



Luxor university
Faculty of Computer and Information
Computer Science Department



Egypt Smart Journey Planner

Graduation Project - Part 1

Under Supervision

Dr. Mohamed Fouad

Dr. Mohamed Ramadan

Submitted by

Ahmed Yahia

Muhammad Saady

Hitham Omar

Sharaf Ahmed

Camellia Mohamed

Rwida Gaber



2025

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

Abstract	II
Table of Content	III
List of tables	v
List of Figures	ix
Chapter 1: System Overview	1
1.1 Introduction	2
1.2 Motivation	2
1.3 Problem Statement	3
1.4 Overview	4
1.5 Conclusion	5
Chapter 2: Related Work	6
2.1 Introduction	7
2.2 Kayak Explore: A Personalized Travel Discovery Tool	7
2.3 Routepointer: Travel Itinerary Planning Platform	8
2.4 Viator: Global Travel Experiences Marketplace	10
2.5 GetYourGuide: A Global Platform for Booking Local Tours and Activities	11
2.6 Our Project: A New Era of Travel Solutions	12
2.7 Conclusion	14
Chapter 3: Domain Analysis and Technique	15
3.1 Introduction	16
3.2 Domain Analysis	16
3.2.1 Clients and Users	16
3.2.2 The Environment	17
3.2.3 Tasks and Action Currently Being Performed	17
3.3 Risks	19
3.4 Project plan	21
3.5 Feasibility Study	22
3.5.1 Technical Feasibility	22
3.5.2 Operational Feasibility	23
3.6 Quality Assurance Plan	24
3.7 System Requirements	25
1. Functional Requirements	25
2. Non-Functional Requirements	27
3.8 Techniques and tools	28
3.8.1 React.js	28
3.8.2 Spring Boot	29
3.8.3 MySQL	30
3.8.4 RASA	30
3.8.5 FastAPI	31

3.9 Application Components	31
3.10 Conclusion	33
Chapter 4: Proposed System & Methodology	34
4.1 Introduction	35
4.2 System Use-Cases	36
4.2.1: Client use-case	36
4.2.2: Admin Use-Case	37
4.3 Use Case Description (Use case scenario)	38
4.3.1 Client-side scenarios	38
4.3.1.1 Register	38
4.3.1.2 Login	39
4.3.1.3 Write Preference	40
4.3.1.4 Process User Preferences	41
4.3.1.5 Recommend Destinations	42
4.3.1.6 Recommend Trips	43
4.3.1.7 View Suggested Destinations	44
4.3.1.8 View Chat History	45
4.3.1.9 View Suggested Trips	46
4.3.1.10 Book a Trip	47
4.3.1.11 Make Payments	48
4.3.1.12 Create a Trip	49
4.3.1.13 Manage Trip	50
4.3.1.14 View Booking	51
4.3.2 Admin-side scenario	52
4.3.2.1 View Payment Transactions	52
4.3.2.2 Moderate Content	53
4.3.2.3 Manage Users	54
4.3.2.4 Add Tourism Company	55
4.3.2.5 Add Governorates	56
4.3.2.6 Add Hotels	57
4.3.2.7 Add Landmarks	58
4.3.2.8 Add Activities	59
4.3.2.9 Add Events	60
4.4 System Architecture	61
4.4.1 Methodology	61
1. Requirement Analysis	61
2. System Design	62
3. Implementation (Upcoming Phase)	62
4. Integration and Testing	62
5. Deployment	63
6. Maintenance	63
Current Status and Justification	63
4.5 Analysis Class	64
4.5.1 Swim Lane Diagram	64

4.5.2 State Diagram	65
4.5.2.1 State for company	65
4.5.2.2 State for Admin	65
4.5.2 Context Diagram	66
4.5.2 Data flow diagram	67
4.6 interaction class diagram	68
4.6.1 Sequence Diagram	68
4.6.1.1 NLP sequence	68
4.6.1.2 Payment	69
4.6.1.3 Manage trip	70
4.7 Design Class	71
4.7.1 Class Diagram	71
4.8 Database Schema	72
4.9 Design Mockup	73
4.10 Conclusion	75
Chapter 5: Conclusion & Feature Work	76
5.1 Conclusions	77
5.2 Future work	78
5.3 References	79

List of Tables

Table 1: Risks	19
Table 2: Technical Feasibility Study	22
Table 3: Operational Feasibility Study	23
Table 4: "Register" Use Case Scenario	38
Table 5: "Login" Use Case Scenario	39
Table 6: "Write Preferences" Use Case Scenario	40
Table 7: "Process User Preferences" Use Case Scenario	41
Table 8: "Recommend Destinations" Use Case Scenario	42
Table 9: "Recommend Trips" Use Case Scenario	43
Table 10: "View Suggested Destinations" Use Case Scenario	44
Table 11: "View Chat History" Use Case Scenario	45
Table 12: "View Suggested Trips" Use Case Scenario	46
Table 13: "Book a Trip" Use Case Scenario	47
Table 14: "Make Payments" Use Case Scenario	48
Table 15: "Create a Trip" Use Case Scenario	49
Table 16: "Manage Trip" Use Case Scenario	50
Table 17: "View Booking" Use Case Scenario	51
Table 18: "View Payment Transactions" Use Case Scenario	52
Table 19: "Moderate Content" Use Case Scenario	53
Table 20: "Manage Users" Use Case Scenario	54
Table 21: "Add Tourism Company" Use Case Scenario	55
Table 22: "Add Governorates" Use Case Scenario	56
Table 23: "Add Hotels" Use Case Scenario	57
Table 24: "Add Landmarks" Use Case Scenario	58
Table 25: "Add Activities" Use Case Scenario	59
Table 26: "Add Events" Use Case Scenario	60

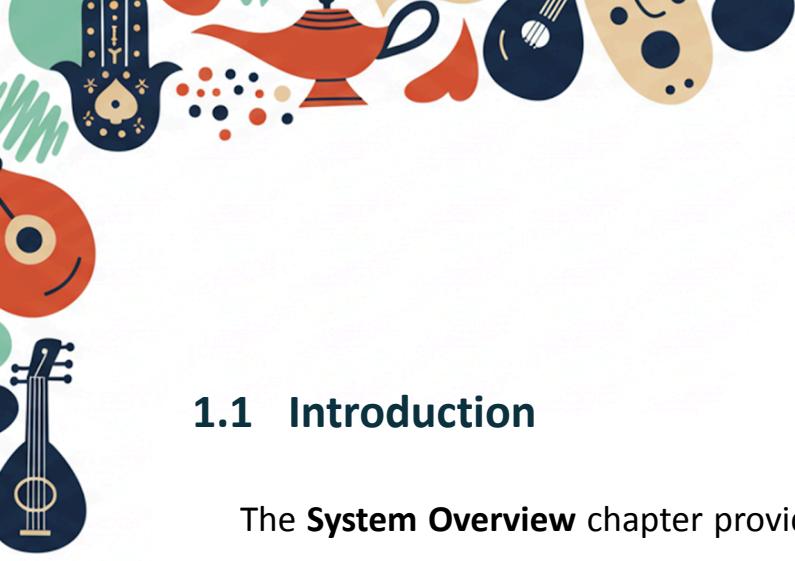
List of Figures

Figure 1: Kayak Explore: A Personalized Travel Discovery Tool	7
Figure 2: Routeperfect: Travel Itinerary Planning Platform	8
Figure 3: Viator: Global Travel Experiences Marketplace	10
Figure 4: GetYourGuide: A Global Platform for Booking Local Tours and Activities	11
Figure 5: Project Plan	21
Figure 6: Client Use Case Diagram	36
Figure 7: Admin Use Case Diagram	37
Figure 8: System Architecture	61
Figure 9: Swim lane Diagram	64
Figure 10: "Company" State Diagram	65
Figure 11: "Admin" State Diagram	65
Figure 12: Context Diagram	66
Figure 13: Data flow diagram	67
Figure 14: "NLP sequence" Sequence Diagram	68
Figure 15: "Payment" Sequence Diagram	69
Figure 16: "Manage trip" Sequence Diagram	70
Figure 17: Class Diagram	71
Figure 18: Database Schema	72
Figure 19: Home UI Mockup	73
Figure 20: Login UI Mockup	73
Figure 21: Chat UI Mockup	74
Figure 22: About UI Mockup	74



Chapter 1

System Overview



1.1 Introduction

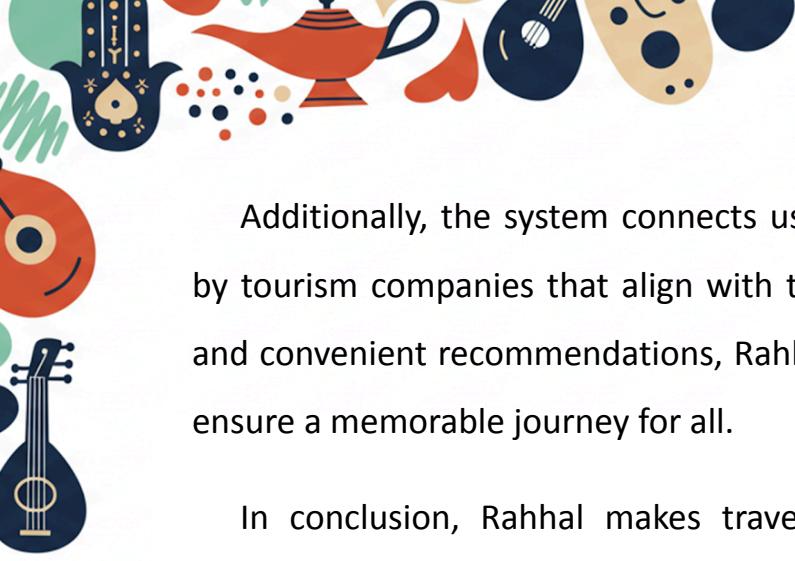
The **System Overview** chapter provides an in-depth look at the core aspects of Rahhal, a smart travel planning system tailored for tourists exploring Egypt. It explores the system's motivations, the challenges it addresses, and how it operates to offer personalized travel experiences. By focusing on delivering convenience, personalization, and efficiency, Rahhal aims to bridge the gap between travelers' unique needs and the vast opportunities Egypt has to offer.

This chapter will detail Rahhal's mission to simplify travel planning, its approach to tackling common travel problems, and the innovative features that set it apart. From identifying user preferences to connecting them with tailored recommendations and tourism company offerings, Rahhal ensures a seamless, enjoyable journey for both local and international tourists.

1.2 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.



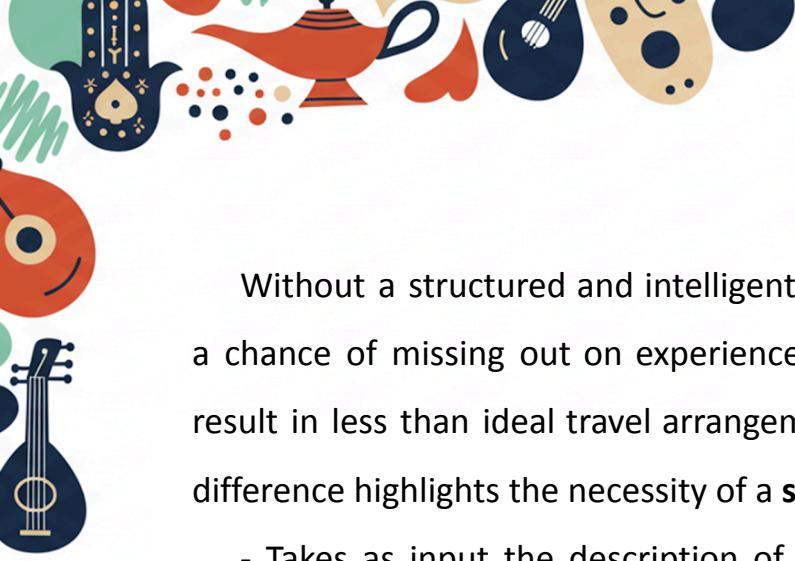
Additionally, the system connects users with recommended trips organized by tourism companies that align with their interests. By offering personalized and convenient recommendations, Rahhal aims to enhance travel planning and ensure a memorable journey for all.

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 citizens have better experiences while also supporting local tourism businesses. With Rahhal, exploring Egypt becomes simple, fun, and memorable for everyone.

1.3 Problem Statement

Egypt is a popular travel destination because of its great history, wide range of landmarks and unique cultural experiences. However, arranging a trip to Egypt might be difficult due to the huge number of cities landmarks, and activities offered. Tourists encounter a number of difficulties:

- 1. Limited Time and Resources:** Tourists often only have a few days to explore, making it difficult to rank the most worthwhile stops and activities.
- 2. Seasonal Preferences:** Tourists may not be aware of the fact that some places are more appropriate for particular seasons.
- 3. Individual Preferences:** Specific hobbies of tourists, including historical research or adventurous sports, can be missed by general planning techniques.
- 4. Information Overload:** Although there is a wealth of information available online, it takes time to locate recommendations that are precise, trustworthy, and customized
- 5. Inconsistency with Arranged Trips:** Tourism companies advertise a range of prearranged trips, but these sometimes don't fit the unique requirements, preferences, and limitations of individual tourists.



Without a structured and intelligent approach to trip planning, tourists run a chance of missing out on experiences that suit their interests, which could result in less than ideal travel arrangements and lower levels of pleasure. This difference highlights the necessity of a **smart, user-focused solution** that:

- Takes as input the description of a tourist's trip, including their budget, time limits, and preferences.
- Provides personalized destinations and activities depending on their specific requirements.
- Recommends pre-planned trips advertised by tourism companies that best meet their needs.
- Simplifies decision-making and improves the whole travel experience.

1.4 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.

1.5 Conclusion

In conclusion, the chapter highlights Rahhal's ability to revolutionize travel planning in Egypt. By addressing key challenges such as information overload, lack of personalization, and limited time, the system provides tourists with customized recommendations for cities, landmarks, accommodations, and trips.

Through its intelligent design, Rahhal not only enhances the travel experience for users but also supports the local tourism industry by facilitating direct connections between travelers and tourism companies. With its innovative features and user-centric approach, Rahhal sets the stage for memorable and efficient travel planning.

The next chapter, **Related Work**, will explore similar projects and systems in the travel and tourism domain, analyzing how they function and identifying what sets Rahhal apart. This comparison will demonstrate the unique value Rahhal brings to the tourism industry.



Chapter 2

Related Work

2.1 Introduction

The **Related Work** chapter explores existing platforms and systems in the travel and tourism industry that share similarities with Rahhal. By analyzing tools like Kayak Explore, Routeperfect, Viator, and GetYourGuide, this chapter identifies the strengths and limitations of current solutions. These platforms offer various features, such as destination discovery, itinerary planning, and activity booking, but often lack the level of personalization and AI-driven insights that Rahhal aims to provide. This chapter will highlight how Rahhal differentiates itself by combining user-driven trip planning with AI-powered recommendations, offering a more tailored and efficient travel planning experience.

2.2 Kayak Explore: A Personalized Travel Discovery Tool



Figure 1: Kayak Explore: A Personalized Travel Discovery Tool

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. (*Kayak Travel*, n.d.)

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.

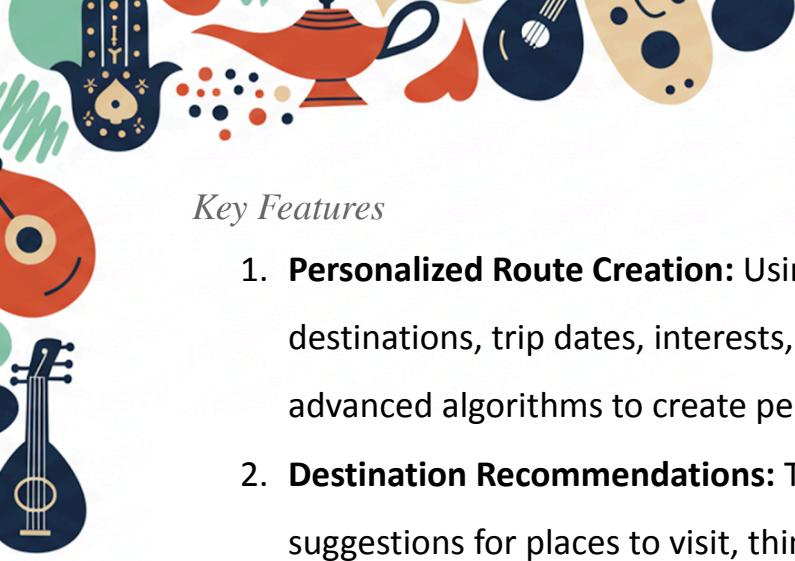
2. **Flexible Search:** The "Anywhere" feature enables users to investigate a broad array of destinations, utilizing filters to refine results based on flight duration, climate, and available activities.
3. **Cost Concerns:** helps with budget-based decision-making by providing predicted flight prices 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.

