

**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**

## 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:

- **HTML5, CSS3, Bootstrap, and JavaScript**

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

<b>Project Title</b>	<b>Hajj and Umrah Trip Management System</b>
<b>Supervisor</b>	Mr. Shadi Bleidi
<b>Project Type</b>	Web Site
<b>Development Methodology</b>	Incremental Model
<b>Programming Languages</b>	PHP, React, Tailwind Css, Vite
<b>Frameworks &amp; Tools</b>	Laravel, Bootstrap, MySQL, VS Code, XAMPP
<b>Target Users</b>	Pilgrims, Administrators, Super Admin
<b>Main Objective</b>	To design and implement a digital platform for managing Hajj and Umrah trips efficiently.
<b>Start Date</b>	[Add your project start date]
<b>End Date</b>	[Add your project end date]

---

## 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.

# **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.

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

### 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.

---

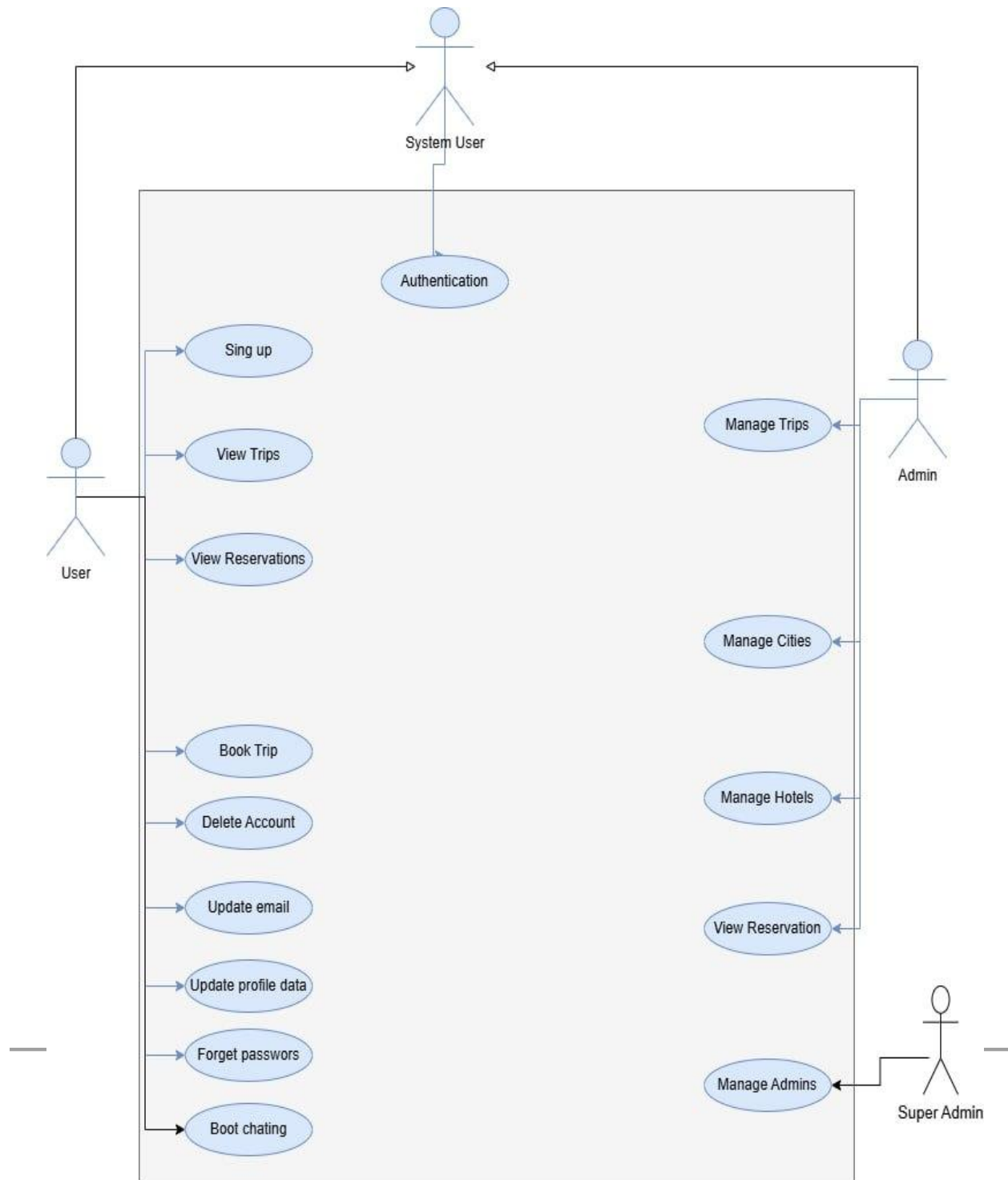
## 4.9 Use Case Diagram (Overview)

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)

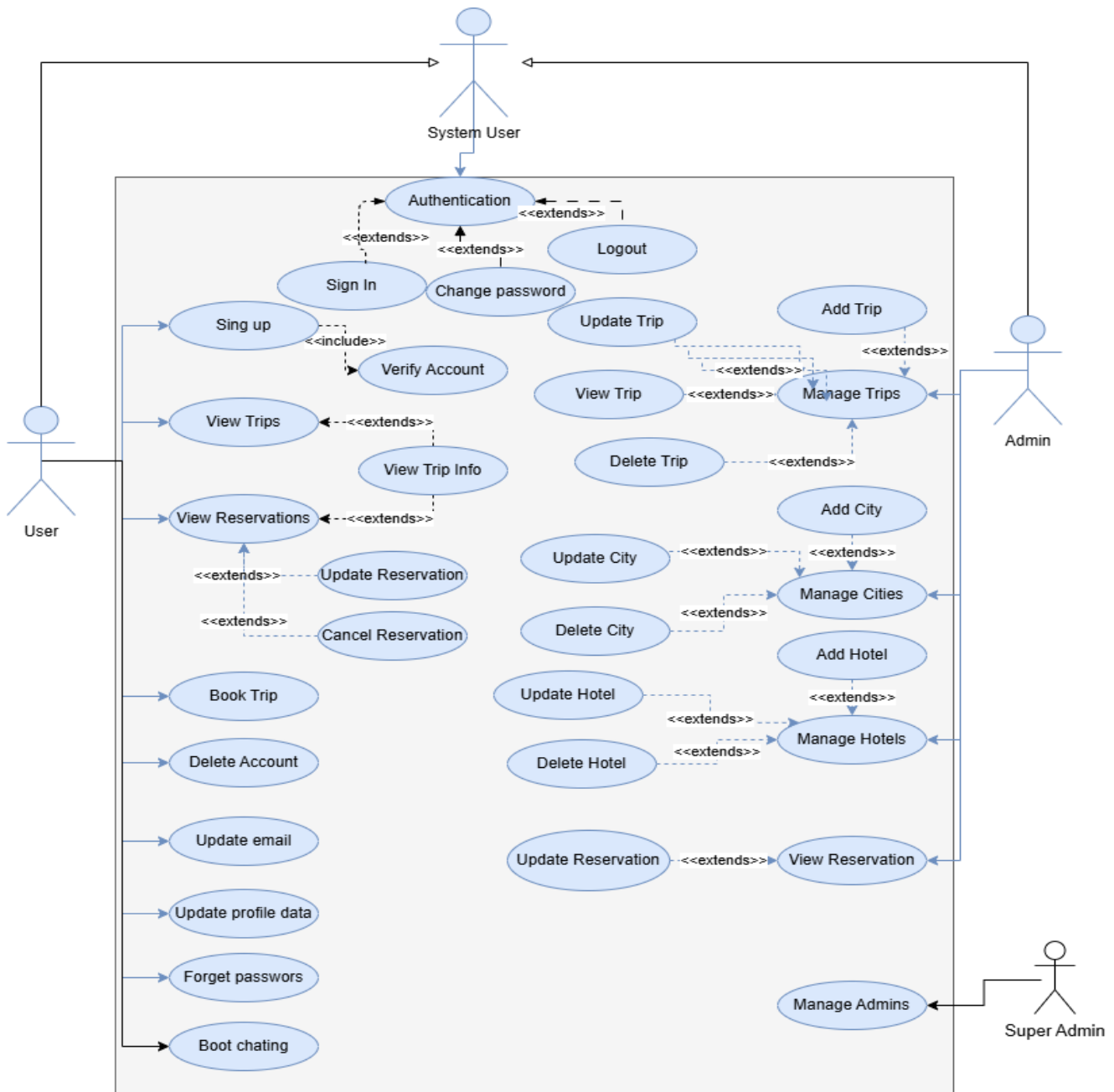


#### 4.9.1 Use Case Diagram (High-Level)



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

#### 4.9.2 Use Case Diagram (Low-Level)



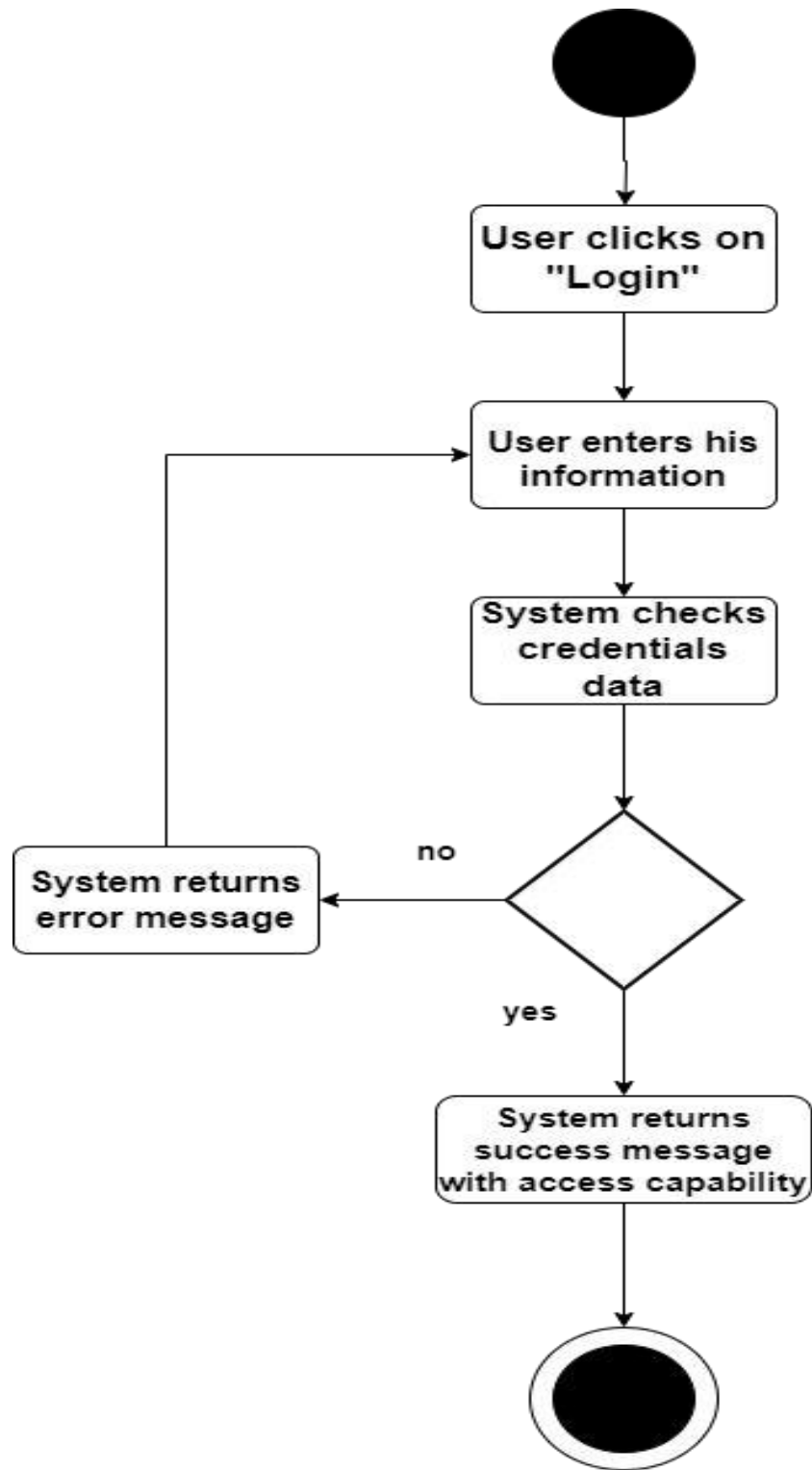
Use Case Diagram (Low-Level-) – Figure 3

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

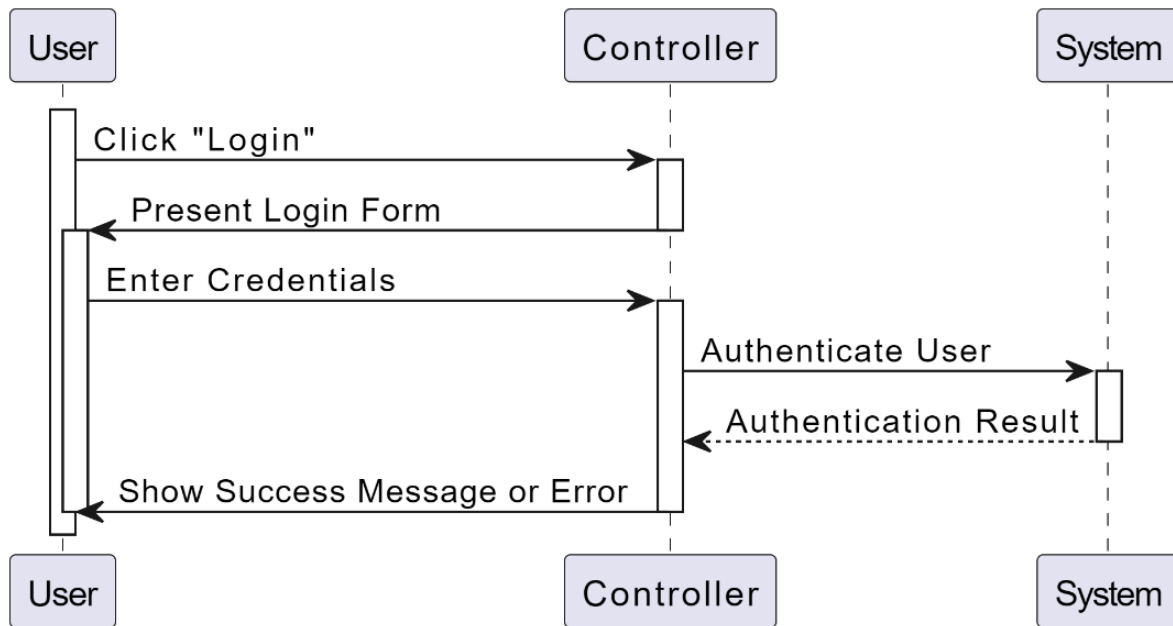
### 5.1. System-User Requirements

#### 5.1.1. Sign in

<b>Use Case ID</b>	UC-1
<b>Use Case</b>	Sign In
<b>Actor</b>	System User
<b>Pre-Condition</b>	Has an account in the system
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Login"</li><li>2. User enter (username or email) and password</li><li>3. System checks credentials data</li><li>4. System returns success message with capability for access to the system</li></ol>
<b>Alternative Scenario</b>	In step 3 if the credentials data is incorrect the system returns error message and redirect user to login again



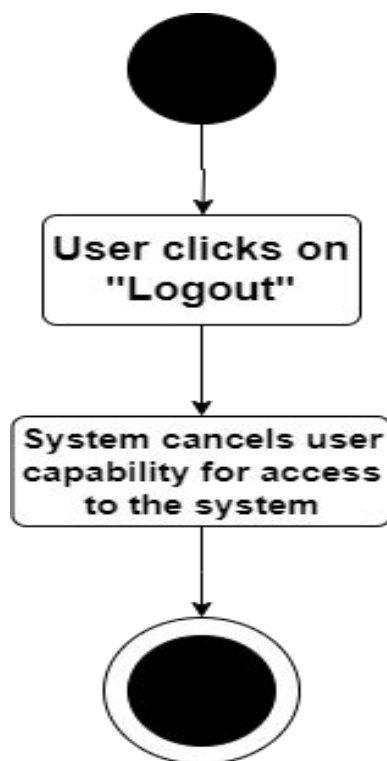
**Activity diagram(log in) – Figure 4**



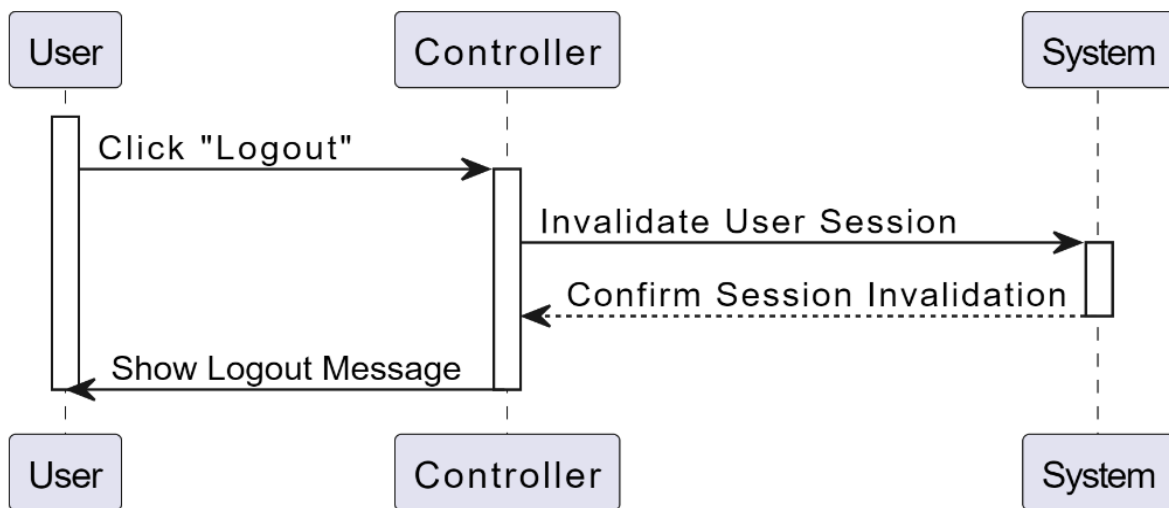
**Sequence diagram(log in) – Figure 5**

### 5.1.2. Log out

<b>Use Case ID</b>	UC-2
<b>Use Case</b>	Logout
<b>Actor</b>	System User
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Logout"</li><li>2. System cancels user capability for access to the system</li></ol>
<b>Alternative Scenario</b>	



