

---

# TRAVEL MATE

---

"Plan, Explore, and Enjoy – All in One App"



Project submitted in the context of the course  
I3301 Software Engineering

Prepared by  
Malk Neyef Serhan 108933  
Hussein Mohamed El Mazbough 105715

LEBANESE UNIVERSITY  
FACULTY OF SCIENCES I  
DEPARTMENT OF COMPUTER SCIENCES  
2024 - 2025

## Abstract

“Discover the world with **Travel Mate** – your ultimate travel companion! Whether you're exploring bustling cities or hidden gems, **Travel Mate** ensures your journey is seamless and unforgettable. We bring everything you need right to your fingertips. Our project operates within the **travel and tourism** domain. It focuses on creating a comprehensive travel app designed to help tourists plan and enjoy their trips effortlessly. The app offers a range of services such as hotel, taxi and car rental searches, event discovery, and information on tourist attractions, tailored to the user's location. The goal is to provide a seamless, stress-free experience for travelers by offering everything they need in one place. Say goodbye to travel stress and hello to your next adventure, because with **Travel Mate**, every trip is a perfect trip!”

### Major Steps in the Project:

First, we choose the topic, introduce it and gather all requirements features. Then we implement and design the diagrams starting from use case to state diagram. Finally, we gather our work, discuss and review all topics.

## Table of Contents

|   |    |
|---|----|
| Abstract .....  | 2  |
| Table of Contents .....   | 3  |
| Chapter I - Introduction.....   | 6  |
| I.1. The business domain .....  | 6  |
| I.2. About the modeled application (Purpose, Users Persona and Product Perspective)....                           | 7  |
| I.3. Analysis of the Existing Similar Apps .....  | 7  |
| I.4. Plan of the document .....   | 7  |
| Chapter II - Requirement Gathering, Analysis, and Specification (extension of SRS document provided by IEEE)..... | 8  |
| II.1. Introduction .....  | 8  |
| 1.1. Purpose .....  | 8  |
| 1.2. Document Conventions .....   | 8  |
| 1.3. Intended Audience and Reading Suggestions .....  | 8  |
| 1.4. Product Scope .....  | 9  |
| II.2. Overall Description.....  | 9  |
| 2.1. Product Perspective .....  | 9  |
| 2.2. Product Functions .....  | 10 |
| 2.3. User Classes and Characteristics .....   | 11 |
| 2.4. Operating Environment .....  | 13 |
| 2.5. Design and Implementation Constraints.....   | 13 |
| 2.6. User Documentation .....   | 14 |
| 2.7. Assumptions and Dependencies .....   | 14 |
| II.3. External Interface Requirements .....   | 14 |

|               |                                       |    |
|---------------|---------------------------------------|----|
| 3.1.          | User Interfaces .....                 | 14 |
| 3.2.          | Hardware Interfaces.....              | 14 |
| 3.3.          | Software Interfaces .....             | 14 |
| 3.4.          | Communications Interfaces .....       | 15 |
| II.4.         | Requirements Gathering .....          | 15 |
| II.5.         | System Features .....                 | 18 |
| 5.1.          | Sign in and log in.....               | 18 |
| 5.2.          | Managing Activity .....               | 19 |
| 5.3.          | Hotel and Accommodation Booking ..... | 19 |
| 5.4.          | Taxi and Car Rentals .....            | 20 |
| 5.5.          | Event Listings .....                  | 20 |
| II.6.         | Other Nonfunctional Requirements..... | 21 |
| 6.1.          | Performance Requirements.....         | 21 |
| 6.2.          | Safety Requirements.....              | 21 |
| 6.3.          | Security Requirements.....            | 22 |
| 6.4.          | Software Quality Attributes.....      | 22 |
| 6.5.          | Business Rules.....                   | 23 |
| II.7.         | Other Requirements .....              | 23 |
| II.8.         | Requirements Analysis .....           | 23 |
| 8.1.          | Use Cases.....                        | 28 |
| 8.2.          | Use Cases Textual Description.....    | 30 |
| II.9.         | High-Level Design Specification ..... | 31 |
| II.10.        | Conclusion .....                      | 32 |
| Chapter III - | Application Conception .....          | 34 |
| III.1.        | Introduction .....                    | 34 |

|              |                            |    |
|--------------|----------------------------|----|
| III.2.       | Database.....              | 34 |
| III.3.       | UML Class Diagram.....     | 34 |
| III.4.       | Sequence Diagrams .....    | 35 |
| III.5.       | State Chart Diagram .....  | 36 |
| III.6.       | Conclusion.....            | 37 |
| Chapter IV - | Conclusion .....           | 38 |
| IV.1.        | Future Considerations..... | 38 |
|              | Bibliography .....         | 39 |

## Chapter I - Introduction

This chapter introduces the **Travel Mate** app, designed to help tourists efficiently plan and manage their trips. It highlights the growing travel and tourism industry and the need for an all-in-one solution that integrates services such as accommodations, car rentals, taxi services, events, and tourist attractions.

### I.1. The business domain

The travel and tourism industry is a major global sector, with increasing demand for integrated travel apps as more travelers use smartphones to plan their trips. However, many face the challenge of using multiple apps for different services, leading to a disjointed experience. This app simplifies the planning process, making travel more enjoyable and stress-free. As the industry continues to grow, **Travel Mate** has great potential to meet the demand for all-in-one travel solutions.



Figure 1 Travel and Tourism Industry Growth Statistics

## I.2. About the modeled application (Purpose, Users Persona and Product Perspective)

### I.3. Analysis of the Existing Similar Apps

There are several existing applications that offer some of the features needed for the Travel Mate app, such as booking accommodations, finding transportation options, and providing information on events and attractions. However, none of them provide an all-in-one solution that consolidates all of these features in a user-friendly interface.

|             | Accommodation Booking | Taxi/Ride Services | Local Events & Activities | Tourist Attractions |
|-------------|-----------------------|--------------------|---------------------------|---------------------|
| Booking.com | X                     |                    |                           |                     |
| Uber        |                       | X                  |                           |                     |
| TripAdvisor |                       |                    | X                         | X                   |
| Airbnb      | X                     |                    |                           |                     |
| Eventbrite  |                       |                    | X                         |                     |

Table 1 Comparison of Existing Similar Application

Agile is about creating a flexible, collaborative, and adaptive environment that prioritizes customer satisfaction and high-quality delivery. By breaking down large projects into manageable iterations, Agile ensures that teams can respond quickly to changing needs and continually improve the product based on real-world feedback.

