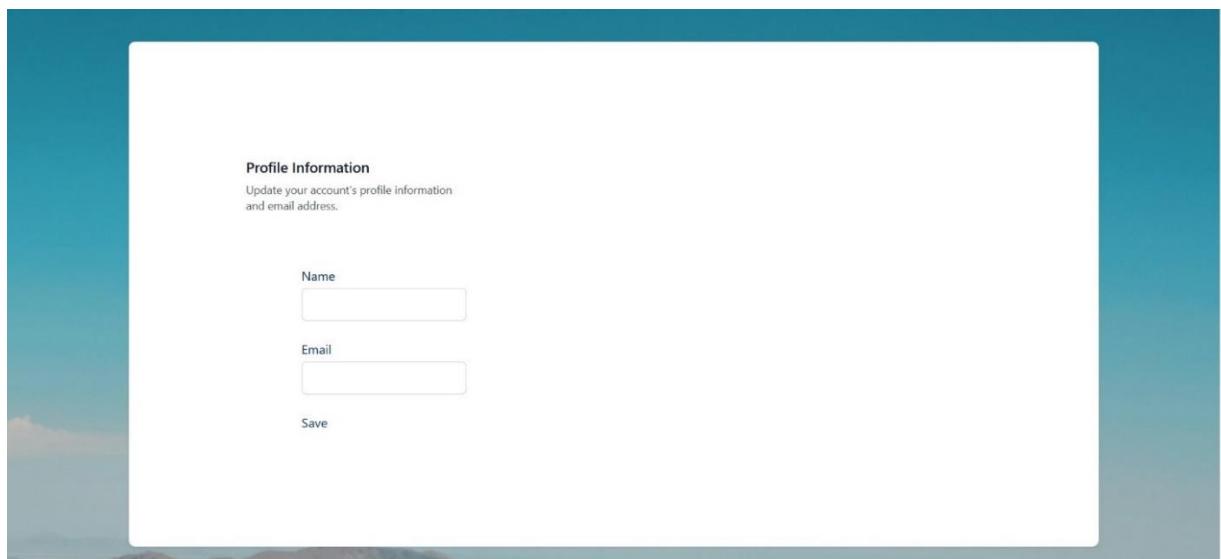
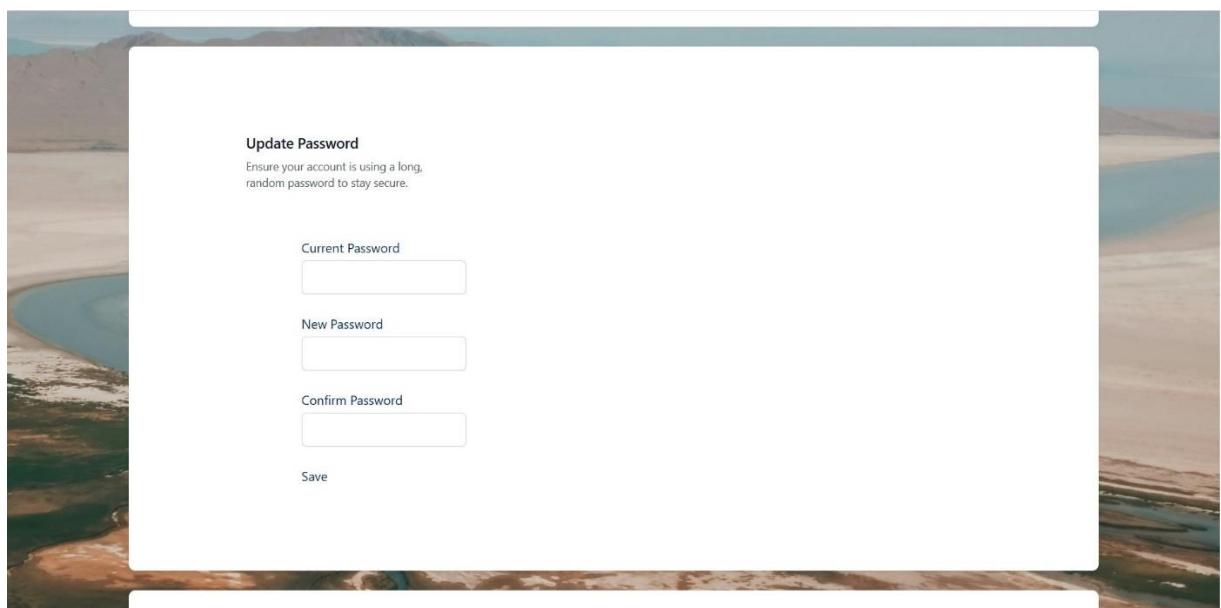


Figure 124 Main Dashboard

**- Profile Management:**



*Figure 125 profile management*



*Figure 126 profile management*

# Traveler Interfaces:

## - Destinations:

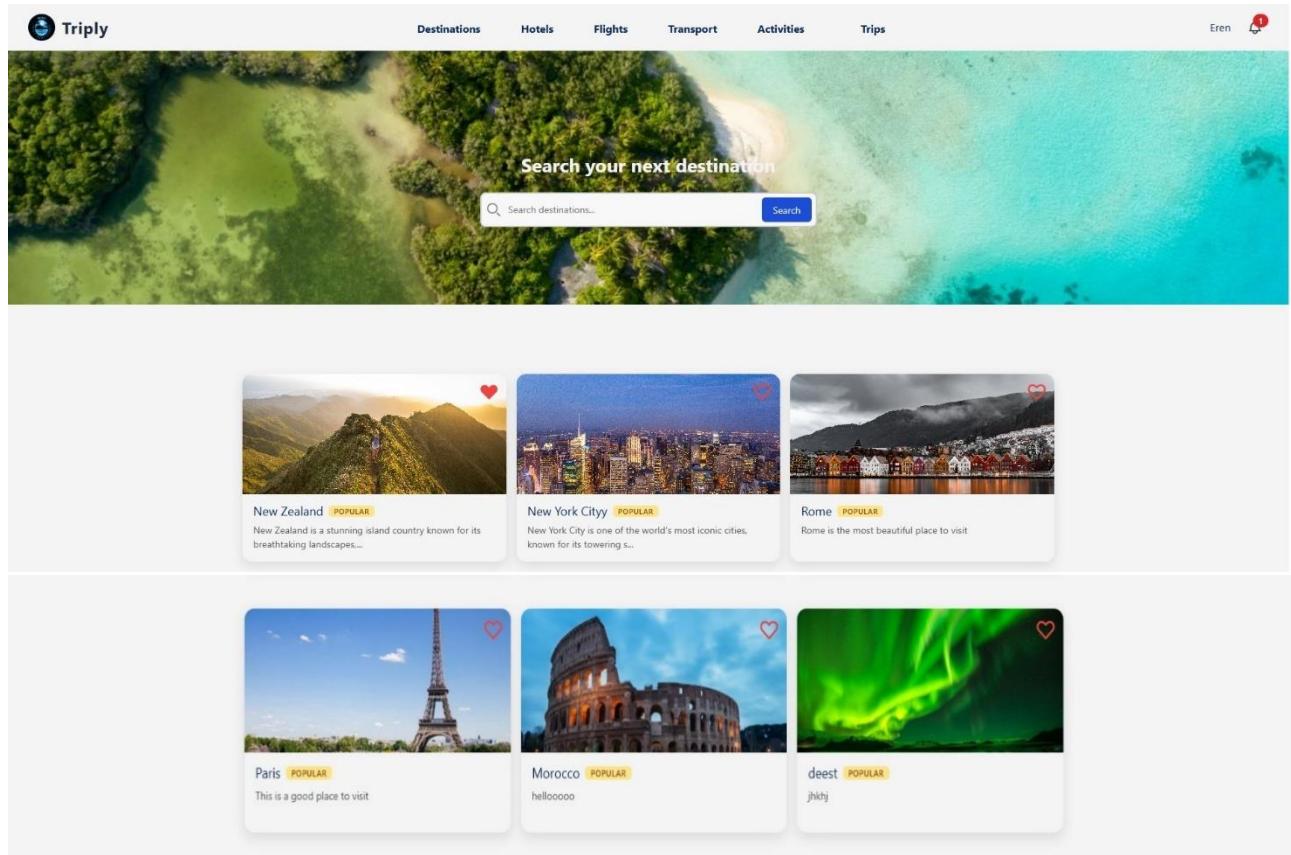


Figure 1 YY Traveler Interfaces-destinations

## -Destination Details:

The screenshot displays the 'Destination Details' interface for New Zealand. At the top, a large banner features a scenic mountain landscape at sunset with the title 'New Zealand'. Below the banner, a brief description states: 'New Zealand is a stunning island country known for its breathtaking landscapes, including mountains, beaches, lakes, and lush forests. It offers a perfect mix of adventure, nature, and rich Maori culture, making it a top destination for travelers seeking unique experiences.' A location pin indicates 'Auckland, New Zealand' and a review count of '4.9 (12487 reviews)'. The main content area is titled 'Must-See Highlights' and includes three categories: 'Dramatic landscapes', 'Adventure sports', and 'The filming location of The Lord of the Rings'. Below this, the 'Unforgettable Experiences' section lists three activities with images and 'More Details' buttons: 'Auckland Coastal Walk', 'Hobbiton Movie Set', and 'Skydiving in Queenstown'. The 'Essential Information' section provides travel details like timezone, language, currency, and weather, along with tips for getting around and local advice. The 'Photo Gallery' section shows four scenic images: a mountain peak at sunset, a turquoise lake, hikers on a trail, and an airplane flying over clouds.

# New Zealand

New Zealand is a stunning island country known for its breathtaking landscapes, including mountains, beaches, lakes, and lush forests. It offers a perfect mix of adventure, nature, and rich Maori culture, making it a top destination for travelers seeking unique experiences.

Auckland, New Zealand

4.9 (12487 reviews)

## Must-See Highlights

Dramatic landscapes Adventure sports The filming location of The Lord of the Rings

## Unforgettable Experiences

The absolute essentials that define this destination

A person walking along a coastal path through a field of white flowers.

**Auckland Coastal Walk**  
A relaxing guided walk along Auckland's coastline, offering scenic ocean views and fresh air, perfect for nature lovers.

[More Details](#)

A person sitting on a bench in front of a movie set.

**Hobbiton Movie Set**  
Visit the iconic Hobbiton movie set and explore the magical village featured in The Lord of the Rings and The Hobbit films.

[More Details](#)

Two people skydiving against a clear blue sky.

**Skydiving in Queenstown**  
Experience an adrenaline-filled skydiving adventure over Queenstown, offering breathtaking views of mountains, lakes, and valleys while free-falling from high altitude with a professional instructor.

[More Details](#)

## Essential Information

Everything you need to know before you go

### Location Details

TIMEZONE	LANGUAGE	CURRENCY	WEATHER
New Zealand Standard Time (NZST), UTC +12	English, Maori	New Zealand Dollar (NZD)	<a href="#">10-Day Forecast</a>

### Getting Around

NEAREST AIRPORT  
Auckland International Airport

PUBLIC TRANSPORT  
[Click here](#) to view available transport options.

### Travel Tips & Essentials

BEST TIME TO VISIT  
October to April, when the weather is mild and ideal for outdoor activities and sightseeing.

EMERGENCY NUMBERS  
111 (Police, Fire, Ambulance)

LOCAL TIP  
Renting a car is highly recommended to fully explore New Zealand's scenic routes and hidden natural attractions.

## Photo Gallery

Discover the beauty of New Zealand through stunning imagery

A person standing on a grassy mountain ridge at sunset.

A person sitting on a rock overlooking a deep turquoise lake surrounded by mountains.

Two people hiking up a rocky trail in a mountainous area.

An airplane flying high above a layer of white clouds.

Figure 128 Destination Details interface

## -Hotels:

The screenshot displays the Triply travel website interface. At the top, there is a navigation bar with links for Destinations, Hotels, Flights, Transport, Activities, and Trips. A user profile icon for 'aya222' is also present. Below the navigation bar, a banner features a hallway with framed pictures and a search bar with the placeholder 'Search hotels...'. A prominent button labeled 'Find Your Hotel' is centered. The main content area shows several hotel cards, each with a thumbnail image, the hotel name, a user ID, and a 'Book' button. The cards include:

- Rotana** (User ID: dtgujl) - Shows an ornate lobby.
- Samira Mes** (User ID: huoh) - Shows a modern building with palm trees.
- Hilton** (User ID: dkajidjiojeqid) - Shows an outdoor terrace with tables and chairs.
- The Four Season** (User ID: 350 5th Ave, New York, NY 10118) - Shows a rooftop lounge at sunset.
- The Wall Street Hotel** (User ID: 750 5th Ave) - Shows the exterior of a building at night.