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

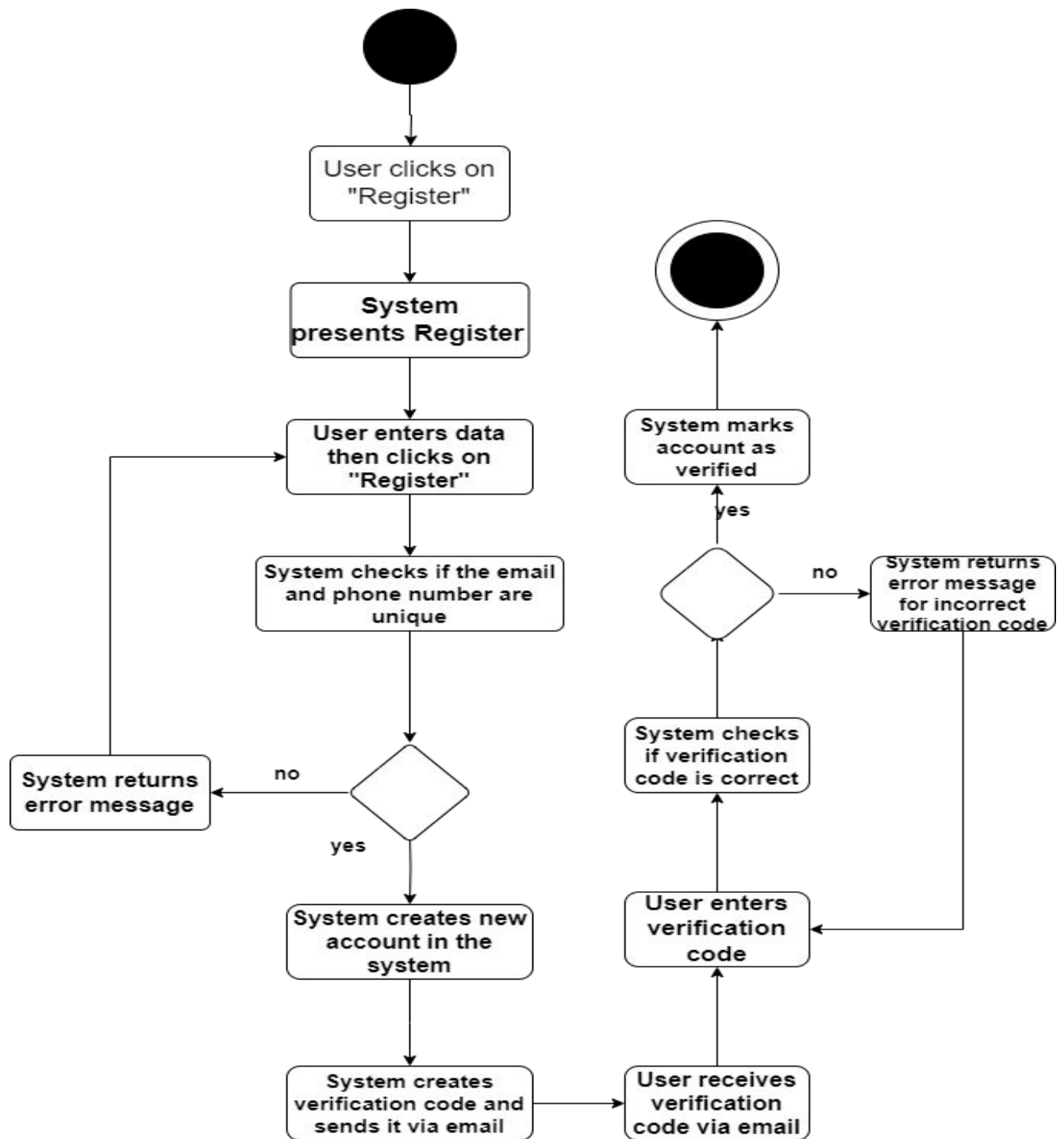


**Sequence diagram(log out) – Figure 7**

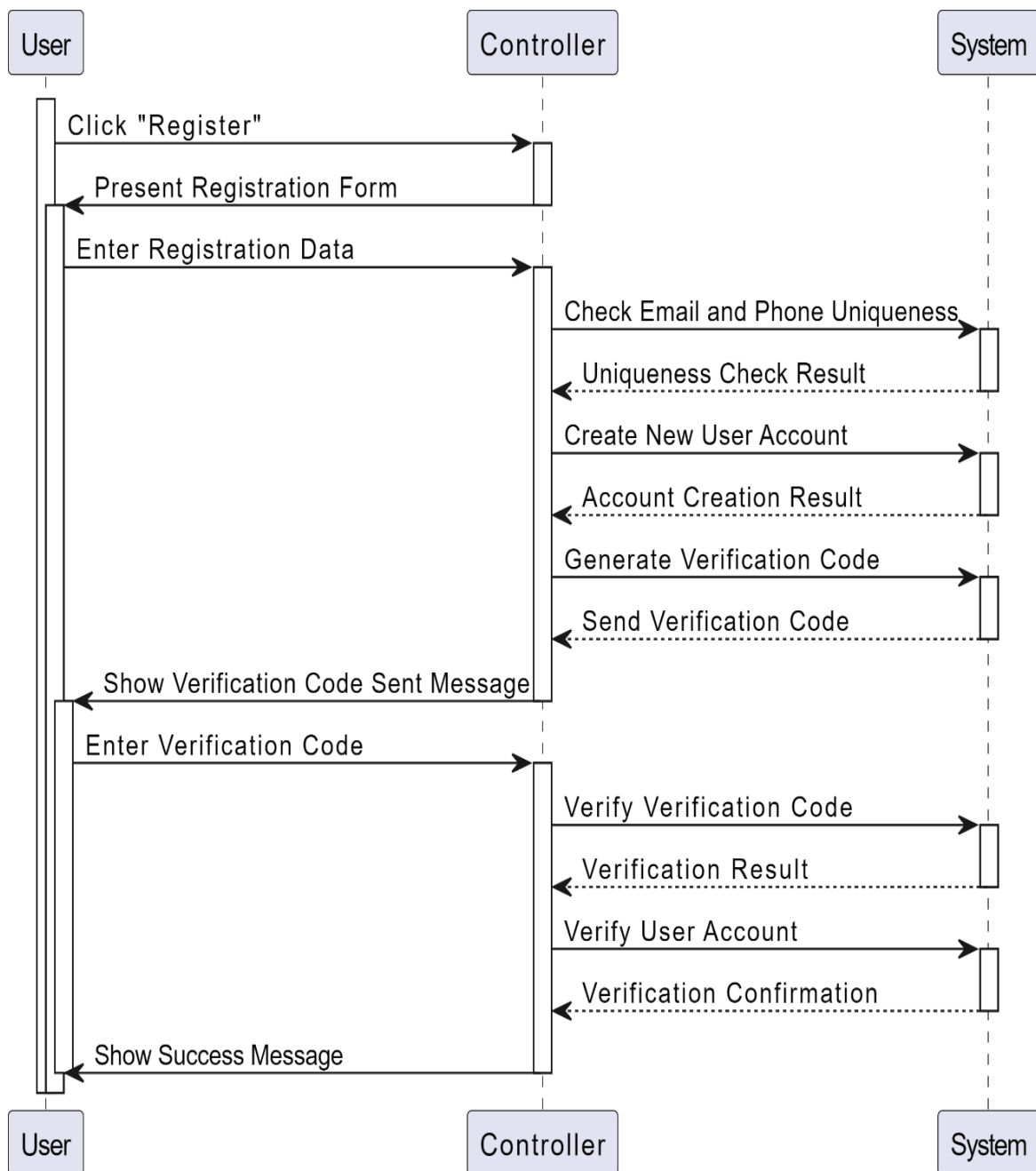
### 5.1.3. Sign up

<b>Use Case ID</b>	UC-5
<b>Use Case</b>	Sign up
<b>Actor</b>	User
<b>Pre-Condition</b>	-
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Register"</li><li>2. System presents form with (name, email, address, phone number)</li><li>3. User enter data then clicks on "Register"</li><li>4. System checks if the email and phone number are unique</li><li>5. System creates new account in the system</li><li>6. System creates verification code and send it via email</li><li>7. User receive verification code via his email</li><li>8. User enter verification code</li><li>9. System check if verification code is correct</li><li>10. System marks account as verified</li></ol>
<b>Alternative Scenario</b>	<p>In step 4 if the email or phone number are not unique, system return error message and redirect user to enter data again</p> <p>In step 9 if the verification code incorrect the system return error message and redirect user to enter code again</p>





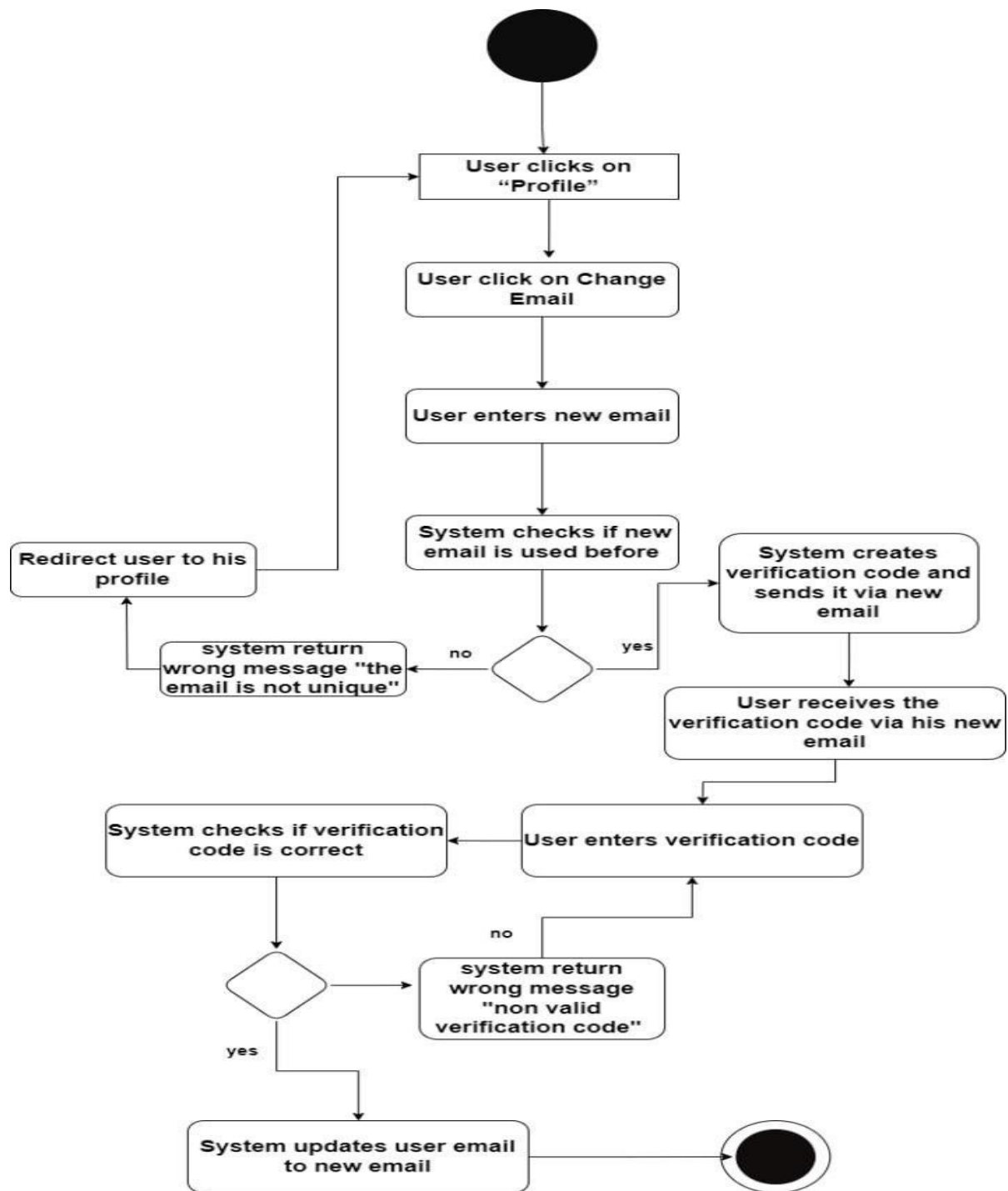
**Activity diagram(Sign up) – Figure 8**



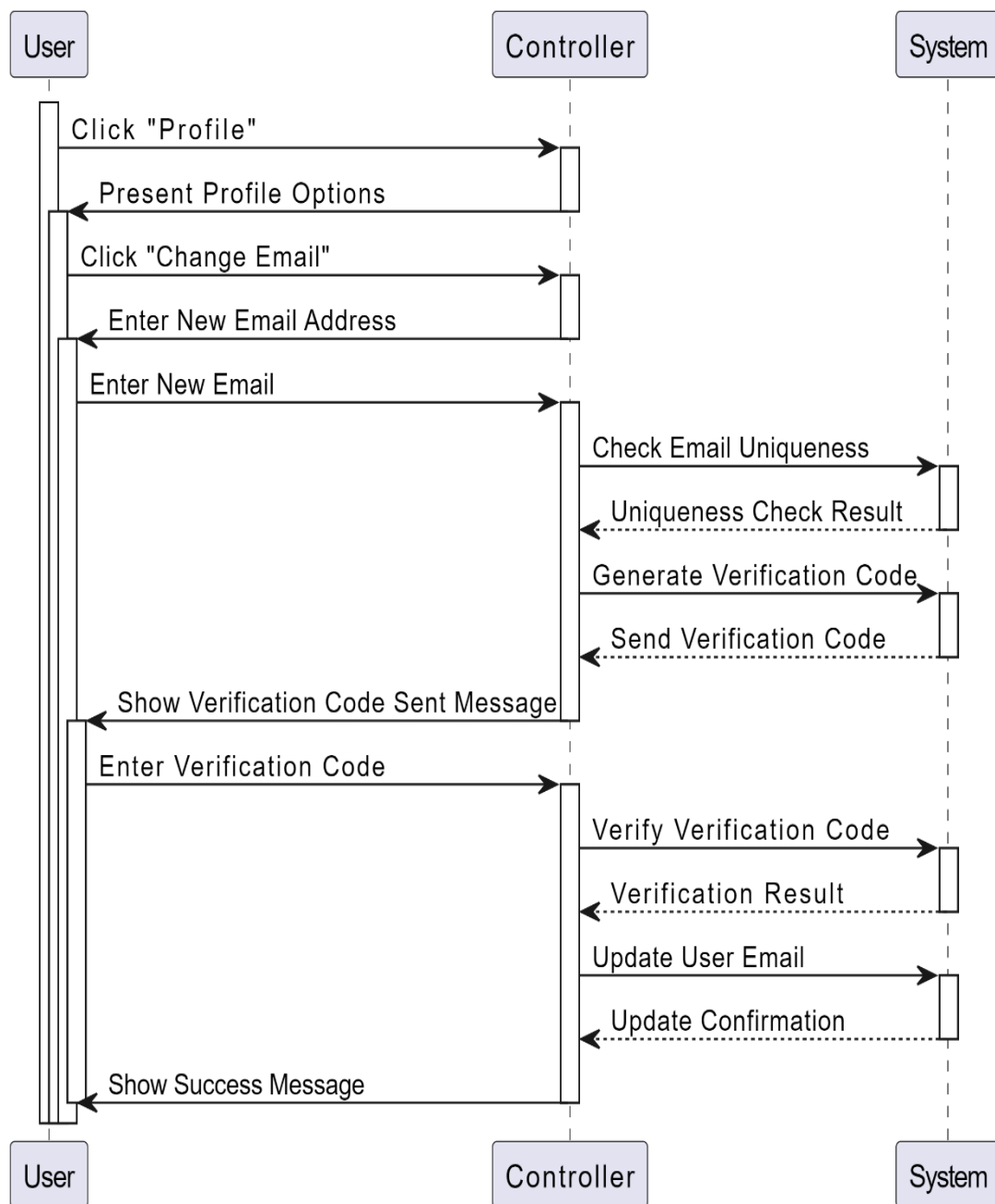
**Sequence diagram(Sign up) – Figure 9**

#### 5.1.4 Update email

<b>Use Case ID</b>	UC-6
<b>Use Case</b>	Update email
<b>Actor</b>	User
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Profile"</li><li>2. User clicks on "Change email"</li><li>3. User enter new email</li><li>4. System checks if new email is used before</li><li>5. System creates verification code and send it via new email</li><li>6. User receive the verification code via his new email</li><li>7. User enter verification code</li><li>8. System checks if verification code is correct</li><li>9. System update user email to new email</li></ol>
<b>Alternative Scenario</b>	<p>In step 4 if the entered email is not unique, system return error message and redirect user to his profile</p> <p>In step 8 if the verification code is incorrect, system return error message and redirect user to enter code again</p>