### I.4. Plan of the document

In this document, we propose a specification, design, and project planning for the **Travel Mate** application. We start by showing an introduction to the application, outlining its business domain and the specific needs it aims to address.

Next, we analyze existing similar applications in the market to highlight the gaps and opportunities that Travel Mate aims to fill.

Then it comes to an important step which is requirements gathering, functional and non-functional features. After that we do the high-level specifications and diagrams.

## Chapter II - Requirement Gathering, Analysis, and Specification (extension of SRS document provided by IEEE)

### II.1. Introduction

#### 1.1. Purpose

**Travel Mate** application, a travel planning tool that helps users organize their trips. This document outlines the software requirements for version 1.0 of the application. The scope of the product covered includes key functionalities such as accommodation searches (hotels), transportation options (taxis and car rentals), and real-time information on local events and tourist attractions. Additionally, the app includes a payment method, allowing users to securely book and pay for services within the app.

#### 1.2. Document Conventions

This Software Requirements Specification (SRS) follows standard documentation conventions to ensure clarity and consistency. The key typographical conventions used throughout this document are:

- **Bold** text is used to highlight important terms, headings, and key features.
- Requirements are numbered sequentially
- **Bullet points** are used for listing unordered items or features.

#### 1.3. Intended Audience and Reading Suggestions

This document is intended for multiple audiences, each with different needs and focuses:

- **Developers:** To understand the technical requirements and build the application according to the specifications.
- **Project Managers:** To ensure that project milestones align with the requirements and deliverables.



- **Marketing Staff:** To better understand the features and selling points of the app for promotional purposes.
- **Testers:** To create test plans based on the specified requirements and ensure product quality.
- **Documentation Writers:** To refer to this document while creating user manuals or help guides.

The document is organized as follows:

1. **Overview Sections** (such as Purpose, Scope...) should be read first to understand the general objectives and features of the application.
2. **Functional Requirements** and **Non-Functional Requirements** sections are important for developers and testers to implement and verify the functionality.
3. **System Specifications** are primarily for developers and testers.

#### 1.4. Product Scope

This application provides a one-stop platform for travelers, combining all essential services into a single app. This includes searching for accommodations, booking transportation, discovering local events, and providing real-time tourist information. The goal of the app is to enhance the travel experience by offering a seamless solution for planning, booking, and payment. The software aligns with corporate goals to simplify the travel process, and offer secure in-app transactions.

## II.2. Overall Description

### 2.1. Product Perspective

While there are several standalone apps for booking hotels, taxis, or finding events, **Travel Mate** seeks to combine all the essential services into one platform. This eliminates the need for tourists to download multiple apps, visit different websites, and face the challenge of coordinating diverse services. The app's mission is to simplify travel planning, enhance the travel experience, and ensure tourists have all the information they need in one place.

This app is a self-contained product, but it may also interact with several external systems and services.

- **Hotel Booking Service:** For integration with hotel booking platforms, enabling users to make reservations.
- **Maps and Navigation Service:** For providing directions, routes, and maps to the user during their travels.
- **Payment Gateway:** To process payments for booking, hotels, or other services through the app.
- **Currency Conversion API:** To provide real-time currency conversion rates.
- **User Authentication Service:** For logging in, managing accounts, and securing user data.
- **Local Events API:** To display local events, activities, or points of interest near the user's travel destination.
- **Taxi Booking Service:** Integration with taxi platforms (Uber, local taxi services) to book rides directly through the app.

## 2.2. Product Functions

The Travel Mate App must provide several core functions to ensure users have a seamless travel experience. These major functions include:

- **User Account Management:**
  - User registration, login.
  - Password recovery.
- **Hotel Booking:**
  - Search available hotels based on location, budget.
  - Book hotel accommodations and view booking details.

- **Taxi Booking:**
  - Search and book taxi services from the app.
  - Track the location of the booked ride in real-time.
- **Navigation and Maps:**
  - Provide turn-by-turn navigation for travelers.
- **Currency Conversion:**
  - Display real-time currency exchange rates for different countries.
- **Local Events and Activities:**
  - Display a list of local events, attractions, and activities in the user's destination.
- **Payment Processing:**
  - Process payments for hotel, taxi bookings, and other services.

### 2.3. User Classes and Characteristics

The **Travel Mate App** is designed for a variety of users, each with different needs and levels of engagement. Below are the users that have an Interaction with the App, their characteristics, and specific requirements for each:

#### 1. Travelers (primary)

- **Characteristics:** main users for this app.
- **Requirements:**
  - Search and book hotels, taxis, and car rentals.
  - Discover nearby attractions and events.

## 2. Hotels

- **Characteristics:** Hotels that list their available rooms for booking.
- **Requirements:**
  - Manage room availability, pricing, and booking requests.
  - Receive and respond to customer reservations.
  - Provide detailed descriptions and images of accommodations.

## 3. Taxi Services

- **Characteristics:** Taxi providers offering rides to travelers.
- **Requirements:**
  - List available taxis and service areas.
  - Accept ride requests and provide estimated fares.
  - Manage driver availability and traveler bookings.

## 4. Car Rental Services

- **Characteristics:** Businesses or individuals offering vehicles for rent.
- **Requirements:**
  - List available cars and rental rates.
  - Manage bookings and track vehicle availability.
  - Provide details on pick-up and drop-off locations.

## 5. Event Organizers & Admins (Support & Content Providers)

- **Characteristics:** Admins responsible for managing and updating events and attractions.

- **Requirements:**

- Add, update, and manage event listings.
- Provide details about tourist attractions and entertainment options.
- Ensure accurate and up-to-date information is available to travelers.

## 2.4. Operating Environment

The **Travel Mate** can be installed as an application or visited as a website. It will operate on modern web browsers, including **Google Chrome, Safari, Firefox, and Microsoft Edge**, supporting both **desktop** and **mobile** platforms. It will be compatible with **Windows, macOS, Linux**, and **mobile operating systems** like **iOS** and **Android**.

As an application, in case of no network connectivity, travelers will still be able to access their current accommodations and scheduled activities, ensuring they can view their plans offline. However, booking new services, making payments, or accessing real-time updates will require an active internet connection.

