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


# Hajj and Umrah Booking and Trip Management System
# Senior Project

**Prepared by**

**Ahmad Kassem  Wael Ali  Hasan mnjed**

**Supervised by**

**Dr.Shadi bleidi**

# Contents

# Abstract

In the era of rapid technological advancement and digital transformation in religious services, the **Hajj and Umrah Booking System** emerges as an innovative solution designed to simplify and manage religious travel arrangements for individuals and organizations alike. This system provides an integrated digital platform that allows users to book trips, manage accommodations and hotels, and track travel details easily and securely.

The platform offers a unified environment enabling pilgrims and visitors to browse available packages, book trips, hotels, and cities, and manage their reservations through an intuitive and efficient interface. Administrators and service providers are granted extended privileges to manage cities, hotels, and trips while ensuring the quality of services provided.

The project aims to support digital transformation in the field of religious tourism and enhance user experience by simplifying booking procedures, ensuring continuous information updates, and improving operational efficiency for service providers through flexible and effective data management.

Through this system, full integration can be achieved across all stages of the Hajj or Umrah journey — from registration and verification to booking and accommodation management — ensuring a smooth, organized, and spiritually fulfilling experience for every pilgrim.

# Chapter 1
# Introduction

## 1.1 Purpose

The Hajj and Umrah Trip Management System aims to create a safe and user-friendly digital platform that enables pilgrims to explore, book, and manage pilgrimage trips efficiently.
The system seeks to simplify the booking process for Hajj and Umrah journeys by providing access to detailed trip information — including hotel data, city destinations, duration, and cost — and allowing online reservations.
It also provides an administrative dashboard for managing trips, hotels, and destinations, ensuring smooth coordination between users and administrators.

---

## 1.2 Project Scope

The Hajj and Umrah Booking System is designed to provide a comprehensive digital environment for managing pilgrimage trips.
It allows users to explore available Hajj and Umrah packages, book trips online, and manage their reservations through an intuitive web interface.
Administrators can manage trip data, hotel information, city destinations, and website content through a dedicated dashboard.
A super administrator has additional privileges to monitor platform activity and control administrator access.
The project focuses on improving user experience, reliability, and administrative efficiency in organizing religious journeys.

---

### 1.2.1 First Phase (MVP – Minimum Viable Product)

The first version of the system focuses on developing the essential functions that ensure a complete and usable core experience for both users and administrators:

- **User Account Management:**
  - User registration, login, logout, and profile management.
- **Trip Browsing:**
  - View available Hajj and Umrah trips with details (name, price, duration, hotel, and destination).

- **Trip Booking:**
  - Reserve a selected trip, specify ticket quantity, and confirm booking.
  - View booking history and track reservation status (Pending / Accepted / Canceled).
- **Admin Management:**
  - Add, edit, delete, and view trips.
  - Manage hotels and city destinations.
  - Monitor and update user reservations.

- **Hotel and City Management:**
  - Add and manage hotels with details such as name, location, and room information.
  - Add and manage cities and associate them with trips.

---

### 1.2.2 Second Phase

- Introduce a **Super Admin role** for managing administrators' access and monitoring the system.
- Enable **online payment options** for secure financial transactions.
- Integrate **Google Maps API** to display hotel locations and trip destinations.
- Add **notification and alert features** to keep users informed about trip updates and booking confirmations.

---

### 1.2.3 Third Phase

- Implement **secure e-payment gateways** for integrated financial transactions.
- Add **statistical reports** for administrators to analyze bookings and performance.
- Enhance **system performance and scalability** to support more users and trips.

## 1.3 Technical Scope

The technical scope of the **Hajj and Umrah Trip Management System** includes all the technological components necessary to build a secure, interactive, and scalable web-based platform.
The system has been developed using **modern web technologies** to ensure reliability, maintainability, and flexibility for future enhancements.

---

### 1.3.1 Frontend (User Interface)

The system's frontend was developed using:

- React, Tailwind Css, Vite
  These technologies were used to build responsive and user-friendly interfaces that allow users to browse trips, view details, and make bookings easily.
  The design focuses on **clarity, simplicity, and accessibility**, ensuring a smooth user experience across all devices.

---

### 1.3.2 Backend (Server Side)

- The backend was developed using the **Laravel Framework (PHP)**.
  Laravel provides a **robust MVC (Model-View-Controller) structure** that enhances system organization and scalability.
  It supports essential features such as:
  - Secure authentication and session management.
  - Efficient handling of trip, hotel, and reservation data.
  - Integration with third-party services such as Google Maps and email notifications.
    Laravel's built-in routing and ORM (Eloquent) ensure clean and maintainable code with strong performance.

---

### 1.3.3 Database

- The system uses **MySQL** as its primary **relational database management system (RDBMS)**.
  It stores structured data related to users, trips, hotels, cities, and bookings.
  Proper indexing and relational design have been implemented to ensure fast data retrieval and integrity.

---

### 1.3.4 Map Integration

- The system integrates **Google Maps API** to display hotel locations and trip destinations.
  This feature allows users to view geographical locations and choose suitable accommodations based on proximity and preference.

---

### 1.3.5 Security and Data Protection

- Secure login and role-based access (User, Admin, Super Admin).
- Encrypted passwords and sensitive data.
- HTTPS encryption for secure communication between client and server.
- Input validation and regular backups to ensure data safety.

---

### 1.3.6 Development Tools

- **Visual Studio Code (VS Code):** main development environment.
- **XAMPP / PHP Artisan:** for local server and Laravel environment setup.
- **phpMyAdmin:** for database management.
- **GitHub:** for version control and collaboration.

---

### 1.3.7 Future Enhancements

- Integration with **payment gateways** for online payments.
- **Mobile application version** for Android and iOS.
- Advanced **reporting and analytics dashboard** for administrators.
- **AI-based trip recommendation system** for personalized suggestions.

# Chapter2
# Project management

## 2.1 Project Overview

The **Hajj and Umrah Trip Management System** is a web-based platform designed to simplify the process of exploring, booking, and managing pilgrimage trips for users while providing administrative tools for managing trip, hotel, and city data.
The project focuses on providing a user-friendly interface, reliability, and security for both users and administrators.

The system was developed under the supervision of **Mr. Shadi Bleidi**, who guided the planning, design, and implementation phases.

---

## 2.2 Project Charter

| | |
|---|---|
| **Project Title** | **Hajj and Umrah Trip Management System** |
| **Supervisor** | Mr. Shadi Bleidi |
| **Project Type** | Web Site |
| **Development Methodology** | Incremental Model |
| **Programming Languages** | PHP, React, Tailwind Css, Vite |
| **Frameworks & Tools** | Laravel, Bootstrap, MySQL, VS Code, XAMPP |
| **Target Users** | Pilgrims, Administrators, Super Admin |
| **Main Objective** | To design and implement a digital platform for managing Hajj and Umrah trips efficiently. |
| **Start Date** | [November 2025] |
| **End Date** | [February 2026] |

---

## 2.3 Project Objectives

1. Facilitate the process of booking Hajj and Umrah trips online.
2. Provide detailed trip, hotel, and city information for users.
3. Enable administrators to manage trips, hotels, and reservations efficiently.
4. Implement a secure login and user management system.
5. Integrate Google Maps for location-based trip exploration.
6. Support notifications and alerts for users and admins.
7. Ensure system scalability and future expansion for new features.

---

## 2.4 Development Methodology – Incremental Model

The **Incremental Software Development Model** was chosen for this project.
This approach divides the system into multiple incremental modules, each developed, tested, and delivered independently.
Each increment adds new functionality while maintaining full compatibility with previous modules.

**Advantages of the Incremental Model in this project:**

- Early delivery of core functions (MVP).
- Reduced development risk by validating each stage.
- Flexibility to incorporate feedback from users and supervisors.
- Easier maintenance and future feature expansion.

**Increments planned for this project:**

- **Increment 1:** User account management and trip browsing.
- **Increment 2:** Trip booking, hotel and city management.
- **Increment 3:** Admin and Super Admin dashboards.
- **Increment 4:** Online payment integration and reporting module.

---

## 2.5 Stakeholders

| Stakeholder | Role / Responsibility |
|---|---|
| Project Supervisor | Provide academic and technical guidance (Mr. Shadi Bleidi) |
| Developers | Design, implement, and test system modules |
| Users (Pilgrims) | Explore and book Hajj & Umrah trips |
| Admins | Manage trips, hotels, cities, and user bookings |
| Super Admin | Monitor the system, assign roles, and control access |

---

## 2.6 Work Breakdown Structure (WBS)

1. **Requirement Analysis**
   - Collect system requirements
   - Define system functionalities
   - Prepare SRS document
2. **System Design**
   - Design UI mockups
   - Create database schema
   - Design system architecture
3. **Implementation (Incremental)**
   - Increment 1: Authentication and user interface
   - Increment 2: Trip and booking management
   - Increment 3: Admin dashboard and reporting
4. **Testing & Validation**
   - Unit testing and integration testing
   - Bug fixing and optimization
5. **Deployment & Maintenance**
   - Host system on web server
   - Conduct user training and final evaluation

---

## 2.7 Risk Management

| Risk | Probability | Impact | Mitigation Strategy |
|---|---|---|---|
| Delay in development | Medium | High | Apply incremental delivery and schedule reviews |
| Data loss or corruption | Low | High | Implement regular database backups |
| Integration issues with APIs | Medium | Medium | Conduct early testing of external APIs |
| Security vulnerabilities | Medium | High | Use encryption, HTTPS, and role-based access |
| Server or hosting failure | Low | Medium | Maintain backups and use reliable hosting service |

---

## 2.8 Project Schedule

A **Gantt chart** was prepared to illustrate the timeline of tasks, including analysis, design, development (per increment), testing, and deployment phases.
Each phase overlaps slightly to maintain progress and support incremental delivery.


**Gantt Chart – Figure (1)**

# Chapter 3
# Fundamental Concepts and Literature Review

## 3.1 Introduction

This chapter presents the fundamental concepts underlying the Hajj and Umrah Trip Management System and provides a comprehensive literature review. The purpose is to establish a theoretical foundation that guided the design, structure, and development of the system. The chapter includes an examination of digital platforms, usability principles, web technologies, and existing pilgrimage-related systems.

---

## 3.2 Fundamental Concepts

### 3.2.1 Digital Trip Management Systems

Digital trip management systems enable users to browse, book, and manage travel services through interactive online platforms. These systems typically include trip listings, hotel information, reservation modules, payment integration, and user dashboards. Digital transformation has significantly enhanced trip management, particularly in large-scale activities like Hajj and Umrah (Saudi Ministry of Hajj and Umrah, 2021).

### 3.2.2 Hajj and Umrah Digital Services

Pilgrimage services require a reliable and well-structured system that organizes trip schedules, hotel stays, transportation, and packages. Digital solutions improve accessibility, accuracy, and efficiency, supporting millions of pilgrims annually.

### 3.2.3 User Roles

The system defines three main roles:

- **User:** Browses trips, views details, manages profile, and books trips.
- **Admin:** Manages trips, hotels, cities, reservations, and system content.
- **Super Admin:** Controls administrative access and oversees all system operations.

### 3.2.4 Reservation Workflow

The reservation workflow follows a standard structured process:

1. User selects a trip
2. User reviews trip details
3. User completes an online reservation
4. Admin reviews, approves, or rejects the booking
5. User receives booking status notifications
   This workflow aligns with the procedures used by major booking platforms such as UmrahBooking.com (2023).

### 3.2.5 Incremental Development Model

The system was developed using the **Incremental Model**, where modules are built and delivered in increments.
Benefits include:

- Early delivery of essential features
- Reduced development risk
- Easier feedback incorporation
- High system scalability
  This approach follows IEEE recommendations for evolutionary software development (IEEE, 2004).

---

## 3.3 Literature Review

### 3.3.1 Reference Platforms

One of the platforms referenced in the project documentation is the **Expo Saudi Arabia Platform**, which features:

- Organized service structure
- High-quality UI/UX
- Clear navigation
- Effective content presentation
  (Expo2030SaudiArabia.com, 2023)

These elements inspired the trip browsing and content organization in the developed system.

### 3.3.2 Existing Pilgrimage Booking Platforms

Platforms such as **UmrahBooking.com** offer features including:

- Umrah packages
- Hotel listings
- Online reservations
  These platforms highlight industry standards but often lack:
- A multi-role permission system
- A dedicated Super Admin control
- A scalable incremental architecture
  The proposed system addresses these gaps.

### 3.3.3 Technical Foundation

The technical literature that supported the system includes:

- **Laravel Documentation:** backend security, authentication, routing, ORM
- **MySQL Documentation:** relational data modeling and optimization
- **Google Maps API Documentation:** integrating map-based location services
- **Nielsen's Usability Principles (1994):** emphasizing intuitive, accessible, and responsive UI design
- **Travel System Usability Research:** emphasizes importance of user-centered design (Alsmadi & Prybutok, 2018)

---

## 3.4 Summary

This chapter presented the essential concepts and relevant literature that shaped the Hajj and Umrah Trip Management System. By analyzing digital platforms, academic studies, and technological documentation, the system was designed using proven practices that enhance usability, reliability, and scalability.

| النظام المقترح (مشروعكم) | haj.gov.sa / Nusuk Platform | UmrahBooking.com & Similar Platforms | Expo Saudi Arabia Platform | الميزة |
|---|---|---|---|---|
| نعم (إدارة رحلات، فنادق، مدن بشكل متكامل) | نعم (حزم شاملة رسمية) | جزئيًا (حزم عمرة وفنادق) | نعم (منظم جيد، إلهام لتنظيم المحتوى) | **هيكل خدمات منظم وواضح** |
| نعم (Nielsen's مستوحاة من Principles لتكون responsive و intuitive) | جيدة (رسمية وآمنة) | متوسطة (تركيز على الحجز البسيط) | نعم (عالية الجودة، تنقل واضح) | **واجهة مستخدم عالية الجودة (UI/UX)** |
| نعم (تصفح رحلات مع صور، خرائط، ليالي في مكة/مدينة) | نعم (حزم حج شاملة: إقامة، نقل، تأشيرات) | نعم (حزم عمرة، فنادق، نقل) | محدود (معارض وفعاليات) | **تصفح رحلات/حزم مع تفاصيل (تواريخ، فنادق، تكلفة)** |
| حجز تذاكر، إدارة ) نعم حجوزات pending/accepted) | نعم (حجز حزم حج رسمية مع تأشيرات) | نعم (حجز فنادق، نقل، حزم مخصصة) | لا | **حجز عبر الإنترنت** |
| نعم (User vs Admin vs Super Admin) | محدود (رسمي للحجوزات) | (فقط B2C غالباً) نادرًا | لا | **إدارة متعددة الأدوار (Multi-role Permissions)** |
| نعم (إدارة رحلات، فنادق، مدن، حجوزات، محتوى) | نعم (للإدارة الرسمية) | جزئيًا (للوكلاء في بعض الأنظمة) | لا | **لوحة تحكم إدارية مخصصة (Dedicated Admin Dashboard)** |
| نعم (Super Admin للتحكم في حسابات الـ Admins) | لا | لا | لا | **Super Admin تحكم (لإدارة الوصول)** |
| نعم (عرض مواقع فنادق على خريطة) | نعم (مواقع مقدسة) | جزئيًا (مواقع فنادق) | محدود | **Google) تكامل خرائط Maps)** |
| هندسة قابلة للتوسع، ) عالية كما في Agile Scrum (الوثيقة | عالية (رسمية لملايين الحجاج) | متوسطة (حزم ثابتة غالبًا) | جيدة (للمعارض) | **قابلية التوسع والمرونة** |
| نعم (Nielsen مستوحى من usability) ودراسات 1994 | جيدة | متوسطة | نعم (إلهام رئيسي) | **التركيز على سهولة الاستخدام (Usability Principles)** |

## 3.5 References


1. Saudi Ministry of Hajj and Umrah. https://www.haj.gov.sa
2. Expo Saudi Arabia Platform. https://www.expo2030saudiarabia.com
3. UmrahBooking.com. https://www.umrahbooking.com
4. Nielsen, J. (1994). *Usability Engineering*. Academic Press.
5. W3C. *Responsive Web Design Standards*. https://www.w3.org/TR/css-flexbox-1/
6. Laravel Documentation. https://laravel.com/docs
7. MySQL Documentation. https://dev.mysql.com/doc/
8. Google Maps API Documentation. https://developers.google.com/maps/documentation
9. IEEE (2004). *Software Development Models*. https://ieeexplore.ieee.org/document/1702235
10. Alsmadi, I., & Prybutok, V. (2018). *Developing and testing a travel system usability model.* https://doi.org/10.1016/j.ijinfomgt.2018.01.010

# Chapter 4
# System Analysis

## 4.1 Introduction

This chapter presents a detailed system analysis for the Hajj and Umrah Trip Management System. It defines the system's overall behavior, identifies stakeholders and users, and specifies both functional and non-functional requirements. The analysis ensures a clear understanding of system operations before proceeding to design and implementation phases.

---

## 4.2 System Overview

The Hajj and Umrah Trip Management System is a web-based platform designed to facilitate the exploration, booking, and management of pilgrimage trips. The system allows users to view available trips, review hotel and destination information, and make reservations online.

Administrators manage trip data, reservations, hotels, and cities through a dedicated dashboard, while the Super Admin oversees administrative access and system-wide configurations.

---

## 4.3 Stakeholders

The primary stakeholders involved in the system are:

- **Pilgrims (Users):** End-users who browse trips and make reservations.
- **System Administrators:** Manage operational data and reservations.
- **Super Administrator:** Oversees system security, permissions, and administration.
- **Project Supervisor:** Shadi Blaidy.
- **Development Team:** Responsible for implementation and maintenance.

---

## 4.4 User Classes and Characteristics

### 4.4.1 User (Pilgrim)

- Basic technical knowledge
- Uses the system to browse trips, view details, and make bookings

- Requires a simple, intuitive interface

### 4.4.2 Admin

- Moderate to advanced technical knowledge
- Manages trips, hotels, cities, reservations, and content
- Requires efficient management tools and data accuracy

### 4.4.3 Super Admin

- Advanced technical knowledge
- Manages admins, permissions, and system settings
- Responsible for system reliability and security

---

## 4.5 Operating Environment

The system operates in a web-based environment with the following specifications:

- **Frontend:** HTML, CSS, JavaScript, Bootstrap
- **Backend:** Laravel (PHP Framework)
- **Database:** MySQL
- **Server:** Web server compatible with PHP (Apache / Nginx)
- **APIs:** Google Maps API (for location-based services)
- **Browsers:** Chrome, Firefox, Edge, Safari

---

## 4.6 Functional Requirements

### 4.6.1 Authentication and Account Management

- The system shall allow users to register and log in.
- The system shall allow users to update personal profile information.
- The system shall provide role-based access control.

---

### 4.6.2 Trip Management

- The system shall allow admins to add, edit, and delete trips.

- The system shall display trip details including schedule, price, and availability.
- The system shall allow users to view all available trips.

---

### 4.6.3 Hotel and City Management

- The system shall allow admins to manage hotel information.
- The system shall allow admins to manage city and destination data.
- The system shall associate trips with specific hotels and cities.

---

### 4.6.4 Reservation Management

- The system shall allow users to book trips online.
- The system shall allow admins to approve, reject, or update reservations.
- The system shall notify users of reservation status updates.

---

### 4.6.5 Content Management

- The system shall allow admins to manage website content.
- The system shall allow the Super Admin to control admin access.

---

## 4.7 Non-Functional Requirements

### 4.7.1 Performance

- The system shall support multiple concurrent users.
- System response time should not exceed 3 seconds under normal load.

### 4.7.2 Security

- The system shall protect user data through authentication and authorization.
- Passwords shall be encrypted.
- Access shall be restricted based on user roles.

### 4.7.3 Usability

- The system shall provide a user-friendly and intuitive interface.
- The system shall be accessible on multiple devices.

### 4.7.4 Scalability

- The system shall support future expansion such as online payments and advanced reports.

### 4.7.5 Reliability

- The system shall ensure data integrity and minimize downtime.

---

## 4.8 Assumptions and Constraints

### 4.8.1 Assumptions

- Users have access to the internet.
- Users possess basic web browsing skills.

### 4.8.2 Constraints

- The system is limited to web-based deployment.
- Development timeline follows the incremental development methodology.

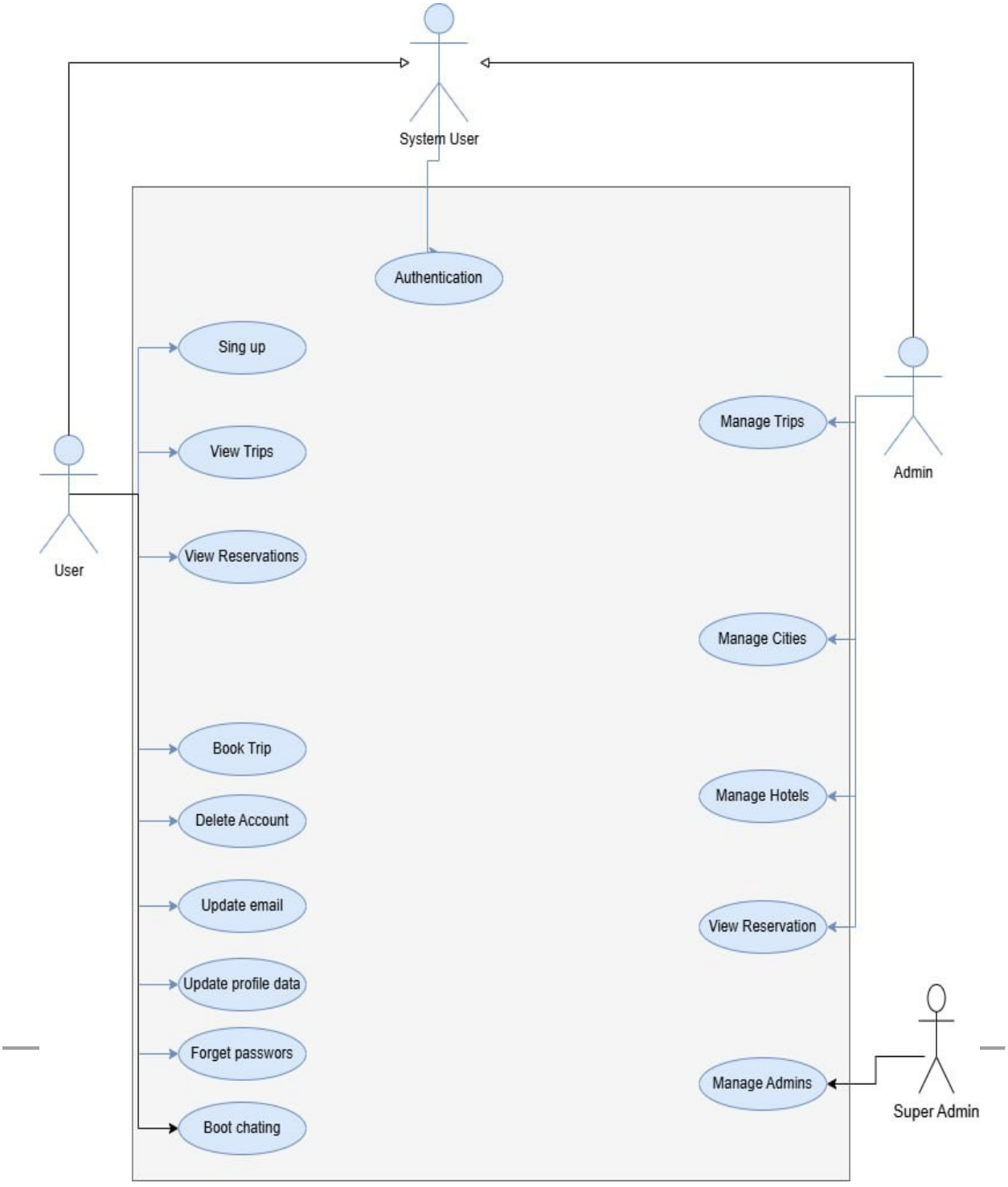
---

## 4.9 Use Case Diagram (Overview)

The system supports the following main use cases:

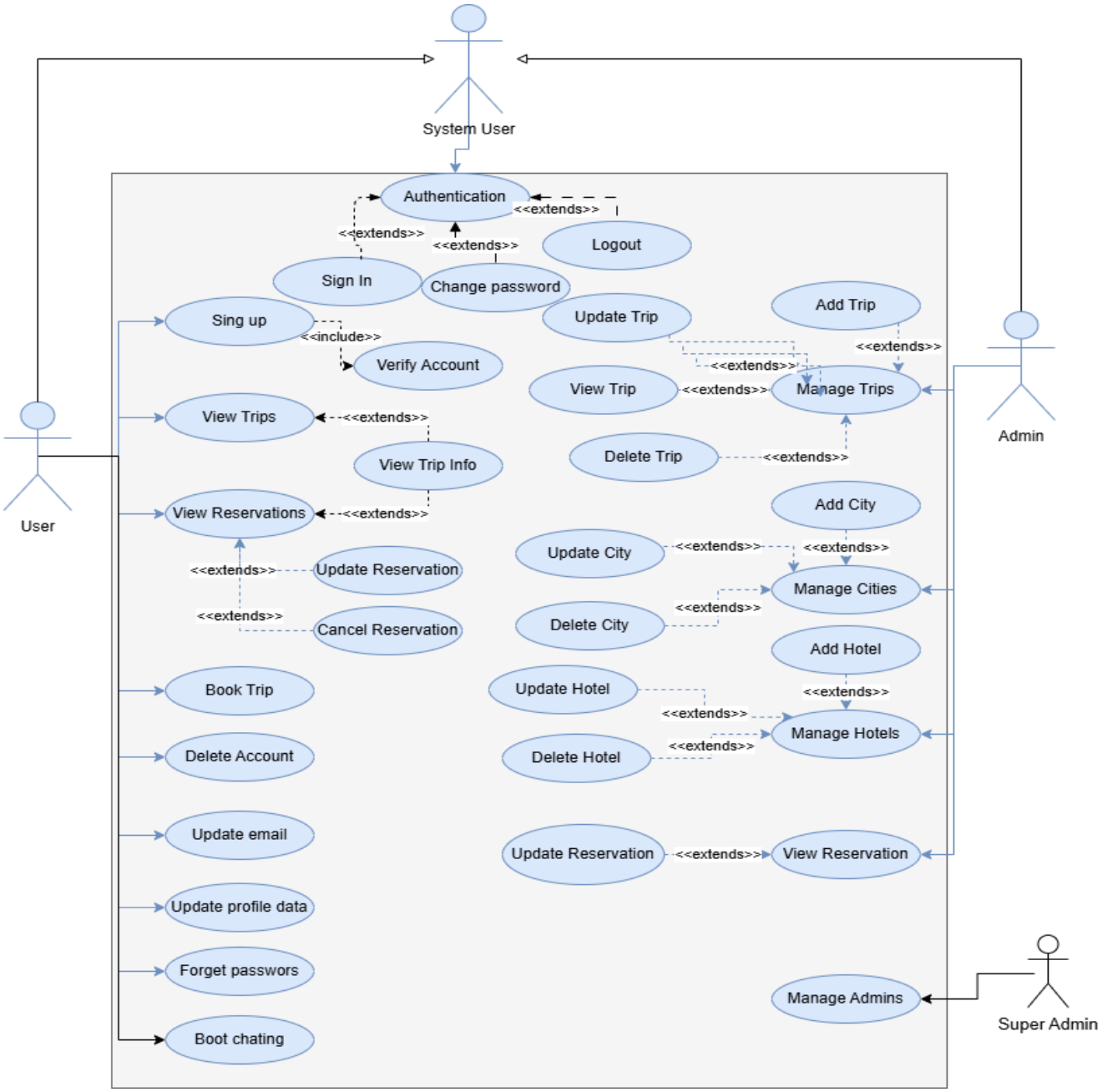
- Register / Login
- Browse Trips
- View Trip Details
- Book Trip
- Manage Trips (Admin)
- Manage Hotels and Cities (Admin)
- Manage Reservations
- Manage Admins (Super Admin)

### 4.9.1 Use Case Diagram (High-Level)




**Use Case Diagram (High-Level-) – Figure 2**

## 4.9.2 Use Case Diagram (Low-Level)



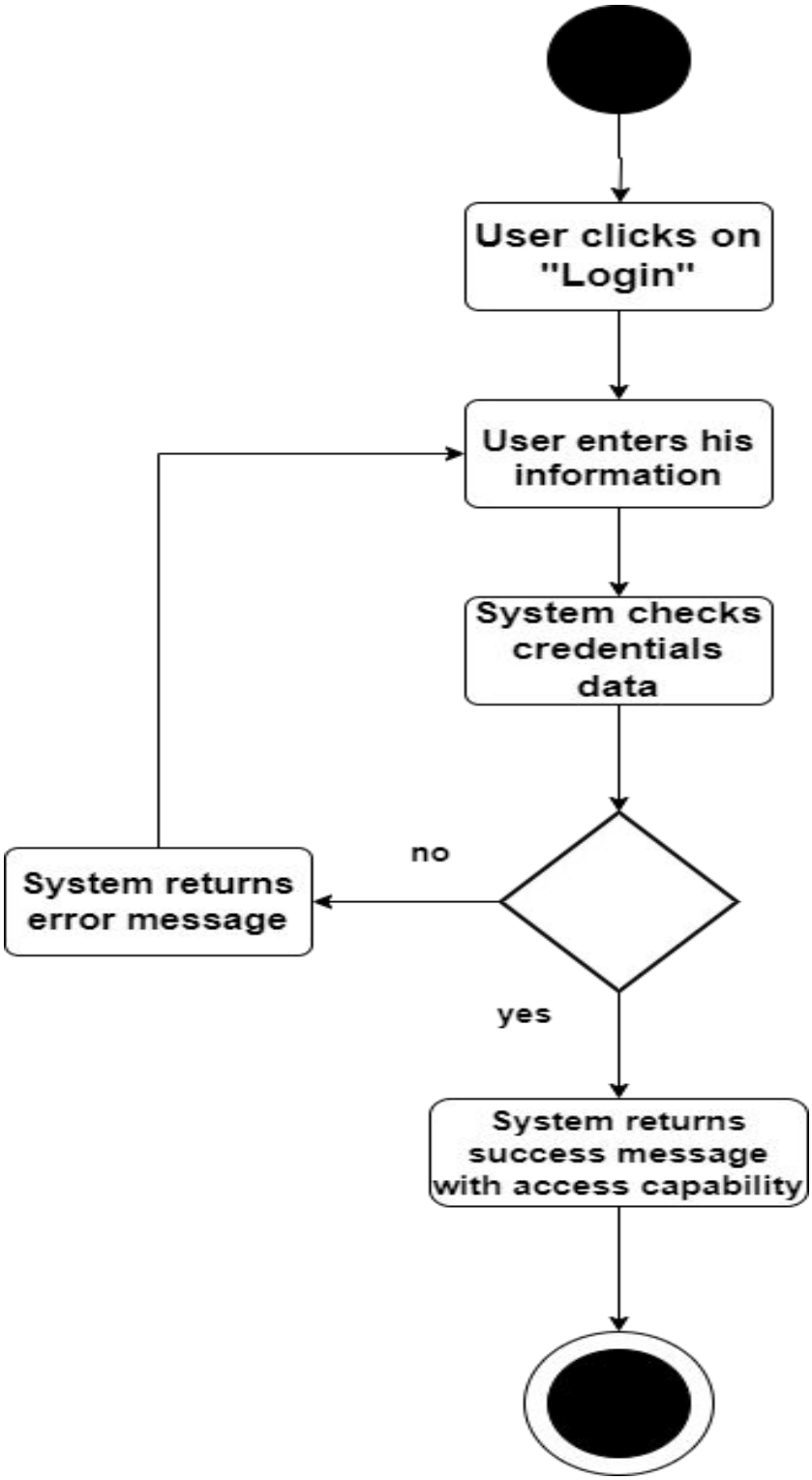

**Use Case Diagram (Low-Level-) – Figure 3**

# 5. Use Case Specification & Activity Diagrams & Sequence Diagrams

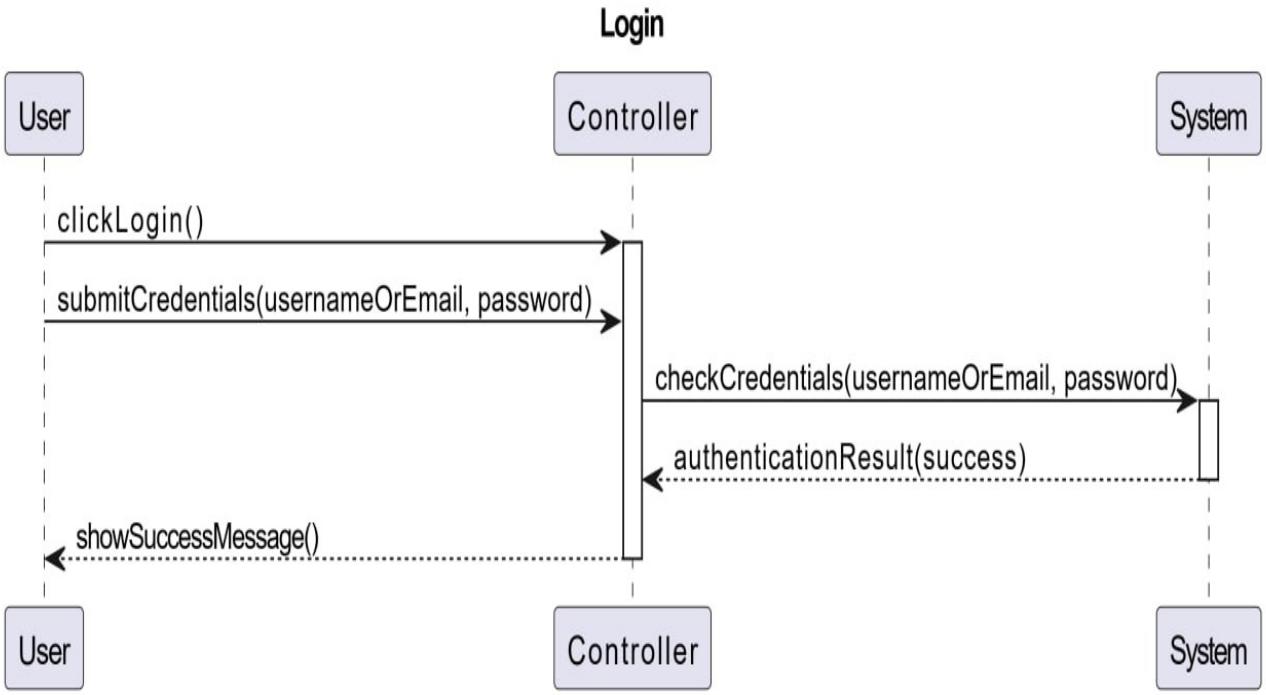
## 5.1. System-User Requirements

### 5.1.1. Sign in

| | |
|---|---|
| **Use Case ID** | UC-1 |
| **Use Case** | Sign In |
| **Actor** | System User |
| **Pre-Condition** | Has an account in the system |
| **Main Scenario** | 1. User clicks on “Login”<br>2. User enter (username or email) and password<br>3. System checks credentials data<br>4. System returns success message with capability for access to the system |
| **Alternative Scenario** | In step 3 if the credentials data is incorrect the system returns error message and redirect user to login again |



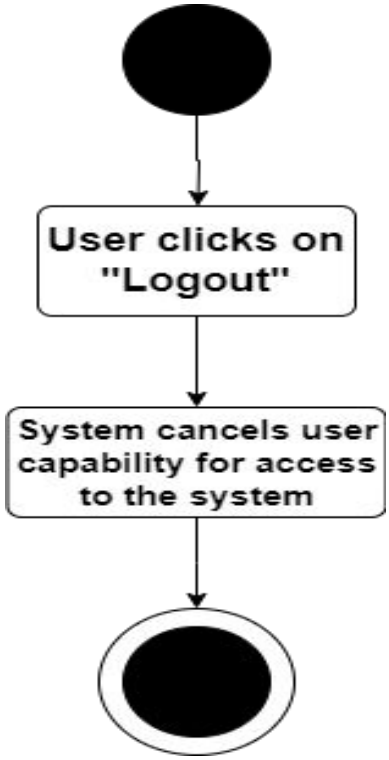

**Activity diagram(log in) – Figure 4**



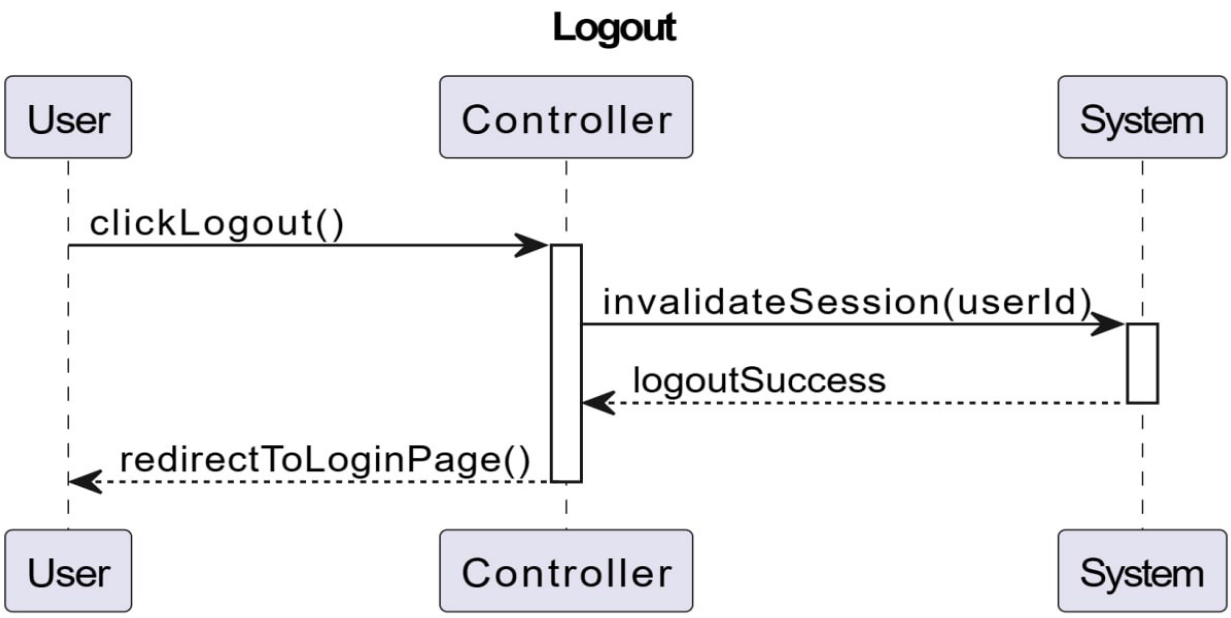

**Sequence diagram(log in) – Figure 5**

5.1.2. Log out

| | |
|---|---|
| **Use Case ID** | UC-2 |
| **Use Case** | Logout |
| **Actor** | System User |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Logout”<br>2. System cancels user capability for access to the system |
| **Alternative Scenario** | _____ |