Figure 129 Traveller interfaces Hotels

## -Hotel Details:

Destinations Hotels Flights Transport Activities Trips



The Wall Street Hotel  
A luxurious hotel with a modern touch in a historic building (Tontine Building) that blends boutique chic with contemporary design. The rooms offer a "residential" atmosphere with artworks and high-quality linens. The location is excellent, close to the World Trade Center, museums, the sea, and the Bridge Tower.

New York, United States GLOBAL RATING: 4 \*



Must-See Amenities  
The absolute essentials that defines this Hotel

 Wifi  Parking  Pool  Spa  Restaurant  Gym  Laundry

## Essential Information

Everything you need to know before you book

 General Info

 ADDRESS  
750 5th Ave

 Nearby landmarks  
One World Observatory & 9/11 Memorial

**Ratings & Rooms**

**STARS**  
★★★★★

**PRICE PER TIME**  
\$500.00

**GLOBAL RATING**  
4

**TOTAL ROOMS**  
180

---

**Policies & Pets**

**PETS**  
allowed

**POLICIES**  
You must notify the hotel at least 48 hours before the arrival date if you wish to cancel.

**CHECK IN TIME**  
03:00 AM

**CHECK OUT TIME**  
11:00 AM

---

**Contact & Website**

**PHONE NUMBER**  
0952585

**EMAIL**  
mailto:reservations@thewallsthotel.com

**WEBSITE**  
[https://www.thewallsthotel.com/?utm\\_source=chatgpt.com](https://www.thewallsthotel.com/?utm_source=chatgpt.com)

---

## Photo Gallery

Discover The Wall Street Hotel hotel through stunning imagery



The photo gallery displays six images: 1. Exterior view of the hotel entrance at night. 2. A rooftop lounge area with couches and tables overlooking a city skyline. 3. The hotel's exterior facade at night, illuminated by warm lights. 4. An interior lounge area with large windows, patterned curtains, and comfortable seating. 5. A long, modern hallway with potted plants and bright lighting. 6. A spacious room with a large window, a flat-screen TV, and a sofa.

Figure 130 Hotel Details interface

## -Book Hotel:

Make a Reservation at Southern Coast Hotel

Room price per night in this hotel is 50

Number of Guests

Number of Rooms

Check-In Date  
 mm/dd/yyyy

Check-Out Date  
 mm/dd/yyyy

First Name  
 Eren

Last Name  
 yeager

Country  
 Armenia

Email  
 eren80@gmail.com

Phone Number  
 6589985555

Figure 131 book hotel interface

## -Flights:

 Triply Destinations Hotels Flights Transport Activities Trips

Fly Anywhere  
Fly Budget and Comfortably

Enter your details

\*Fill the Return field only if your trip type is round-trip

Where from? Where to? Departure Return seats Trip Type

select country  select country  mm/dd/yyyy   mm/dd/yyyy  1  One-way-trip

Figure 132 flights interface

## -Flights Results:

The screenshot displays a search results page for flights from JFK to LAX. The results are organized into a grid of cards, each representing a different flight option. The columns represent the carrier, flight number, departure time, arrival time, and price. The flights listed are all operated by carrier B6 and have a price of \$116.52.

Outbound Flights	
CARRIER: B6 From: JFK at 2025-10-07T06:00:00 To: LAX at 2025-10-07T08:50:00 Price: \$116.52	Flight #: 123 CARRIER: B6 From: JFK at 2025-10-07T07:00:00 To: LAX at 2025-10-07T09:55:00 Price: \$116.52
CARRIER: B6 From: JFK at 2025-10-07T11:00:00 To: LAX at 2025-10-07T13:58:00 Price: \$116.52	Flight #: 423 CARRIER: B6 From: JFK at 2025-10-07T08:00:00 To: LAX at 2025-10-07T10:59:00 Price: \$116.52
CARRIER: B6 From: JFK at 2025-10-07T09:00:00 To: LAX at 2025-10-07T12:01:00 Price: \$116.52	Flight #: 323 CARRIER: B6 From: JFK at 2025-10-07T21:30:00 To: LAX at 2025-10-08T00:34:00 Price: \$116.52
CARRIER: B6 From: JFK at 2025-10-07T12:00:00	Flight #: 623 CARRIER: B6 From: JFK at 2025-10-07T10:00:00 Flight #: 1123

Figure 133 flights results interface

## -Activities:

The screenshot shows the activities section of a travel website. At the top, there is a search bar with the placeholder "Search by activity or destination" and a "Search" button. Below the search bar are several filter options: Availability, Difficulty Level, Guide Language, Requires Booking, Family Friendly, Pets Allowed, Category, Filter, and Reset. The main area features a large, scenic image of a man sitting on a rocky outcrop overlooking a turquoise lake and mountains. Below this image, four activity thumbnails are displayed: "Auckland Coastal Walk", "Hobbiton Movie Set", "Park Tour", and "Skydiving in Queenstown". Each thumbnail includes a small image, the activity name, and a "More Details" button.