The website will rely on **APIs** for services like **maps, booking systems, payment gateways**, and **event information**, requiring a stable **internet connection**. It will be hosted on **cloud platforms** for scalability and performance, with secured data.

## 2.5. Design and Implementation Constraints

The development of this app will be influenced by several constraints. These include compliance with privacy laws and regulations, such as data protection requirements. The app must also be optimized to run efficiently on a variety of devices with different performance capabilities. Integration with third-party services for bookings and payments will be necessary, and the app should handle multiple users and requests simultaneously. Security will be a key focus, with measures to protect user data and transactions. Additionally, the app must support multiple languages and ensure smooth communication between systems.

## 2.6. User Documentation

The app will be equipped with a range of user-friendly documentation to ensure a smooth and enjoyable experience. Users will have access to a comprehensive **User Manual** that covers all app features, along with easy-to-follow **Interactive Tutorials** for first-time users. In-app **On-line Help**, our team is always ready to provide quick solutions to common issues.

## 2.7. Assumptions and Dependencies

The app depends on few things, like third-party services for bookings, payments, and transportation, as well as users having a stable internet connection for active services. We also rely on cloud hosting and payment gateways, and the app needs users to have compatible devices. If any of these things change, it could impact how the app works or cause delays.

# II.3. External Interface Requirements

## 3.1. User Interfaces

The system is using to implement user interfaces:

As frontend we use **HTML**, **CSS** and **JavaScript**.

As backend we use **PHP** and **MySQL**.

## 3.2. Hardware Interfaces

As hardware interfaces we use:

- Windows
- Any browser that supports **HTML**, **CSS**, **PHP** and **JS**

## 3.3. Software Interfaces

- **Connections:** Frontend (HTML, CSS, JS) communicates with backend (PHP), which connects to the MySQL database.

- **Data Flow:** User input goes to backend, and the backend sends results to the frontend.
- **Purpose:** PHP processes data, MySQL stores it, and frontend displays it.
- **Communication:** Data exchanged via HTTP requests.
- **Data Sharing:** User and booking data shared between frontend and backend.

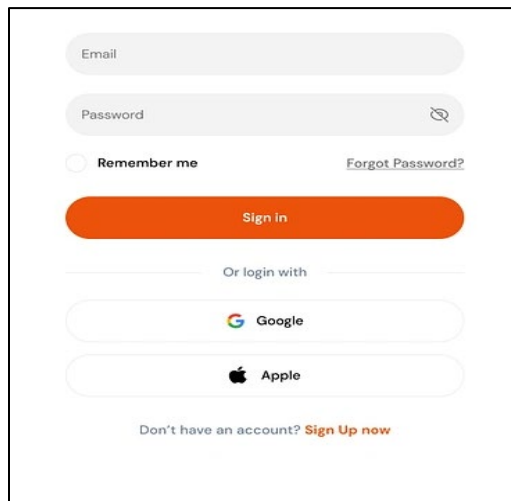
### 3.4. Communications Interfaces

The **Travel Mate** app will use PHP for backend, JavaScript for frontend, and HTTP/HTTPS for secure communication. It supports all types of web browsers.

## II.4. Requirements Gathering

To gather these requirements, we used surveys, brainstorming sessions, interviews, and prototype testing. We started by sending surveys to travelers, hotels, taxi services, and car rentals to learn what they need. Then, we talked to Traveler for more details and created early versions of the app to test and improve based on feedback. This will help us make sure the app meets user needs.

Sample prototype:




The image shows a sample prototype of a login form. It features a clean, modern design with a white background and rounded rectangular input fields. The form includes the following elements:

- Email Field:** A light gray rounded rectangle with the placeholder text "Email".
- Password Field:** A light gray rounded rectangle with the placeholder text "Password" and a small eye icon on the right for toggling visibility.
- Remember me:** A radio button followed by the text "Remember me".
- Forgot Password?** A link in blue text located to the right of the "Remember me" checkbox.
- Sign In Button:** A prominent orange rounded rectangle with the text "Sign In" in white.
- Or login with:** A horizontal line with the text "Or login with" centered below it.
- Google Login:** A rounded rectangle containing the Google logo and the text "Google".
- Apple Login:** A rounded rectangle containing the Apple logo and the text "Apple".
- Sign Up Link:** At the bottom, the text "Don't have an account?" is followed by a link "Sign Up now" in orange.


# Create an account


Already have an account? [Log in](#)



☒ I agree to the [Terms & Conditions](#)


Or register with

 Google

 Apple

choose country that you want to travel

Item 1





Welcome to //Country

|                |                |
|----------------|----------------|
| Hotel          | transportation |
| Tourist Places | Events         |

You Activity

Search

|  |  |  |
|--|--|--|
|  |  |  |
|  |  |  |
|  |  |  |

Details about room

Book

Credit Card Details

|                     |   |
|---------------------|---|
| First Name          | Last Name   |
| 1234 1234 1234 1234 | <div>VISA</div> <div>MasterCard</div> <div>AMERICAN EXPRESS</div> |
| MM / YY             | CVC   |
| Lebanon             |   |

Cookie Lover Name

|            |           |
|------------|-----------|
|            |           |
| First Name | Last Name |

E-mail

|                        |     |
|------------------------|-----|
| ex: myname@example.com | Pay |
|------------------------|-----|

## II.5. System Features

This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product.

### 5.1. Sign in and log in

- **Description and Priority:**

Users can securely sign in or log in using email/password or social media accounts(google).

**Priority:** High

- **Stimulus/Response Sequences:**

- Stimulus: User enters valid credentials or selects a social media login.

Response: System validates credentials and redirects to the homepage.

- Stimulus: User enters invalid credentials.

Response: System shows an error message, prompting user to retry.

- Stimulus: User clicks "Log Out".

Response: System logs the user out and redirects to the login screen

- **Functional Requirements:**

- REQ-1: Allow users to sign up (Sign In) with email/password or social media accounts.
- REQ-2: Allow users to log in using email/password or social media accounts.
- REQ-3: Validate credentials and show error for invalid login attempts.
- REQ-4: After successful login, redirect users to the homepage.
- REQ-5: Allow users to log out and end their session.

## 5.2. Managing Activity

- **Description and Priority:**  
Helps travelers manage their bookings and activities.  
**Priority:** Medium
- **Stimulus/Response Sequences:**
  - Stimulus: user selects destination and dates.  
Response: System generates all categories.
  - Stimulus: user selects “Your Activity”.  
Response: System generates all activities.
  - Stimulus: user deletes an activity.  
Response: System removes this activity.
- **Functional Requirements:**
  - REQ-6: Generate services based on dates/destination.
  - REQ-7: Allow users to manage activities.