2.3 Routeprofet: Travel Itinerary Planning Platform



Figure 2: Routeprofet: 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. (*Routeprofet*, n.d.)



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.

2.4 Viator: Global Travel Experiences Marketplace

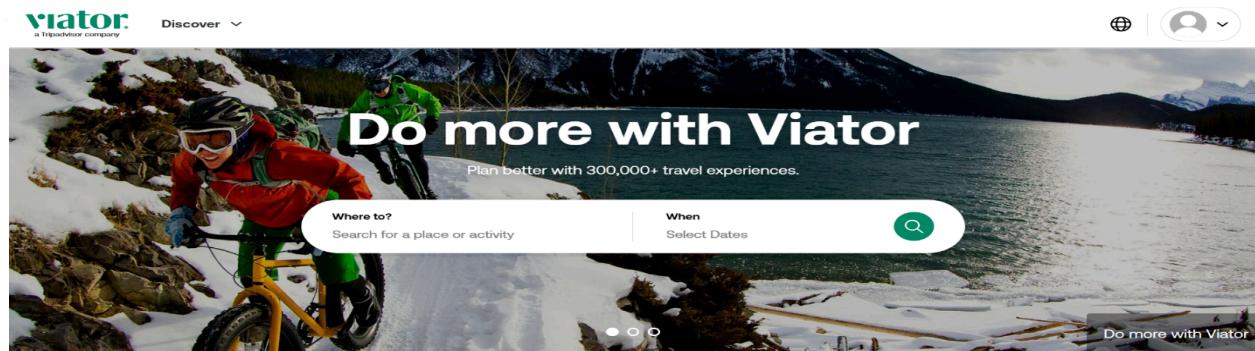


Figure 3: Viator: Global Travel Experiences Marketplace

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.

(*Viator*, n.d.)

Key Features

- 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.
- 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.
- 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.

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

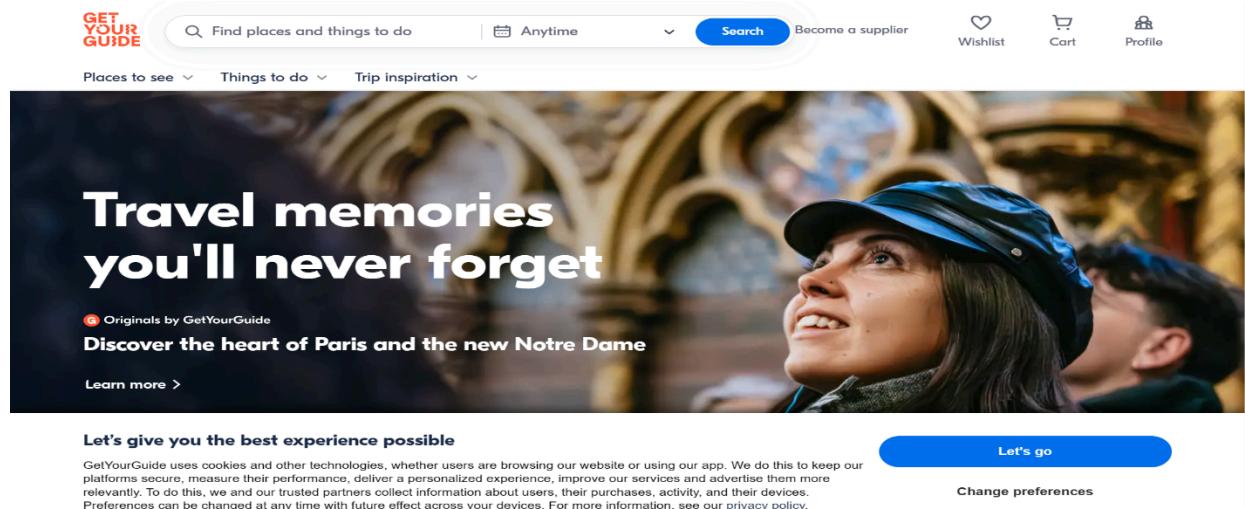


Figure 4: GetYourGuide: A Global Platform for Booking Local Tours and Activities

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. (*GetYourGuide*, n.d.)

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.

2.6 Our Project: A New Era of Travel Solutions

Previous projects faced several challenges and limitations that led us to develop our current project, aiming to provide a more personalized and enhanced user experience. Key issues identified include:

1. **Limited Personalization:** Most platforms offer generic recommendations without tailoring them to the user's specific interests and preferences.
2. **Complexity in Trip Planning:** Users often find it difficult to plan and organize trips due to scattered information and limited guidance.
3. **Limited Integration:** Lack of integration with multiple service providers for a seamless booking experience across different components of a trip



Our project introduces a unique blend of user-driven trip planning with AI-powered recommendation systems to offer a more personalized and efficient travel experience. Unlike traditional platforms, our system allows users to describe their trips in natural language, providing a comprehensive overview of their preferences, interests, and requirements. Leveraging machine learning and Natural Language Processing (NLP), we analyze user inputs to recommend tailored destinations, activities, and pre-designed trips from leading tourism companies.

This approach empowers travelers to plan their vacations more effectively, with the aid of user-centric feedback and advanced AI. Our chatbot facilitates intelligent decision-making by swiftly presenting curated vacations and activities from trusted sources.

Additionally, our system is flexible, accommodating a broad range of customer preferences—from families seeking all-inclusive packages to solo travelers exploring unique destinations. By optimizing travel dates and costs, we enhance the planning experience with personalized recommendations.

Ultimately, this innovative solution marks a significant advancement in travel technology, reshaping how people plan their vacations. It meets the needs of modern travelers while setting a new benchmark for future travel technology developments through the integration of AI insights and natural language input adaptability.

2.7 Conclusion

In conclusion, this chapter demonstrates that while existing platforms provide valuable tools for travel planning, they often fall short in delivering personalized, AI-driven recommendations. Rahhal stands out by leveraging Natural Language Processing (NLP) and machine learning to understand user preferences and generate tailored suggestions. By addressing the gaps in current systems, Rahhal offers a more user-centric and innovative approach to travel planning. The next chapter, **Domain Analysis and Technique**, will delve into the technical and operational aspects of Rahhal, exploring how the system is designed to meet the needs of its users and stakeholders.

Chapter 3

Domain Analysis and Technique

3.1 Introduction

The **Domain Analysis and Technique** chapter provides a detailed examination of the technical and operational foundations of Rahhal. It begins by analyzing the domain—identifying key users, their interactions, and the environment in which the system operates. This chapter also explores the risks and constraints associated with building a travel recommendation system, such as data accuracy, privacy concerns, and scalability. Additionally, it outlines the feasibility of the project, assessing whether the proposed solution is technically and operationally viable. Finally, the chapter introduces the quality assurance plan, ensuring that the system meets high standards of performance, reliability, and user satisfaction.

3.2 Domain Analysis

3.2.1 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.

3.2.2 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.

3.2.3 Tasks and Action 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.

3.3 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	<ul style="list-style-type: none"> - Partner with reliable tourism companies and government tourism boards to ensure accurate and up-to-date data. - Integrate third-party APIs (e.g., Google Places, TripAdvisor) to supplement data collection.
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.	Medium	<ul style="list-style-type: none"> - Continuously train and refine NLP models with diverse datasets to improve understanding of user inputs. - Use advanced algorithms to ensure relevant and personalized suggestions.
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	<ul style="list-style-type: none"> - Conduct regular security audits and penetration testing to identify and fix vulnerabilities. - Implement multi factor authentication (MFA) and role-based access control to restrict unauthorized access.

Table 1: Risks

Risk	Description	Impact	Likelihood	Mitigation
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.	Low	<ul style="list-style-type: none"> - Diversify data sources by including smaller and independent providers to ensure a wider range of options. - Implement user preference filters (e.g., budget, provider type) to offer more personalized and unbiased results.
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.	Low	<ul style="list-style-type: none"> - Regularly test the platform under high traffic conditions. - Use cloud-based infrastructure (e.g., AWS, Google Cloud) for dynamic scaling to handle traffic spikes. - Optimize database queries and implement caching mechanisms to improve performance.
Technical Failures	The platform may experience technical issues such as bugs, server downtime, or integration failures.	Disruption of service and frustrated users.	Medium	<ul style="list-style-type: none"> - Implement automated testing (unit, integration, and stress testing) to identify and fix bugs early. - Maintain a disaster recovery plan to quickly restore services in case of failures.

3.4 Project plan

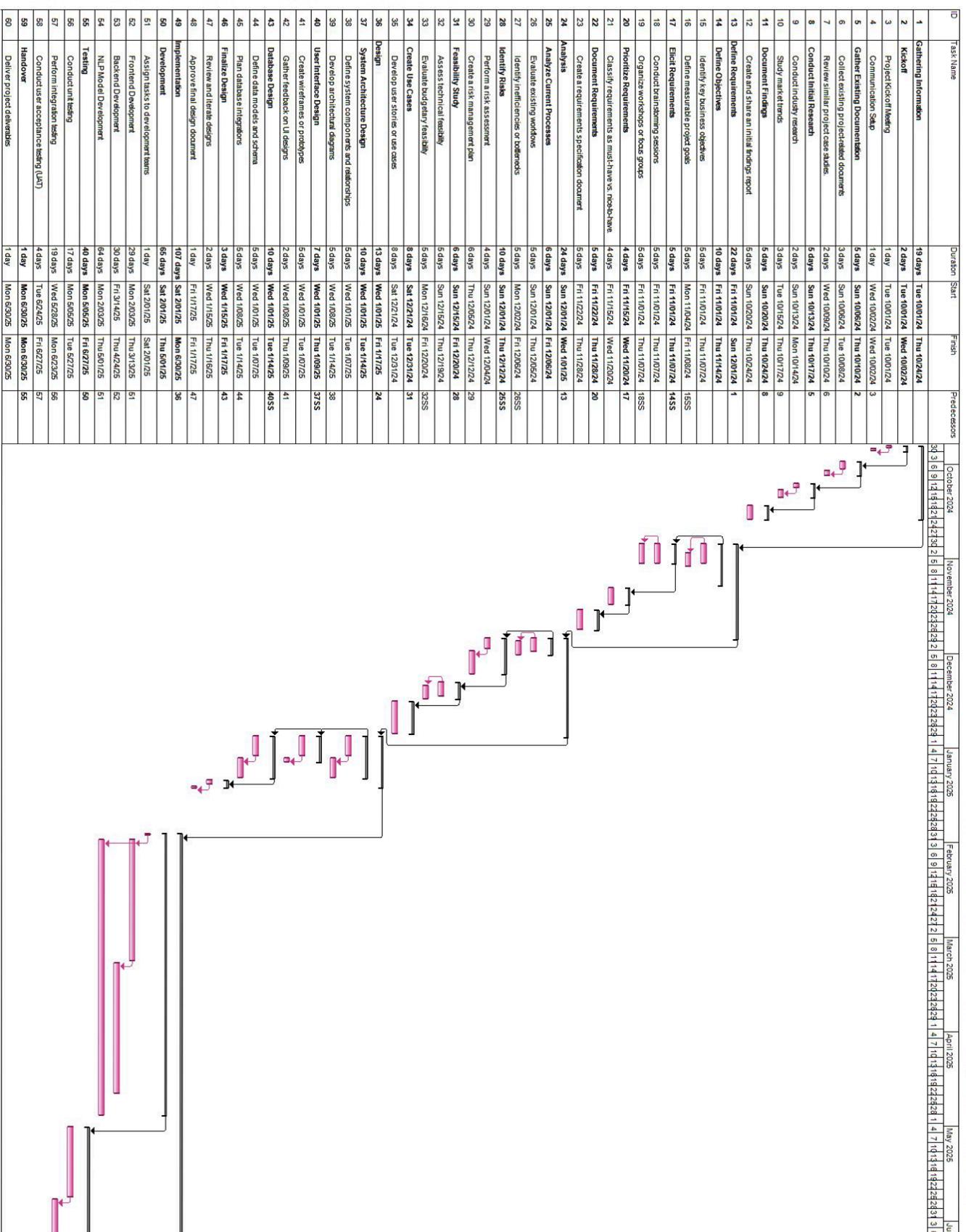


Figure 5: Project Plan

3.5 Feasibility Study

3.5.1 Technical Feasibility

SECTION	DESCRIPTION
Software Requirements	<ul style="list-style-type: none">- The platform will use React for the frontend, providing a responsive and interactive user interface.- Spring Boot will power the backend, offering a robust framework for developing secure and scalable RESTful APIs.- MySQL is selected as the relational database system for its reliability and ability to handle structured data efficiently.
Hardware Requirements	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- React is chosen for its component-based architecture and extensive library support, enabling fast and maintainable UI development.- Spring Boot is selected for its ability to simplify backend development and integrate seamlessly with MySQL.- MySQL is chosen for its efficiency in handling structured data, ensuring robust transaction management and data integrity.
Technical Expertise	<ul style="list-style-type: none">- 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	<ul style="list-style-type: none">- IDEs: IntelliJ IDEA, Visual Studio Code.- Version Control: Git.
Security Measures	<ul style="list-style-type: none">- SSL encryption for secure data transfer between the client and server.- Regular database backups and security patches to safeguard data integrity.

Table 2: Technical Feasibility Study

3.5.2 Operational Feasibility

SECTION	DESCRIPTION
User Acceptance Rate	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Anticipated 25% improvement in operational efficiency for tourism companies, streamlining processes such as trip creation, bookings, and user communication.
Information Accessibility	<ul style="list-style-type: none"> - 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	<ul style="list-style-type: none"> - Users (tourists and companies) are expected to adapt within 1-2 weeks, thanks to the platform's intuitive design and straightforward features.
User Satisfaction	<ul style="list-style-type: none"> - Satisfaction Goals: Targeting a 90% satisfaction rate post-implementation by focusing on personalized recommendations, efficient booking processes.

Table 3: Operational Feasibility Study

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.

3.7 System Requirements

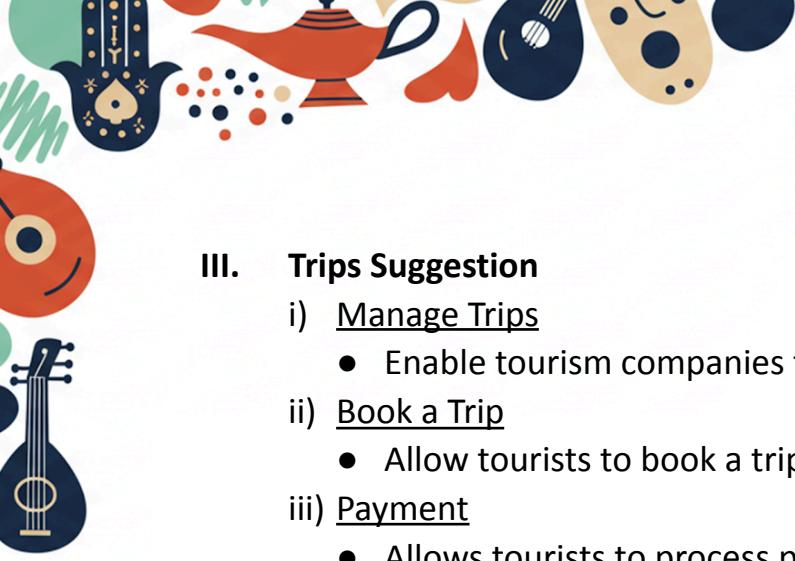
1. Functional Requirements

I. Chatbot Functionalities

- i) 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.
- ii) Preference Extraction:
 - Extract key features from the input text, such as destination type, activities, and time preferences.
- iii) Natural Language Understanding (NLU):
 - Handle synonyms and variations in user language.
- iv) Response Generation:
 - Respond with recommended destinations ranked by similarity scores.
 - Provide additional information about suggested destinations.

II. Recommendation System

- i) Data Matching:
 - Compare user preferences with destination data.
 - Compute similarity scores and rank destinations based on relevance.
- ii) Dynamic Suggestions:
 - Adapt recommendations based on real-time updates to user preferences.
- iii) Feedback Loop:
 - Allow users to provide feedback on recommendations to improve future suggestions.
 - Update destination data based on user interactions.