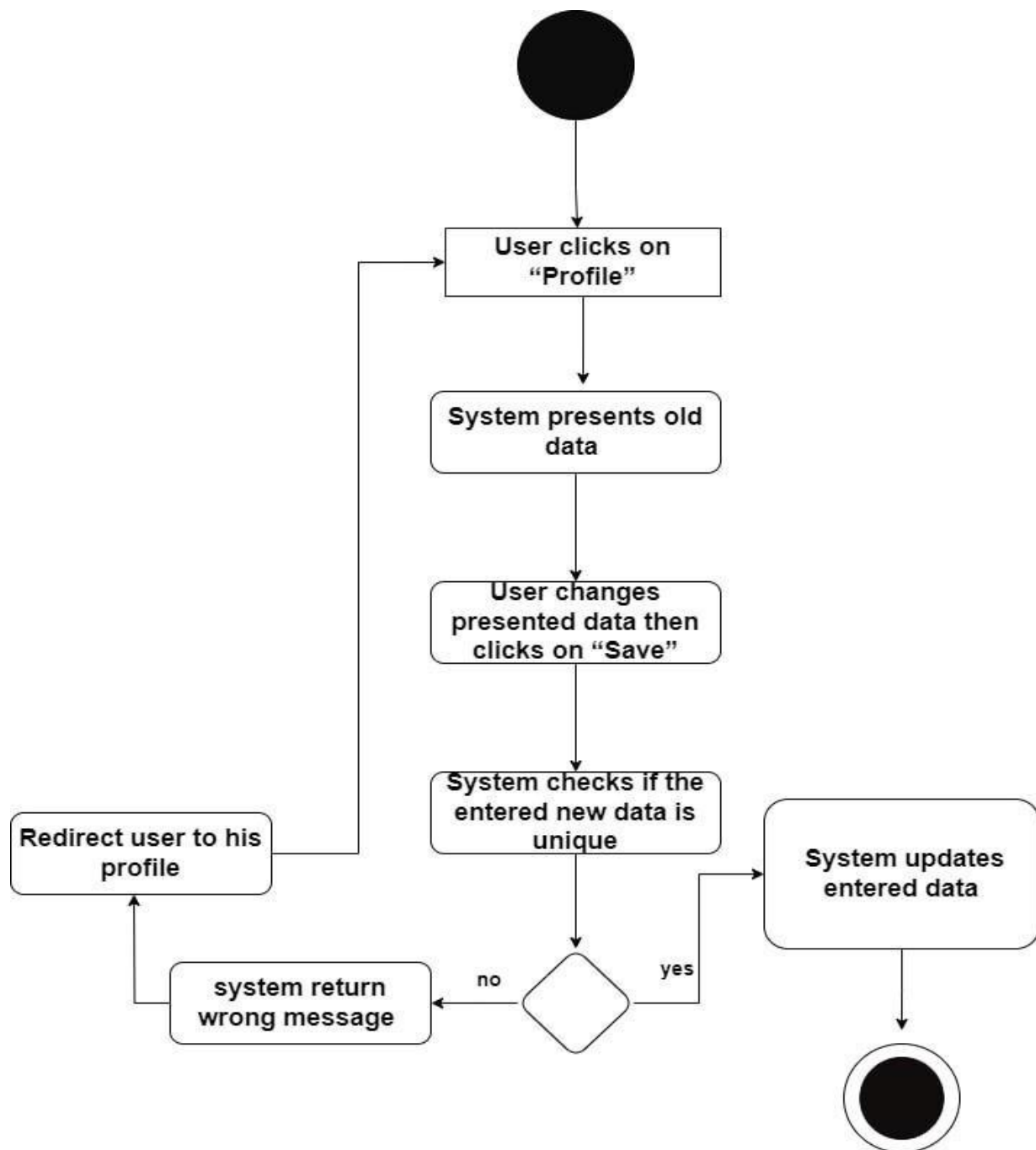
**Activity diagram(Update email) – Figure 10**



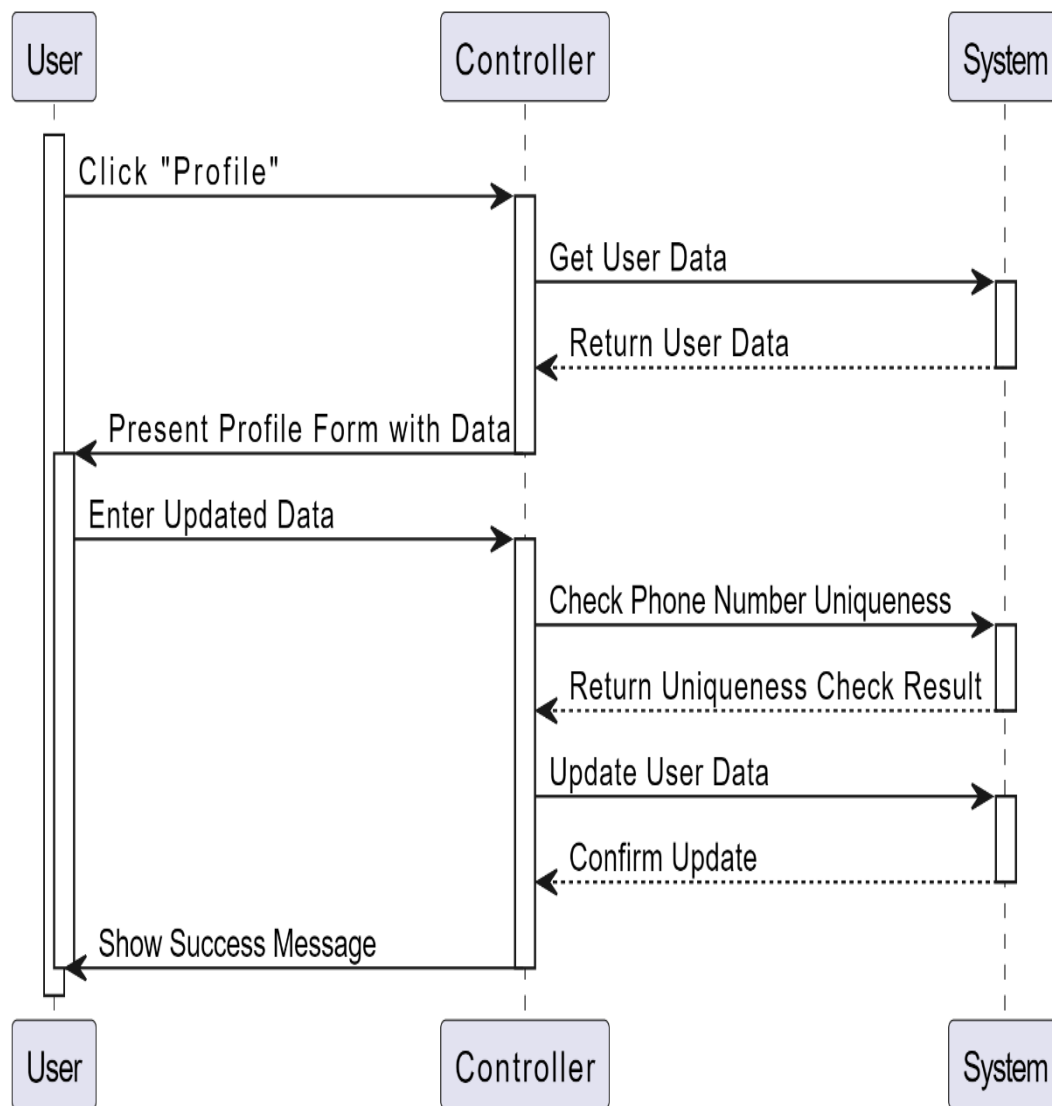
**Sequence diagram(Update email) – Figure 11**

### 5.1.5 Update profile data

<b>Use Case ID</b>	UC-7
<b>Use Case</b>	Update profile data
<b>Actor</b>	User
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Profile”</li><li>2. System present old data (name, phone number, address)</li><li>3. User change presented data then clicks on “Save”</li><li>4. System checks if the entered phone number is unique</li><li>5. System update entered data</li></ol>
<b>Alternative Scenario</b>	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**

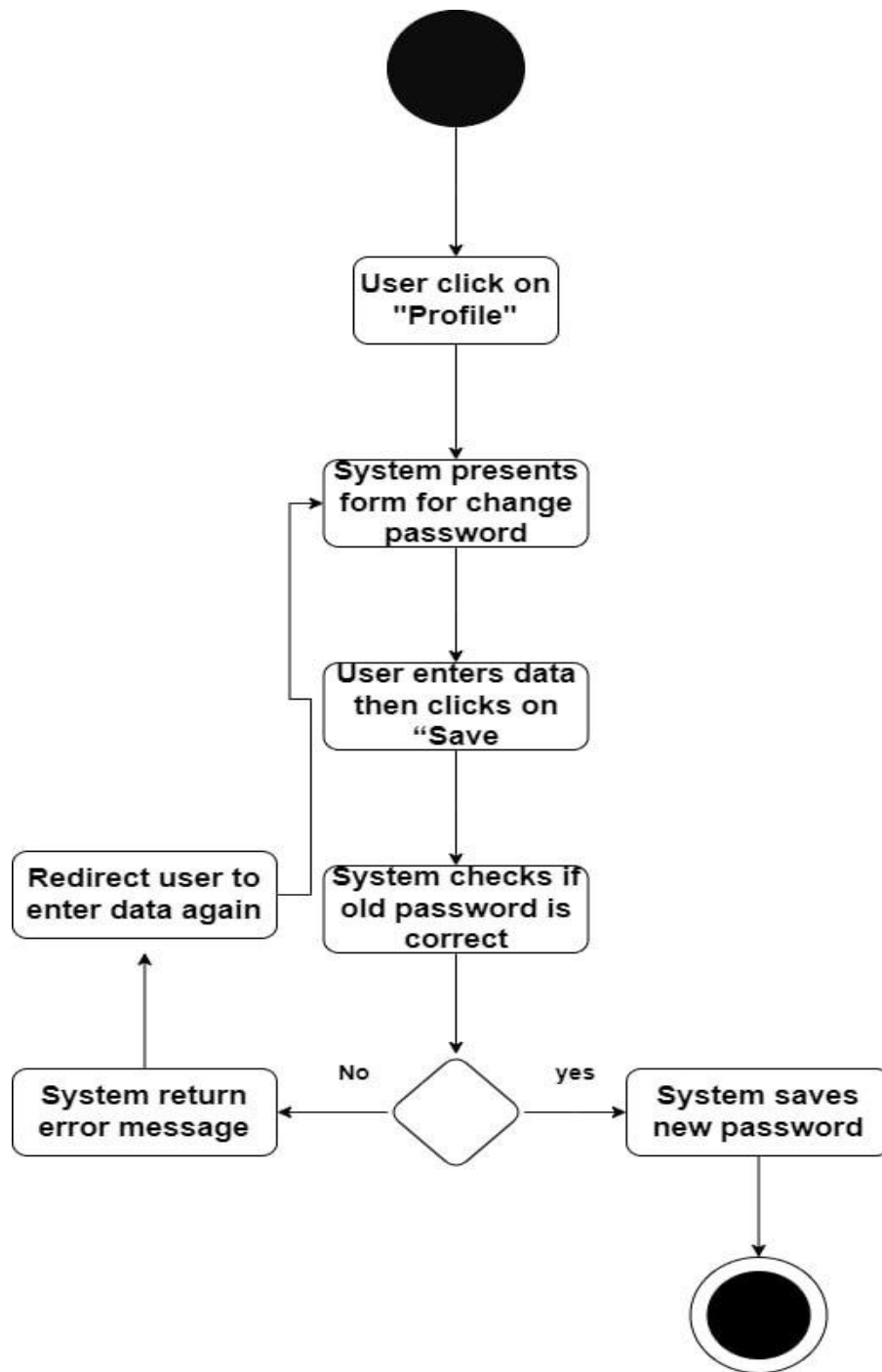


**Sequence diagram(Update profile data) – Figure 13**

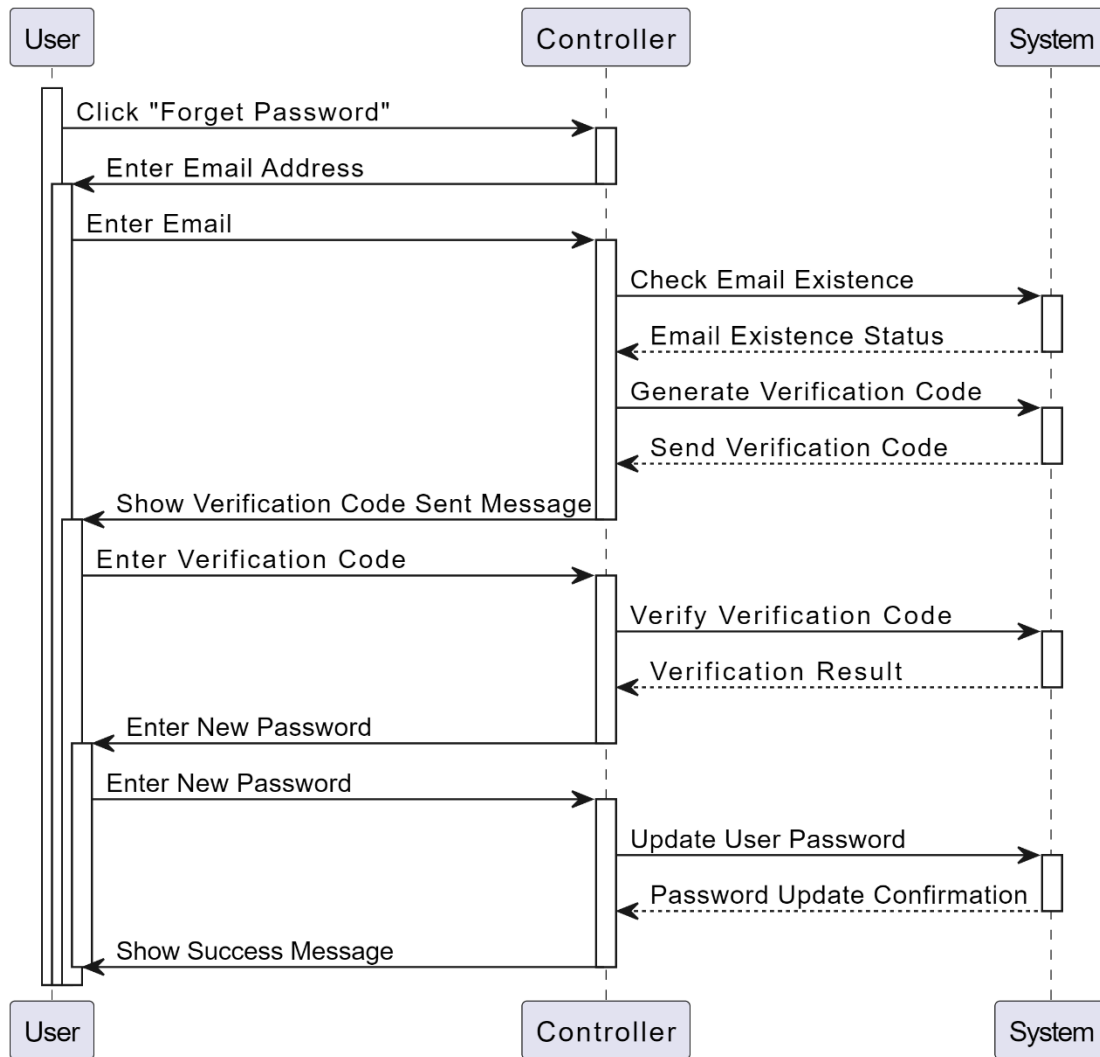


### 5.1.6 Forget password

<b>Use Case ID</b>	UC-8
<b>Use Case</b>	Forget password
<b>Actor</b>	User
<b>Pre-Condition</b>	Has an account in the system, User in sign in page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Forget Password"</li><li>2. User enter his email</li><li>3. System checks if the email exists in the system</li><li>4. System creates verification code and send it via entered email</li><li>5. User receive forget password code</li><li>6. User enter verification code</li><li>7. System checks if the verification code is correct</li><li>8. User enter new password</li><li>9. System update user password</li></ol>
<b>Alternative Scenario</b>	<p>In step 3 if the email dose not exists the system return error message and redirect user to enter email again</p> <p>In step 7 if the verification code is incorrect the system return error message and redirect user to enter password again</p>



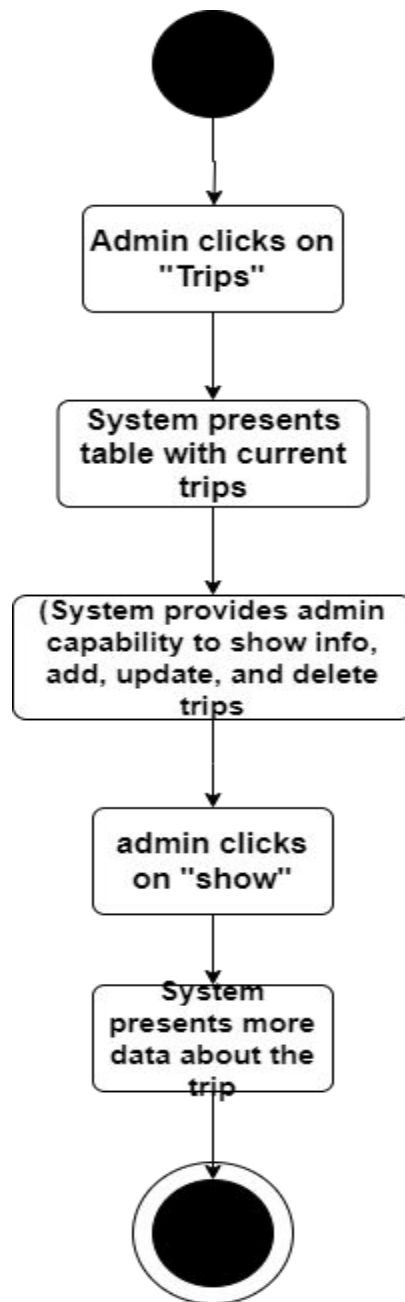
**Activity diagram(Forget password) – Figure 14**