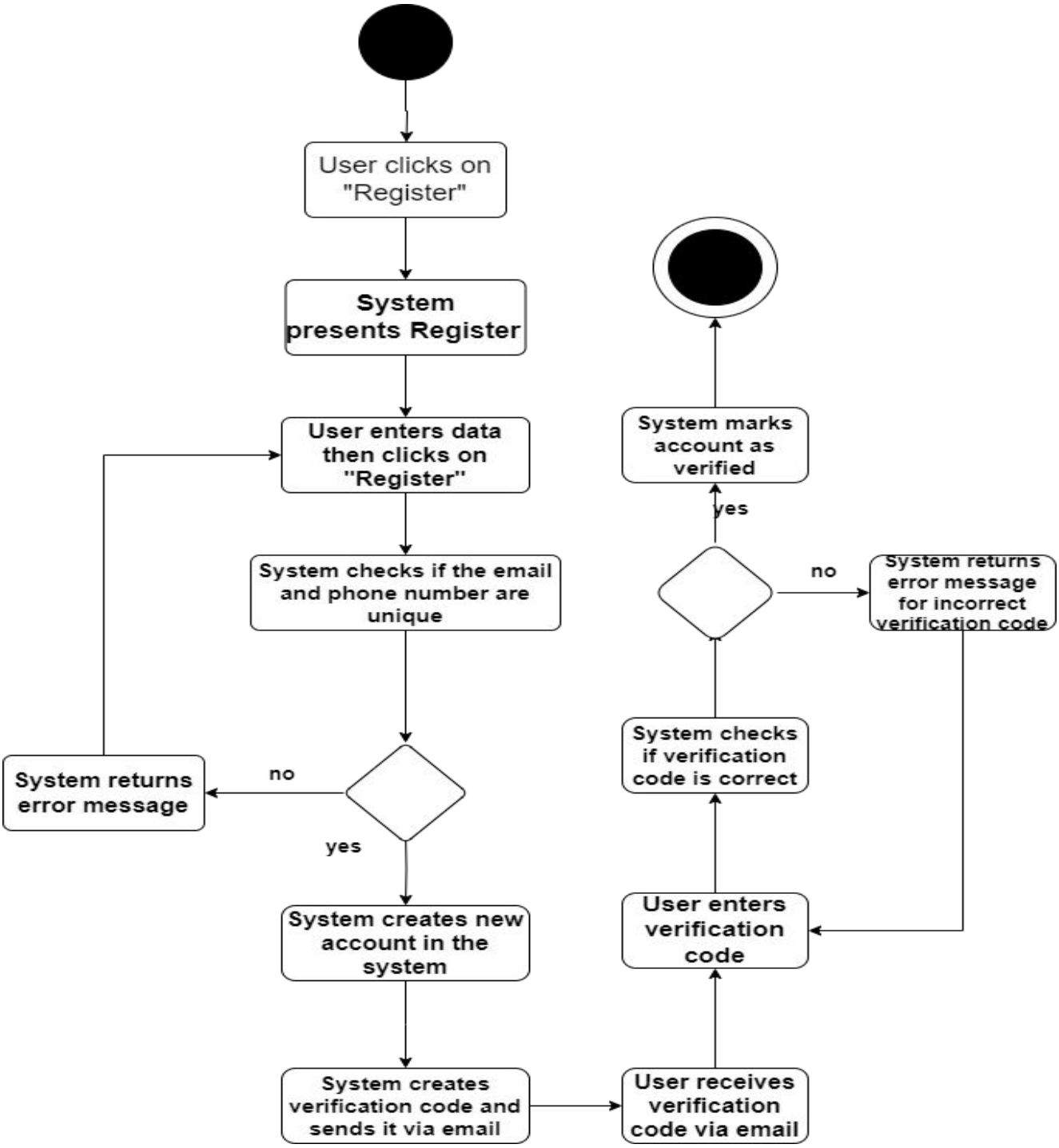

**Activity diagram(log out) – Figure 6**



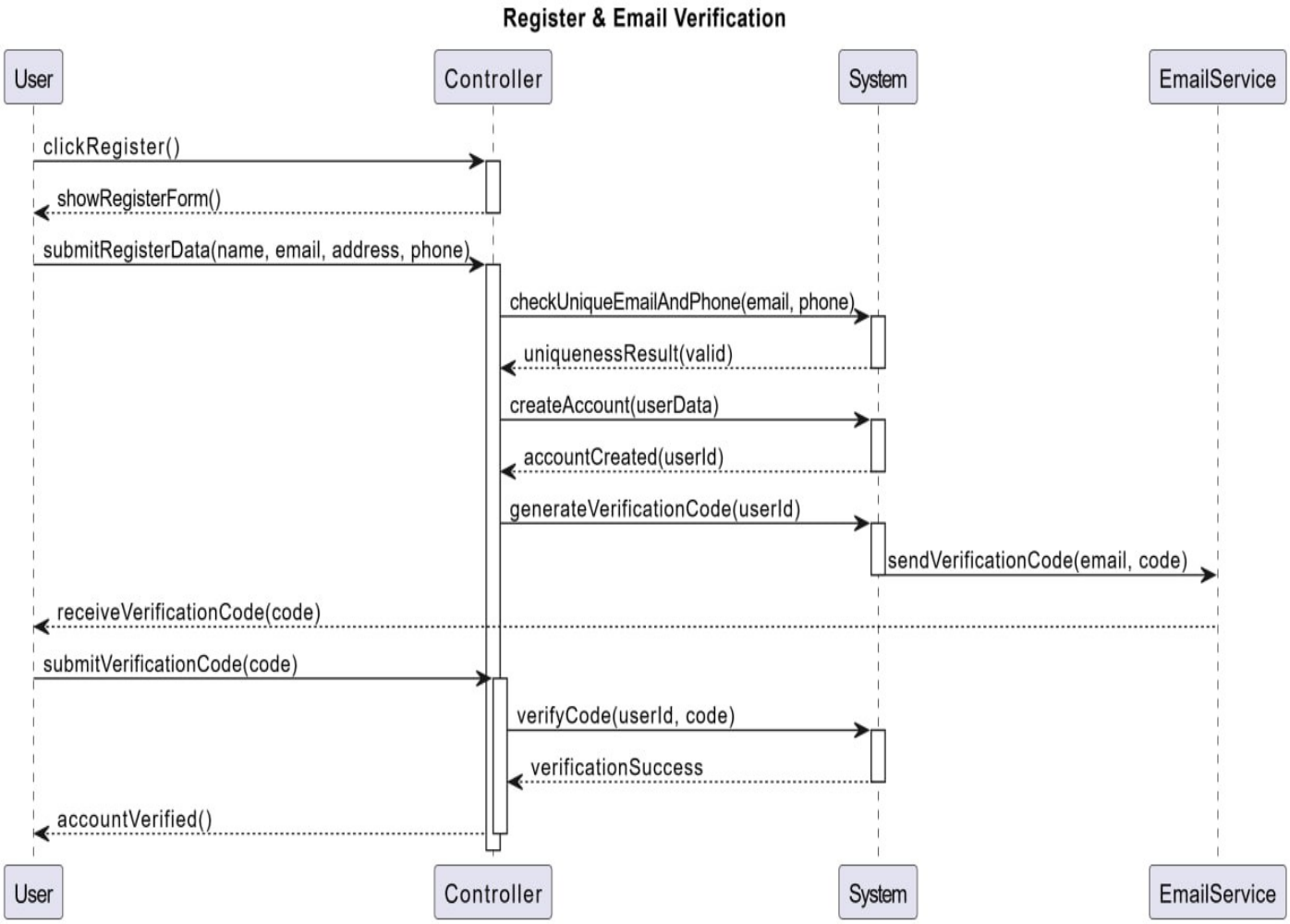

**Sequence diagram(log out) – Figure 7**

### 5.1.3. Sign up

| | |
|---|---|
| **Use Case ID** | UC-5 |
| **Use Case** | Sign up |
| **Actor** | User |
| **Pre-Condition** | - |
| **Main Scenario** | 1. User clicks on “Register”<br>2. System presents form with (name, email, address, phone number)<br>3. User enter data then clicks on “Register”<br>4. System checks if the email and phone number are unique<br>5. System creates new account in the system<br>6. System creates verification code and send it via email<br>7. User receive verification code via his email<br>8. User enter verification code<br>9. System check if verification code is correct<br>10. System marks account as verified |
| **Alternative Scenario** | In step 4 if the email or phone number are not unique, system return error message and redirect user to enter data again<br>In step 9 if the verification code incorrect the system return error message and redirect user to enter code again |



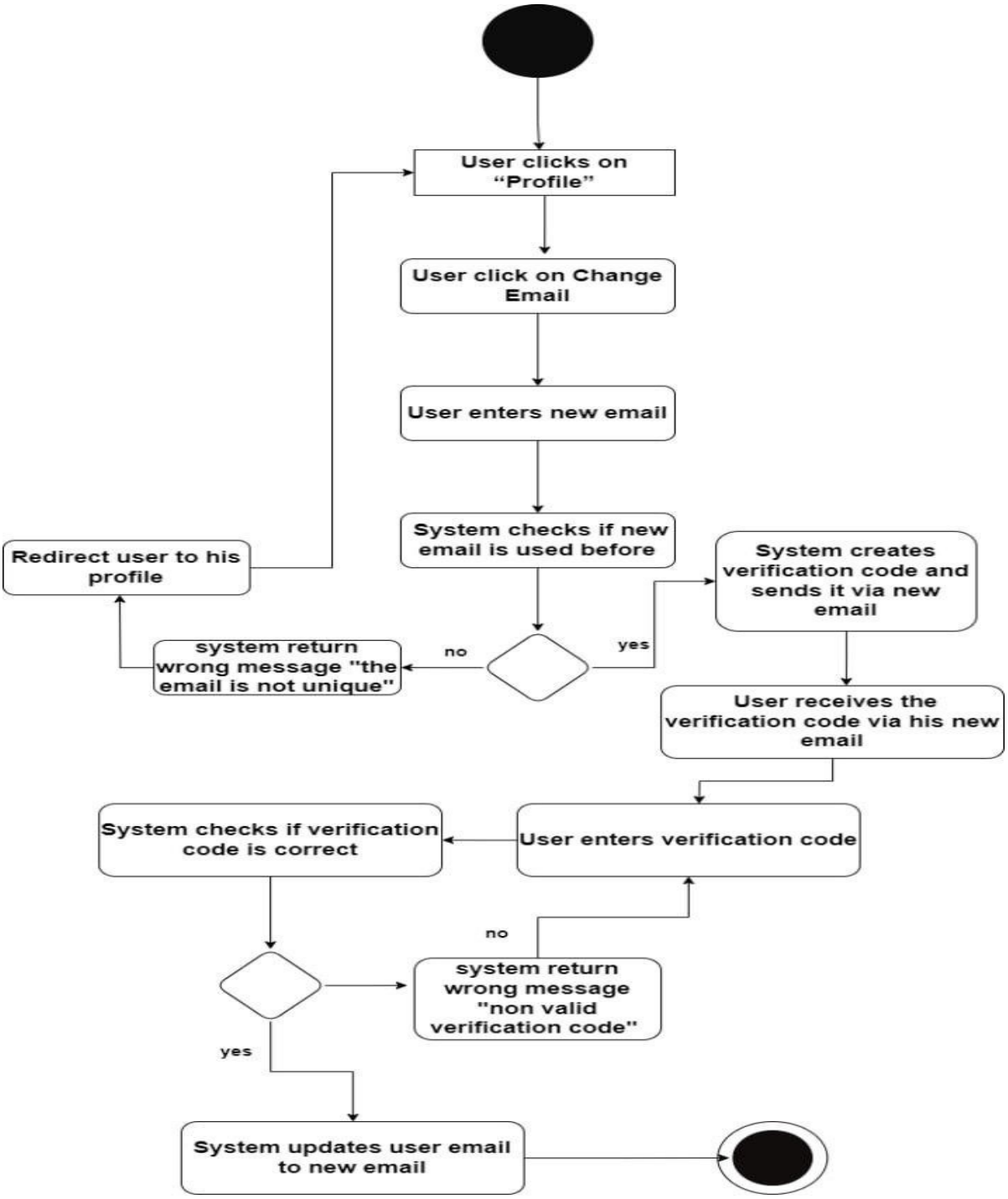

**Activity diagram(Sign up) – Figure 8**



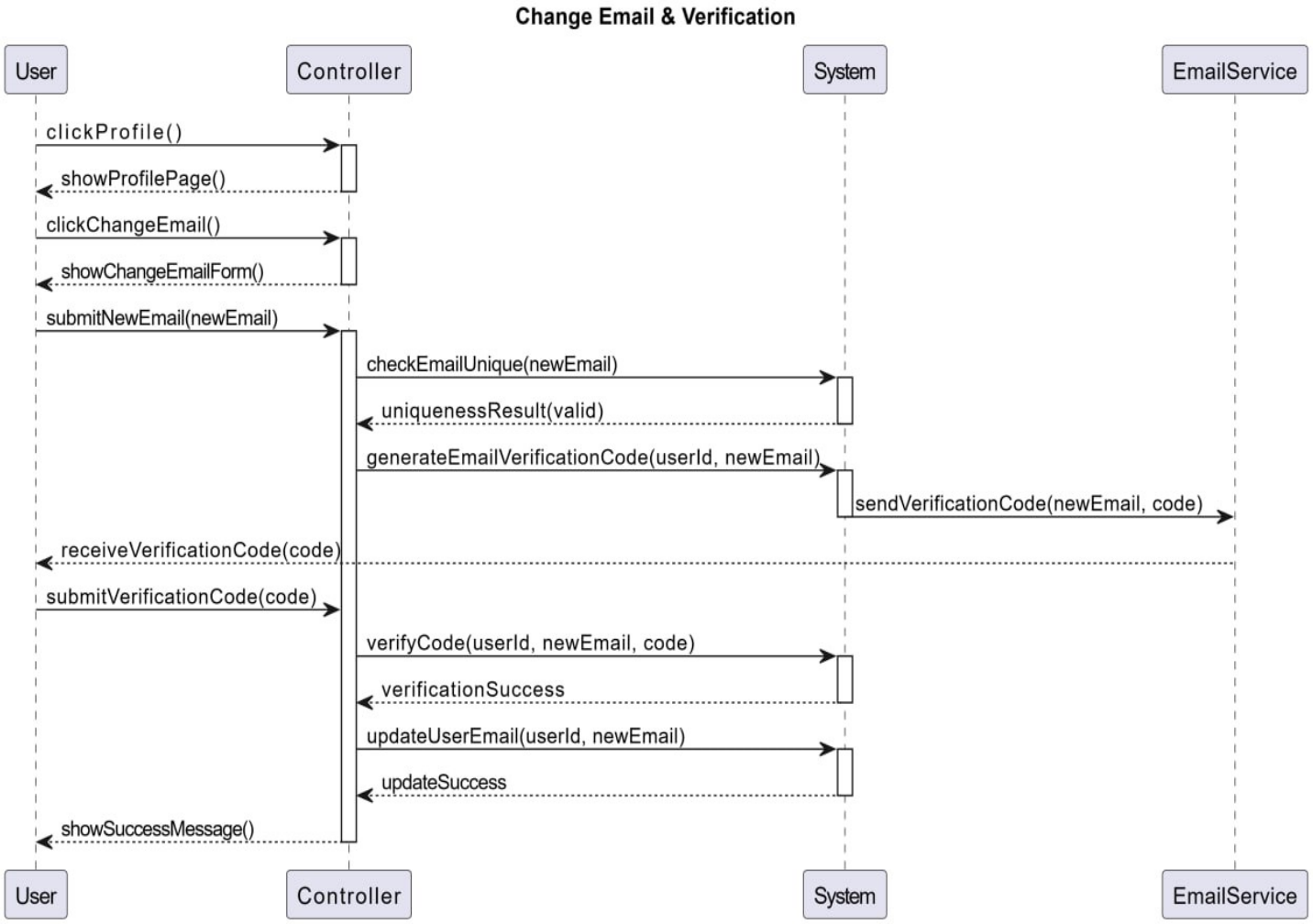

**Sequence diagram(Sign up) – Figure 9**

### 5.1.4 Update email

| | |
|---|---|
| **Use Case ID** | UC-6 |
| **Use Case** | Update email |
| **Actor** | User |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Profile”<br>2. User clicks on “Change email”<br>3. User enter new email<br>4. System checks if new email is used before<br>5. System creates verification code and send it via new email<br>6. User receive the verification code via his new email<br>7. User enter verification code<br>8. System checks if verification code is correct<br>9. System update user email to new email |
| **Alternative Scenario** | In step 4 if the entered email is not unique, system return error message and redirect user to his profile<br>In step 8 if the verification code is incorrect, system return error message and redirect user to enter code again |



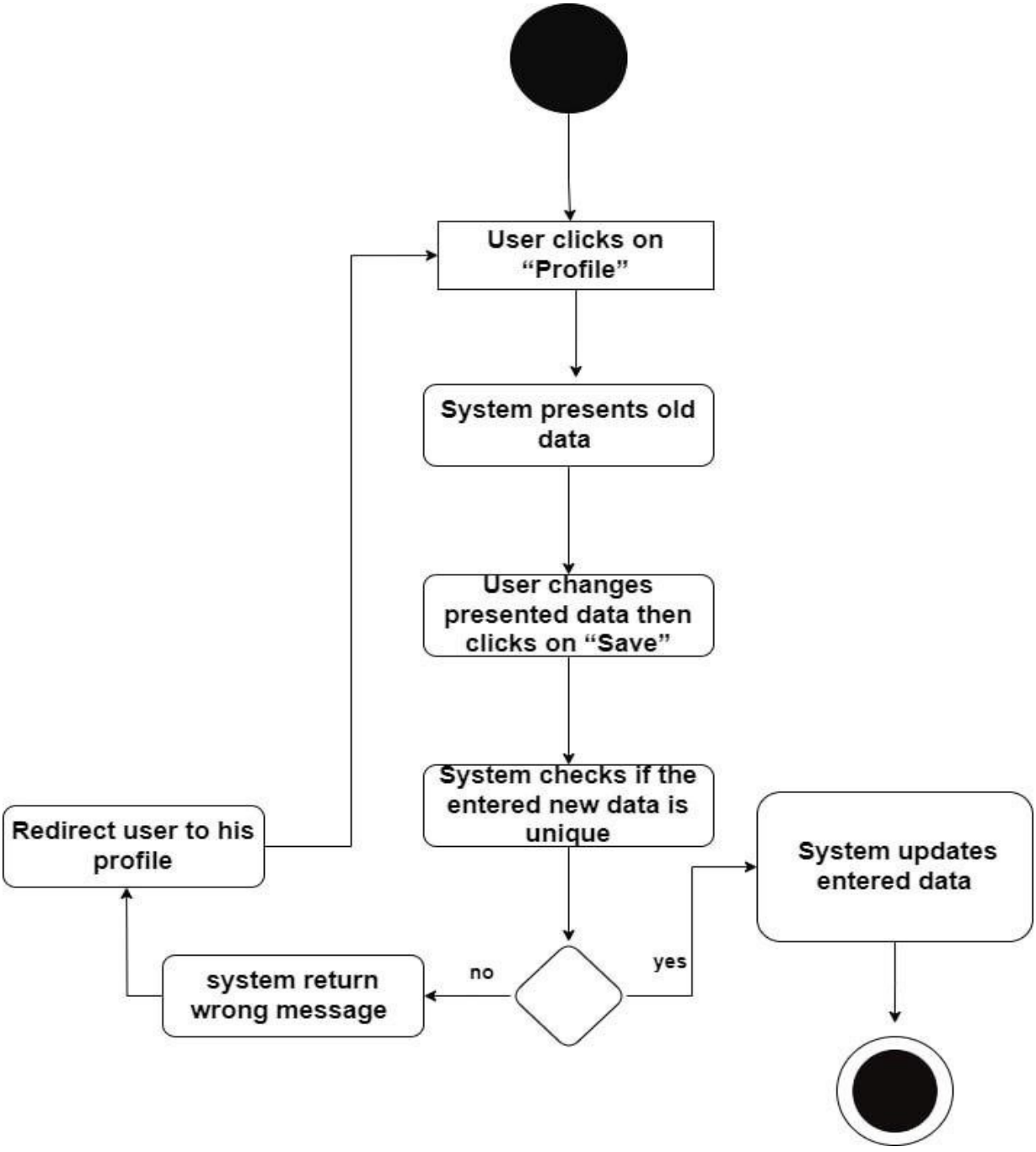

**Activity diagram(Update email) – Figure 10**



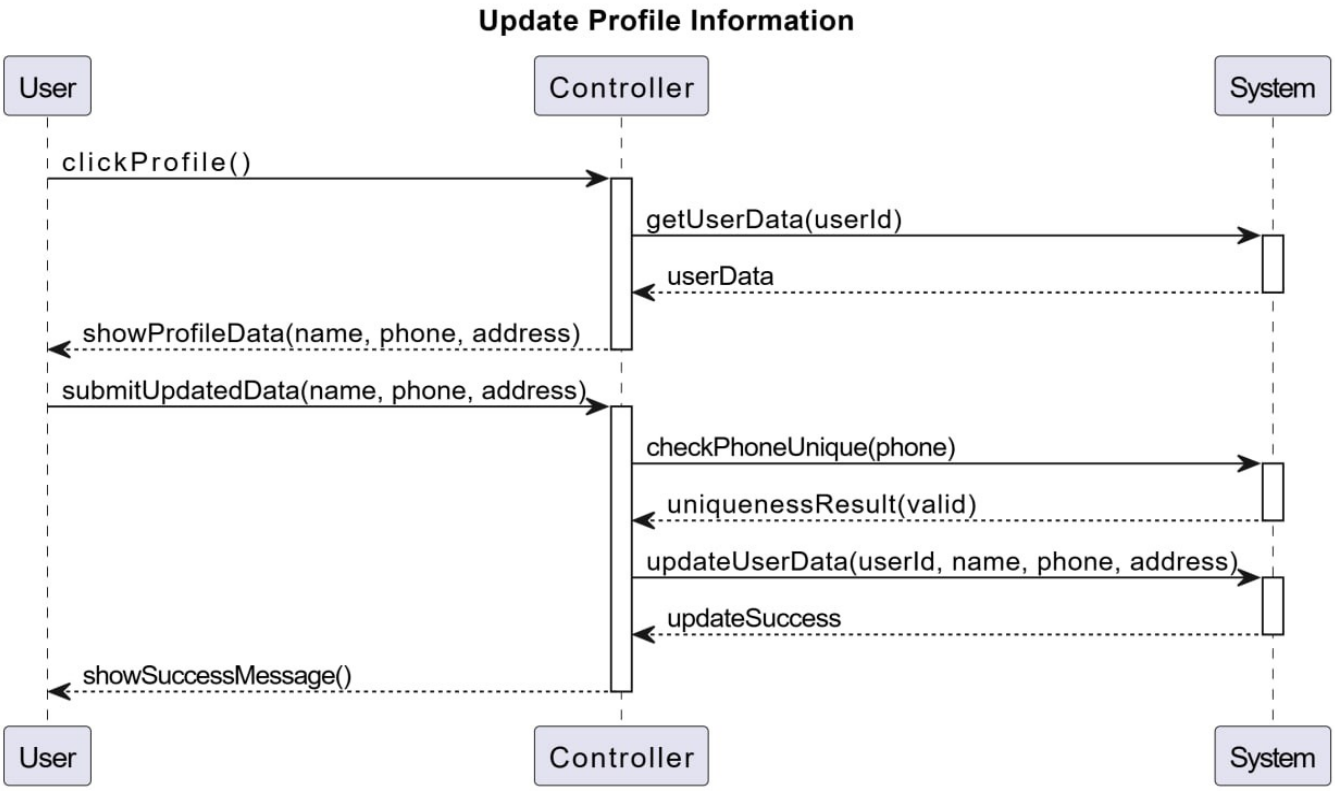

**Sequence diagram(Update email) – Figure 11**

### 5.1.5 Update profile data

| | |
|---|---|
| **Use Case ID** | UC-7 |
| **Use Case** | Update profile data |
| **Actor** | User |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Profile”<br>2. System present old data (name, phone number, address)<br>3. User change presented data then clicks on “Save”<br>4. System checks if the entered phone number is unique<br>5. System update entered data |
| **Alternative Scenario** | In step 4 if the entered phone number is not unique, system return error message and redirect user to his profile |



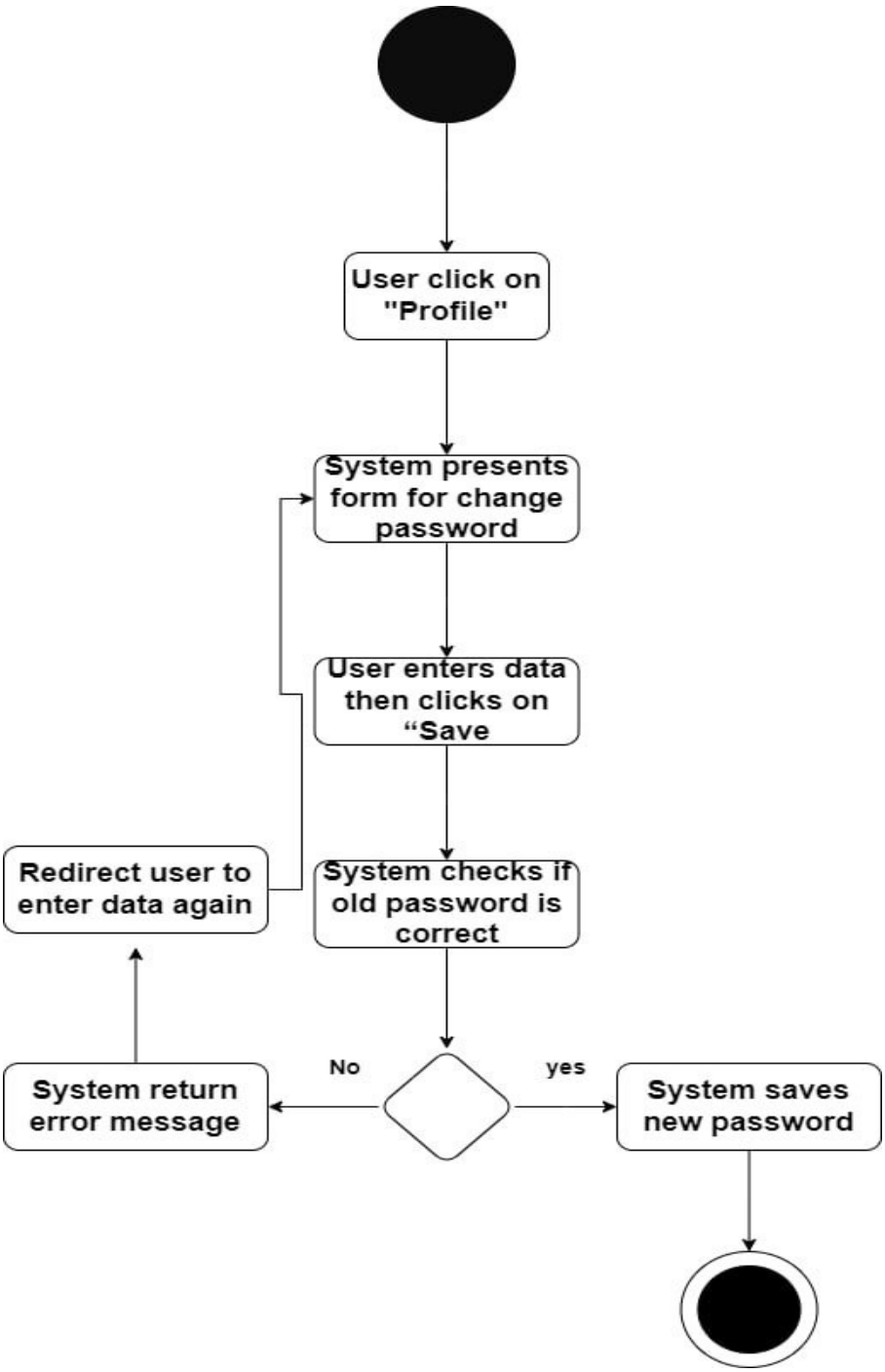

**Activity diagram(Update profile data) – Figure 12**



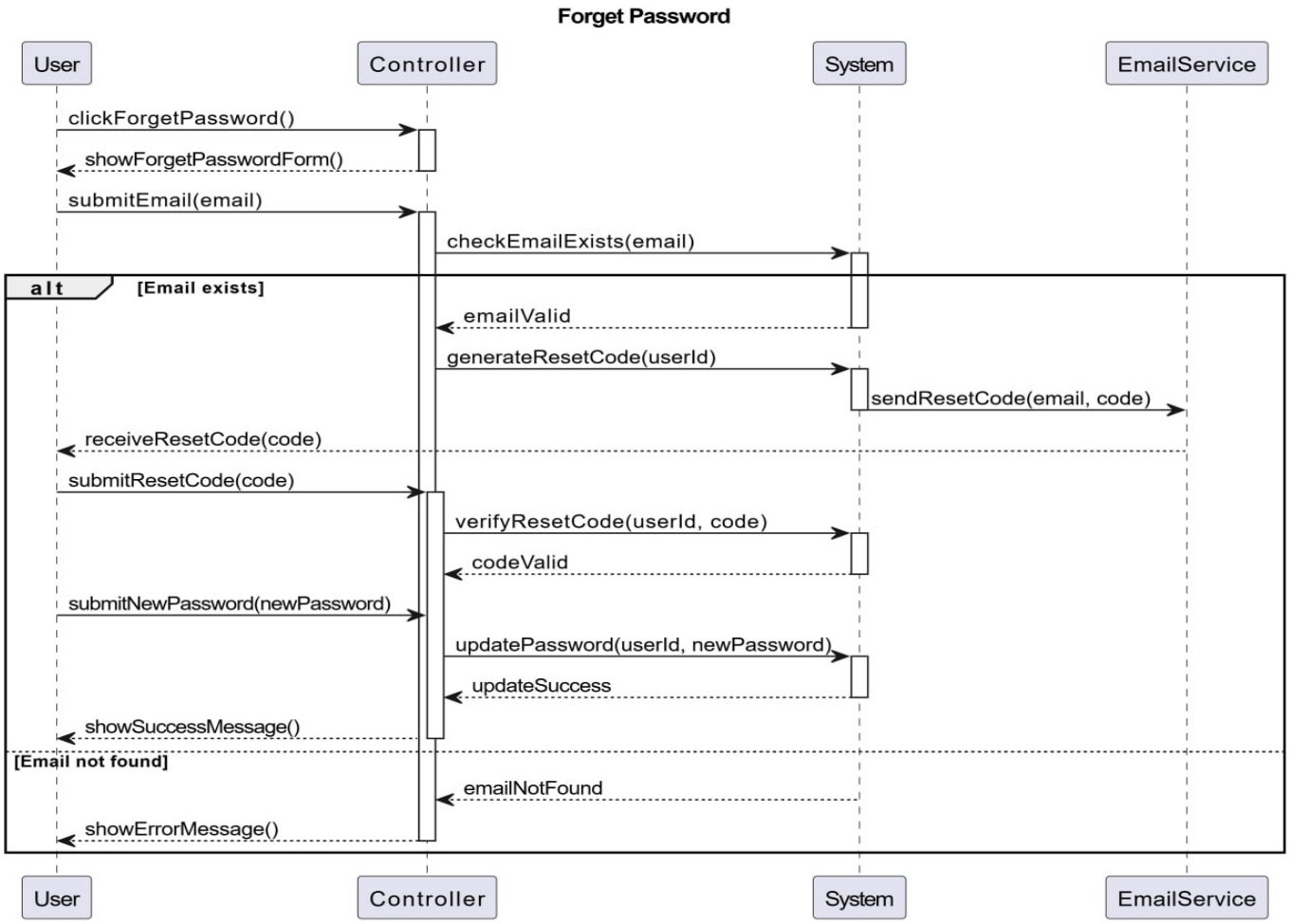

**Sequence diagram(Update profile data) – Figure 13**

### 5.1.6 Forget password

| | |
|---|---|
| **Use Case ID** | UC-8 |
| **Use Case** | Forget password |
| **Actor** | User |
| **Pre-Condition** | Has an account in the system, User in sign in page |
| **Main Scenario** | 1. User clicks on “Forget Password”<br>2. User enter his email<br>3. System checks if the email exists in the system<br>4. System creates verification code and send it via entered email<br>5. User receive forget password code<br>6. User enter verification code<br>7. System checks if the verification code is correct<br>8. User enter new password<br>9. System update user password |
| **Alternative Scenario** | In step 3 if the email dose not exists the system return error message and redirect user to enter email again<br><br>In step 7 if the verification code is incorrect the system return error message and redirect user to enter password again |



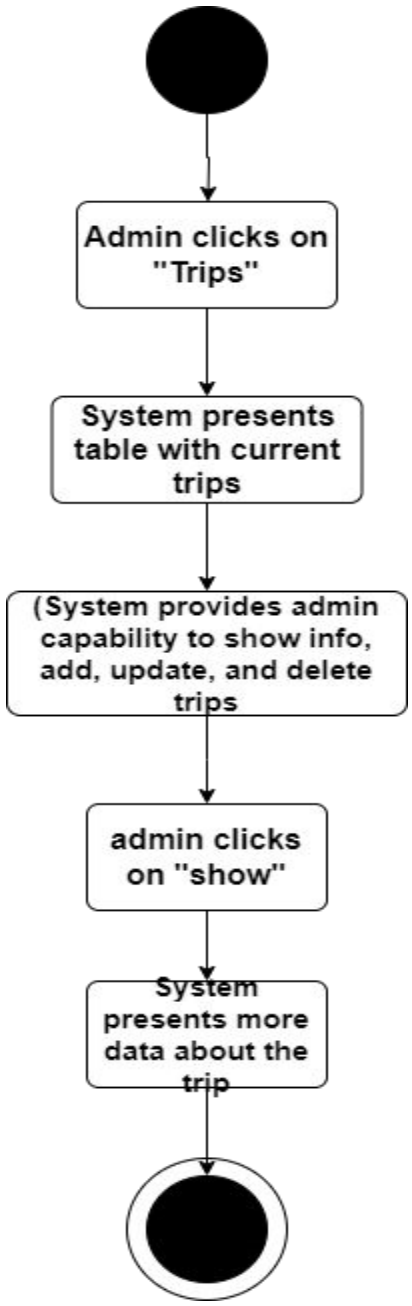

**Activity diagram(Forget password) – Figure 14**



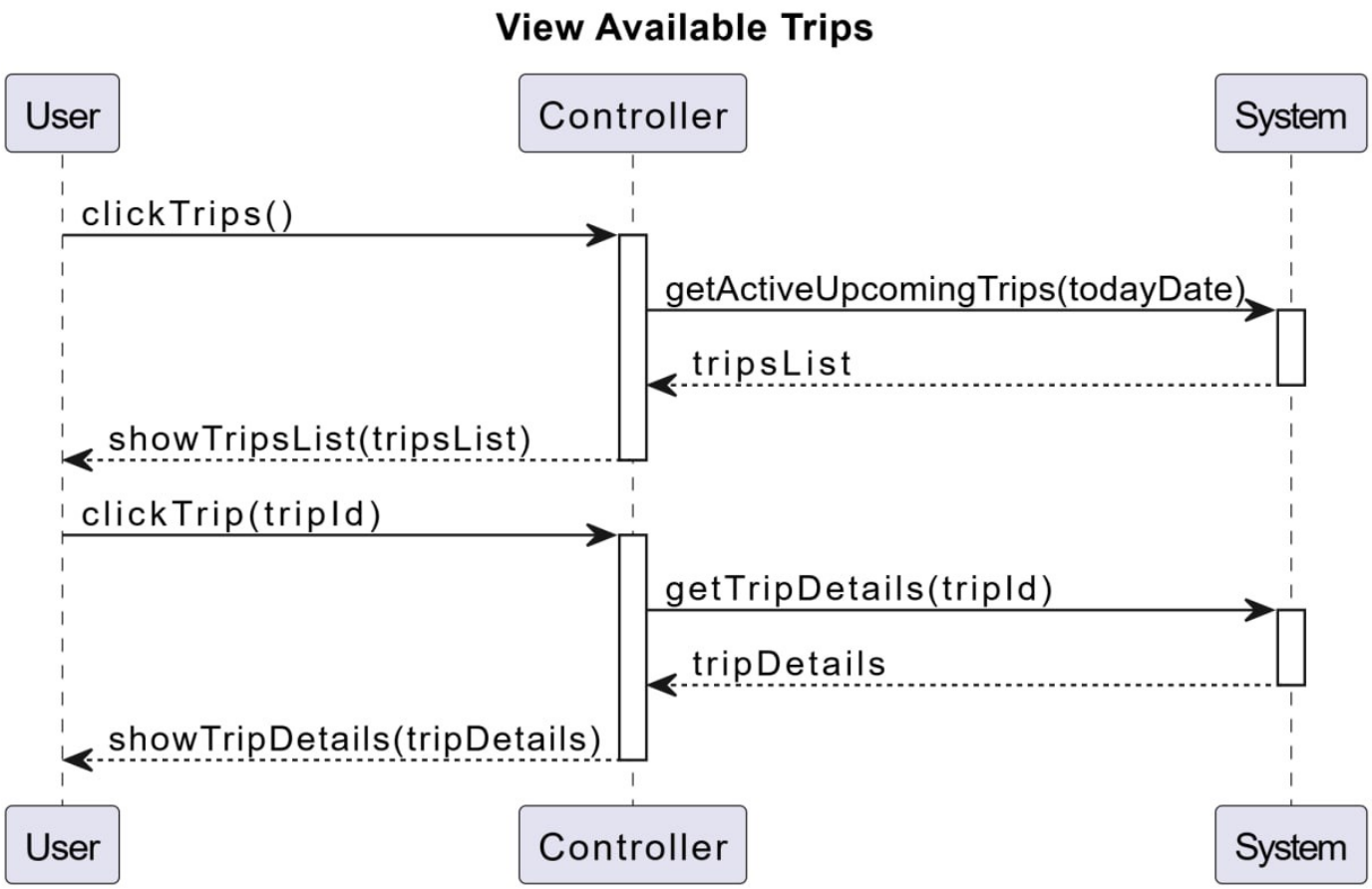

**Sequence diagram(Forget password) – Figure 15**

### 5.1.7 View trip

| | |
|---|---|
| **Use Case ID** | UC-9 |
| **Use Case** | View Trips |
| **Actor** | User |
| **Pre-Condition** | User in home page |
| **Main Scenario** | 1. System presents list of active and the start date is after today trips (name, price, photo, start date, end date, list of nights (e.g. 3 nights in Makkah) and description) and each trip is clickable to show more details |
| **Alternative Scenario** | - |



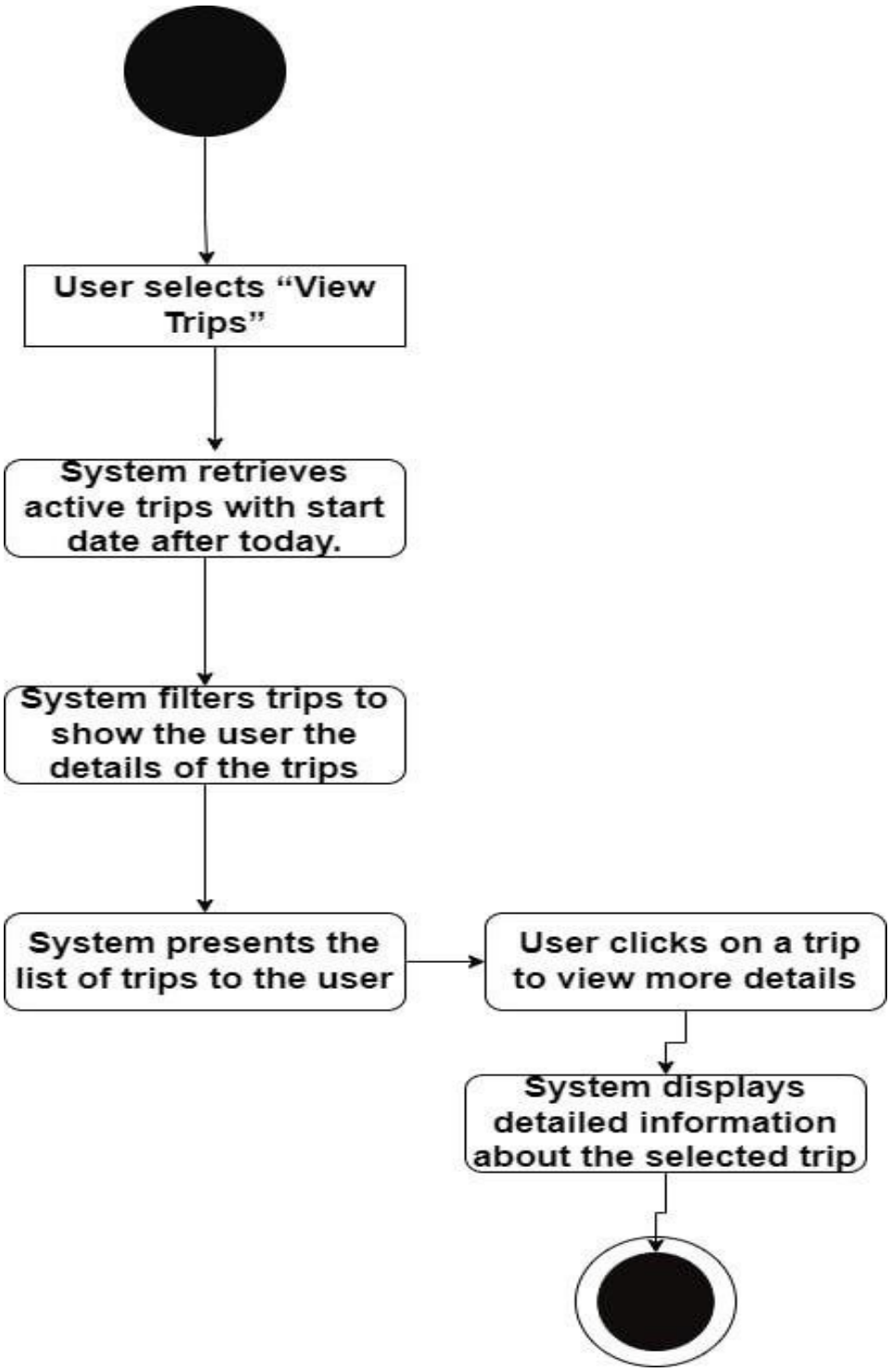

**Activity diagram(View Trips) – Figure 16**



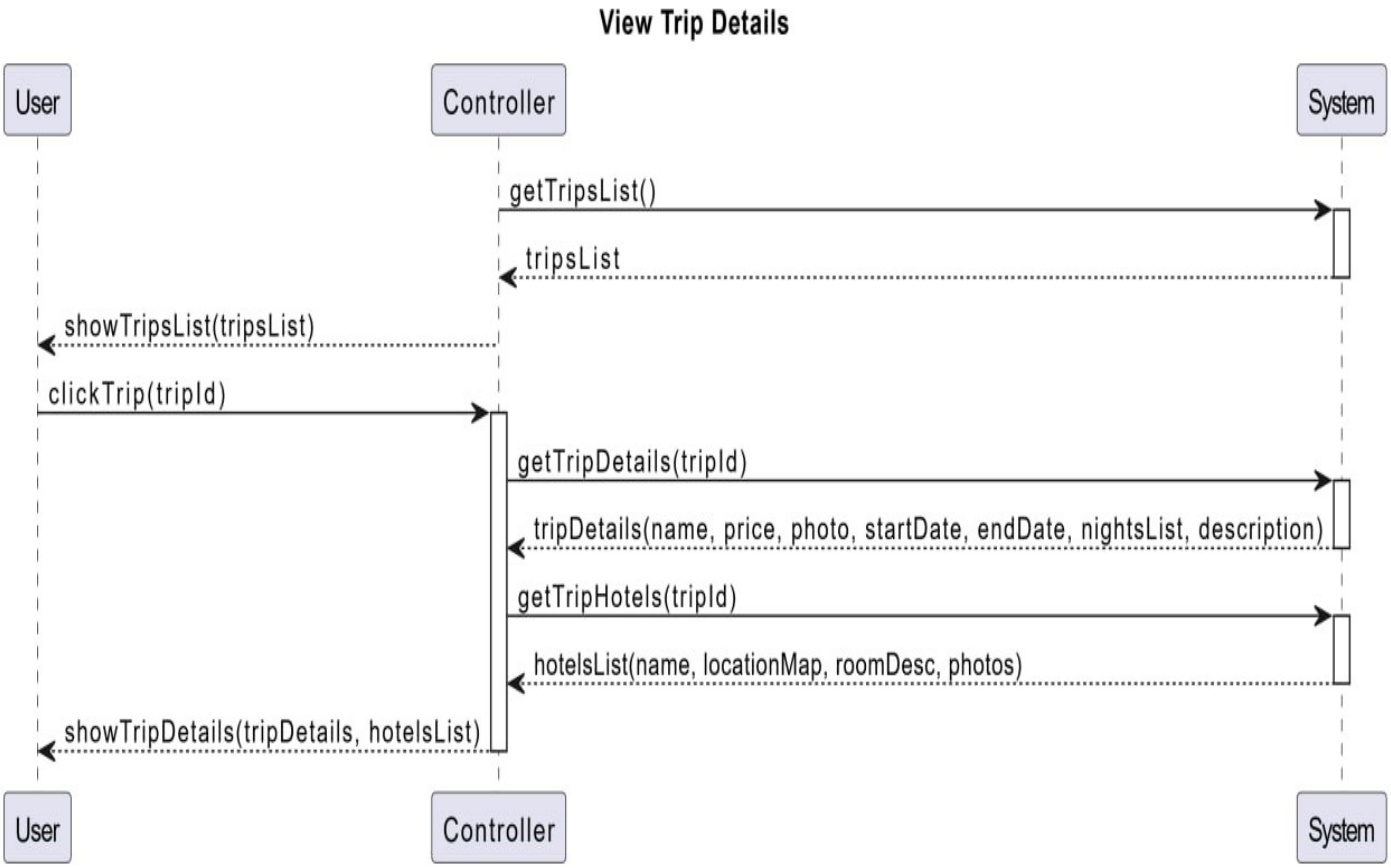

**Sequence diagram(view trips) – Figure 17**

### 5.1.8 View Trip info

| | |
|---|---|
| **Use Case ID** | UC-10 |
| **Use Case** | View Trip Info |
| **Actor** | User |
| **Pre-Condition** | User in home page |
| **Main Scenario** | 1. System presents list of trips each trip is clickable to show more details<br>2. User clicks on any trip<br>3. System presents more info about trip (name, price, photo, start date, end date, list of nights (e.g. 3 nights in Makkah), description, list of hotels info (name, location (as map), room description, photos)) |
| **Alternative Scenario** | - |