## 5.3. Hotel and Accommodation Booking

- **Description and Priority:**  
Search and book hotels.  
**Priority:** High
- **Stimulus/Response Sequences:**
  - Stimulus: User enters search criteria.  
Response: System displays available accommodations.
  - Stimulus: User selects a hotel.  
Response: System shows details (pricing, availability).
  - Stimulus: User clicks "Book Now".  
Response: System processes booking and sends confirmation message.
  - Stimulus: User cancels booking.  
Response: System cancels the reservation and notifies the user.
- **Functional Requirements:**
  - REQ-8: Search accommodations by location, price, etc.

- REQ-9: Display real-time availability.
- REQ-10: Allow booking through the app.
- REQ-11: Notify user on successful booking.
- REQ-12: Notify user on successful booking.
- REQ-13: user can cancel booking.

#### 5.4. Taxi and Car Rentals

- **Description and Priority:**

Provides transport options (taxis, rentals).

**Priority:** High

- **Stimulus/Response Sequences:**

- Stimulus: User requests taxi or car rental service.  
Response: System displays available options based on location.
- Stimulus: User selects a car/taxi.  
Response: System processes booking and provides confirmation.
- Stimulus: User cancels booking.  
Response: System cancels the reservation and notifies the user.

- **Functional Requirements:**

- REQ-14: Show available taxis and car rentals.
- REQ-15: Allow booking within the app.
- REQ-16: Notify user on booking status.

#### 5.5. Event Listings

- **Description and Priority:**

Displays local events (parties, concerts).

**Priority:** High

- **Stimulus/Response Sequences:**

- Stimulus: User requests local events.  
Response: System displays events based on location and date.

- Stimulus: User selects an event.

Response: System shows detailed information about the event.

- Stimulus: User clicks "Register" or "Buy Tickets".

Response: System processes registration or purchase and confirms with a notification.

- **Functional Requirements:**

- REQ-17: Display local events.
- REQ-18: Allow event registration or ticket purchase.

## II.6. Other Nonfunctional Requirements

### 6.1. Performance Requirements

These performance requirements are important to ensure that the app remains fast, responsive, and reliable under typical load conditions. They help avoid frustration for users when interacting with the app, especially when traveling and in areas with unreliable internet connections.

- **Response Time:** The app should have a response time of less than **2 seconds** for loading most pages. Complex searches (multi-criteria queries for accommodations, car rentals, etc.) may take up to **5 seconds**, to ensure a smooth user experience.
- **Throughput:** The app should be able to handle **1000 concurrent users** at peak times without noticeable degradation in performance.
- **Data Load Handling:** The system must be able to handle and display large sets of data (500+ hotel listings) efficiently, without causing significant delays or crashes.
- **Offline Availability:** The app must allow users to access some features (such as user's activity) without an internet connection, with data being synchronized when the internet is available again.

### 6.2. Safety Requirements

Ensuring safety is critical to maintain user trust.

- **Data Privacy:** Personal data, including user preferences, booking details, and payment information, must be securely encrypted and never exposed to unauthorized parties.

- **Security of Financial Transactions:** All transactions related to bookings and payments must be secure to prevent unauthorized access to users' financial data.
- **Location Safety:** The app should provide safety warnings for specific locations (e.g, travel or risk warnings related to political instability, natural disasters, or pandemics). Users should be informed of potential safety concerns and how to reduce risks.

### 6.3. Security Requirements

These security measures are essential to protect user privacy

- **Authentication:** Users must authenticate using a secure password and optionally a Two-Factor Authentication method to access their accounts.
- **Data Encryption:** Sensitive data, such as user payment details and other data must be encrypted.
- **Access Control:** The app must ensure that users only have access to the information and actions they are permitted to. For example, admin users should have full access to all data, while other users like taxi and hotel can't access all data.
- **Privacy:** User location data, preferences, and other personally information should only be used with the app's privacy policy, and users must have the ability to request deletion of their data at any time.

### 6.4. Software Quality Attributes

Essential for providing users with a seamless, enjoyable experience.

- **Reliability:** The app must be highly available and accessible to users at all times.
- **Maintainability:** The app's codebase should be modular, well-documented, and follow standard coding practices to facilitate easy updates and fixes. The system should support easy bug tracking and version management.
- **Usability:** The app should have an intuitive and user-friendly interface.
- **Interoperability:** The app should work seamlessly across different platforms (iOS, Android, Web), and integrate well with third-party APIs for booking, maps, updates, etc.

- **Testability:** The app should be easy to test, with unit tests, integration tests, and end-to-end tests implemented for critical features, including payment processes, account management, and bookings.
- **Portability:** The app should be able to function across a variety of devices (smartphones, tablets, desktop browsers).

### 6.5. Business Rules

1. **User Roles and Permissions** admin has full control over app, you must have an account for login and booking.
2. **Booking Policies:** All bookings confirmed via notification after successful payment.
3. **Payment and Currency Handling:** Supports multiple payment methods and real-time currency conversion.
4. **Safety Alerts:** Real-time alerts (e.g, emergencies) based on user location
5. **Notifications:** Users receive booking confirmations and reminders.

## II.7. Other Requirements

The app should store user data, booking information, and event details in a relational database. Data backup must occur regularly to prevent data loss. This app must support multiple languages, with an initial focus on Arabic, English, Spanish, and French. The interface should adjust based on the traveler's location and language preferences. It must also comply with consumer protection laws in different regions, such as refund policies for cancellations.

## II.8. Requirements Analysis

### Requirements Categorization:

#### User Requirements Category:

- **Traveler (Primary User):**
  - User can sign up or log in using email/password or social media accounts.

- User can choose a country destination.
- User can search and book accommodations.
- User can view event listings and tourist attractions.
- User can access current activities.
- User can modify trip details, including cancellations, or book new accommodations.
- **Hotel (Secondary User):**
  - Hotel owners can add and manage their available rooms for booking.
  - Hotel owners can update pricing, availability, and check-in/check-out policies.
- **Car Rental and Taxi Services (Secondary Users):**
  - Car rental companies and taxi service providers can display available vehicles to the user.
  - They can manage availability and pricing.
- **Admin:**
  - Admin can add, remove, or update events and tourist attractions.
  - Admin ensures the system functions correctly and securely.