Figure 134 activities interface

### -Activity Details:

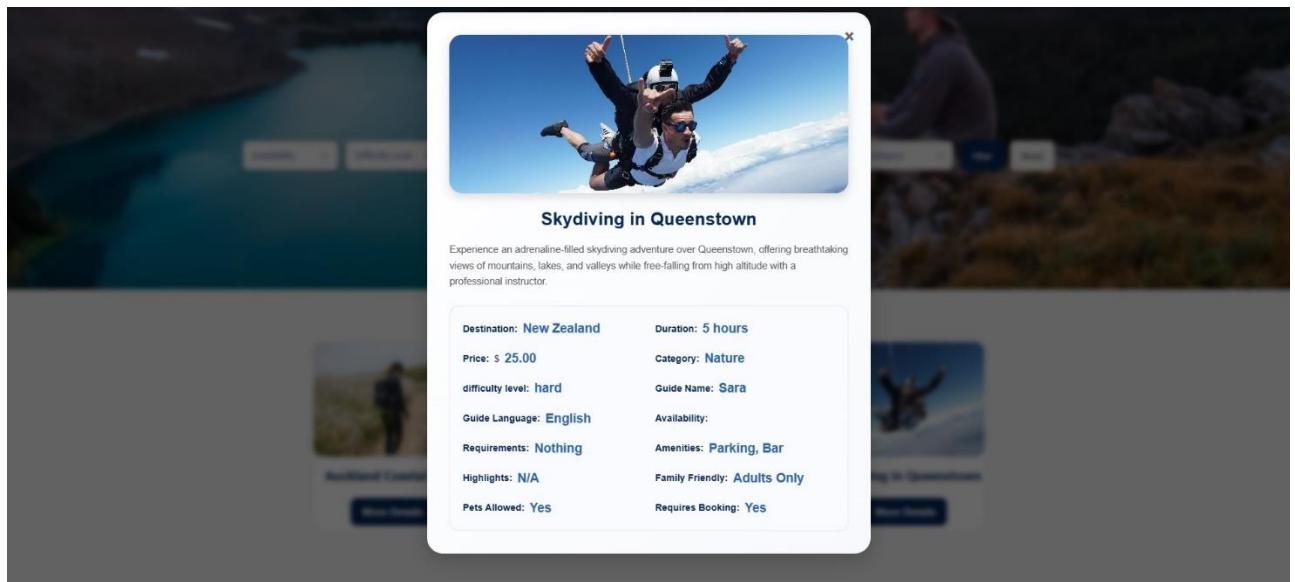


Figure 135 Activity Details interface

### -Transport Services:

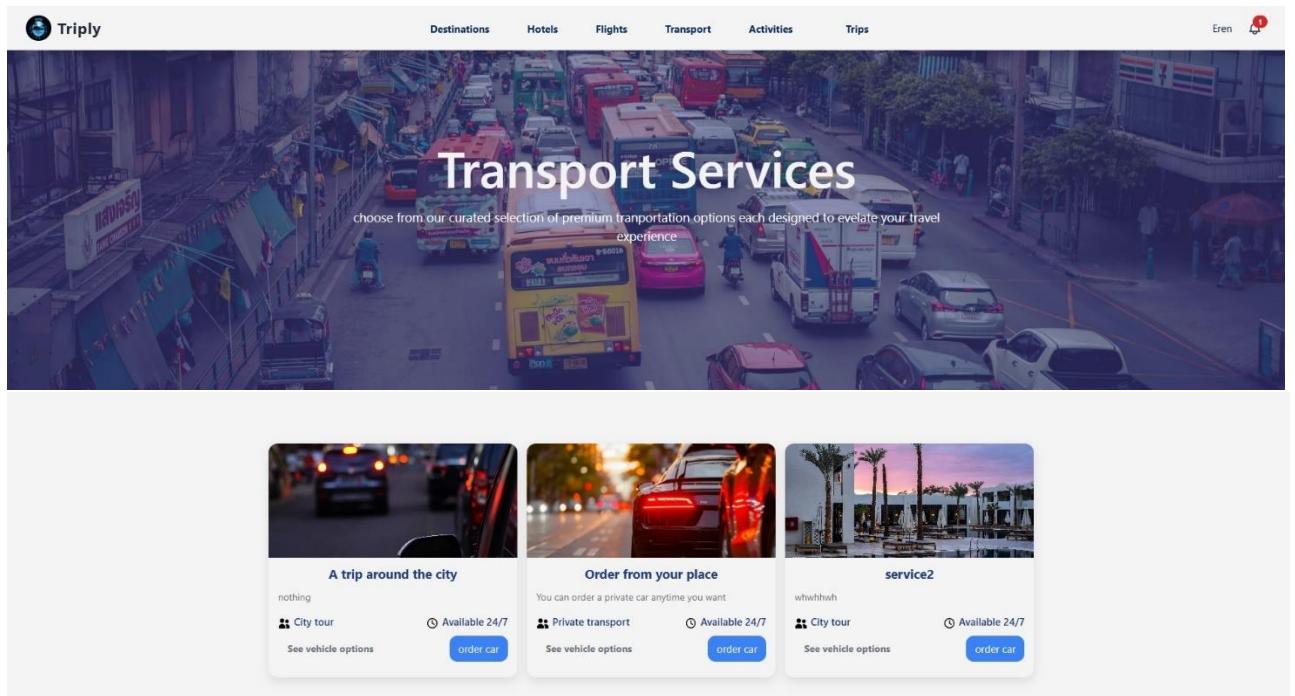


Figure 136 Transport Services interface

**-Order Vehicle:**

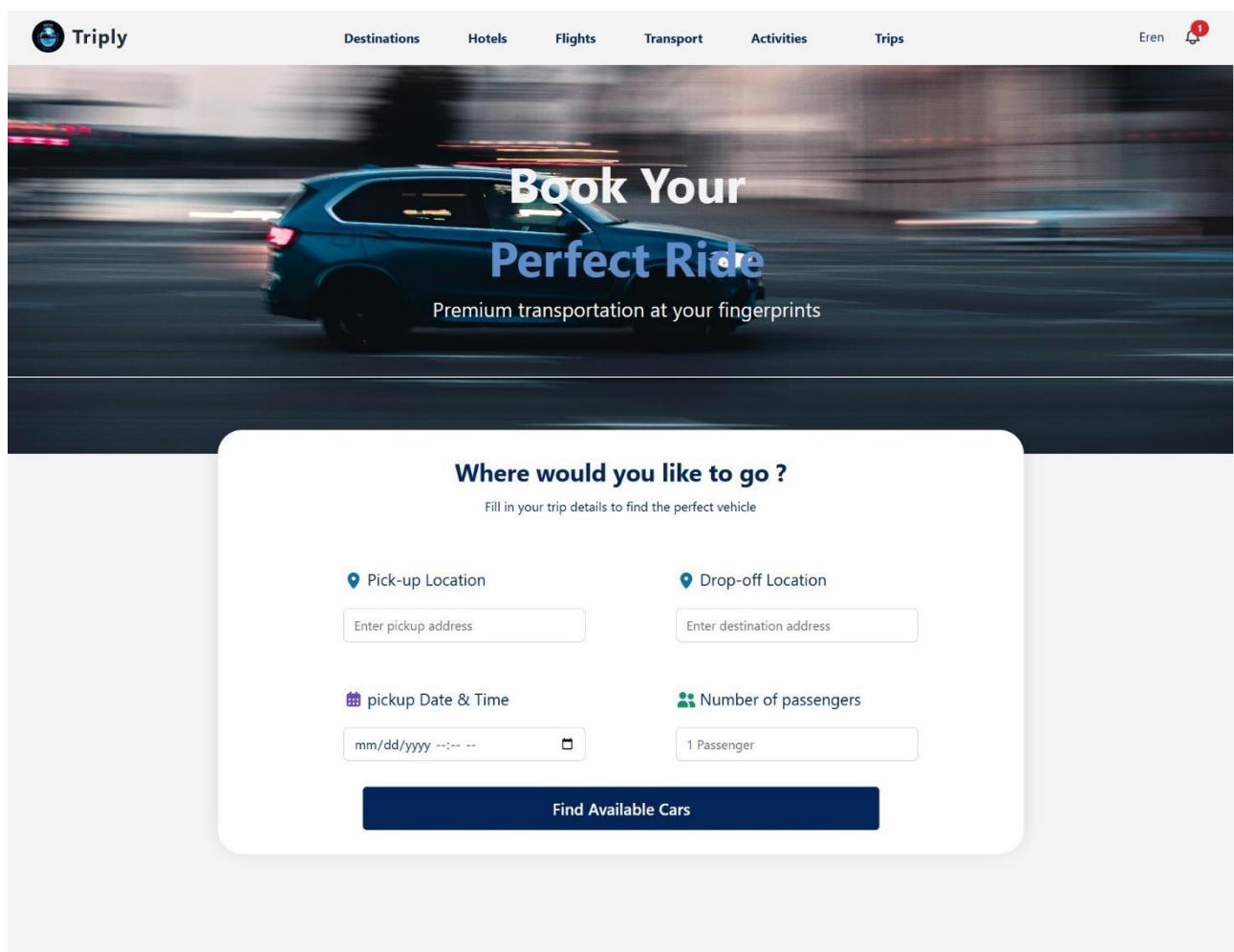


Figure 137 Order Vehicle interface

## -Vehicles Results:

Screenshot of the Triply Vehicles Results interface showing available cars for a trip from Statue of Liberty to Times Square.

**Triply** Destinations Hotels Flights Transport Activities Trips Aya

Back Available cars 14.77 km | 23.8 min

**Trip overview**

- Pickup Location: Statue of Liberty, New York, USA
- Destination Location: Times Square, New York, USA
- Passengers: 2
- Date & Time: 2026-01-03T11:01
- Distance: 14.77 km
- Duration: 23.8 min



**Mercedes-Benz S-class1030** LUXURY

Up to 5 passengers / LUX-001333333

Driver Name : sese55 ll  
6589985

Reserve

**Mercedes-Benz** LUXURY

Up to 8 passengers / LUX-00122345

Driver Name : erwin er  
6589985

Reserve


Figure 138 Vehicles Results interface

**-Book Vehicle:**

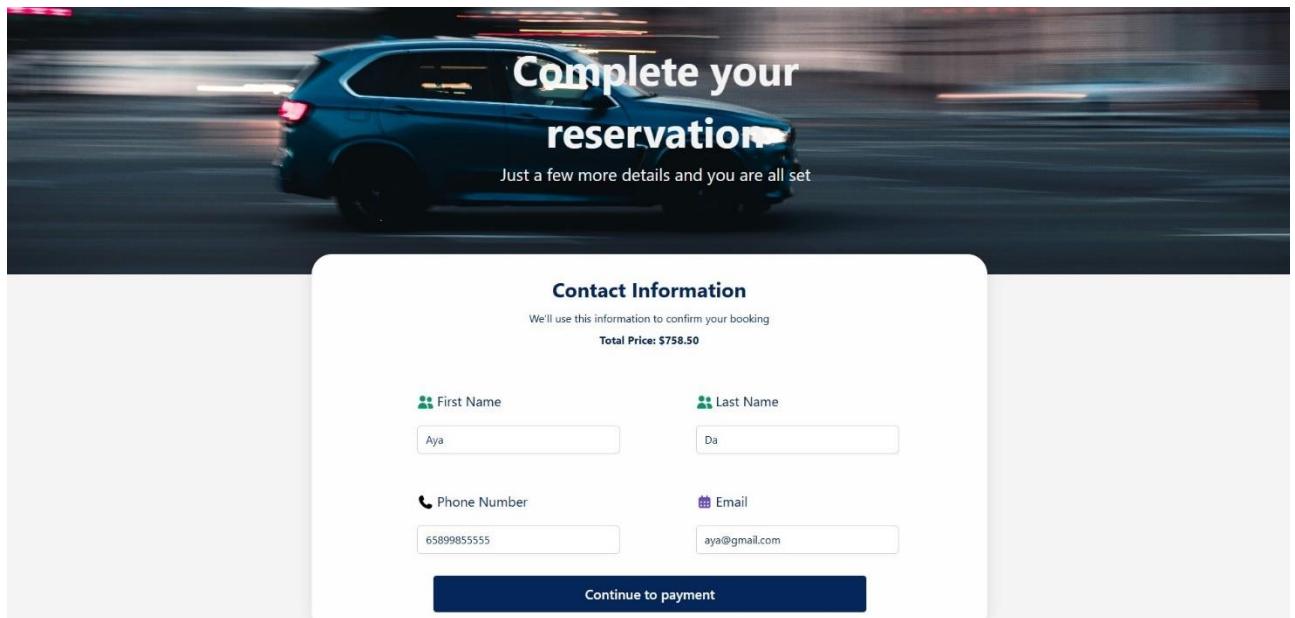


Figure 139 Book Vehicle interface

**-Notifications:**

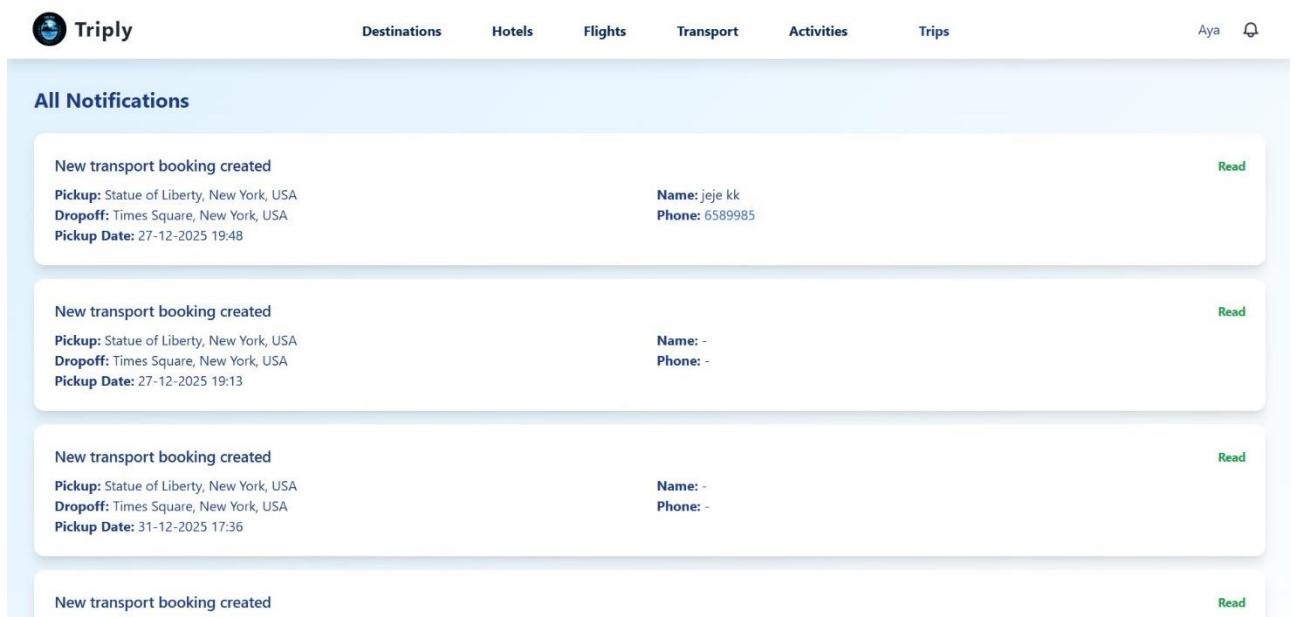


Figure 140 Notifications interface

**-Favorites:**

The screenshot shows the Triply mobile application's "My Favorites" screen. At the top, there is a navigation bar with the Triply logo, followed by links for Destinations, Hotels, Flights, Transport, Activities, and Trips. A user profile icon for "Eren" is also present, with a red notification badge showing the number "1". Below the navigation bar, the title "My Favorites" is displayed. The main content area is divided into two sections: "Favorite Destinations" and "Favorite Hotels".

**Favorite Destinations:**

- **New Zealand**  
Auckland

**Favorite Hotels:**

- **Southern Coast Hotel**  
Damascus
- **The empire hotel**  
Syria

Figure 141 Favorites interface

## -Reservations:

The screenshot shows the Triply web application interface for managing reservations. At the top, there is a navigation bar with links for Destinations, Hotels, Flights, Transport, Activities, and Trips. A user profile for 'Eren' is shown on the right with a notification badge. Below the navigation is a search bar containing 'Pickup, Dropoff' and filter options for Month (All), Year (All), Status (All), with 'Filter' and 'Reset' buttons.

**Transport Reservations**

A list of all your transport reservations.

User	Pickup	Dropoff	Pickup Date	Passengers	Total Price	Status
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2026-01-26 16:51:00	1	\$758.50	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2026-01-03 07:24:00	1	\$93.85	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-31 06:15:00	1	\$93.85	Pending
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-31 06:15:00	1	\$93.85	Pending
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-27 13:19:00	1	\$93.85	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-27 10:31:00	1	\$93.85	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-26 08:06:00	1	\$93.85	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-25 13:33:00	1	\$93.85	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-20 15:14:00	1	\$93.85	Completed

Figure 142 reservation interface

## -Trips:

The screenshot shows the Triply web application interface for managing trips. At the top, there is a navigation bar with links for Destinations, Hotels, Flights, Transport, Activities, and Trips. A user profile for 'Aya' is shown on the right. Below the navigation is a section titled 'My Trips'.

**My Trips**

<b>AI Trip: A trip to see the northern lig...</b> <small>AI Trip</small> A trip to see the northern lights Dates: 2026-01-02 - 2026-01-04 Travelers: 1 <a href="#">View Details</a> <a href="#">Delete</a>	<b>Northern Lights Trip</b> adventure trip to the northern lights Dates: 2026-01-03 - 2026-01-05 Travelers: 2 <a href="#">View Details</a> <a href="#">Delete</a>	<b>AI Trip: A romantic trip to paris...</b> <small>AI Trip</small> A romantic trip to paris Dates: 2026-01-02 - 2026-01-04 Travelers: 2 <a href="#">View Details</a> <a href="#">Delete</a>
<b>AI Generated Trip: A luxury 4-day shopping and food tour in Dubai...</b> A luxury 4-day shopping and food tour in Dubai Dates: 2025-12-31 - 2026-01-03 Travelers: 2 <a href="#">View Details</a> <a href="#">Delete</a>	<b>AI Generated Trip: A luxury 4-day shopping and food tour in Dubai...</b> A luxury 4-day shopping and food tour in Dubai Dates: 2025-12-31 - 2026-01-03 Travelers: 2 <a href="#">View Details</a> <a href="#">Delete</a>	<b>AI Generated Trip: A luxury 4-day shopping and food tour in Dubai...</b> A luxury 4-day shopping and food tour in Dubai Dates: 2025-12-31 - 2026-01-03 Travelers: 2 <a href="#">View Details</a> <a href="#">Delete</a>

Figure 143 Trips interface

## -Trip Details:

The screenshot shows the Triply website interface for a trip titled "AI Trip: A trip to see the northern lig...". Key details include a duration of 3 days, 1 traveler, a budget of \$5,000.00, and dates from Jan 02 - Jan 04, 2026. The itinerary is for a Northern Lights trip and includes sections for Trip Overview, Day 3 (Morning, Afternoon, Evening), Budget Breakdown, and a note about the total estimated budget of \$5000. Buttons for "Back to Trips" and "Print Itinerary" are visible at the bottom.

Figure 144 Trip Details interface

## -Create Trip:

The screenshot shows the Triply website interface for creating a new trip plan. It offers two options: "Manual Creation" (represented by a pen icon) and "AI-Powered Creation" (represented by a sparkler icon). The "Manual Creation" section describes building a trip step-by-step with full control over every detail, listing tasks like choosing destination, selecting hotels, adding activities, and arranging transportation. The "Time required" is 5-10 minutes. The "AI-Powered Creation" section describes letting AI create a complete itinerary based on preferences, listing steps like describing the ideal trip, suggesting hotels, activities, and transport, and perfecting the itinerary in minutes. The "Time required" is 2-3 minutes. A "Back" button is also visible.