III. Trips Suggestion

i) Manage Trips

- Enable tourism companies to add, update, or delete trips.

ii) Book a Trip

- Allow tourists to book a trip if it has available slots.

iii) Payment

- Allows tourists to process payments for their bookings.
- Tourists can choose a payment method from various methods (e.g., credit card, PayPal).

IV. Destination Data Management

i) Data Storage:

- Maintain a database of destinations with metadata.

ii) Data Enrichment:

- Allow administrators to add or update destination details regularly.
- Include photos, descriptions, and links for more information.

iii) Categorization:

- Classify destinations into categories (e.g., beach, city, adventure) to assist in matching.

V. User Account Management

i) Registration and Login:

- Enable users to create accounts and log in securely.
- Store user preferences for personalized recommendations.

ii) Preference History:

- Allow users to view and update previously entered preferences.

VI. User Experience Enhancements

i) Interactive Filters:

- Allow users to refine results (e.g., filter by budget or location).

ii) Visual Feedback:

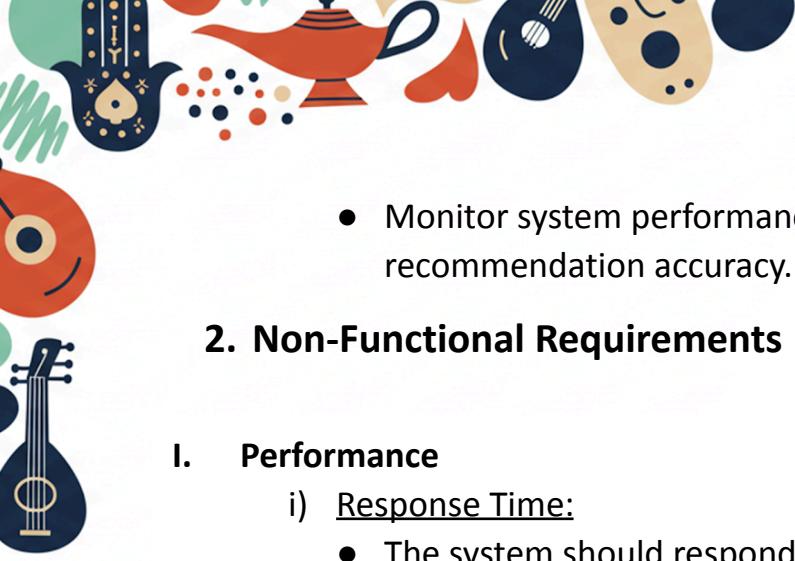
- Display recommendations with images, maps, and brief descriptions for each destination.

VII. Reporting and Analytics

i) User Insights:

- Generate reports on user preferences and popular destinations.

ii) Performance Metrics:

- 
- Monitor system performance, including response time and recommendation accuracy.

2. Non-Functional Requirements

I. Performance

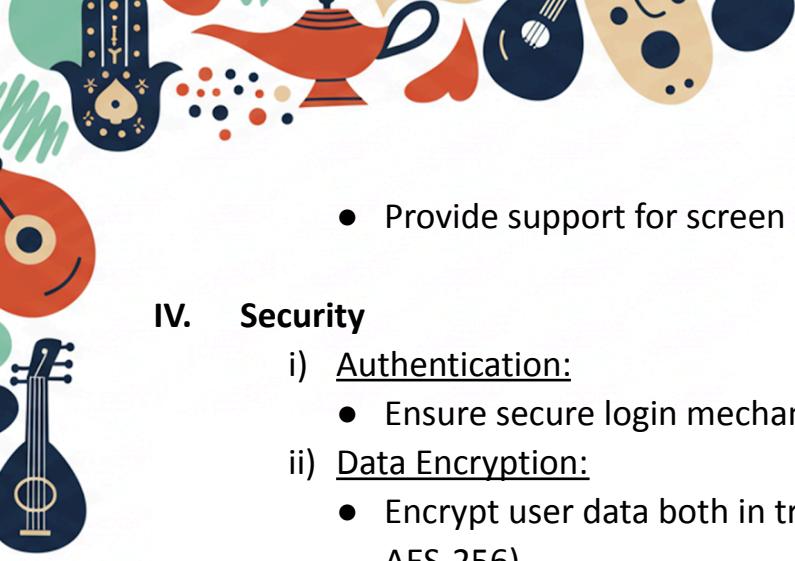
- i) Response Time:
 - The system should respond to user queries within 2 seconds for most inputs.
 - Recommendations should be generated in under 3 seconds after processing the user's preferences.
- ii) Scalability:
 - The system should handle at least 10,000 concurrent users without degradation in performance.
- iii) Throughput:
 - The system should support a high volume of transactions, such as processing 1,000 recommendations per minute.

II. Reliability

- i) Availability:
 - The system should have 99.9% uptime to ensure users can access the service at any time.
- ii) Fault Tolerance:
 - In case of a server or database failure, the system should automatically redirect traffic to backup servers.
- iii) Error Handling:
 - The system should provide clear error messages and recover gracefully from unexpected inputs or failures.

III. Usability

- i) Ease of Use:
 - The chatbot should be intuitive and require minimal learning effort.
 - Provide guided prompts or suggestions to help users frame their preferences.
- ii) Accessibility:
 - Ensure the interface complies with accessibility standards, such as WCAG 2.1.

- 
- Provide support for screen readers and keyboard navigation.

IV. Security

- i) Authentication:
 - Ensure secure login mechanisms with multi-factor authentication.
- ii) Data Encryption:
 - Encrypt user data both in transit (using HTTPS) and at rest (e.g., AES-256).
- iii) Privacy Compliance:
 - Provide users with options to view, update, or delete their personal data.

V. Maintainability

- i) Code Modularity:
 - Use a modular architecture to simplify maintenance and updates.
- ii) Documentation:
 - Provide detailed technical documentation for developers, including API specifications and deployment instructions.
- iii) Error Logging:
 - Implement comprehensive logging for debugging and monitoring purposes.

VI. Scalability

- i) Cloud Deployment:
 - Use cloud services (e.g., AWS, Google Cloud, or Azure) for dynamic scaling based on demand.

3.8 Techniques and tools

3.8.1 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.

3.8.2 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.

3.8.3 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.

3.8.4 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.

3.8.5 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.

3.9 Application Components

1. Frontend:

Tourist Interface:

- Login page.
- Chatbot for trip planning (AI-powered).
- Display of suggested travel plans, trips, hotels.
- Trip booking and payment page.
- User dashboard to manage bookings and tickets.

Tourism Company Interface:

- Login page.
- Dashboard to advertise new trips and manage trips.
- List of tourists who booked and their payment status.

Admin Interface:

- Dashboard for managing accounts and content.

- Dashboard for viewing bookings, payments, and tourism company interactions.
- Dashboard to enter data that used in AI model

2. Backend:

AI Model:

- Processes tourist input and generates personalized travel plans.
- Suggests trips, hotels, restaurants, and tourism companies.

Database:

- Stores data for tourists, tourism companies, trips, hotels, restaurants, bookings, and payments.

APIs to handle:

- Tourist queries and AI suggestions.
- Booking and payment processes.
- Admin management operations.

Booking and Payment System:

- Securely handle transactions.
- Manages bookings, payments, and generates unique ticket number for each confirmed payment.

Notifications System:

- Send notifications to tourists (e.g., booking confirmation, ticket details).

3.10 Conclusion

In conclusion, this chapter establishes the technical and operational feasibility of Rahhal, demonstrating that the system is well-equipped to handle the challenges of modern travel planning. By addressing risks such as data accuracy and scalability, and by implementing a robust quality assurance plan, Rahhal ensures a reliable and efficient user experience. The next chapter, **Proposed System & Methodology**, will provide a detailed blueprint of the system's architecture, use cases, and interaction models, showcasing how Rahhal integrates cutting-edge technology to deliver a seamless travel planning experience.



Chapter 4

Proposed System & Methodology

4.1 Introduction

The **Proposed System & Methodology** chapter serves as a comprehensive blueprint for the design, functionality, and operational framework of Rahhal. This section elaborates on the system's structure, including its core use cases, user scenarios, and technical components. By highlighting both client-side and admin-side functionalities, it demonstrates how the system meets the diverse needs of its users while maintaining seamless interaction and efficient management.

The chapter delves into the system's architecture, interaction flow, and underlying database schema, showcasing how data is processed, stored, and utilized to deliver personalized recommendations. It also incorporates design mockups to provide a visual understanding of the user interface, ensuring that both usability and functionality are well-aligned.

Through detailed diagrams, such as sequence, state, and data flow diagrams, the chapter provides a clear visualization of Rahhal's internal processes and user interactions. This structured approach illustrates how the system integrates cutting-edge technology to deliver an intuitive and efficient travel planning experience.

4.2 System Use-Cases

4.2.1: Client use-case

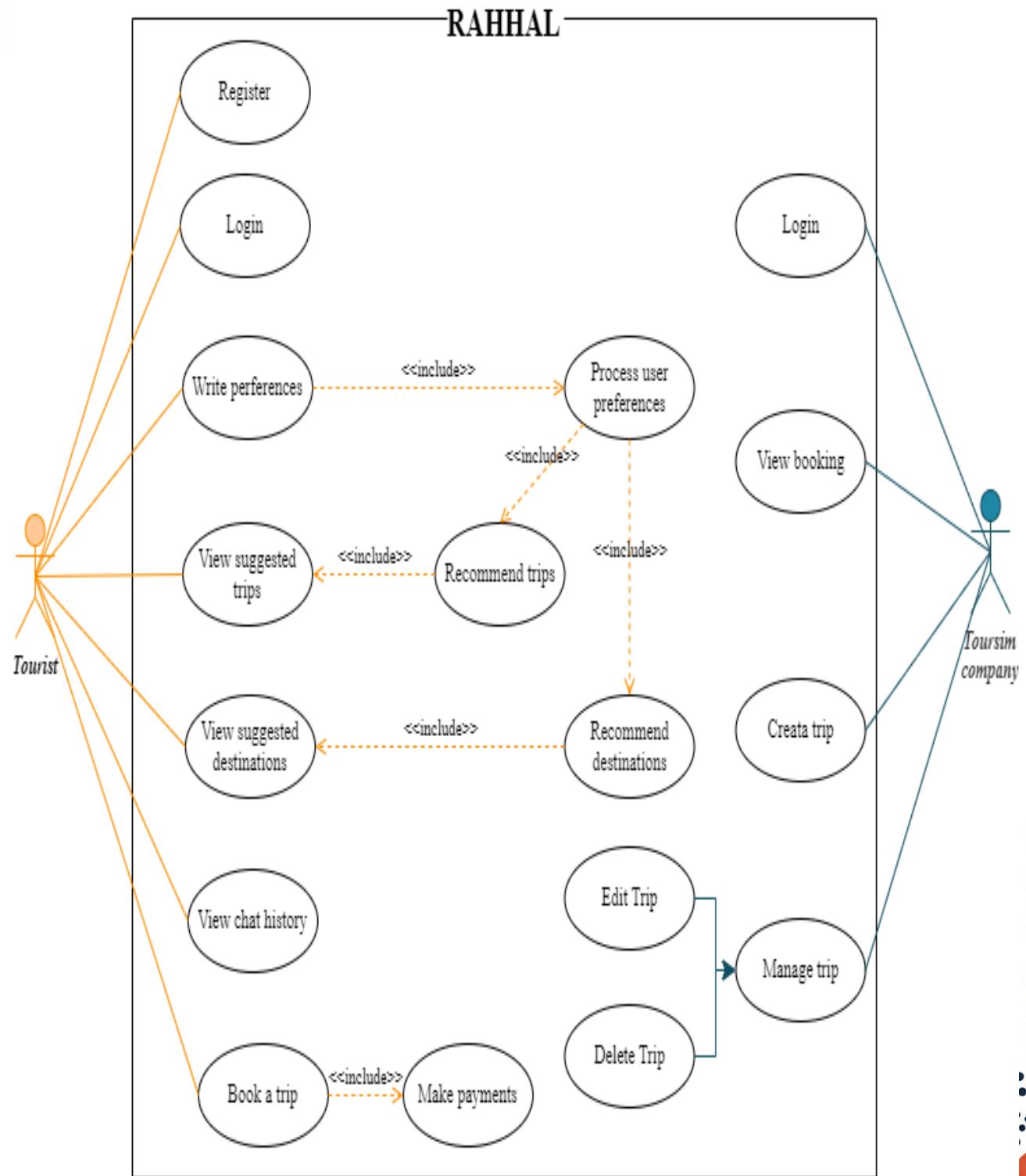


Figure 6: Client Use Case Diagram

4.2.2: Admin Use-Case

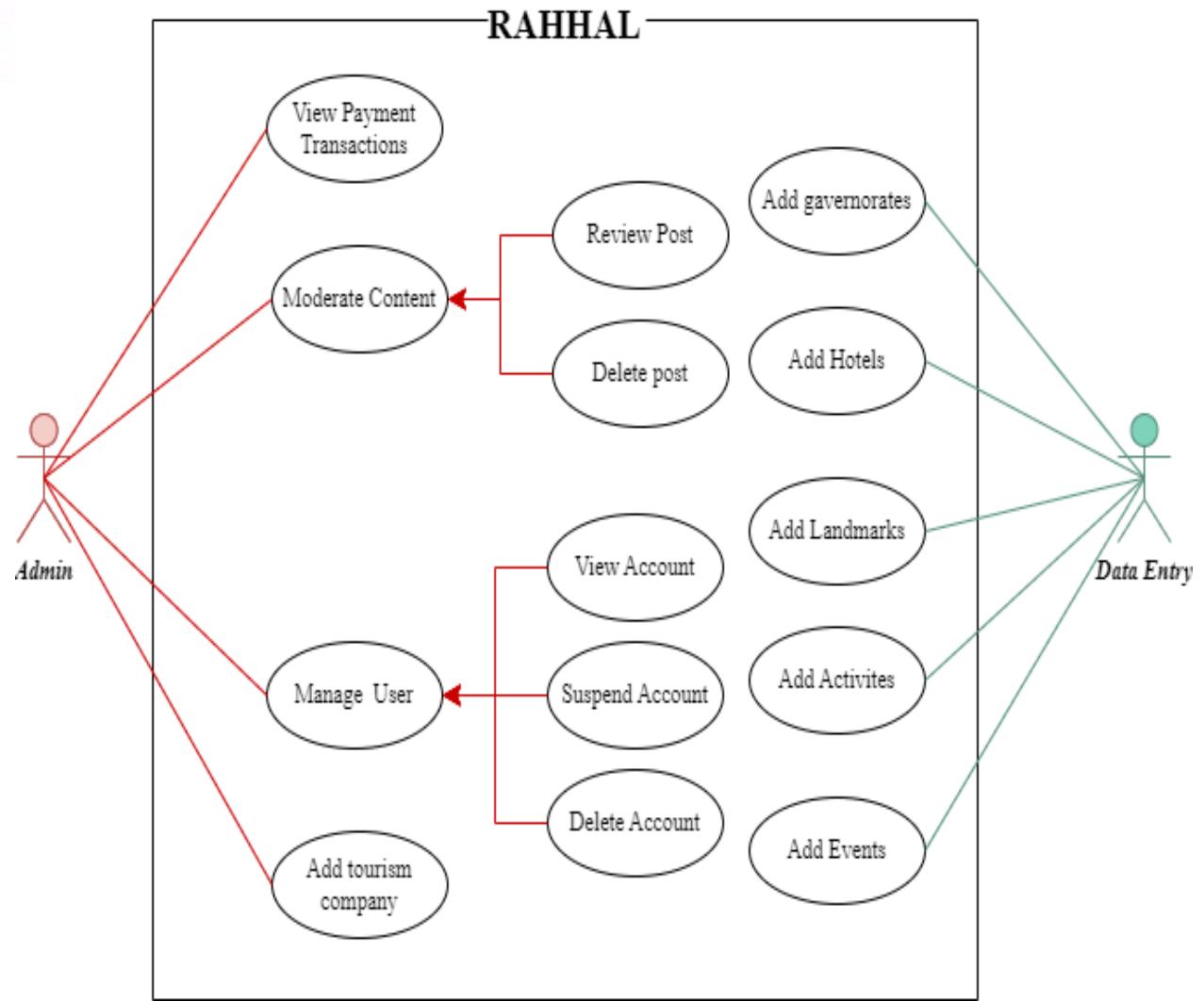


Figure 7: Admin Use Case Diagram

4.3 Use Case Description (Use case scenario)

4.3.1 Client-side scenarios

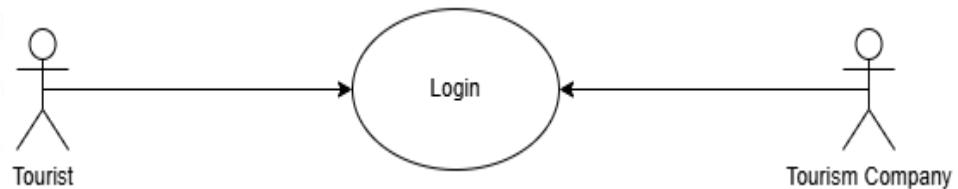
4.3.1.1 Register



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:	<ol style="list-style-type: none">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. <p>The system sends a confirmation email to the tourist.</p>
Alternative Flow:	If invalid data is provided, the system prompts for corrections.