#### **Functional Requirements Category:**

- **Core Functionalities:**
  - The system shall allow users to log in or sign up through email/password or social media.
  - The system shall allow users to search for hotels, taxis, and car rentals by location and dates.
  - The system shall enable users to view and book available events and attractions.
  - The system shall allow the admin to manage events and tourist attractions.



- **Booking Management:**

- The system shall check availability for accommodations, cars, and taxis before allowing bookings.
- The system shall notify users of successful bookings or any conflicts.

- **System Notifications:**

- The system shall notify users about upcoming events, cancellations, or delays in bookings.

**Business Requirements Category:**

- **Travel Mate** should increase booking efficiency for users, providing a seamless experience for accommodations, rentals, and attractions.
- **Travel Mate** should enhance the revenue of hotels, car rental services, and taxi providers by exposing their services to more tourists.
- The app should drive customer loyalty and satisfaction by offering a one-stop solution for travel needs.

**Implementation/Transition Requirements Category:**

- The app should be designed to scale with increasing numbers of users and service providers.
- The system should be hosted on a cloud platform for flexibility and scalability.
- All transactions should be secure, with encrypted data for payment processing.

**Non-functional Requirements Category:**

- **Performance:**

- The app should load within 2 seconds for the main interface and user searches.

- The app should support up to 1,000 active users simultaneously without performance degradation.
- **Usability:**
  - The app should have an intuitive and user-friendly interface for easy navigation.
  - The system should provide real-time updates for travel-related services (e.g, taxi availability, accommodation booking status).
- **Security:**
  - The app should employ end-to-end encryption for user data, especially for payment information.

### **Criticism of the Requirements:**

- **Overly broad or unclear requirements:** The requirement to "support up to 1,000 active users simultaneously" is ambitious, but may be difficult to implement without specifying the underlying technical constraints. The system's infrastructure needs to be clearly outlined to handle this scale efficiently.
- **Conflicting priorities:** The desire for a "seamless experience" while maintaining secure transaction processing may present conflicts in terms of speed vs. security checks. This needs to be carefully balanced.

### **MOSCOW Categorization:**

#### **M-MUST:**

- It must run on all java applications.
- User must be able to sign up or log in using email/password or social media accounts.
- The system must allow users to search and book accommodations, taxis, and car rentals.
- The system must support booking confirmation and cancellations.
- Admin must be able to add and manage events and attractions.

- The system must provide a secure payment gateway.

#### **S-SHOULD:**

- The system should support real-time updates for bookings.
- Support additional languages to make the app more accessible internationally.

#### **C-COULD:**

- Implement augmented reality features to enhance the tourist experience (view attractions in AR).
- User can modify bookings (cancel or change).

#### **W-WON'T:**

- Native support for other transportation modes (e.g. bike rentals) in the initial release.

#### **Non-Functional Requirements Validation and Verification:**

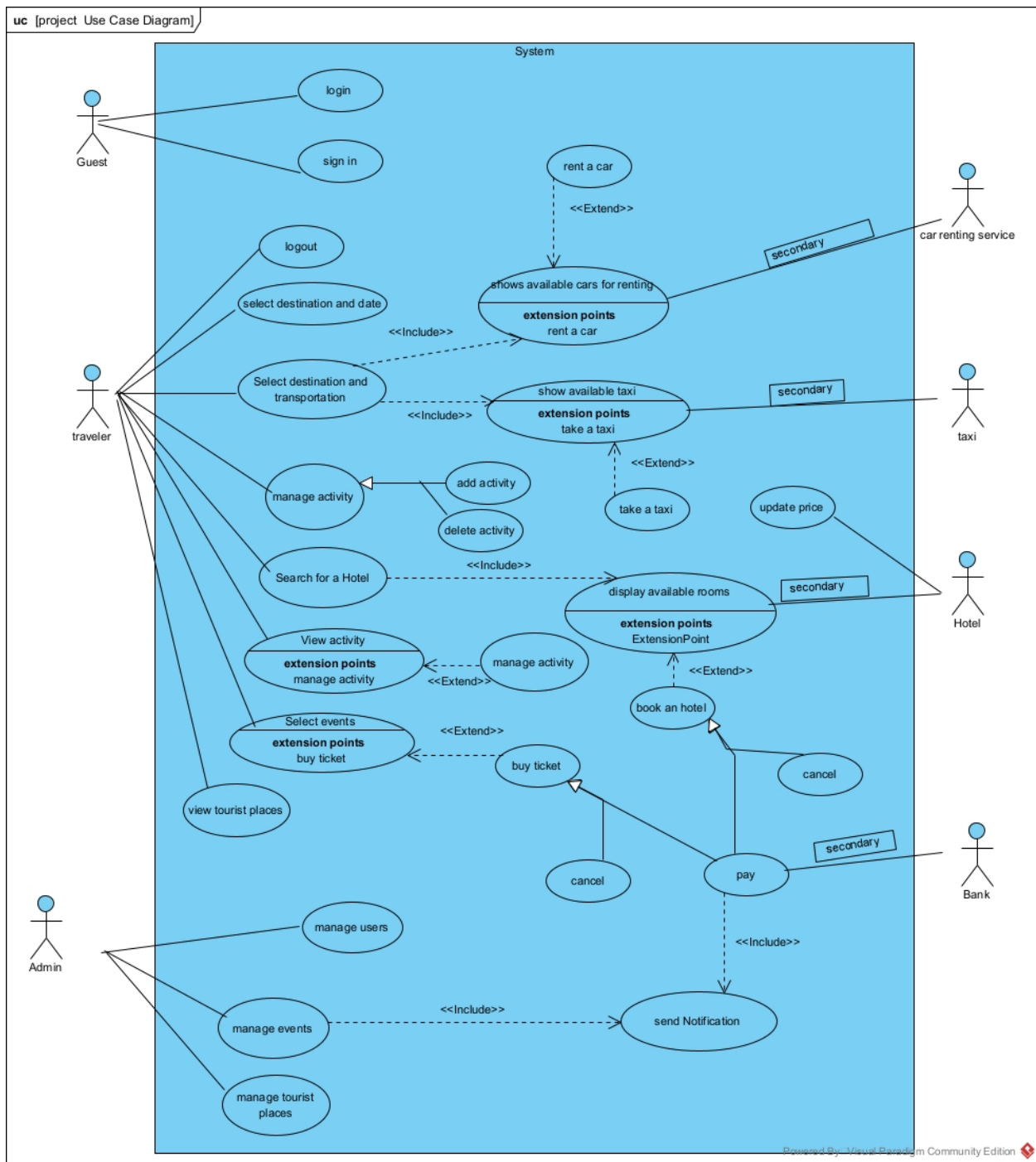
##### **Validation (Are we doing the right things?):**

- During the development process, we will perform user acceptance testing (UAT) to ensure that the app meets the identified non-functional requirements.
- We will test performance under simulated high loads to confirm that the system can handle the expected number of users.