Figure 145 create Trip interface

### -Create AI Trip:

The screenshot shows the 'AI Trip Creator' interface. At the top left is a 'Back' button. The title 'AI Trip Creator' is centered above a subtitle 'Describe your ideal trip and let AI do the planning.' Below this is a section titled 'Tell us about your dream trip' with a yellow star icon. A text input field for 'Trip Description' contains placeholder text 'e.g., A romantic 5-day trip to Paris...'. To the right are fields for 'Number of Travelers' (set to 1) and 'Duration (Days)' (set to 3). Below these are date inputs for 'Start Date (Optional)' and 'End Date (Optional)'. Further down are fields for 'Estimated Budget (\$)' (set to 'Optional') and 'Language' (set to 'English'). At the bottom is a purple 'Generate My Trip' button.

Figure 146 create ai Trip interface

### -Create Manual Trip – Step1:

The screenshot shows the 'Create New Trip Plan' interface, specifically 'Manual planning — step: 1'. At the top left is a 'Back' button. The title 'Create New Trip Plan' is at the top center, with 'Manual Planning' below it. A progress bar shows steps 1 through 5. The main area is titled 'Basic Trip Information' with a sub-instruction 'Let's start with essentials'. It contains fields for 'Trip Name' (placeholder 'e.g., European Adventure 2024'), 'Description' (placeholder 'Brief description of your trip ...'), 'Start Date' (date input), 'End Date' (date input), 'Number of Travelers' (text input placeholder 'Enter number of Travelers'), and 'Estimated Budget \$' (text input placeholder 'Enter your Budget'). At the bottom is a 'Next Step' button.

Figure 147 Create Manual Trip – Step1 -interface

## -Create Manual Trip – Step2:

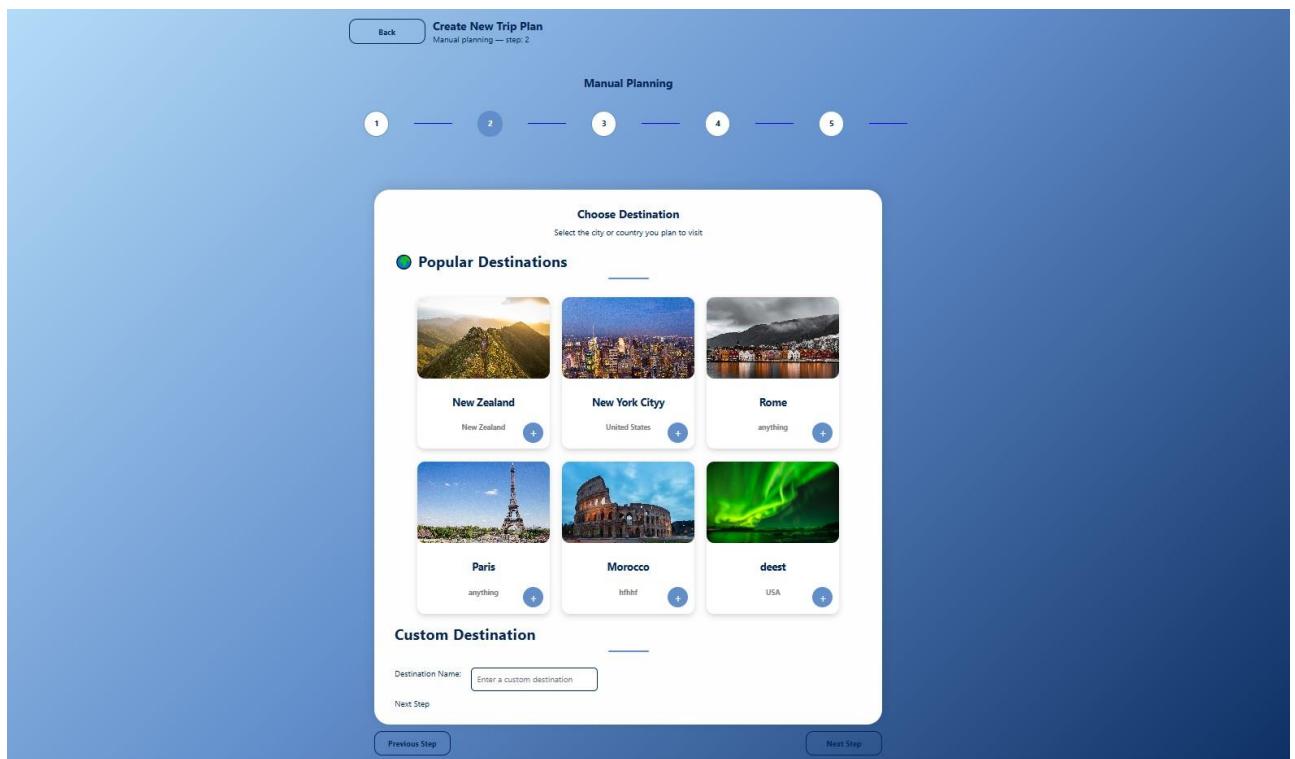


Figure 148 Create Manual Trip – Step2 interface

### -Create Manual Trip – Step3:

The screenshot shows the 'Manual Planning' interface for creating a manual trip, specifically Step 3: Choose Your Hotels. At the top, there is a navigation bar with five numbered steps: 1, 2, 3 (highlighted in blue), 4, and 5. Below the navigation bar is a section titled 'Choose Your Hotels' with the sub-instruction 'Select accommodations for each day of your trip'. This section contains a heading 'Available Hotels' with a small icon. Three hotel options are listed in cards:

- Southern Coast Hotel**: \$50 / night. Rating: 5 stars. Features: Free Wi-Fi, Breakfast included, Pool access.
- The empire hotel**: \$50 / night. Rating: 5 stars. Features: Free Wi-Fi, Breakfast included, Pool access.
- Harbour View Auckland Hotel**: \$20 / night. Rating: 5 stars. Features: Free Wi-Fi, Breakfast included, Pool access.

Below this is a section titled 'Assign Hotels to Days' with the sub-instruction 'Select which hotel you'll stay at for each day:'. It includes three dropdown menus labeled 'Day 1', 'Day 2', and 'Day 3', each with the placeholder text '-- Select Hotel --'. Underneath this is a section titled 'Assign Custom Hotels To Days' with the sub-instruction 'You can custom your hotels and assign them to your trip's days'. It features three input fields labeled 'Day 1', 'Day 2', and 'Day 3', each with the placeholder text 'Enter hotel name for Day 1', 'Enter hotel name for Day 2', and 'Enter hotel name for Day 3' respectively. At the bottom of the screen are two buttons: 'Previous Step' on the left and 'Next Step' on the right.

Figure 149 Create Manual Trip – Step3 interface

### -Create Manual Trip – Step4:

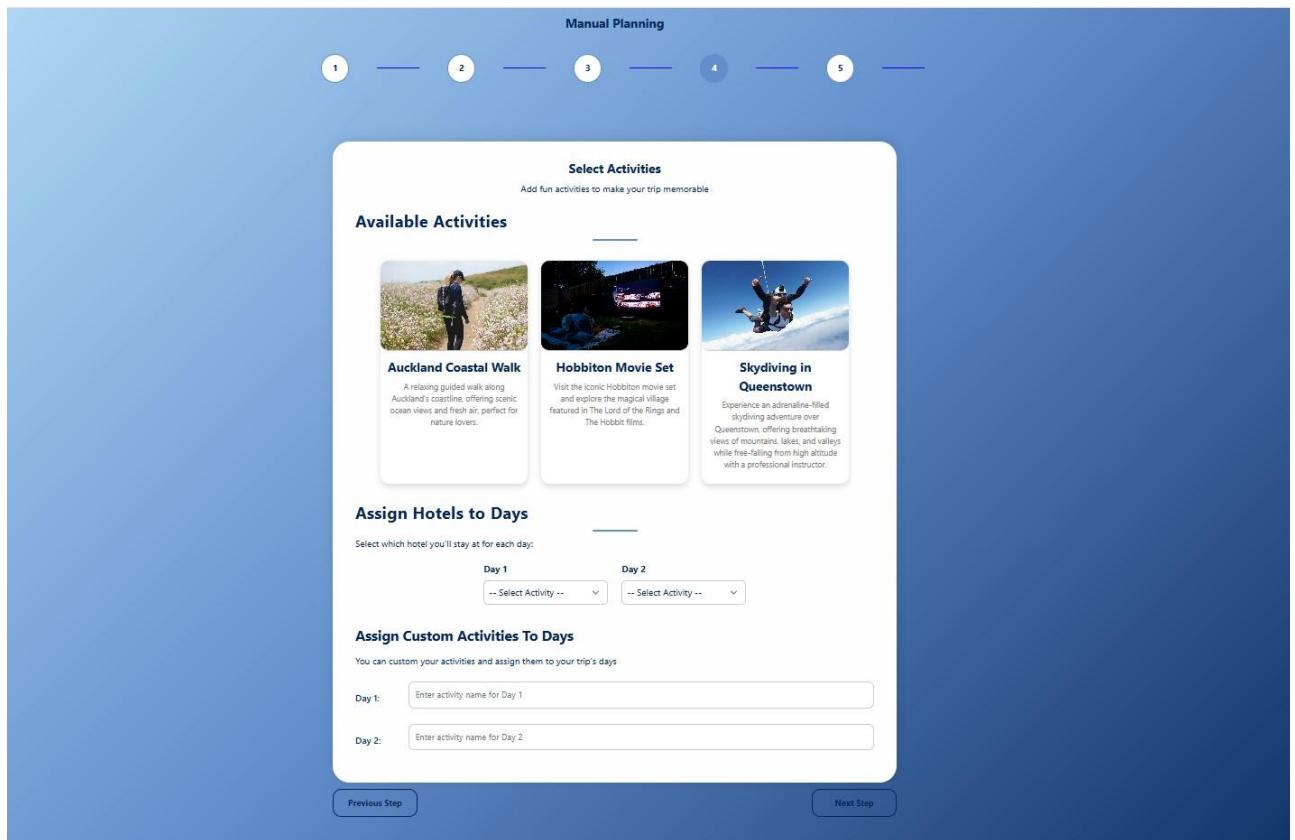


Figure 150 Create Manual Trip – Step4 interface

### -Create Manual Trip – Step5:

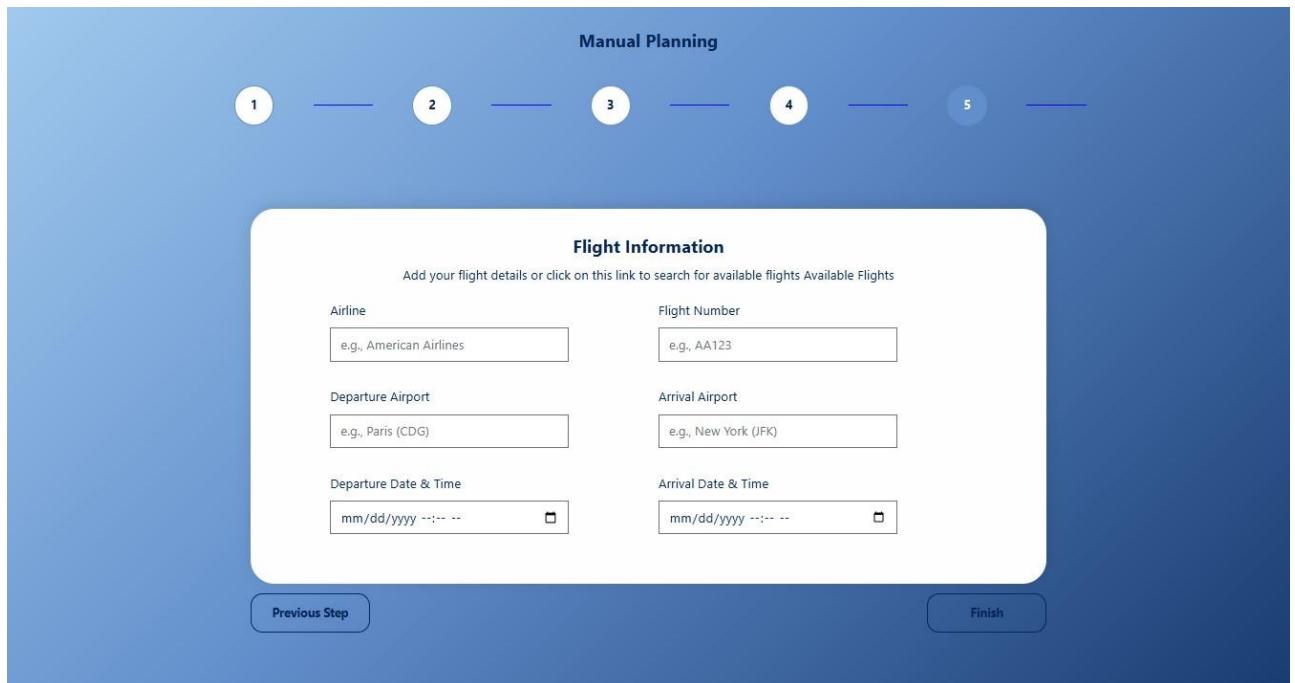


Figure 151 Create Manual Trip – Step5 interface

# Admin Interfaces:

## -Destinations:

The screenshot shows the 'Destinations' section of the Triply admin interface. At the top, there's a navigation bar with links for Destinations, Hotels, Flights, Transport, Activities, Trips, Drivers, and Reservations. On the far right, there are 'Admin' and a search icon. Below the navigation is a large, scenic aerial photograph of a tropical island. A search bar with placeholder text 'Search your next destination' and a 'Search' button is positioned over the image. Below the search bar, there are three destination cards:

- New Zealand** (POPULAR): Described as a stunning island country known for its breathtaking landscapes. Includes 'Edit' and 'Delete' buttons.
- New York City** (POPULAR): Described as one of the world's most iconic cities, known for its towering skyscrapers. Includes 'Edit' and 'Delete' buttons.
- Rome** (POPULAR): Described as the most beautiful place to visit. Includes 'Edit' and 'Delete' buttons.