Table 4: "Register" Use Case Scenario

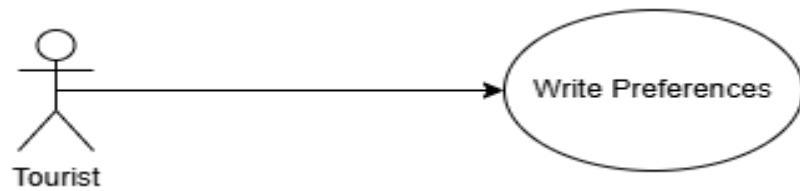
4.3.1.2 Login



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:	<ol style="list-style-type: none"> 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. <p>The system validates the credentials and grants access.</p>
Alternative Flow:	The system prompts the actor to retry or reset their password.

Table 5: "Login" Use Case Scenario

4.3.1.3 Write Preference



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:	<ul style="list-style-type: none"> • The tourist has logged into the system. • The preferences section is available.
Post conditions:	The preferences are saved successfully in the system.
Main Flow:	<ol style="list-style-type: none"> 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.

Table 6: "Write Preferences" Use Case Scenario

4.3.1.4 Process User Preferences



Use Case ID:	UC004
Use Case Name:	Process User Preferences
Area:	Personalization
Actor:	Preferences Extractor
Description:	Processes the preferences provided by tourists to extract key themes and keywords.
Preconditions:	<ul style="list-style-type: none"> The tourist has written their preferences. The system is connected to the natural language processing (NLP) engine.
Post conditions:	The extracted preferences are stored for recommendation purposes.
Main Flow:	<ol style="list-style-type: none"> The tourist writes their preferences in free text. The system processes the input using an NLP engine to identify themes and keywords. The system stores the extracted preferences in the database.
Alternative Flow:	If the input is incomplete, the system prompts the tourist to provide additional details.

Table 7: "Process User Preferences" Use Case Scenario

4.3.1.5 Recommend Destinations



Use Case ID:	UC005
Use Case Name:	Recommend Destinations
Area:	Personalization
Actor:	Trip Seeker
Description:	Uses tourist preferences to suggest destinations from the system database.
Preconditions:	<ul style="list-style-type: none"> User preferences are processed and stored. The system has access to a database of destinations.
Post conditions:	A list of recommended destinations is displayed to the tourist.
Main Flow:	<ol style="list-style-type: none"> The system retrieves the stored preferences. The system computes similarity between preferences and destinations in the database. The system ranks destinations based on relevance and displays them to the tourist.
Alternative Flow:	The system displays a message suggesting broader options, when there is no relevant destinations.

Table 8: "Recommend Destinations" Use Case Scenario

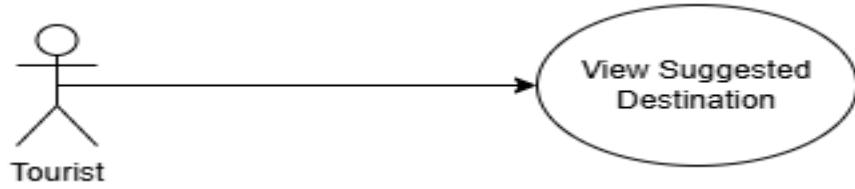
4.3.1.6 Recommend Trips



Use Case ID:	UC006
Use Case Name:	Recommend Trips
Area:	Trip Recommendation
Actor:	Trip Seeker
Description:	Suggests trips from the stored trips created by tourism companies.
Preconditions:	Trips are available in the system database.
Post conditions:	A list of recommended trips is displayed to the tourist.
Main Flow:	<ol style="list-style-type: none"> 1. The system retrieves trips created by tourism companies from the database. 2. The system ranks trips based on popularity, reviews, and relevance. 3. The system displays the ranked trips to the tourist.
Alternative Flow:	The system displays a message suggesting broader options, when no relevant trips available.

Table 9: "Recommend Trips" Use Case Scenario

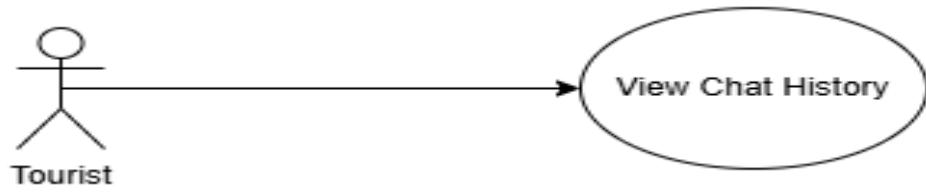
4.3.1.7 View Suggested Destinations



Use Case ID:	UC007
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:	<ul style="list-style-type: none"> The tourist has saved preferences. The system has generated destination suggestions.
Post conditions:	The tourist views a list of suggested destinations.
Main Flow:	<ol style="list-style-type: none"> The tourist logs into the system. The system retrieves destination suggestions based on the tourist's preferences. 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.

Table 10: "View Suggested Destinations" Use Case Scenario

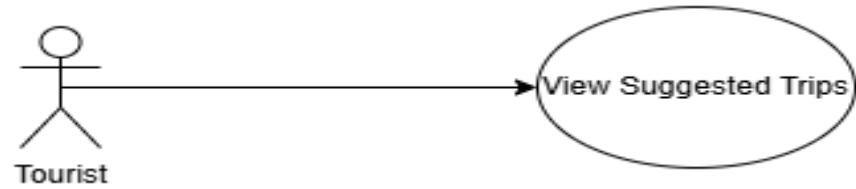
4.3.1.8 View Chat History



Use Case ID:	UC008
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:	<ol style="list-style-type: none">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.

Table 11: "View Chat History" Use Case Scenario

4.3.1.9 View Suggested Trips



Use Case ID:	UC009
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:	<ol style="list-style-type: none"> 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 at the current time.

Table 12: "View Suggested Trips" Use Case Scenario

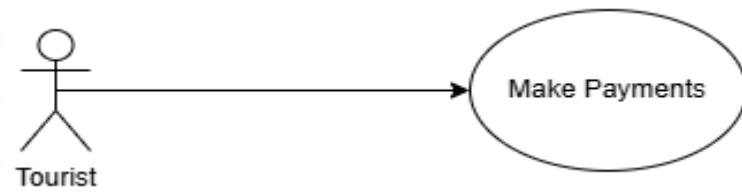
4.3.1.10 Book a Trip



Use Case ID:	UC010
Use Case Name:	Book a Trip
Area:	Booking
Actor:	Tourist
Description:	Allows tourists to select and book trips offered by the system.
Preconditions:	<ul style="list-style-type: none"> The tourist has selected a trip. The trip has available slots.
Post conditions:	The trip is booked, and the payment is processed.
Main Flow:	<ol style="list-style-type: none"> The tourist selects a trip from the suggested list. The system displays the trip details, including price and itinerary. The tourist clicks the "Book Now" button. The system navigates to the payment page. The tourist completes the payment (triggers the "Make Payments" use case). The system confirms the booking and generates a receipt.
Alternative Flow:	<ul style="list-style-type: none"> The system informs the tourist and suggests alternate trips. If payment fails, the system prompts the tourist to retry or use another method.

Table 13: "Book a Trip" Use Case Scenario

4.3.1.11 Make Payments



Use Case ID:	UC011
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:	<ol style="list-style-type: none"> 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.

Table 14: "Make Payments" Use Case Scenario

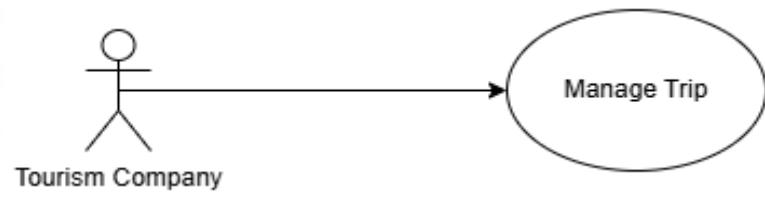
4.3.1.12 Create a Trip



Use Case ID:	UC012
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:	<ol style="list-style-type: none"> 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.

Table 15: "Create a Trip" Use Case Scenario

4.3.1.13 Manage Trip



Use Case ID:	UC013
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:	<ol style="list-style-type: none"> 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.

Table 16: "Manage Trip" Use Case Scenario

4.3.1.14 View Booking



Use Case ID:	UC014
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:	<ol style="list-style-type: none"> 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.

Table 17: "View Booking" Use Case Scenario

4.3.2 Admin-side scenario

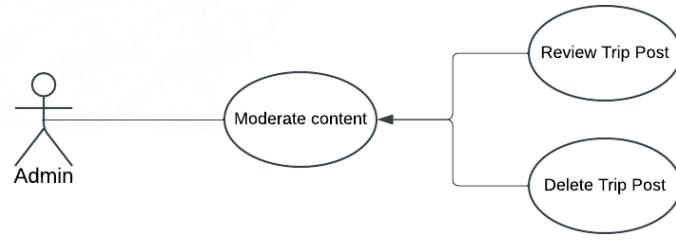
4.3.2.1 View Payment Transactions



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 can solve errors in the payment process if they exist.
Main Flow:	<ol style="list-style-type: none"> 1. The admin navigates to the Transactions section. 2. Filters transactions by date, user, type or payment ID. 3. Review 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.

Table 18: "View Payment Transactions" Use Case Scenario

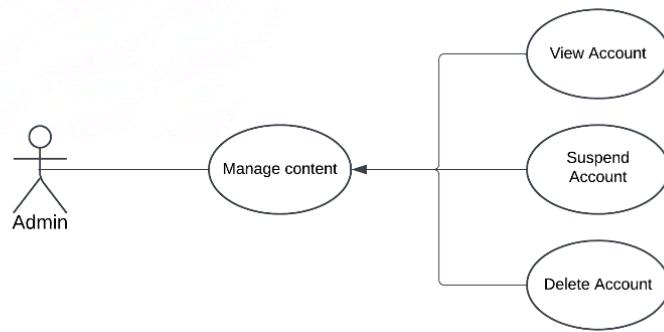
4.3.2.2 Moderate Content



Use Case ID:	UC_012
Use Case Name:	Moderate content
Area:	Content Management
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 has missing necessary data, the admin rejects it and issues a warning to the tourism company.

Table 19: "Moderate Content" Use Case Scenario

4.3.2.3 Manage Users



Use Case ID:	UC_013
Use Case Name:	Manage Users
Area:	User Management
Actor:	Admin
Description:	The admin views 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- Save 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

Table 20: "Manage Users" Use Case Scenario

4.3.2.4 Add Tourism Company



Use Case ID:	UC_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.

Table 21: "Add Tourism Company" Use Case Scenario

4.3.2.5 Add Governorates



Use Case ID:	UC_015
Use Case Name:	Add Governorates
Area:	Data Management
Actor:	Data Entry
Description:	The Data Entry adds government information
Preconditions:	<ul style="list-style-type: none"> 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:	<ul style="list-style-type: none"> 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.

Table 22: "Add Governorates" Use Case Scenario

4.3.2.6 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.

Table 23: "Add Hotels" Use Case Scenario

4.3.2.7 Add Landmarks



Use Case ID:	UC_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 are incomplete, the Data Entry role saves it as a draft for later completion.

Table 24: "Add Landmarks" Use Case Scenario

4.3.2.8 Add Activities



Use Case ID:	UC_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 are correct and available.
Post conditions:	Activity details added to the database.
Main Flow:	1- The admin navigates to the Add Activity section. 2- Enter 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.

Table 25: "Add Activities" Use Case Scenario

4.3.2.9 Add Events



Use Case ID:	UC_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 are 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.

Table 26: "Add Events" Use Case Scenario

4.4 System Architecture

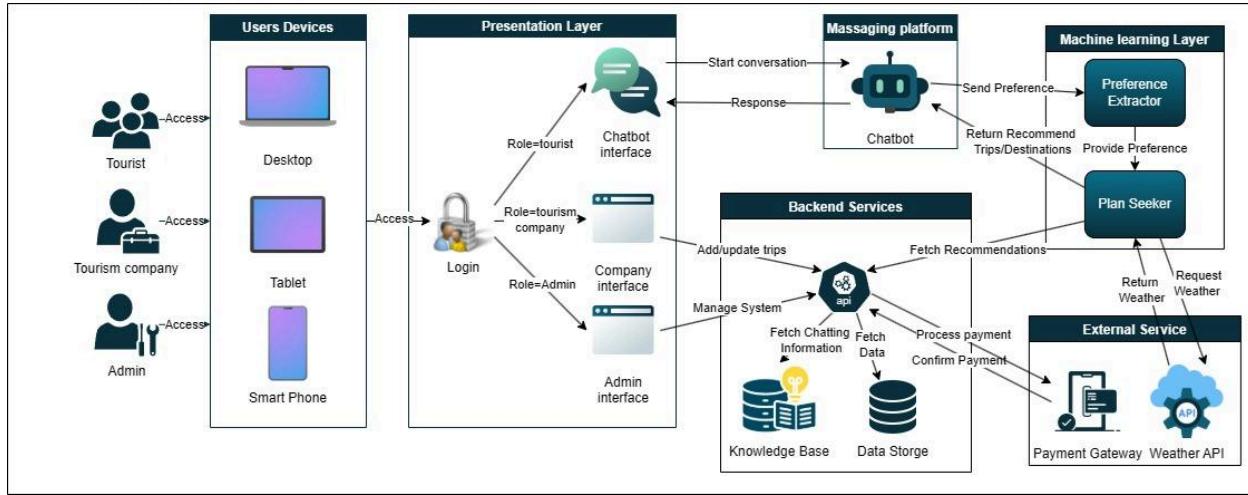


Figure 8: System Architecture

4.4.1 Methodology

Our project adopts the **Waterfall Methodology**, a linear and structured approach to software development. This methodology divides the development process into distinct phases, each completed sequentially. Given the nature of our project, the Waterfall approach ensures clarity, thorough documentation, and systematic progress through predefined stages. Below, we outline the phases, focusing on the progress achieved thus far.

1. Requirement Analysis

- This phase involved gathering and analyzing detailed project requirements from stakeholders, including users and tourism companies.
- Techniques such as surveys, and competitive analysis were used to capture the functional and non-functional requirements.
- The outcome was a comprehensive **Requirement Specification Document**, which serves as the foundation for subsequent phases.

2. System Design

- We have successfully completed the **System Design** phase.
- During this phase, we transformed the requirements into a structured system architecture:
 - Created **use case diagrams, class diagrams, sequence diagrams**, and a detailed **database schema**.
 - Defined the integration of key technologies such as **React.js, Spring Boot, MySQL, RASA, and FastAPI**.
 - Developed mockups for the system's user interface to ensure alignment with user expectations.
- This phase provided a clear blueprint for the implementation phase.

3. Implementation (Upcoming Phase)

- The next step involves developing the system based on the design documents.
- Key tasks include:
 - Frontend development using React.js.
 - Backend implementation using Spring Boot.
 - Integration of Machine Learning and NLP components for recommendation features using RASA and FastAPI.

4. Integration and Testing

- After implementation, the system components will be integrated and tested as a unified platform.
- Testing will include:
 - **Unit Testing** for individual modules.

- **Integration Testing** to validate interactions between components.
- **System Testing** to ensure the complete system meets requirements.

5. Deployment

- In this phase, the system will be deployed to a live environment for real-world use.
- Deployment will include server setup, database configuration, and user access testing.

6. Maintenance

- Following deployment, the system will enter the maintenance phase to address any issues and accommodate evolving requirements.
- This includes updates, optimizations, and user feedback incorporation.

Current Status and Justification

We have successfully completed the **System Design** phase and are transitioning to the **Implementation** phase. The **Waterfall Methodology** was chosen for its structured approach, which ensures each phase is thoroughly completed and documented before proceeding. This approach is particularly suitable as we have clearly defined requirements and a comprehensive system design, minimizing the risk of major changes during development.

Our progress so far demonstrates a strong foundation for building a robust and innovative travel solution.

