**Luxor university**

**Faculty of Computer and Information**

**Computer Science Department**

**Project Name**

**Graduation Project**

**Under Supervision**

**Submitted by**

**2023**

**ABSTRACT**

Rahhal is an intelligent system designed to make travel experiences in Egypt easier and more enjoyable. It creates personalized plans for tourists, suggesting historical places to visit, nearby hotels, restaurants, and more based on the type of experience they want to have.

Users provide a description of their desired experience, whether they are international tourists visiting Egypt or Egyptians exploring their own country. Based on this input, Rahhal generates a tailored plan that meets their preferences and expectations.

Additionally, the system connects users with recommended trips organized by tourism companies that align with their interests. By offering personalized and convenient recommendations, Rahhal aims to enhance travel planning and ensure a memorable journey for all.

In conclusion, Rahhal makes travel planning in Egypt easier and more enjoyable by offering personalized recommendations based on each person’s preferences. It helps both tourists and locals have better experiences while also supporting local tourism businesses. With Rahhal, exploring Egypt becomes simple, fun, and memorable for everyone.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
|  | **Content** | **Page** |
| Abstract | | II |
| Table of Content   |  |  | | --- | --- | | List of abbreviation | V | | | III  v |
| List of tables | | ix |
| List of Figures | | x |

[**Chapter 1 System Overview** 1](#_Toc123970747)

[1.1 Introduction 2](#_Toc123912870)

[1.1.1 Motivation 2](#_Toc123912874)

[1.1.2 Problem Statement 2](#_Toc123912874)

[1.1.3 Overview 3](#_Toc123912874)

[**Chapter 2 Related Work** 4](#_Toc123970750)

[**Chapter 3 Domain Analysis and Technique** 12](#_Toc123970754)

[3.1 Domain Analysis 13](#_Toc123912870)

[3.1.1 Clients and users 15](#_Toc123912874)

[3.1.2 The Environment 15](#_Toc123912874)

[3.1.3 Tasks and actions currently being performed 15](#_Toc123912874)

[3.2 Risk 16](#_Toc123912865)

[3.3 Constrain 17](#_Toc123912865)

[3.4 Project Plan 18](#_Toc123912865)

[3.5 Feasibility Study 19](#_Toc123912865)

[3.6 Quality Assurance Plan  20](#_Toc123912865)

[3.8 System Requirements 24](#_Toc123912865)

[3.9 Techniques and tools 25](#_Toc123912866)

[3.10 Application Components 26](#_Toc123912865)

[3.11 System Request 28](#_Toc123912865)

[**Chapter 4 Proposed System and Methodology** 29](#_Toc123970757)

[4.1 System Use-Cases 30](#_Toc123970759)

[4.2 Use Case Description (Use case scenario) 33](#_Toc123970760)

[4.3 System Architecture 60](#_Toc123970761)

[4.4 Analysis Class 61](#_Toc123970762)

[4.4.1 State Diagram 61](#_Toc123970763)

4.4.2 Data flow diagram (Level Zero diagram) 63

4.4.3: Context Diagram 64

[4.5 interaction class diagram 65](#_Toc123970764)

[4.5.1 System diagram 65](#_Toc123970765)

4.5.2 Sequence Diagram 68

[4.6 Design Class 69](#_Toc123970766)

[4.6.1 Class Diagram 69](#_Toc123970767)

[4.7 Database Schema 70](#_Toc123970768)

[4.8 ER Diagram 71](#_Toc123970769)

4.9 Design Mockup

**Chapter 5: Conclusion& Future work** 103

7.1 Conclusion 104

7.2. Future work 104

7.3. References 105

**List of Abbreviation**

|  |  |
| --- | --- |
| **Keyword** | **Meaning** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**List of Tables**

|  |  |  |
| --- | --- | --- |
|  | **Table** | **Page** |

**List of Figures**

|  |  |  |
| --- | --- | --- |
|  | **Figure** | **Page** |

**Chapter 1**

**System Overview**

* 1. **Introduction**
     1. **Motivation**

Rahhal is an intelligent system designed to make travel experiences in Egypt easier and more enjoyable. It creates personalized plans for tourists, suggesting historical places to visit, nearby hotels, restaurants, and more based on the type of experience they want to have.

Users provide a description of their desired experience, whether they are international tourists visiting Egypt or Egyptians exploring their own country. Based on this input, Rahhal generates a tailored plan that meets their preferences and expectations.

Additionally, the system connects users with recommended trips organized by tourism companies that align with their interests. By offering personalized and convenient recommendations, Rahhal aims to enhance travel planning and ensure a memorable journey for all.

In conclusion, Rahhal makes travel planning in Egypt easier and more enjoyable by offering personalized recommendations based on each person’s preferences. It helps both tourists and locals have better experiences while also supporting local tourism businesses. With Rahhal, exploring Egypt becomes simple, fun, and memorable for everyone.

‎**1.1.2 Problem Statement**

Egypt, with its rich history, diverse attractions, and unique cultural experiences, is a dream destination for many tourists. However, the vast variety of cities, landmarks, and activities can overwhelm travelers, especially when their time is limited. Choosing the most suitable destinations and experiences becomes a challenging task, particularly when:

1. **Limited Time and Resources**: Tourists often have only a few days to explore Egypt and need to decide which cities and attractions will provide the best experience.
2. **Seasonal Preferences**: Certain destinations are better suited for specific seasons, such as summer or winter, but tourists may not be aware of these nuances.
3. **Individual Preferences**: Travelers have unique interests, whether it’s exploring historical landmarks or engaging in adventurous activities. Traditional methods of planning, like asking friends or browsing online reviews, often fail to consider these individual needs.
4. **Information Overload**: The internet provides a wealth of information, but sifting through it to find accurate, reliable, and personalized recommendations is time-consuming and inefficient.

Without a structured and personalized approach, tourists may miss out on experiences that align with their preferences, leading to suboptimal travel plans and reduced satisfaction. This gap highlights the need for an intelligent solution that simplifies decision-making and delivers tailored recommendations, including the ability to book trips directly through the system.

**1.1.3 Overview**

This project introduces a smart, recommendation-based model designed to enhance the travel planning experience for tourists visiting Egypt. By addressing the challenges of information overload, lack of personalization, and time constraints, our model offers tailored suggestions for destinations, accommodations, landmarks, and curated trips, all of which can be booked directly through the system.

The system operates by collecting specific input from tourists, such as:

* **Preferred Activities**: Whether they wish to explore historical sites, enjoy adventurous experiences, or relax in serene locations.
* **Travel Season**: The time of year they plan to visit, ensuring recommendations are seasonally appropriate.

Using this information, the model generates personalized recommendations, including:

* **Cities**: Suggesting the most suitable cities to visit based on the tourist’s interests and the season.
* **Hotels**: Identifying accommodations that match their preferences for comfort and proximity to attractions.
* **Landmarks**: Highlighting iconic sites and hidden gems aligned with their interests.
* **Organized Trips**: Proposing pre-planned trips offered by local companies that cater to their described preferences, which tourists can book directly through the system.

By streamlining the decision-making process, this project aims to save time, reduce uncertainty, and improve the overall travel experience. It also benefits local tourism companies by connecting them with tourists through targeted trip suggestions and booking options.

**Chapter 2**

**Related Work**

**2.1** **Kayak Explore: A Personalized Travel Discovery Tool**  
Figure 1, Kayak Explore Pl
at form

Kayak Explore is a comprehensive destination discovery tool integrated into the Kayak travel platform. It offers a simplified way of travel planning by taking consideration of preferences, budgetary constraints, and preferred vacation dates in order to help users find suitable travel destinations.

Key Features

1. Destination Discovery: Users input budget, departure city, and travel dates, with flexibility to adjust parameters. Kayak Explore generates a map displaying potential destinations that align with the user's criteria, accompanied by estimated flight prices.
2. Flexible Search: The "Anywhere" feature enables users to investigate a broad array of destinations, utilizing filters to refine results based on flight duration, climate, and available activities.
3. Cost Concerns: helps with budget-based decision-making by providing predicted flight prices to various locations.
4. Interactive Map: With the flexibility to zoom in and out to examine regions or individual cities, the interactive map that shows destinations helps consumers see vacation possibilities that fit within their budget.
5. Motivating Travel: highlights popular vacation spots, special offers, and unusual locations; perfect for last-minute budget travel or tourists looking for inspiration.
6. Cost-effective Choices: helps find places where flights are less expensive at particular times.