Below these cards are three smaller thumbnail images: the Eiffel Tower, the Colosseum, and the Aurora Borealis.

Figure 152 Destinations interface

## -Create Destination:

The screenshot shows the 'Add New Destination' form. The title 'Add New Destination' is at the top. A note below it says: 'Note: You can add a new activity for this destination from here [Activities](#)'. The form contains the following fields:

Destination Name	timezone
City	language
Country	currency
Description	nearest_airport
location_details	best_time_to_visit
IATA Code (for the most famous airport in this destination) you can find the right IATA code here <a href="#">IATA-CODES</a>	emergency_numbers
Images	local_tip
<input type="button" value="Choose Files"/> No file chosen	Highlight <input type="text" value="Enter a highlight (e.g., Famous landmark)"/>

Below the form, there's a note: 'Click (+) to add more highlights'. At the bottom are two buttons: 'Create Destination' (green) and 'Cancel'.

Figure 153 Create Destination interface

## -Edit Destination:

### Edit Destination

\*Note: You can add a new activity for this destination from here [Activities](#)

Destination Name	timezone
New Zealand	New Zealand Standard Time (NZST, UTC +12)
City	language
Auckland	English, Māori
Country	currency
New Zealand	New Zealand Dollar (NZD)
Description	nearest_airport
New Zealand is a stunning island country known for its breath-taking landscapes and natural beauty.	Auckland International Airport
location_details	best_time_to_visit
Located in the southwestern Pacific Ocean, New Zealand consists of two main islands: North Island and South Island.	October to April, when the weather is mild and ideal for outdoor activities.
IATA Code (for the most famous airport in this destination) you can find the right IATA code here <a href="#">IATA-CODES</a>	emergency_numbers
AKL	111 (Police, Fire, Ambulance)
Images	local_tip
<input type="button" value="Choose Files"/> No file chosen	Renting a car is highly recommended to fully explore New Zealand.
Highlight	
Dramatic landscapes	
Adventure sports	
The filming location of The Lord of the Rings	
Click (-) to add more highlights	
<input type="button" value="+"/>	
<input type="button" value="Update Destination"/> <input type="button" value="Cancel"/>	

**Current Images**

Figure 154 Edit Destination interface

## -Hotels:

The screenshot shows the Triply software interface with the 'Hotels' tab selected. At the top, there is a navigation bar with links for Destinations, Hotels, Flights, Transport, Activities, Trips, Drivers, and Reservations. On the far right, there are 'Admin' and a lock icon. A blue button labeled '+ Add New Hotel' is located in the top right corner. Below the navigation bar, there is a search bar with the placeholder 'Search hotels...' and a 'Search' button. The main area displays three hotel cards: 'Southern Coast Hotel' (123 Quay Street, Auckland, New Zealand), 'The empire hotel' (mnfdjnjkgfknkdfjdf), and 'Harbour View Auckland Hotel' (15 Quay Street, Auckland, New Zealand). Each card includes a thumbnail image, the hotel name, address, and two buttons: 'Edit' (blue) and 'Delete' (red).

Figure 155 Hotels interface

**-Create Hotel:**

### Add New Hotel

<p><b>Hotel Name</b> <input type="text" value="Enter Hotel name"/></p> <p><b>Associated destination</b> <input type="text" value="-- Select the associated destination --"/></p> <p><b>City</b> <input type="text"/></p> <p><b>Country</b> <input type="text"/></p> <p><b>Description</b> <input type="text" value="write a description"/></p> <p><b>address</b> <input type="text" value="Enter address"/></p> <p><b>global_rating</b> <input type="text" value="Enter global_rating(max:5)"/></p> <p><b>price_per_night</b> <input type="text" value="Enter price_per_night \$"/></p> <p><b>total_rooms</b> <input type="text" value="Enter total_rooms"/></p> <p><b>stars</b> <input type="text" value="Select how many stars stars"/></p> <p><b>Images</b> <input type="button" value="Choose Files"/> No file chosen</p>	<p><b>Pets Allowed</b> <input type="text" value="-- Select --"/></p> <p><b>check_in_time</b> <input type="text" value="--:-- --"/></p> <p><b>check_out_time</b> <input type="text" value="--:-- --"/></p> <p><b>policies</b> <input type="text" value="Enter policies"/></p> <p><b>phone_number</b> <input type="text" value="phone_number"/></p> <p><b>email</b> <input type="text" value="Enter @email.com"/></p> <p><b>website</b> <input type="text" value="Enter A website"/></p> <p><b>nearby_landmarks</b> <input type="text" value="Enter nearby landmarks"/></p> <p><b>Amenities</b></p> <table border="0"><tr><td><input type="checkbox" value="Wifi"/> Wifi</td><td><input type="checkbox" value="Parking"/> Parking</td></tr><tr><td><input type="checkbox" value="Pool"/> Pool</td><td><input type="checkbox" value="Spa"/> Spa</td></tr><tr><td><input type="checkbox" value="Restaurant"/> Restaurant</td><td><input type="checkbox" value="Gym"/> Gym</td></tr><tr><td><input type="checkbox" value="Laundry"/> Laundry</td><td><input type="checkbox" value="Air Condition"/> Air Condition</td></tr><tr><td colspan="2"><input type="checkbox" value="Free Breakfast"/> Free Breakfast</td></tr></table>	<input type="checkbox" value="Wifi"/> Wifi	<input type="checkbox" value="Parking"/> Parking	<input type="checkbox" value="Pool"/> Pool	<input type="checkbox" value="Spa"/> Spa	<input type="checkbox" value="Restaurant"/> Restaurant	<input type="checkbox" value="Gym"/> Gym	<input type="checkbox" value="Laundry"/> Laundry	<input type="checkbox" value="Air Condition"/> Air Condition	<input type="checkbox" value="Free Breakfast"/> Free Breakfast	
<input type="checkbox" value="Wifi"/> Wifi	<input type="checkbox" value="Parking"/> Parking										
<input type="checkbox" value="Pool"/> Pool	<input type="checkbox" value="Spa"/> Spa										
<input type="checkbox" value="Restaurant"/> Restaurant	<input type="checkbox" value="Gym"/> Gym										
<input type="checkbox" value="Laundry"/> Laundry	<input type="checkbox" value="Air Condition"/> Air Condition										
<input type="checkbox" value="Free Breakfast"/> Free Breakfast											

Figure 156Create Hotel interface

**-Edit Hotel:**

### Edit Hotel

Hotel Name <input type="text" value="Harbour View Auckland Hotel"/>	Pets Allowed <input type="text" value="-- Select --"/>
Associated destination <input type="text" value="New Zealand"/>	check_in_time <input type="text" value="06:36 PM"/>
City <input type="text" value="Auckland"/>	check_out_time <input type="text" value="06:36 PM"/>
Country <input type="text" value="New Zealand"/>	policies <input type="text" value="Check-in requires a valid ID. No smoking inside rooms. Free Wi-Fi."/>
Description <input type="text" value="New York City is one of the world's most iconic cities, known for its art, culture, and history."/>	phone_number <input type="text" value="65899855555"/>
address <input type="text" value="15 Quay Street, Auckland, New Zealand"/>	email <input type="text" value="s@s.com"/>
global_rating <input type="text" value="5"/>	website <input type="text" value="http://127.0.0.1:8000/hotels/1/edit"/>
price_per_night <input type="text" value="20"/>	nearby_landmarks <input type="text" value="Free cancellation up to 24 hours before check-in"/>
total_rooms <input type="text" value="3"/>	Amenities
stars <input type="text" value="Select how many stars stars"/>	<input checked="" type="checkbox"/> Wifi <input type="checkbox"/> Parking
Images <input type="text" value="Choose Files No file chosen"/>	<input checked="" type="checkbox"/> Pool <input type="checkbox"/> Spa
	<input checked="" type="checkbox"/> Restaurant <input type="checkbox"/> Gym
	<input type="checkbox"/> Laundry <input type="checkbox"/> Air Condition
	<input type="checkbox"/> Free Breakfast

Current Images



Figure 157 Edit Hotel interface

**-Activities:**

The screenshot shows the 'Activities' section of the Triply platform. At the top, there's a navigation bar with links for Destinations, Hotels, Flights, Transport, Activities (which is the active tab), Trips, Drivers, and Reservations. On the far right, there are 'Admin' and a lock icon. Below the navigation is a large banner image of a person sitting on a rocky outcrop overlooking a turquoise lake and mountains. The word 'Activities' is centered in a bold, white font. Below the banner is a search bar with the placeholder 'Search by activity or destination' and a 'Search' button. Underneath the search bar are several filter options: Availability, Difficulty Level, Guide Language, Requires Booking, Family Friendly, Pets Allowed, Category, Filter, and Reset. A blue 'Add New Activity' button is located below the filters. The main content area displays four activity cards, each with a thumbnail image, the activity name, and two buttons: 'More Details' and either 'Edit' or 'Delete'. The activities listed are Auckland Coastal Walk, Hobbiton Movie Set, Park Tour, and Skydiving in Queenstown.