4.5 Analysis Class

4.5.1 Swim Lane Diagram

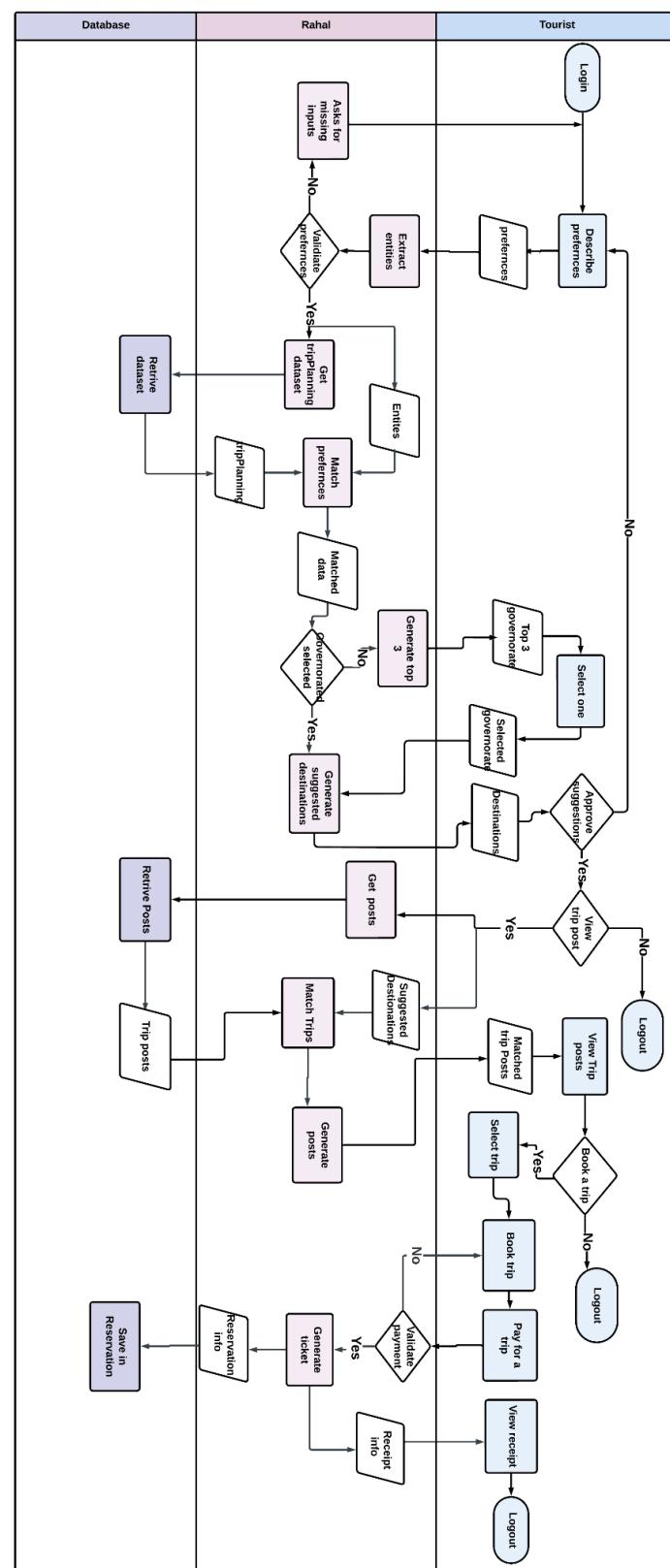


Figure 9: Swim Lane Diagram

4.5.2 State Diagram

4.5.2.1 State for company

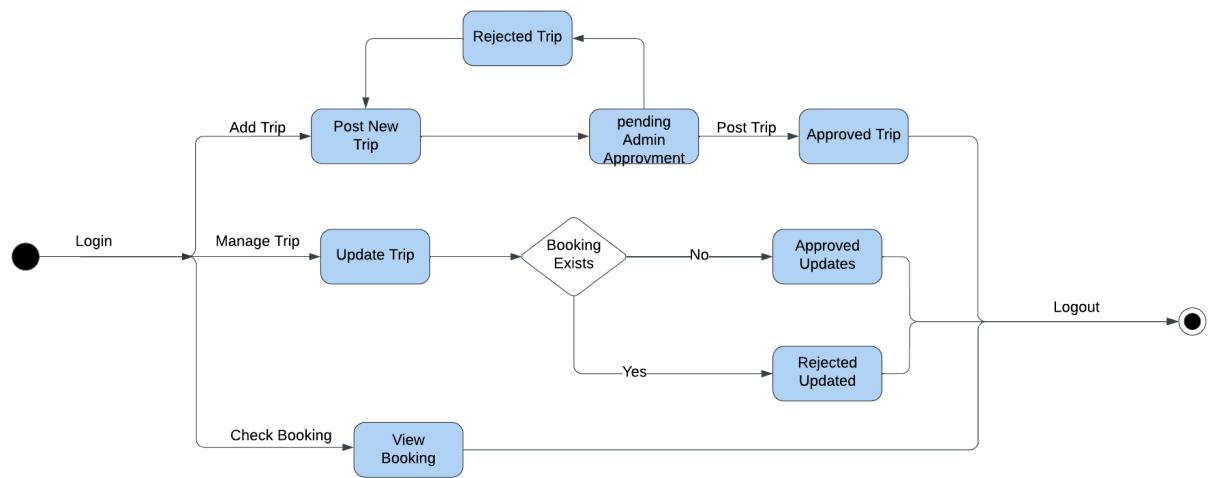


Figure 10: "Company" State Diagram

4.5.2.2 State for Admin

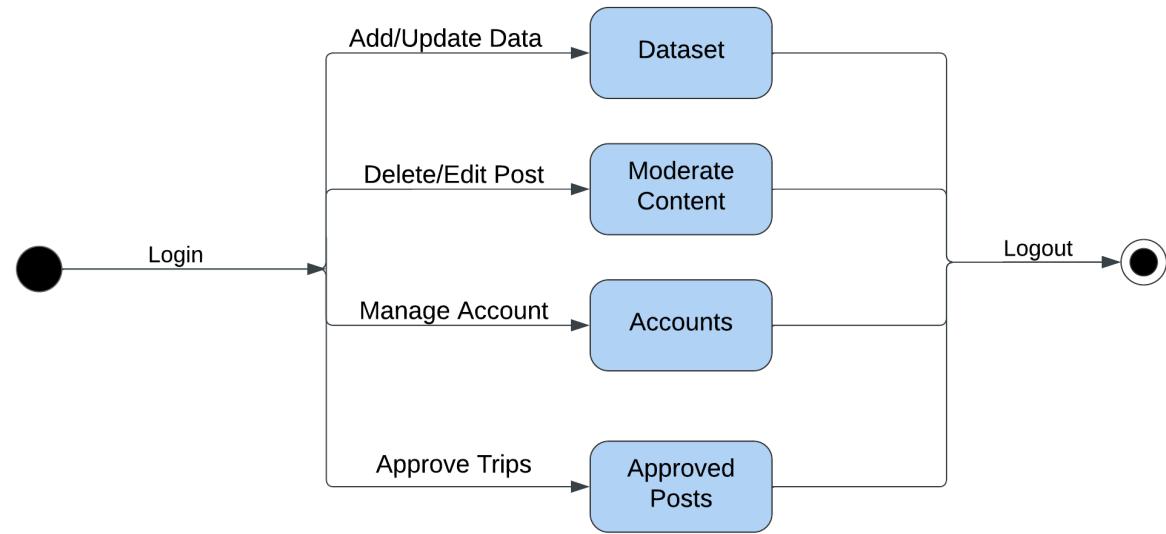


Figure 11: "Admin" State Diagram

4.5.2 Context Diagram

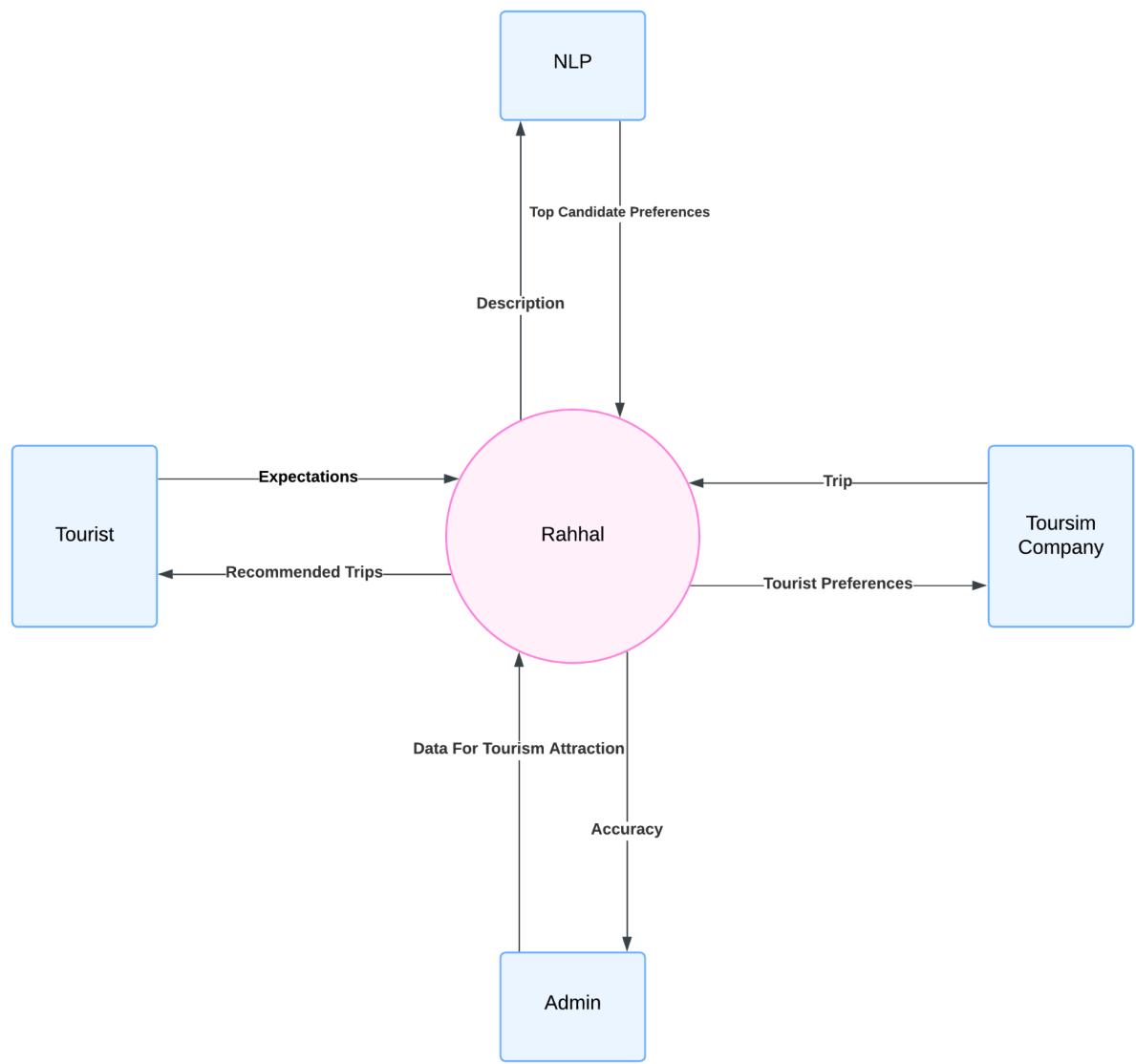


Figure 12: Context Diagram

4.5.2 Data flow diagram

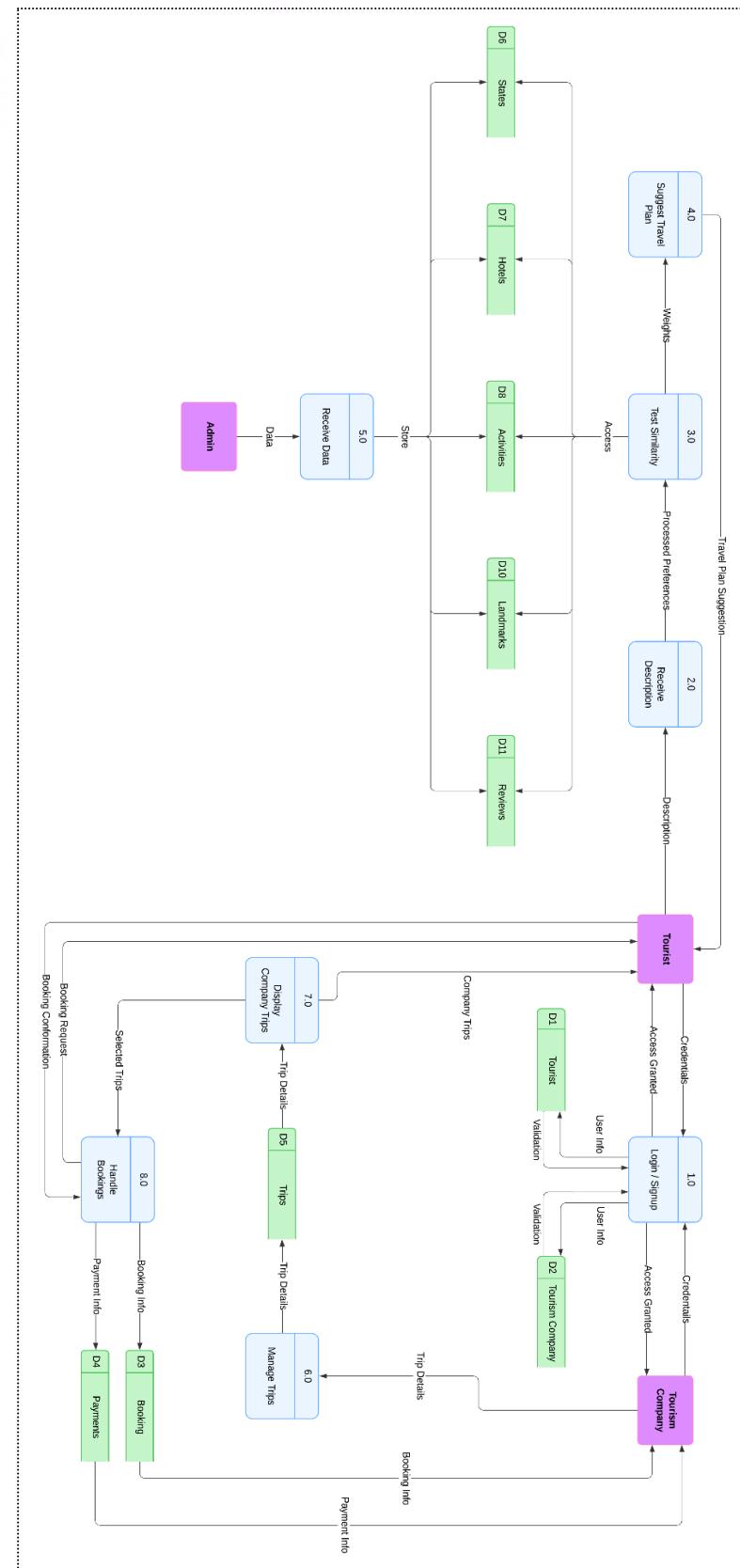


Figure 13: Data Flow Diagram

4.6 interaction class diagram

4.6.1 Sequence Diagram

4.6.1.1 NLP sequence

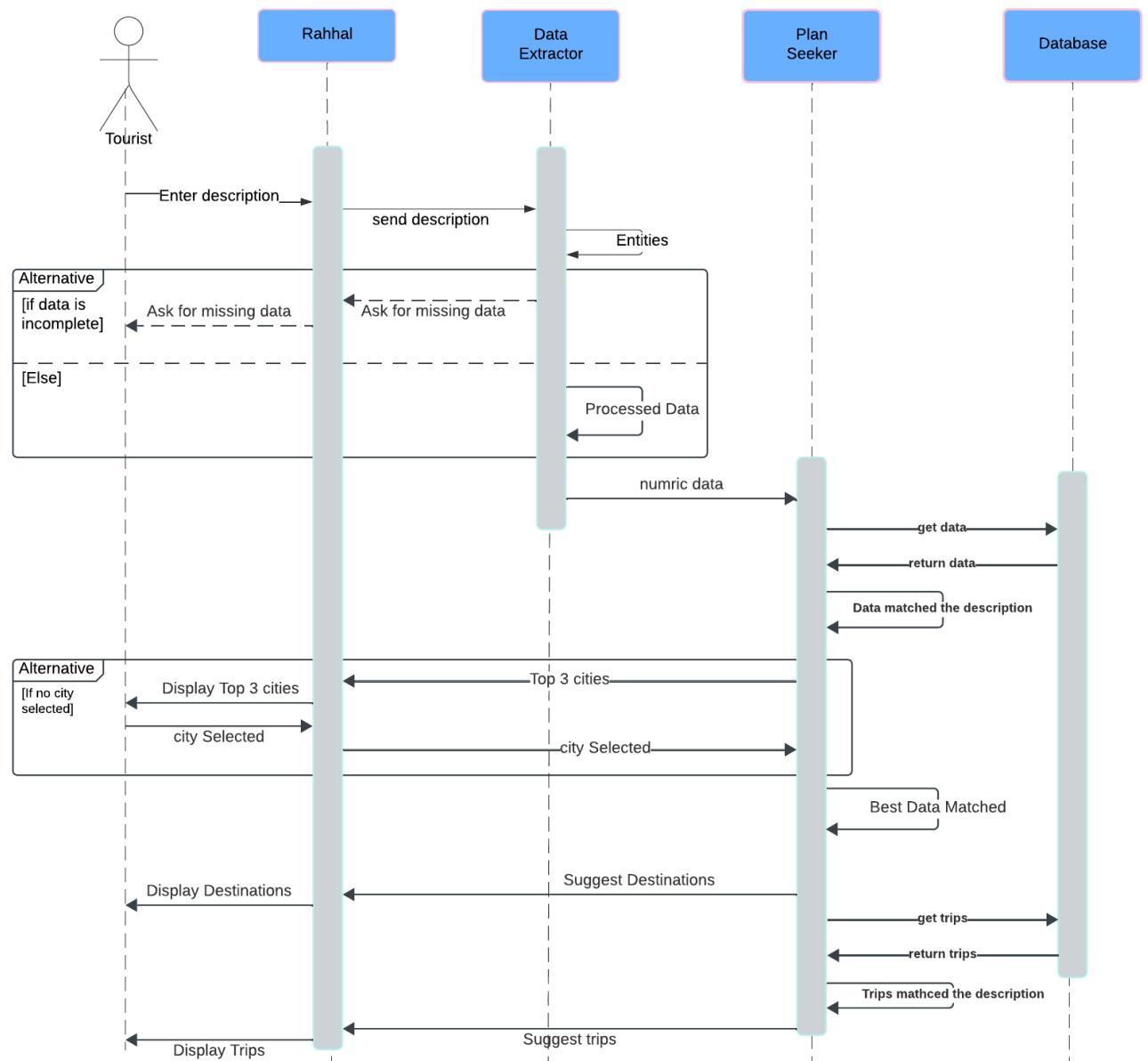


Figure 14: "NLP sequence" Sequence Diagram

4.6.1.2 Payment

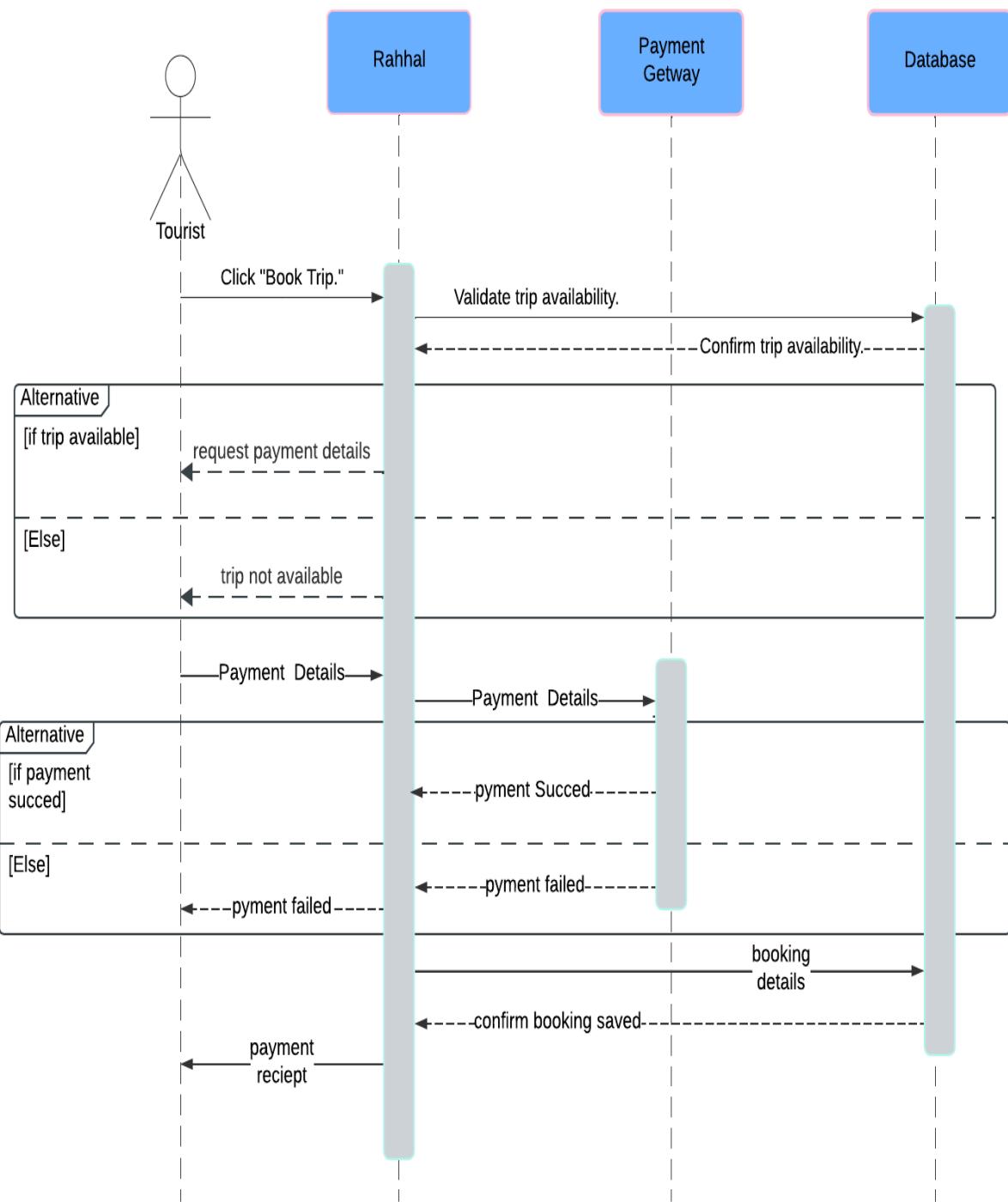


Figure 15: "Payment" Sequence Diagram

4.6.1.3 Manage trip

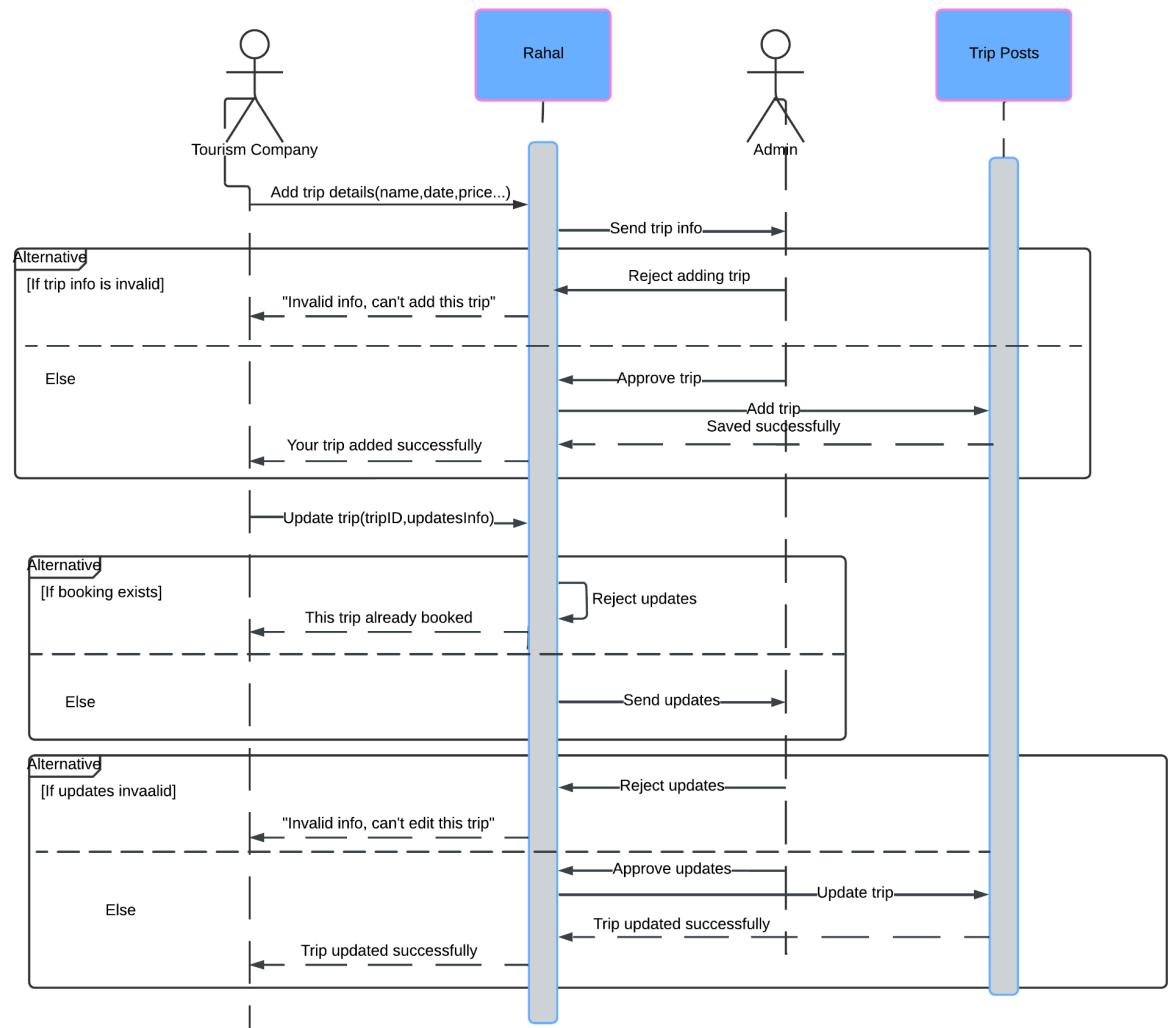


Figure 16: "Manage trip" Sequence Diagram

4.7 Design Class

4.7.1 Class Diagram

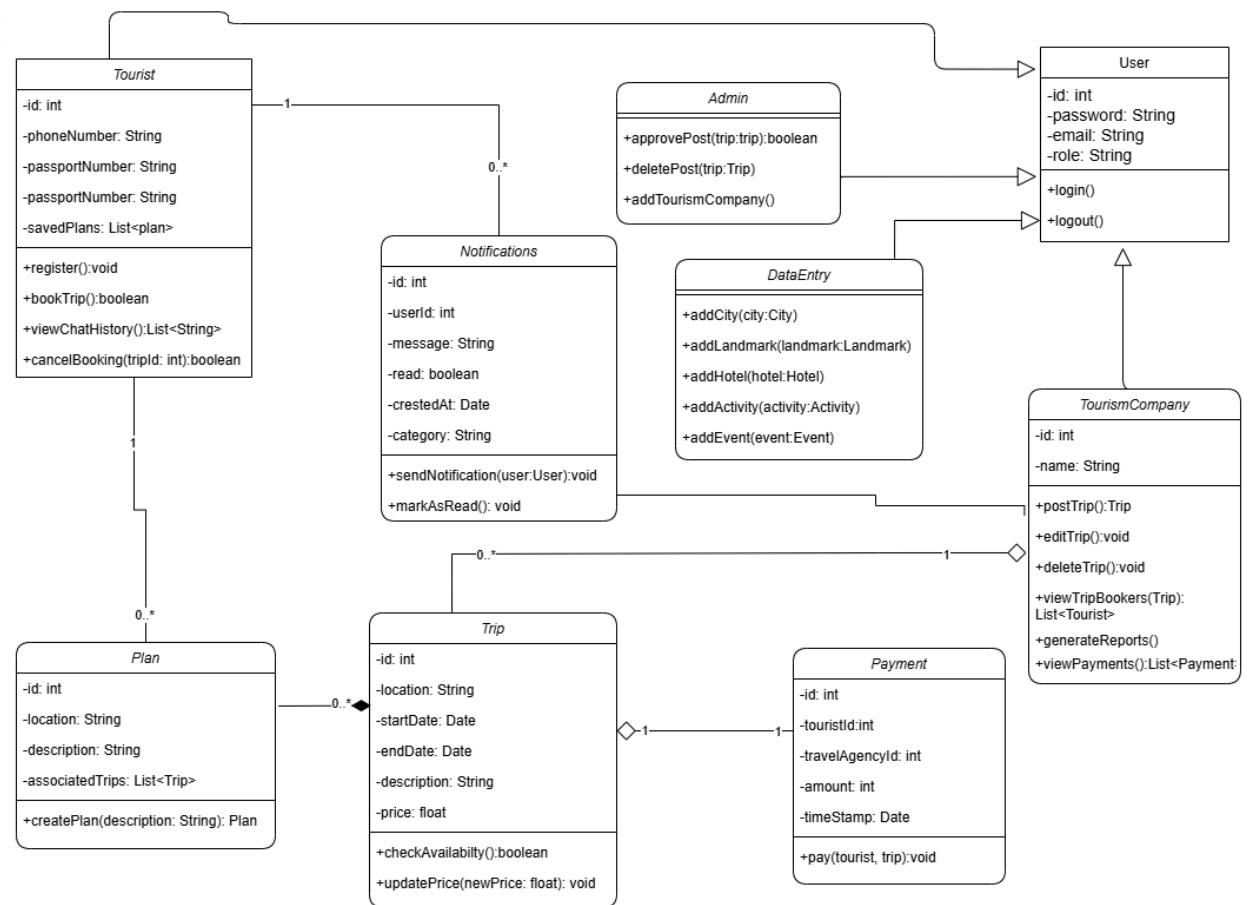


Figure 17: Class Diagram

4.8 Database Schema

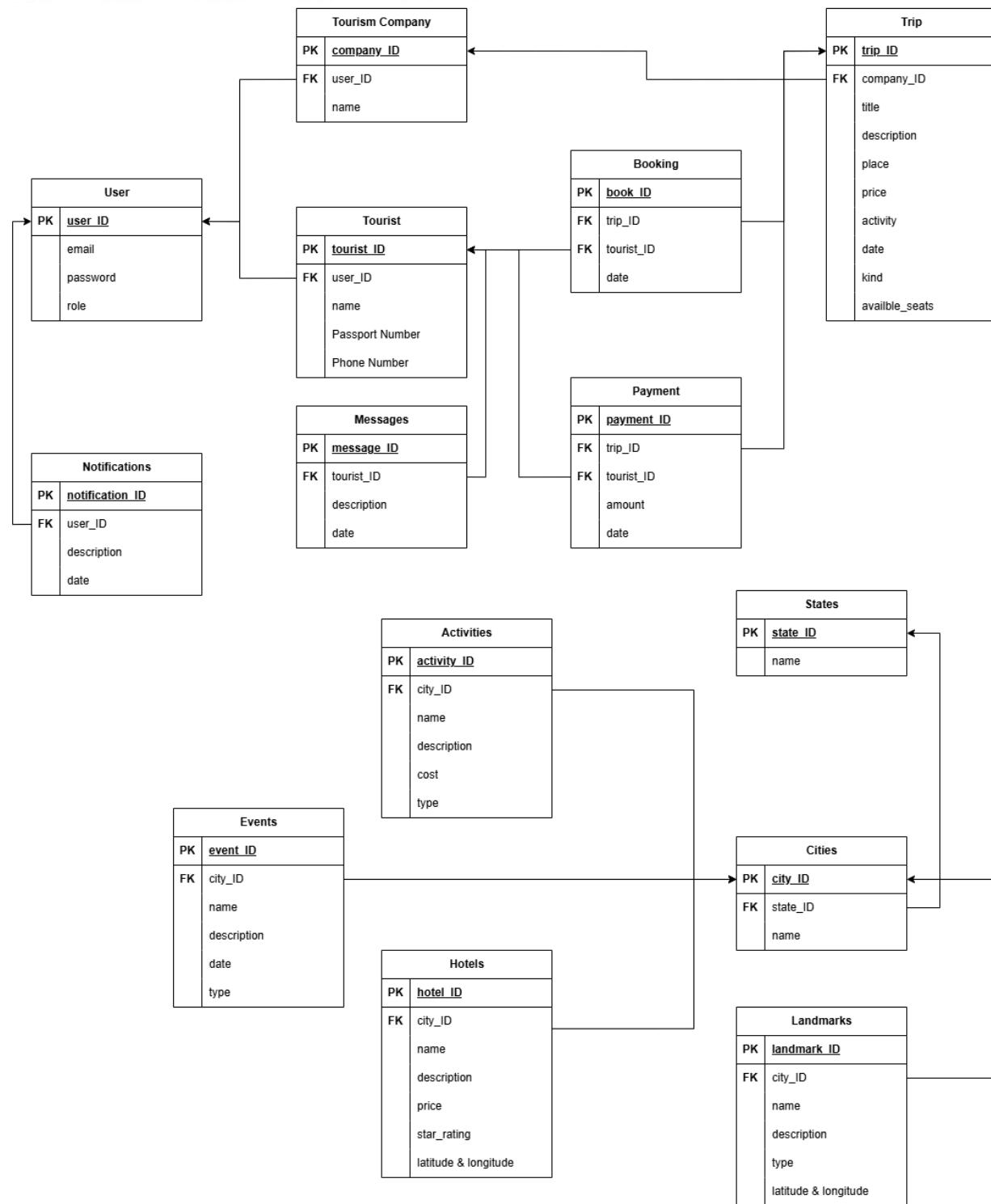


Figure 18: Database Schema

4.9 Design Mockup

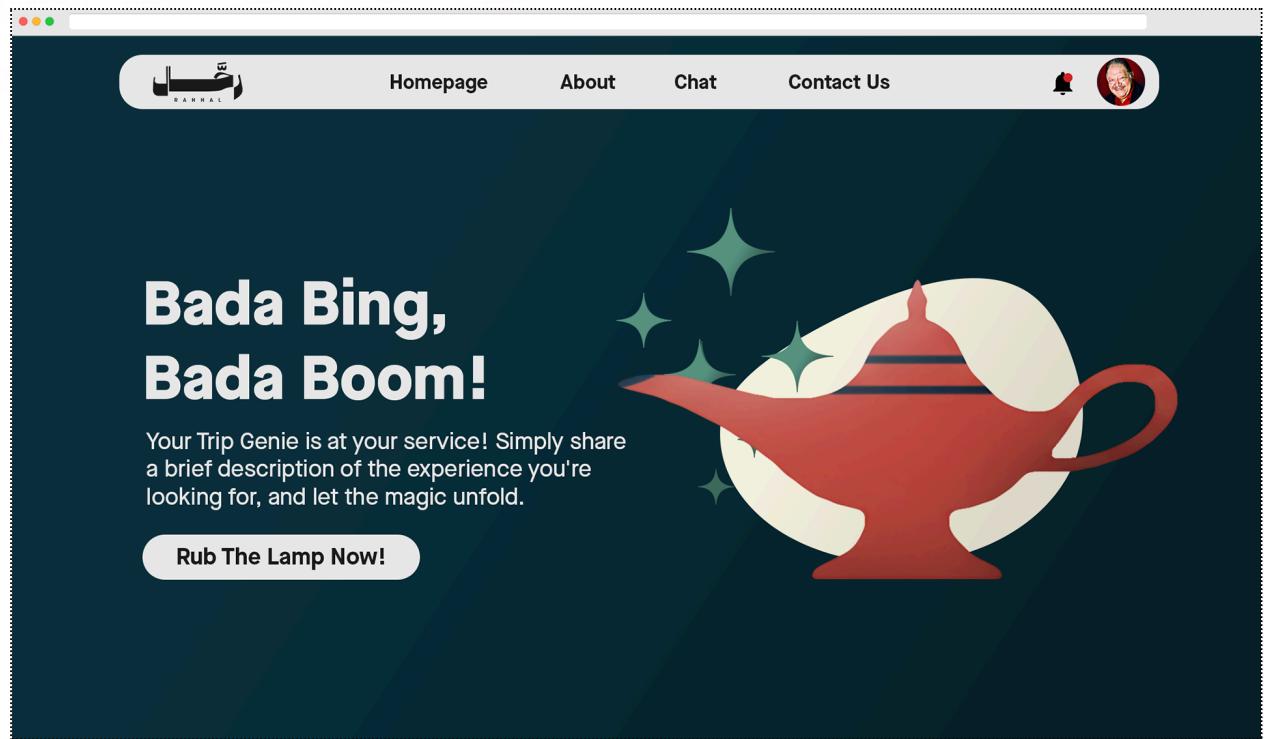


Figure 19: Home UI Mockup

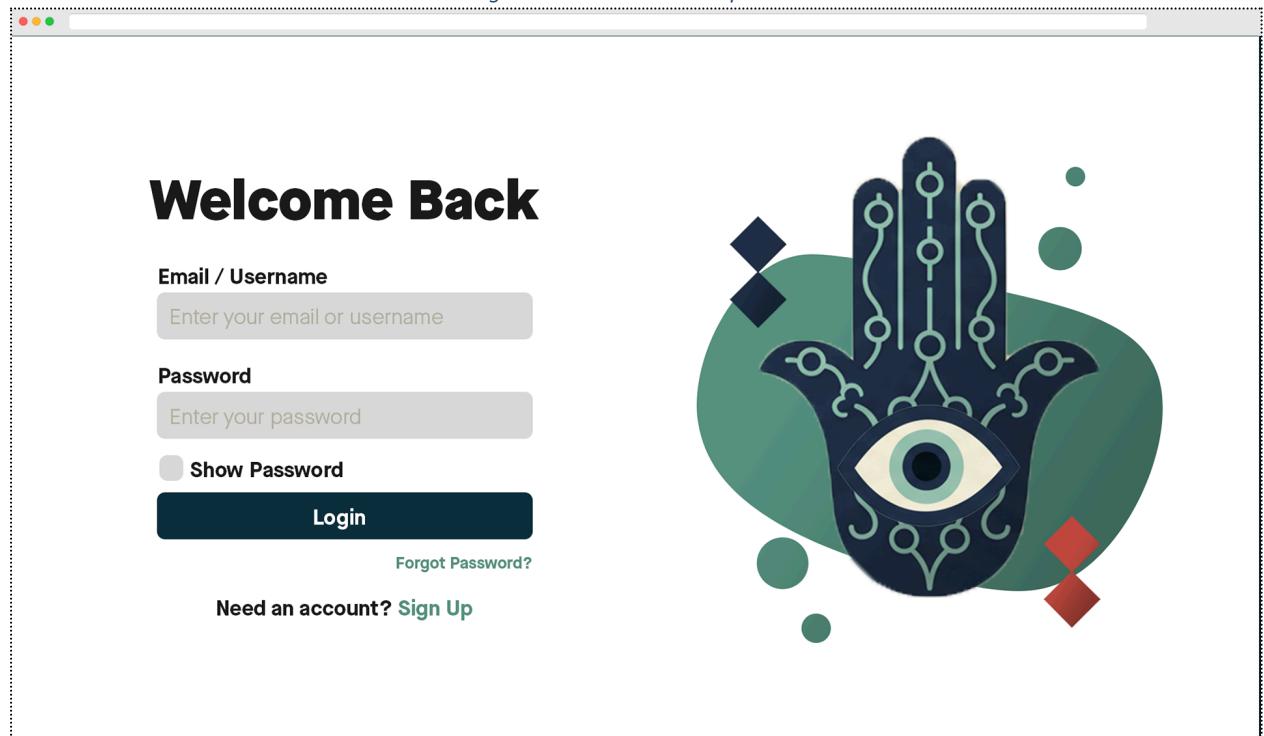


Figure 20: Login UI Mockup

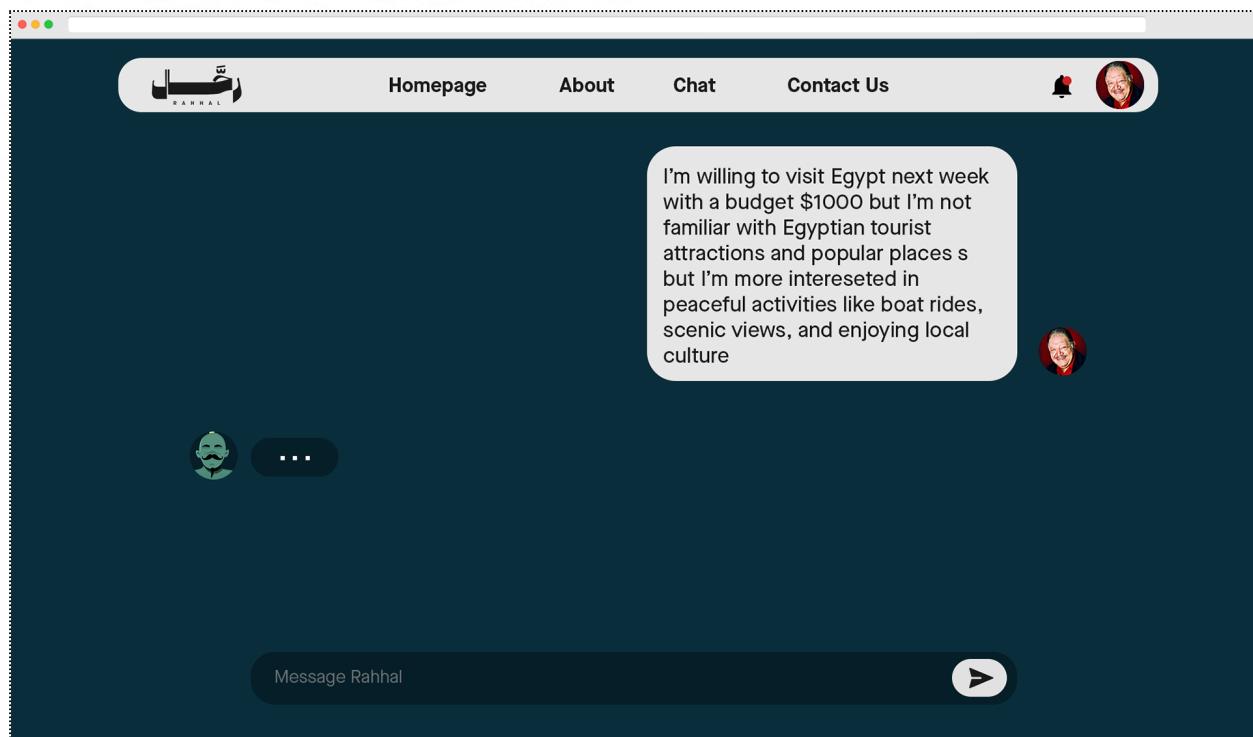


Figure 21: Chat UI Mockup

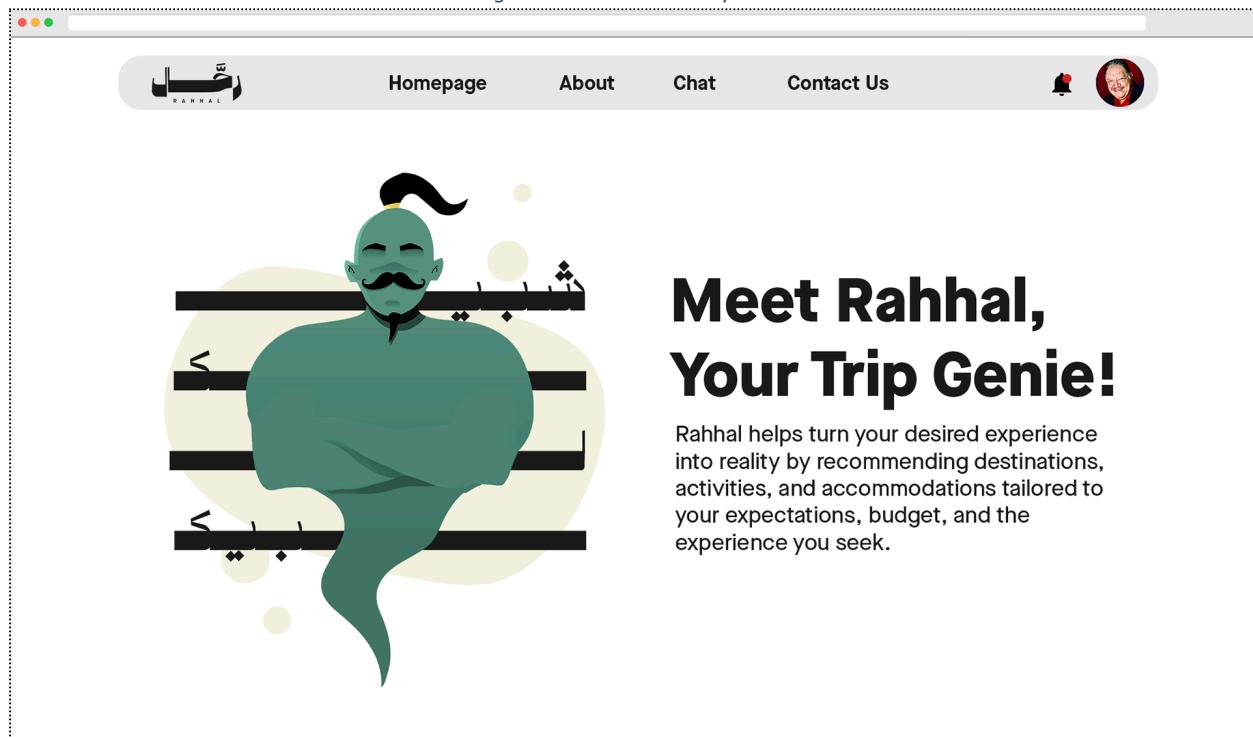