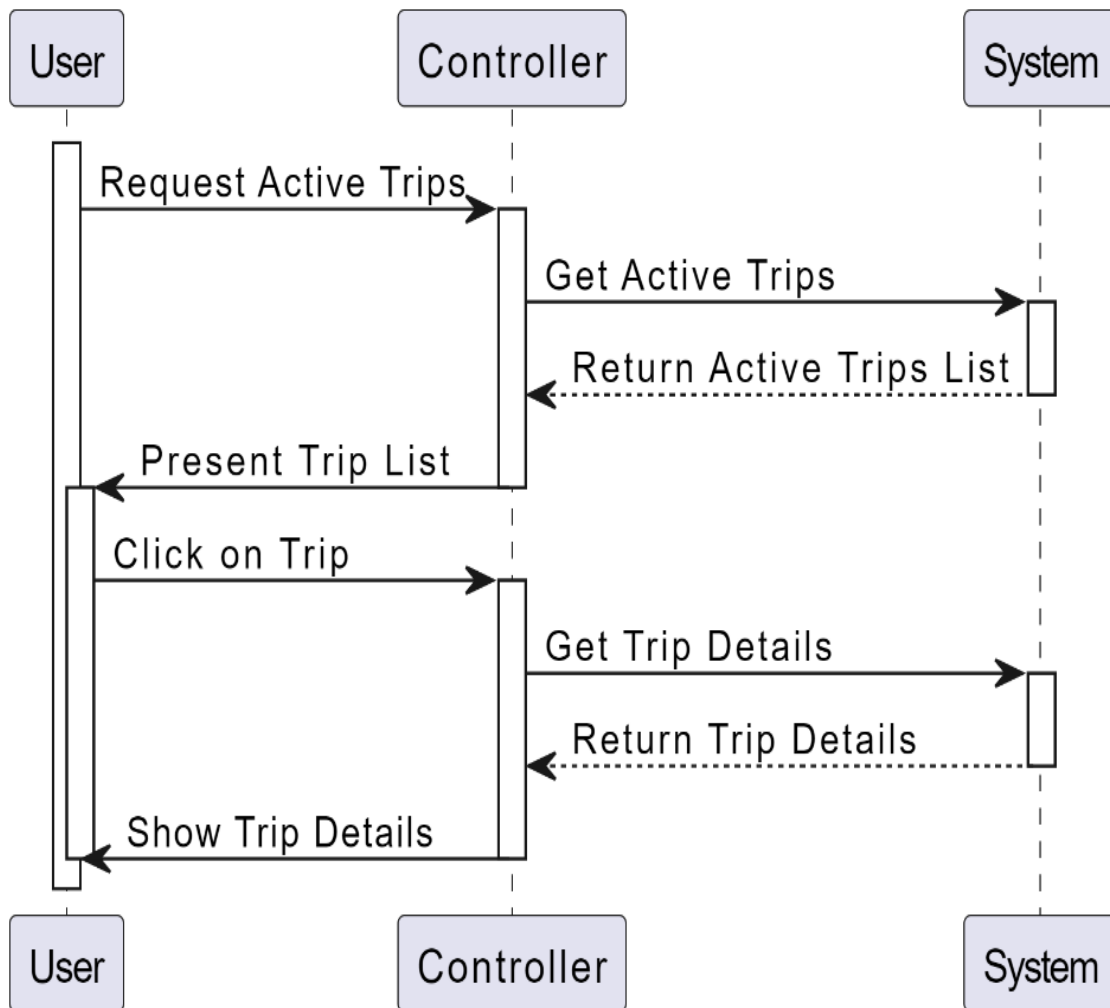
**Sequence diagram(Forget password) – Figure 15**

### 5.1.7 View trip

<b>Use Case ID</b>	UC-9
<b>Use Case</b>	View Trips
<b>Actor</b>	User
<b>Pre-Condition</b>	User in home page
<b>Main Scenario</b>	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
<b>Alternative Scenario</b>	-



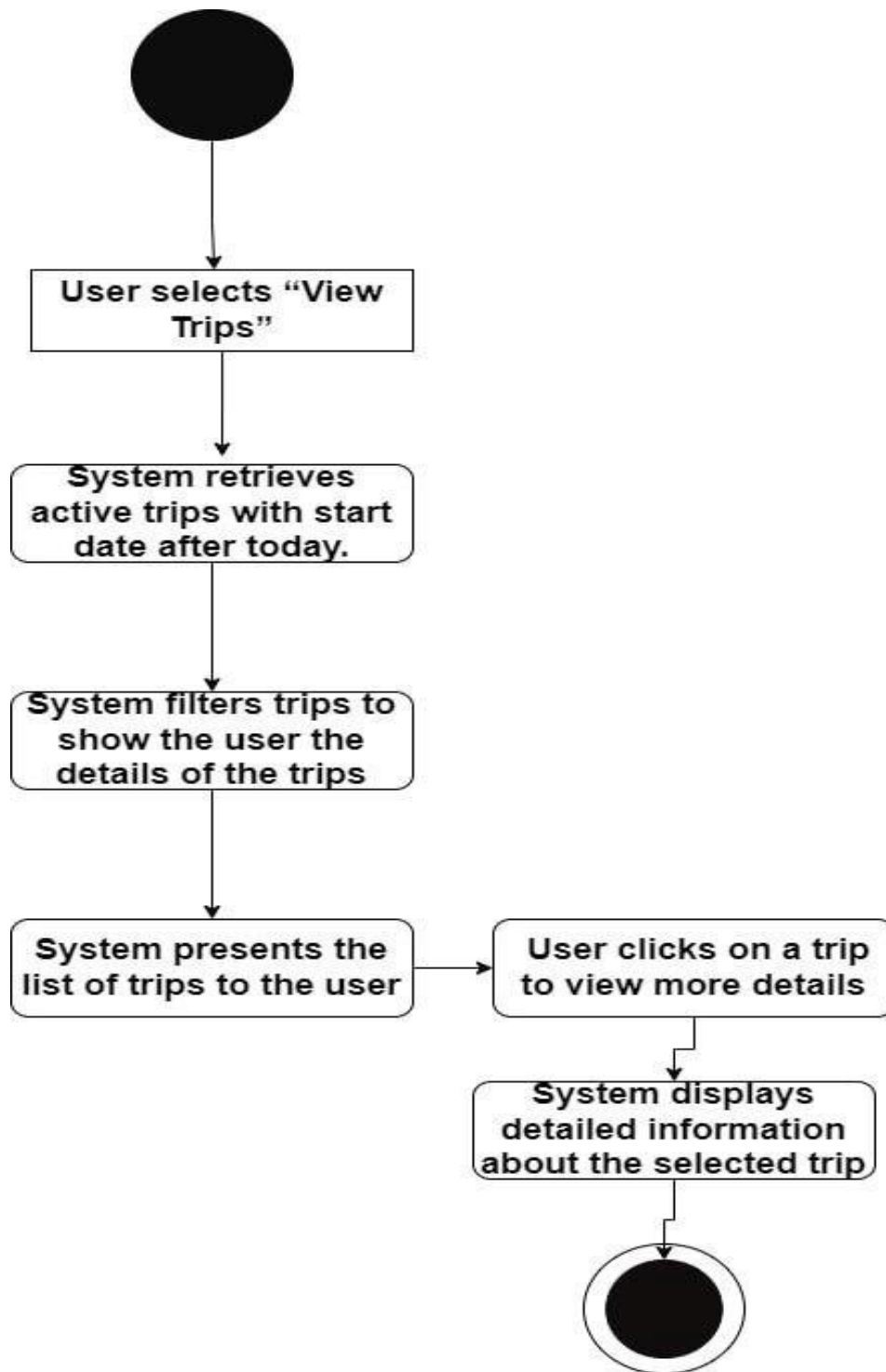
**Activity diagram(View Trips) – Figure 16**



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

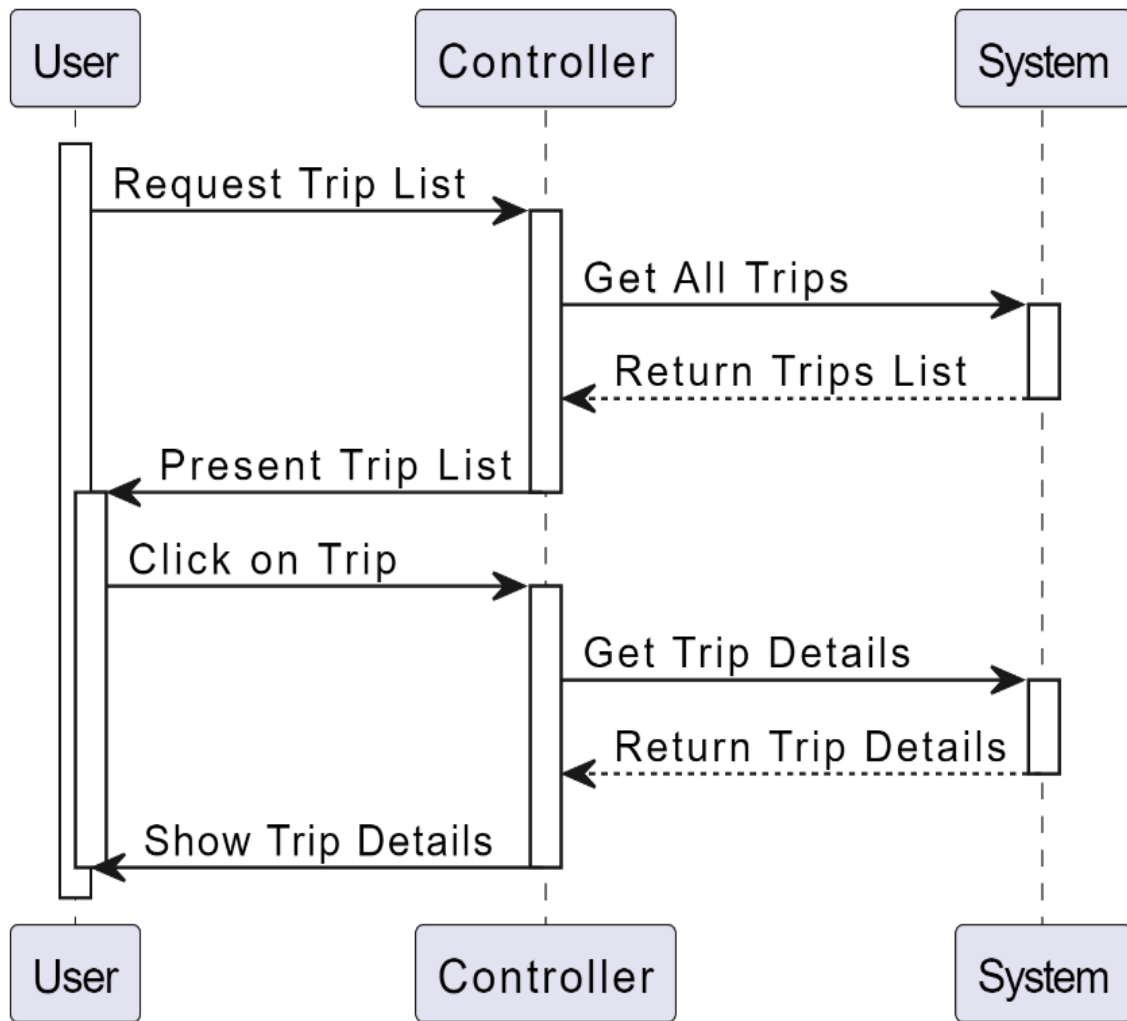
### 5.1.8 View Trip info

<b>Use Case ID</b>	UC-10
<b>Use Case</b>	View Trip Info
<b>Actor</b>	User
<b>Pre-Condition</b>	User in home page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. System presents list of trips each trip is clickable to show more details</li><li>2. User clicks on any trip</li><li>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))</li></ol>
<b>Alternative Scenario</b>	-



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



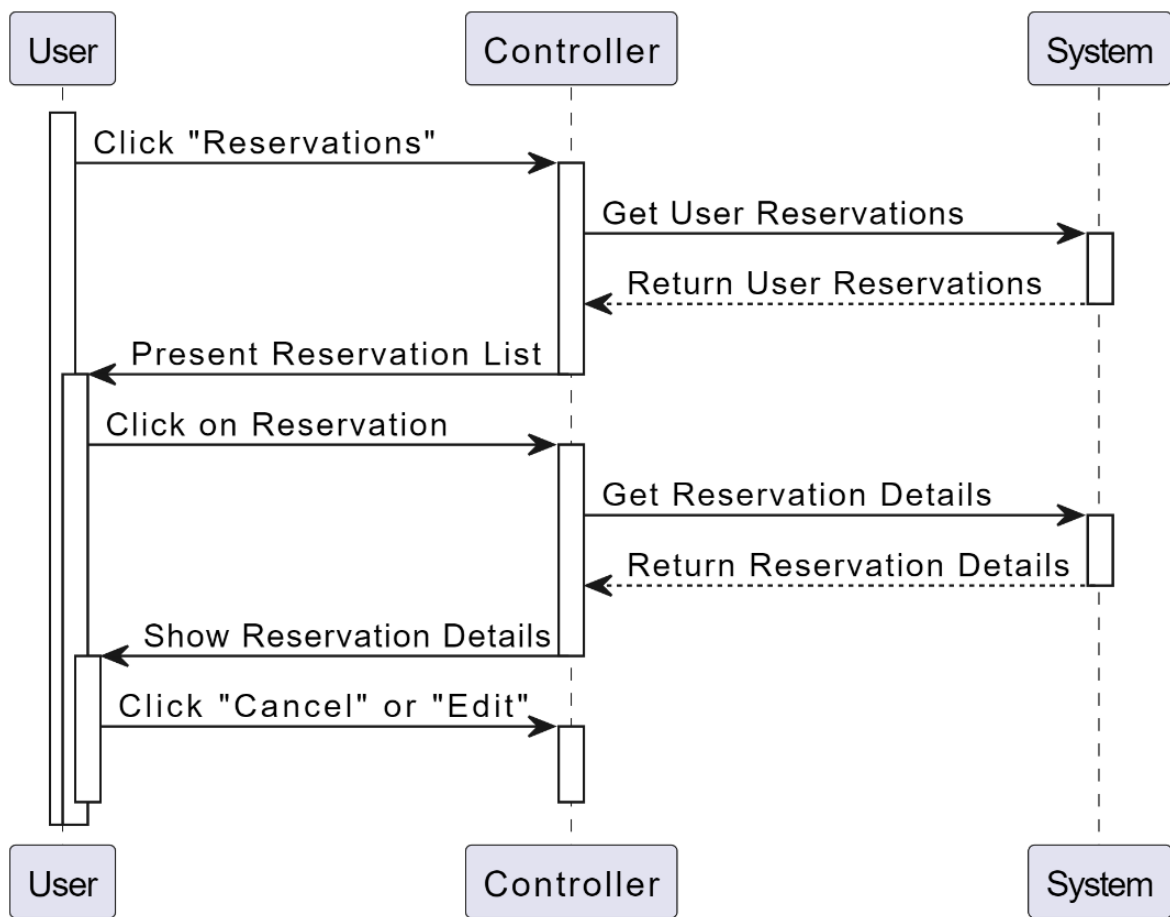


**Sequence diagram(view trips info) – Figure 19**

### 5.1.9 View Reservations

<b>Use Case ID</b>	UC-11
<b>Use Case</b>	View Reservations
<b>Actor</b>	User
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Reservations”</li><li>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)</li></ol>
<b>Alternative Scenario</b>	-

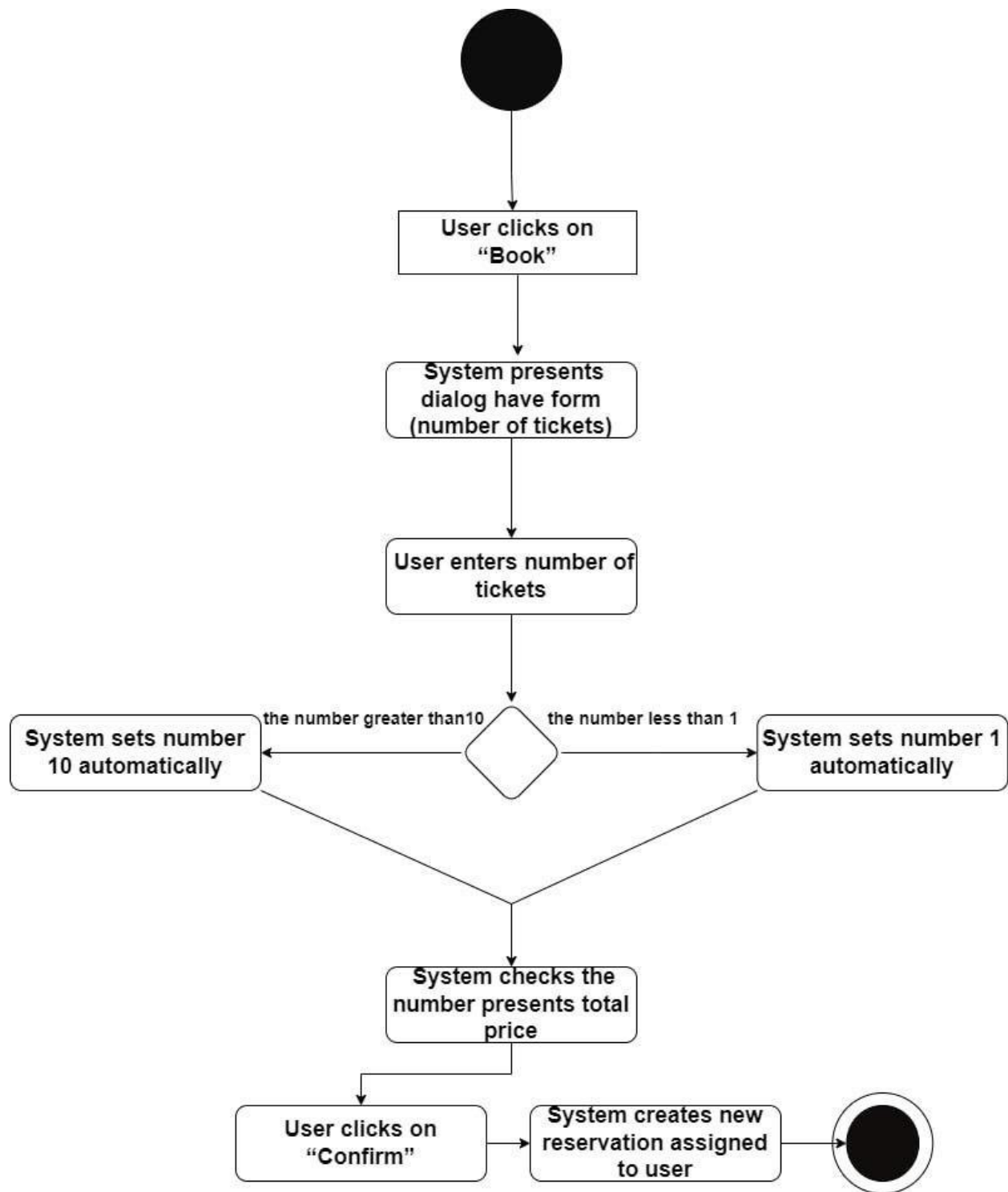
**Activity diagram(View Reservation) – Figure 20**