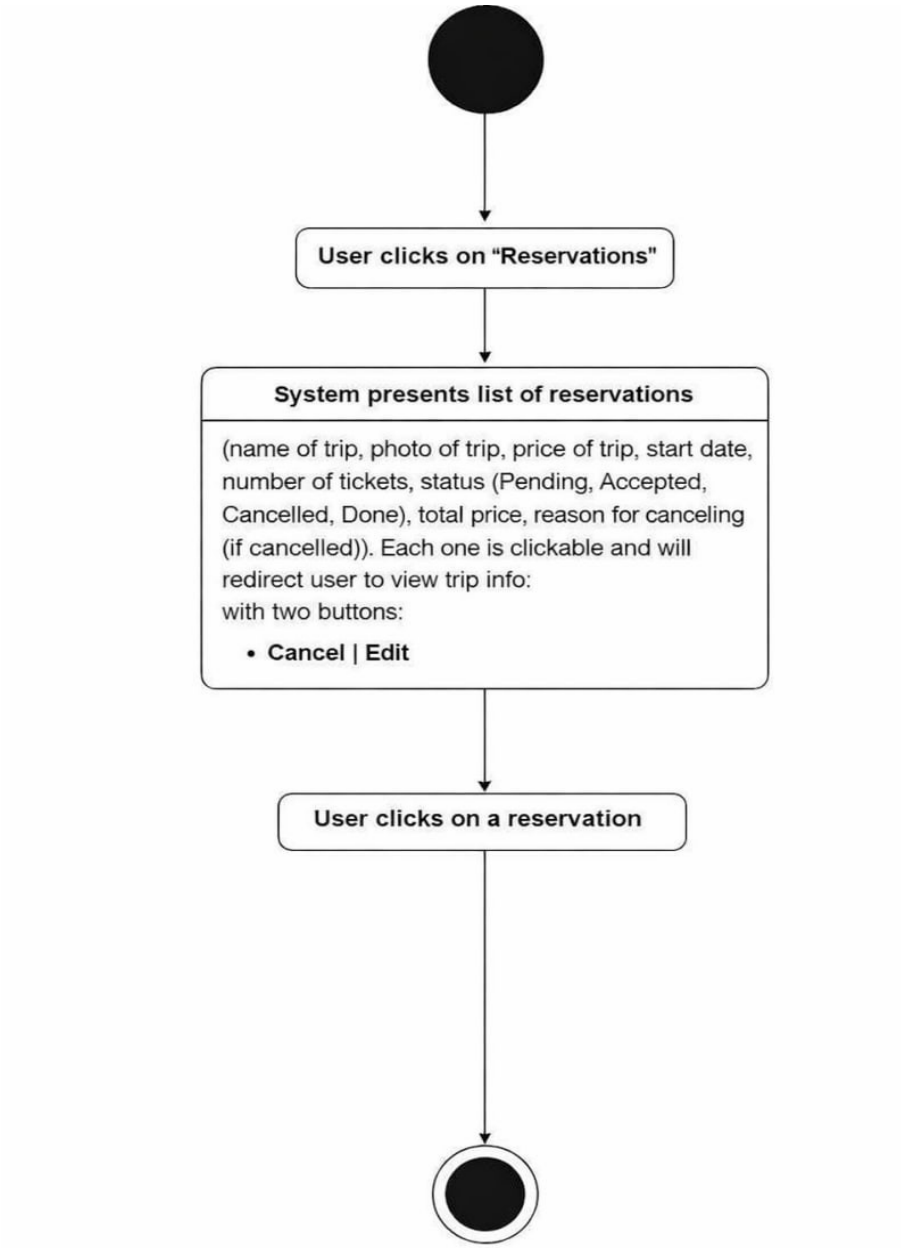

**Activity diagram(View Trips info) – Figure 18**



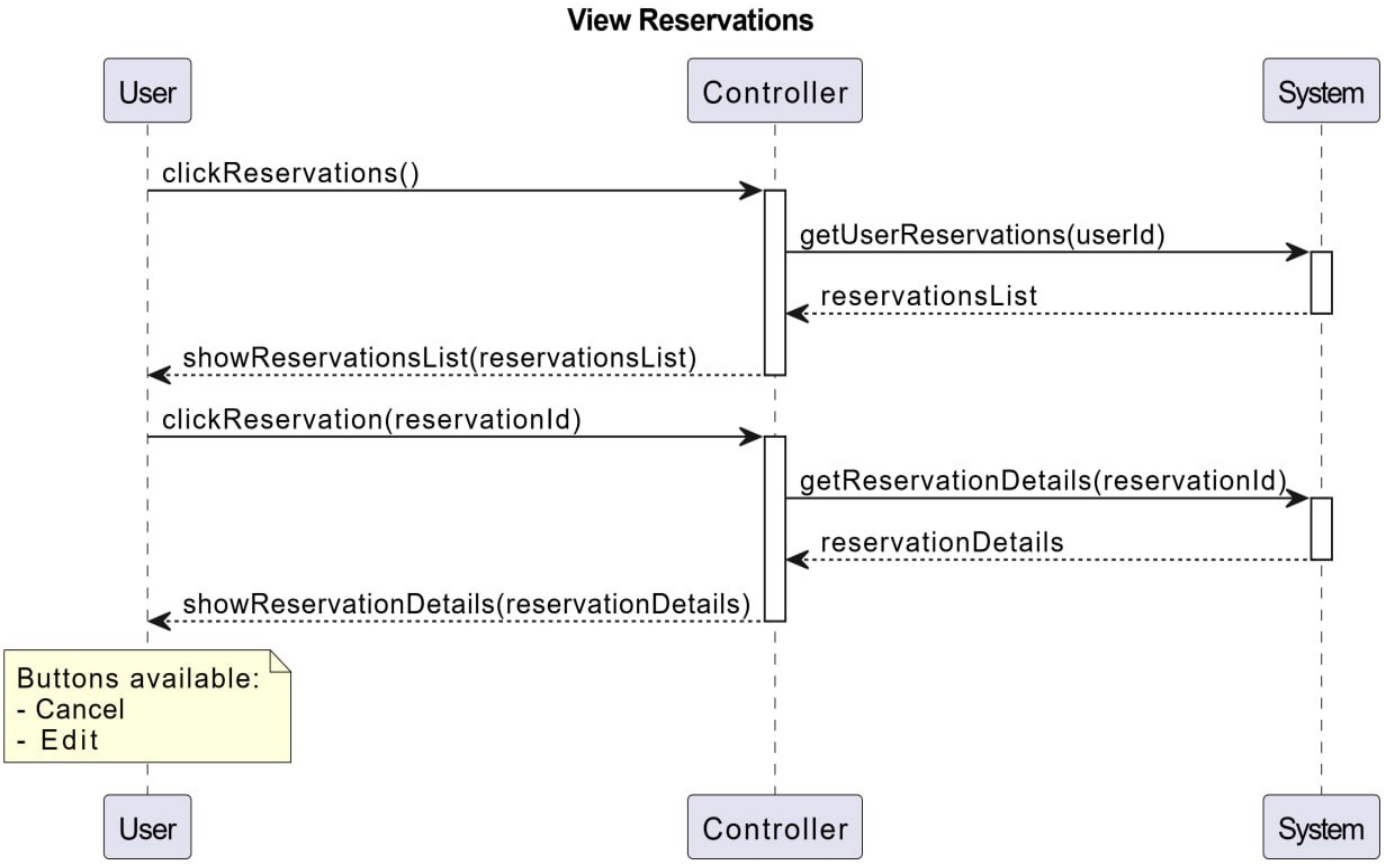

**Sequence diagram(view trips info) – Figure 19**

### 5.1.9 **View Reservations**

| | |
|---|---|
| **Use Case ID** | UC-11 |
| **Use Case** | View Reservations |
| **Actor** | User |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Reservations”<br>2. System present list of reservations (name of trip, photo of trip, price of trip, start date, number of tickets, status (Pending, Accepted, Cancelled, Done), total price, reason of canceling If it is canceled) and each one is clickable and will redirect user to view trip info and tow buttons (Cancel, Edit) |
| **Alternative Scenario** | - |



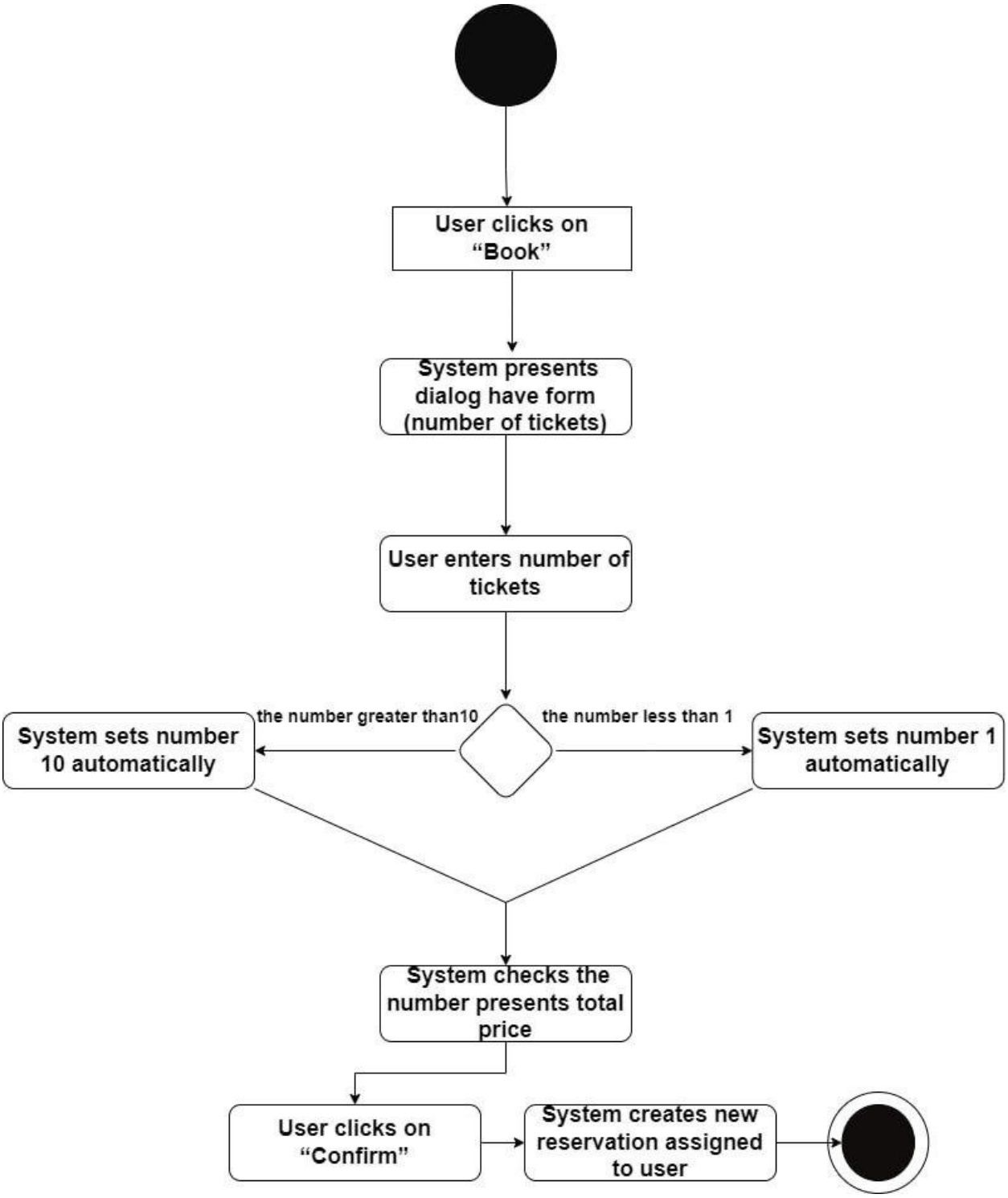

**Activity diagram(View Reservation) – Figure 20**



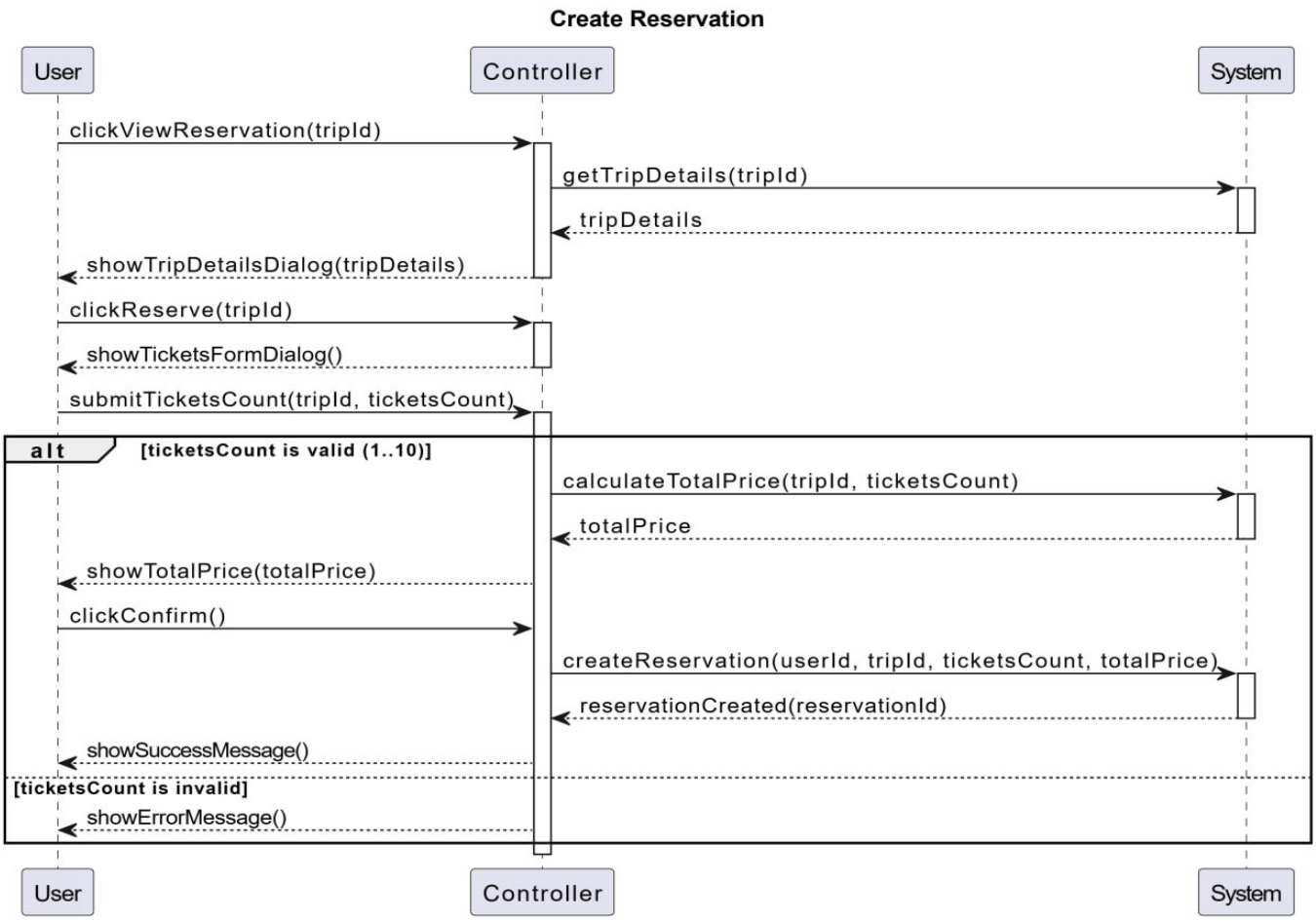

**Sequence diagram(View Reservation) – Figure 21**

### 5.1.10 Book trip

| | |
|---|---|
| **Use Case ID** | UC-12 |
| **Use Case** | Book Trip |
| **Actor** | User |
| **Pre-Condition** | Already logged in, and in view trip info page |
| **Main Scenario** | 1. User clicks on “view reservation”<br>2. System presents dialog have details of trip<br>3. user click on reservation<br>4. System presents dialog have form to select number of ticket<br>5. User enter number greater than 0 and less or equals 10<br>6. System checks the number presents total price<br>7. User clicks on “Confirm”<br>8. system create new reservation assigned to user |
| **Alternative Scenario** | in step 4 if the number is less than 1 make it 1 automatically and if the number greater than 10 make 10 automatically |



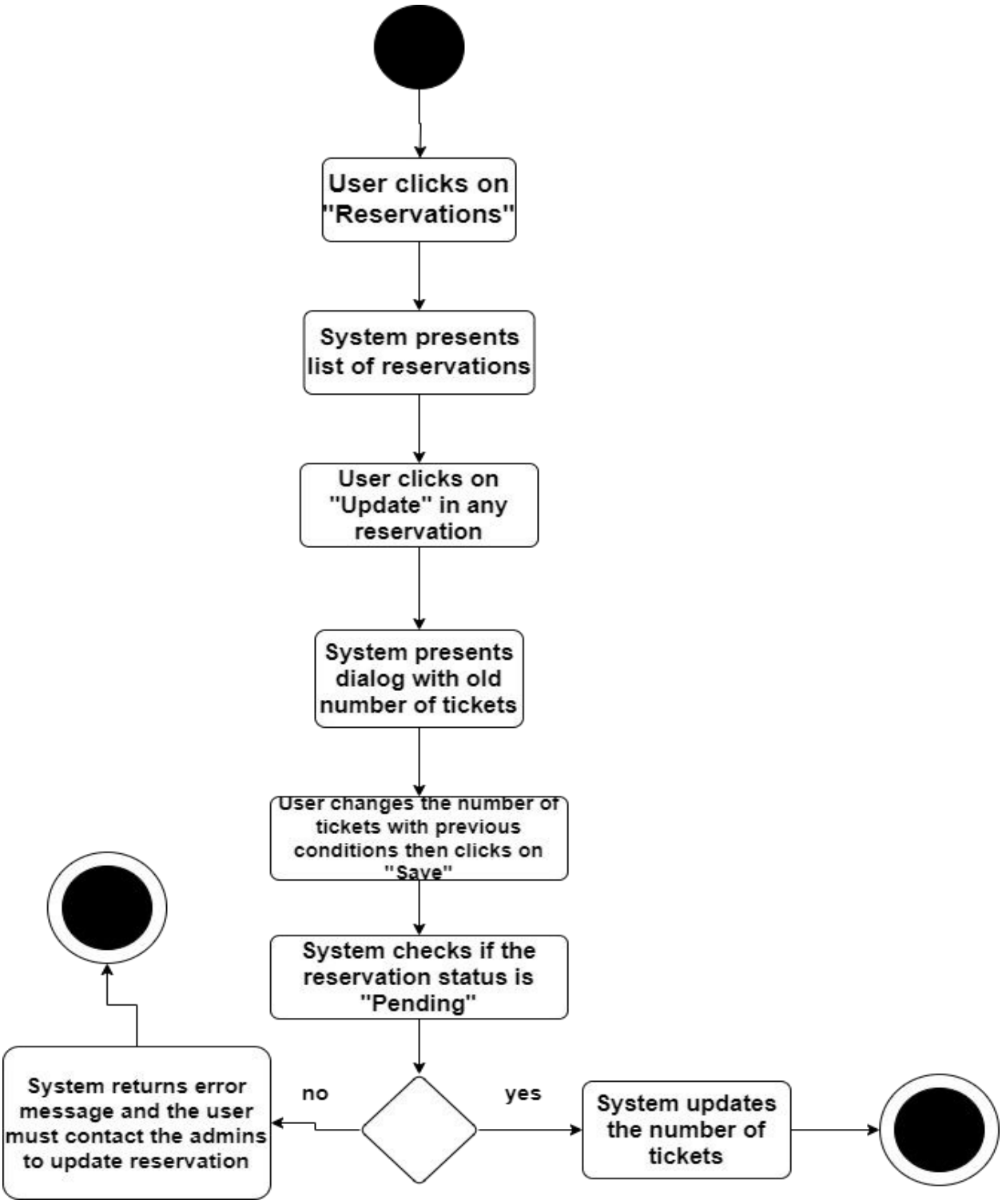

**Activity diagram(Book Trip) – Figure 22**



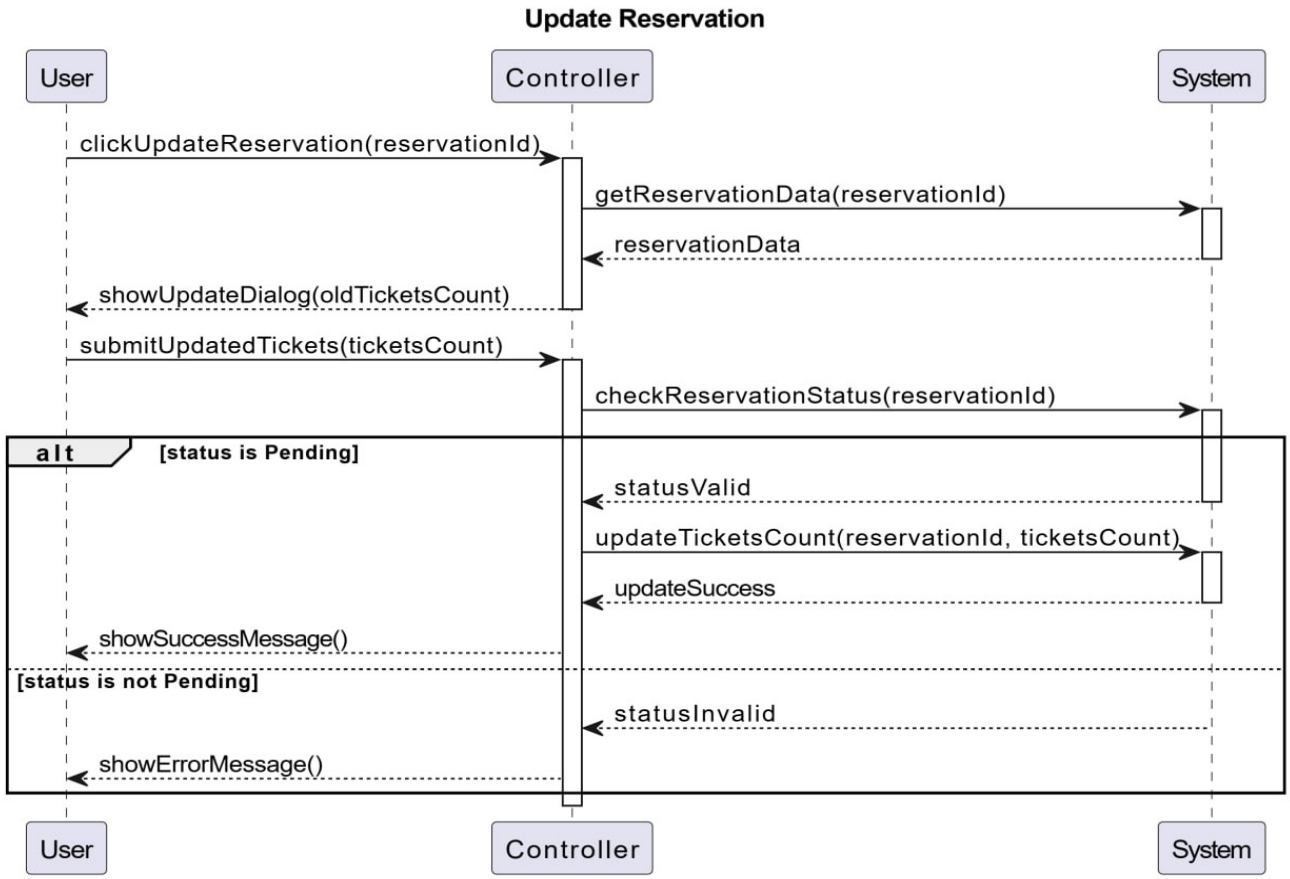

**Sequence diagram(Book trip) – Figure 23**

### 5.1.11 Update Reservation

| | |
|---|---|
| **Use Case ID** | UC-13 |
| **Use Case** | Update Reservation |
| **Actor** | User |
| **Pre-Condition** | Already logged in, and in view reservations page |
| **Main Scenario** | 1. User clicks on “Update” in any reservation<br>2. System present dialog with old number of tickets<br>3. User change the number of tickets with previous conditions then clicks on “Save”<br>4. System checks the reservation status is “Pending”<br>5. System updates number of tickets |
| **Alternative Scenario** | In step 4 if the status is not “Pending” the system return error message and the user must contact the admins to update reservation |



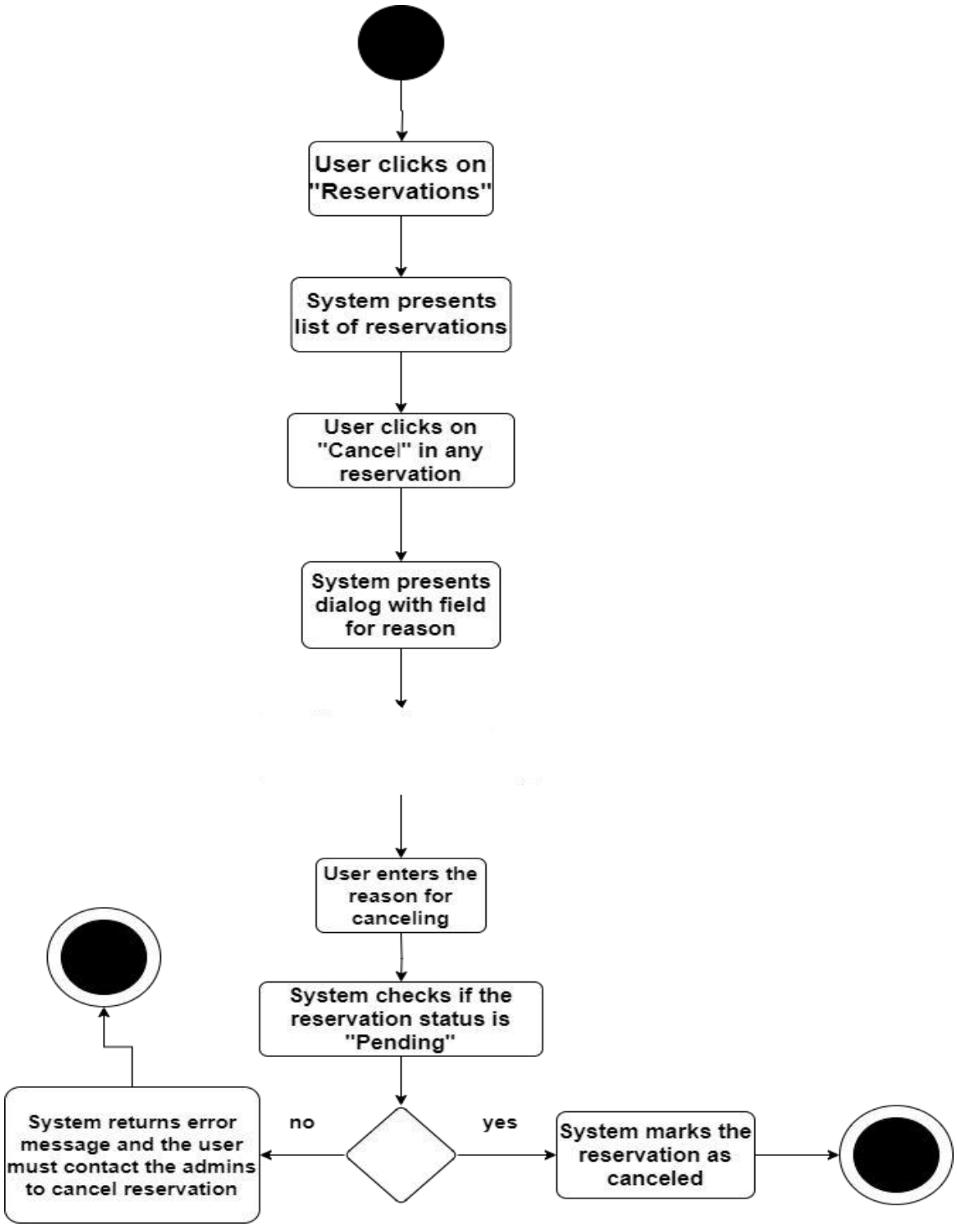

**Activity diagram(**Update Reservation**) – Figure 24**



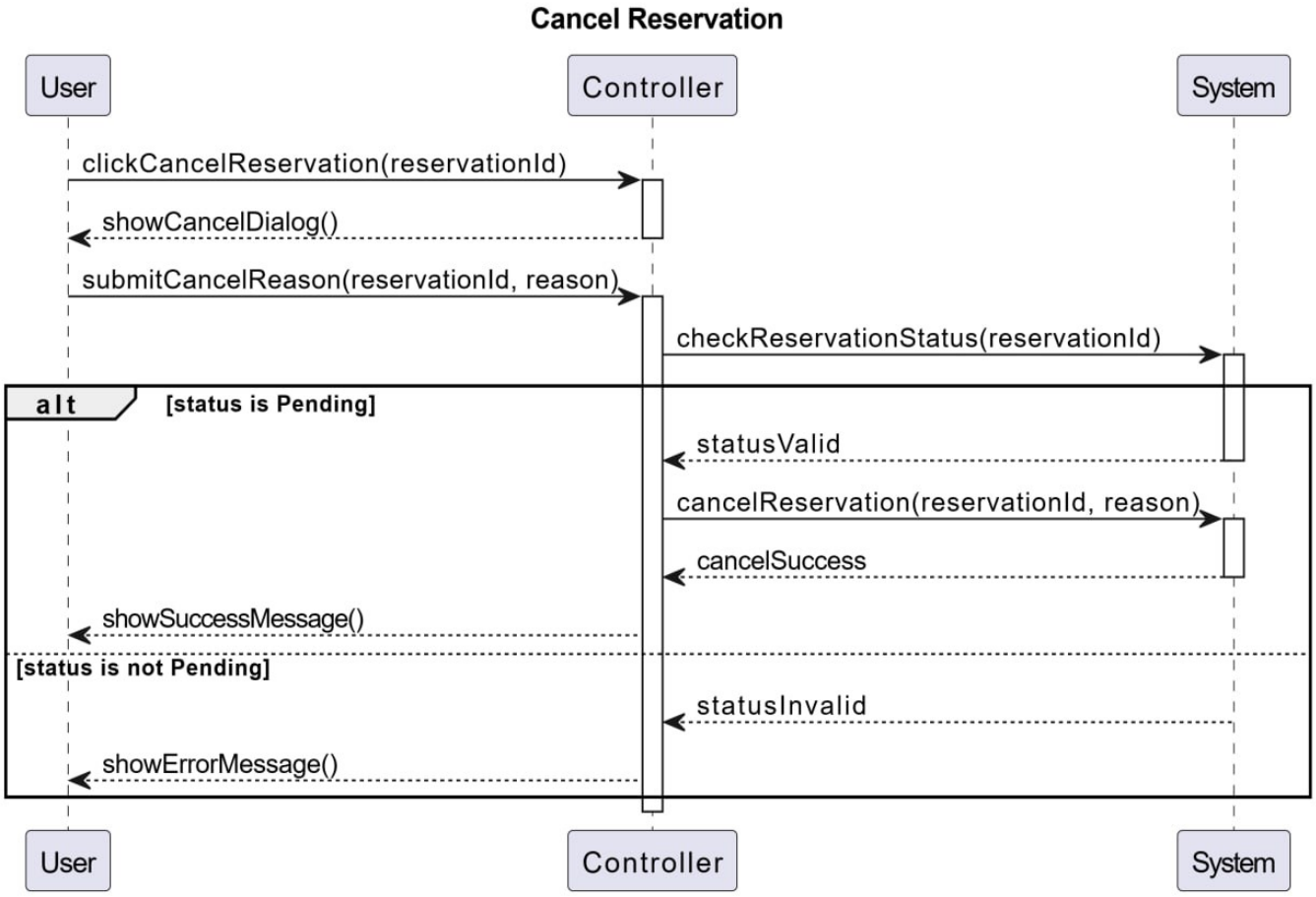

**Sequence diagram(**Update Reservation**) – Figure 25**

### 5.1.12 Cancel Reservation

| | |
|---|---|
| **Use Case ID** | UC-14 |
| **Use Case** | Cancel Reservation |
| **Actor** | User |
| **Pre-Condition** | Already logged in, and in view reservations page |
| **Main Scenario** | 1. User clicks on “Cancel” in any reservation<br>2. System present dialog with field reason<br>3. User enter the reason of canceling<br>4. System checks the reservation status is “Pending”<br>5. System marks the reservation as canceled |
| **Alternative Scenario** | In step 4 if the status is not “Pending” the system return error message and the user must contact the admins to cancel reservation |



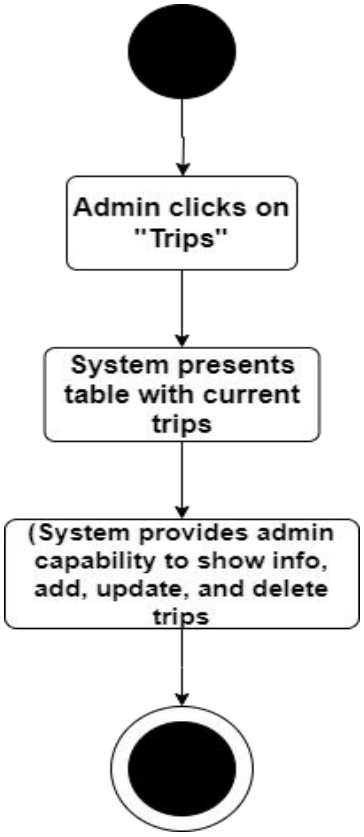

**Activity diagram(Cancel Reservation) – Figure 26**



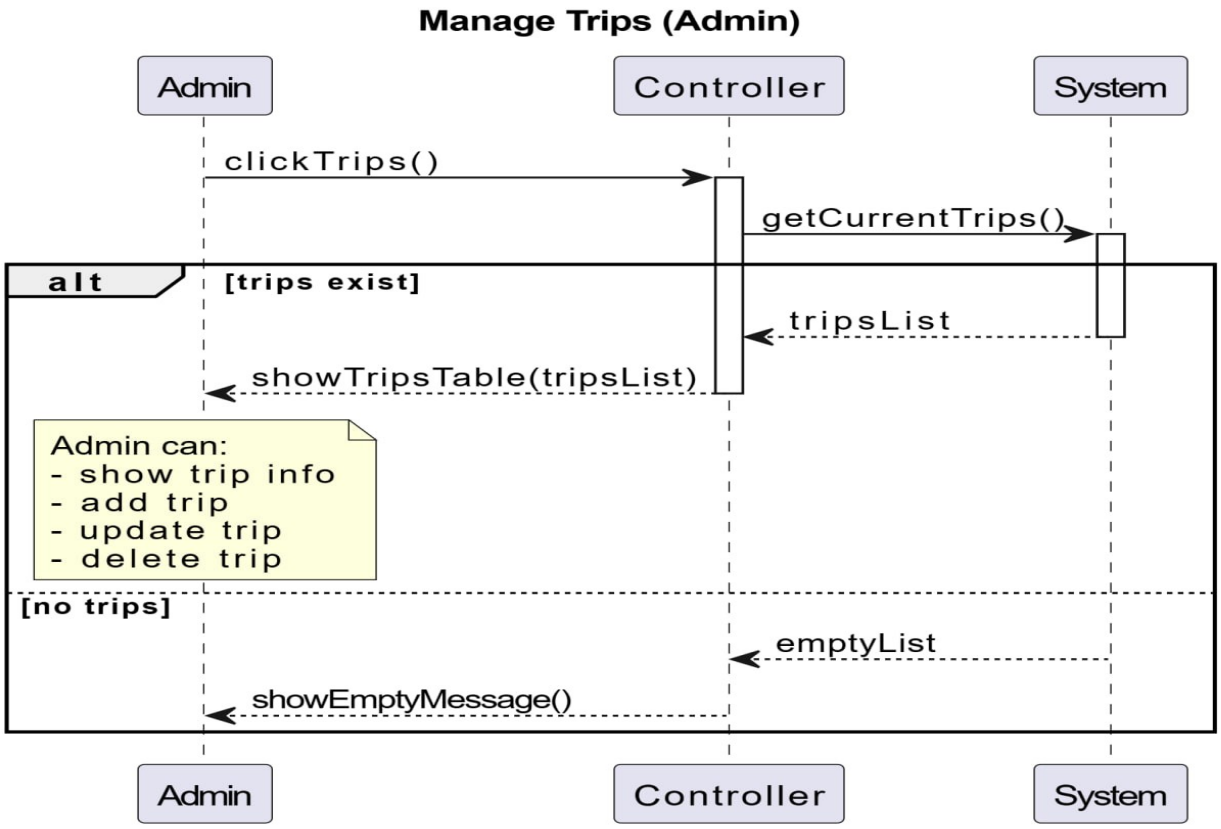

**Sequence diagram(Cancel Reservation) – Figure 27**

### 5.1.13 Manage trips

| | |
|---|---|
| **Use Case ID** | UC-15 |
| **Use Case** | Manage Trips |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Trips”<br>2. System Presents table with current tips (name, price, photo, start date, end date)<br>3. System provides admin capability to show info, add, update and delete trips |
| **Alternative Scenario** | |



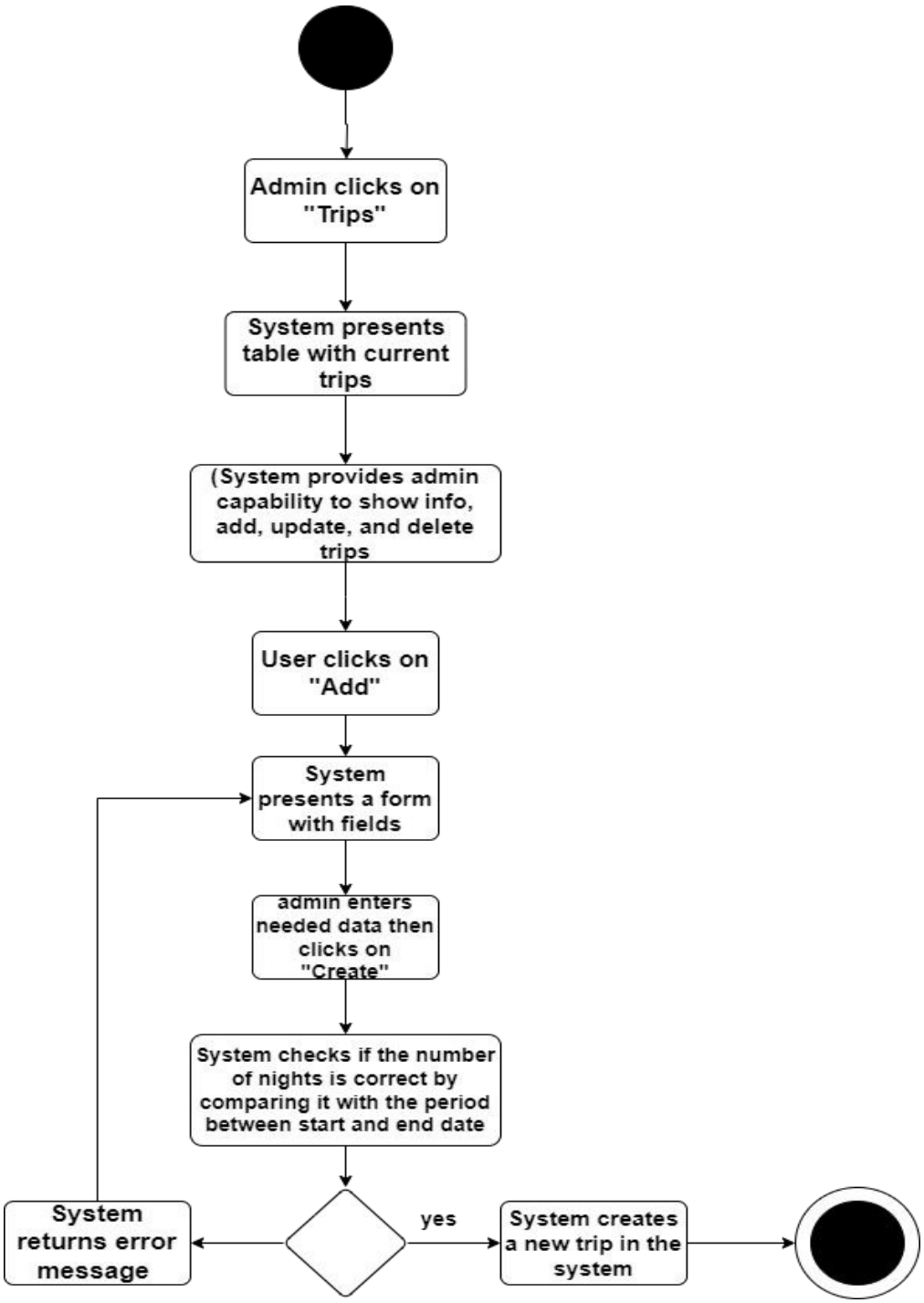

**Activity diagram(Manage trips) – Figure 28**



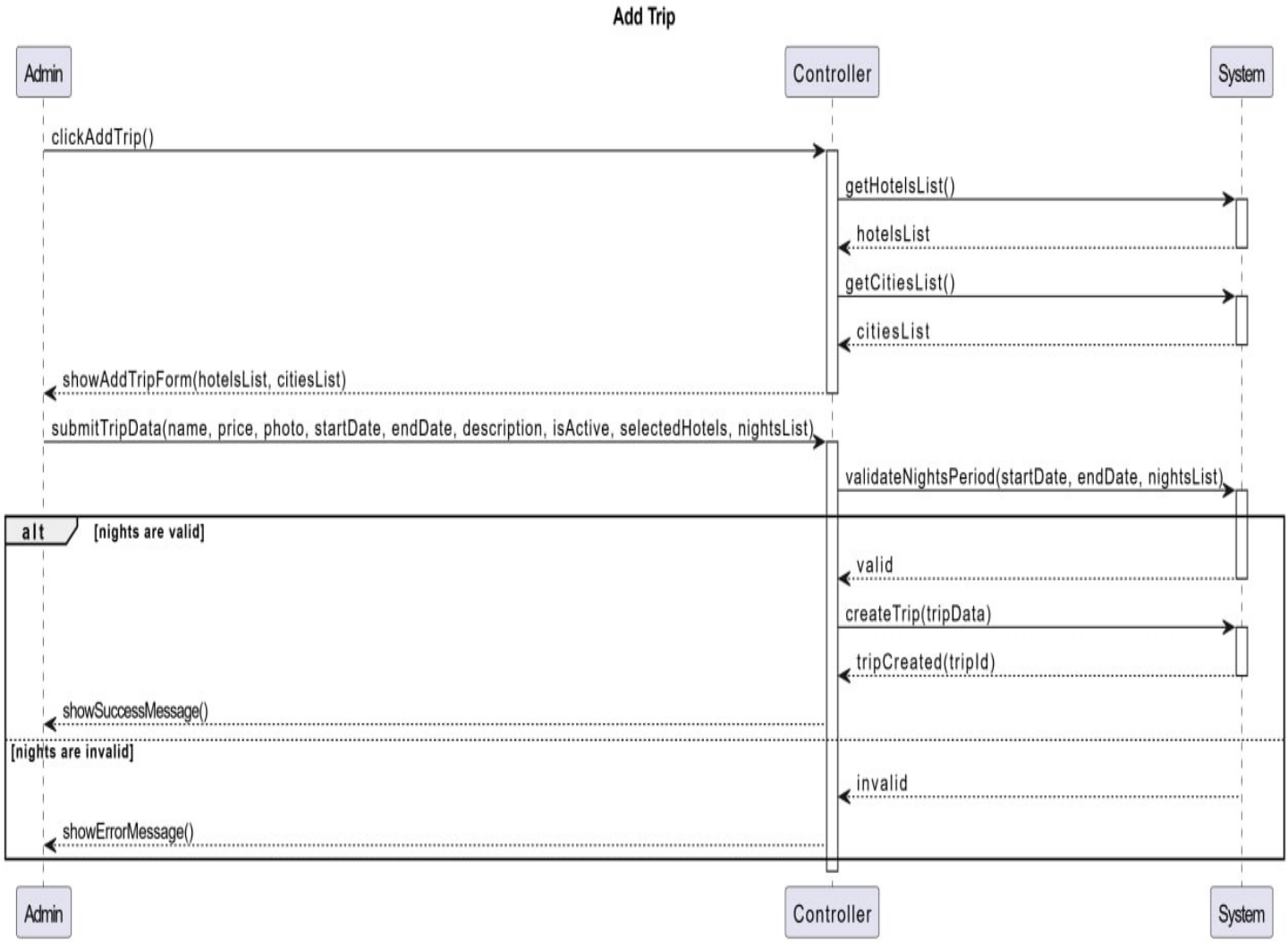

**Sequence diagram(Manage trips) – Figure 29**

### 5.1.13 Add Trip

| | |
|---|---|
| **Use Case ID** | UC-16 |
| **Use Case** | Add Trip |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, and in manage trips page |
| **Main Scenario** | 1. User clicks on “Add”<br>2. System Presents from with (name, price, photo, start date, end date, description, is active, list of selected hotels (add 1 or more hotel in this trip by select any hotel stored in the system before), list of nights (add 1 or more nights by set number of nights and select any city stored in the system before)<br>3. User enter needed data then clicks on “Create”<br>4. System checks if the number on nights is correct by comparing it with period between start and end date<br>5. System creates new trip in the system |
| **Alternative Scenario** | In step 4 if the number of nights is not correct the system returns error message and redirect user to add trip page to correct data |