2.2 Routeperfect: Travel Itinerary Planning Platform

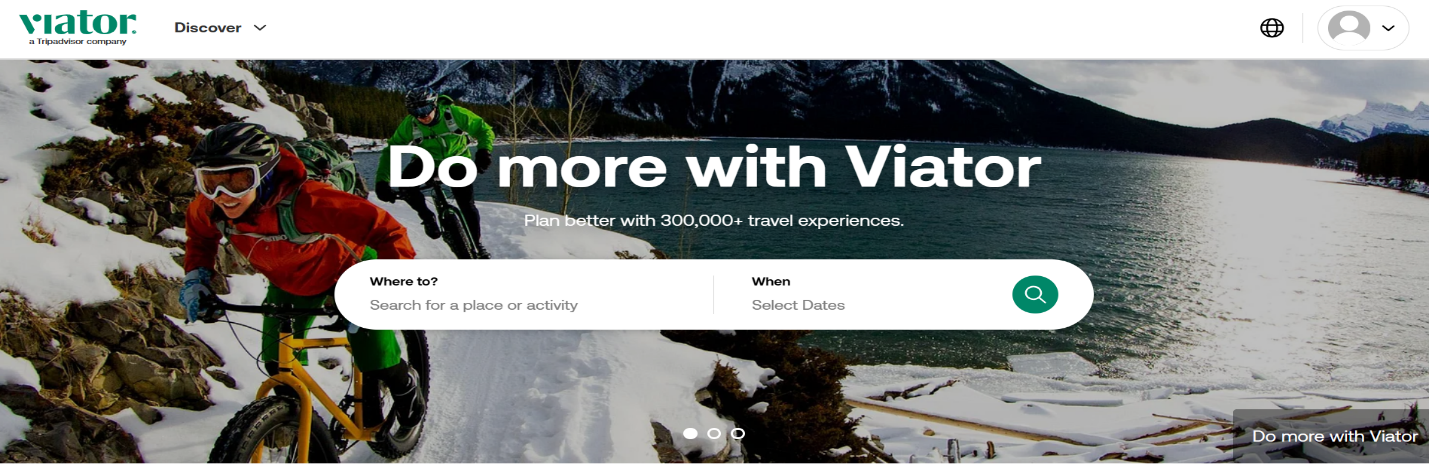


RoutePerfect is an online trip planning tool made to make the process of making customized and efficient travel plans easier. Through the use of advanced algorithms and user settings, RoutePerfect helps travelers create specific routes that fit their time, money, and interests.

Key Features

1. Personalized Route Creation: Using user inputs including preferred destinations, trip dates, interests, and budget, RoutePerfect employs advanced algorithms to create personalized travel plans.
2. Destination Recommendations: The website offers carefully chosen suggestions for places to visit, things to do, and activities, assisting users in finding popular hotspots and hidden treasures based on their interests.
3. Optimized Route Planning: RoutePerfect makes sure customers can make the most of their trip without needless delays or detours by optimizing travel routes to save travel time and maximize efficiency.
4. Budget Management Tools: Users can choose their spending limits, and the site makes recommendations for lodging, entertainment, and modes of transportation that fit within their means.
5. Interactive Maps and Visualizations: The platform integrates interactive maps to help users visualize their itinerary, making it easier to understand the geographical layout of their trip and adjust plans as needed.
6. Multi-Destination Trips: RoutePerfect supports the planning of complex, multi-destination trips, allowing users to seamlessly connect multiple cities or countries within a single itinerary.

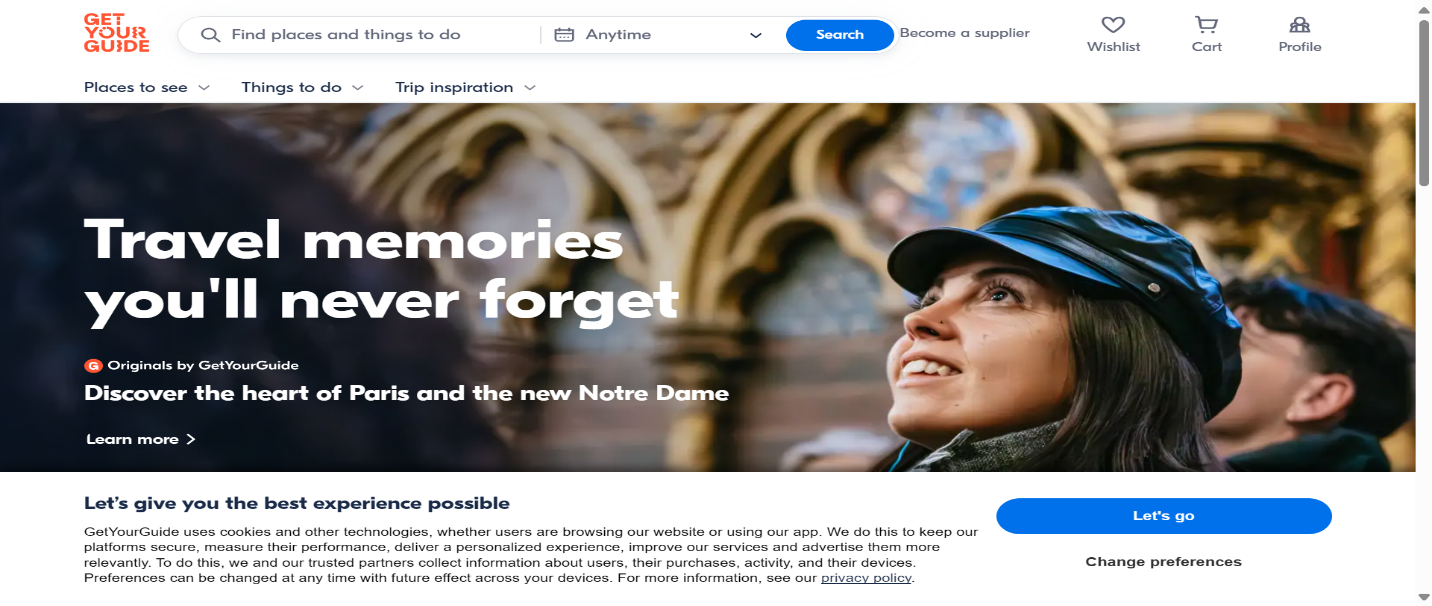
2.3 Viator: Global Travel Experiences Marketplace

Viator is a popular global marketplace for travel experiences, providing a thorough platform for tourists to find, evaluate, and reserve tours, activities, and attractions across the world. Viator, a subsidiary of Tripadvisor, links millions of tourists with regional tour companies, giving them access to a wide range of well-chosen experiences in global locations. From gourmet experiences and family-friendly excursions to adventure sports and cultural trips, the platform is made to accommodate a wide range of travel interests.

Key Features

1. Wide Selection of Travel Experiences: Viator makes sure that travelers may find experiences that suit their interests and preferences by providing a vast array of more than 300,000 tours, activities, and attractions in locations across the globe.
2. User-friendly Search and Booking Interface: Users may browse the platform by destination, activity type, duration, price, and customer ratings with ease thanks to its user-friendly search and filtering system.
3. Verified Reviews and Ratings: Viator gives consumers access to real traveler reviews and ratings, enabling them to base their choices on prior customers' experiences.
4. Instant Confirmation and Mobile Tickets: The majority of reservations made on Viator are confirmed instantly, and users may access mobile tickets for increased convenience by doing away with printed paperwork.
5. Managed Collections and Suggestions: The website offers carefully chosen collections and tailored suggestions, like "Top Attractions," "Bucket List Experiences," and "Hidden Gems," to motivate tourists and make choosing easier.
6. Integration with Tripadvisor: Viator, a member of the Tripadvisor family, easily incorporates into Tripadvisor's ecosystem, giving consumers access to extra travel resources in one location, like restaurant reviews and hotel reservations.

2.4 GetYourGuide: A Global Platform for Booking Local Tours and Activities

GetYourGuide is a popular platform that offers a wide range of carefully chosen activities and guided tours in many different countries, all aimed at improving travel experiences. It acts as a marketplace where tourists can easily find and reserve local experiences, guaranteeing a fun and easy trip.

Key Features

1. Extensive Activity Selection: GetYourGuide provides a comprehensive range of activities, from cultural tours and adventure experiences to gourmet workshops, catering to diverse interests and preferences.
2. Easy Booking Process: Users can effortlessly browse, select, and book activities with a simple, intuitive interface.
3. User Reviews and Ratings: Trustworthy reviews and ratings from fellow travelers help users make informed decisions.
4. Mobile Accessibility: The platform is fully optimized for mobile devices, allowing users to book and manage activities on the go.
5. Instant Confirmation: Users receive immediate confirmation upon booking, ensuring peace of mind.
6. Filters for Personalized Search: Advanced filters enable users to find activities based on interests, duration, price, and more.
7. Special Offers and Deals: GetYourGuide often features exclusive discounts and deals, offering value for money.

2.5 Aspects of our project

A distinctive feature of our project lies on its new combination of user-driven trip planning with AI-powered recommendation systems, resulting in a more personalized and efficient experience for travelers. Instead of providing generic travel platforms, our system enables users to input trip descriptions in natural language, giving them a comprehensive overview of their preferences, interests, and needs. We use machine learning and Natural Language Processing (NLP) to analyze user inputs, resulting in recommendations for personalized destinations, activities, and pre-designed trips from leading tourism companies.

Travelers may now more effectively plan their vacations and engage with tourism companies by using user-centric feedback and advanced artificial intelligence (AI). Our chatbot helps customers make intelligent choices fast and easily by showing them pre-planned vacations and activities from reliable sources.

Furthermore, the system's flexibility enables it to deal with a wide range of customer preferences, from families looking for all-inclusive packages to lone travelers looking for unusual destinations. It optimizes travel dates and costs by offering consumers tailored recommendations, improving their planning experience.

In the end, this idea offers a new advancement in travel technology that will transform how people organize their vacations. This solution not only satisfies the demands today's travelers but also establishes a standard for upcoming developments in travel technology by combining AI-driven insights with the adaptability of natural language input.

**Chapter 3**

**Domain Analysis and Technique**

**3.1 Domain Analysis**

## 3.1.1 CLIENTS AND USERS

USERS:

- Tourists: The primary users of the system, tourists input their preferences and receive tailored travel recommendations. They can explore various cities and trips and make bookings directly through the platform.