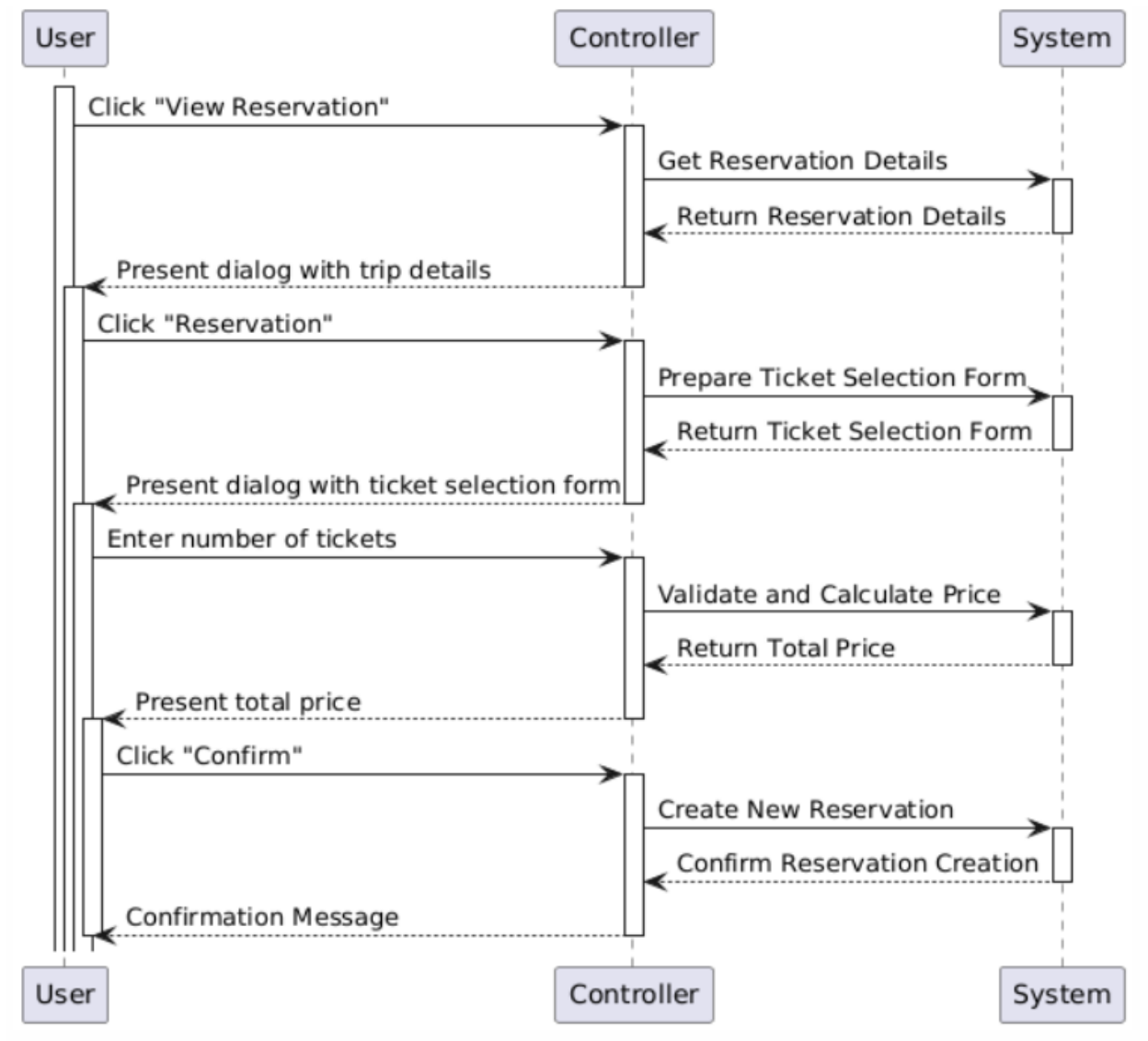
**Sequence diagram(View Reservation) – Figure 21**

### 5.1.10 Book trip

<b>Use Case ID</b>	UC-12
<b>Use Case</b>	Book Trip
<b>Actor</b>	User
<b>Pre-Condition</b>	Already logged in, and in view trip info page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “view reservation”</li><li>2. System presents dialog have details of trip</li><li>3. user click on reservation</li><li>4. System presents dialog have form to select number of ticket</li><li>5. User enter number greater than 0 and less or equals 10</li><li>6. System checks the number presents total price</li><li>7. User clicks on “Confirm”</li><li>8. system create new reservation assigned to user</li></ol>
<b>Alternative Scenario</b>	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**

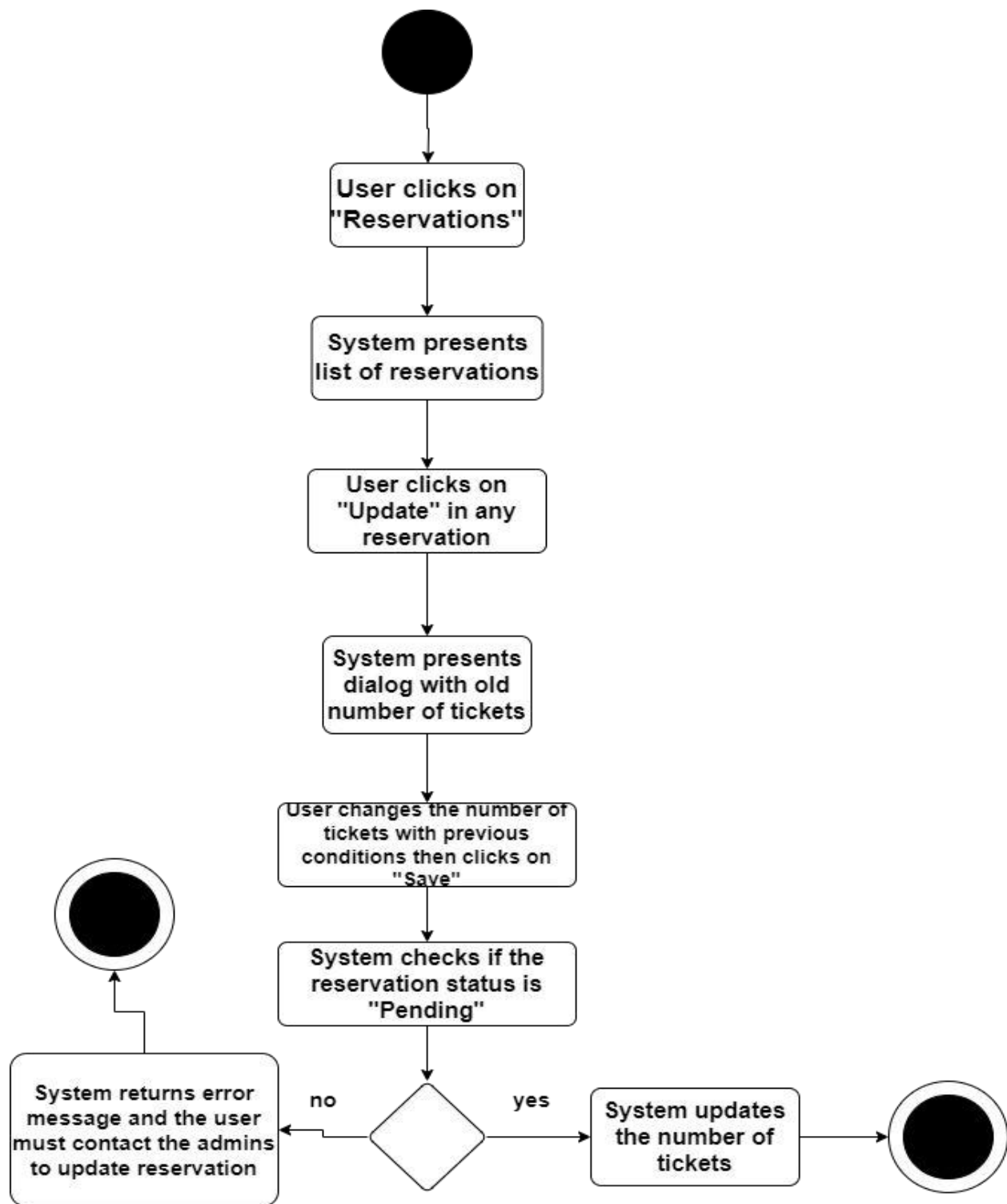


**Sequence diagram(Book trip) – Figure 23**

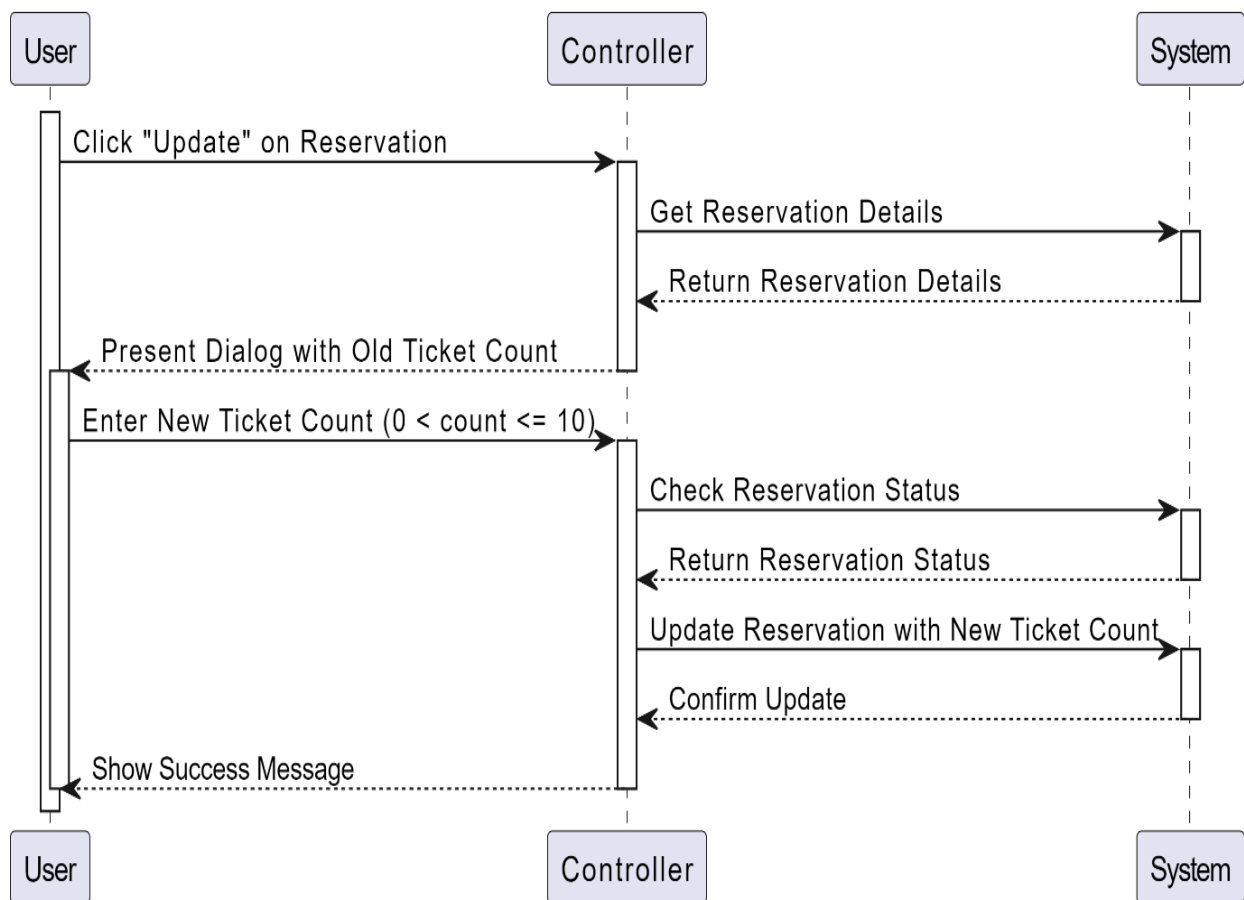
### 5.1.11 Update Reservation

<b>Use Case ID</b>	UC-13
<b>Use Case</b>	Update Reservation
<b>Actor</b>	User
<b>Pre-Condition</b>	Already logged in, and in view reservations page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Update" in any reservation</li><li>2. System present dialog with old number of tickets</li><li>3. User change the number of tickets with previous conditions then clicks on "Save"</li><li>4. System checks the reservation status is "Pending"</li><li>5. System updates number of tickets</li></ol>
<b>Alternative Scenario</b>	In step 4 if the status is not "Pending" the system return error message and the user must contact the admins to update reservation





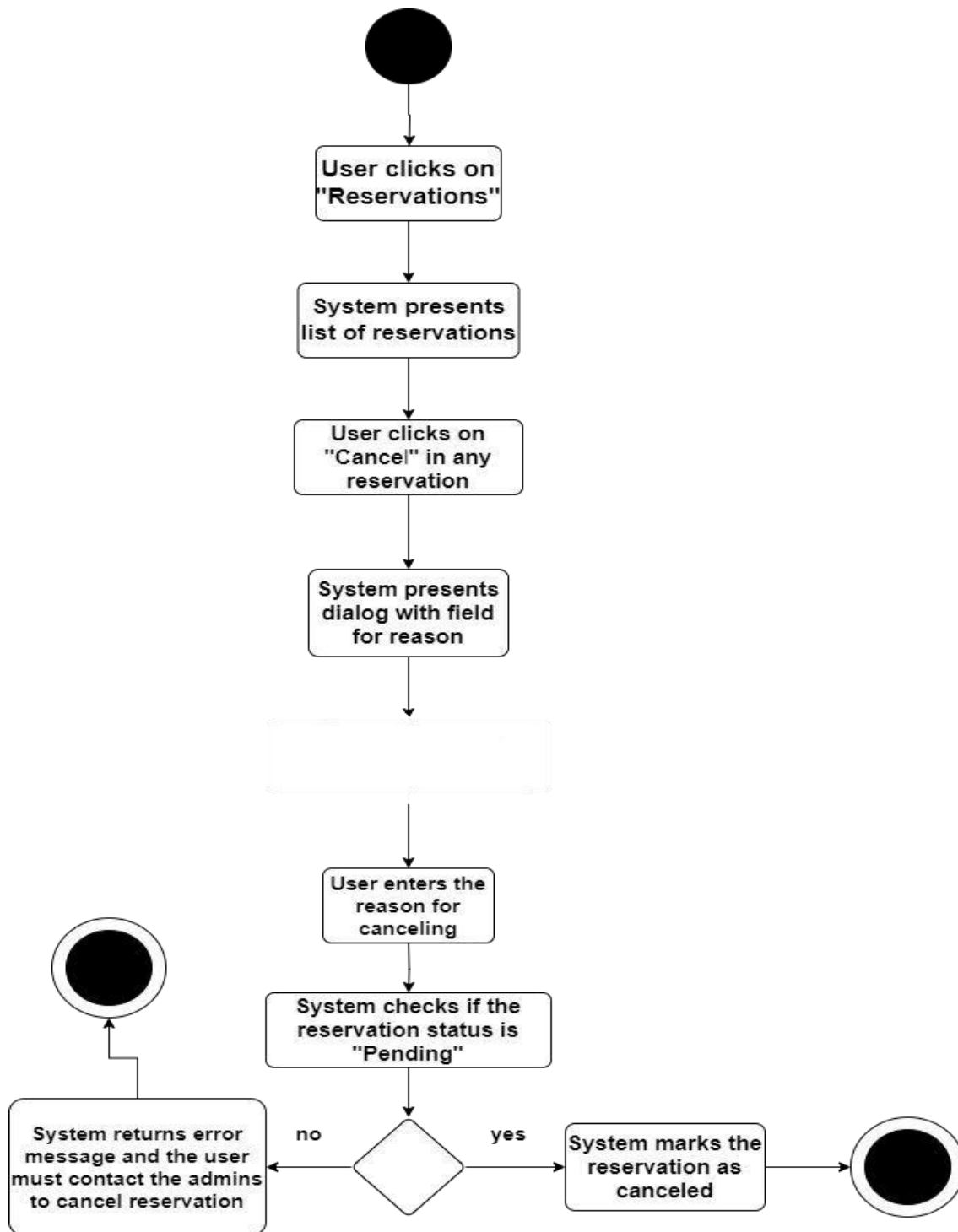
**Activity diagram(Update Reservation) – Figure 24**