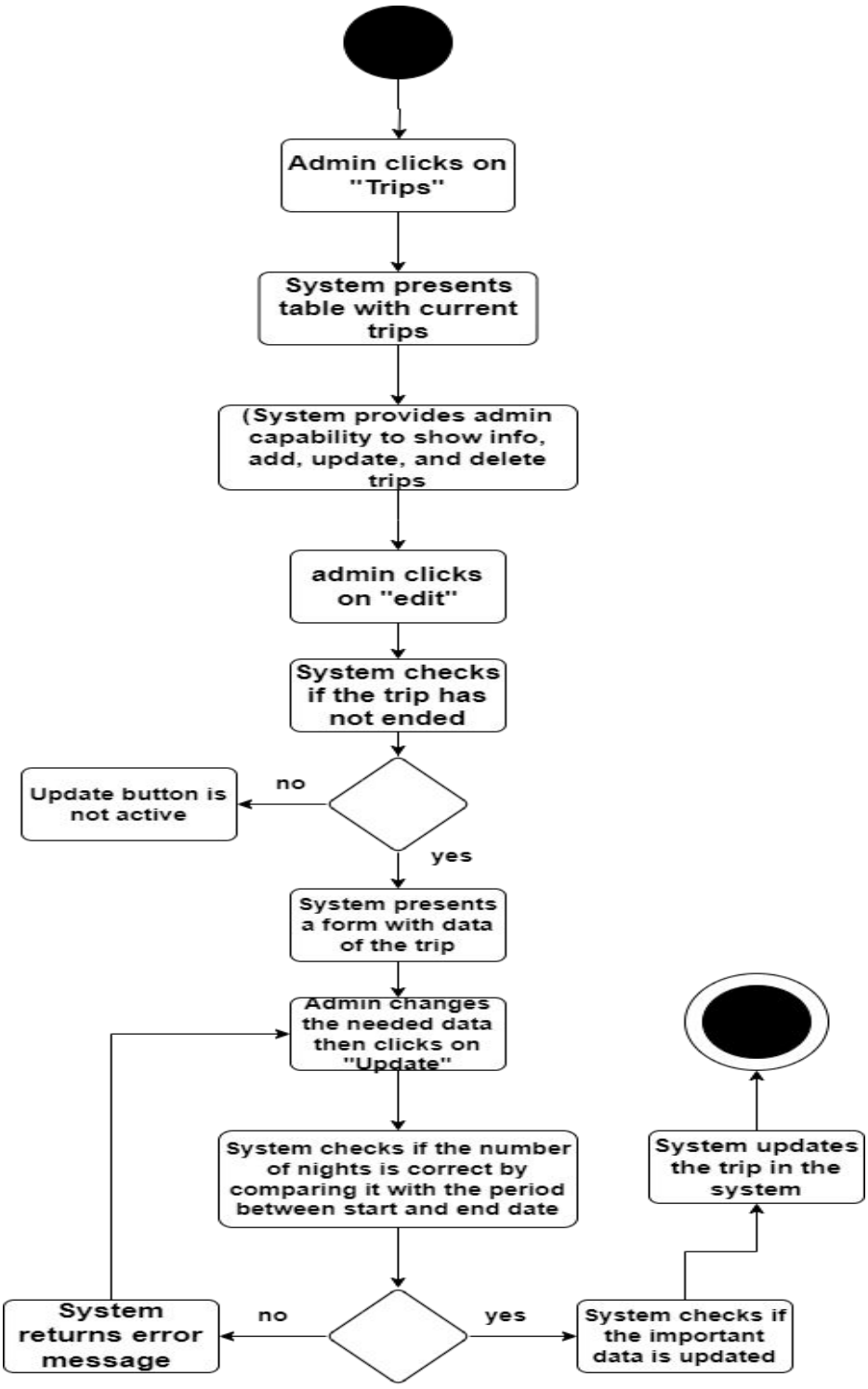

**Activity diagram(Add Trip) – Figure 30**



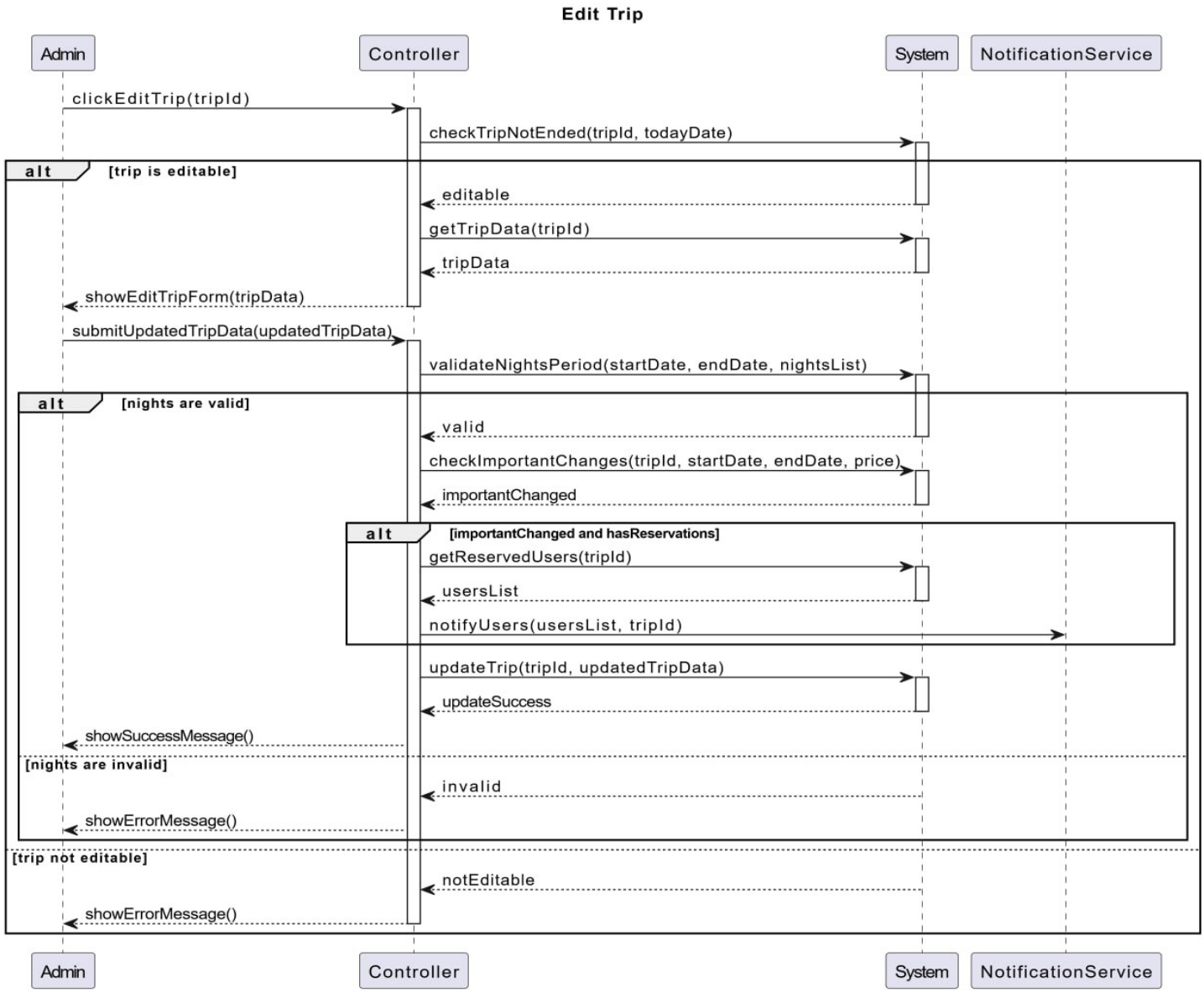

**Sequence diagram(Add Trip) – Figure 31**

### 5.1.14 Update Trip

| | |
|---|---|
| **Use Case ID** | UC-17 |
| **Use Case** | Update Trip |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, and in manage trips page |
| **Main Scenario** | 1. User clicks on “Edit” in any trip<br>2. System checks if the trip does not end (start date before today)<br>3. System Presents from with data of trip (presented before in add)<br>4. User change needed data then clicks on “Update”<br>5. System checks if the number on nights is correct by comparing it with period between start and end date<br>6. System checks if the important data (start date, end date, price) is updated and any user have been reserved this trip the system shall notify the users<br>7. System update trip in the system |
| **Alternative Scenario** | In step 2 if the trip ends the update button is not active |



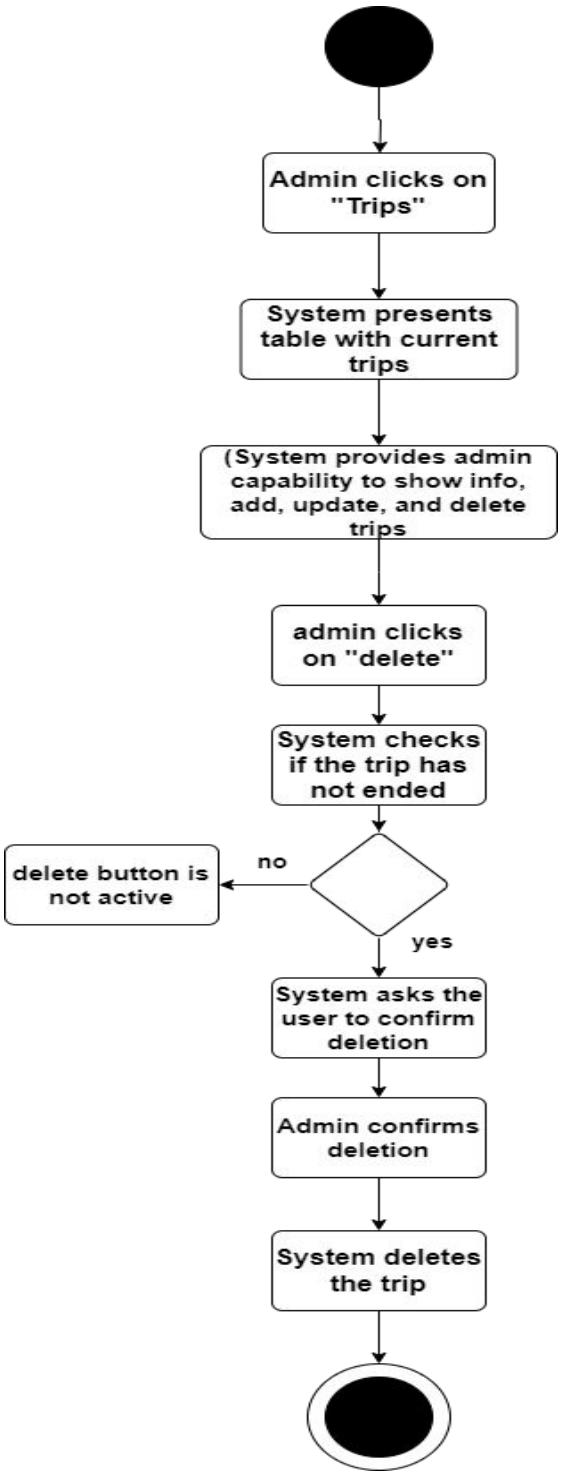

**Activity diagram(Update Trip) – Figure 32**



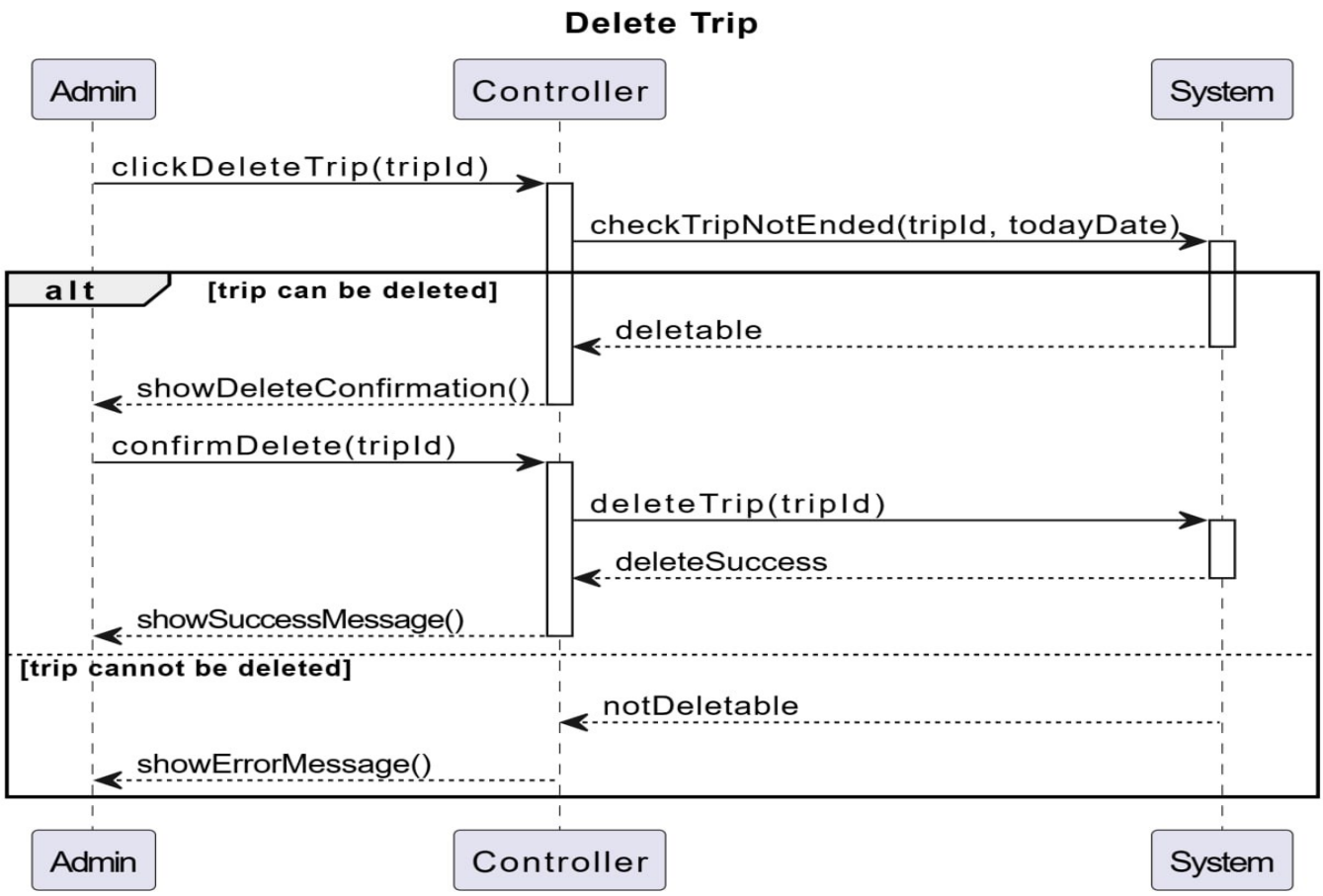

**Sequence diagram(Update Trip) – Figure 33**

### 5.1.15 Delete Trip

| | |
|---|---|
| **Use Case ID** | UC-18 |
| **Use Case** | Delete Trip |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, and in manage trips page |
| **Main Scenario** | 1. User clicks on “Delete” in any trip<br>2. System checks if the trip does not end (start date before today)<br>3. System ask user to delete (confirm)<br>4. User confirm delete<br>5. System delete trip |
| **Alternative Scenario** | In step 2 if the trip ends the update button is not active |



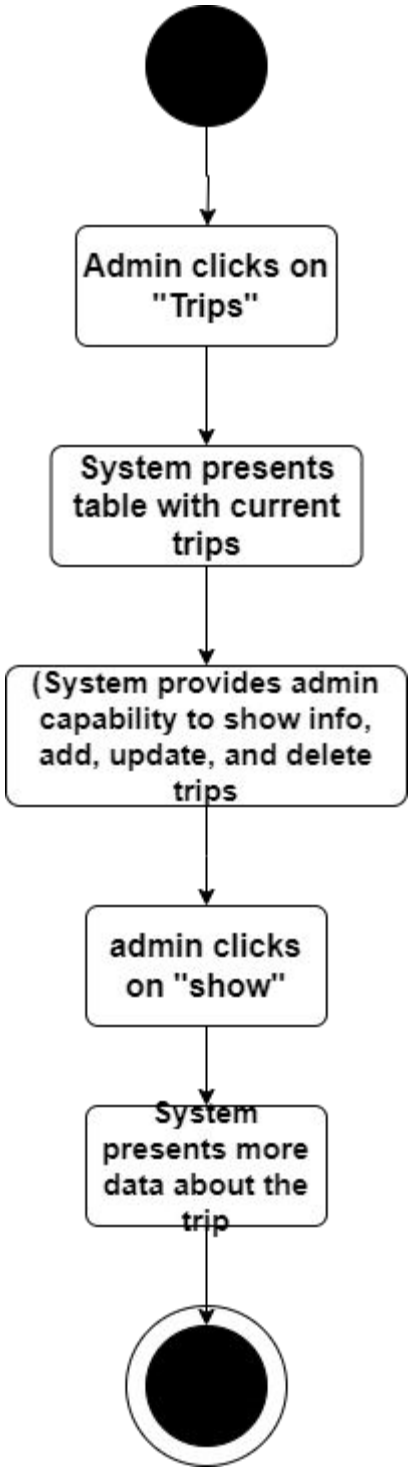

**Activity diagram(Delete Trip) – Figure 34**



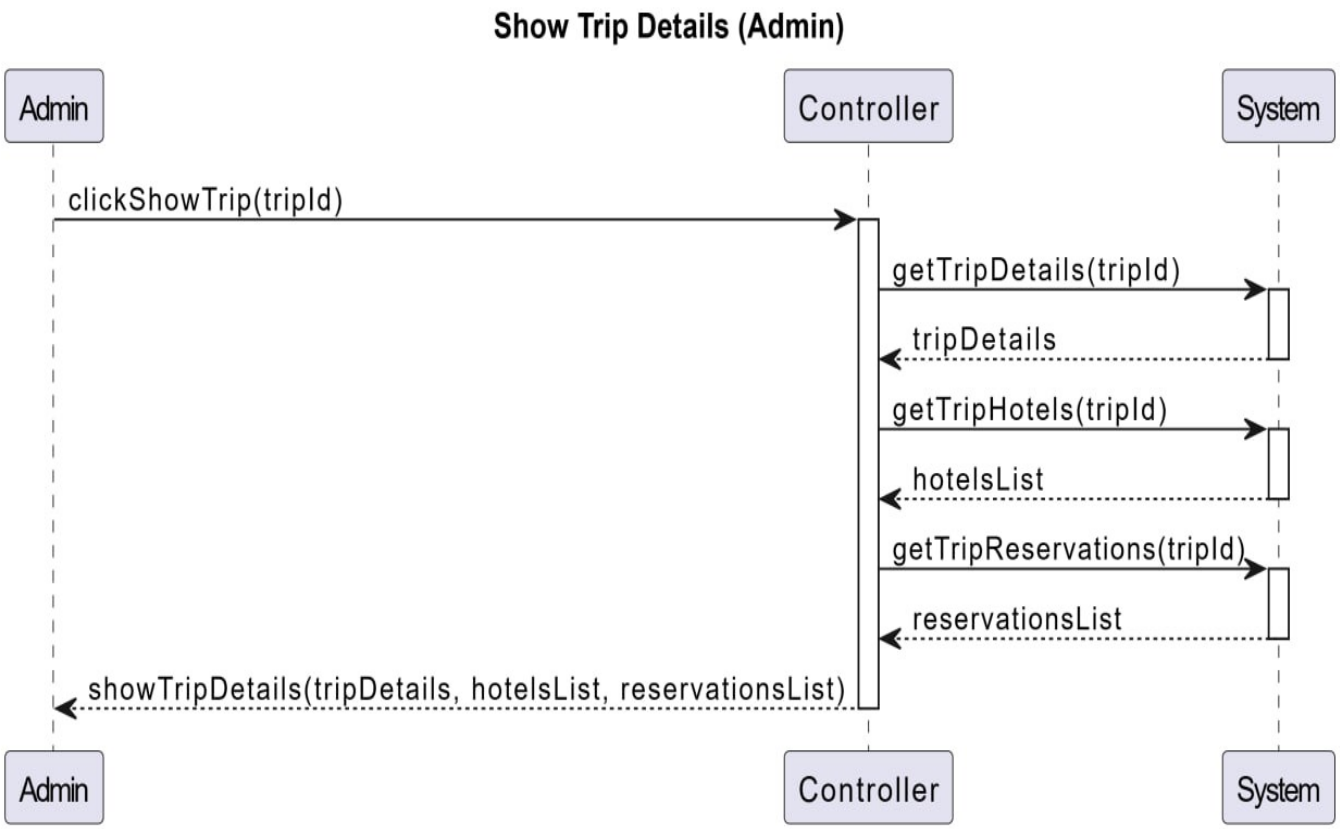

**Sequence diagram(Delete Trip) – Figure 35**

### 5.1.16 View Trip

| | |
|---|---|
| **Use Case ID** | UC-19 |
| **Use Case** | View Trip |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, and in manage trips page |
| **Main Scenario** | 1. User clicks on “Show” in any trip<br>2. System presents more data about trip (name, price, photo, start date, end date, is active, list of nights (e.g. 3 nights in Makkah), description, list of hotels info (name, location (as map), room description, photos), list of reservations (name of user, email, number of tickets, total price)) |
| **Alternative Scenario** | - |



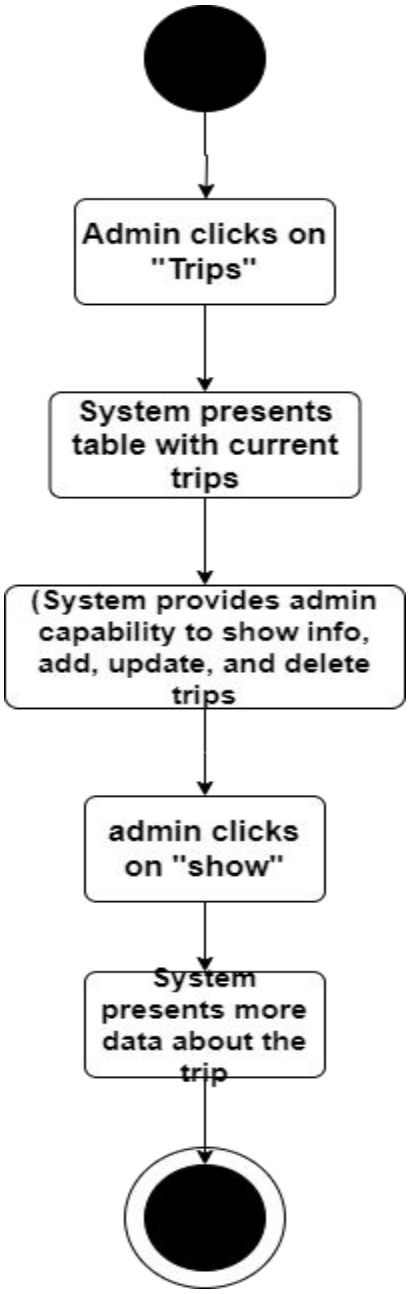

**Activity diagram(View Trip) – Figure 35**



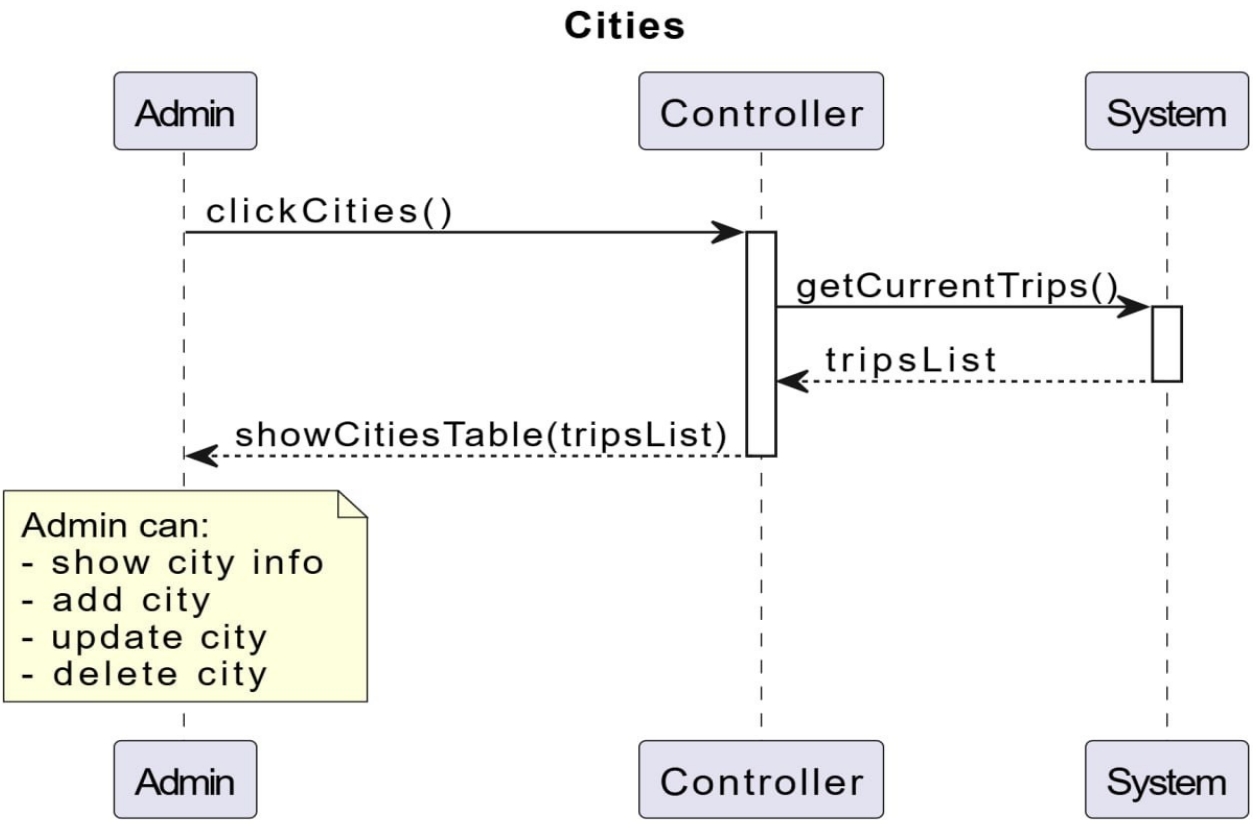

**Sequence diagram(View Trip) – Figure 36**

### 5.1.17 Manage Cities

| | |
|---|---|
| **Use Case ID** | UC-20 |
| **Use Case** | Manage Cities |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Cities”<br>2. System Presents table with current trips<br>3. System provides admin capability to show info, add, update and delete cities |
| **Alternative Scenario** | - |



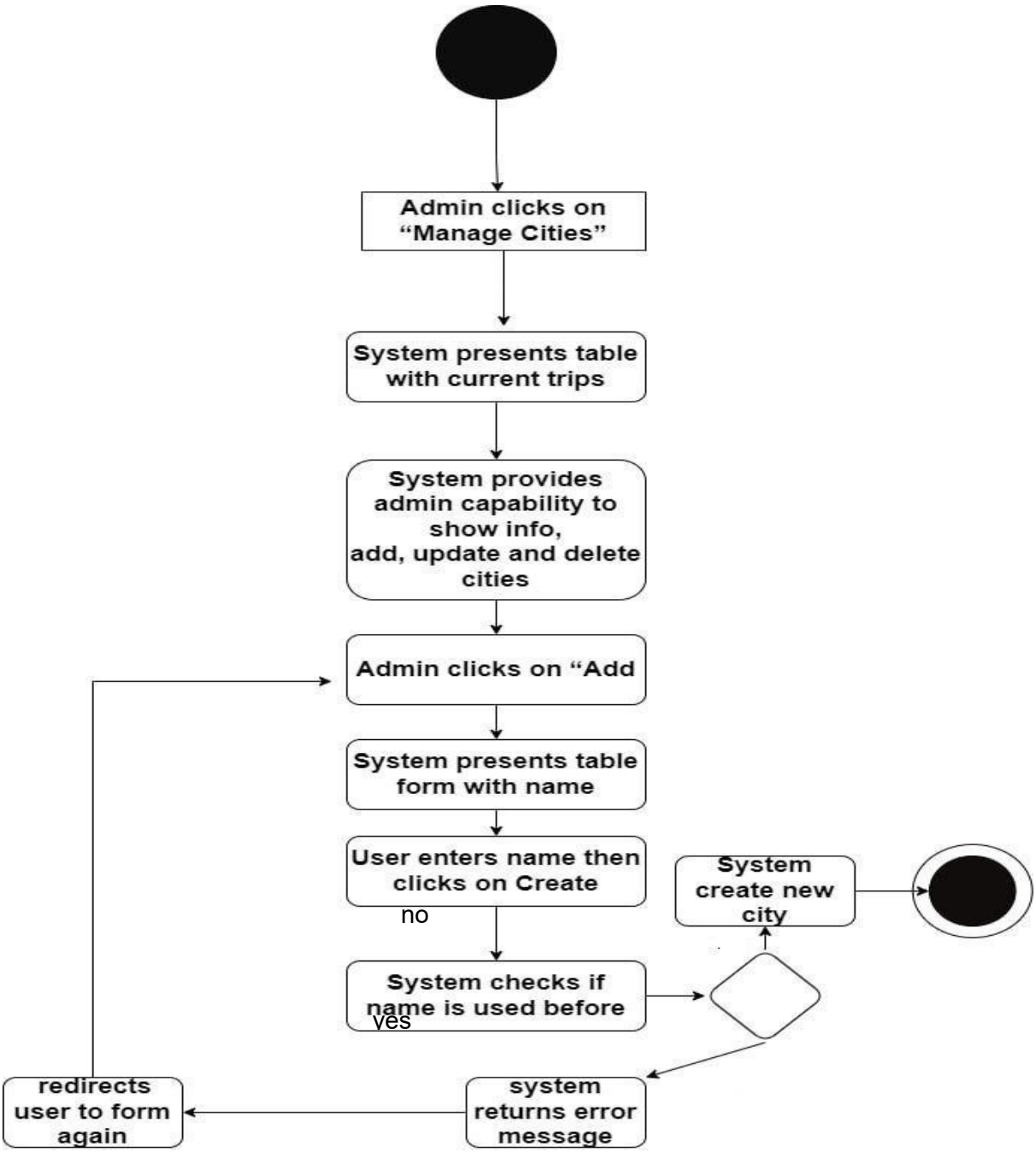

**Activity diagram(Manage Cities) – Figure 37**



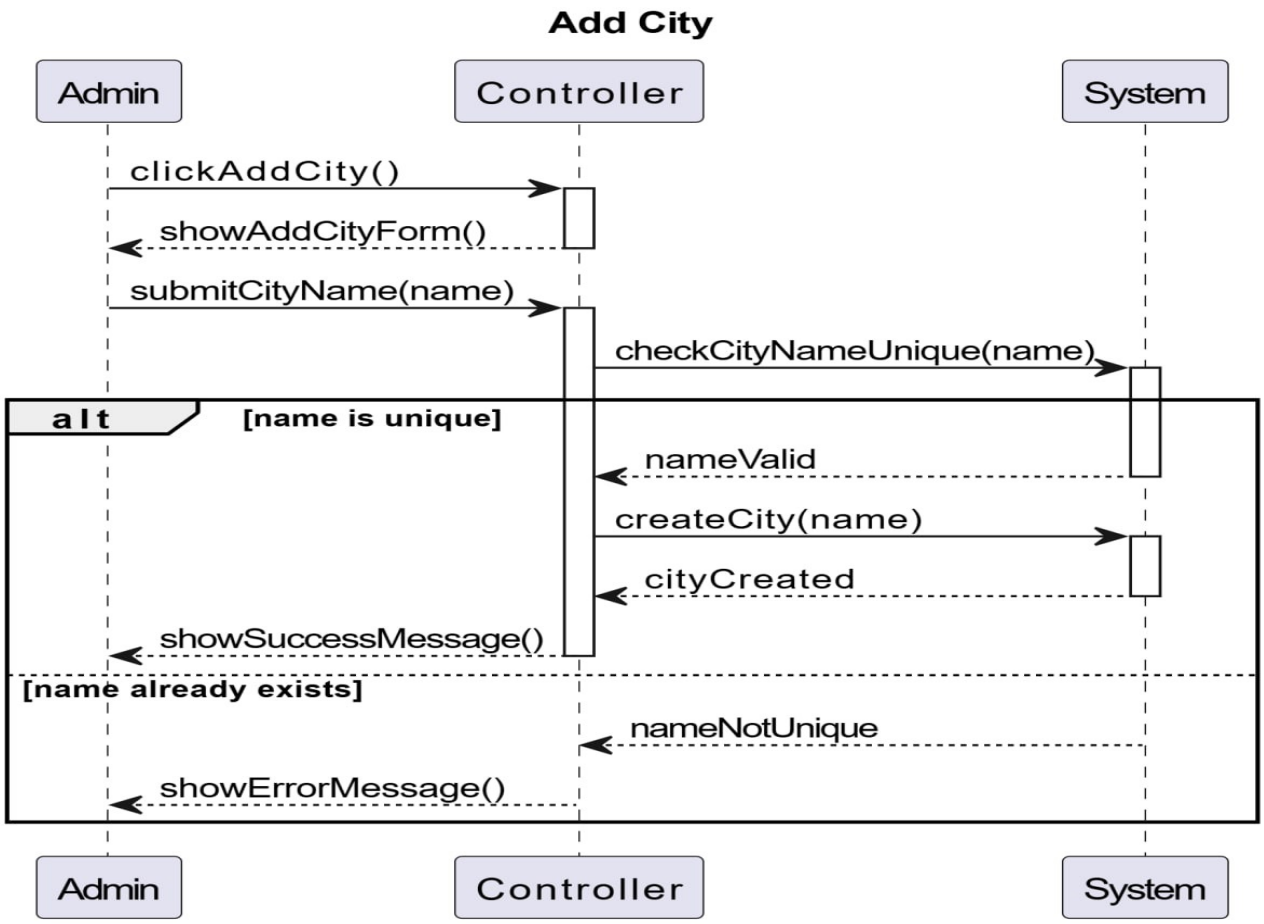

**Sequence diagram(Manage Cities) – Figure 38**

### 5.1.18 Add City

| | |
|---|---|
| **Use Case ID** | UC-21 |
| **Use Case** | Add City |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Manage Cities page |
| **Main Scenario** | 1. User clicks on “Add”<br>2. System Presents table form with name field<br>3. User enter name then clicks on “Create”<br>4. System checks if name is used before<br>5. System creates new city |
| **Alternative Scenario** | In step 4 if the name is not unique system return error message and redirect user to form again |



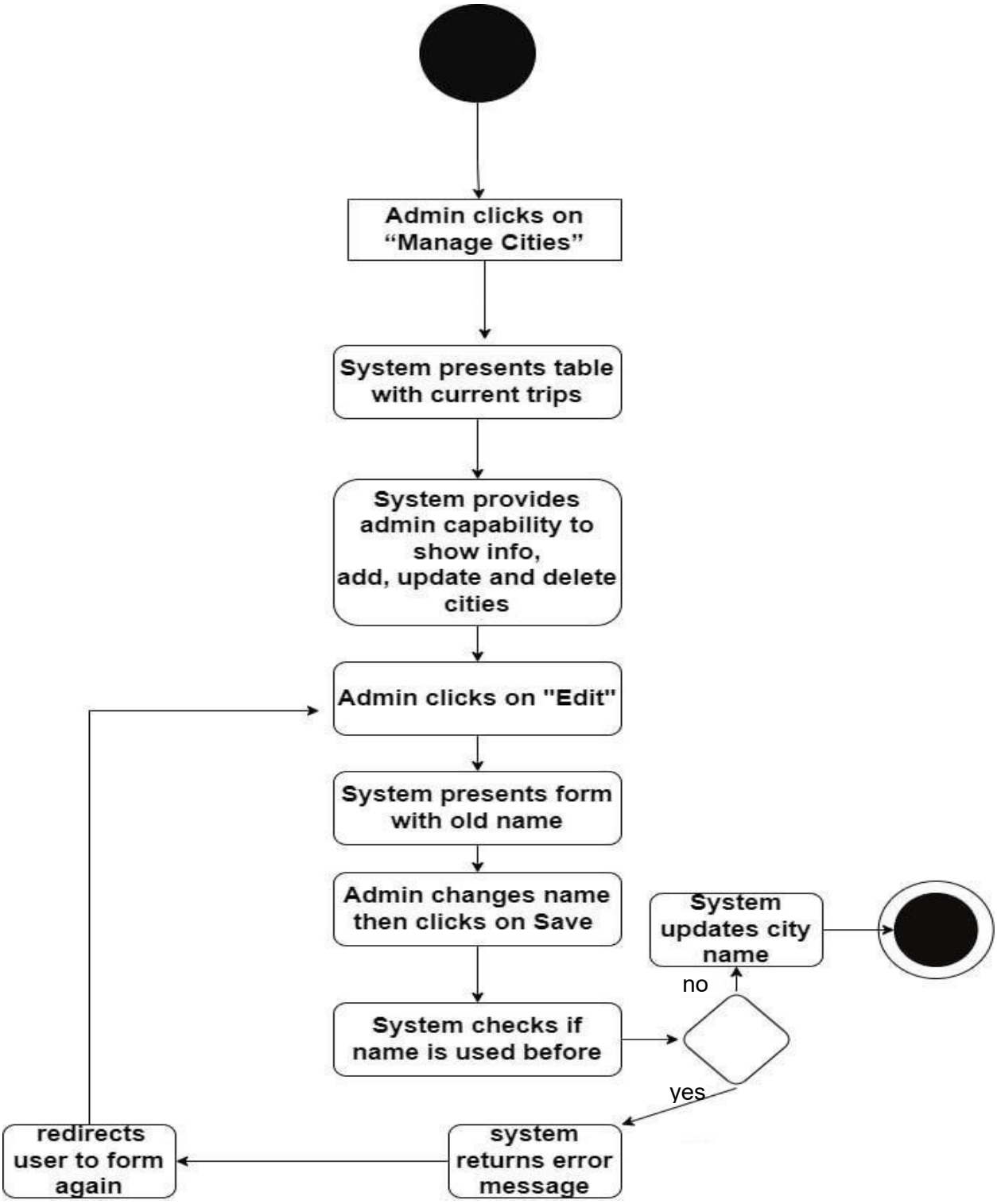

**Activity diagram(Add City) – Figure 39**



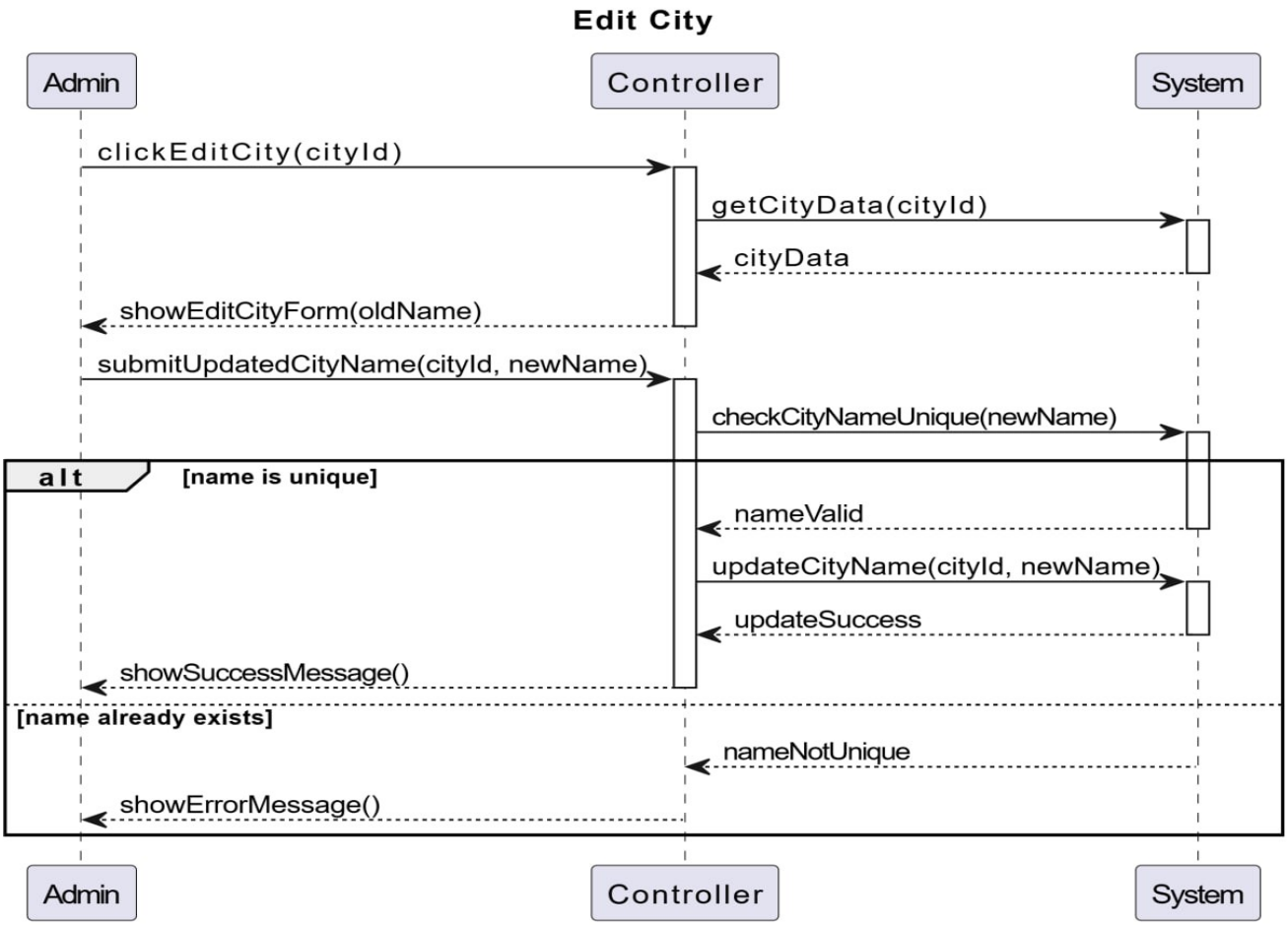

**Sequence diagram(Add City) – Figure 40**

### 5.1.19 Update City

| | |
|---|---|
| **Use Case ID** | UC-22 |
| **Use Case** | Update City |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Manage Cities page |
| **Main Scenario** | 1. User clicks on “Edit” in any city<br>2. System presents form with old name<br>3. User change name then clicks on “Save”<br>4. System checks if name is used before<br>5. System update city name |
| **Alternative Scenario** | In step 4 if the name is not unique system return error message and redirect user to form again |