Figure 22: About UI Mockup

4.10 Conclusion

In conclusion, this chapter outlines the technical backbone and operational flow of Rahhal. By breaking down use cases, system architecture, and interaction models, it offers a detailed view of how the system functions effectively for both tourists and administrators.

The chapter highlights the seamless integration of NLP for personalized recommendations, secure payment processing, and robust content management. It also emphasizes the importance of a well-structured database and user-centric design mockups to enhance usability and efficiency.

The next chapter, **Conclusion & Future Work**, will provide a summary of the project's vision and achievements while exploring potential improvements and future enhancements to expand Rahhal's capabilities and impact in the tourism industry.



Chapter 5

Conclusion & Feature Work

5.1 Conclusions

Rahhal appears as a pioneering solution in the tourism industry, seamlessly blending artificial intelligence with travel planning to revolutionize how tourists explore destinations. By providing tailored itineraries, accommodation options, recommendations, and secure booking capabilities, Rahhal transforms the often complex process of trip planning into a simple and enjoyable experience. At its core, Rahhal serves as more than just a travel planner; it acts as a comprehensive platform connecting tourists with trusted tourism companies. The AI-powered chatbot personalizes recommendations based on user preferences, ensuring relevance and satisfaction. Meanwhile, its robust backend infrastructure allows admins to manage data efficiently, create new trips, and oversee user payments, maintaining operational excellence. A standout feature of Rahhal is its ticketing system, which generates unique identifiers for every booking, ensuring smooth operations and transparency. Furthermore, its secure payment integration builds trust, while the intuitive interface encourages seamless engagement from both tourists and administrators. With its innovative design, Rahhal not only caters to individual travelers but also enhances collaboration between admins and tourism companies, contributing to a dynamic and responsive ecosystem. The project sets a solid foundation for the future of smart, AI-driven tourism.

5.2 Future work

The future roadmap for Rahhal aims to expand its capabilities and enhance the user experience.

Key areas of development include:

AI Optimization:

Incorporating advanced machine learning techniques to provide more precise and predictive trip recommendations.

Adding sentiment analysis to better understand user preferences and feedback.

Mobile Application Development:

Launching a dedicated mobile app for Rahhal to cater to tourists planning trips on the go.