##### **Verification (Are we doing the things right?):**

- Once developed, we will perform stress testing to verify that the app can handle the expected user load without compromising performance.
- We will review security measures such as encryption and transaction safety through external audits.

## 8.1. Use Cases



**Associations:**

- Guests can log in, and sign up.
- Travelers can select a destination, date, and transportation options.
- Renting a car extends to viewing available rental cars.
- Taking a taxi includes showing available taxis, with an extension to book one.
- Hotel search includes displaying available rooms, which extends to booking a hotel.
- Booking a hotel requires payment and includes an option to cancel.
- Activity management allows travelers to add, delete, and view activities.
- Event selection extends to buying tickets.
- Payments are processed by a Bank, ensuring secure transactions.
- Admins manage users, events, and tourist places.
- Send Notification is triggered when an event is added or when a payment is made

## 8.2. Use Cases Textual Description

|                           |  |   |
|---------------------------|--|---|
| <b>Number</b>             | 1  |   |
| <b>Name</b>               | Hotel Booking Payment  |   |
| <b>Summary</b>            | The user pays for a hotel room using a bank payment method.              |   |
| <b>Priority</b>           | 5  |   |
| <b>Preconditions</b>      | The user has selected a hotel and room to book.                          |   |
| <b>Postconditions</b>     | The user successfully completes the payment and receives a confirmation. |   |
| <b>Primary Actor(s)</b>   | Traveler   |   |
| <b>Secondary Actor(s)</b> | Bank system, Hotel system  |   |
| <b>Trigger</b>            | The user has chosen to proceed with payment.                             |   |
| <b>Main Scenario</b>      | <b>Step</b>  | <b>Action</b>   |
|                           | 1  | User initiates the payment process after selecting a hotel room                       |
|                           | 2  | The system displays available payment methods (Credit Card, Bank Transfer, etc.).     |
|                           | 3  | User selects a payment method (e.g., Credit Card).                                    |
|                           | 4  | The system prompts the user to enter payment details.                                 |
|                           | 5  | User enters payment details and confirms the payment.                                 |
|                           | 6  | The system verifies the payment details with the bank.                                |
|                           | 7  | If verification is successful, the system deducts the amount from the user's account. |
|                           | 8  | The system generates a payment confirmation receipt.                                  |
|                           | 9  | The system sends a payment confirmation notification to the user.                     |
|                           | 10   | The system updates the hotel booking status as confirmed.                             |
| <b>Extensions</b>         | <b>Step</b>  | <b>Branching Action</b>   |
|                           | 6a   | The system notifies the user that account fund are insufficient                       |
|                           | 6b   | System gives current account balance and decline the payment.                         |
|                           | 6c   | The system gives the user the option to retry with another method.                    |
| <b>Open Issues</b>        | 1  | Should the system allow partial payments or installments?                             |
|                           | 2  | Should the system store payment details for future use?                               |

Figure 3: Example for a DTD

## II.9. High-Level Design Specification

### Security:

**Travel Mate** integrates robust security measures to protect user data and ensure system integrity:

- User Authentication & Authorization: Secure login via email/password or social media accounts(e.g., google).
- Data Protection: Important information is encrypted to prevent unauthorized access.
- API Security: Secure protocols are used to protect data exchanges.
- Access Control: Only authorized users can access the app.
- Privacy Compliance: The app follows data protection user privacy.

### Hardware:

The system is designed to function on the following hardware platforms:

- Mobile Devices: Supports both Android and iOS smartphones.
- Servers: Uses cloud hosting for scalability and reliability. (aws...)
- Database: Manages data securely with a reliable database system.

### User Interface:

- Intuitive Design: The app layout is simple and easy to navigate.
- Essential Screens: Includes home, search, booking, and payment sections.
- Responsive & Accessible: Works well on different devices for a seamless experience.

**Internal Interfaces:**

- Data Exchange: Uses secure APIs to connect the app with the server.
- Backend System: Built with a reliable framework to handle requests and process data.
- Database Management: Efficiently organizes and retrieves data for smooth performance.

**External Interfaces:**

- Third-party APIs: Integration with services like Booking.com ...
- Payment Gateway: Secure transactions through visa, PayPal, or Apple Pay.
- Mapping Services: Google Maps API for route planning and location-based services.
- Notification Services

**Architecture:**

- Client-Server Model.
- Microservices Architecture: Independent services for authentication, booking, and recommendations.
- Database Design: Uses a relational database. (MySQL)

**Database:**

- Relational Database: MySQL for structured data.
- Core Tables: User, Bookings, Payments, Events, Hotels, Car Rentals, Tourist places user activity.

## **II.10. Conclusion**

This chapter outlined the high-level design specifications of the **Travel Mate** application, covering security measures, hardware and software requirements, user interface, system



architecture, and database structure. The design ensures seamless integration with third-party services while maintaining performance, security, and scalability.

Challenges encountered included defining the scope of services and integrating external APIs. These were addressed through market research, stakeholder feedback, and an agile development approach. With this solid foundation, **Travel Mate** is set to provide a reliable and efficient travel planning experience.

## Chapter III - Application Conception

### III.1. Introduction

In the previous chapters, we have discussed the features that should be offered by our application, focusing on the needs of both travelers and service providers. These features involve key entities such as Traveler, Hotel, Payment, Event, Transportation, and Notification systems. In this chapter, we use UML diagrams to show how these entities are structured, their roles in the application, and how they interact with each other.