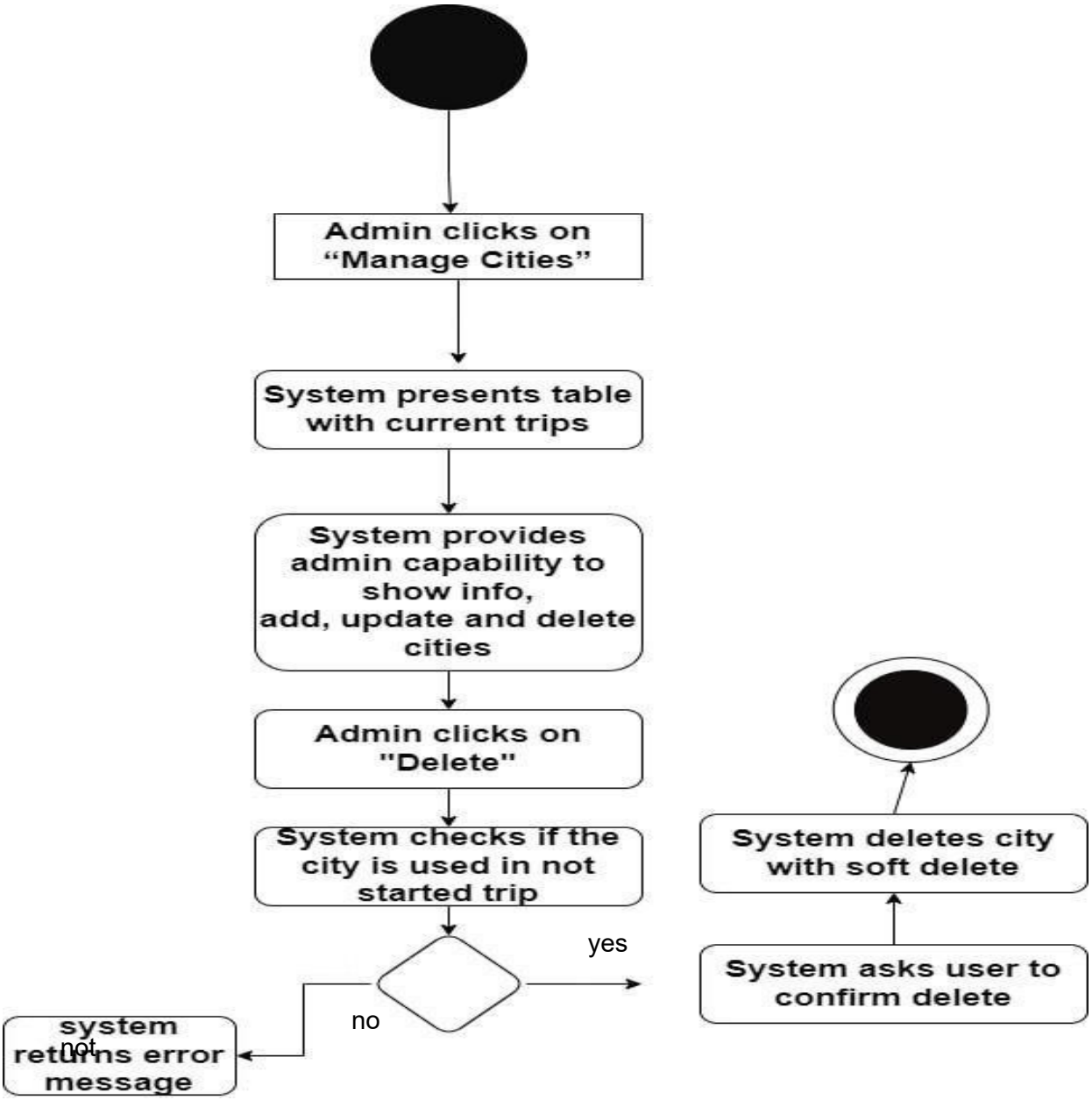

**Activity diagram(Update City) – Figure 41**



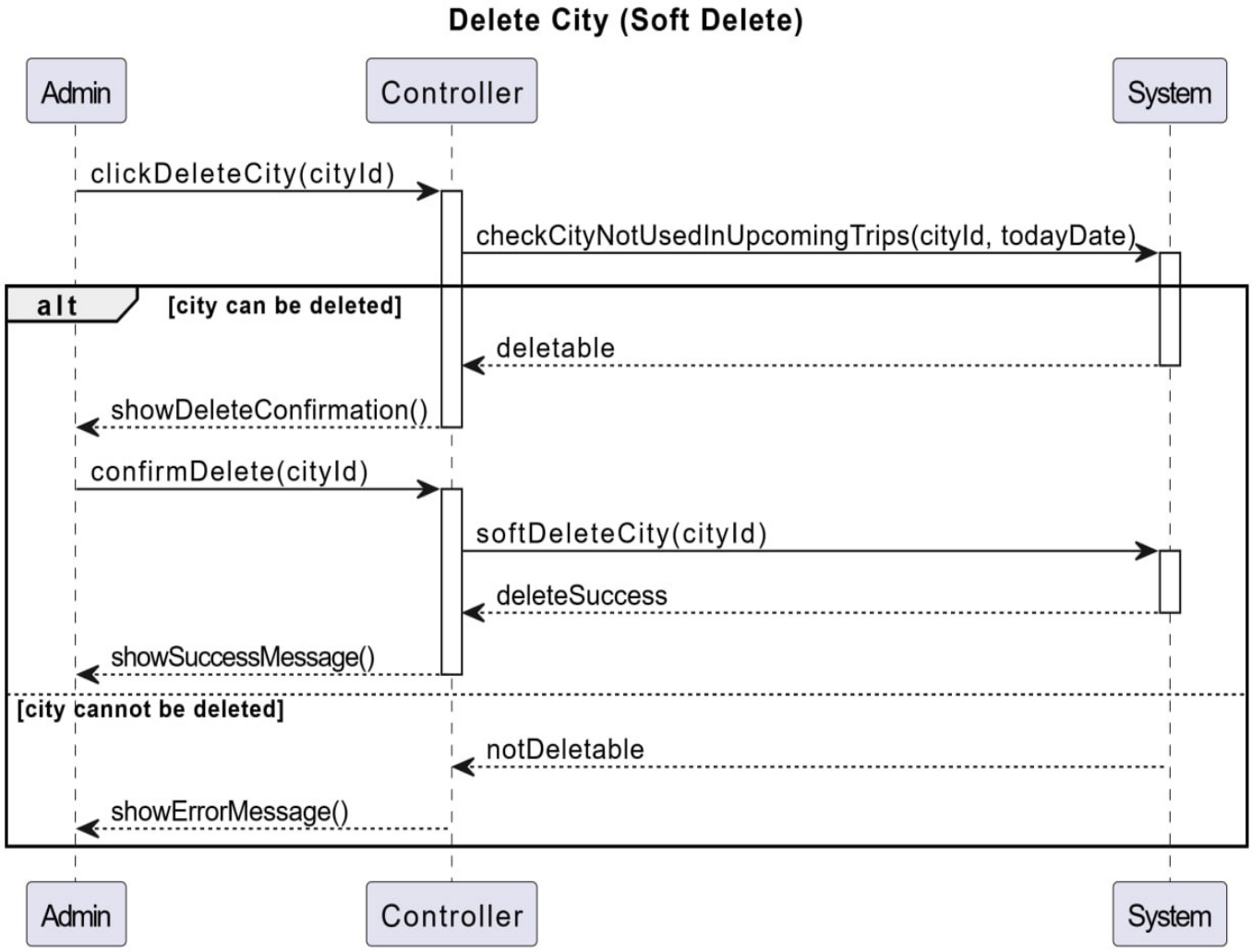

**Sequence diagram(Update City) – Figure 42**

### 5.1.20 Delete City

| | |
|---|---|
| **Use Case ID** | UC-23 |
| **Use Case** | Delete City |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Manage Cities page |
| **Main Scenario** | 1. User clicks on “Delete” in any city<br>2. System checks if the city does not use in not started trips<br>3. System ask user to delete (confirm)<br>4. User confirm delete<br>5. System deletes city (soft delete) |
| **Alternative Scenario** | In step 2 if the city used in trips not started yet, system return error message |



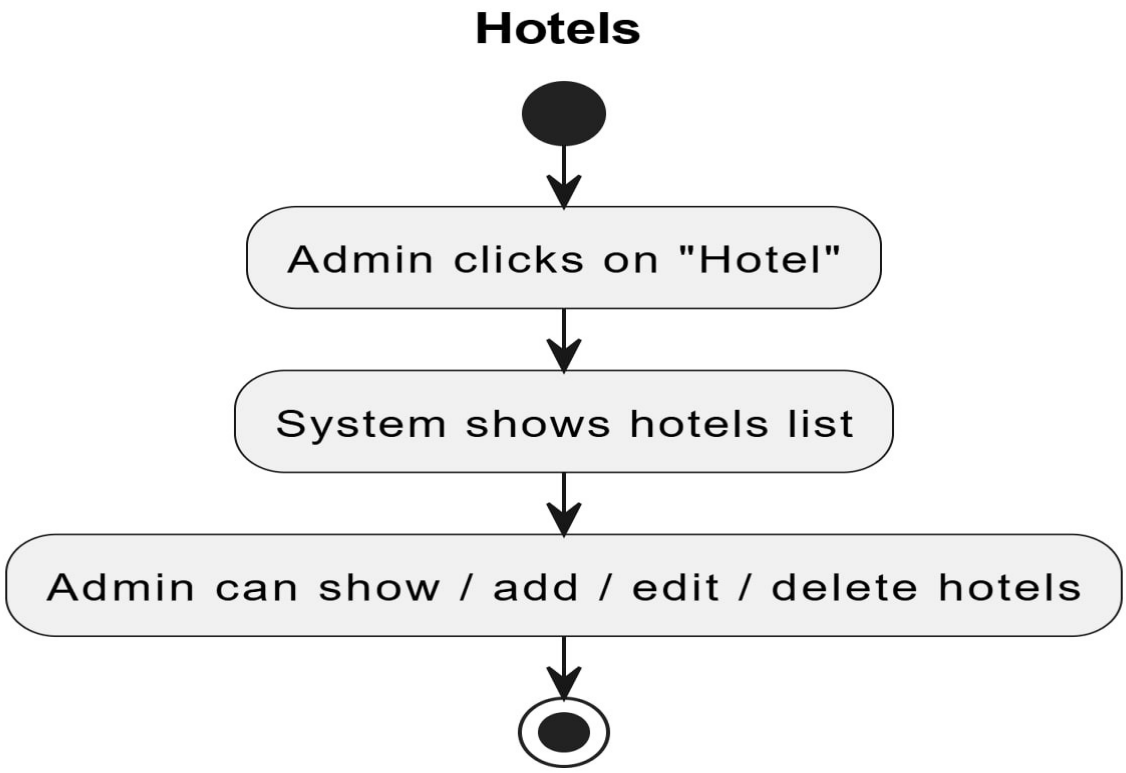

**Activity diagram(Delete City) – Figure 43**



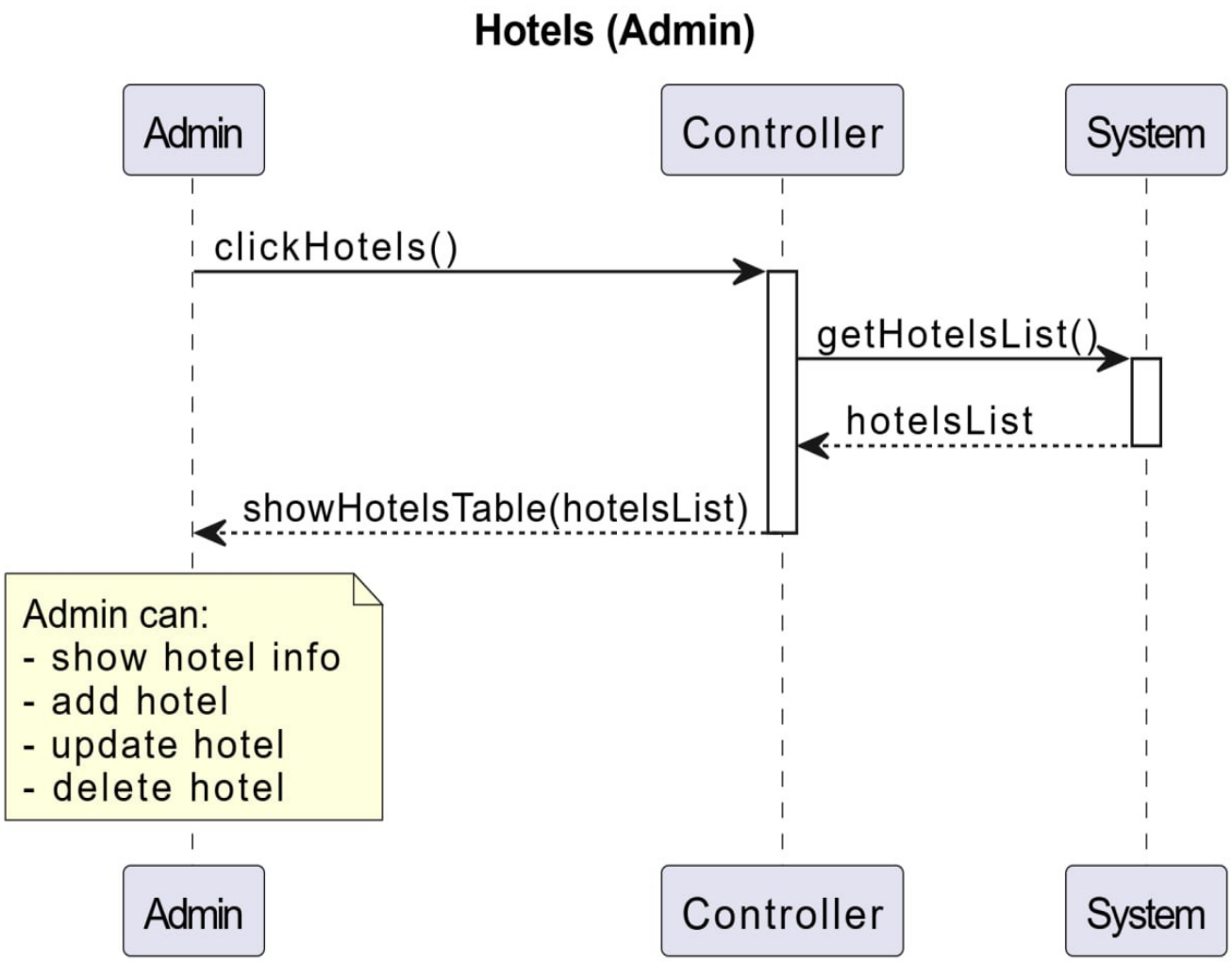

**Sequence diagram(Delete City) – Figure 44**

### 5.1.21 Manage Hotels

| | |
|---|---|
| **Use Case ID** | UC-24 |
| **Use Case** | Manage Hotels |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Hotel”<br>2. System Presents table with current hotels (name, location (as map), room description, photos)<br>3. System provides admin capability to show info, add, update and delete hotels |
| **Alternative Scenario** | - |



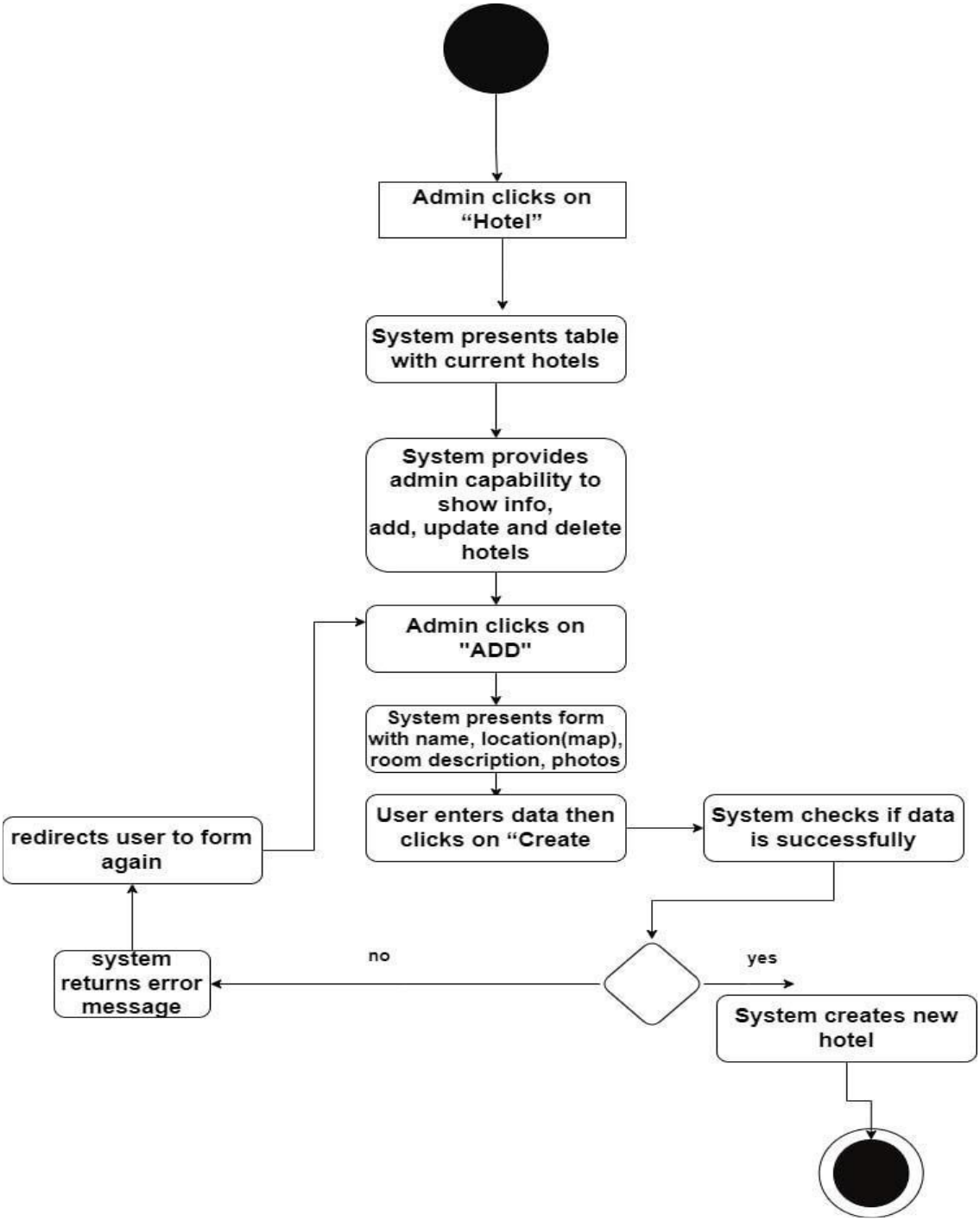

**Activity diagram(Manage Hotels) – Figure 45**



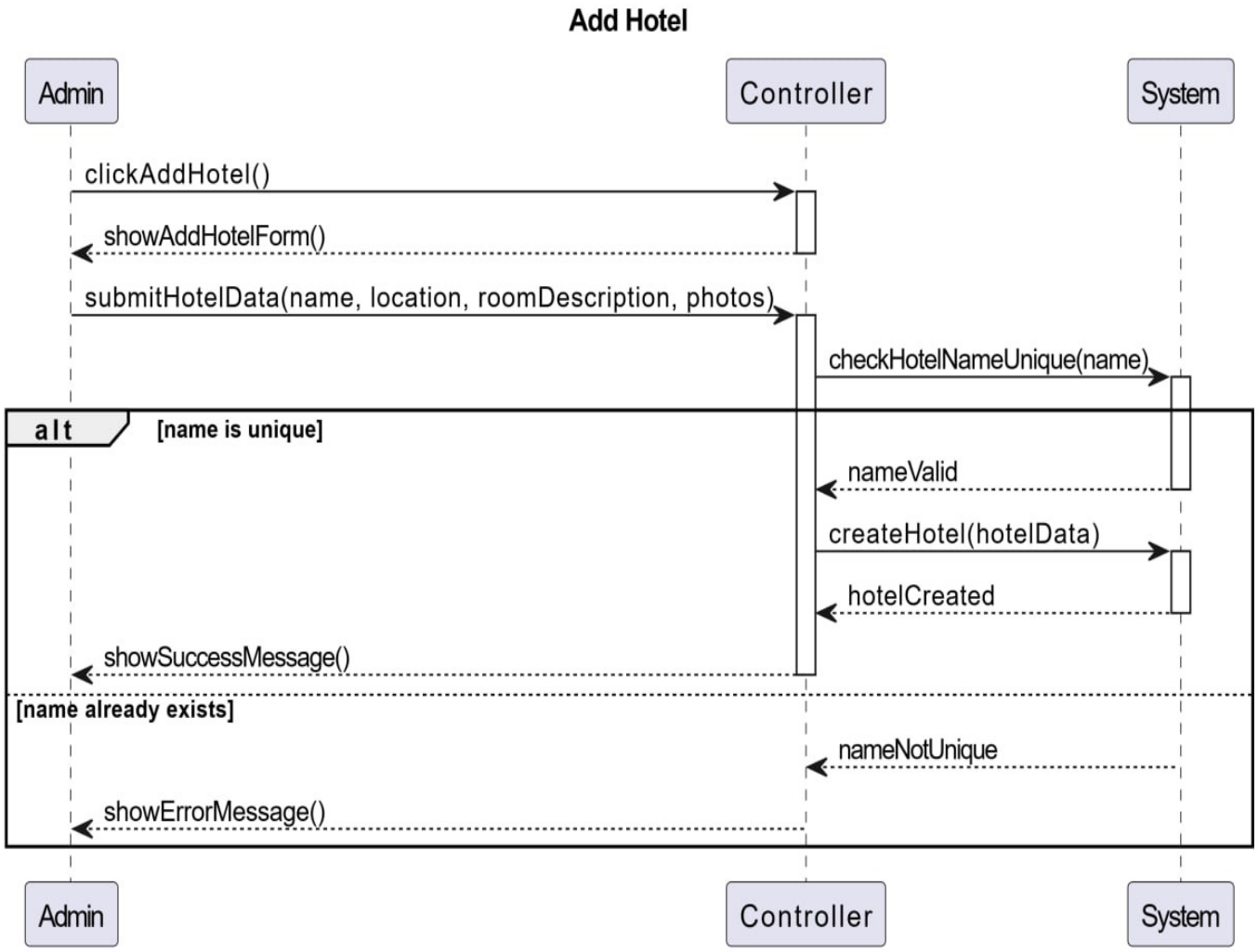

**Sequence diagram(Manage Hotels) – Figure 46**

### 5.1.22 Add Hotels

| | |
|---|---|
| **Use Case ID** | UC-25 |
| **Use Case** | Add Hotel |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Manage Hotels page |
| **Main Scenario** | 1. User clicks on “Add”<br>2. System presents form with (name, location (as map), room description, photos)<br>3. User enter date then clicks on “Create”<br>4. System checks if name is used before<br>5. System creates new hotel |
| **Alternative Scenario** | In step 4 if the name is not unique system return error message and redirect user to form again |



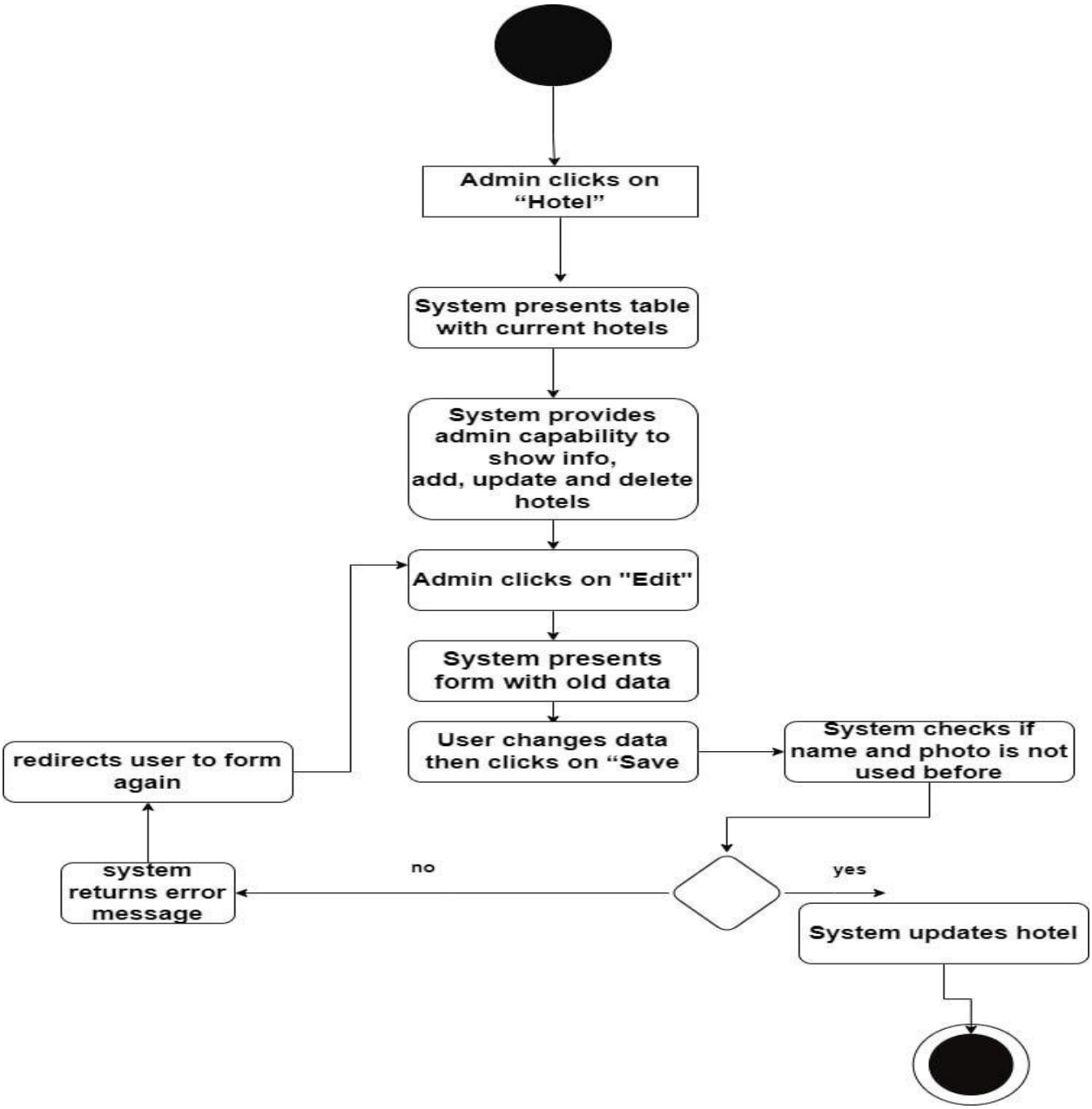

**Activity diagram(Add Hotels) – Figure 47**



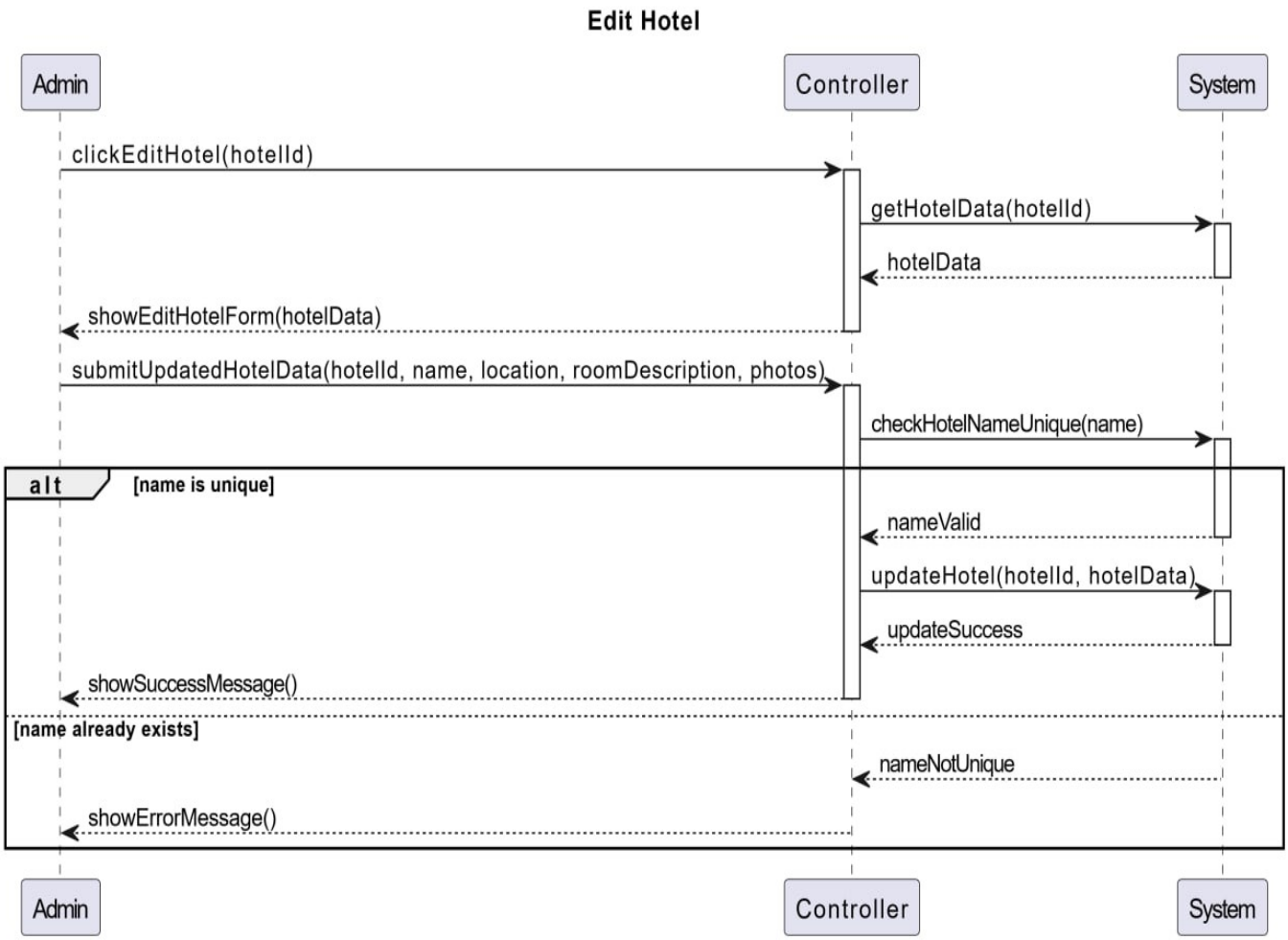

**Sequence diagram(Add Hotels) – Figure 48**

### 5.1.23 Update Hotel

| | |
|---|---|
| **Use Case ID** | UC-26 |
| **Use Case** | Update Hotel |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Manage Hotels page |
| **Main Scenario** | 1. User clicks on “Edit” in any city<br>2. System presents form with old data<br>3. User change data then clicks on “Save”<br>4. System checks if name is used before<br>5. System update hotel |
| **Alternative Scenario** | In step 4 if the name is not unique system return error message and redirect user to form again |



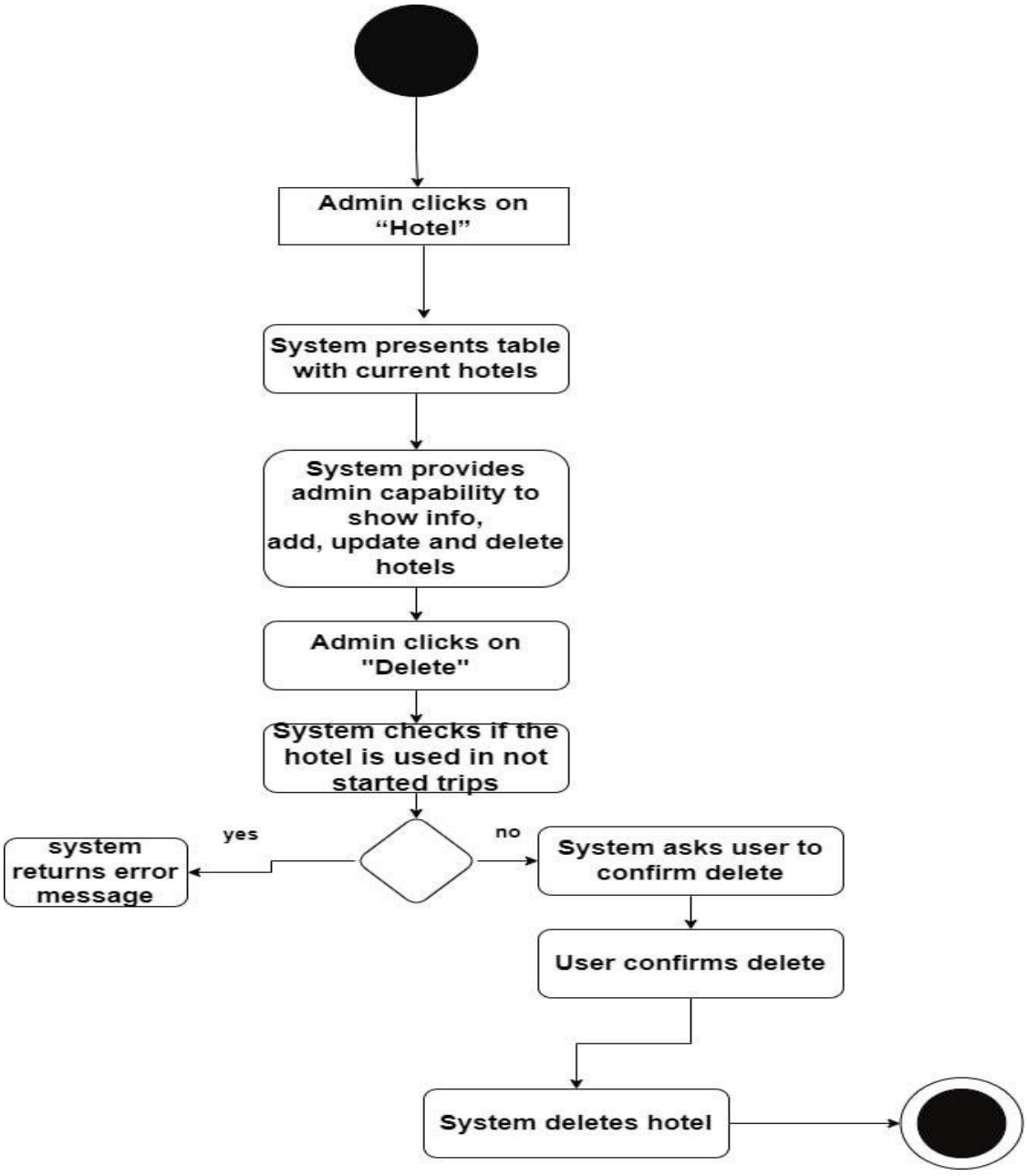

**Activity diagram(Update Hotels) – Figure 49**



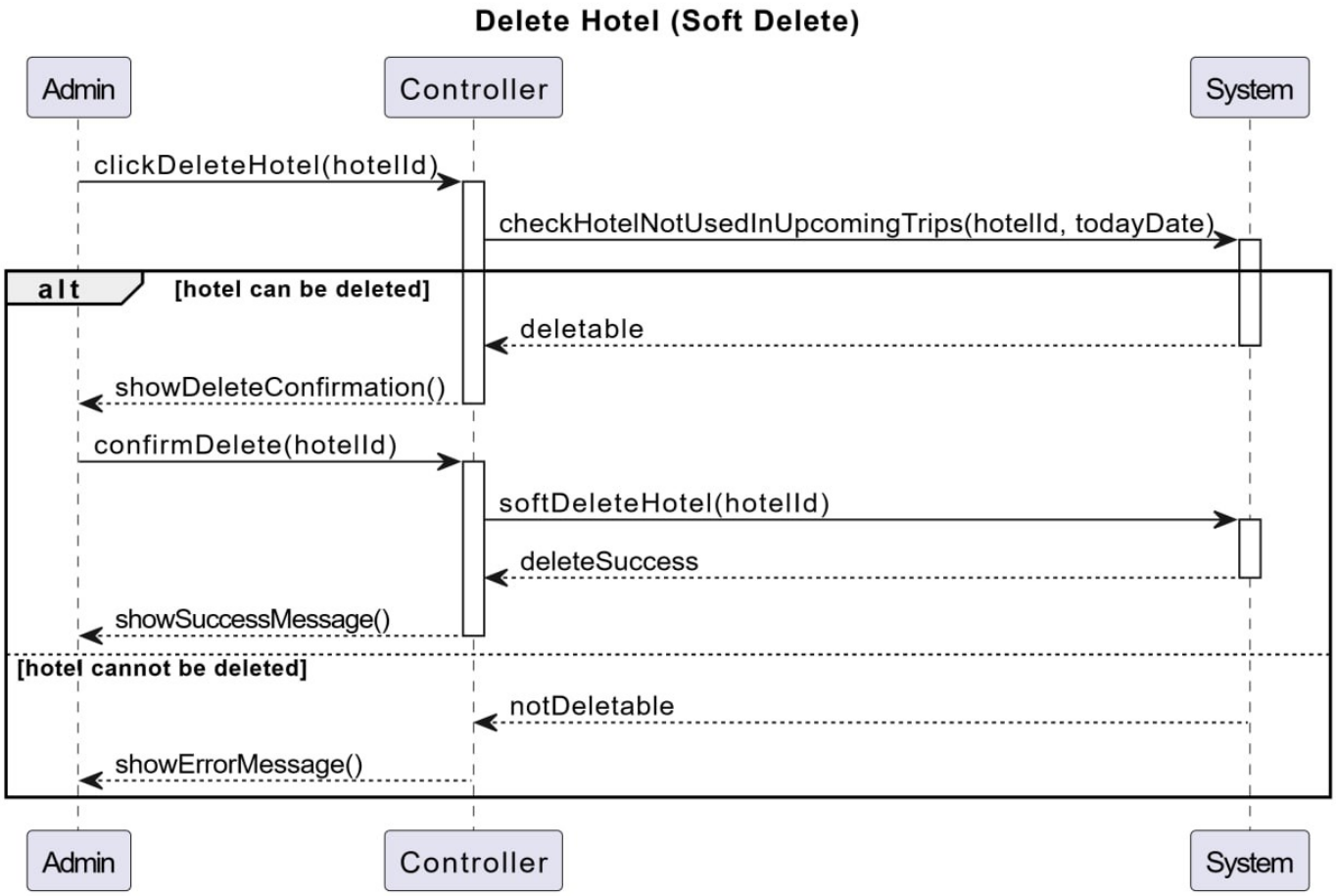

**Sequence diagram(Update Hotels) – Figure 50**

### 5.1.24 Delete Hotel

| | |
|---|---|
| **Use Case ID** | UC-27 |
| **Use Case** | Delete Hotel |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Manage Hotels page |
| **Main Scenario** | 1. User clicks on “Delete” in any hotel<br>2. System checks if the hotel does not use in not started trips<br>3. System ask user to delete (confirm)<br>4. User confirm delete<br>5. System deletes city (soft delete) |
| **Alternative Scenario** | In step 2 if the hotel used in trips not started yet, system return error message |