- System Administrators: Responsible for managing the platform, ensuring data accuracy, updating content, and maintaining the overall system functionality. They handle user accounts, oversee bookings, and ensure smooth operations.

CLIENTS:

- Tourism Companies: These entities collaborate with the system to list their trips and packages, providing tourists with curated travel options. They benefit from the platform's exposure and integrated booking system.

3.1.2 THE ENVIRONMENT

Web-Based Application:

- Framework: The frontend of the application uses HTML, CSS, Bootstrap, JavaScript, and React, ensuring a responsive and interactive user interface. This combination allows for a seamless user experience across different devices and screen sizes.

-Backend Operations: The backend is built with Spring Boot and MySQL, providing robust server-side operations and efficient data management.

Server Infrastructure:

- Hosting Services: The application is hosted on a cloud platform, ensuring scalability, reliability, and security for user interactions and data storage.

- Spring Boot Integration: Optimized to support backend functionalities, enabling real-time processing and interaction with the frontend.

Development and Testing:

- GitHub: Employed as the version control system, facilitating Agile Development and continuous integration, allowing for collaborative updates by the development team.

- Jira: Used for project management and issue tracking, helping the team stay organized and meet project deadlines.

## 3.1.3 TASKS AND ACTIONS CURRENTLY BEING PERFORMED

- Tourist Interaction: Tourists are currently interacting with the system by inputting their preferences, receiving recommendations, and making bookings for trips.

- System Administration: Administrators are managing the platform's content, ensuring data accuracy, and maintaining system operations to provide a seamless user experience.

- Content Updates: Regular updates to the database of cities and trips are being performed to keep the information current and relevant.

- Trip Management: Tourism companies are listing and updating trip packages to provide tourists with the latest and most suitable options.

- Feedback and Support: The system is collecting user feedback and providing support to address any issues or improve the platform's functionality based on user suggestions.

**3.2 Risks**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | impact | Likelihood | Mitigation |
| Data Collection and Accuracy | Collecting accurate, comprehensive, and up-to-date data for (e.g. places, trips) can be challenging. due to limited sources or uncooperative tourism companies. | Incomplete or outdated suggestions, leading to poor user experience. | High | Use multiple sources like APIs, user-generated content. |
| Inaccurate Suggestions | The system may suggest irrelevant, inappropriate places or trips due to poor data quality, insufficient algorithms, or failure to understand user input | Users may lose trust in our website, leading to decreased engagement and negative reviews. | High | Use robust Natural Language Processing (NLP) models to accurately understand user descriptions. |
| Data Privacy and Security Risks | User data (e.g. trip descriptions, preferences) could be exposed to unauthorized parties due to weak security measures. | Legal consequences, loss of user trust, and reputational damage. | High | Implement strong authentication and authorization mechanisms. |
| Dependency on Tourism Company Data | the platform relies heavily on data from tourism companies, So it may suggest biased or limited options, excluding smaller or independent providers. | Users may think the platform is unfair or incomplete, making it less attractive. | Medium | Allow users to filter results based on their preferences (e.g., budget, type of provider). |
| Scalability Issues | The platform may struggle to handle a large number of users or trips, leading to slow performance or crashes. | Poor user experience and potential loss of users. | Medium | Regularly test the platform under high traffic conditions. |
| Dependency on Third-Party APIs | The platform will rely on third-party APIs (Payment gateway) that could become unavailable or change their terms of service. | Disruption of service and loss of functionality. | Medium | Regularly monitor the status and terms of third-party APIs. |
| Technical Failures | The platform may experience technical issues such as bugs, server downtime, or integration failures. | Disruption of service and frustrated users. | High | Conduct strict testing (e.g., unit, integration, and stress testing). |

**3.3 Constrains**

1. Tourist Interaction:

* Can interact with the AI chatbot, receive travel plans, view suggested trips, book, and pay.

1. AI chatbot:

* AI chatbot should support free-form queries and structured questions for planning trips.
* Ensure suggestions consider the tourist's preferences, budget, and time constraints.

1. Booking and Payment:

* Bookings should be confirmed only after successful payment.
* Each payment must generate a unique ticket number.
* Payment gateway integration must be secure.

1. Admin Panel:

* Admin accounts require authentication and authorization.
* Admins can add, update, or delete: Locations (e.g., attractions, restaurants, hotels) and Tourism company accounts.
* Admins should view payment and booking details securely.

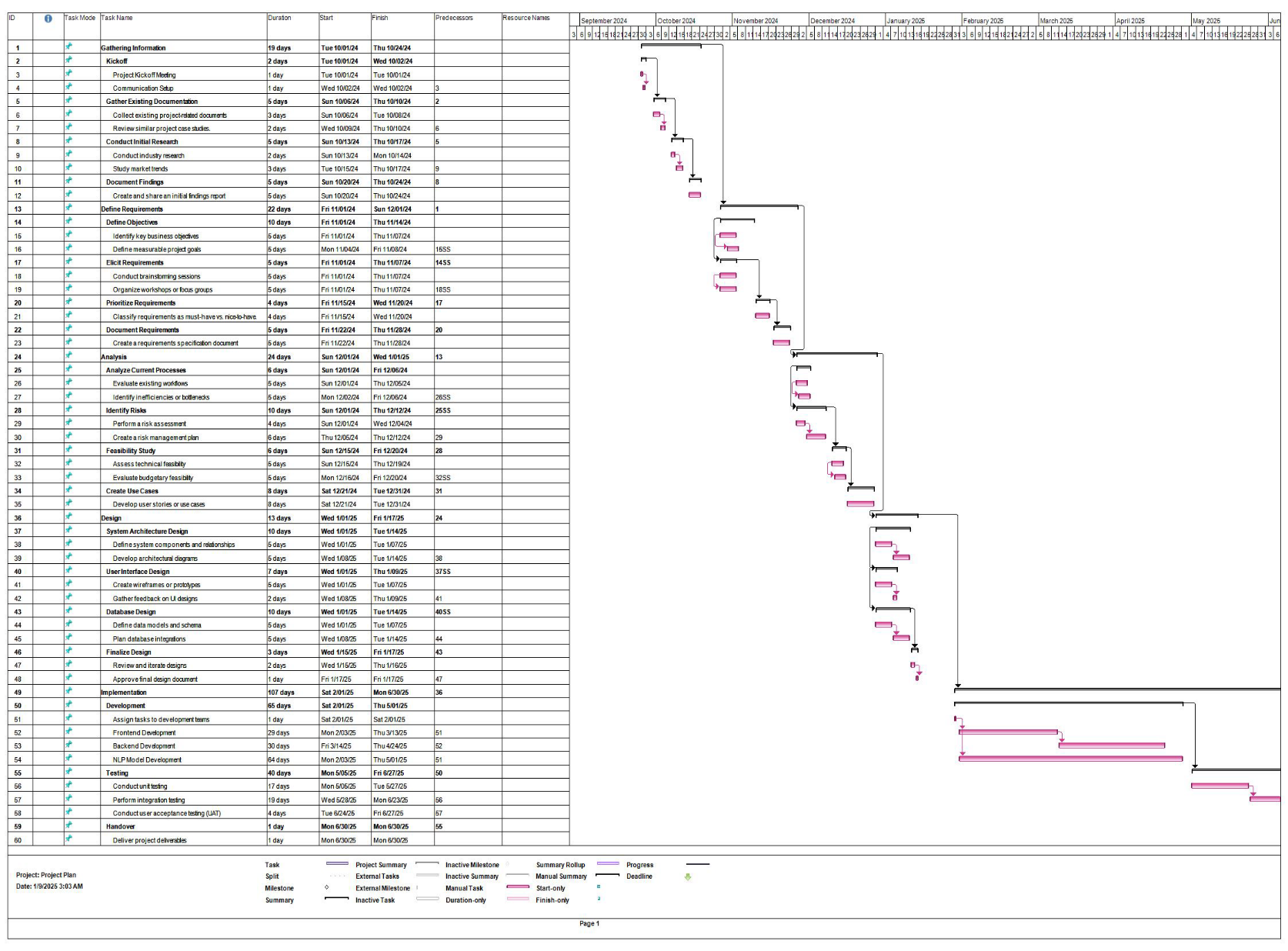
1. Data Validation:

* Ensure input validation for all user-provided data (e.g., dates, preferences, trips).

1. System Scalability:

* Handle a growing number of tourists, companies, and data entries.
* AI model must efficiently query a large dataset without performance issues.