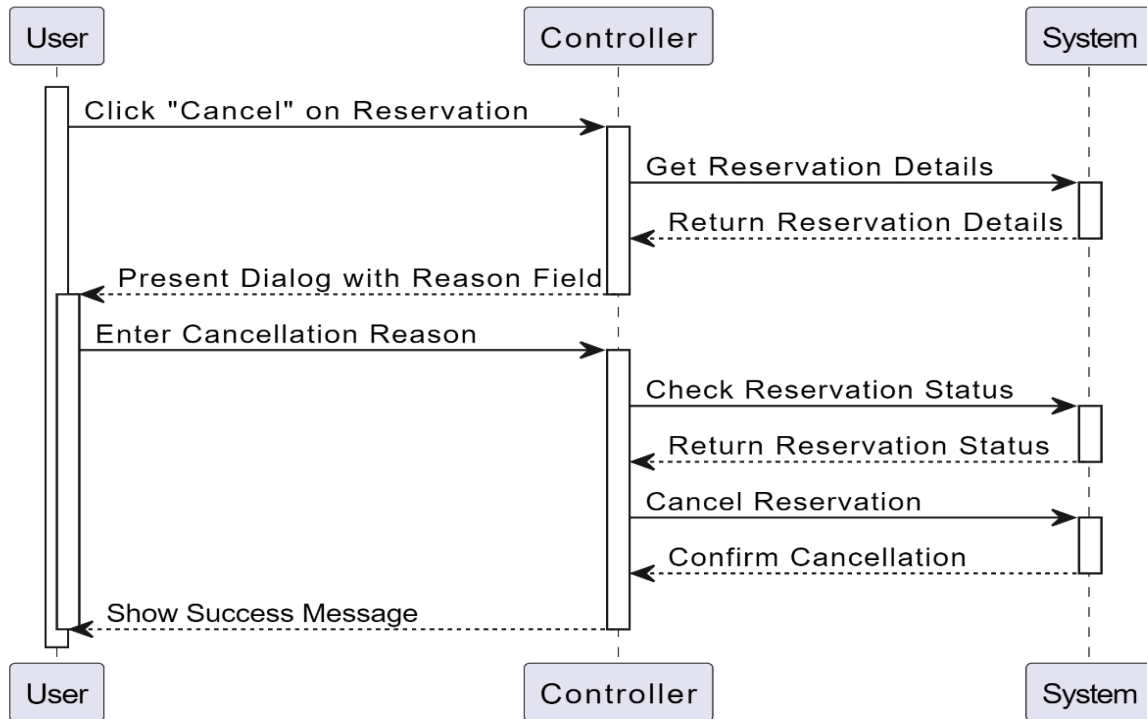
**Sequence diagram(Update Reservation) – Figure 25**

### 5.1.12 Cancel Reservation

<b>Use Case ID</b>	UC-14
<b>Use Case</b>	Cancel Reservation
<b>Actor</b>	User
<b>Pre-Condition</b>	Already logged in, and in view reservations page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Cancel" in any reservation</li><li>2. System present dialog with field reason</li><li>3. User enter the reason of canceling</li><li>4. System checks the reservation status is "Pending"</li><li>5. System marks the reservation as canceled</li></ol>
<b>Alternative Scenario</b>	In step 4 if the status is not "Pending" the system return error message and the user must contact the admins to cancel reservation



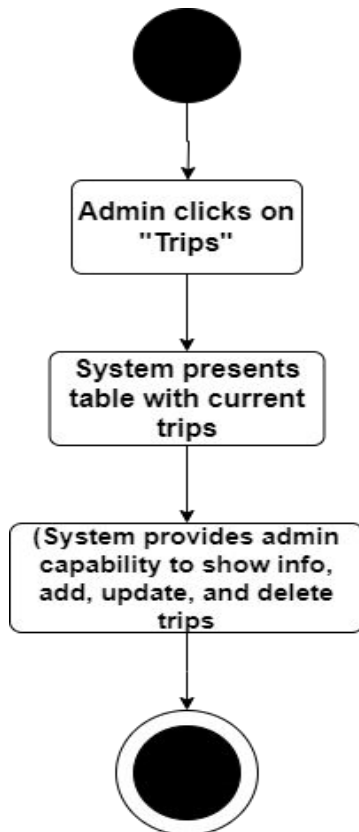
**Activity diagram(Cancel Reservation) – Figure 26**



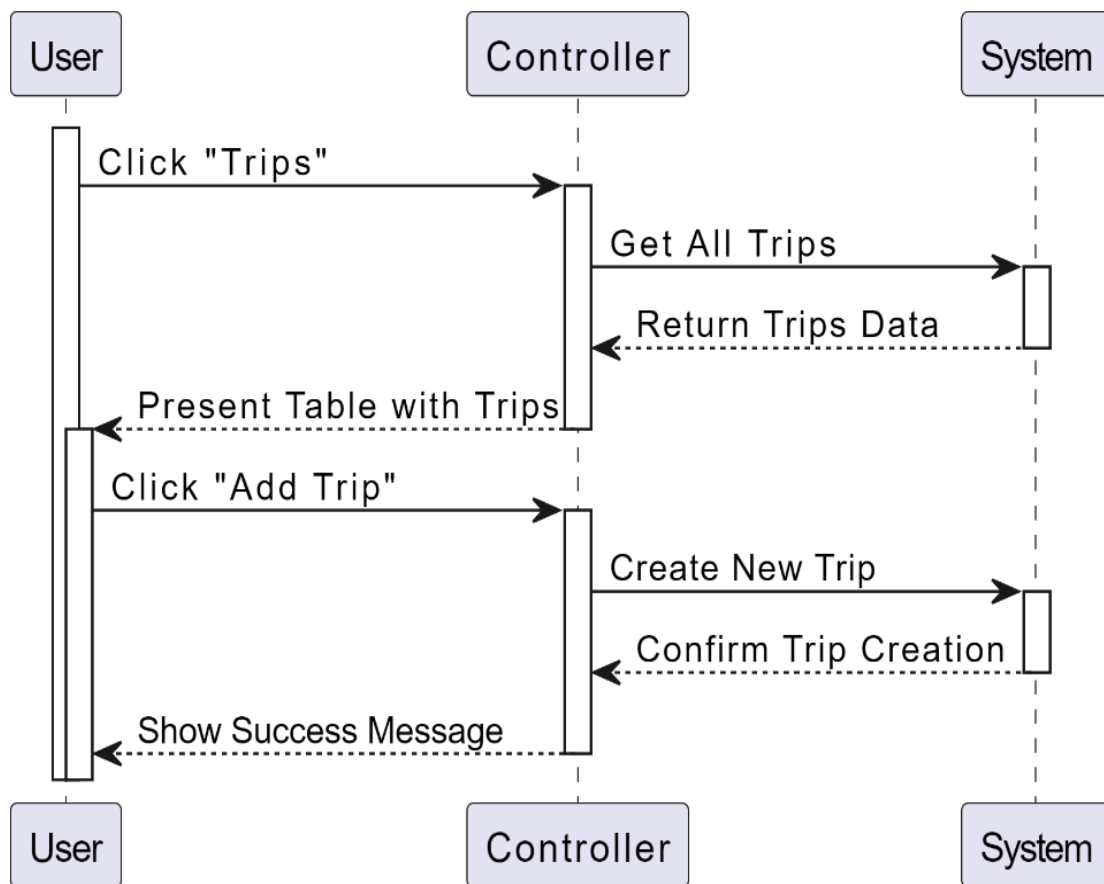
**Sequence diagram(Cancel Reservation) – Figure 27**

### 5.1.13 Manage trips

<b>Use Case ID</b>	UC-15
<b>Use Case</b>	Manage Trips
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Trips"</li><li>2. System Presents table with current tips (name, price, photo, start date, end date)</li><li>3. System provides admin capability to show info, add, update and delete trips</li></ol>
<b>Alternative Scenario</b>	



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

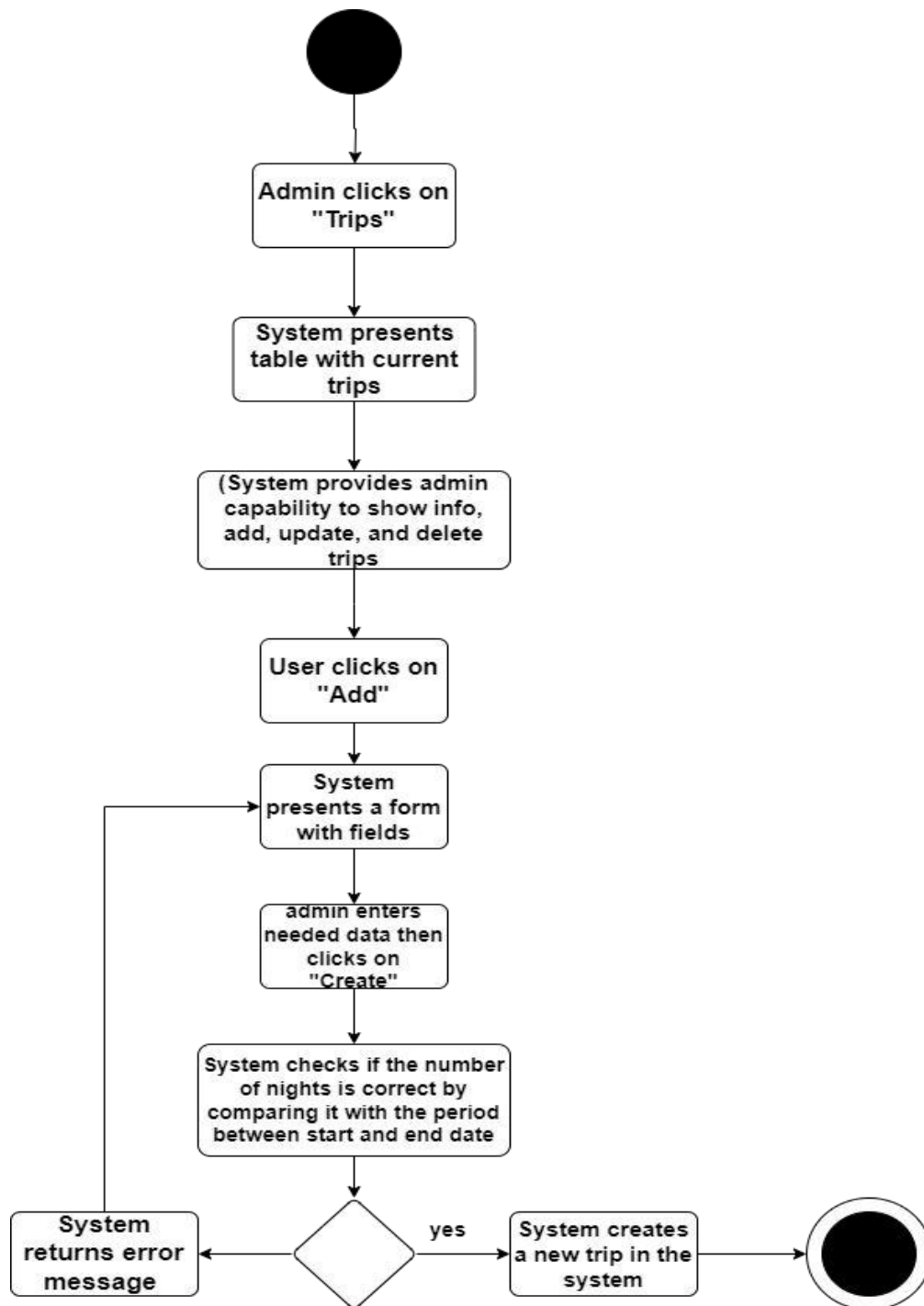


**Sequence diagram(Manage trips) – Figure 29**

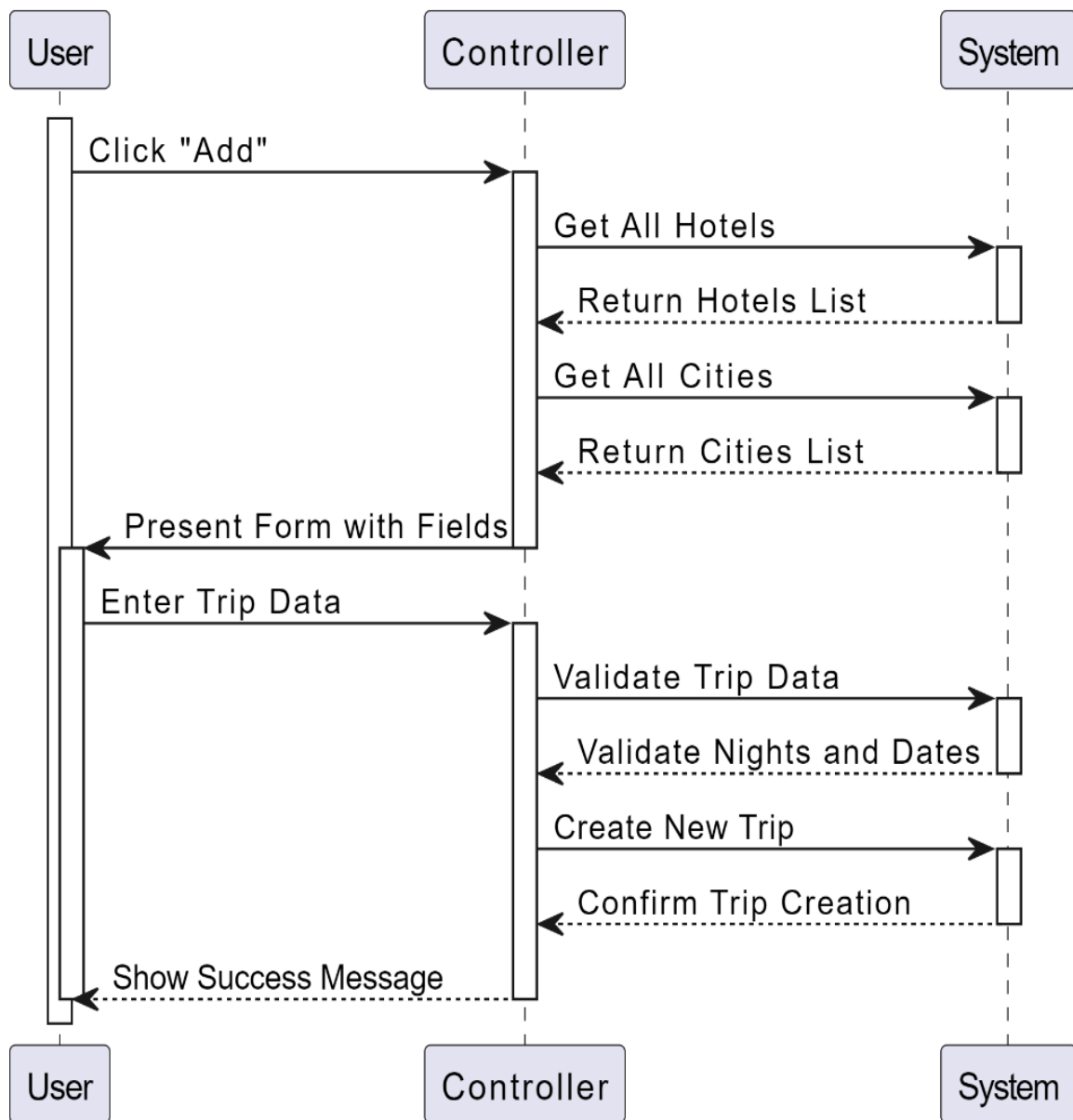
### 5.1.13 Add Trip

<b>Use Case ID</b>	UC-16
<b>Use Case</b>	Add Trip
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, and in manage trips page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Add”</li><li>2. System Presents form 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)</li><li>3. User enter needed data then clicks on “Create”</li><li>4. System checks if the number on nights is correct by comparing it with period between start and end date</li><li>5. System creates new trip in the system</li></ol>
<b>Alternative Scenario</b>	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





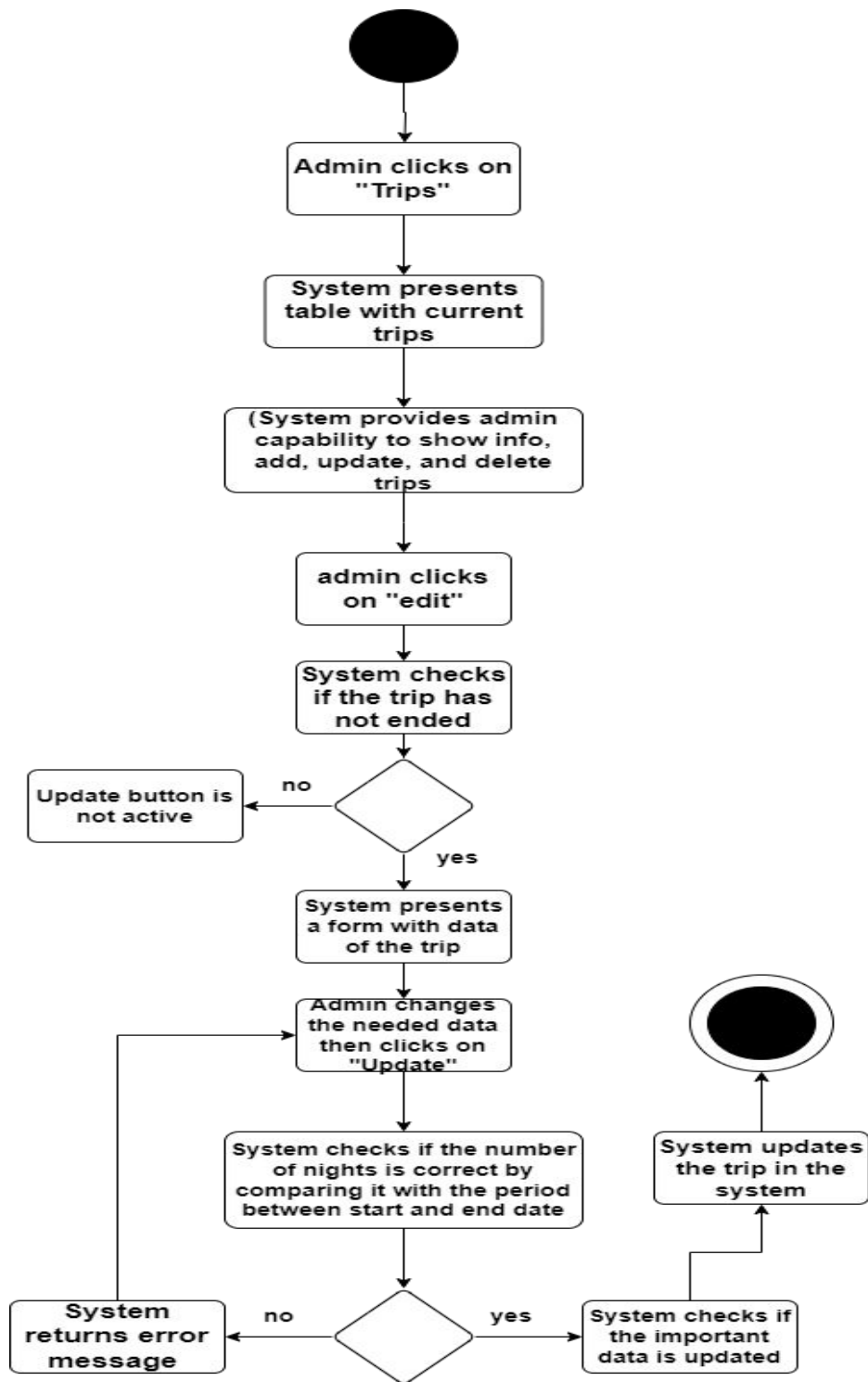
**Activity diagram(Add Trip) – Figure 30**