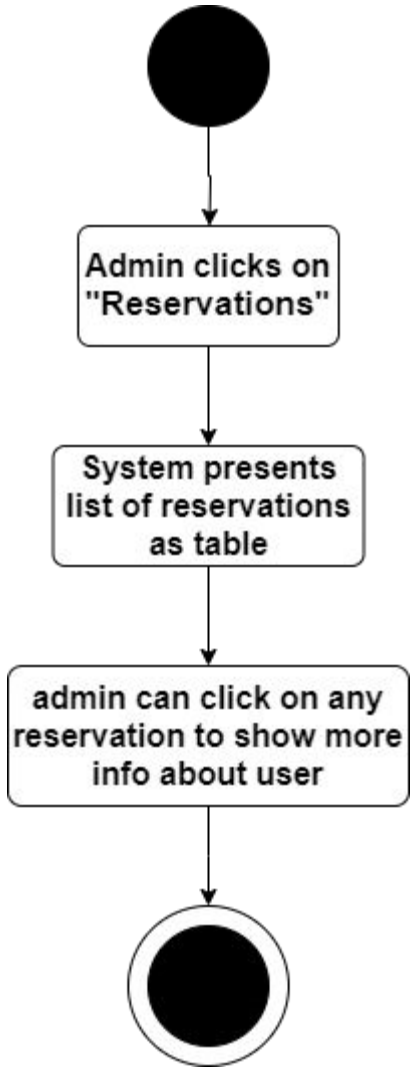

**Activity diagram(Delete Hotels) – Figure 51**




**Sequence diagram(Delete Hotels) – Figure 52**

### 5.1.25 Manage Reservations

| | |
|---|---|
| **Use Case ID** | UC-28 |
| **Use Case** | Manage Reservations |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in |
| **Main Scenario** | 1. User clicks on “Reservations”<br>2. System presents list of reservations as table with (name of trip, name of user, email of user, number of tickets, price of trip, total price, status)<br>3. User can filter reservations by (status, trip)<br>4. User can search via name or email of user<br>5. User can click on any reservation to show more info about user (name, email, phone number, address) via dialog |
| **Alternative Scenario** | - |

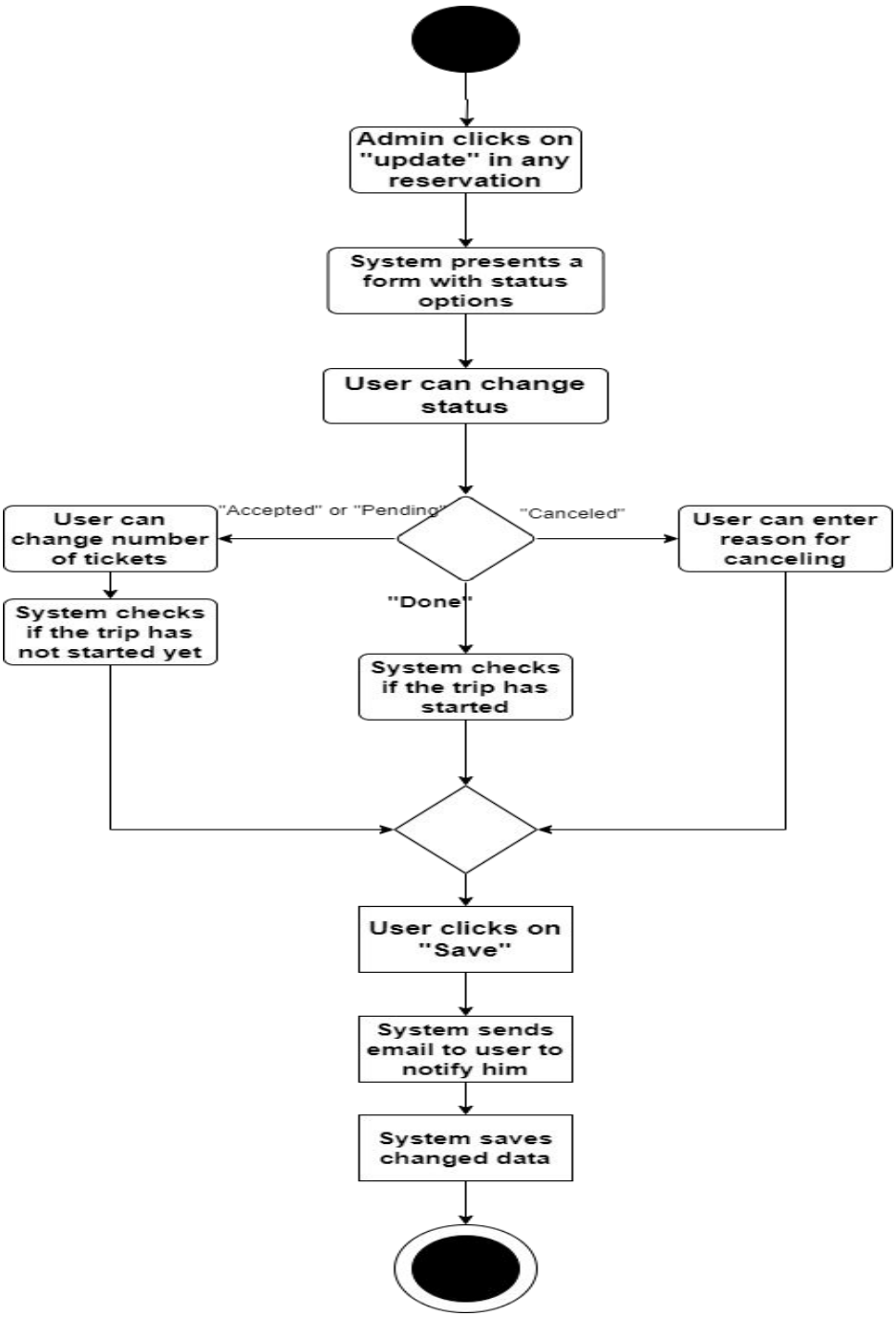



**Activity diagram(**Manage Reservations**) – Figure 53**

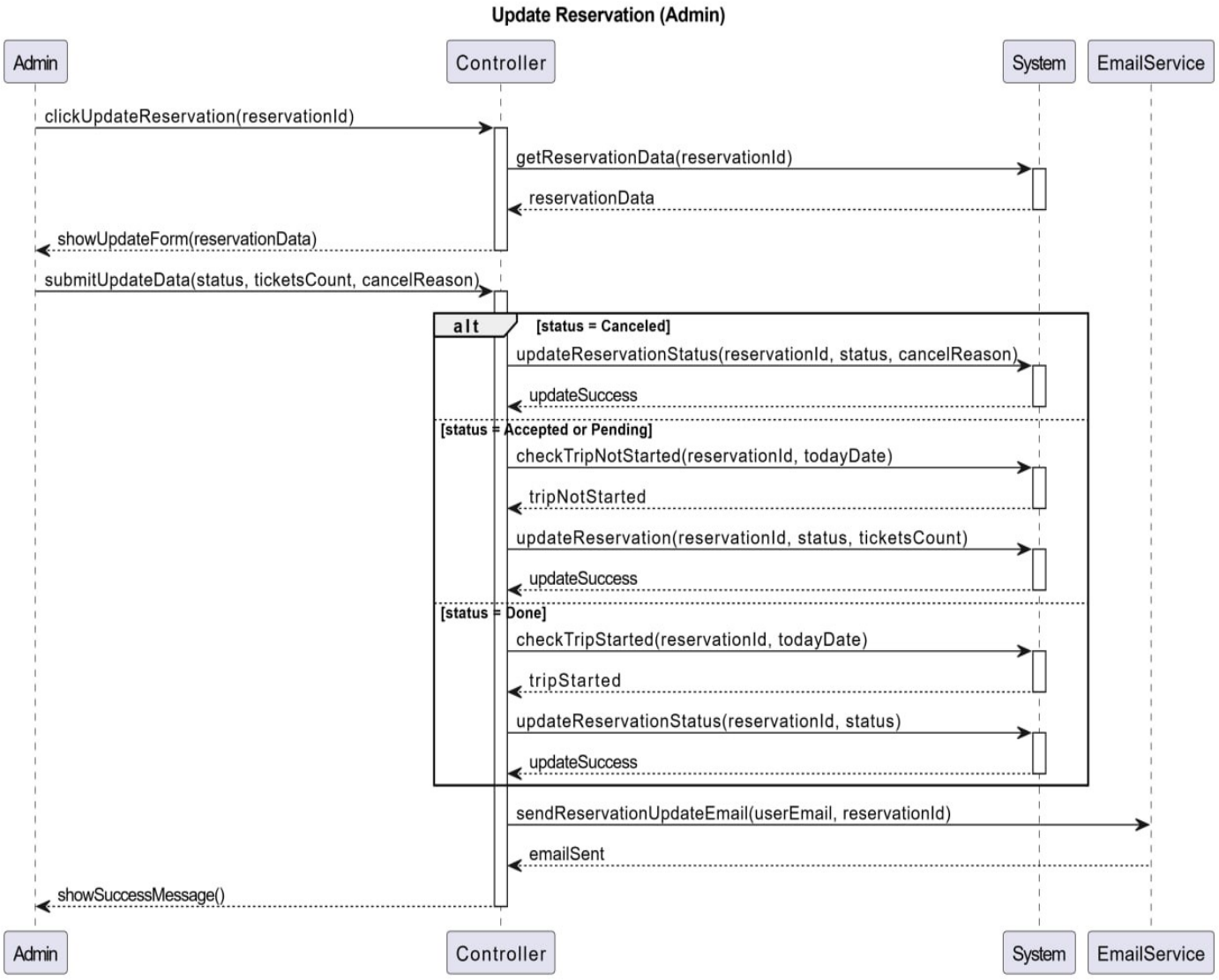

**Sequence diagram(**Manage Reservations**) – Figure 54**

### 5.1.26 Update Reservations

| | |
|---|---|
| **Use Case ID** | UC-29 |
| **Use Case** | Update Reservation |
| **Actor** | Admin |
| **Pre-Condition** | Already logged in, User in Reservations page |
| **Main Scenario** | 1. User clicks on “update” in any reservation<br>2. System presents from with status (Pending, Accepted, Canceled, Done)<br>3. User can change status<br>4. If selected status is “Canceled”, User can enter reason of canceling<br>5. If selected status is “Accepted” or “Pending”, User can change number of tickets but the system shall check if the trip not started yet<br>6. If selected status is “Done” system shall if the trip started<br>7. User clicks on “Save”<br>8. System send email to user to notify him<br>9. System saves changed data |
| **Alternative Scenario** | - |



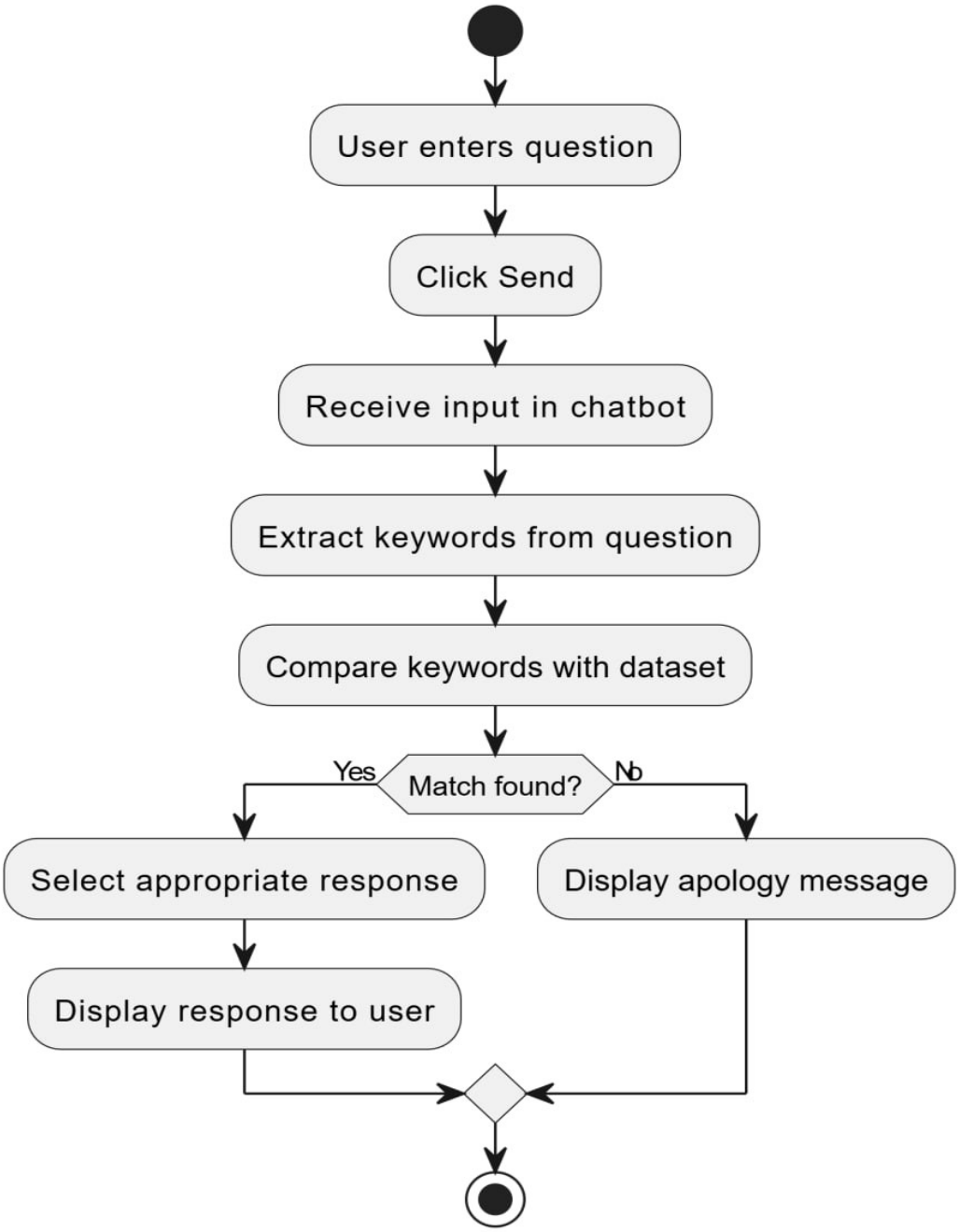

**Activity diagram(**Update Reservations**) – Figure 55**



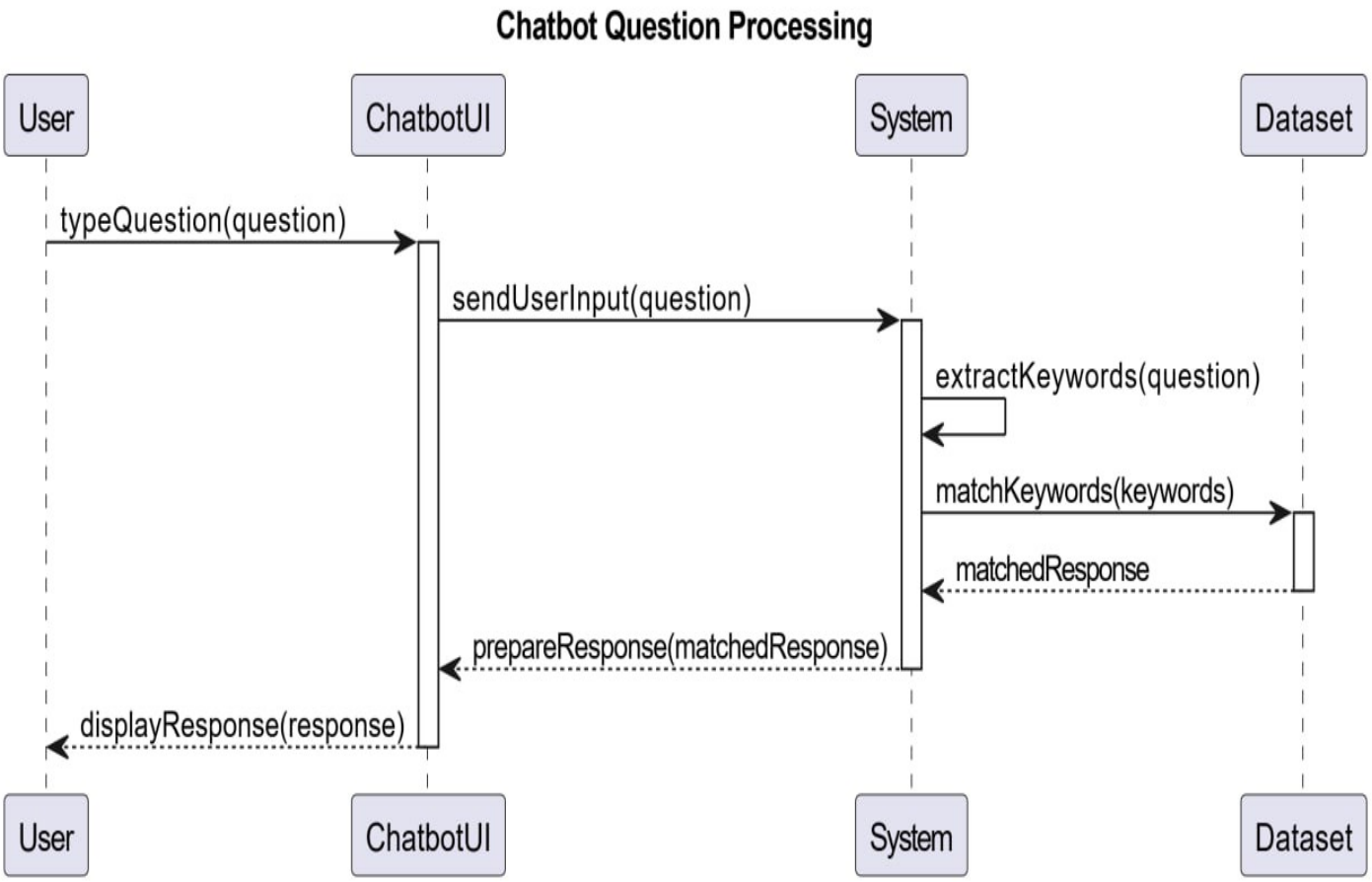

**Sequence diagram(**Update Reservations**) – Figure 56**

5.1.27 Chat Boot

| Item | Description |
|---|---|
| **Use Case ID** | UC-30 |
| **Use Case Name** | Chatbot Question Processing for Hajj & Umrah Website |
| **Actor** | System User (Website User) |
| **Description** | This use case describes how the chatbot processes user questions related to Hajj and Umrah by extracting keywords and matching them with a predefined dataset to generate an appropriate response. |
| **Pre-Conditions** | - The user is browsing the Hajj & Umrah website - Chatbot interface is loaded (Chatbot Component Loaded) - Dataset is available in the Frontend Component - Keyword matching logic is enabled |
| **Trigger** | The user enters a question in the chatbot interface |
| **Main Success Scenario** | 1. The user types a question into the chatbot.<br>2. The system receives the user input.<br>3. The system extracts keywords from the entered question.<br>4. The system compares the extracted keywords with the dataset.<br>5. A suitable keyword match is found.<br>6. The system identifies the response associated with the matched keywords.<br>7. The system displays the response to the user in the chat interface. |
| **Alternative Scenario** | **A1: No Keyword Match Found (at Step 6)** 1. No matching keywords are found in the dataset. 2. The system displays an apology message to the user. *Example:* “Sorry, your question is not related to Hajj or Umrah. Please ask a question within this scope.” |
| **Post-Conditions** | - The chatbot response is displayed to the user - The chat session remains active for further questions |

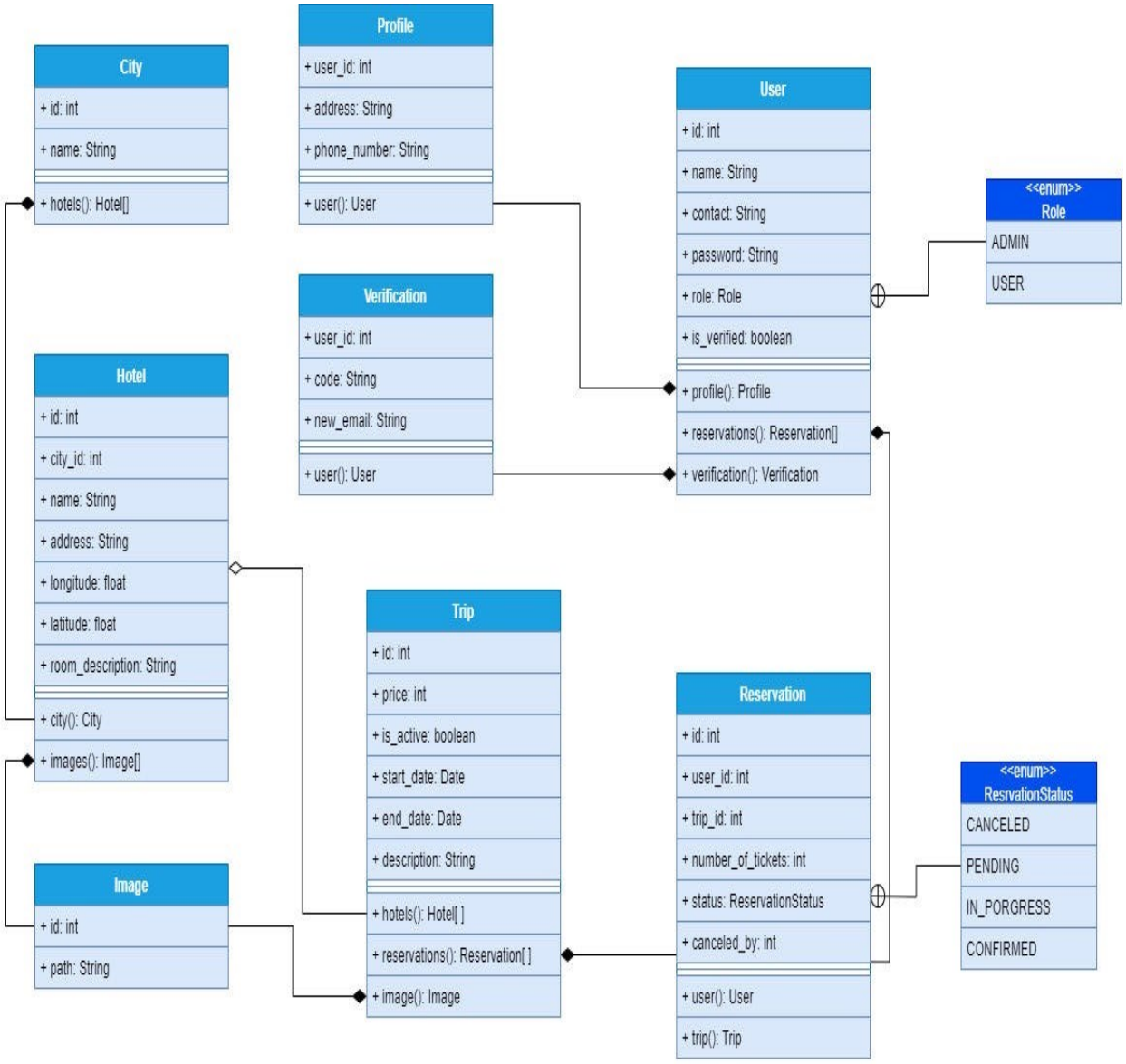



**Activity diagram(**Chat Boot**) – Figure 57**



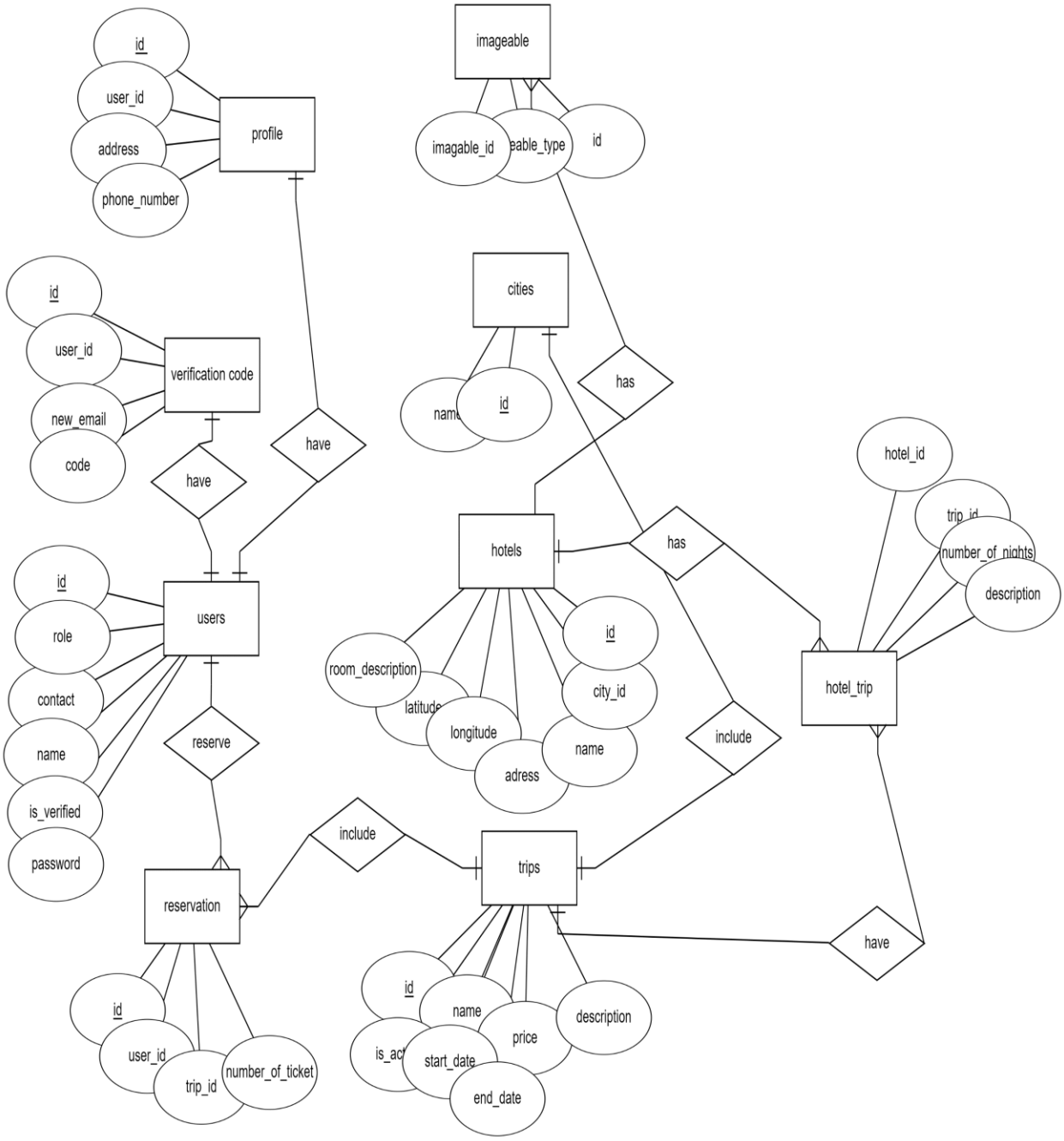

**Sequence diagram(**Chat Boot**) – Figure 58**

## 6. Analysis class diagram




Class diagram - **Figure 58**

## 7. ERD Diagram




ERD diagram - **Figure 59**

## 8. Table Diagram

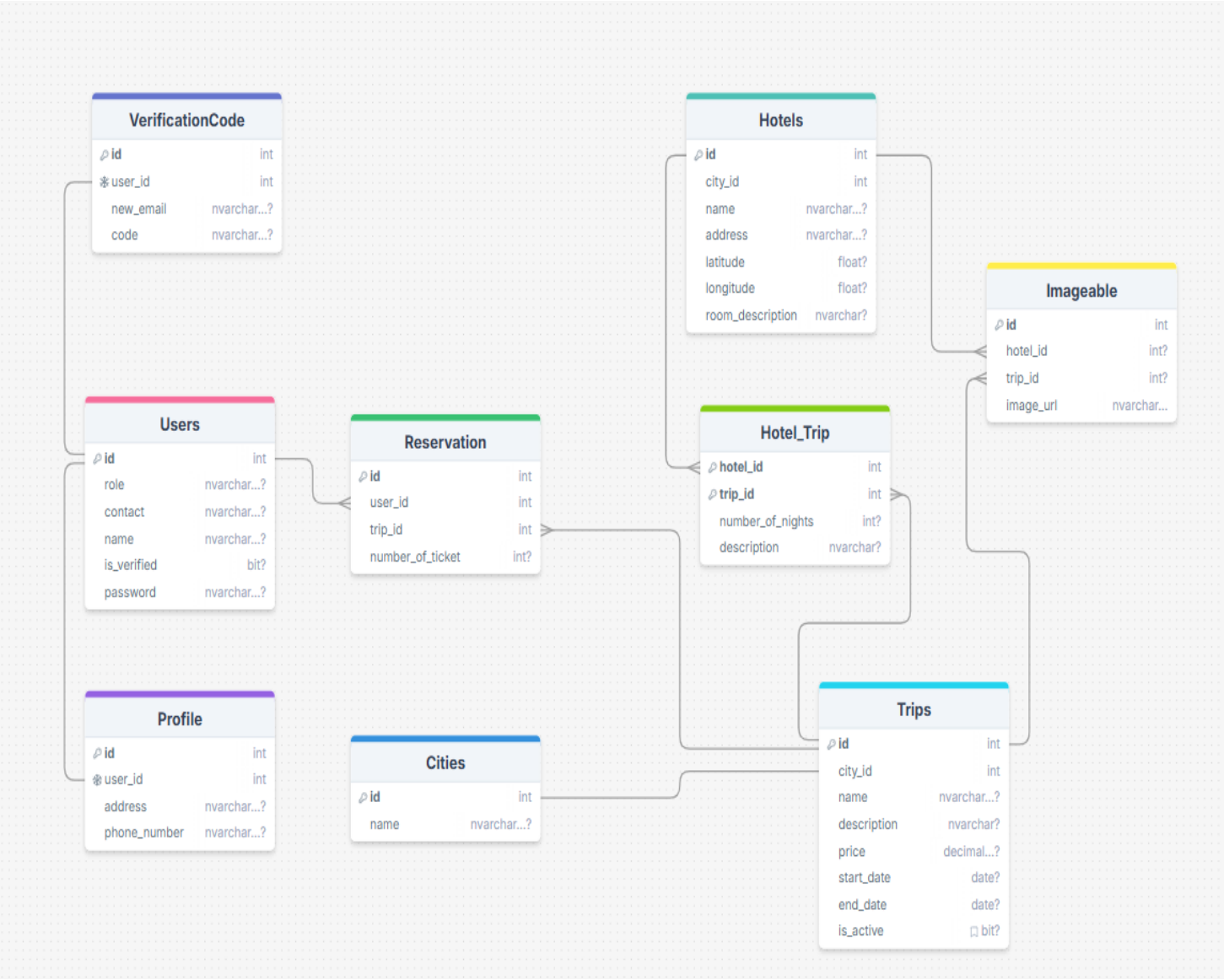



Table diagram - **Figure 60**

## 9. Initial test cases

| Test Case ID | Related Use Case | Scenario Type | Preconditions | Test Steps | Expected Result |
|---|---|---|---|---|---|
| TC-UC1-01 | UC-1 Sign In | Main | User has an account | Enter valid username/email and password → Login | User logged in successfully |
| TC-UC1-02 | UC-1 Sign In | Alternative | User has an account | Enter invalid credentials → Login | Error message displayed and redirect to login |
| TC-UC2-01 | UC-2 Logout | Main | User logged in | Click Logout | User logged out successfully |
| TC-UC5-01 | UC-5 Sign Up | Main | User not registered | Enter valid data → Register → Enter correct verification code | Account created and verified |
| TC-UC5-02 | UC-5 Sign Up | Alternative | Email or phone already used | Enter duplicated email/phone → Register | Error message shown |
| TC-UC5-03 | UC-5 Sign Up | Alternative | Verification code incorrect | Enter wrong verification code | Error message and retry verification |
| TC-UC6-01 | UC-6 Update Email | Main | User logged in | Enter new email → Enter correct verification code | Email updated successfully |
| TC-UC6-02 | UC-6 Update Email | Alternative | Email already exists | Enter duplicated email | Error message shown |
| TC-UC6-03 | UC-6 Update Email | Alternative | Verification code incorrect | Enter wrong code | Email not updated |
| TC-UC7-01 | UC-7 Update Profile | Main | User logged in | Update profile data → Save | Profile data updated |
| TC-UC7-02 | UC-7 Update Profile | Alternative | Phone number already exists | Enter duplicated phone number | Error message shown |

| Test Case ID | Related Use Case | Scenario Type | Preconditions | Test Steps | Expected Result |
|---|---|---|---|---|---|
| TC-UC8-01 | UC-8 Forget Password | Main | User has an account | Enter email → Enter correct code → Enter new password | Password updated successfully |
| TC-UC8-02 | UC-8 Forget Password | Alternative | Email does not exist | Enter invalid email | Error message shown |
| TC-UC8-03 | UC-8 Forget Password | Alternative | Verification code incorrect | Enter wrong code | Password not updated |
| TC-UC9-01 | UC-9 View Trips | Main | User on home page | Open trips list | Active trips displayed |
| TC-UC10-01 | UC-10 View Trip Info | Main | Trips available | Click on trip | Trip details displayed |
| TC-UC11-01 | UC-11 View Reservations | Main | User logged in | Open reservations page | Reservations list displayed |
| TC-UC12-01 | UC-12 Book Trip | Main | User logged in, Trip active | Enter valid number of tickets → Confirm | Reservation created with status PENDING |
| TC-UC12-02 | UC-12 Book Trip | Alternative | Tickets < 1 | Enter 0 tickets | System sets tickets to 1 automatically |
| TC-UC12-03 | UC-12 Book Trip | Alternative | Tickets > 10 | Enter more than 10 tickets | System sets tickets to 10 automatically |
| TC-UC13-01 | UC-13 Update Reservation | Main | Reservation status = PENDING | Update number of tickets → Save | Reservation updated |
| TC-UC13-02 | UC-13 Update Reservation | Alternative | Reservation status ≠ PENDING | Try to update reservation | Error message shown |
| TC-UC14-01 | UC-14 Cancel Reservation | Main | Reservation status = PENDING | Enter cancel reason → Confirm | Reservation canceled |
| TC-UC14-02 | UC-14 Cancel Reservation | Alternative | Reservation status ≠ PENDING | Try to cancel reservation | Error message shown |

| Test Case ID | Related Use Case | Scenario Type | Preconditions | Test Steps | Expected Result |
|---|---|---|---|---|---|
| TC-UC15-01 | UC-15 Manage Trips | Main | Admin logged in | Open manage trips page | Trips list displayed |
| TC-UC16-01 | UC-16 Add Trip | Main | Admin logged in | Enter valid trip data → Create | Trip created successfully |
| TC-UC16-02 | UC-16 Add Trip | Alternative | Number of nights incorrect | Enter invalid nights | Error message shown |
| TC-UC17-01 | UC-17 Update Trip | Main | Admin logged in, Trip not ended | Update trip data → Save | Trip updated successfully |
| TC-UC17-02 | UC-17 Update Trip | Alternative | Trip already ended | Try to update | Update button disabled |
| TC-UC18-01 | UC-18 Delete Trip | Main | Admin logged in, Trip not ended | Delete trip → Confirm | Trip deleted successfully |
| TC-UC18-02 | UC-18 Delete Trip | Alternative | Trip already ended | Try to delete | Delete disabled |
| TC-UC19-01 | UC-19 View Trip (Admin) | Main | Admin logged in | Click Show trip | Trip details displayed |
| TC-UC20-01 | UC-20 Manage Cities | Main | Admin logged in | Open manage cities page | Cities list displayed |
| TC-UC21-01 | UC-21 Add City | Main | Admin logged in | Enter city name → Create | City created successfully |
| TC-UC21-02 | UC-21 Add City | Alternative | City name duplicated | Enter existing city name | Error message shown |
| TC-UC22-01 | UC-22 Update City | Main | Admin logged in | Update city name → Save | City updated |
| TC-UC22-02 | UC-22 Update City | Alternative | City name duplicated | Enter existing name | Error message shown |

| Test Case ID | Related Use Case | Scenario Type | Preconditions | Test Steps | Expected Result |
|---|---|---|---|---|---|
| TC-UC23-01 | UC-23 Delete City | Main | Admin logged in, City not used | Delete city → Confirm | City deleted (soft delete) |
| TC-UC23-02 | UC-23 Delete City | Alternative | City used in trips | Try to delete city | Error message shown |
| TC-UC24-01 | UC-24 Manage Hotels | Main | Admin logged in | Open manage hotels page | Hotels list displayed |
| TC-UC25-01 | UC-25 Add Hotel | Main | Admin logged in | Enter hotel data → Create | Hotel created successfully |
| TC-UC25-02 | UC-25 Add Hotel | Alternative | Hotel name duplicated | Enter existing hotel name | Error message shown |

# Chapter 5
# System design

## 5.1 Introduction

This chapter presents the system design of the Hajj and Umrah Management Website.
The system is implemented using a **React-based frontend** and a **backend API following the Model–View–Controller (MVC) architectural pattern**.

Unlike traditional MVC systems where Views are rendered on the server side, this system separates the frontend and backend. The frontend is implemented using React, while the backend exposes RESTful APIs that handle business logic and data processing.

This design approach improves scalability, flexibility, and maintainability, similar to the architecture used in the Marketing System project.

---

## 5.2 Overall System Architecture

The system follows a **Client–Server Architecture** with clear separation between frontend and backend.

The architecture consists of the following main layers:

- Frontend Layer (React)
- Backend Layer (MVC-based API)
- Data Layer (Database)

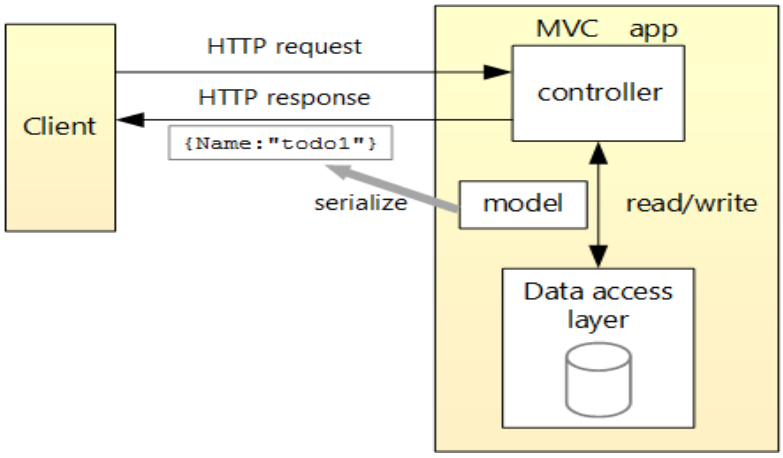



**Figure (61): Overall System Architecture (React + MVC API)**
*(Diagram showing React Client → API Controllers → Models → Database)*

## 5.3 Frontend Architecture (React Layer)

The frontend of the system is implemented using **React**, which is responsible only for the user interface and user interaction.

### Frontend responsibilities:

- Display pages, forms, tables, and dialogs
- Collect user input
- Send HTTP requests to backend APIs
- Display responses received from the backend
- Handle routing and state management
- Integrate chatbot interface

The frontend does not contain business logic or direct database access.

---

## 5.4 Backend Architecture (MVC-Based API)

The backend is designed using the **MVC architectural pattern**, but without server-side views.
Instead, Controllers return **JSON responses** to the React frontend.

### Backend layers include:

- Controller Layer
- Model Layer
- Service (Business Logic) Layer *(optional but recommended)*
- Data Access Layer

### 5.4.1 Controller Layer

Controllers handle HTTP requests sent from the React frontend.

*Controller responsibilities:*

- Receive API requests
- Validate request data
- Call appropriate business logic
- Return JSON responses

*Main Controllers:*

- AuthController
- UserController
- TripController
- ReservationController
- CityController
- HotelController
- ChatbotController
- AdminController

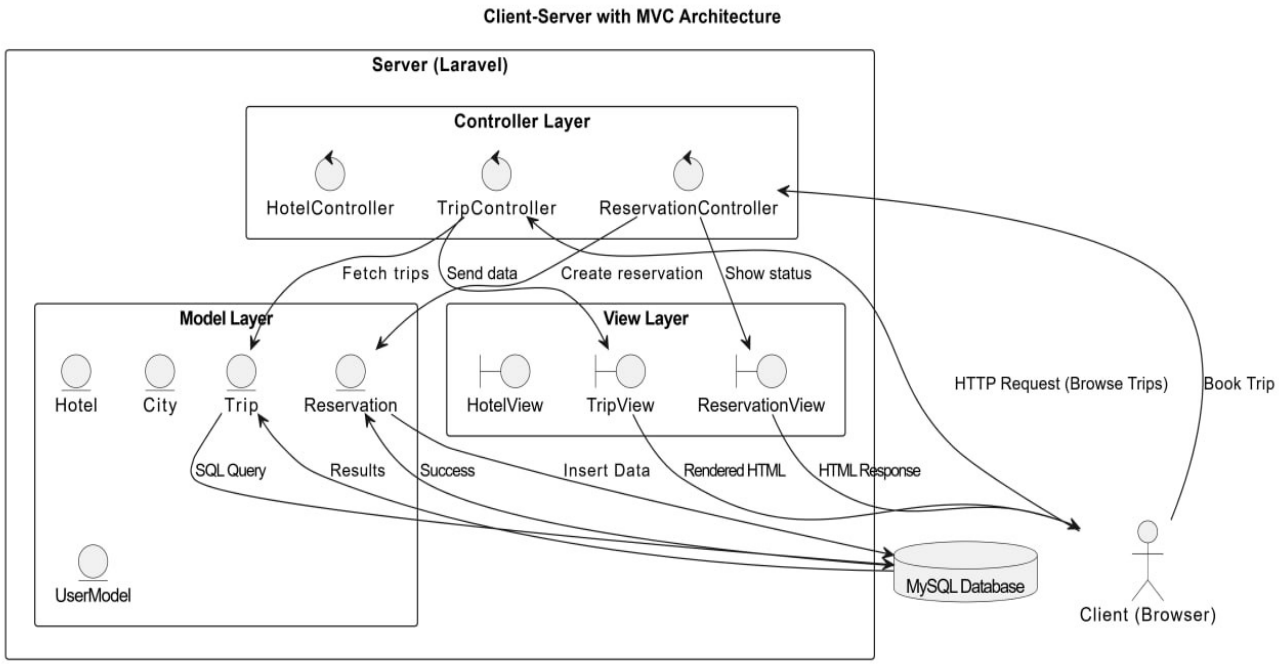



**Figure (62): API Request Flow (React → Controller → Model)**

### 5.4.2 Model Layer

The Model layer represents the system data and business rules.

*Model responsibilities:*

- Represent database entities
- Apply business rules
- Perform CRUD operations
- Support soft delete logic

*Main Models:*

- User
- Trip
- Reservation
- City
- Hotel
- TripNight
- TripHotel
- ChatbotDataset
- Notification

Models interact with the database through ORM or direct queries.

### 5.4.3 Service Layer (Business Logic)

The Service layer contains complex business logic and rules.

*Service responsibilities:*

- Reservation validation
- Trip status checks
- Notification triggering
- Chatbot keyword matching
- Data consistency checks

This layer improves separation of concerns and keeps controllers lightweight.

## 5.5 API Communication Design

Communication between the frontend and backend is performed using **RESTful APIs**.