**3.4 Project plan**

****

**3.5 Feasibility Study**

**3.5.1 Technical FEASIBILITY**

|  |  |
| --- | --- |
| SECTION | DESCRIPTION |
| Software Requirements | * The platform will use **React** for the frontend, providing a responsive and interactive user interface. * S**pring Boot** will power the backend, offering a robust framework for developing secure and scalable RESTful APIs. * M**ySQL** is selected as the relational database system for its reliability and ability to handle structured data efficiently. |
| Hardware Requirements | * The platform will be hosted on a scalable cloud service such as **AWS EC2**, **Google Cloud**, or **Azure**, ensuring high availability and performance. * Database storage will use cloud-hosted MySQL services or self-managed servers, depending on cost and performance needs. |
| Technology Assessment | * R**eact** is chosen for its component-based architecture and extensive library support, enabling fast and maintainable UI development. * S**pring Boot** is selected for its ability to simplify backend development and integrate seamlessly with MySQL. * M**ySQL** is chosen for its efficiency in handling structured data, ensuring robust transaction management and data integrity. |
| Technical Expertise | * The development team is skilled in **React**, **Spring Boot**, and **MySQL**, minimizing additional training or recruitment costs. * Expertise in integrating third-party APIs for payments and notifications is also available. |
| Development Tools | * IDEs: **IntelliJ IDEA**, **Visual Studio Code**. * Version Control: **Git**. |
| Security Measures | * S**SL encryption** for secure data transfer between the client and server. * Regular database backups and security patches to safeguard data integrity. |

**3.5.2 OPERATIONAL FEASIBILITY**

|  |  |
| --- | --- |
| SECTION | DESCRIPTION |
| User Acceptance Rate | * Anticipated to exceed **85%** based on the platform's intuitive design and user-friendly features tailored to tourists and tourism companies, enabling straightforward navigation and efficient functionality. |
| System Integration Period | * The system is expected to be fully operational within **4 months**, ensuring seamless onboarding for tourism companies and effective adoption of trip planning and booking features. |
| Throughput Enhancement | * Anticipated 25% improvement in operational efficiency for tourism companies, streamlining processes such as trip creation, bookings, and user communication. |
| Information Accessibility | * A projected **30% enhancement** in timely and accurate access to travel information, curated trip plans, and user preferences, leading to better decision-making for both tourists and tourism companies. |
| Adaptation Period | * Users (tourists and companies) are expected to adapt within 1-2 weeks, thanks to the platform's intuitive design and straightforward features. |
| User Satisfaction | * **Satisfaction Goals**: Targeting a **90% satisfaction rate** post-implementation by focusing on personalized recommendations, efficient booking processes. |

**3.6 Quality Assurance Plan**

**Testing Levels**

* **Unit Testing**: Validate individual components like user authentication, payment processing, and trip creation to ensure functionality.
* **Integration Testing**: Ensure seamless interaction between the frontend, backend, and database for consistent performance across features.
* **System Testing**: Test the entire platform to confirm all workflows, such as booking trips and sending messages, function as intended.

**Performance Metrics**

* **Responsiveness: Ensure fast and reliable interactions throughout the platform.**
* **System Reliability:** Maintain consistent functionality during peak usage periods.
* **Scalability:** Confirm the platform can handle increased user activity smoothly.

**Monitoring and Feedback**.

* **Real-Time Monitoring**: Use tools to track system performance and detect issues proactively.
* **User Feedback**: Provide options for users to report issues and suggest improvements.
* **Continuous Improvement**: Incorporate feedback and monitoring insights into regular updates to enhance usability and performance.

**Expected Outcomes**

* A reliable, high-performing application with accurate recommendations.
* Seamless integration between all components.
* Continuous improvement based on monitoring and user feedback.
  1. **System Requirements**

## Functional Requirements

* 1. **Chatbot Functionalities**
     1. User Input Handling:
        + Accept free-text input from tourists regarding their travel preferences.
        + Provide guidance or prompts to help users structure their preferences if needed.
     2. Preference Extraction:
        + Extract key features from the input text, such as destination type, activities, and time preferences.
     3. Natural Language Understanding (NLU):
        + Handle synonyms and variations in user language.
     4. Response Generation:
        + Respond with recommended destinations ranked by similarity scores.
        + Provide additional information about suggested destinations.
  2. **Recommendation System**
     1. Data Matching:
        + Compare user preferences with destination data.
        + Compute similarity scores and rank destinations based on relevance.
     2. Dynamic Suggestions:
        + Adapt recommendations based on real-time updates to user preferences.
     3. Feedback Loop:
        + Allow users to provide feedback on recommendations to improve future suggestions.
        + Update destination data based on user interactions.
  3. **Destination Data Management**
     1. Data Storage:
        + Maintain a database of destinations with metadata.
     2. Data Enrichment:
        + Allow administrators to add or update destination details regularly.
        + Include photos, descriptions, and links for more information.
     3. Categorization:
        + Classify destinations into categories (e.g., beach, city, adventure) to assist in matching.
  4. **User Account Management**
     1. Registration and Login:
        + Enable users to create accounts and log in securely.
        + Store user preferences for personalized recommendations.
     2. Preference History:
        + Allow users to view and update previously entered preferences.
  5. **User Experience Enhancements**
     1. Interactive Filters:
        + Allow users to refine results (e.g., filter by budget or location).
     2. Visual Feedback:
        + Display recommendations with images, maps, and brief descriptions for each destination.
  6. **Reporting and Analytics**
     1. User Insights:
        + Generate reports on user preferences and popular destinations.
     2. Performance Metrics:
        + Monitor system performance, including response time and recommendation accuracy.

## Non-Functional Requirements

1. **Performance**
   * 1. Response Time:
        + The system should respond to user queries within 2 seconds for most inputs.
        + Recommendations should be generated in under 3 seconds after processing the user’s preferences.
     2. Scalability:
        + The system should handle at least 10,000 concurrent users without degradation in performance.
     3. Throughput:
        + The system should support a high volume of transactions, such as processing 1,000 recommendations per minute.
2. **Reliability**
   * 1. Availability:
        + The system should have 99.9% uptime to ensure users can access the service at any time.
     2. Fault Tolerance:
        + In case of a server or database failure, the system should automatically redirect traffic to backup servers.
     3. Error Handling:
        + The system should provide clear error messages and recover gracefully from unexpected inputs or failures.
3. **Usability**
   * 1. Ease of Use:
        + The chatbot should be intuitive and require minimal learning effort.
        + Provide guided prompts or suggestions to help users frame their preferences.
     2. Accessibility:
        + Ensure the interface complies with accessibility standards, such as WCAG 2.1.
        + Provide support for screen readers and keyboard navigation.
4. **Security**
   * 1. Authentication:
        + Ensure secure login mechanisms with multi-factor authentication.
     2. Data Encryption:
        + Encrypt user data both in transit (using HTTPS) and at rest (e.g., AES-256).
     3. Privacy Compliance:
        + Provide users with options to view, update, or delete their personal data.
5. **Maintainability**
   * 1. Code Modularity:
        + Use a modular architecture to simplify maintenance and updates.
     2. Documentation:
        + Provide detailed technical documentation for developers, including API specifications and deployment instructions.
     3. Error Logging:
        + Implement comprehensive logging for debugging and monitoring purposes.
6. **Scalability**
   * 1. Cloud Deployment:
        + Use cloud services (e.g., AWS, Google Cloud, or Azure) for dynamic scaling based on demand.

**3.8 Techniques and tools**

3.9.1 React.js

Why React.js?

• It’s a powerful frontend library for building dynamic and responsive user interfaces.

• Offers component-based architecture, making it easy to reuse and maintain code.

• Provides a rich ecosystem of libraries and tools for enhanced functionality.

How It Fits with Our Project:

• User Interface: Build an intuitive and interactive UI for users to input trip descriptions and view recommendations.

• State Management: Use tools like Redux or Context API to manage user preferences and application state.

• Integration: Seamlessly connect with the backend (Spring Boot) to fetch and display personalized recommendations.

3.9.2 Spring Boot

Why Spring Boot?

• It’s a robust backend framework for building scalable and secure web applications.

• Offers built-in features like dependency injection, security, and database integration.

• Simplifies development with auto-configuration and a wide range of plugins.

How It Fits with Our Project:

• API Development: Create RESTful APIs to handle user inputs, process data, and return recommendations.

• Business Logic: Implement logic for analyzing user preferences, querying the database, and integrating with AI models.

• Integration: Connect with the frontend (React) and database (MySQL) to ensure smooth data flow.

3.9.3 MySQL

Why MySQL?

• It’s a reliable and widely-used relational database management system (RDBMS).

• Offers scalability and ease of use for managing structured data.

• Provides strong data integrity and ACID compliance for secure transactions.

How It Fits with Our Project:

• Data Storage: Store user profiles, trip descriptions, destination details, and tourism company data.

• Querying: Efficiently query the database to fetch personalized recommendations based on user inputs.

• Integration: Connect with the backend (Spring Boot) to retrieve and update data dynamically.

3.9.4 RASA

Why Rasa?

• It’s designed for conversational AI with built-in NLP capabilities.

• Create custom actions to fetch trip suggestions based on user input.

• Easy to train models to understand complex trip descriptions.

How It Fits with Our Project:

• Intent Recognition: Detect intents like "describe trip", "suggest destinations" or "match trips”.

• Entity Extraction: Extract key entities like location, budget, trip type, duration, etc.

• Custom Actions: Use custom Python actions to query database and return results.

3.9.5 FastAPI

Why FastAPI?

• Acts as a bridge between the chatbot, database, and tourism companies.

• Offers flexibility to handle additional logic, authentication, or pre/post-processing

• Implement custom APIs to query trips and destinations efficiently.

How It Fits with Our Project:

• Database Queries: FastAPI can handle MySQL queries to fetch trip and destination data.