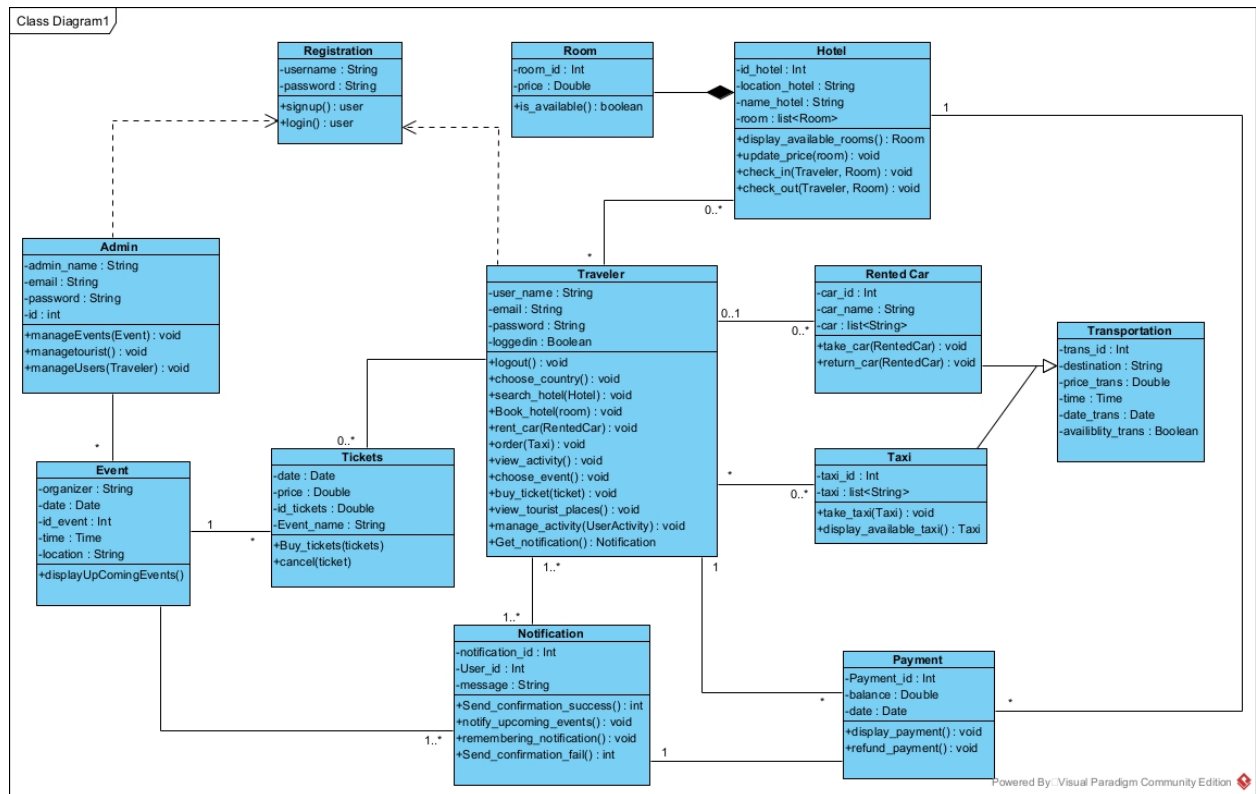
### III.2. Database

Database is defined as “a structured set of data held in a computer, especially one that is accessible in various ways.” For our **Travel Mate** application, a relational database management system (RDBMS) is an appropriate choice due to its ability to handle structured data and support complex queries. The database will store essential information such as user profiles (Traveler, Admin, Hotel Providers, etc.), booking details, payment transactions, hotel room availability, event schedules, and notifications. This structure will ensure efficient data retrieval, consistency, and scalability as the application grows.

### III.3. UML Class Diagram

- The Traveler is the main user who can book hotels, rent cars, order taxis, and buy tickets. Each traveler can receive multiple Notifications (1..\*) related to payments and events.
- The Payment class handles transactions, and every payment generates a Notification (1..1) to inform the traveler.
- The Event class is managed by an Admin, and when a new event is added, notifications are sent to travelers (1..\*).
- The Hotel class contains multiple Rooms (1..\*), and travelers can book rooms.

- The Transportation class includes Taxi and Rented Car, allowing travelers to select their mode of travel.
- Tickets are linked to Events, allowing travelers to purchase and cancel them.