### API characteristics:

- JSON request and response format
- HTTP methods (GET, POST, PUT, DELETE)
- Stateless communication
- Role-based access control

---

## 5.6 UML Diagrams and System Design

UML diagrams are used to represent system behavior and interaction:

- Use Case Diagrams: Describe system functionality
- Sequence Diagrams: Show interaction between React, API Controllers, and Models
- Activity Diagrams: Describe system workflows

These diagrams support the MVC-based API design and frontend-backend interaction.

---

## 5.7 Security Design

Security is enforced at the backend level:

- Authentication using tokens
- Role-based authorization
- Input validation
- Secure password storage
- Protected admin APIs

---

## 5.8 Advantages of React + MVC API Architecture

This architecture provides:

- Clear separation between frontend and backend
- Scalability
- Better performance
- Easier frontend development
- Independent deployment of frontend and backend

---

## 5.9 Summary

This chapter described the system design of the Hajj and Umrah Management Website using a React frontend and an MVC-based backend API.
The design ensures modularity, scalability, and alignment with modern web application architectures.

The next chapter will discuss system implementation and testing.

# Chapter 6
# Practical Implementation

## 6.1 Introduction

This chapter describes the implementation and testing phases of the **Hajj and Umrah Trip Management System**.
The system was implemented using a **React-based frontend** and a **Laravel backend API**, following an incremental development approach.

The implementation focuses on delivering a scalable, maintainable, and user-friendly web application that meets the functional and non-functional requirements defined in previous chapters.

---

## 6.2 Development Environment

The system was developed using the following environment and tools:

- **Operating System:** Windows
- **Code Editor:** Visual Studio Code (VS Code)
- **Frontend Build Tool:** Vite
- **Backend Framework:** Laravel (PHP)
- **Local Server:** XAMPP
- **Database Management:** phpMyAdmin
- **Version Control:** GitHub

These tools supported efficient development, testing, and deployment of the system

---

## 6.3 Technologies and Tools Used

### 6.3.1 Frontend Technologies

The frontend of the system was implemented using modern JavaScript technologies:

- **React:** Used to build reusable UI components and manage application state
- **Tailwind CSS:** Used for responsive and utility-first UI styling
- **Vite:** Used as a fast build and development tool

This combination provides high performance, modularity, and a modern user experience

---

### 6.3.2 Backend Technologies

- **Programming Language:** PHP
- **Framework:** Laravel

Laravel was used to implement the backend as a RESTful API following the MVC architectural pattern.
It provides routing, authentication, ORM (Eloquent), and secure session handling.

---

### 6.3.3 Database Technology

- **Database Management System:** MySQL

The database stores structured data related to users, trips, hotels, cities, and reservations.
Relational design and indexing were applied to ensure data integrity and efficient querying

---

### 6.3.4 External APIs

- **Google Maps API**

Google Maps API was integrated to display hotel locations and trip destinations within the system interface

---

## 6.4 Backend Implementation

The backend was implemented using Laravel following the MVC architectural pattern.

### 6.4.1 Controller Layer

Controllers receive HTTP requests from the React frontend and return JSON responses.

Main controllers include:

- AuthController
- UserController
- TripController
- ReservationController
- CityController
- HotelController
- ChatbotController

---

### 6.4.2 Model Layer

Models represent database entities and encapsulate business rules.

Main models include:

- User
- Trip
- Reservation
- City
- Hotel

Laravel Eloquent ORM was used to map models to database tables.

## 6.5 Frontend Implementation (React)

The frontend was implemented as a **single-page application (SPA)** using React.

Key features include:

- Component-based architecture
- API communication using HTTP requests
- Dynamic rendering of trips, reservations, and admin dashboards
- Integration of chatbot interface
- Responsive design using Tailwind CSS

---

## 6.6 Chatbot Implementation

The chatbot feature was implemented as a React component supported by backend logic.

### Chatbot workflow:

1. The user submits a question through the chatbot interface.
2. The system extracts keywords from the input.
3. Keywords are compared with a predefined dataset.
4. The matched response is returned to the frontend.
5. The response is displayed to the user.

---

## 6.7 System Testing

System testing was conducted to verify correct functionality.

### 6.7.1 Testing Types

- Manual Testing
- Functional Testing

---

### 6.7.2 Test Case Execution

Test cases were derived directly from system use cases and validated:

- Authentication and authorization
- Trip browsing and booking
- Reservation management
- Admin and Super Admin operations
- City and hotel management
- Chatbot responses

All test cases are documented in the test case tables presented earlier in this chapter.

---

## 6.8 Testing Results

Testing results confirmed that:

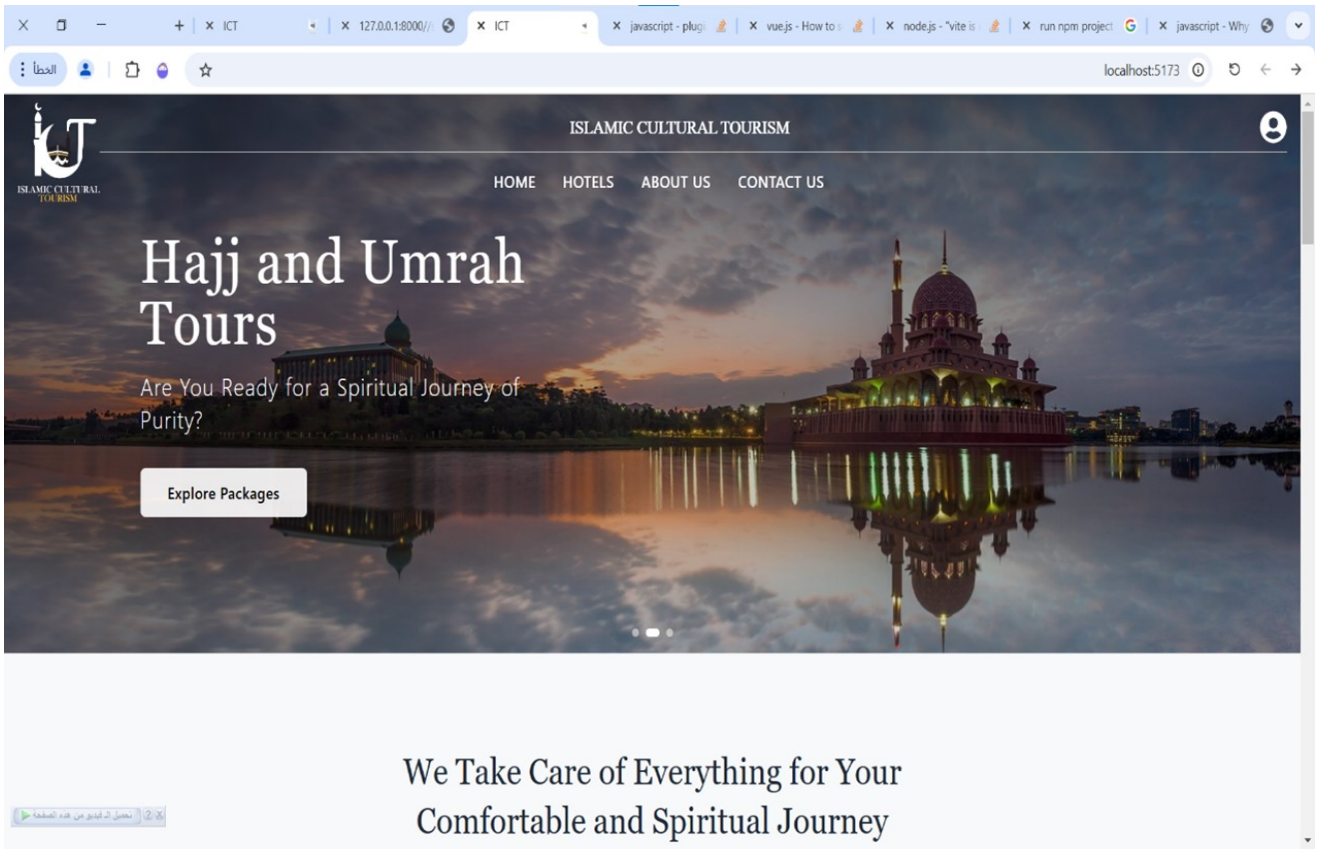
- All core system functionalities operate correctly
- Input validation is enforced
- User roles and permissions are applied correctly
- The system meets its defined requirements

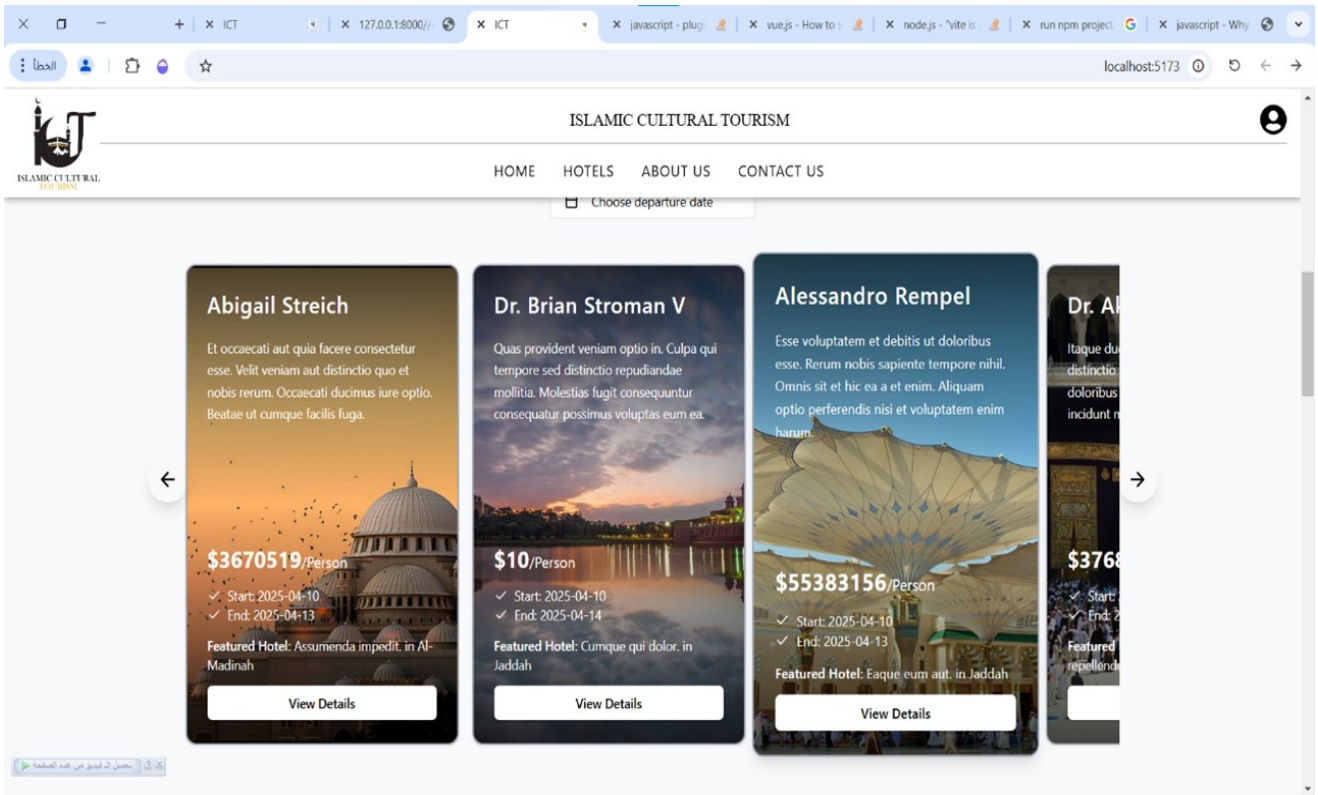
---

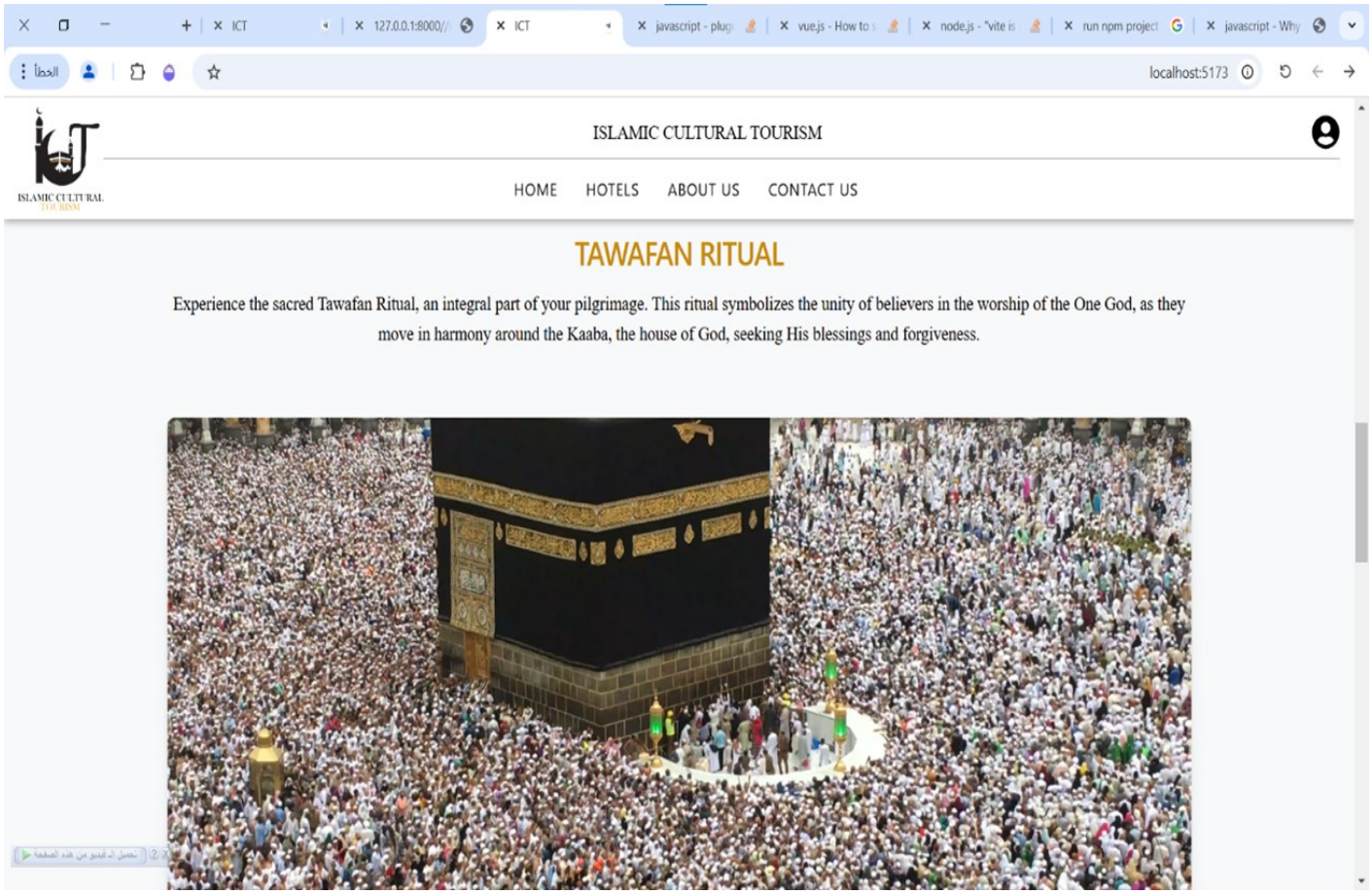
## 6.9 Summary

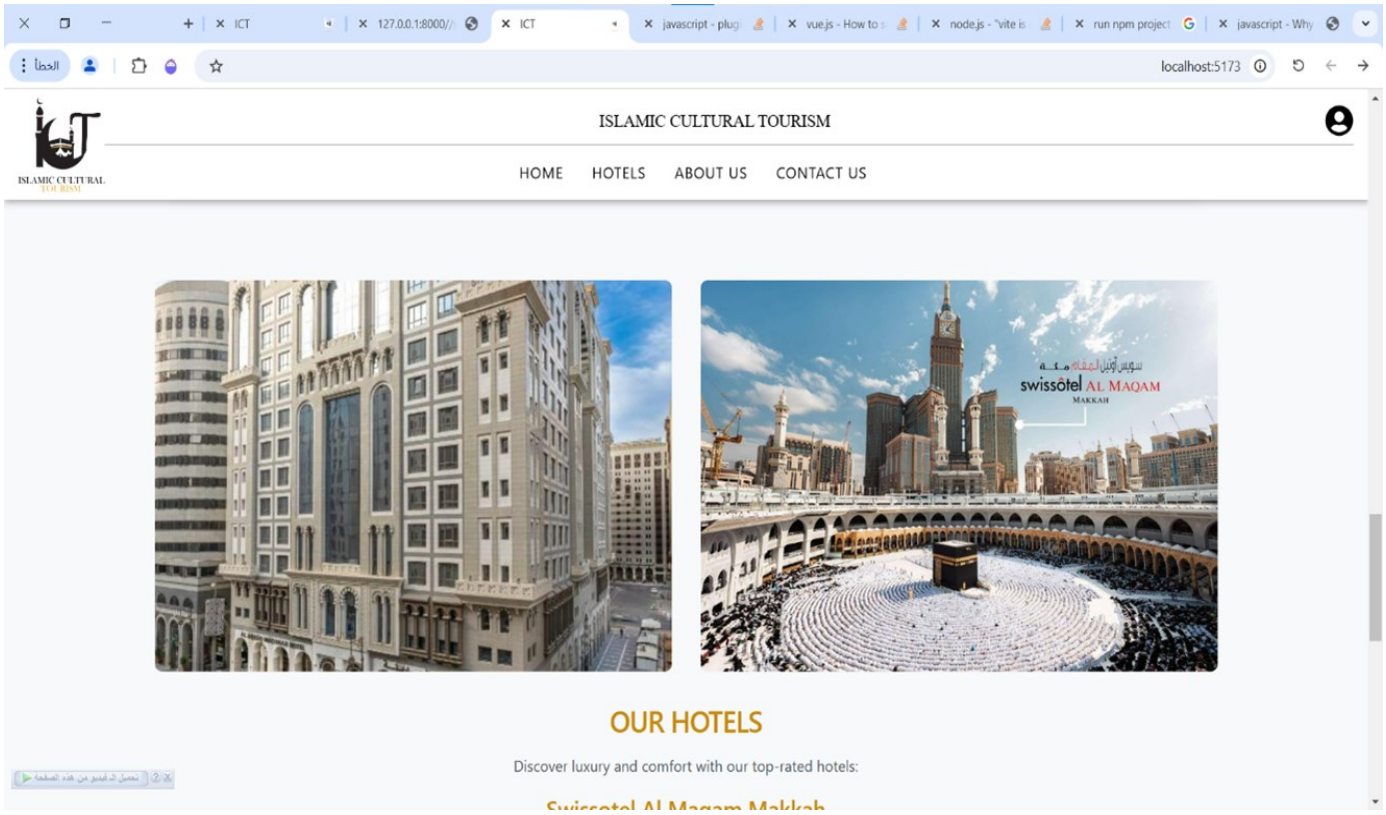
This chapter presented the implementation and testing details of the system using **React, Tailwind CSS, Vite, Laravel, and MySQL**.
The incremental development approach ensured early validation of features and reduced implementation risks.

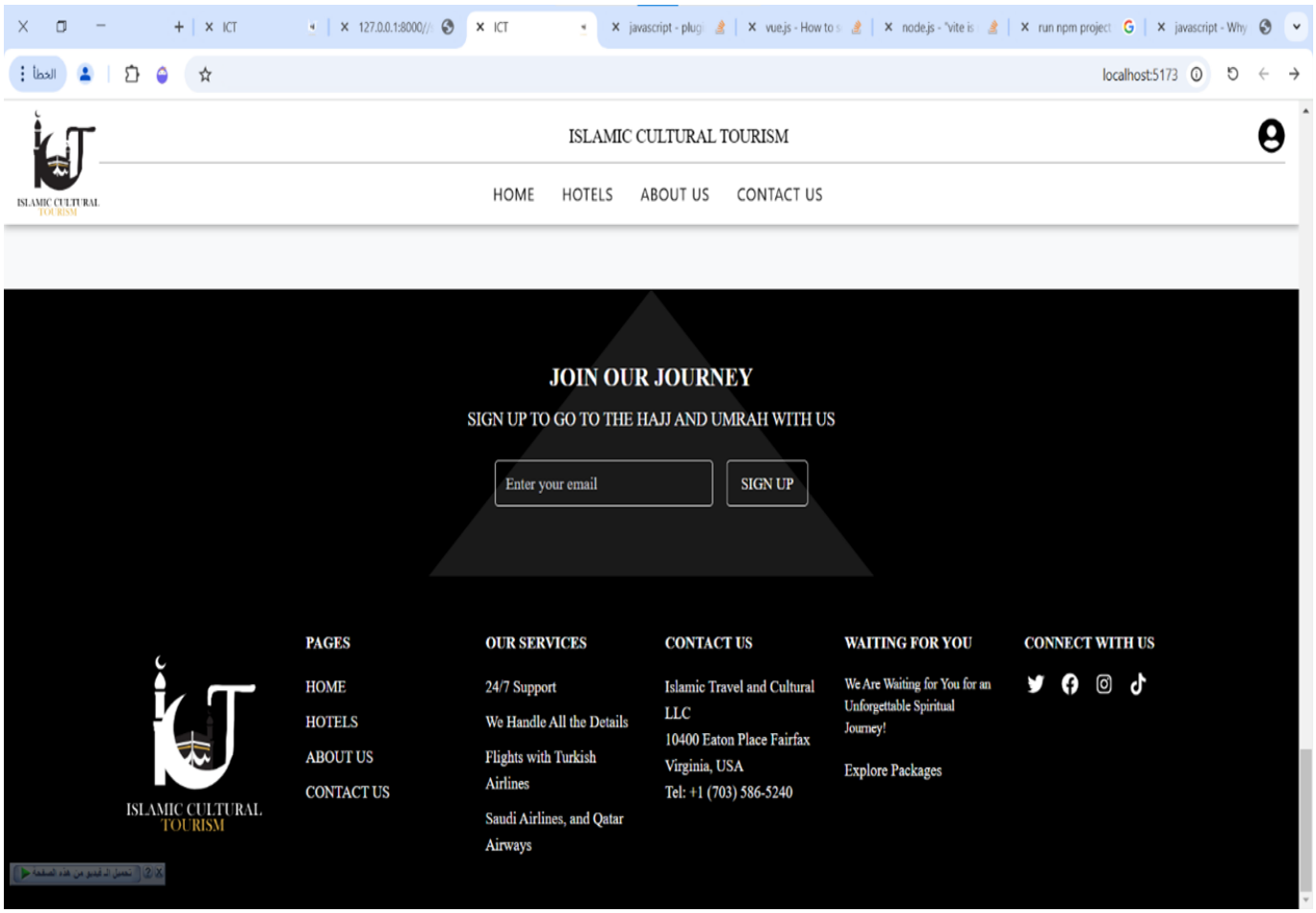
## 7.System Interfaces

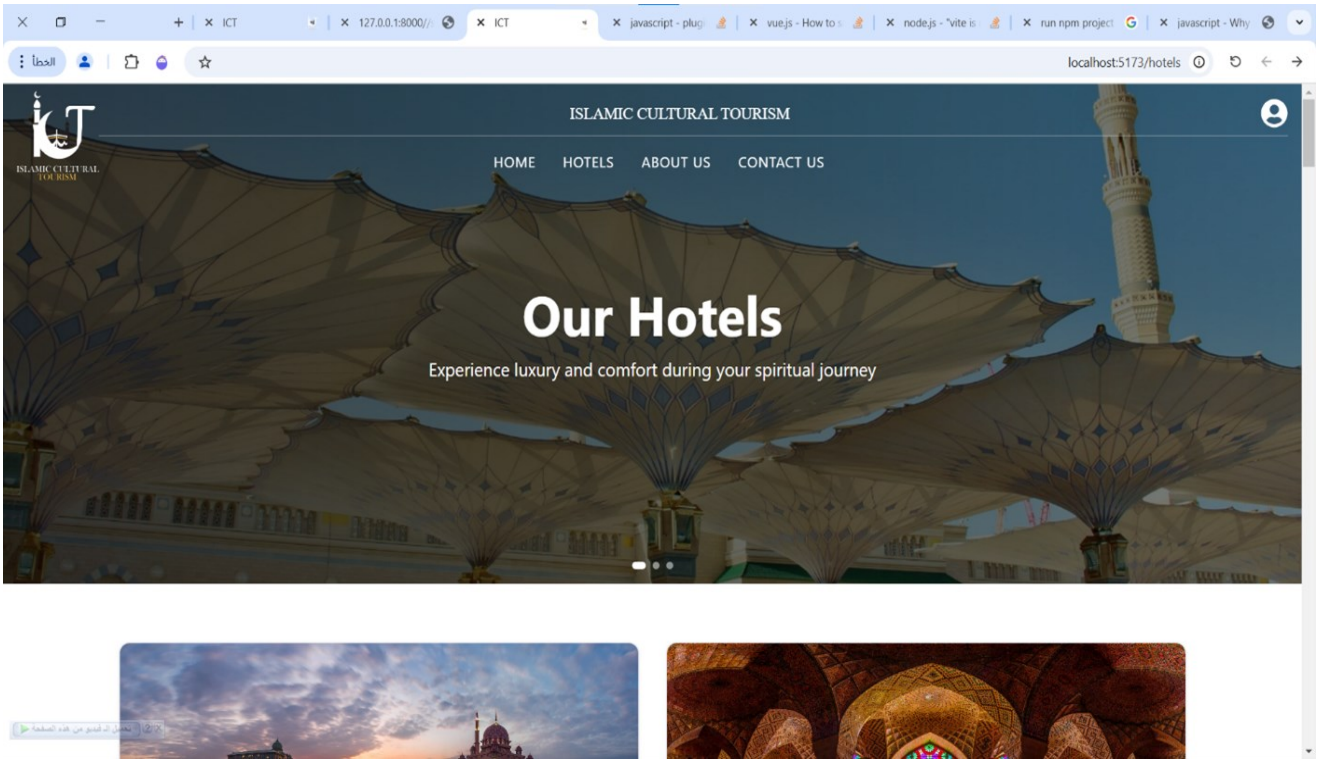


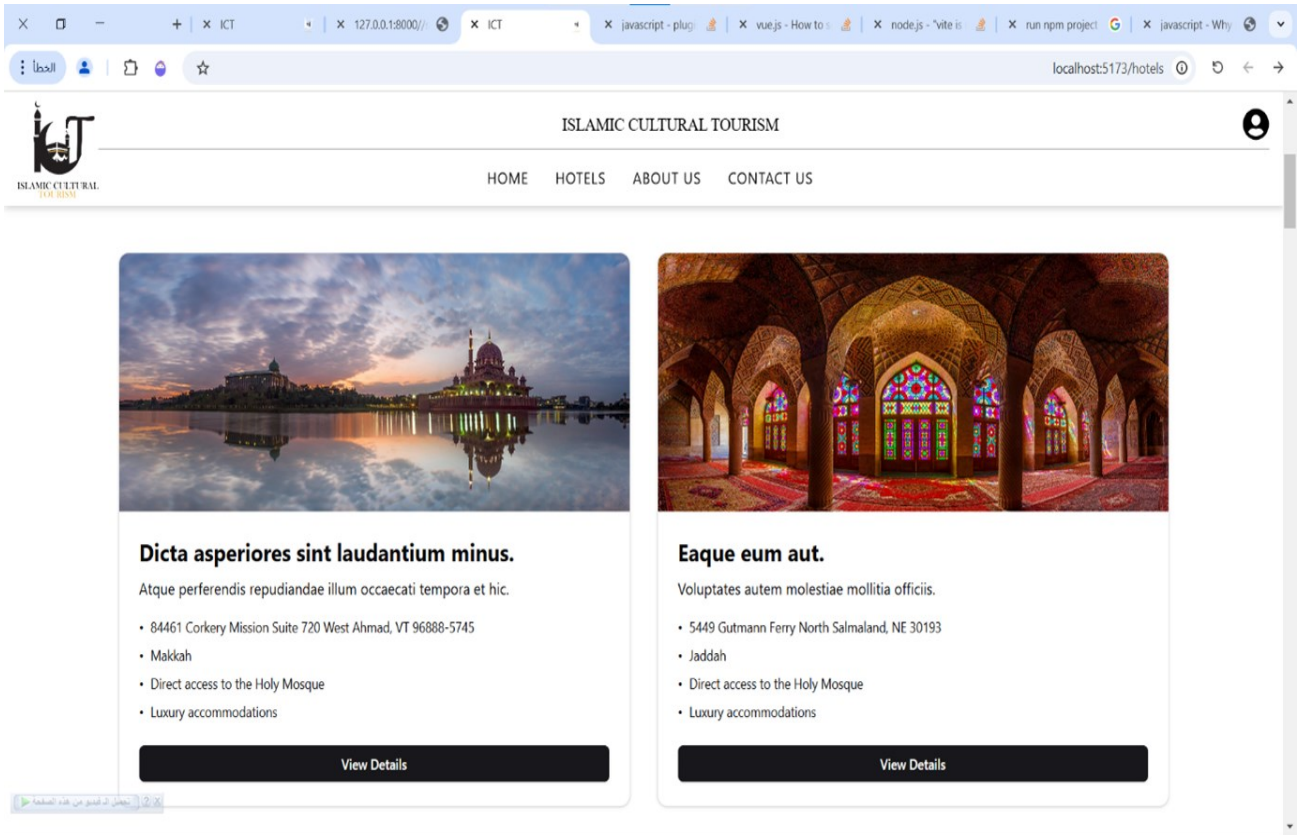



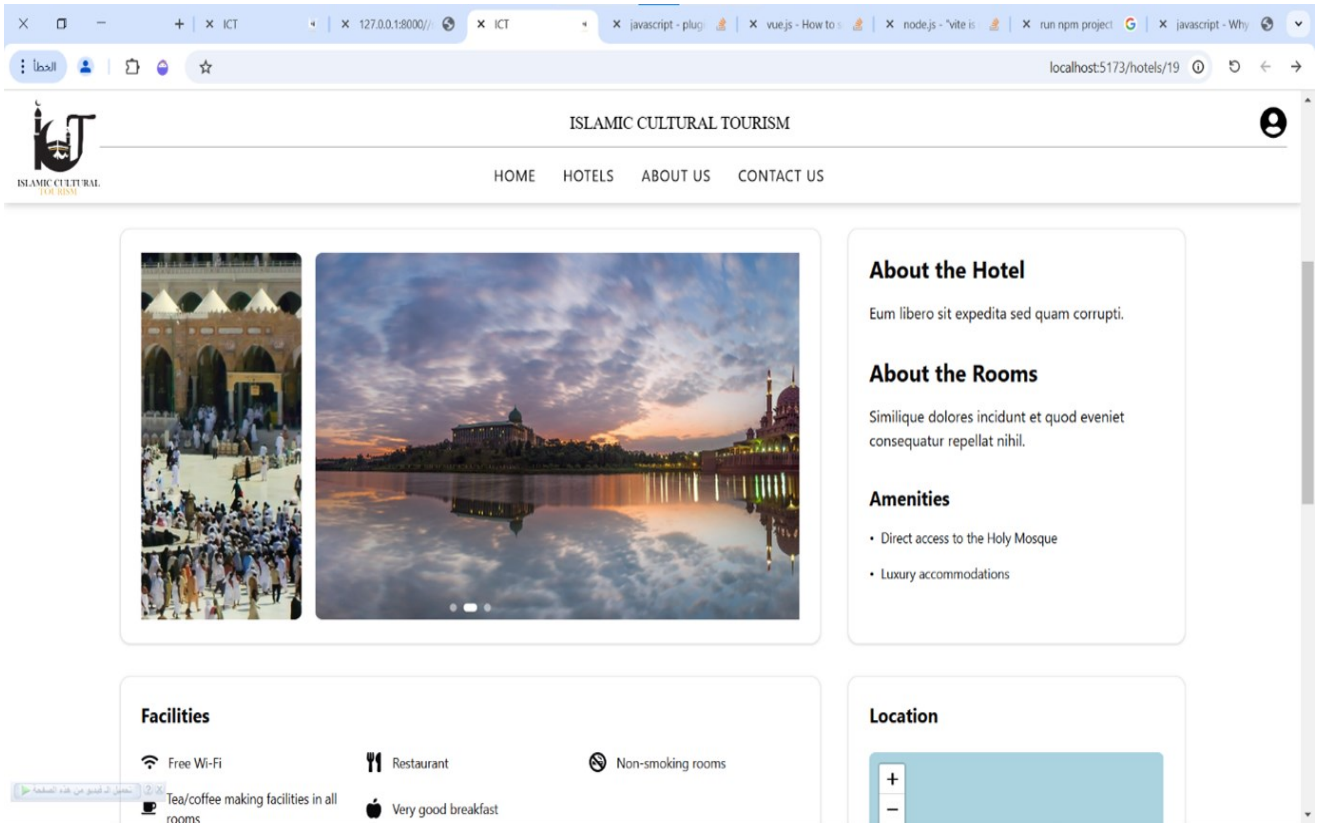

ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
TAWAFAN RITUAL
Experience the sacred Tawafan Ritual, an integral part of your pilgrimage. This ritual symbolizes the unity of believers in the worship of the One God, as they move in harmony around the Kaaba, the house of God, seeking His blessings and forgiveness.

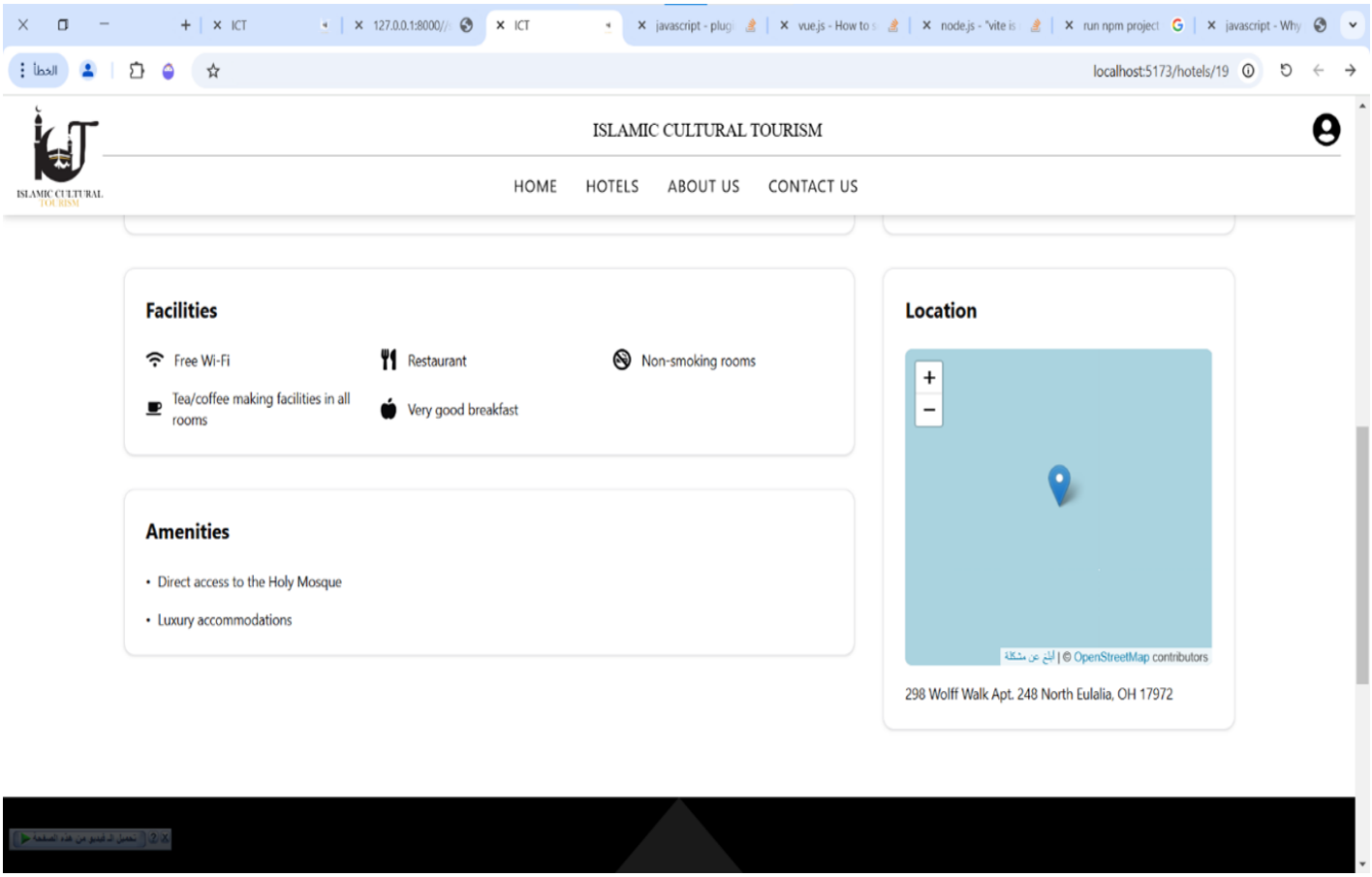

ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
swissôtel AL MAQAM
MAKKAH
OUR HOTELS
Discover luxury and comfort with our top-rated hotels:

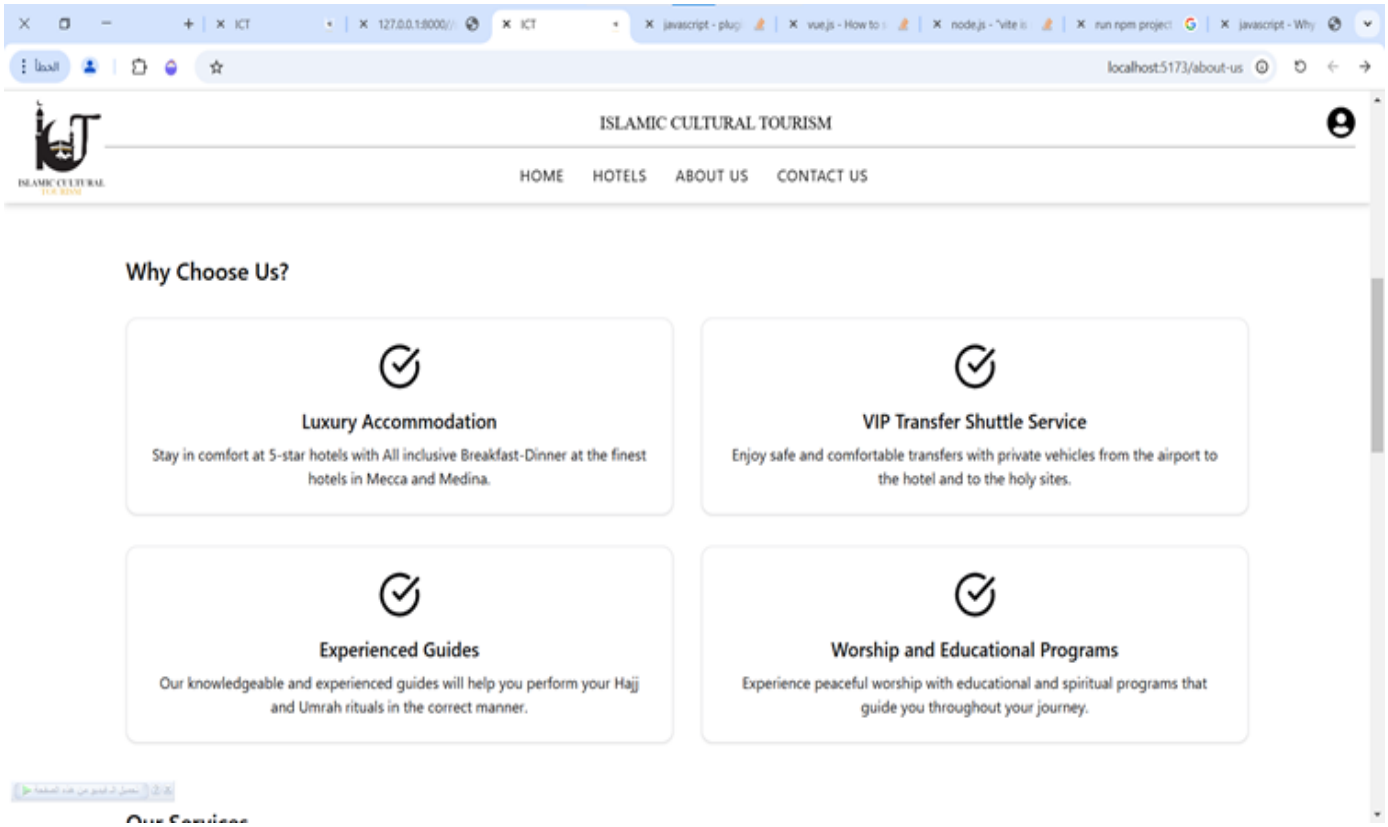

ISLAMIC CULTURAL TOURISM
HOME HOTELS ABOUT US CONTACT US
JOIN OUR JOURNEY
SIGN UP TO GO TO THE HAJJ AND UMRAH WITH US
Enter your email
SIGN UP
ISLAMIC CULTURAL TOURISM
PAGES
HOME
HOTELS
ABOUT US
CONTACT US
OUR SERVICES
24/7 Support
We Handle All the Details
Flights with Turkish Airlines
Saudi Airlines, and Qatar Airways
CONTACT US
Islamic Travel and Cultural LLC
10400 Eaton Place Fairfax
Virginia, USA
Tel: +1 (703) 586-5240
WAITING FOR YOU
We Are Waiting for You for an Unforgettable Spiritual Journey!
Explore Packages
CONNECT WITH US