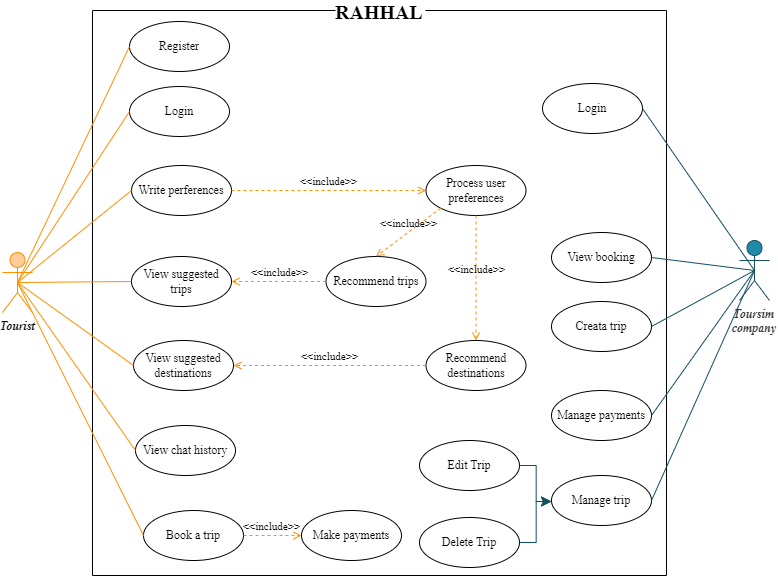
• Response Formatting: Format responses to be more user-friendly before sending them back to Rasa.

**Chapter 4**

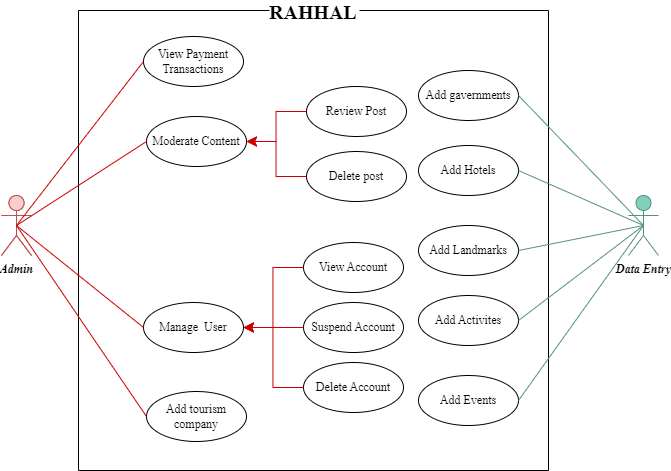
**Proposed System & Methodology**

**4.1 System Use-Cases**

**4.1.1: Client use-case**



**4.1.2: Admin Use-Case**

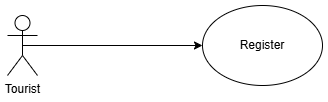


**4.2 Use Case Description (Use case scenario)**

**4.2.1 client usecases**

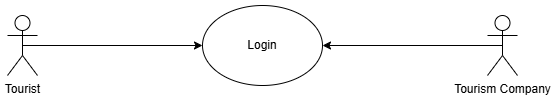
|  |  |
| --- | --- |
| **Use Case ID:** | UC001 |
| **Use Case Name:** | Register |
| **Area:** | User Authentication |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to create an account in the system to access its features. |
| **Preconditions:** | The tourist is not already registered. |
| **Post conditions:** | The tourist's account is successfully created. |
| **Main Flow:** | 1. The tourist navigates to the registration page. 2. The system displays a form with fields for required details. 3. The tourist fills out the form and submits it. 4. The system validates the input and creates the account.   The system sends a confirmation email to the tourist. |
| **Alternative Flow:** | If invalid data is provided, the system prompts for corrections. |

Register



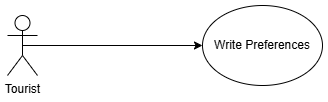
|  |  |
| --- | --- |
| **Use Case ID:** | UC002 |
| **Use Case Name:** | Log In |
| **Area:** | User Authentication |
| **Actor:** | Tourist, Tourism Company |
| **Description:** | Allows registered users (Tourists and Tourism Companies) to securely access their accounts. |
| **Preconditions:** | The actor has a registered account. |
| **Post conditions:** | The actor is successfully logged into the system. |
| **Main Flow:** | 1. The actor navigates to the login page. 2. The system displays a form for username and password. 3. The actor enters their credentials and submits the form.   The system validates the credentials and grants access. |
| **Alternative Flow:** | The system prompts the actor to retry or reset their password. |

Log In



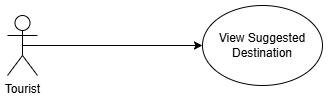
|  |  |
| --- | --- |
| **Use Case ID:** | UC003 |
| **Use Case Name:** | Write Preferences |
| **Area:** | Personalization |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to specify their travel preferences to receive personalized recommendations. |
| **Preconditions:** | * The tourist has logged into the system. * The preferences section is available. |
| **Post conditions:** | The preferences are saved successfully in the system. |
| **Main Flow:** | 1. The tourist logs into the system. 2. The tourist navigates to the preferences section. 3. The system displays a form with a field for description. 4. The tourist fills out and submits the form.   The system confirms the preferences are saved. |
| **Alternative Flow:** | If the tourist enters invalid data, the system prompts for corrections. |

Write Preferences



|  |  |
| --- | --- |
| **Use Case ID:** |  |
| **Use Case Name:** | V |
| **Area:** | Personalization |

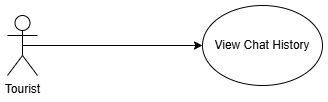
View Suggested Destinations



|  |  |
| --- | --- |
| **Use Case ID:** | UC004 |
| **Use Case Name:** | View Suggested Destinations |
| **Area:** | Personalization |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to browse destinations suggested by the system based on their preferences. |
| **Preconditions:** | * The tourist has saved preferences. * The system has generated destination suggestions. |
| **Post conditions:** | The tourist views a list of suggested destinations. |
| **Main Flow:** | 1. The tourist logs into the system. 2. The system retrieves destination suggestions based on the tourist's preferences. 3. The tourist browses the suggested destinations and clicks on individual destinations for more details. |
| **Alternative Flow:** | If no suggestions match preferences, the system suggests broader options. |

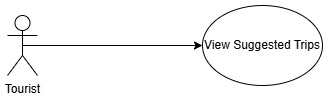
|  |  |
| --- | --- |
| **Use Case ID:** | UC005 |
| **Use Case Name:** | View Chat History |
| **Area:** | Customer Support |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to access their past communications with customer service. |
| **Preconditions:** | The tourist has written preferences before. |
| **Post conditions:** | The tourist successfully views past chat interactions. |
| **Main Flow:** | 1. The tourist logs into the system. 2. The tourist clicks on the "Chat History" option. 3. The system retrieves and displays past conversations. |
| **Alternative Flow:** | The system informs the user that no previous chats are available. |

View Chat History



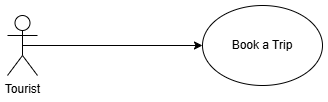
|  |  |
| --- | --- |
| **Use Case ID:** | UC006 |
| **Use Case Name:** | View Suggested Trips |
| **Area:** | Suggestion |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to browse curated trip options. |
| **Preconditions:** | Suggested trips created by tourism companies. |
| **Post conditions:** | The tourist views a list of curated trip options. |
| **Main Flow:** | 1. The tourist logs into the system. 2. The system displays a list of suggested trips. 3. The tourist clicks on a trip to view detailed information. |
| **Alternative Flow:** | The system notifies the user that there are no trips available ate the current time. |

View Suggested Trips



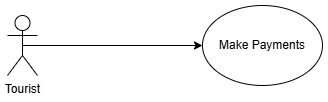
|  |  |
| --- | --- |
| **Use Case ID:** | UC007 |
| **Use Case Name:** | Book a Trip |
| **Area:** | Booking |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to select and book trips offered by the system. |
| **Preconditions:** | * The tourist has selected a trip. * The trip has available slots. |
| **Post conditions:** | The trip is booked, and the payment is processed. |
| **Main Flow:** | 1. The tourist selects a trip from the suggested list. 2. The system displays the trip details, including price and itinerary. 3. The tourist clicks the "Book Now" button. 4. The system navigates to the payment page. 5. The tourist completes the payment (triggers the "Make Payments" use case). 6. The system confirms the booking and generates a receipt. |
| **Alternative Flow:** | * The system informs the tourist and suggests alternate trips. * If payment fails, the system prompts the tourist to retry or use another method. |

Book a Trip



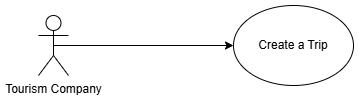
|  |  |
| --- | --- |
| **Use Case ID:** | UC008 |
| **Use Case Name:** | Make Payments |
| **Area:** | Payment Processing |
| **Actor:** | Tourist |
| **Description:** | Allows tourists to securely process payments for their bookings. |
| **Preconditions:** | The tourist has initiated the payment process. |
| **Post conditions:** | Payment is successfully processed. |
| **Main Flow:** | 1. The tourist selects a payment method (e.g., credit card, PayPal). 2. The system processes the payment. 3. A confirmation receipt is generated and sent to the tourist. |
| **Alternative Flow:** | The system prompts the user to retry with another method. |