Activity	More Details	Edit / Delete
Auckland Coastal Walk	<a href="#">More Details</a>	<a href="#">Edit</a> <a href="#">Delete</a>
Hobbiton Movie Set	<a href="#">More Details</a>	<a href="#">Edit</a> <a href="#">Delete</a>
Park Tour	<a href="#">More Details</a>	<a href="#">Edit</a> <a href="#">Delete</a>
Skydiving in Queenstown	<a href="#">More Details</a>	<a href="#">Edit</a> <a href="#">Delete</a>

Figure 158 activites interface

**-Add Activity:**

### Add New Activity

<p>Activity Name <input type="text" value="Enter activity name"/></p> <p>description <input type="text" value="Enter activity description"/></p> <p>Select Destination <input type="text" value="-- Choose a destination --"/></p> <p>Duration <input type="text" value="e.g. 2"/></p> <p>Duration Unit <input type="text" value="Hours"/></p> <p>Start Time <input type="text" value="--:-- --"/></p> <p>End Time <input type="text" value="--:-- --"/></p> <p>Start Date <input type="text" value="mm/dd/yyyy"/></p> <p>End Date <input type="text" value="mm/dd/yyyy"/></p> <p>Availability <input type="text" value="Available"/></p> <p>Guide Name <input type="text"/></p> <p>Guide Language <input type="text"/></p> <p>Contact Number <input type="text"/></p> <p>Requirements <input type="text"/></p>	<p>Price <input type="text" value="e.g. 50.00"/></p> <p>Category <input type="text" value="-- Choose category --"/></p> <p>Is Active <input type="text" value="Inactive"/></p> <p>Difficulty Level <input type="text" value="-- Select --"/></p> <p>Amenities <input type="checkbox"/> WiFi   <input type="checkbox"/> Parking   <input type="checkbox"/> Pool <input type="checkbox"/> Restaurant   <input type="checkbox"/> Bar</p> <p>Address <input type="text"/></p> <p>Requires Booking? <input checked="" type="checkbox"/> Yes   <input type="checkbox"/> No</p> <p>Pets Allowed? <input checked="" type="checkbox"/> Yes   <input type="checkbox"/> No</p> <p>Family Friendly <input type="text" value="All Ages"/></p> <p>Highlights <input type="text"/></p> <p>Activity Image <input type="button" value="Choose File"/> No file chosen</p>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Figure 159 Add Activity interface

**-Edit Activity:**

### Edit Activity

Activity Name	Price
Skydiving in Queenstown	25.00
description	Category
Experience an adrenaline-filled skydiving adventure over Qu	Nature
Select Destination	Is Active
New Zealand	Inactive
Duration	Difficulty Level
5	Hard
Duration Unit	Amenities
Hours	<input type="checkbox"/> WiFi <input checked="" type="checkbox"/> Parking <input type="checkbox"/> Pool
Start Time	<input type="checkbox"/> Restaurant <input checked="" type="checkbox"/> Bar
--:-- --	
End Time	Address
--:-- --	Queenstown Airport, Queenstown, New Zealand
Start Date	Requires Booking?
mm/dd/yyyy	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
End Date	Pets Allowed?
mm/dd/yyyy	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
Availability	Family Friendly
Not Available	Adults Only
Guide Name	Highlights
Sara	
Guide Language	Activity Image
English	<input type="button" value="Choose File"/> No file chosen
Contact Number	
24154446655	
Requirements	
Nothing	

Figure 160 Edit Activity interface

## -Transport Services:

The screenshot shows the 'Transport Services' section of the Triply platform. At the top, there's a banner with the text 'choose from our curated selection of premium transportation options each designed to elevate your travel experience'. Below the banner, there are three service cards:

- A trip around the city**: Description: 'nothing'. Options: 'City tour' (checkbox), 'Available 24/7' (checkbox). Buttons: 'Edit' and 'Delete'.
- Order from your place**: Description: 'You can order a private car anytime you want'. Options: 'Private transport' (checkbox), 'Available 24/7' (checkbox). Buttons: 'Edit' and 'Delete'.
- service2**: Description: 'whhhhhh'. Options: 'City tour' (checkbox), 'Available 24/7' (checkbox). Buttons: 'Edit' and 'Delete'.

At the bottom right of the card area, there are buttons for 'Add Vehicles' and 'View Vehicles'.

Figure 161 Transport Services interface

## -Create Transport Service:

The screenshot shows the 'Add New Service' dialog box over a blurred background of a busy street scene. The dialog has the following fields:

- Service Name**: Input field with placeholder 'Enter service name'.
- Service Type**: Drop-down menu currently set to 'Airport pick up'.
- Description**: Input field with placeholder 'Enter description'.
- Service Image**: Input field with placeholder 'Choose File' and note 'No file chosen'.

At the bottom right of the dialog are 'Save' and 'Cancel' buttons.

Figure 162 Create Transport Service interface

## -Edit Transport Service:

The screenshot shows the Triply platform's 'Edit Service' interface. A modal window titled 'Edit Service' is open, containing fields for 'Service Name' (set to 'Order from your place'), 'Service Type' (set to 'Private transport'), and 'Description' (containing the text 'You can order a private car anytime you want'). Below these fields is a 'Service Image' section with a placeholder 'Choose File' button and a 'No file chosen' message. At the bottom of the modal are 'Save' and 'Cancel' buttons. In the background, there are blurred sections of the platform, including a 'A trip around the city' section with a car image and a 'service2' section with a sunset image.

Figure 163 Edit Transport Service interface

## -Vehicles:

The screenshot shows the 'Available cars' interface. It displays two vehicle entries. The first entry is for a 'Mercedes-Benz S-class1030' (LUX-001333333), which is labeled as 'LUXURY'. It shows a white car image, the driver information ('Driver Name : sese55 II' and '6589985'), and the price details ('Base Price : 20.00\$' and '+ \$50.00/km'). The second entry is for a 'Mercedes-Benz' (LUX-00122345), also labeled as 'LUXURY'. It shows a red car image, the driver information ('Driver Name : erwin er' and '6589985'), and the price details ('Base Price : 20.00\$' and '+ \$5.00/km'). Both entries have 'Edit Vehicle' and 'Delete Vehicle' buttons at the bottom.

Figure 164 Vehicles interface

**-Create Vehicle:**

### Add New Vehicle

<p>Car Model</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Enter car model</div>	<p>Max Passengers</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">e.g. 4</div>
<p>Plate Number</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">Enter plate number</div>	<p>Base Price</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">e.g. 50.00</div>
<p>Select Driver</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">-- Choose a driver --</div>	<p>Price per KM</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;">e.g. 5.00</div>
<p>Vehicle Image</p> <div style="border: 1px solid #ccc; padding: 5px; width: 100%;"><input type="button" value="Choose File"/> No file chosen</div>	
<div style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; margin: 0 10px;">Save</div> <div style="border: 1px solid #ccc; border-radius: 5px; padding: 2px 10px; background-color: #f0f0f0;">Cancel</div>	

Figure 165 Create Vehicle interface

**-Edit Vehicle:**

### Edit Vehicle

Car Model	Max Passengers
Mercedes-Benz S-class1030	5
Plate Number	Base Price
LUX-001333333	20.00
Select Driver	Price per KM
sese55	50.00
Category	
LUXURY	

Vehicle Image

Current Image:



Leave the file input empty to keep the current image.

No file chosen

Figure 166 Edit Vehicle interface

## -Transport Reservations:

**Transport Reservations**  
A list of all the transport reservations in your system.

User	Pickup	Dropoff	Pickup Date	Passengers	Total Price	Status
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2026-01-26 16:51:00	1	\$758.50	Completed
Aya	Statue of Liberty, New York, USA	Times Square, New York, USA	2026-01-03 16:40:00	1	\$758.50	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2026-01-03 07:24:00	1	\$93.85	Completed
Aya	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-31 17:36:00	1	\$758.50	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-31 06:15:00	1	\$93.85	Pending
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-31 06:15:00	1	\$93.85	Pending
Aya	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-27 19:48:00	1	\$93.85	Completed
Aya	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-27 19:13:00	1	\$93.85	Completed
Aya	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-27 15:00:00	1	\$93.85	Completed
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	2025-12-27 13:19:00	1	\$93.85	Completed

Figure 167 Reservations interface

## -Hotel Reservations:

**Hotel Reservations**  
A list of all the hotel reservations in your system.

User	Hotel	Check-in	Check-out	Rooms	Guests	Total Price	Status
Eren yeager	The empire hotel	2026-05-02	2026-05-04	2	2	\$200.00	Paid
Iaita la	Southern Coast Hotel	2026-01-03	2026-01-06	2	2	\$300.00	Paid

Figure 168 Hotel Reservations interface

## -Drivers:

**Driver List**  
A list of all the drivers in your system.

FIRST NAME	LAST NAME	EMAIL	CONTACT	COUNTRY	ADDRESS	EXPERIENCE	LICENSE CATEGORY	PHOTO OF LICENSE	STATUS	DATE OF HIRE	Actions	
sese	ss	sese@gmail.com	6589985	USA	njjj	kkkkdkdkjk	A	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	17-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
del	rey	del@gmail.com	6589985	Algeria	hhhhhhhhhh		A	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
lele	le	lele2@gmail.com	6589985555	Australia	jjjjjjjjjj	kkkkkkkkkk	A	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
jan	ll	jan@gmail.com	6589985	Austria	hfhfhfhfhf	-----	B	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
levi	oo	levi2@gmail.com	6589985	Azerbaijan	cjqcjqc	pppppppppp	B	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
new	nn	new@gmail.com	6589985555	Azerbaijan	dhdhdhdhd	shshshsh	B	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
laila	laila	laila55@gmail.com	65289466	Bahamas	shhshsh	shshshshshs	A	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
erwin	er	erwin@gmail.com	6589985	Austria	dhdhdh	edhdhdhdh	A	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	18-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
sedra44	ll	sedrarr@gmail.com	6589985555	Australia	whwhwhwh	whwhwhwh	A	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	20-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>
sese55	ll	sese55@gmail.com	6589985	Austria	sgsgsg	sgsgsg	B	<a href="#">View License</a>	Approved	<a href="#">Confirm</a>	20-12-2025	<a href="#">Delete</a> <a href="#">Completed Reservations</a> <a href="#">Pending Reservations</a>

Figure 169 Drivers interface

## -Driver's Reservations:

**Reservations jan**  
A list of all the Pending Reservations in your system for a driver.

CLIENT NAME	CLIENT PHONE NUMBER	PICKUP LOCATION	DROPOFF LOCATION	PICKUP DATETIME	PASSENGER	TOTAL PRICE	STATUS
Eren	6589985555	Statue of Liberty, New York, USA	Times Square, New York, USA	20-12-2025 15:14	1	\$93.85	Pending
sese	6589985	Statue of Liberty, New York, USA	Times Square, New York, USA	26-12-2025 16:13	1	\$93.85	Pending

Figure 170 Driver's Reservations interface

# Driver Interfaces:

## -Pending Reservations:

The screenshot shows the 'My Pending Reservations' section of the Triply app. At the top, there's a header with the Triply logo, navigation links for 'Pending Bookings' and 'My Completed Bookings', and a notification badge 'jan 2'. Below the header, a sub-header reads 'My Pending Reservations' with a subtitle 'A list of all my reservations.' A table lists two pending bookings:

CLIENT NAME	CLIENT PHONE NUMBER	PICKUP LOCATION	DROPOFF LOCATION	PICKUP DATETIME	PASSENGER	TOTAL PRICE	STATUS	ACTIONS
Eren	6589985555	Statue of Liberty, New York, USA	Times Square, New York, USA	20-12-2025 15:14	1	\$93.85	Pending	<button>Complete</button> <button>Cancel</button>
sese	6589985	Statue of Liberty, New York, USA	Times Square, New York, USA	26-12-2025 16:13	1	\$93.85	Pending	<button>Complete</button> <button>Cancel</button>

Figure 171 Pending Reservations: interface

## -Completed Reservations:

The screenshot shows the 'My completed Reservations' section of the Triply app. At the top, there's a header with the Triply logo, navigation links for 'Pending Bookings' and 'My Completed Bookings', and a notification badge 'jan 2'. Below the header, a sub-header reads 'My completed Reservations' with a subtitle 'A list of all my reservations.' A table lists two completed bookings:

CLIENT NAME	PICKUP LOCATION	DROPOFF LOCATION	PICKUP DATETIME	PASSENGERS	TOTAL PRICE	STATUS
Eren	Statue of Liberty, New York, USA	Times Square, New York, USA	20-12-2025 15:14	1	\$93.85	Completed
sese	Statue of Liberty, New York, USA	Times Square, New York, USA	26-12-2025 16:13	1	\$93.85	Completed

Figure 172 Completed Reservations interface

## -Notifications

The screenshot shows the 'All Notifications' section of the Triply app. At the top, there is a header with the Triply logo, navigation links for 'Pending Bookings' and 'My Completed Bookings', and a user profile icon for 'jeje' with a red notification badge containing the number '2'. Below the header, the title 'All Notifications' is displayed. Two notifications are listed:

- New transport booking created** (Unread)  
Name: Aya Da  
Phone: 65899855555  
Details:  
**Pickup:** Statue of Liberty, New York, USA  
**Dropoff:** Times Square, New York, USA  
**Pickup Date:** 27-12-2025 19:48
- New transport booking created** (Unread)  
Name: -  
Phone: -  
Details:  
**Pickup:** Statue of Liberty, New York, USA  
**Dropoff:** Times Square, New York, USA  
**Pickup Date:** 27-12-2025 19:13