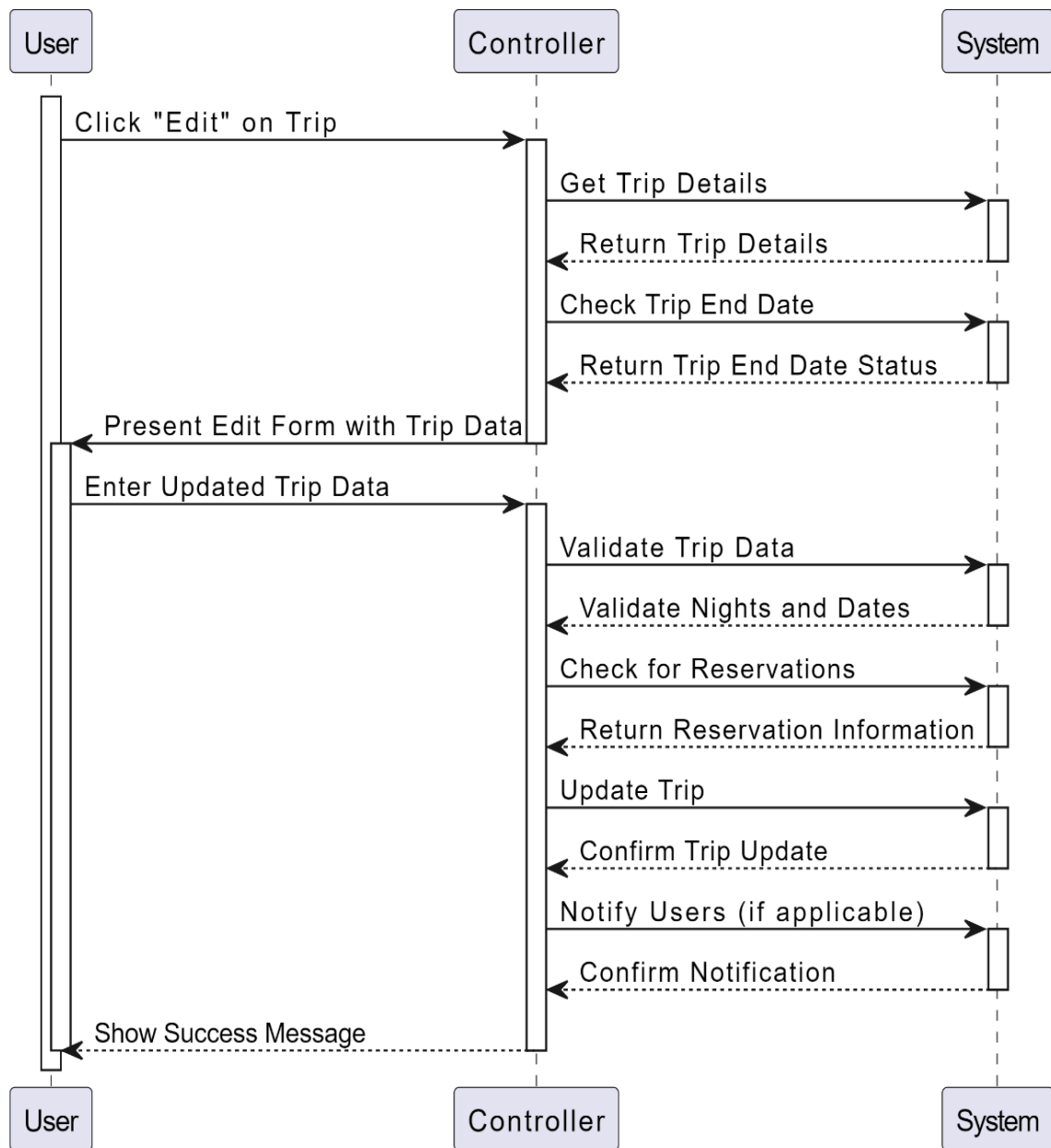
**Sequence diagram(Add Trip) – Figure 31**

#### 5.1.14 Update Trip

<b>Use Case ID</b>	UC-17
<b>Use Case</b>	Update Trip
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, and in manage trips page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Edit" in any trip</li><li>2. System checks if the trip does not end (start date before today)</li><li>3. System Presents form with data of trip (presented before in add)</li><li>4. User change needed data then clicks on "Update"</li><li>5. System checks if the number on nights is correct by comparing it with period between start and end date</li><li>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</li><li>7. System update trip in the system</li></ol>
<b>Alternative Scenario</b>	In step 2 if the trip ends the update button is not active



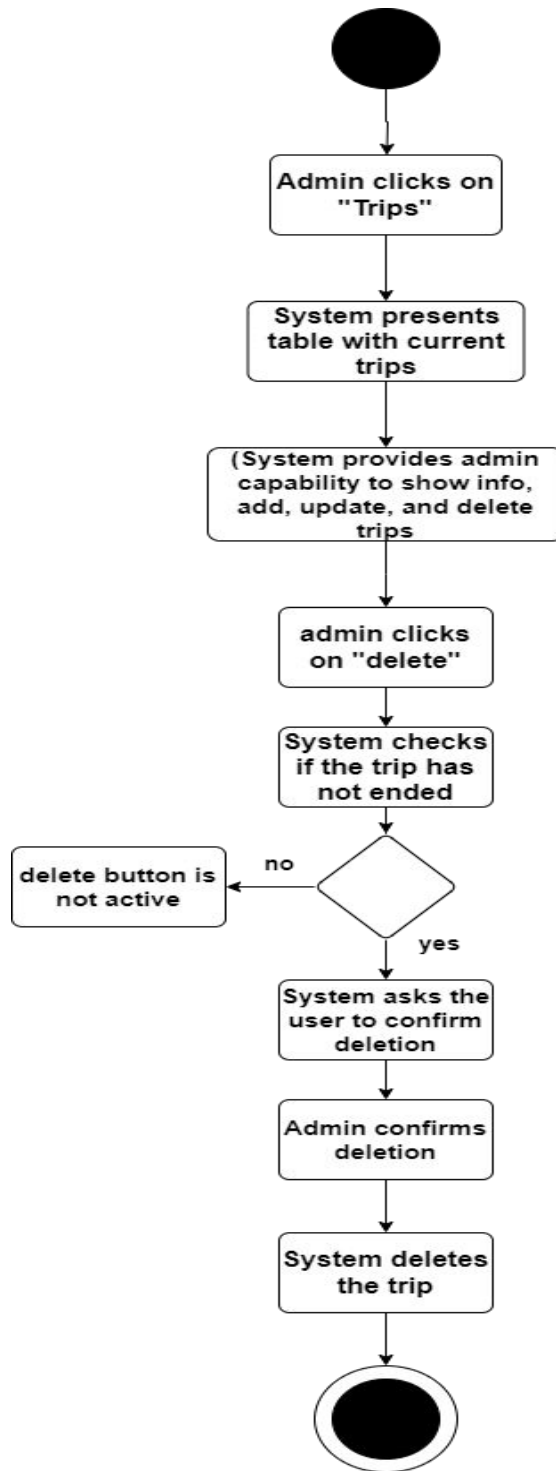
**Activity diagram(Update Trip) – Figure 32**



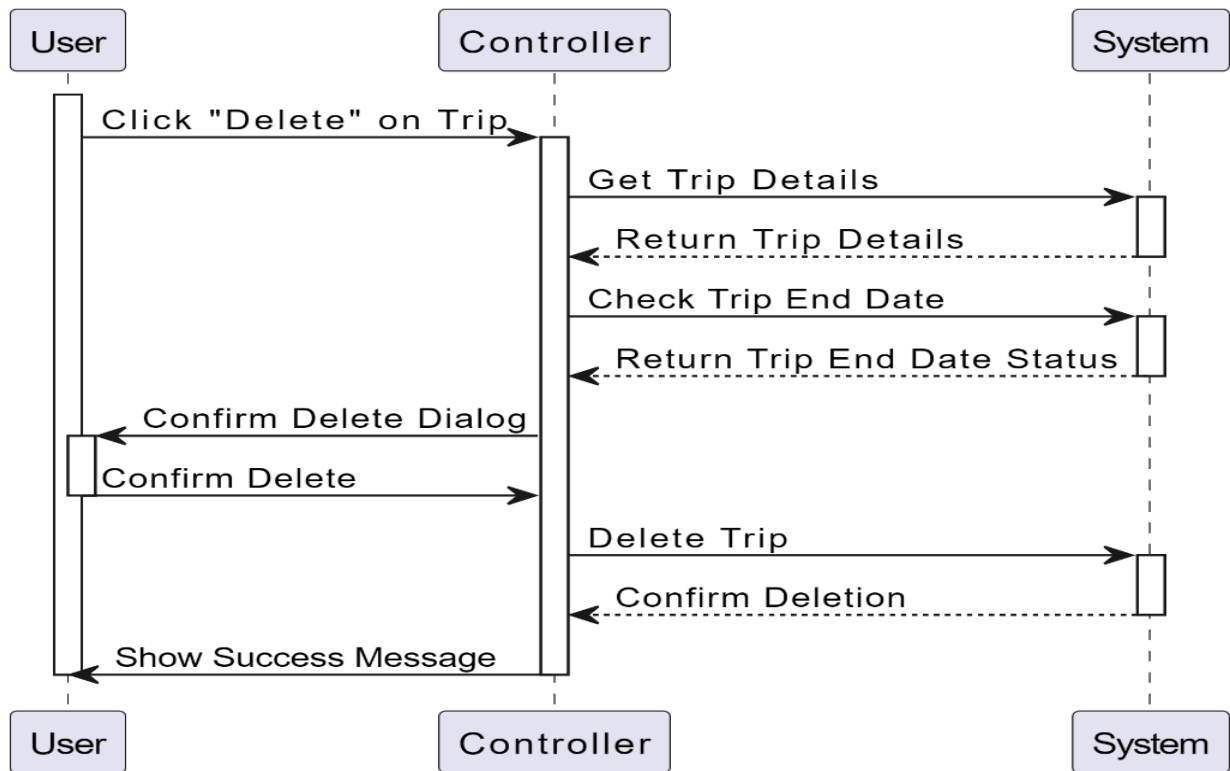
**Sequence diagram(Update Trip) – Figure 33**

### 5.1.15 Delete Trip

<b>Use Case ID</b>	UC-18
<b>Use Case</b>	Delete Trip
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, and in manage trips page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Delete” in any trip</li><li>2. System checks if the trip does not end (start date before today)</li><li>3. System ask user to delete (confirm)</li><li>4. User confirm delete</li><li>5. System delete trip</li></ol>
<b>Alternative Scenario</b>	In step 2 if the trip ends the update button is not active



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

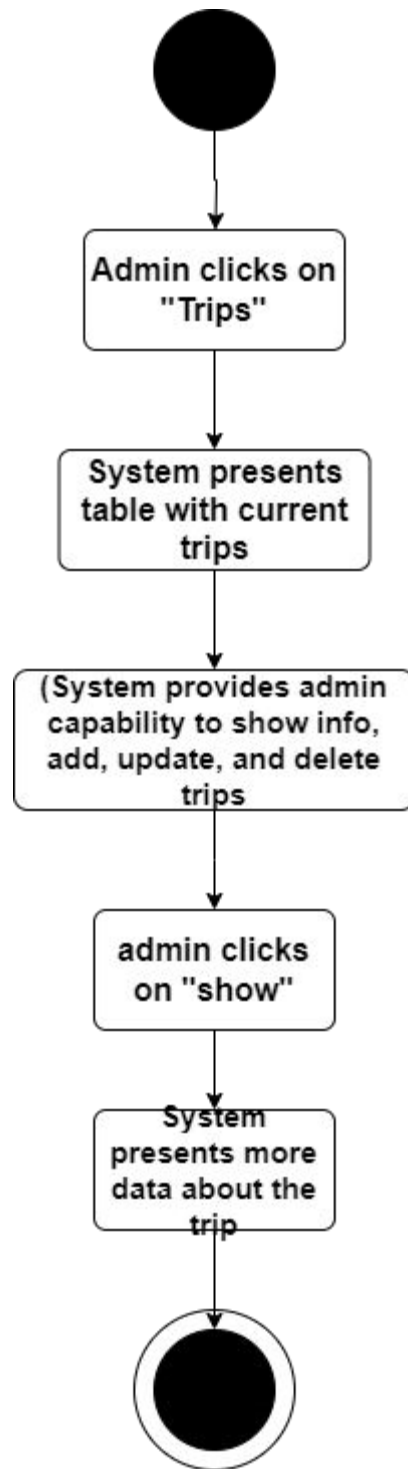


**Sequence diagram(Delete Trip) – Figure 35**



### 5.1.16 View Trip

<b>Use Case ID</b>	UC-19
<b>Use Case</b>	View Trip
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, and in manage trips page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Show” in any trip</li><li>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))</li></ol>
<b>Alternative Scenario</b>	-

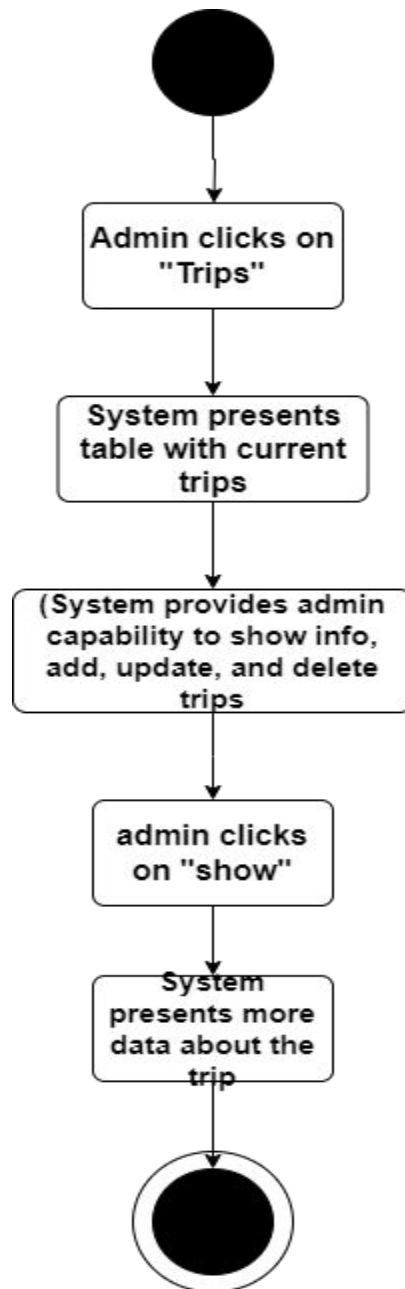


**Activity diagram(View Trip) – Figure 35**

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

### 5.1.17 Manage Cities

<b>Use Case ID</b>	UC-20
<b>Use Case</b>	Manage Cities
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Cities”</li><li>2. System Presents table with current trips</li><li>3. System provides admin capability to show info, add, update and delete cities</li></ol>
<b>Alternative Scenario</b>	-