Make Payments



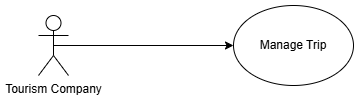
|  |  |
| --- | --- |
| **Use Case ID:** | UC009 |
| **Use Case Name:** | Create a Trip |
| **Area:** | Trip Management |
| **Actor:** | Tourism Company |
| **Description:** | Allows tourism companies to add new trips to the system for tourists to book. |
| **Preconditions:** | The company has access to the trip creation feature. |
| **Post conditions:** | A new trip is successfully created and listed. |
| **Main Flow:** | 1. The tourism company logs into the system. 2. They navigate to the "Create Trip" section. 3. The company inputs details like destination, activities, price, and availability. 4. The system saves the trip and makes it available for tourists. |
| **Alternative Flow:** | The system prompts the user to complete missing fields. |

Create a Trip



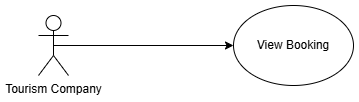
|  |  |
| --- | --- |
| **Use Case ID:** | UC010 |
| **Use Case Name:** | Manage Trip |
| **Area:** | Trip Management |
| **Actor:** | Tourism Company |
| **Description:** | Allows tourism companies to update and modify details of their trips. |
| **Preconditions:** | The trip is already created. |
| **Post conditions:** | The trip details are updated successfully. |
| **Main Flow:** | 1. The tourism company logs into the system. 2. They navigate to the "Manage Trip" section. 3. The company updates trip details, such as itinerary or pricing. 4. The system saves the changes. |
| **Alternative Flow:** | If the trip already booked, display error message. |

Manage Trip

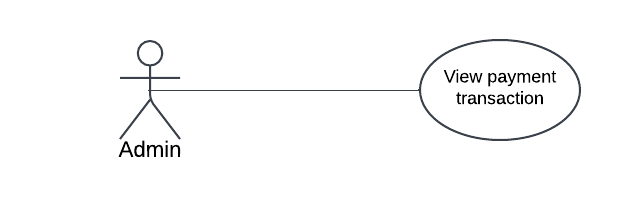


|  |  |
| --- | --- |
| **Use Case ID:** | UC011 |
| **Use Case Name:** | View Booking |
| **Area:** | Booking Management |
| **Actor:** | Tourism Company |
| **Description:** | Allows tourism companies to access and review bookings made by tourists. |
| **Preconditions:** | Bookings have been made by tourists. |
| **Post conditions:** | The company successfully views the booking details. |
| **Main Flow:** | 1. The company logs into the system. 2. They navigate to the "View Booking" section. 3. The system displays a list of all bookings. 4. The company selects a specific booking to view details such as customer information, trip details, and payment status. |
| **Alternative Flow:** | The system notifies the company that no bookings are currently available. |

View Booking

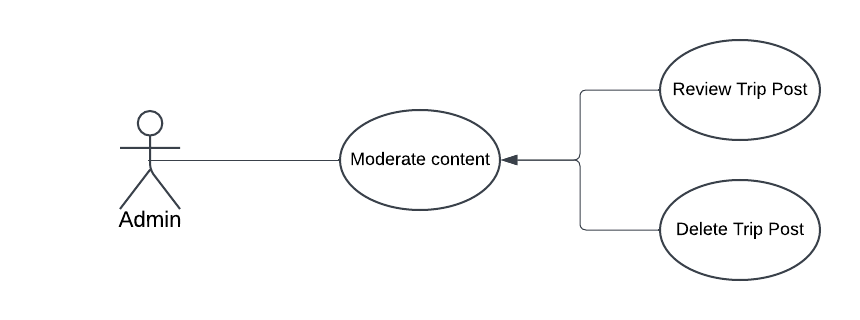


### 4.2.11. View Payment Transactions



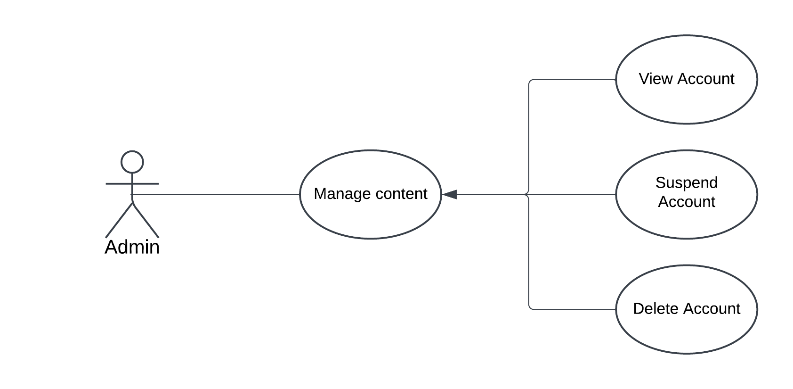
|  |  |
| --- | --- |
| **Use Case ID:** | UC\_011 |
| **Use Case Name:** | View Payment Transactions |
| **Area:** | Payment Management |
| **Actor:** | Admin |
| **Description:** | The admin reviews and monitors payment transaction. |
| **Preconditions:** | 1. Admin must be logged in 2. Payment transactions is existed |
| **Post conditions:** | The admin has a clear understanding of payment transactions and solve errors in the payment process if exists. |
| **Main Flow:** | 1. The admin navigates to the Transactions section. 2. Filters transactions by date, user, type or payment ID. 3. Reviews the transaction details. 4. Exports the data if needed. |
| **Alternative Flow:** | If no transactions are found, the admin receives a notification indicating no records are available. |

### 4.2.12. Moderate Content



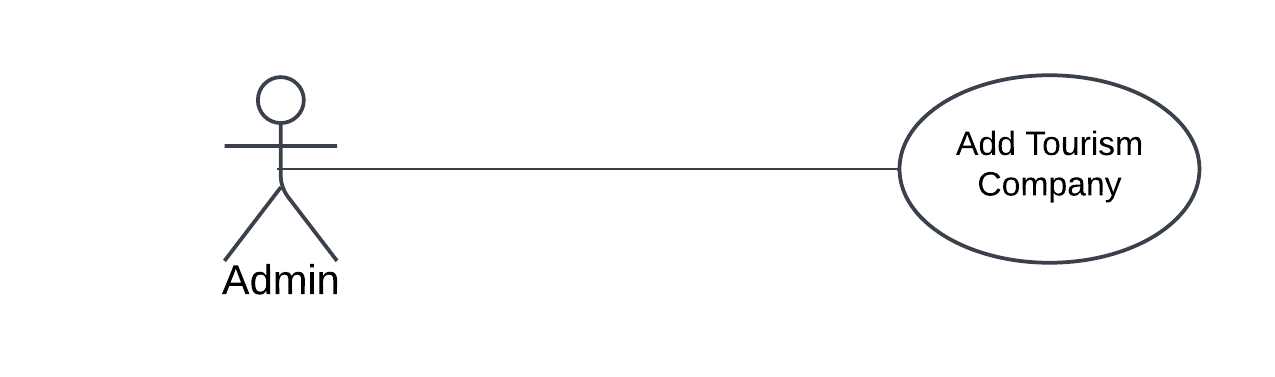
|  |  |
| --- | --- |
| **Use Case ID:** | UC\_012 |
| **Use Case Name:** | Moderate content |
| **Area:** | Content Mangement |
| **Actor:** | Admin |
| **Description:** | The admin reviews and approves or rejects tourism company posts. |
| **Preconditions:** | 1. Admin must be logged in 2. Trip posts existed |
| **Post conditions:** | Posts is either approved or rejected |
| **Main Flow:** | 1. The admin navigates to the Moderate Content section. 2. Reviews trip posts that have been posted 3. Approves or rejects the post. 4. Notifies the tourism company if necessary. |
| **Alternative Flow:** | If the post does not obey the website rules or have missing necessary data, the admin rejects it and issues a warning to the tourism company. |

### 4.2.13. Manage Users



|  |  |
| --- | --- |
| **Use Case ID:** | UD-013 |
| **Use Case Name:** | Manage Users |
| **Area:** | User Management |
| **Actor:** | Admin |
| **Description:** | The admin view user accounts and permissions. |
| **Preconditions:** | 1. The admin is logged into the system. 2. User accounts exist on the platform. |
| **Post conditions:** | User accounts are updated or managed as required. |
| **Main Flow:** | 1. The admin navigates to the Manage User section. 2. Selects a user account. 3. Updates roles or permissions. 4. Saves the changes. |
| **Alternative Flow:** | If the user violates website rules, inactive for a long time or user requests to suspend or delete his account, the admin can delete or suspend his account |

### 4.2.14. Add Tourism Company



|  |  |
| --- | --- |
| **Use Case ID:** | UD\_014 |
| **Use Case Name:** | Add Tourism Company |
| **Area:** | Trip management |
| **Actor:** | Admin |
| **Description:** | The admin adds a new tourism company. |
| **Preconditions:** | 1. The admin is logged into the system. 2. The tourism company details are available. |
| **Post conditions:** | The new tourism company is added to the data |
| **Main Flow:** | 1. The admin navigates to the Add Tourism Company section. 2. Enters company details (e.g., name, location, services, contact information). 3. Publishes the company listing. |
| **Alternative Flow:** | If the company already exists, the admin updates the existing listing. |

4.2.15. Add Governments



|  |  |
| --- | --- |
| **Use Case ID:** | UD\_015 |
| **Use Case Name:** | Add Governments |
| **Area:** | Data Management |
| **Actor:** | Data Entry |
| **Description:** | The Data Entry adds government information |
| **Preconditions:** | 1. The Data Entry is logged into the system. 2. Government information is correct and available. |
| **Post conditions:** | Government information added to the database. |
| **Main Flow:** | 1. The admin navigates to the Add Governments section. 2. Enters government details (e.g., name, description). 3. Verifies the information for accuracy. 4. Save the information. |
| **Alternative Flow:** | If the information is incomplete, the Data Entry role saves it as a draft for later completion. |