Figure 173 Notifications interface

### 3.5 Test Cases:

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Status
TC-FR-SU-01-01	Login with valid credentials	1. Go to login page 2. Enter valid email & password 3. Click Login	Valid email & password	User logged in successfully, redirected to dashboard	Passed
TC-FR-SU-01-02	Login with wrong password	1. Go to login page 2. Enter valid email & wrong password 3. Click Login	Valid email, wrong password	Error message “Invalid credentials” displayed	Passed
TC-FR-SU-01-03	Login with unregistered email	1. Go to login page 2. Enter non-existent email & any password 3. Click Login	Non-existent email	Error message “Invalid credentials” displayed	Passed
TC-FR-SU-02-01	Logout	1. Login 2. Click Logout button	-	User logged out, redirected to login page	Passed
TC-FR-SU-03-01	Edit profile with valid data	1. Login 2. Go to Profile 3. Update email/password 4. Click Save	Valid email & password	Profile updated successfully, confirmation shown	Passed
TC-FR-SU-03-02	Edit profile with empty password	1. Login 2. Go to Profile 3. Leave password empty 4. Save	Empty password	Error message “Password cannot be empty”	Passed
TC-FR-USER-01-01	Register with valid info	1. Go to registration page 2. Fill valid info 3. Submit	Valid name, email, password	User registered successfully, redirected to login	Passed
TC-FR-USER-01-02	Register with existing email	1. Go to registration page 2. Enter email already registered 3. Submit	Existing email	Error message “Email already exists”	Passed
TC-FR-USER-02-01	View destinations list	1. Login 2. Navigate to Destinations page	-	Destinations list displayed	Passed
TC-FR-USER-03-01	View destination details	1. Navigate to destinations page 2. Click on a destination	Existing destination	Full details displayed	Passed
TC-FR-USER-04-01	Search destination	1. Navigate to destinations page 2. Enter valid name/location 3. Click Search	Existing name/location	Matching destinations displayed	Passed

<b>TC-FR-USER-04-02</b>	Search destination with no match	1. Login 2. Enter non-existent name/location 3. Search	Random name/location	Message “No results found”	Passed
<b>TC-FR-USER-05-01</b>	View hotels list	1. Login 2. Navigate to Hotels page	-	Hotels list displayed	Passed
<b>TC-FR-USER-06-01</b>	View hotel details	1. Navigate to Hotels page 2. Click on a hotel	Existing hotel	Hotel details displayed	Passed
<b>TC-FR-USER-07-01</b>	Search hotels	1. Navigate to Hotels page 2. Enter valid criteria 3. Click Search	Name/location /rating	Matching hotels displayed	Passed
<b>TC-FR-USER-07-02</b>	Search hotel with no match	1. Login 2. Enter non-matching criteria 3. Click Search	Random criteria	Message “No hotels found”	Passed
<b>TC-FR-USER-08-01</b>	View transport services	1. Login 2. Navigate to Transport Services	-	Transport services list displayed	Passed
<b>TC-FR-USER-09-01</b>	View activities list	1. Login 2. Navigate to Activities page	-	Activities list displayed	Passed
<b>TC-FR-USER-10-01</b>	View activity details	1. Navigate to Activities page 2. Click on an activity	Existing activity	Activity details displayed	Passed
<b>TC-FR-USER-10-02</b>	Search activities	1. Login 2. Enter valid name/destination/ filters 3. Click Search	Existing data	Matching activities displayed	Passed
<b>TC-FR-USER-10-03</b>	Search activities no match	1. Navigate to Activities page 2. Search with invalid criteria	Random criteria	Message “No activities found”	Passed
<b>TC-FR-USER-11-01</b>	Search bookings	1. Login 2. Navigate to Bookings 3. Search/filter bookings	Valid criteria	Relevant bookings displayed	Passed
<b>TC-FR-USER-11-02</b>	Search bookings no match	1. Login 2. Navigate to Bookings 3. Search with invalid criteria	Random criteria	Message “No activities found”	Passed
<b>TC-FR-TRV-01-01</b>	Book hotel	1. Login 2. Choose hotel 3. Fill booking form 4. Submit	Valid hotel & dates	Booking confirmed, confirmation shown	Passed

<b>TC-FR-TRV-01-02</b>	Book hotel invalid data	1. Navigate to hotels page 2. Choose hotel 3. Fill form with invalid dates/empty fields 4. Submit	Invalid dates or empty fields	Error message displayed	Passed
<b>TC-FR-TRV-02-01</b>	Search vehicles	1. Choose transport service 2. Enter required date, time 3. Click Search	Valid date/time/service	Available vehicles displayed	Passed
<b>TC-FR-TRV-02-02</b>	Search vehicles no availability	1. Choose transport service 2. Enter date/time/service with no available vehicles 3. Search	Date/time/service with no availability	Message “No vehicles available”	Passed
<b>TC-FR-TRV-03-01</b>	Book vehicle	1. Click Book for a selected vehicle 2. Fill booking info 3. Submit	Valid vehicle & data	Vehicle reserved successfully	Passed
<b>TC-FR-TRV-03-02</b>	Book vehicle invalid data	1. Click Book for a selected vehicle 2. submit with empty fields	Empty fields	Booking failed, error message	Passed
<b>TC-FR-TRV-04-01</b>	Complete payment	1. Select booking 2. Go to payment 3. Enter valid payment info 4. Submit	Valid payment info	Payment successful, confirmation shown, Booking created	Passed
<b>TC-FR-TRV-05-01</b>	Receive booking notification	1. Make a booking 2. Complete payment 3. Check notifications	-	User receives notification of new booking	Passed
<b>TC-FR-TRV-06-01</b>	Search flights	1. Login 2. Navigate to flights page 3. Enter search criteria 3. search	Valid flight info	Flights displayed from API	Passed
<b>TC-FR-TRV-06-02</b>	Search flights invalid	1. Login 2. Navigate to flights page 3. Enter invalid search criteria 3. search	Random invalid flight	No flights returned / error message	Passed
<b>TC-FR-TRV-07-01</b>	Add to favorites	1. Login 2. Select hotel/destination/activity 3. Click Add to Favorites	Existing item	Item added to favorites list	Passed

<b>TC-FR-TRV-08-01</b>	Remove from favorites	1. Login 2. Select favorite item 3. Click Remove	Existing favorite	Item removed from favorites list	Passed
<b>TC-FR-TRV-09-01</b>	View favorites	1. Login 2. Navigate to Favorites	-	Favorites list displayed	Passed
<b>TC-FR-TRV-10-01</b>	Create manual trip plan	1. Login 2. Navigate to Trip Plan 3. Fill plan info 4. Save	Valid plan info	Trip plan saved and listed	Passed
<b>TC-FR-TRV-10-02</b>	Create trip plan invalid	1. Login 2. Navigate to Trip Plan 3. Enter empty or invalid trip plan 4. Save	Empty fields	Error message "Trip plan cannot be empty"	Passed
<b>TC-FR-TRV-11-01</b>	Create AI trip plan	1. Login 2. Navigate to Trip Plan 3. Use AI suggestion 4. Generate	Valid input	Trip plan generated & saved	Passed
<b>TC-FR-TRV-11-02</b>	Create AI trip plan invalid	1. Login 2. 2. Navigate to Trip Plan 3. Provide invalid AI input 3. Generate	Invalid AI input	Error message displayed, plan not created	
<b>TC-FR-TRV-12-01</b>	Delete trip plan	1. Login 2. Open trip plan 3. Click Delete	Existing trip plan	Trip plan deleted, confirmation shown	
<b>TC-FR-TRV-13-01</b>	View trips plan	1. Login 2. Navigate to Trip Plans	-	List of user's trip plans displayed	
<b>TC-FR-TRV-14-01</b>	View bookings	1. Login 2. Navigate to Bookings	-	User's bookings displayed	Passed
<b>TC-FR-DR-01-01</b>	View assigned bookings	1. Login as driver 2. Navigate to Assigned Bookings	-	Assigned bookings displayed	Passed
<b>TC-FR-DR-02-01</b>	Update booking status	1. Login as driver 2. Open booking 3. Change status.	Valid booking	Booking status updated	Passed
<b>TC-FR-DR-03-01</b>	Receive booking notification	1. Traveller Make a booking 2. Login as driver	-	Driver receives notification of new booking	Passed
<b>TC-FR-ADMIN-01-01</b>	Add destination	1. Login as admin 2. Navigate to Add Destination 3. Fill info 4. Submit	Valid data	Destination added successfully	Passed
<b>TC-FR-ADMIN-01-02</b>	Add destination invalid	1. Login 2. Add destination with empty/invalid data	Invalid data	Error message displayed	Passed

<b>TC-FR-ADMIN-02-01</b>	Edit destination	1. Login as admin 2. choose destination 3. Edit info 4. Save	Valid data	Destination updated	Passed
<b>TC-FR-ADMIN-03-01</b>	Delete destination	1. Login as admin 2. choose destination 3. Click Delete	Existing destination	Destination deleted	Passed
<b>TC-FR-ADMIN-04-01</b>	Add hotel	1. Login as admin 2. Navigate to Add Hotel 3. Fill info 4. Submit	Valid data	Hotel added successfully	Passed
<b>TC-FR-ADMIN-04-02</b>	Add hotel invalid	1. Login 2. Add hotel with invalid/empty data	Invalid data	Error message displayed	Passed
<b>TC-FR-ADMIN-05-01</b>	Edit hotel	1. Login as admin 2. choose hotel 3. Edit info 4. Save	Valid data	Hotel updated	Passed
<b>TC-FR-ADMIN-06-01</b>	Delete hotel	1. Login as admin 2. choose hotel 3. Click Delete	Existing hotel	hotel deleted	Passed
<b>TC-FR-ADMIN-06-02</b>	Delete hotel with upcoming bookings	1. Login as admin 2. choose hotel with upcoming bookings 3. Click Delete	Existing hotel with bookings	Error message displayed	Passed
<b>TC-FR-ADMIN-07-01</b>	View system bookings	1. Login as admin 2. Navigate to Bookings	-	All bookings displayed	Passed
<b>TC-FR-ADMIN-08-01</b>	Add transport service	1. Login as admin 2. Navigate to Add Transport 3. Fill info 4. Submit	Valid data	Transport service added	Passed
<b>TC-FR-ADMIN-09-01</b>	Edit transport service	1. Login as admin 2. Open service 3. Edit info 4. Save	Valid data	Service updated	Passed
<b>TC-FR-ADMIN-10-01</b>	Delete transport service	1. Login as admin 2. Open service 3. Delete	Existing service	Service deleted	Passed
<b>TC-FR-ADMIN-11-01</b>	View transport linked vehicles	1. Login as admin 2. Navigate to Transport Service 3. View vehicles	-	List of linked vehicles displayed	Passed
<b>TC-FR-ADMIN-12-01</b>	Add vehicle	1. Login as admin 2. Navigate to Add Vehicle 3. Fill info 4. Submit	Valid vehicle info	Vehicle added & linked	Passed
<b>TC-FR-ADMIN-12-02</b>	Add vehicle invalid	1. Login 2. Add vehicle with invalid/empty data	Invalid data	Error message displayed	Passed

<b>TC-FR-ADMIN-13-01</b>	Edit Vehicle	1. Login as admin 2. Open vehicle 3. Edit info 4. Save	Valid data	Vehicle updated	Passed
<b>TC-FR-ADMIN-14-01</b>	Delete vehicle	1. Login as admin 2. Open vehicle 3. Delete	Existing vehicle	Vehicle deleted	Passed
<b>TC-FR-ADMIN-14-02</b>	Delete vehicle with upcoming bookings	1. Login as admin 2. Open vehicle 3. Delete	Existing vehicle with upcoming bookings	Error message shown and reject deletion	Passed
<b>TC-FR-ADMIN-15-01</b>	Accept/reject drivers	1. Login as admin 2. Navigate to driver requests 3. Approve/Reject	Pending driver	Driver status updated & email sent	Passed
<b>TC-FR-ADMIN-16-01</b>	View drivers	1. Login as admin 2. Navigate to drivers.	-	All drivers displayed	Passed
<b>TC-FR-ADMIN-17-01</b>	Search drivers	1. Login as admin 2. Enter valid criteria 3. Click Search	Existing driver data	Matching drivers displayed	Passed
<b>TC-FR-ADMIN-17-02</b>	Search drivers no match	1. Login 2. Search with invalid criteria	Random data	Message "No drivers found"	Passed
<b>TC-FR-ADMIN-18-01</b>	Delete driver	1. Login as admin 2. Navigate to drivers. 3. delete a specific driver	Existing approved driver	Driver deleted	Passed
<b>TC-FR-ADMIN-18-02</b>	Delete Driver with not-completed bookings	1. Login as admin 2. Navigate to drivers. 3. delete driver with not-completed bookings	Existing driver with not-completed bookings	Error message shown and reject deletion.	Passed
<b>TC-FR-ADMIN-19-01</b>	Add activity	1. Login as admin 2. Navigate to Add Activity 3. Fill info 4. Submit	Valid activity info	Activity added successfully	Passed
<b>TC-FR-ADMIN-19-02</b>	Add activity invalid	1. Login 2. Add activity with empty/invalid data	Invalid data	Error message displayed	Passed
<b>TC-FR-ADMIN-20-01</b>	Edit activity	1. Login as admin 2. Open activity 3. Edit info 4. Save	Valid data	Activity updated	Passed
<b>TC-FR-ADMIN-21-01</b>	Delete activity	1. Login as admin 2. Open activity 3. Delete	Existing activity	Activity deleted	Passed