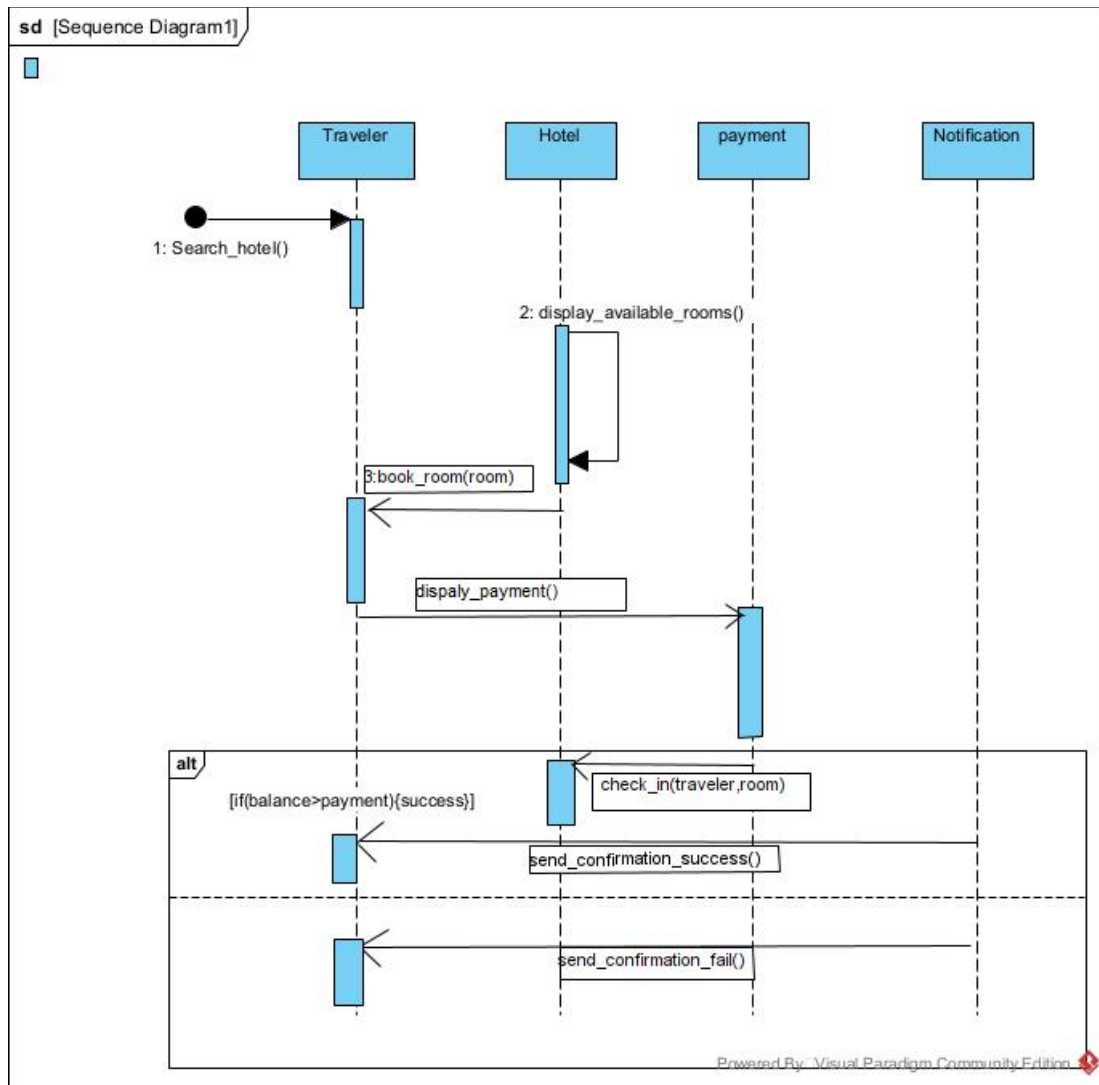


### III.4. Sequence Diagrams

Here's a breakdown of how the Sequence Diagram is structured.

#### Scenario:

1. Traveler searches for a hotel
2. Hotel displays available rooms
3. Traveler selects a room and books it
4. Traveler proceeds with payment
5. Payment is processed
6. Notification is sent to the traveler confirming the booking

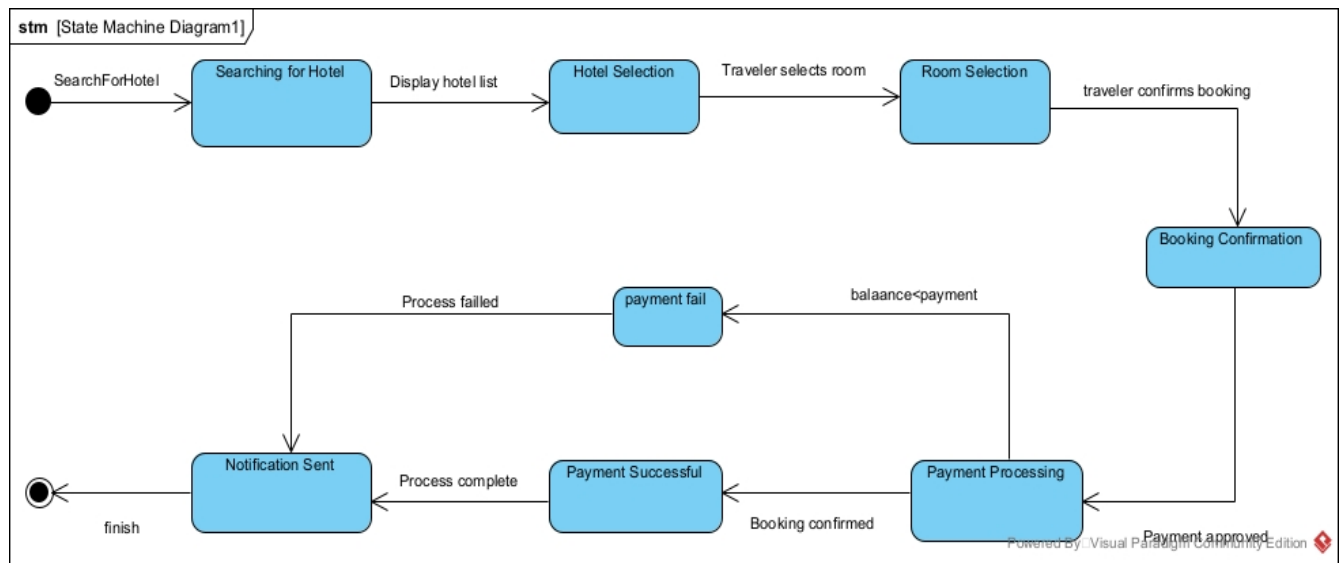


### III.5. State Chart Diagram

Here are the detailed steps for the State Diagram representing a traveler searching for a hotel, booking a room, making a payment, and receiving a notification:

1. Search for Hotel: The traveler initiates a search for available hotels.
2. Hotel Selection: The traveler selects a hotel from the search results.
3. Room Selection: The traveler chooses a room within the selected hotel.
4. Booking Confirmation: The traveler confirms the selected room for booking.

5. Payment Processing: The system moves to the payment process where the traveler provides payment details.
6. Payment Successful: If the payment is approved, the system confirms the transaction.
7. Payment failed: If the payment declined, the system sent a notification. ( $\text{payment} > \text{balance}$ ).
8. Notification Sent: A notification is sent to the traveler confirming the booking.



### III.6. Conclusion

This chapter presented the design of our application, First, we introduced the UML class diagram that clearly illustrated the relationships and roles of various entities, such as the Traveler, Hotel, Payment, Event, and Transportation systems. After that, we presented sequence diagrams to detail how the system functions during the booking and payment process, and a state chart diagram to highlight the system's transitions. In the next chapter, we will conduct a complete feasibility study, outlining the project's realization plan, risk management strategy, and measures for client support and satisfaction.

## Chapter IV - Conclusion

Working on the **Travel Mate** project has been a valuable learning experience. The goal was to create an app that simplifies travel planning by offering a range of services. Through development, we learned a lot about integrating different APIs and managing various features in a single app. While there were some challenges, such as ensuring real-time data and working with third-party services, the project helped us gain practical experience in problem-solving and app design.

### IV.1. Future Considerations

Future work can focus on:

1. Expanding Services: Adding flights, local transportation, and restaurant bookings.
2. AI: Implementing AI for customized recommendations and better trip planning.
3. Local Business Integration: Partnering with local services for exclusive deals.

## Bibliography

[1] Sommerville, Software Engineering, Pearson, 2014.

[2] Beginning Software Engineering, Rod Stephens. John Wiley & Sons, Inc.