Global Support and Localization:

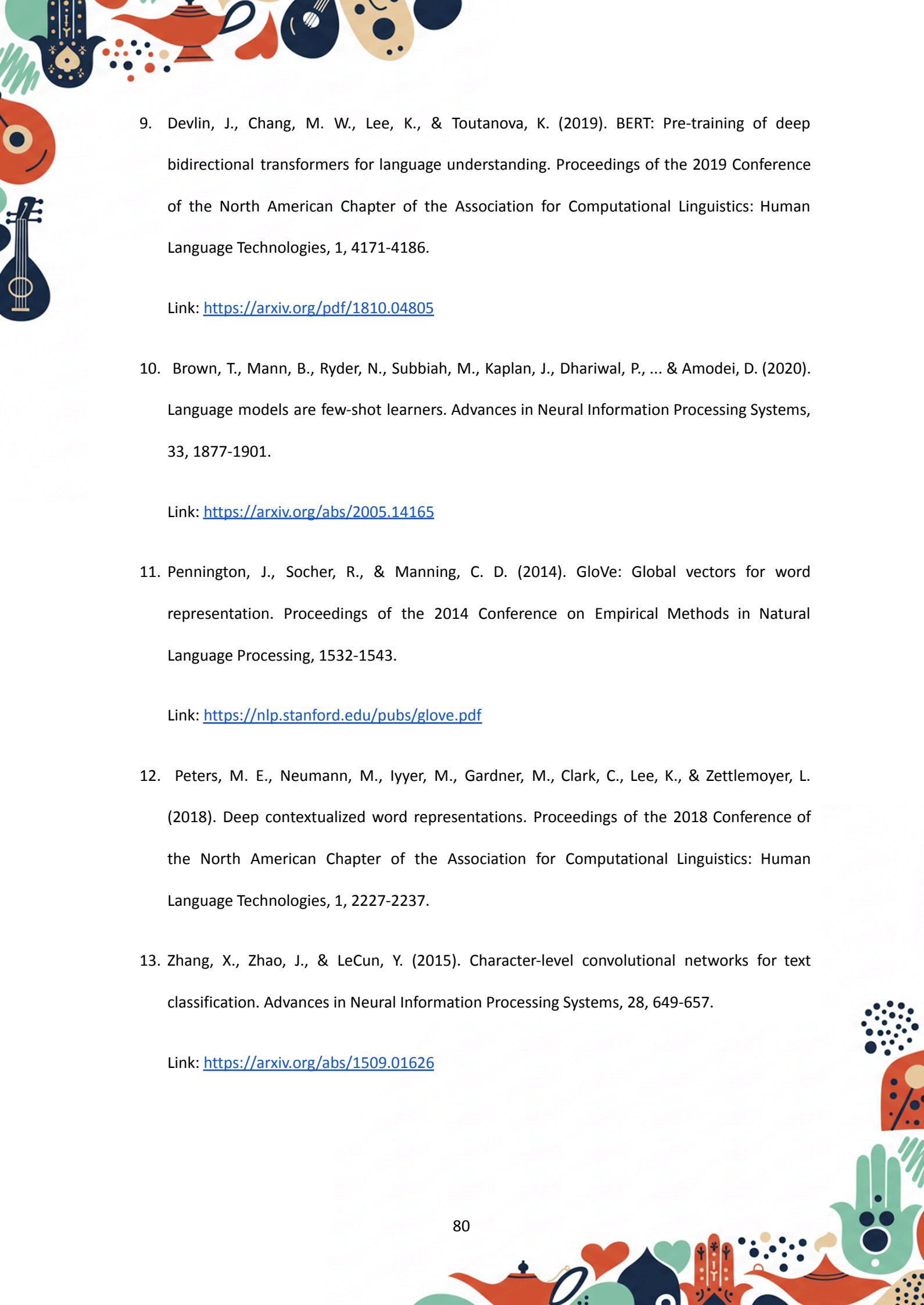
Expanding to support multiple languages and currencies, enabling seamless use for international travelers. Customizing recommendations based on regional and cultural factors.

User Engagement Features:

Adding forums, reviews, and social sharing options to foster a community of engaged travelers. Enabling tourists to save and share itineraries with friends and family.

5.3 References

1. *GetYourGuide*. (n.d.). GetYourGuide: Book Things To Do, Attractions, and Tours. Retrieved January 18, 2025, from <https://www.getyourguide.com/>
2. *Kayak Travel*. (n.d.). Search Flights, Hotels & Rental Cars | KAYAK. Retrieved January 18, 2025, from <https://www.kayak.com/>
3. *Routeperfect*. (n.d.). RoutePerfect: Plan & Book A Perfect Trip With Our Itinerary Planner. Retrieved January 18, 2025, from <https://www.routeperfect.com/>
4. *Viator*. (n.d.). Viator: Travel Tours, Activities, and Things to Do | 2025. Retrieved January 18, 2025, from <https://www.viator.com/>
5. Jurafsky, D., & Martin, J. H. (2021). Speech and Language Processing (3rd ed.). Prentice Hall.
Link: <https://web.stanford.edu/~jurafsky/slp3/>
6. Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit. O'Reilly Media.
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111-3119.
Link: <https://arxiv.org/pdf/1310.4546>
8. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30, 5998-6008.
Link: <https://arxiv.org/abs/1706.03762>



9. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 4171-4186.

Link: <https://arxiv.org/pdf/1810.04805>

10. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

Link: <https://arxiv.org/abs/2005.14165>

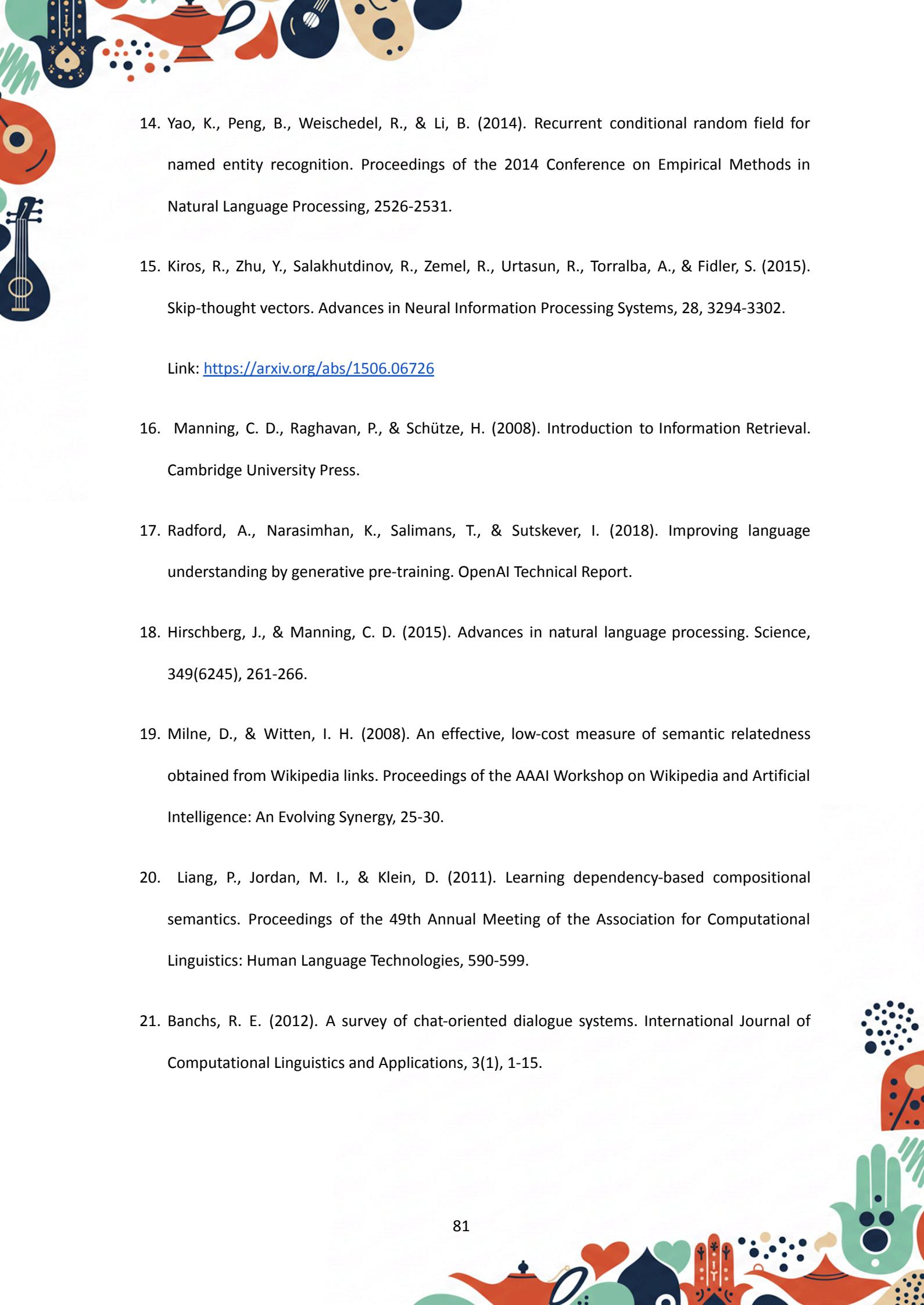
11. Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 1532-1543.

Link: <https://nlp.stanford.edu/pubs/glove.pdf>

12. Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1, 2227-2237.

13. Zhang, X., Zhao, J., & LeCun, Y. (2015). Character-level convolutional networks for text classification. Advances in Neural Information Processing Systems, 28, 649-657.

Link: <https://arxiv.org/abs/1509.01626>

- 
14. Yao, K., Peng, B., Weischedel, R., & Li, B. (2014). Recurrent conditional random field for named entity recognition. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2526-2531.
 15. Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. Advances in Neural Information Processing Systems, 28, 3294-3302.
Link: <https://arxiv.org/abs/1506.06726>
 16. Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.
 17. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. OpenAI Technical Report.
 18. Hirschberg, J., & Manning, C. D. (2015). Advances in natural language processing. Science, 349(6245), 261-266.
 19. Milne, D., & Witten, I. H. (2008). An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. Proceedings of the AAAI Workshop on Wikipedia and Artificial Intelligence: An Evolving Synergy, 25-30.
 20. Liang, P., Jordan, M. I., & Klein, D. (2011). Learning dependency-based compositional semantics. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 590-599.
 21. Banchs, R. E. (2012). A survey of chat-oriented dialogue systems. International Journal of Computational Linguistics and Applications, 3(1), 1-15.

22. Link:

https://www.researchgate.net/publication/252869214_Visual_limiting_magnitude_determination_charts

23. Chen, Q., Zhuo, Z., & Wang, W. (2019). Bert for joint intent classification and slot filling. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 716-721.

24. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 260-270.

25. Honnibal, M., & Montani, I. (2020). spaCy 3: Industrial-strength NLP. Explosion AI.

Link: <https://spacy.io/>

26. Yadav, V., & Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. *Proceedings of the 27th International Conference on Computational Linguistics*, 2145-2158.

27. Cer, D., Yang, Y., Kong, S. Y., Hua, N., Limtiaco, N., John, R. S., ... & Kurzweil, R. (2018). Universal sentence encoder. arXiv preprint arXiv:1803.11175.

Link: <https://arxiv.org/abs/1803.11175>

28. Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982-3992.



Link: <https://arxiv.org/abs/1908.10084>

29. UNESCO. (2021). World Heritage in Egypt. Retrieved from UNESCO Website.

Link: <https://whc.unesco.org/en/statesparties/eg/>

30. Shahrour, I., & Khalifa, G. (2019). Tourism in Egypt: Development and economic impact.

Journal of Tourism and Hospitality, 8(2), 1-10.

Link:

https://www.researchgate.net/publication/331408576_Vorhersagen_fur_das_Bremsverhalten_von_Reifen_durch_Bestimmung_der_Gummireibung

31. Ministry of Tourism and Antiquities, Egypt. (2020). Discover Egypt: Tourist Guide. Cairo, Egypt.

Link: <https://www.experienceegypt.eg/en>

32. Lonely Planet. (2021). Egypt Travel Guide. Lonely Planet Publications.

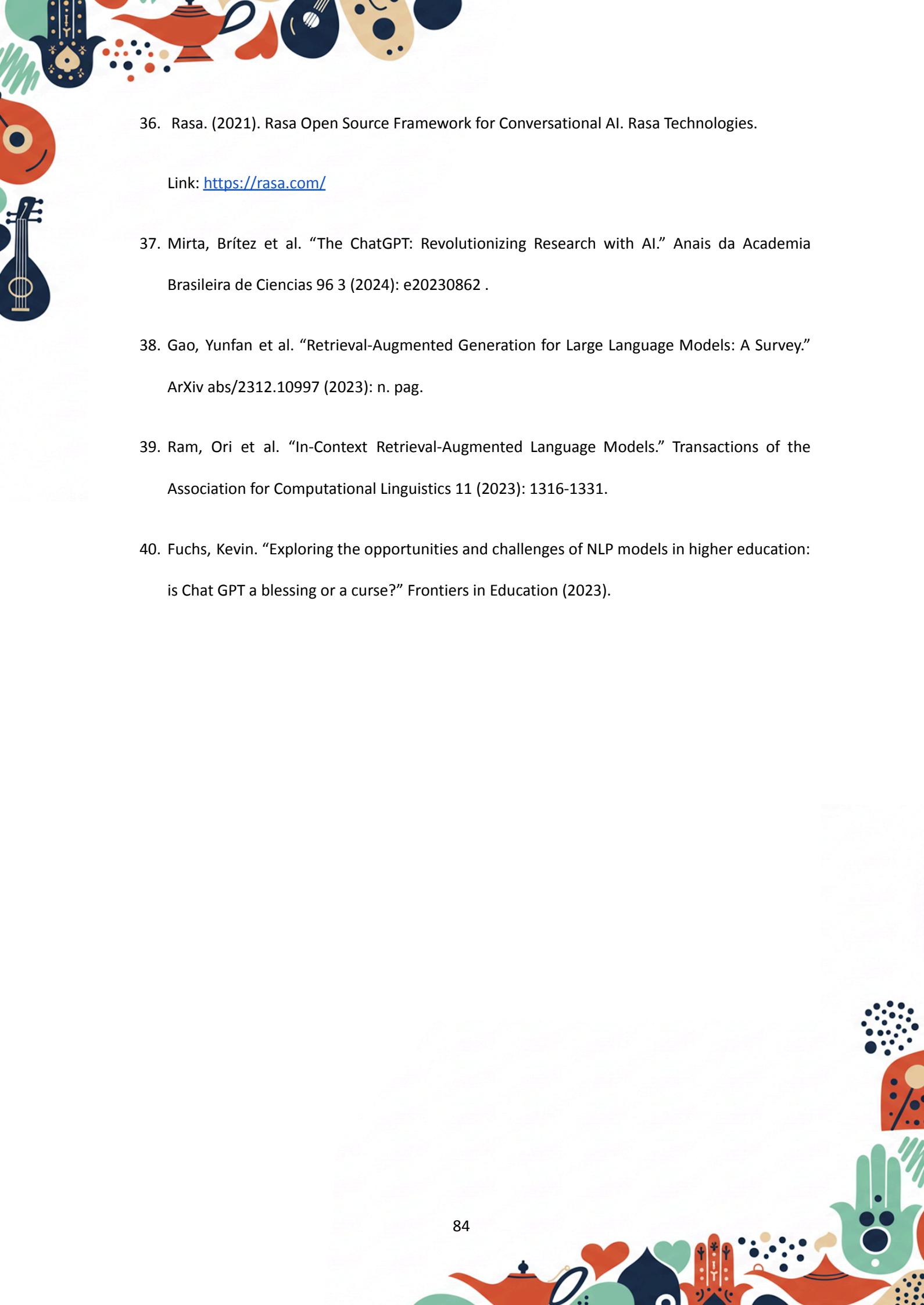
33. Hasan, R. A., & Ismail, A. F. (2018). Identifying tourist preferences for destination planning in Egypt using big data analytics. Journal of Destination Marketing & Management, 10, 1-12.

Link: <https://www.sciencedirect.com/science/article/pii/S2212571X18300289>

34. Elshakankiri, M., Ismail, S., & Salah, A. (2018). Enhancing tourism services in Egypt using AI and IoT. Future Computing and Informatics Journal, 3(2), 22-31.

35. Python Software Foundation. (2021). NLTK: Natural Language Toolkit. Retrieved from [nltk.org](https://www.nltk.org/).

Link: <https://www.nltk.org/>



36. Rasa. (2021). Rasa Open Source Framework for Conversational AI. Rasa Technologies.

Link: <https://rasa.com/>

37. Mirta, Brítez et al. "The ChatGPT: Revolutionizing Research with AI." Anais da Academia Brasileira de Ciencias 96 3 (2024): e20230862 .

38. Gao, Yunfan et al. "Retrieval-Augmented Generation for Large Language Models: A Survey." ArXiv abs/2312.10997 (2023): n. pag.

39. Ram, Ori et al. "In-Context Retrieval-Augmented Language Models." Transactions of the Association for Computational Linguistics 11 (2023): 1316-1331.

40. Fuchs, Kevin. "Exploring the opportunities and challenges of NLP models in higher education: is Chat GPT a blessing or a curse?" Frontiers in Education (2023).