4.2.16. Add Hotels



|  |  |
| --- | --- |
| **Use Case ID:** | UD\_015 |
| **Use Case Name:** | Add Hotels |
| **Area:** | Data Management |
| **Actor:** | Data Entry |
| **Description:** | The Data Entry adds a new hotel. |
| **Preconditions:** | 1. The Data Entry is logged into the system. 2. Hotel details are correct and available. |
| **Post conditions:** | The new hotel is listed on the platform. |
| **Main Flow:** | 1. The admin navigates to the Add Governments section. 2. Enters hotel details (e.g., name, location, star rating). 3. Verifies the information for accuracy. 4. Save the information. |
| **Alternative Flow:** | If the information is incomplete, the Data Entry role saves it as a draft for later completion. |

4.2.17. Add Landmarks



|  |  |
| --- | --- |
| **Use Case ID:** | UD\_017 |
| **Use Case Name:** | Add Landmarks |
| **Area:** | Data Management |
| **Actor:** | Data Entry |
| **Description:** | The Data Entry adds a new landmark. |
| **Preconditions:** | 1. The Data Entry is logged into the system. 2. Landmark details are correct and available. |
| **Post conditions:** | Landmark details added to the database. |
| **Main Flow:** | 1. The admin navigates to the Add Landmark section. 2. Enters Landmark details (e.g., name, description, location). 3. Verifies the information for accuracy. 4. Save the information. |
| **Alternative Flow:** | If the Landmark details is incomplete, the Data Entry role saves it as a draft for later completion. |

4.2.15. Add Activities



|  |  |
| --- | --- |
| **Use Case ID:** | UD\_015 |
| **Use Case Name:** | Add Activities |
| **Area:** | Data Management |
| **Actor:** | Data Entry |
| **Description:** | The Data Entry adds new activity. |
| **Preconditions:** | 1. The Data Entry is logged into the system. 2. Activity details is correct and available. |
| **Post conditions:** | Activity details added to the database. |
| **Main Flow:** | 1. The admin navigates to the Add Activity section. 2. Enters activity details (e.g., name, description, type). 3. Verifies the information for accuracy. 4. Save the information. |
| **Alternative Flow:** | If the information is incomplete, the Data Entry role saves it as a draft for later completion. |

4.2.15. Add Events



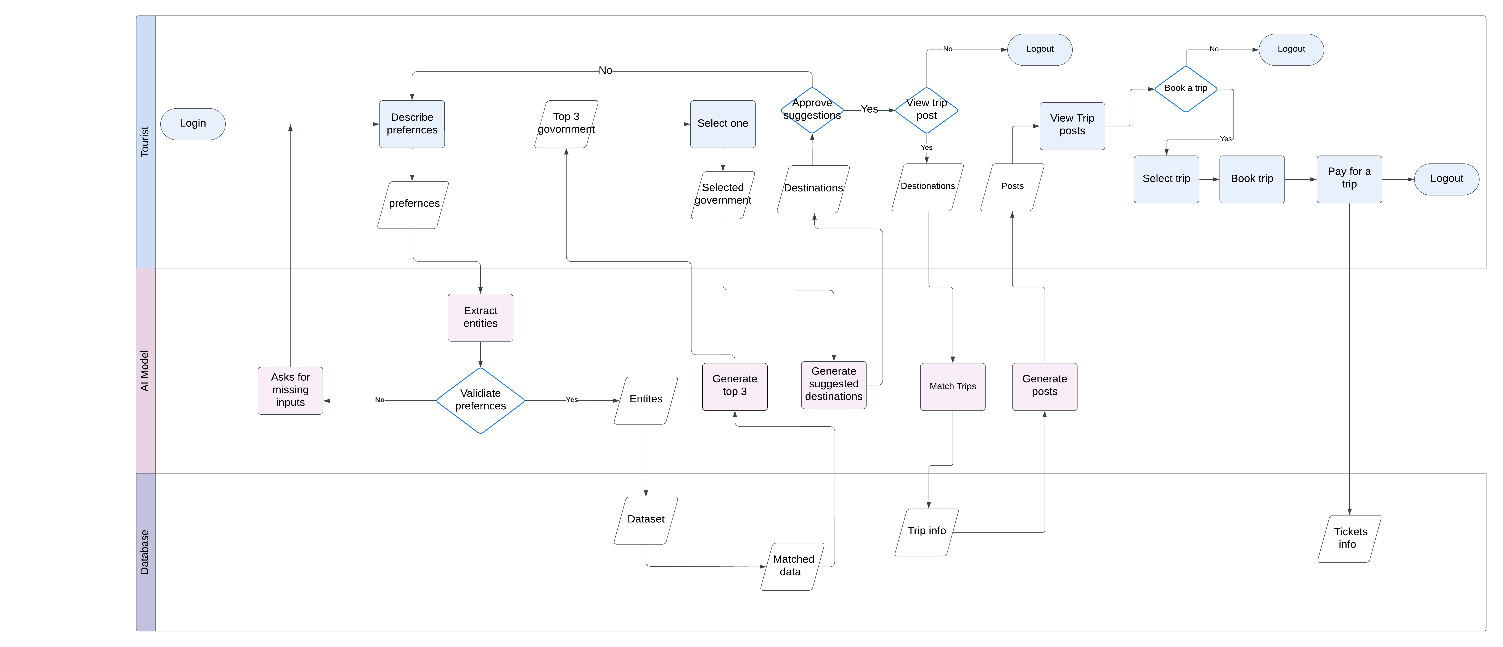
|  |  |
| --- | --- |
| **Use Case ID:** | UD\_015 |
| **Use Case Name:** | Add Events |
| **Area:** | Data Management |
| **Actor:** | Data Entry |
| **Description:** | The Data Entry adds upcoming events |
| **Preconditions:** | 1. The Data Entry is logged into the system. 2. Event details is correct and available. |
| **Post conditions:** | Event details added to the database. |
| **Main Flow:** | 1. The admin navigates to the Add Event section. 2. Enters event details (e.g., name, date, description). 3. Verifies the information for accuracy. 4. Save the information. |
| **Alternative Flow:** | If the information is incomplete, the Data Entry role saves it as a draft for later completion. |