Table 57 Test Cases

### 3.6 Table – RTM (Requirements Traceability Matrix)

RTM ID	Requirement ID	Requirement Title	Analysis Section	Code	Test Cases
TVP-FR-A-M-01	FR-SU-01	<a href="#">Login</a>	<a href="#">Specification &amp; Diagrams</a>	<a href="#">GitHub</a>	<a href="#">TC-FR-SU-01-01, TC-FR-SU-01-02, TC-FR-SU-01-03</a>
TVP-FR-A-M-02	FR-SU-02	<a href="#">Logout</a>	<a href="#">Specification &amp; Diagrams</a>	<a href="#">GitHub</a>	<a href="#">TC-FR-SU-02-01</a>
TVP-FR-A-M-03	FR-SU-03	<a href="#">Edit Profile</a>	<a href="#">Specification &amp; Diagrams</a>	<a href="#">GitHub</a>	<a href="#">TC-FR-SU-03-01, TC-FR-SU-03-02</a>
TVP-FR-A-M-04	FR-USER-01	<a href="#">Register</a>	<a href="#">Specification &amp; Diagrams</a>	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-01-01, TC-FR-USER-01-02</a>
TVP-FR-D-M-05	FR-USER-02	<a href="#">View Destinations</a>	<a href="#">Specification &amp; Diagrams</a>	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-02-01</a>
TVP-FR-D-M-06	FR-USER-03	<a href="#">View Destination Details</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-03-01</a>
TVP-FR-D-M-07	FR-USER-04	<a href="#">Search Destination</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-04-01, TC-FR-USER-04-02</a>
TVP-FR-H-M-08	FR-USER-05	<a href="#">View Hotels</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-05-01</a>
TVP-FR-H-M-09	FR-USER-06	<a href="#">View Hotel Details</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-06-01</a>
TVP-FR-H-M-10	FR-USER-07	<a href="#">Search Hotels</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-07-01, TC-FR-USER-07-02</a>
TVP-FR-T-M-11	FR-USER-08	<a href="#">View Transport Services</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-08-01</a>
TVP-FR-ACT-M-12	FR-USER-09	<a href="#">View Activities</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-09-01</a>
TVP-FR-ACT-M-13	FR-USER-10	<a href="#">View Activity Details</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-10-01</a>
TVP-FR-ACT-M-14	FR-USER-10	<a href="#">Search Activities</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-10-02, TC-FR-USER-10-03</a>
TVP-FR-B-M-15	FR-USER-11	<a href="#">Search Bookings</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-USER-11-01, TC-FR-USER-11-02</a>
TVP-FR-TRV-M-16	FR-TRV-01	<a href="#">Book Hotel</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-01-01, TC-FR-TRV-01-02</a>

<b>TVP-FR-TRV-M-17</b>	FR-TRV-02	<a href="#">Search Vehicles</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-02-01, TC-FR-TRV-02-02</a>
<b>TVP-FR-TRV-M-18</b>	FR-TRV-03	<a href="#">Book Vehicle</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-03-01, TC-FR-TRV-03-02</a>
<b>TVP-FR-TRV-M-19</b>	FR-TRV-04	<a href="#">Receive Notifications (Traveller)</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-05-01</a>
<b>TVP-FR-TRV-M-20</b>	FR-TRV-05	<a href="#">Payment</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-04-01</a>
<b>TVP-FR-TRV-M-21</b>	FR-TRV-06	<a href="#">Search Flights</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-06-01, TC-FR-TRV-06-02</a>
<b>TVP-FR-FAV-M-22</b>	FR-TRV-07	<a href="#">Add to Favorites</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-07-01</a>
<b>TVP-FR-FAV-M-23</b>	FR-TRV-08	<a href="#">Remove from Favorites</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-08-01</a>
<b>TVP-FR-FAV-M-24</b>	FR-TRV-09	<a href="#">View Favorites</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-09-01</a>
<b>TVP-FR-PLAN-M-25</b>	FR-TRV-10	<a href="#">Create Manual Trip Plan</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-10-01, TC-FR-TRV-10-02</a>
<b>TVP-FR-PLAN-M-26</b>	FR-TRV-11	<a href="#">Create AI Trip Plan</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-11-01, TC-FR-TRV-11-02</a>
<b>TVP-FR-PLAN-M-27</b>	FR-TRV-12	<a href="#">Delete Trip Plan</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-13-01</a>
<b>TVP-FR-PLAN-M-28</b>	FR-TRV-13	<a href="#">View Trip Plans</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-14-01</a>
<b>TVP-FR-B-M-29</b>	FR-TRV-14	<a href="#">View Bookings</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-TRV-15-01</a>
<b>TVP-FR-DR-M-31</b>	FR-DR-01	<a href="#">View Driver Assigned Bookings</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-DR-01-01</a>
<b>TVP-FR-DR-M-32</b>	FR-DR-02	<a href="#">Update Booking Status</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-DR-02-01</a>
<b>TVP-FR-DR-M-33</b>	FR-DR-03	<a href="#">Receive Notifications (Driver)</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-DR-03-01</a>
<b>TVP-FR-ADM-M-34</b>	FR-ADMIN-01	<a href="#">Add Destination</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-01-01, TC-FR-ADMIN-01-02</a>

<b>TVP-FR-ADM-M-35</b>	FR-ADMIN-02	<a href="#">Edit Destination</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-02-01</a>
<b>TVP-FR-ADM-M-36</b>	FR-ADMIN-03	<a href="#">Delete Destination</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-03-01</a>
<b>TVP-FR-ADM-M-37</b>	FR-ADMIN-04	<a href="#">Add Hotel</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-04-01, TC-FR-ADMIN-04-02</a>
<b>TVP-FR-ADM-M-38</b>	FR-ADMIN-05	<a href="#">Edit Hotel</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-05-01</a>
<b>TVP-FR-ADM-M-39</b>	FR-ADMIN-06	<a href="#">Delete Hotel</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-06-01, TC-FR-ADMIN-06-02</a>
<b>TVP-FR-ADM-M-40</b>	FR-ADMIN-07	<a href="#">View System Bookings</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-07-01</a>
<b>TVP-FR-ADM-M-41</b>	FR-ADMIN-08	<a href="#">Add Transport Service</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-08-01</a>
<b>TVP-FR-ADM-M-42</b>	FR-ADMIN-09	<a href="#">Edit Transport Service</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-09-01</a>
<b>TVP-FR-ADM-M-43</b>	FR-ADMIN-10	<a href="#">Delete Transport Service</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-10-01</a>
<b>TVP-FR-ADM-M-44</b>	FR-ADMIN-11	<a href="#">View Transport Linked Vehicles</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-11-01</a>
<b>TVP-FR-ADM-M-45</b>	FR-ADMIN-12	<a href="#">Add Vehicle</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-12-01, TC-FR-ADMIN-12-02</a>
<b>TVP-FR-ADM-M-46</b>	FR-ADMIN-13	<a href="#">Edit Vehicle</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-13-01</a>
<b>TVP-FR-ADM-M-47</b>	FR-ADMIN-14	<a href="#">Delete Vehicle</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-14-01, TC-FR-ADMIN-14-02</a>
<b>TVP-FR-ADM-M-48</b>	FR-ADMIN-15	<a href="#">Accept/Reject Drivers</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-15-01</a>
<b>TVP-FR-ADM-M-49</b>	FR-ADMIN-16	<a href="#">View Drivers</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-16-01</a>
<b>TVP-FR-ADM-M-50</b>	FR-ADMIN-17	<a href="#">Search Drivers</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-17-01, TC-FR-ADMIN-17-02</a>
<b>TVP-FR-ADM-M-51</b>	FR-ADMIN-18	<a href="#">Delete Driver</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-18-01, TC-FR-ADMIN-18-02</a>
<b>TVP-FR-ADM-M-52</b>	FR-ADMIN-19	<a href="#">Add Activity</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-19-01, TC-FR-ADMIN-19-02</a>

<b>TVP-FR-ADM-M-53</b>	FR-ADMIN-20	<a href="#">Edit Activity</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-20-01</a>
<b>TVP-FR-ADM-M-54</b>	FR-ADMIN-21	<a href="#">Delete Activity</a>	Specification & Diagrams	<a href="#">GitHub</a>	<a href="#">TC-FR-ADMIN-21-01</a>

*RTM (Requirements Traceability Matrix) ^ A Table*

## **3.7 Future Work and Conclusion:**

### **3.7.1 Future Work:**

To enhance the value proposition of the Travel Planning platform and ensure its continued relevance and competitiveness in the digital travel market, the project can be expanded to include a set of advanced features and structural improvements. The proposed future work focuses on broadening the scope of services, deepening personalization, and optimizing the user experience across various devices.

#### ***3.7.1.1 Enhancing Artificial Intelligence and Personalization:***

##### **- Advanced Recommendation System:**

Transitioning from rule-based planning to a machine learning-driven recommendation system that utilizes collaborative filtering algorithms. This will enable the platform to offer highly personalized suggestions for activities, restaurants, and accommodations based on the travel history of similar users and their past preferences.

#### ***3.7.1.2 Expanding Financial and Logistics Services:***

##### **1. Integration of Additional Payment Gateways:**

Incorporating other global payment options, such as Stripe or Apple Pay, to increase the flexibility of financial transactions and cater to diverse user preferences.

##### **2. Global Car Rental Integration:**

Expanding transportation services to include integration with major international car rental platforms, providing wider options for users who prefer self-driving.

#### ***3.7.1.3 Expanding Service Offerings and Management:***

##### **1. Tour Guide Integration and Management:**

Implementing a dedicated module to onboard, verify, and manage professional Tour Guides. This includes a profile system for guides, a rating and review mechanism, and a scheduling interface to allow users to book guides directly for specific destinations or activities.

##### **2. Comprehensive Activity Booking System:**

Developing a full-fledged booking and reservation system for individual Activities (e.g., museum tickets, tours, adventure sports). This requires integration with activity providers and a robust inventory management system.

##### **3. Pre-packaged Guided Trips:**

Creating a catalog of Pre-planned, Ready-to-Book Trips that include all logistics (transport, accommodation, activities) and are bundled with a professional tour guide. This feature will cater to users seeking seamless, curated travel experiences.

##### **4. Flexible Cancellation and Refund Policies:**

Establishing and implementing clear, automated Cancellation Policies for all bookings (hotels, transport, activities, and full trips). The system must handle refund calculations and processing based on defined policy rules, ensuring transparency and user trust.

## **5. User Feedback and Complaint System:**

Introducing a dedicated system for Complaints and Suggestions. This module will allow users to submit structured feedback, which will be routed to the appropriate administrative or service team for tracking and resolution, thereby facilitating continuous service improvement.

### **3.7.1.3 Data Analytics and Administrative Tools:**

#### **1. Administrative Dashboard and System Charts:**

Developing an advanced Admin Dashboard featuring real-time Charts and Visualizations. These tools will allow administrators to track key performance indicators (KPIs) such as:

- Total bookings and revenue over time.
- User activity and engagement metrics.
- Performance and ratings of drivers and tour guides.
- Complaint resolution status and trends.
- This data-driven approach will be crucial for operational oversight and strategic decision-making.

## **3.8 Conclusion:**

The Travel Planning project represents a comprehensive and innovative solution to the challenges inherent in traditional travel organization. The project successfully achieved its primary goal of integrating multiple travel services—ranging from destination exploration and AI-assisted planning to booking and payment—with a single, unified web platform.

The platform is built upon a robust architectural foundation using the Laravel framework, with MVC. This approach has resulted in clean, maintainable, and flexible code that is ready for future expansion. Furthermore, the project demonstrated a high capability for integration with vital external services, including payment gateways (PayPal) and Artificial Intelligence APIs (Groq) for generating personalized travel itineraries.

The unique combination of manual and AI-supported planning, alongside the integrated transportation and driver management system, positions Travel Planning as a powerful tool designed to simplify the user experience and save significant time and effort. With the ambitious future work planned—including the integration of tour guides, advanced data analytics, and a comprehensive activity booking system—the project holds substantial potential to become a leading, full-service platform in the smart tourism sector. This project has proven its technical viability and its ability to meet the specified functional and non-functional requirements, establishing a strong foundation for a successful commercial venture.