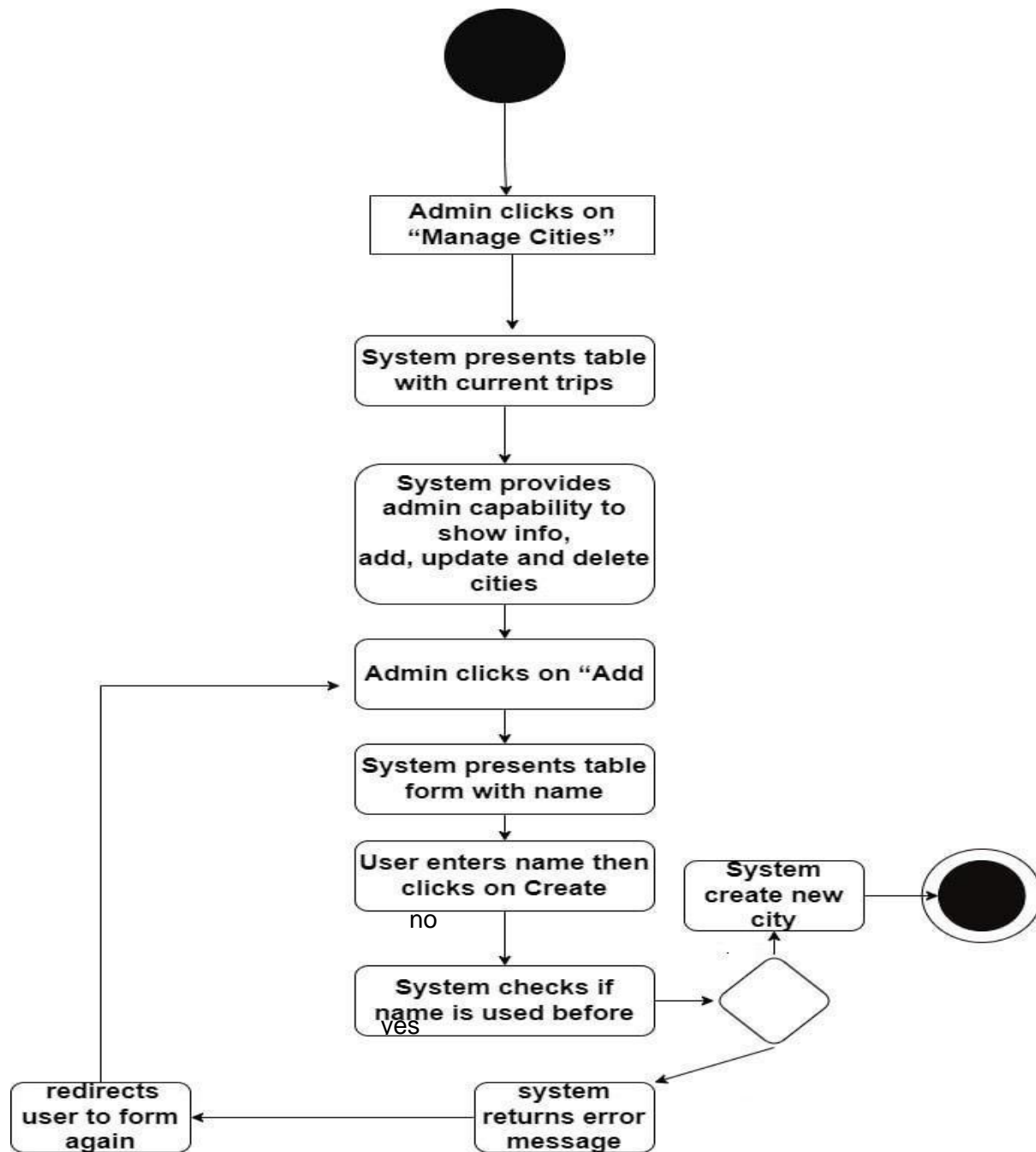


**Activity diagram(Manage Cities) – Figure 37**

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

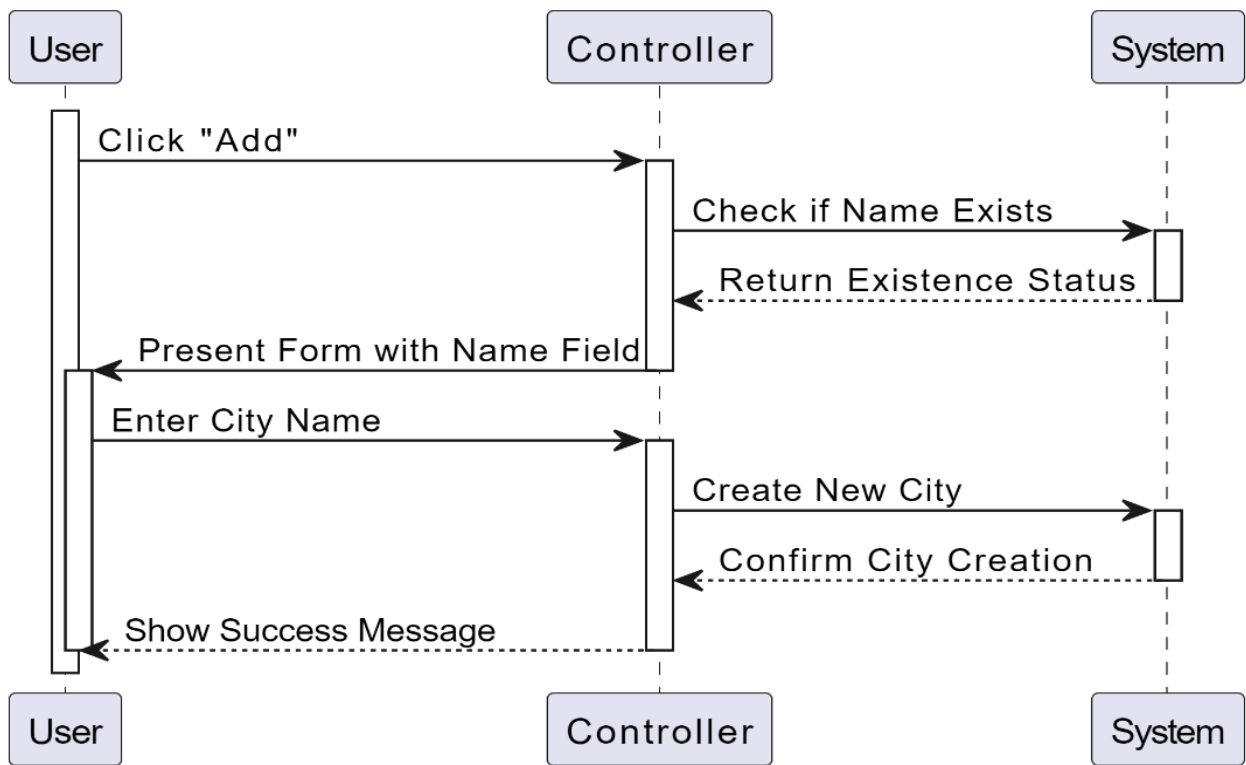
### 5.1.18 Add City

<b>Use Case ID</b>	UC-21
<b>Use Case</b>	Add City
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, User in Manage Cities page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Add”</li><li>2. System Presents table form with name field</li><li>3. User enter name then clicks on “Create”</li><li>4. System checks if name is used before</li><li>5. System creates new city</li></ol>
<b>Alternative Scenario</b>	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**

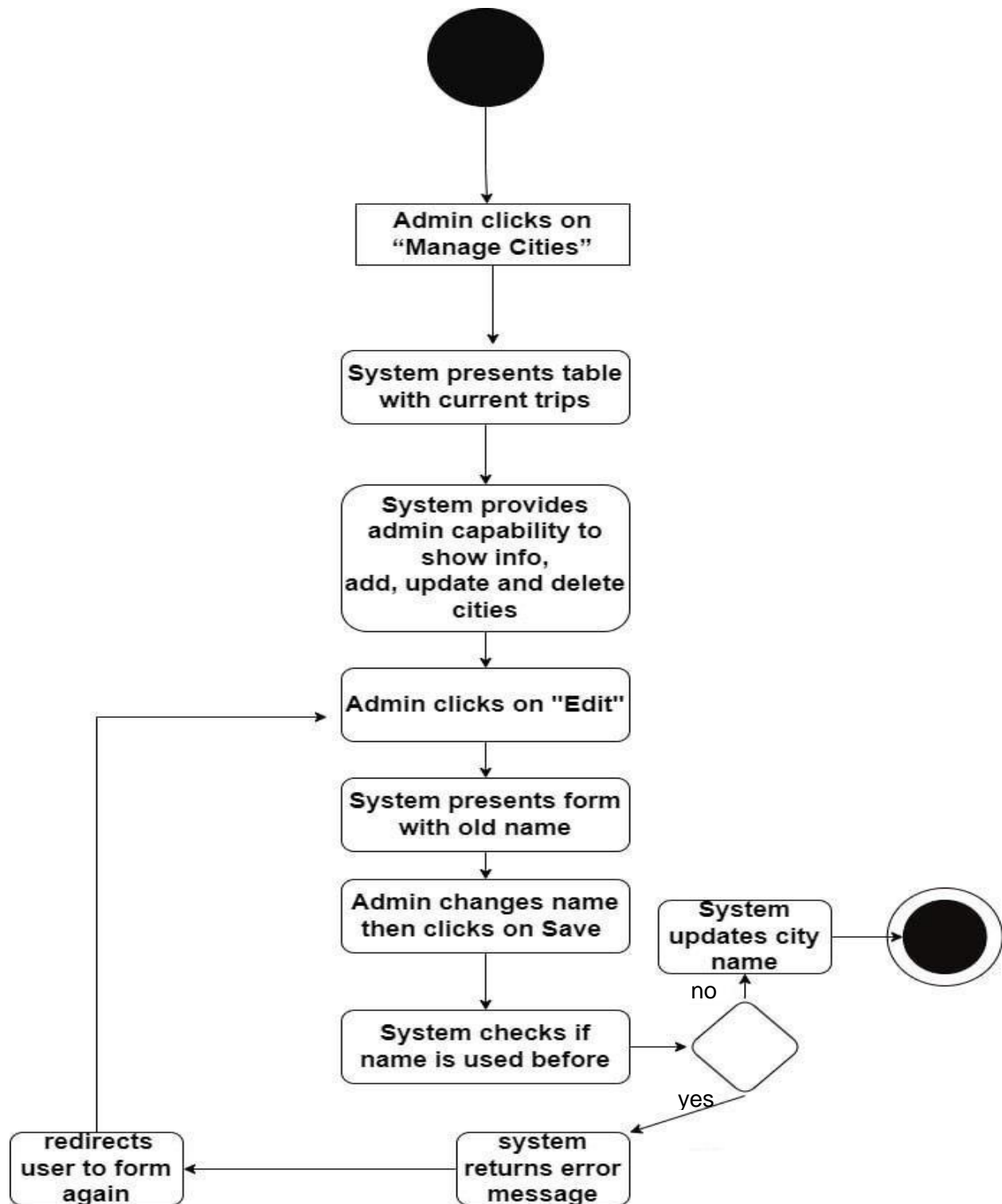




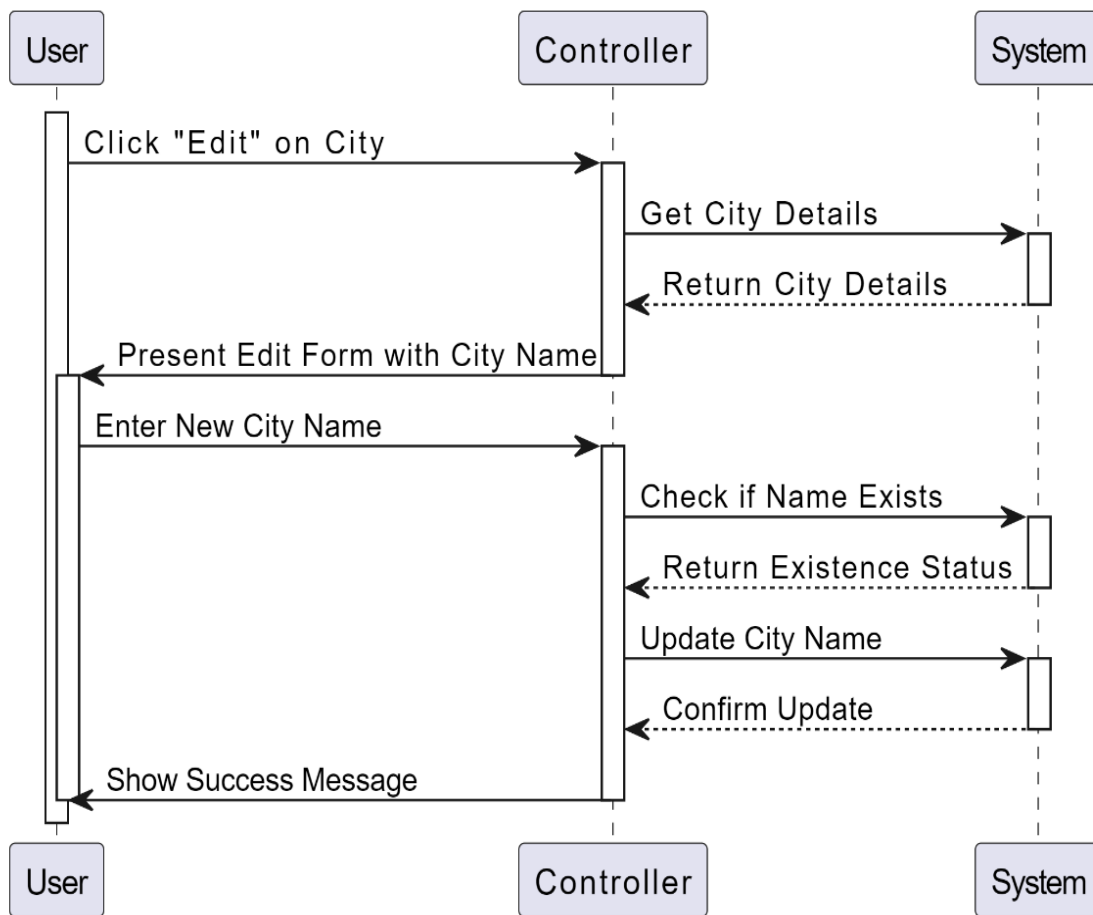
**Sequence diagram(Add City) – Figure 40**

### 5.1.19 Update City

<b>Use Case ID</b>	UC-22
<b>Use Case</b>	Update City
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, User in Manage Cities page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on "Edit" in any city</li><li>2. System presents form with old name</li><li>3. User change name then clicks on "Save"</li><li>4. System checks if name is used before</li><li>5. System update city name</li></ol>
<b>Alternative Scenario</b>	In step 4 if the name is not unique system return error message and redirect user to form again



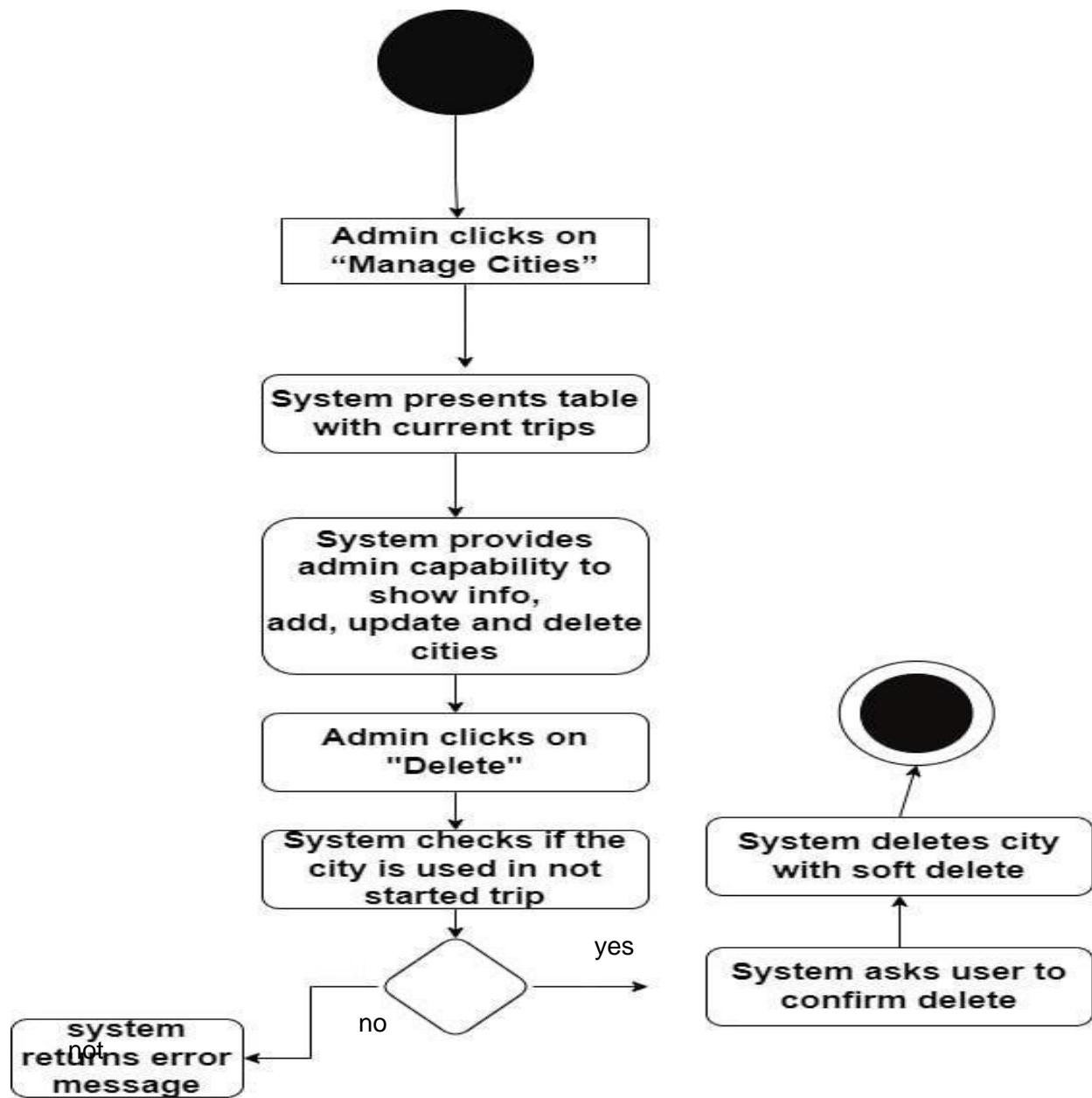
**Activity diagram(Update City) – Figure 41**



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

### 5.1.20 Delete City

<b>Use Case ID</b>	UC-23
<b>Use Case</b>	Delete City
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, User in Manage Cities page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Delete” in any city</li><li>2. System checks if the city does not use in not started trips</li><li>3. System ask user to delete (confirm)</li><li>4. User confirm delete</li><li>5. System deletes city (soft delete)</li></ol>
<b>Alternative Scenario</b>	In step 2 if the city used in trips not started yet, system return error message



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

### 5.1.21 Manage Hotels

<b>Use Case ID</b>	UC-24
<b>Use Case</b>	Manage Hotels
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Hotel”</li><li>2. System Presents table with current hotels (name, location (as map), room description, photos)</li><li>3. System provides admin capability to show info, add, update and delete hotels</li></ol>
<b>Alternative Scenario</b>	-

**Activity diagram(Manage Hotels) – Figure 44**



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

### 5.1.22 Add Hotels

<b>Use Case ID</b>	UC-25
<b>Use Case</b>	Add Hotel
<b>Actor</b>	Admin
<b>Pre-Condition</b>	Already logged in, User in Manage Hotels page
<b>Main Scenario</b>	<ol style="list-style-type: none"><li>1. User clicks on “Add”</li><li>2. System presents form with (name, location (as map), room description, photos)</li><li>3. User enter date then clicks on “Create”</li><li>4. System checks if name is used before</li><li>5. System creates new hotel</li></ol>
<b>Alternative Scenario</b>	In step 4 if the name is not unique system return error message and redirect user to form again