**4.3 System Architecture**

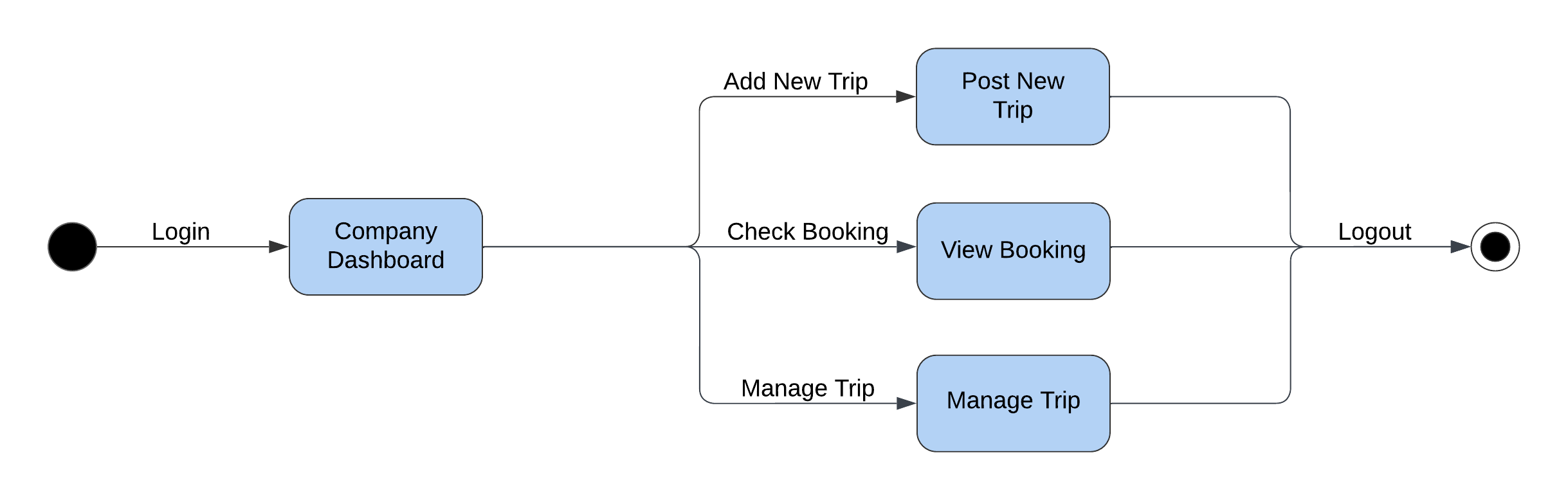
**4.4 Analysis Class**

**4.4.1 State Diagram**

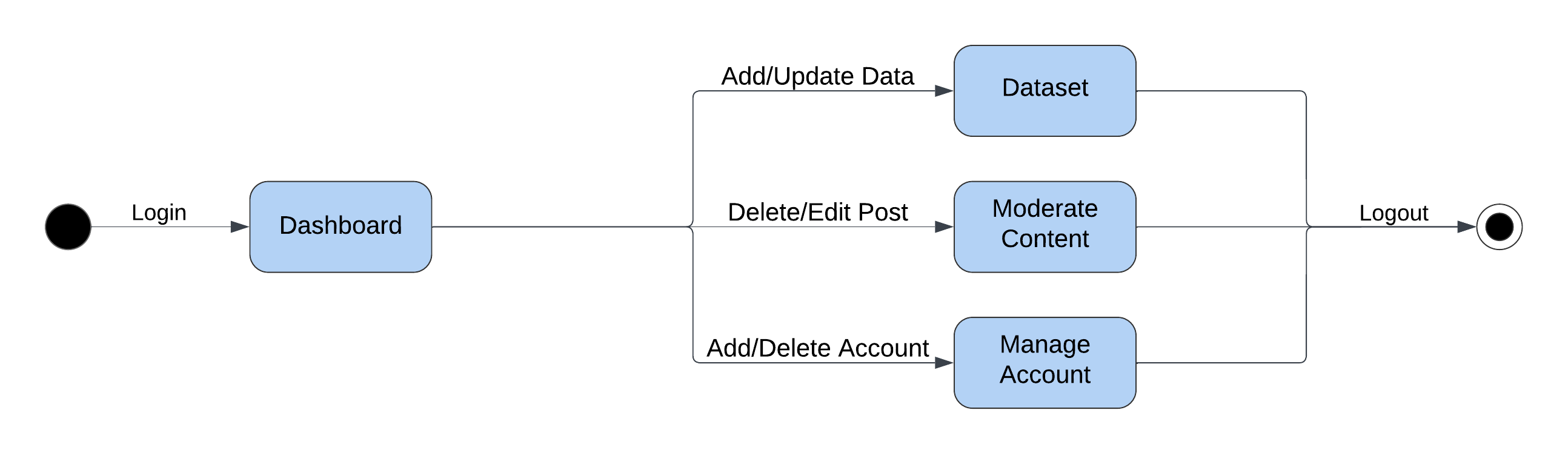
**4.4.1.1 State for tourist**



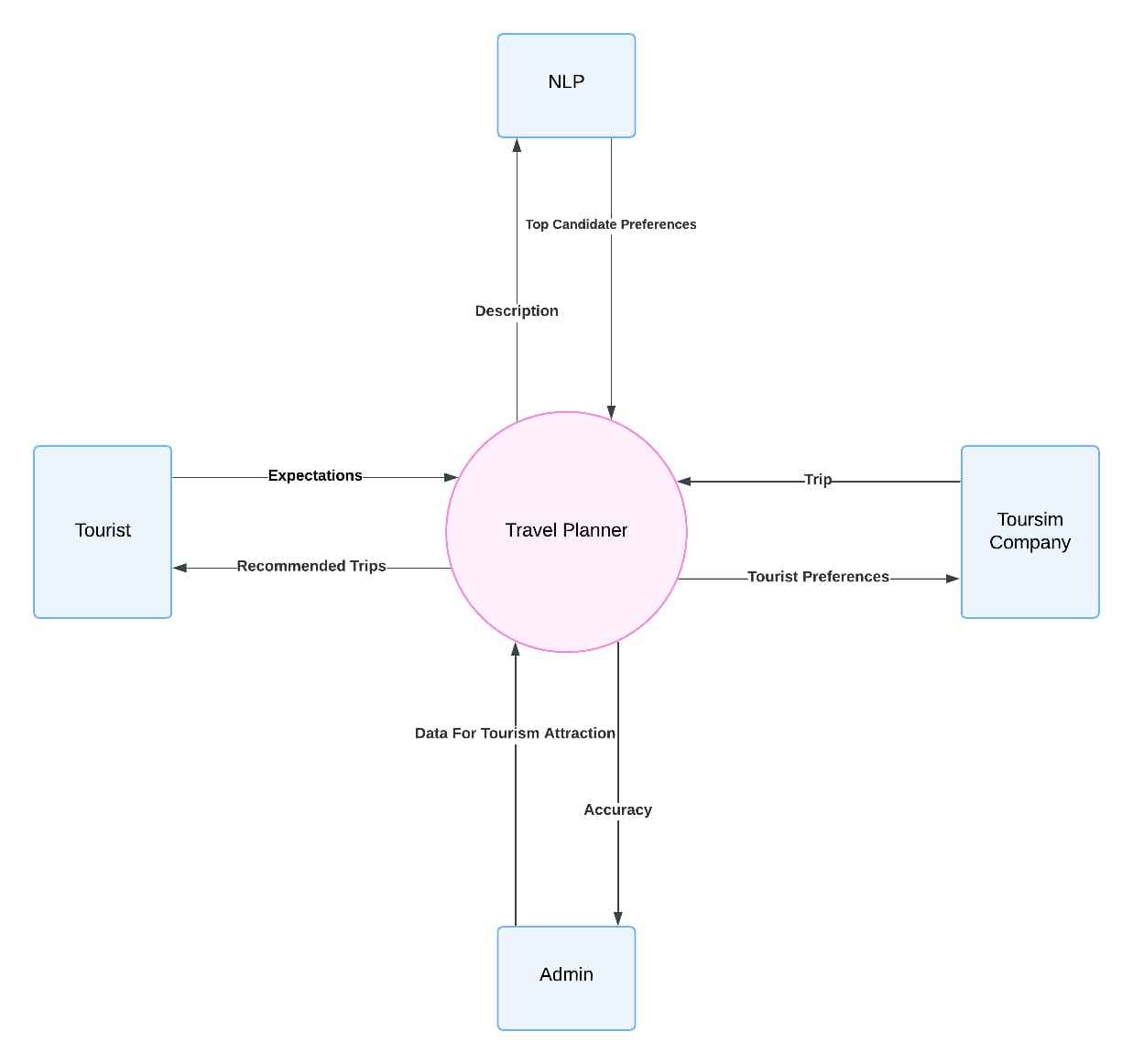
**4.4.1.2 State for company**



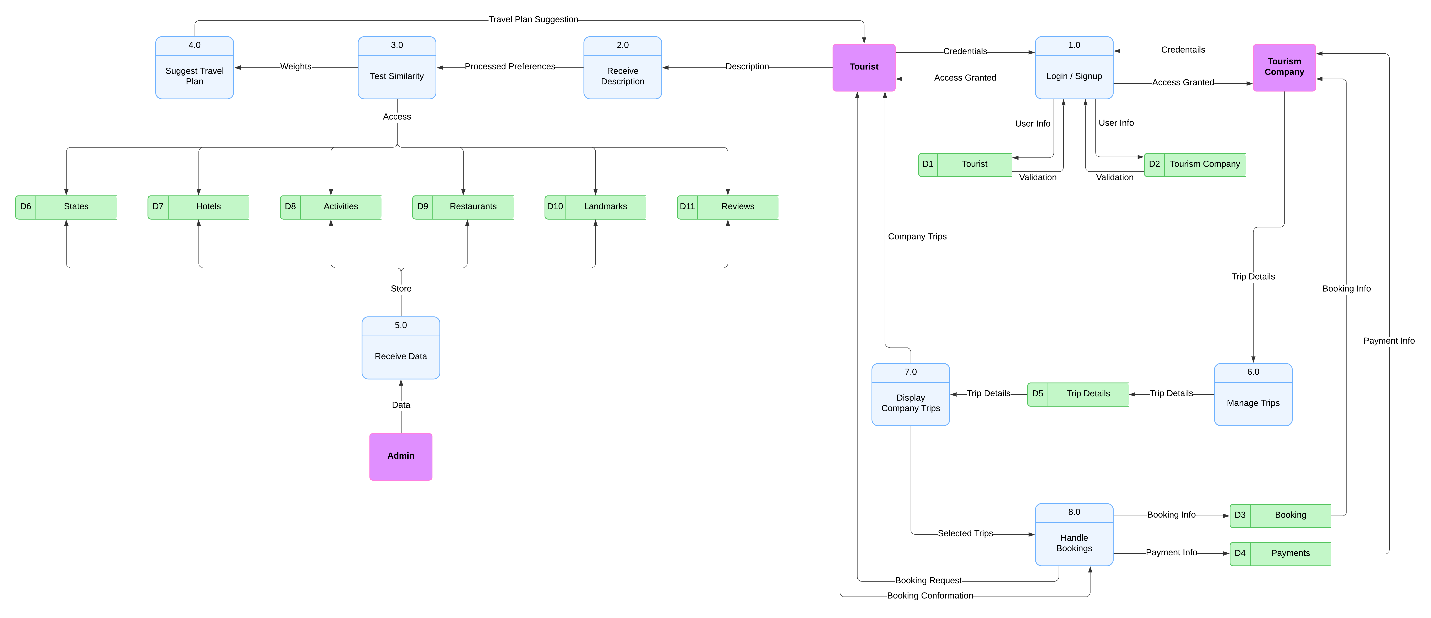
**4.4.1.2 State for Admin**



**4****.4.2 Context Diagram**



**4****.4.3: Data flow diagram**



**4.5 interaction class diagram**

**4.5.1 Sequence Diagram**

**4.5.1.1 NLP sequence**

****

**4.6 Design Class**

**4.6.1 Class Diagram**

**4.7 Database Schema**

* 1. **ER Diagram**
  2. **Design Mockup**

**Chapter 5**

**Conclusion & Feature work**

**5.1 Conclusions**

**5.2 Future work**

**5.3 References**