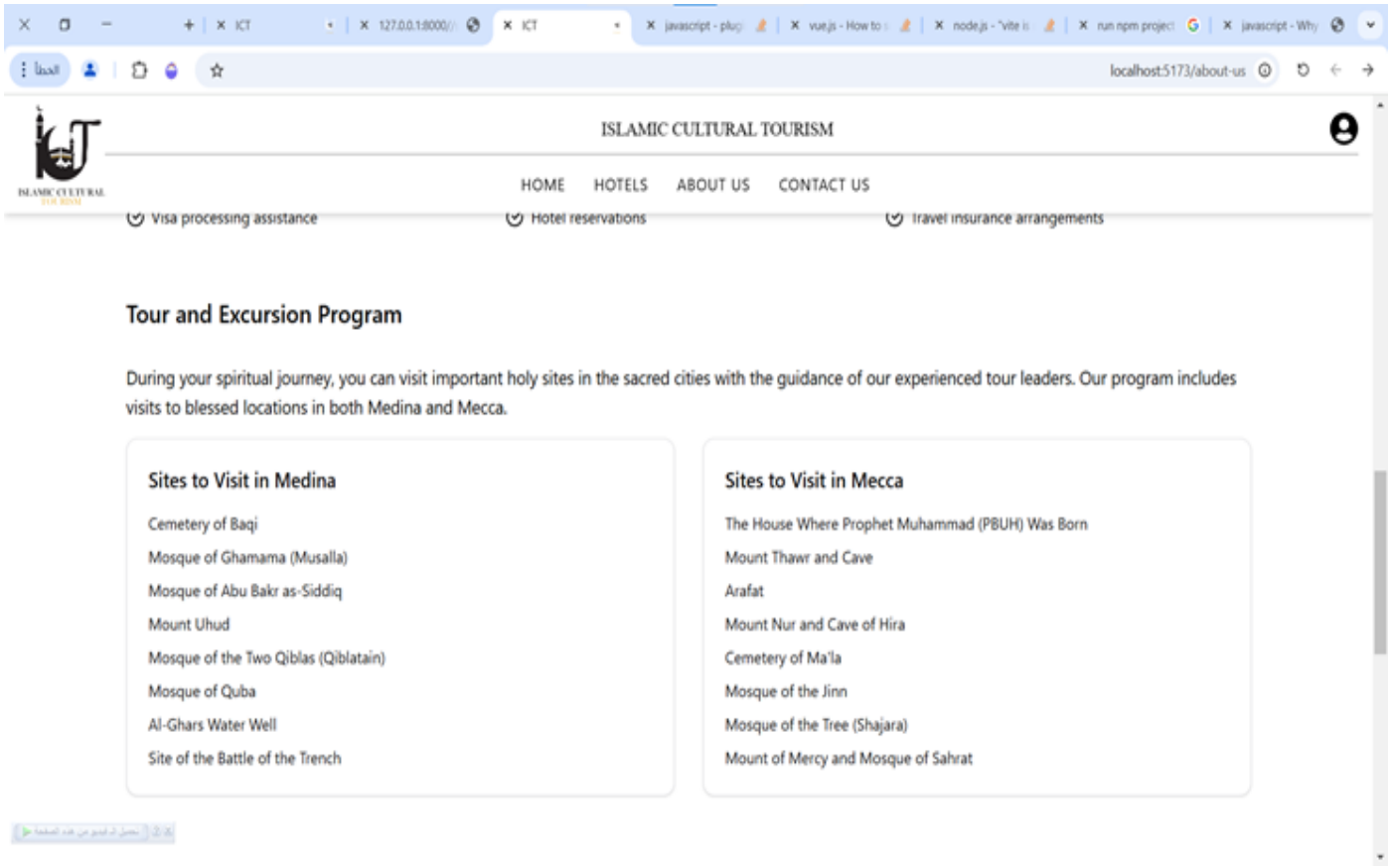

ISLAMIC CULTURAL TOURISM
HOME HOTELS ABOUT US CONTACT US
Our Hotels
Experience luxury and comfort during your spiritual journey

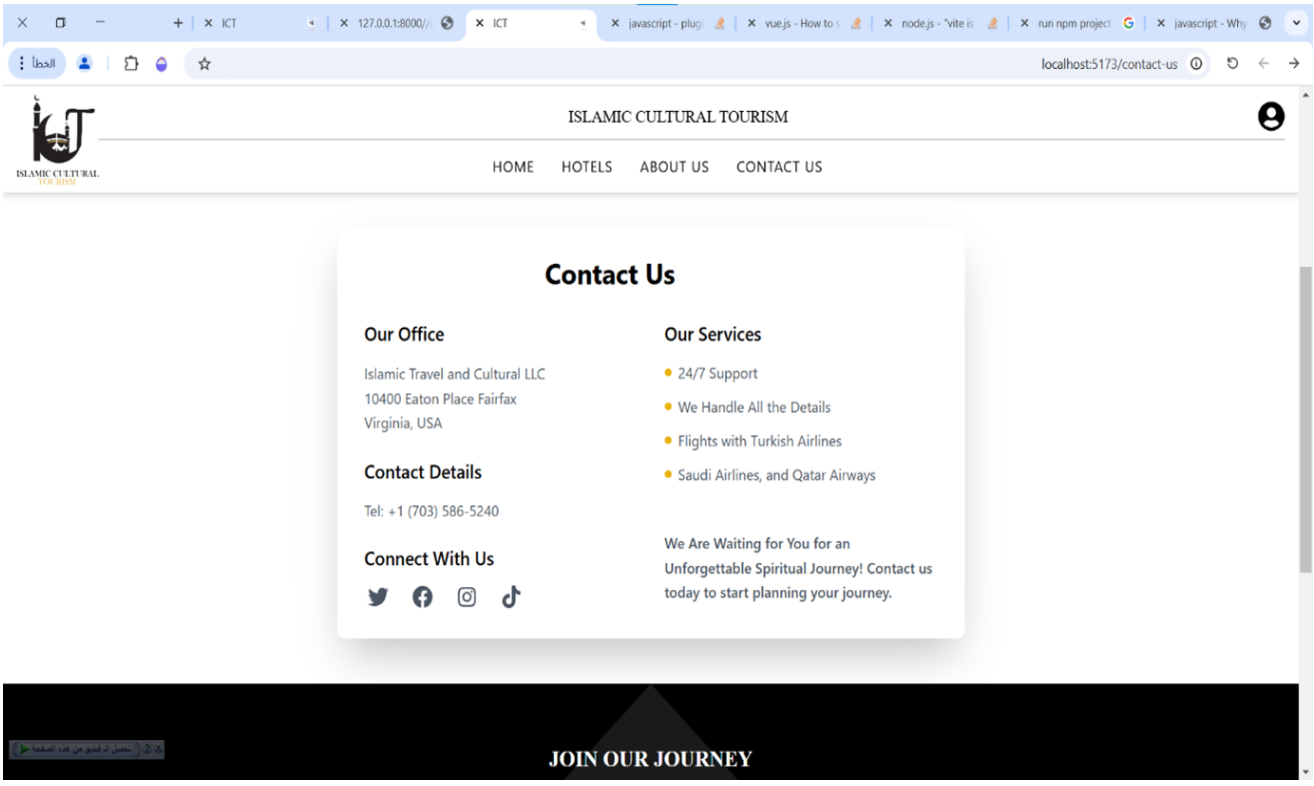

ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
Dicta asperiores sint laudantium minus.
Atque perferendis repudiandae illum occaecati tempora et hic.
• 84461 Corkery Mission Suite 720 West Ahmad, VT 96888-5745
• Makkah
• Direct access to the Holy Mosque
• Luxury accommodations
View Details
Eaque eum aut.
Voluptates autem molestiae mollitia officiis.
• 5449 Gutmann Ferry North Salmaland, NE 30193
• Jaddah
• Direct access to the Holy Mosque
• Luxury accommodations
View Details

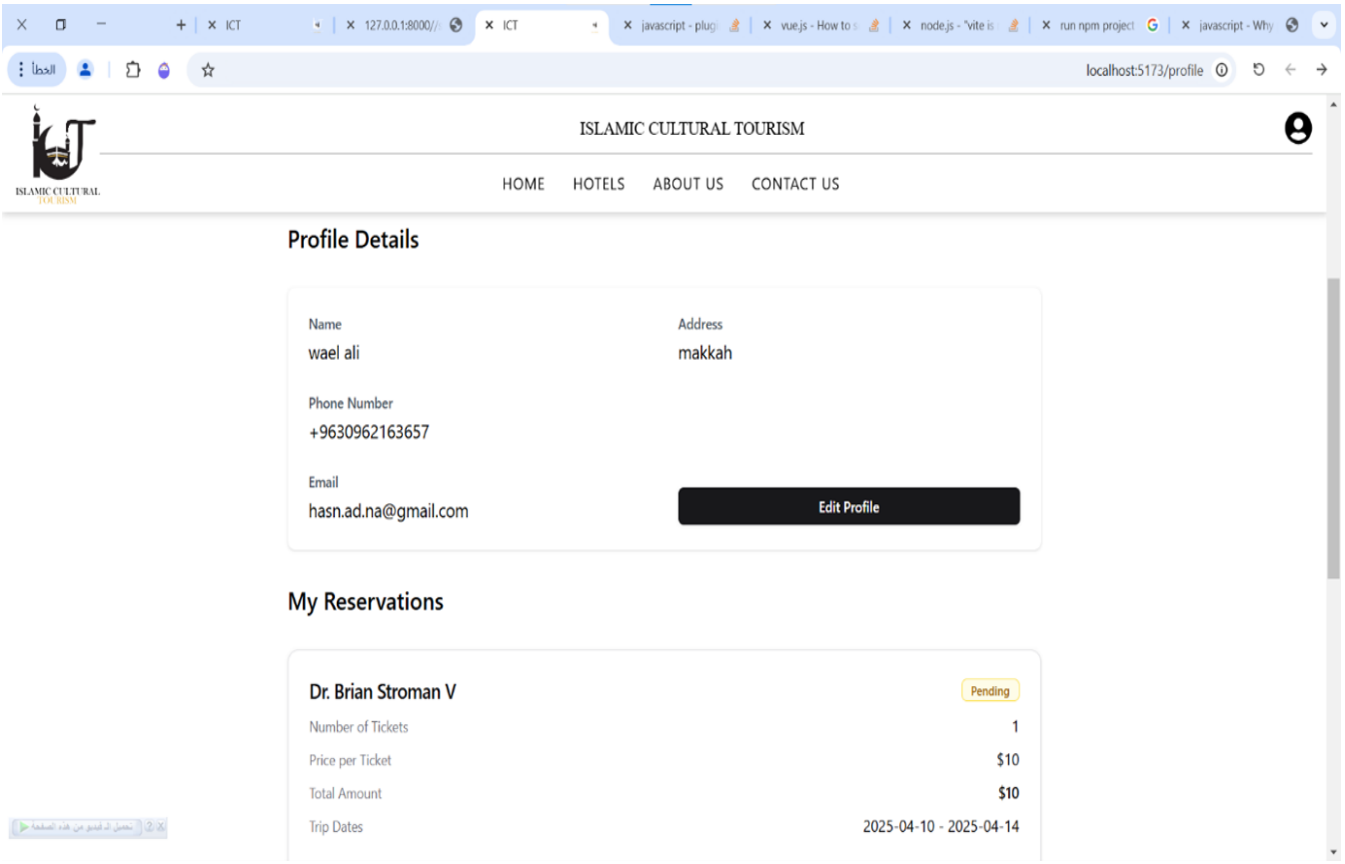

ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
About the Hotel
Eum libero sit expedita sed quam corrupti.
About the Rooms
Similique dolores incidunt et quod eveniet consequatur repellat nihil.
Amenities
• Direct access to the Holy Mosque
• Luxury accommodations
Facilities
Free Wi-Fi
Restaurant
Non-smoking rooms
Tea/coffee making facilities in all rooms
Very good breakfast
Location

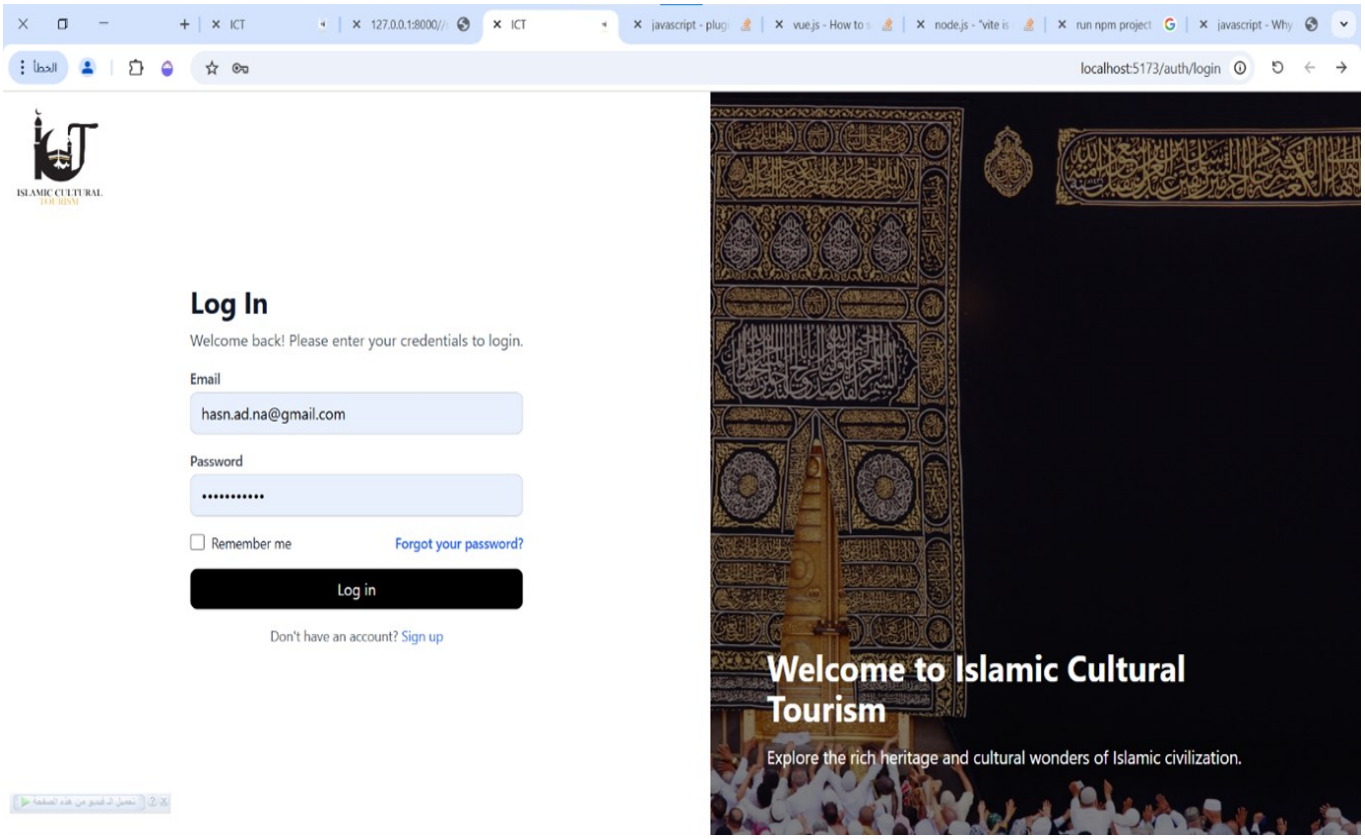

ISLAMIC CULTURAL TOURISM
HOME HOTELS ABOUT US CONTACT US
Facilities
Free Wi-Fi
Restaurant
Non-smoking rooms
Tea/coffee making facilities in all rooms
Very good breakfast
Amenities
• Direct access to the Holy Mosque
• Luxury accommodations
Location
298 Wolff Walk Apt. 248 North Eulalia, OH 17972

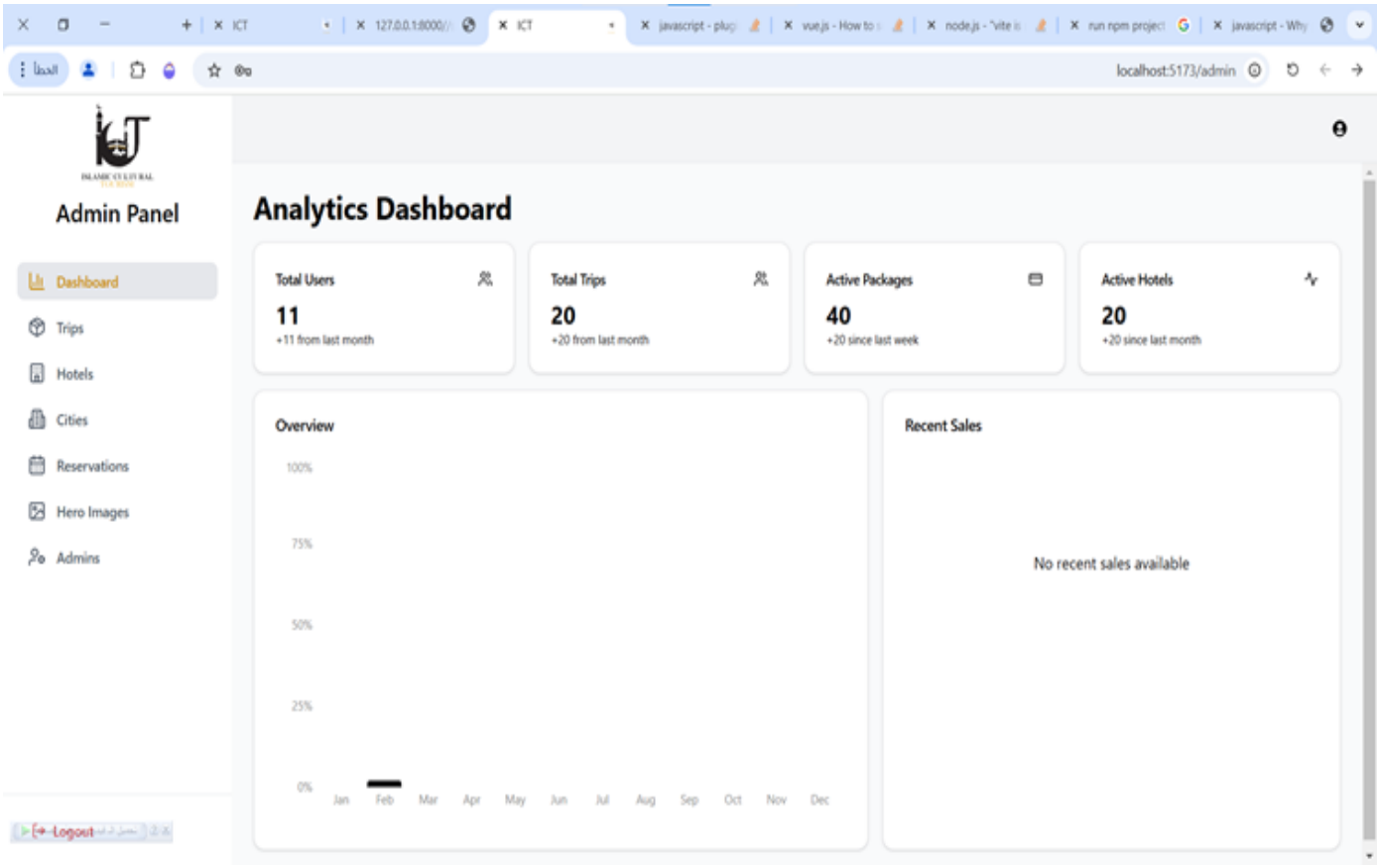

ISLAMIC CULTURAL TOURISM
HOME HOTELS ABOUT US CONTACT US
Why Choose Us?
Luxury Accommodation
Stay in comfort at 5-star hotels with All inclusive Breakfast-Dinner at the finest hotels in Mecca and Medina.
VIP Transfer Shuttle Service
Enjoy safe and comfortable transfers with private vehicles from the airport to the hotel and to the holy sites.
Experienced Guides
Our knowledgeable and experienced guides will help you perform your Hajj and Umrah rituals in the correct manner.
Worship and Educational Programs
Experience peaceful worship with educational and spiritual programs that guide you throughout your journey.

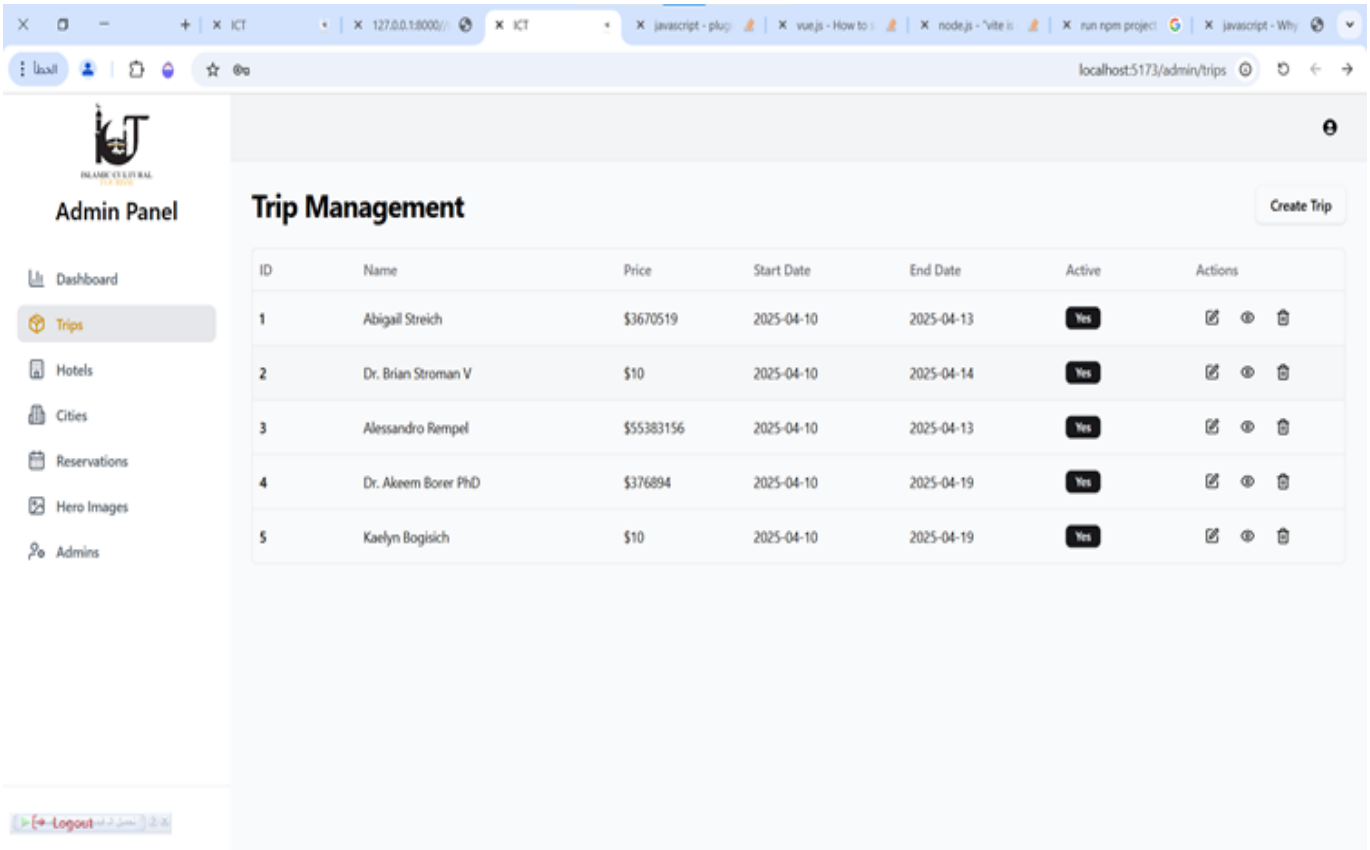

localhost:5173/about-us
ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
Visa processing assistance
Hotel reservations
Travel insurance arrangements
Tour and Excursion Program
During your spiritual journey, you can visit important holy sites in the sacred cities with the guidance of our experienced tour leaders. Our program includes visits to blessed locations in both Medina and Mecca.
Sites to Visit in Medina
Cemetery of Baqi
Mosque of Ghamama (Musalla)
Mosque of Abu Bakr as-Siddiq
Mount Uhud
Mosque of the Two Qiblas (Qiblatain)
Mosque of Quba
Al-Ghars Water Well
Site of the Battle of the Trench
Sites to Visit in Mecca
The House Where Prophet Muhammad (PBUH) Was Born
Mount Thawr and Cave
Arafat
Mount Nur and Cave of Hira
Cemetery of Ma'la
Mosque of the Jinn
Mosque of the Tree (Shajara)
Mount of Mercy and Mosque of Sahrat

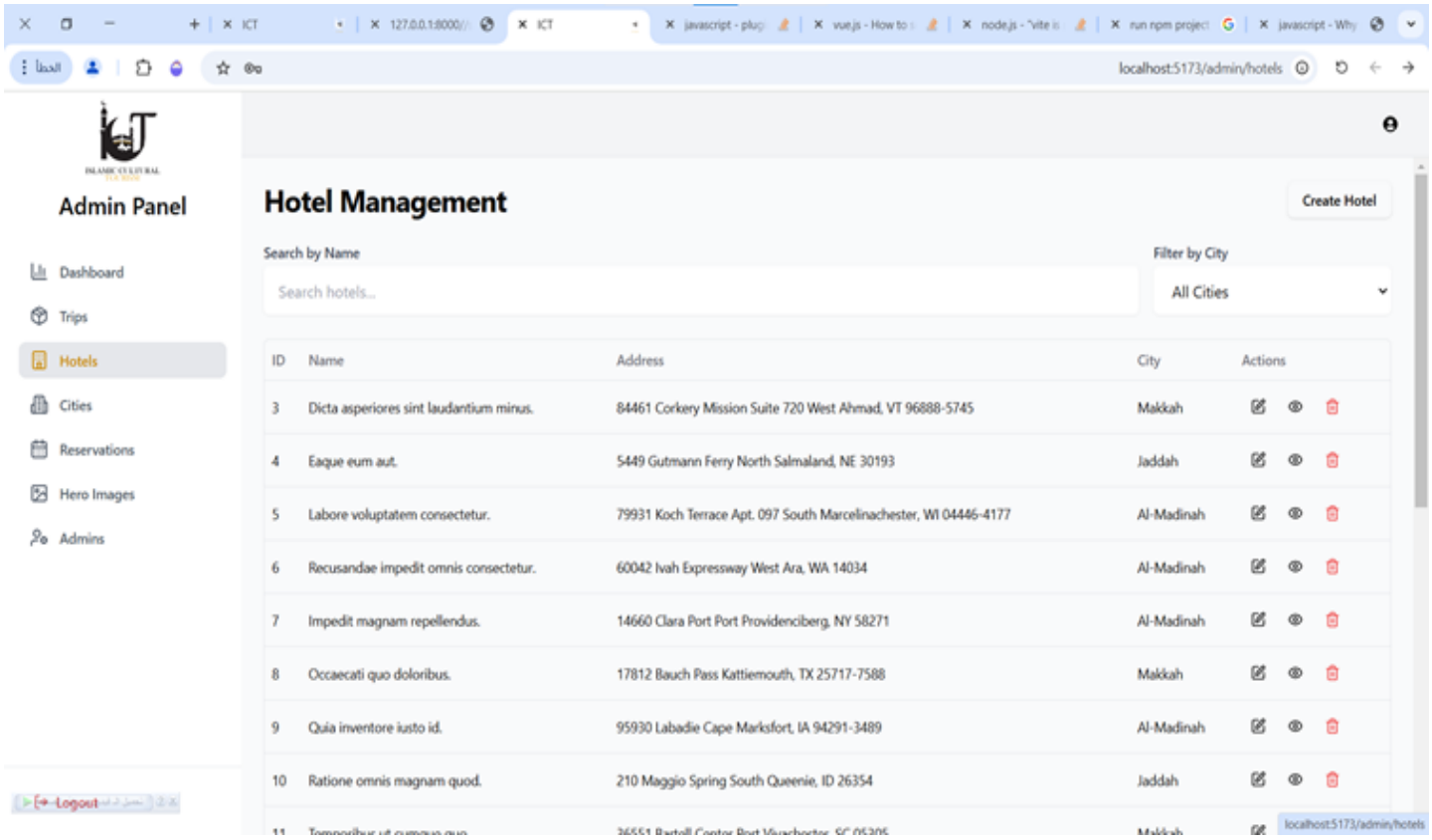

localhost:5173/contact-us
ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
Contact Us
Our Office
Islamic Travel and Cultural LLC
10400 Eaton Place Fairfax
Virginia, USA
Contact Details
Tel: +1 (703) 586-5240
Connect With Us
Our Services
24/7 Support
We Handle All the Details
Flights with Turkish Airlines
Saudi Airlines, and Qatar Airways
We Are Waiting for You for an Unforgettable Spiritual Journey! Contact us today to start planning your journey.
JOIN OUR JOURNEY

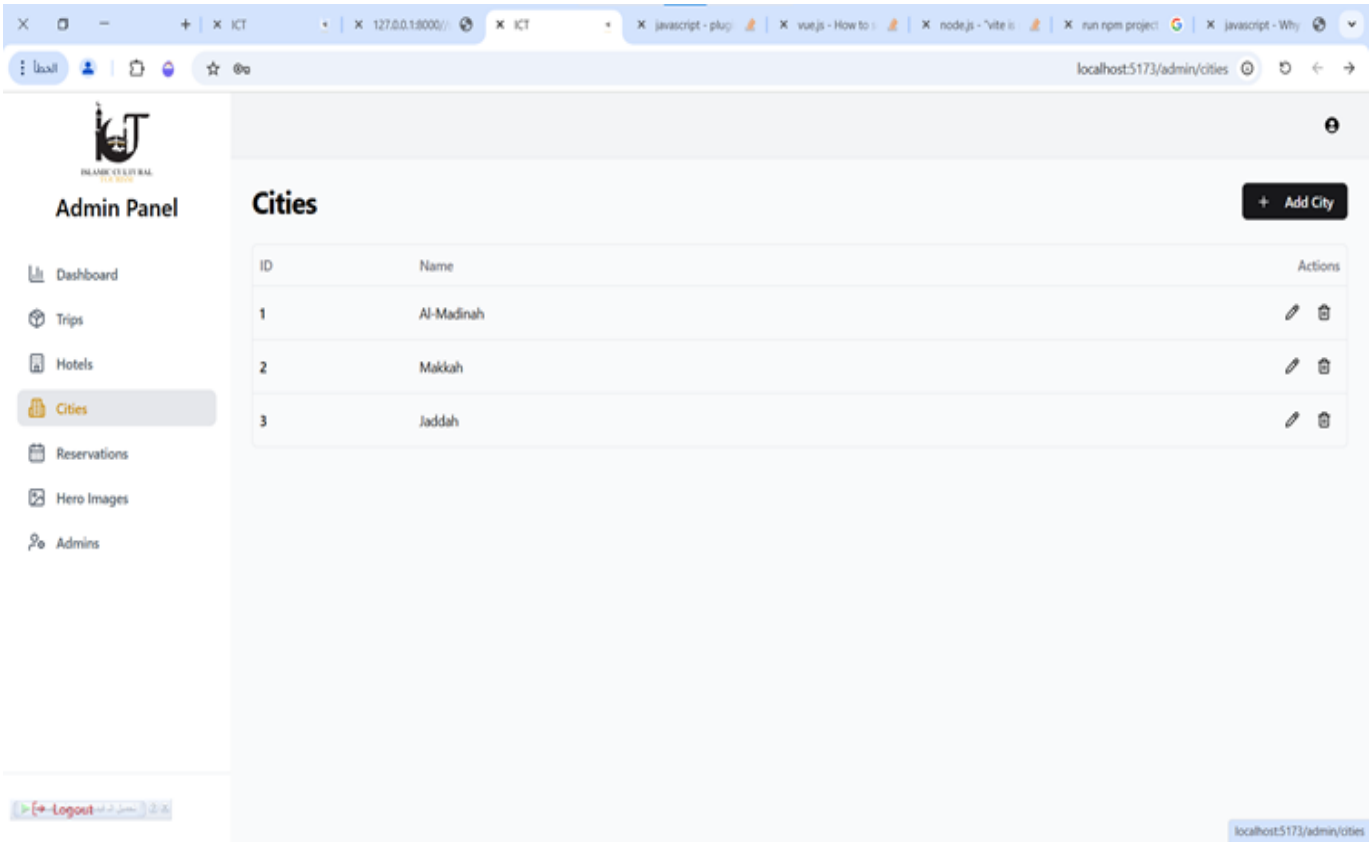

ISLAMIC CULTURAL TOURISM
HOME
HOTELS
ABOUT US
CONTACT US
Profile Details
Name
wael ali
Address
makkah
Phone Number
+9630962163657
Email
hasn.ad.na@gmail.com
Edit Profile
My Reservations
Dr. Brian Stroman V
Pending
Number of Tickets
1
Price per Ticket
$10
Total Amount
$10
Trip Dates
2025-04-10 - 2025-04-14

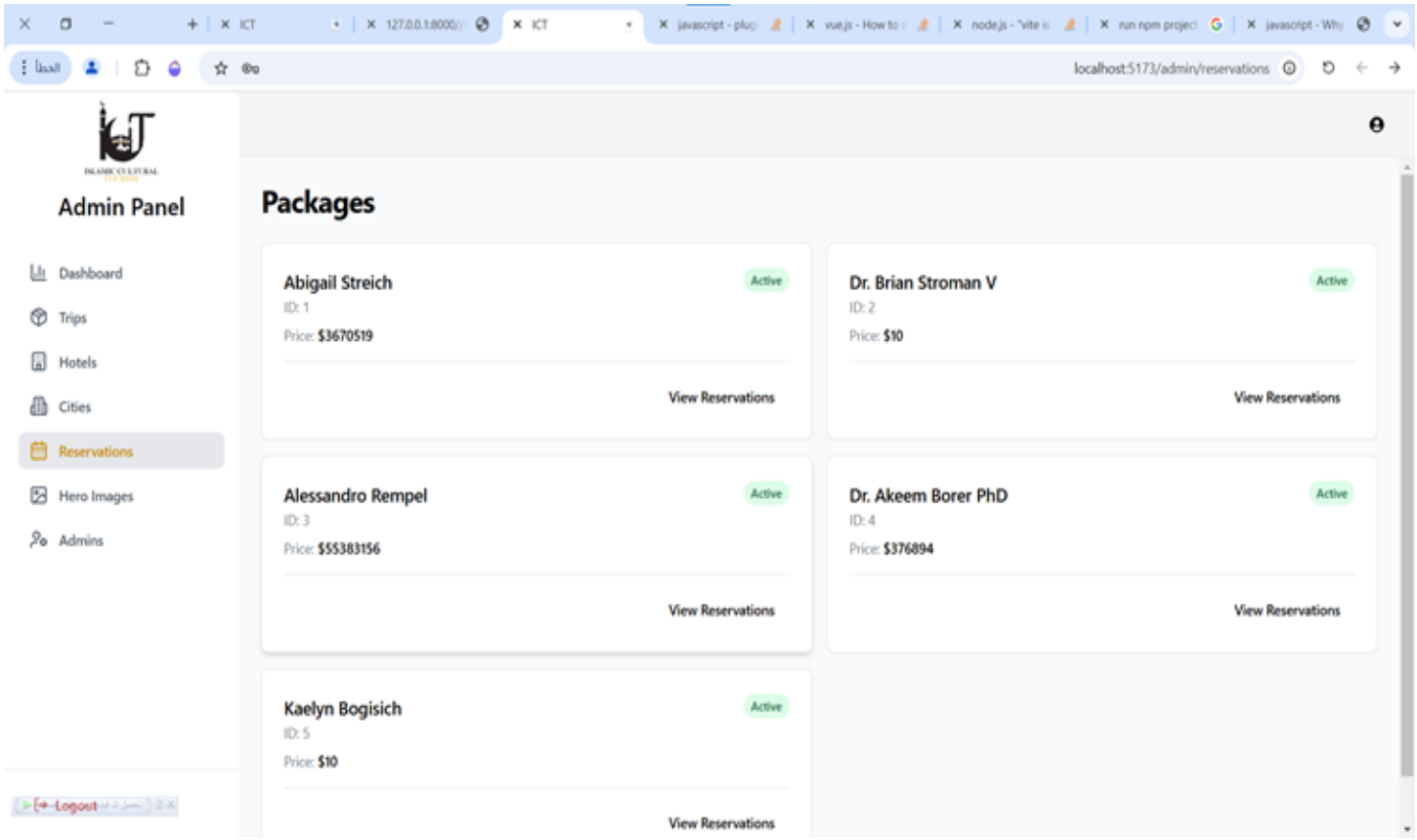

Log In
Welcome back! Please enter your credentials to login.
Email
hasn.ad.na@gmail.com
Password
Remember me
Forgot your password?
Log in
Don't have an account? Sign up
Welcome to Islamic Cultural Tourism
Explore the rich heritage and cultural wonders of Islamic civilization.

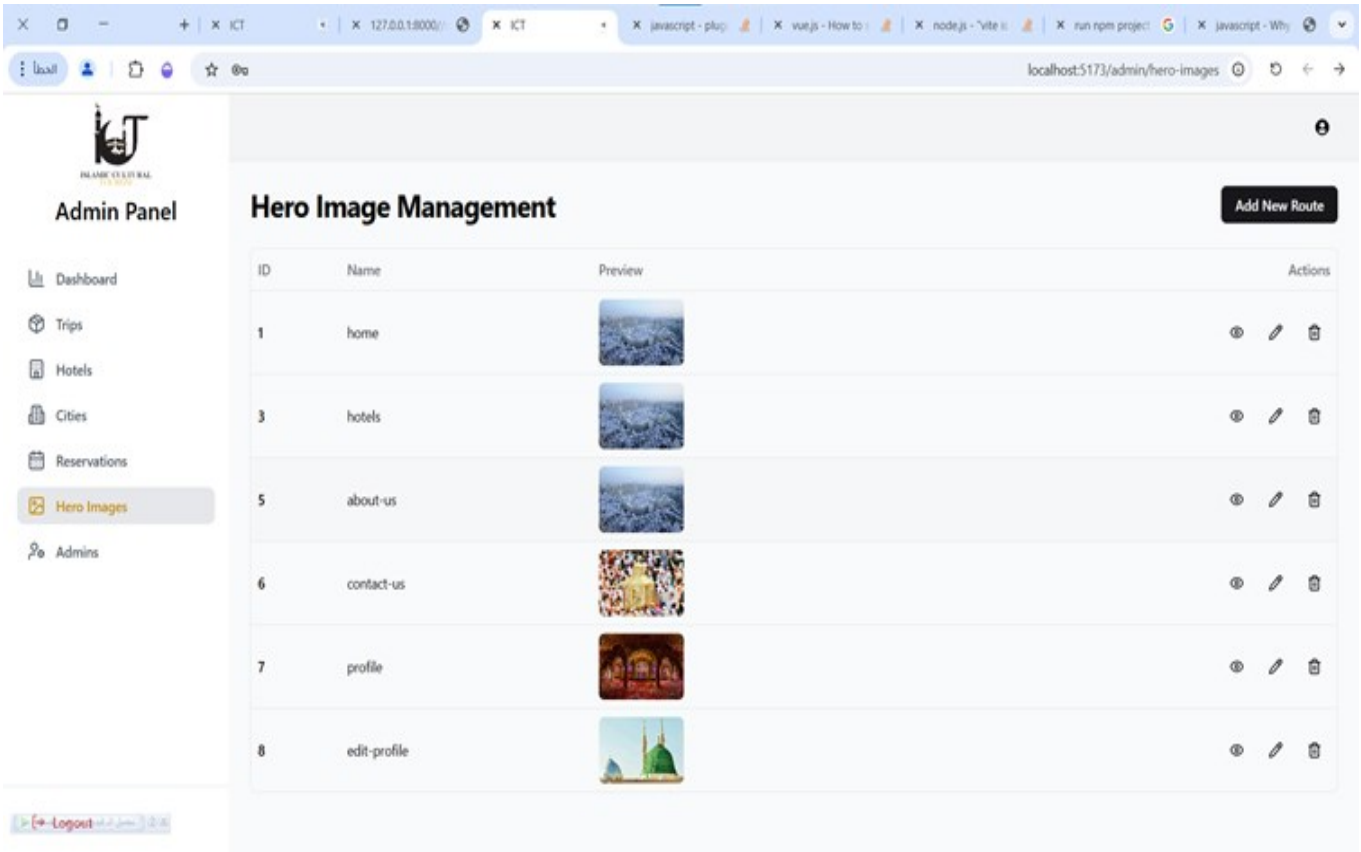

localhost:5173/admin
Admin Panel
Dashboard
Trips
Hotels
Cities
Reservations
Hero Images
Admins
Logout
Analytics Dashboard
Total Users
11
+11 from last month
Total Trips
20
+20 from last month
Active Packages
40
+20 since last week
Active Hotels
20
+20 since last month
Overview
100%
75%
50%
25%
0%
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Recent Sales
No recent sales available


localhost:5173/admin/trips
Admin Panel
Dashboard
Trips
Hotels
Cities
Reservations
Hero Images
Admins
Logout
Trip Management
Create Trip
ID
Name
Price
Start Date
End Date
Active
Actions
1
Abigail Streich
$3670519
2025-04-10
2025-04-13
Yes
2
Dr. Brian Stroman V
$10
2025-04-10
2025-04-14
Yes
3
Alessandro Rempel
$55383156
2025-04-10
2025-04-13
Yes
4
Dr. Akeem Borer PhD
$376894
2025-04-10
2025-04-19
Yes
5
Kaelyn Bogisich
$10
2025-04-10
2025-04-19
Yes


localhost:5173/admin/hotels
Admin Panel
Dashboard
Trips
Hotels
Cities
Reservations
Hero Images
Admins
Logout
Hotel Management
Create Hotel
Search by Name
Search hotels...
Filter by City
All Cities
ID Name Address City Actions
3 Dicta asperiores sint laudantium minus. 84461 Corkery Mission Suite 720 West Ahmad, VT 96888-5745 Makkah
4 Eaque eum aut. 5449 Gutmann Ferry North Salmaland, NE 30193 Jaddah
5 Labore voluptatem consectetur. 79931 Koch Terrace Apt. 097 South Marcelinachester, WI 04446-4177 Al-Madinah
6 Recusandae impedit omnis consectetur. 60042 Ivah Expressway West Ara, WA 14034 Al-Madinah
7 Impedit magnam repellendus. 14660 Clara Port Port Providenciberg, NY 58271 Al-Madinah
8 Occaecati quo doloribus. 17812 Bauch Pass Kattiemouth, TX 25717-7588 Makkah
9 Quia inventore iusto id. 95930 Labadie Cape Marksfort, IA 94291-3489 Al-Madinah
10 Ratione omnis magnam quod. 210 Maggio Spring South Queenie, ID 26354 Jaddah
localhost:5173/admin/hotels


localhost:5173/admin/cities
Admin Panel
Dashboard
Trips
Hotels
Cities
Reservations
Hero Images
Admins
Logout
Cities
+ Add City
ID Name Actions
1 Al-Madinah
2 Makkah
3 Jaddah
localhost:5173/admin/cities


localhost:5173/admin/reservations
Admin Panel
Dashboard
Trips
Hotels
Cities
Reservations
Hero Images
Admins
Logout
Packages
Abigail Streich
ID: 1
Price: $3670519
Active
View Reservations
Dr. Brian Stroman V
ID: 2
Price: $10
Active
View Reservations
Alessandro Rempel
ID: 3
Price: $55383156
Active
View Reservations
Dr. Akeem Borer PhD
ID: 4
Price: $376894
Active
View Reservations
Kaelyn Bogisich
ID: 5
Price: $10
Active
View Reservations


localhost:5173/admin/hero-images
Admin Panel
Dashboard
Trips
Hotels
Cities
Reservations
Hero Images
Admins
Logout
Hero Image Management
Add New Route
ID
Name
Preview
Actions
1
home
3
hotels
5
about-us
6
contact-us
7
profile
8
edit-profile