Intelligent Tour Guide

Software Design and Requirement Specification

Session: 2021 – 2025

Submitted by:

Ateeq Aslam	2021-CS-638
M.Jamal Mustafa	2021-CS-631
Javeria Ijaz	2021-CS-604

Supervised by:

Sir Usman Ghani

Co-Supervised by:

Dr. Irfan Yousuf

Department of Computer Science, New Campus

University of Engineering and Technology

Lahore, Pakistan

Contents

List of Figures

List of Tables

Chapter 1

Software Requirement Specification

Pakistan is a beautiful country with many interesting cultures, offering a unique blend of ancient traditions and modern influences. But it can be hard to travel there because it's difficult to find information about the best places to visit. Planning a trip, finding hidden places, and understanding the language can be difficult. **Intelligent Tour Guide** app helps with all of these things. It gives you special tips, connects you with local people, and helps you understand the language. With our app, you can easily plan your trip, discover hidden treasures, and have a great time in Pakistan.

1.1 Functional Requirement

Functional requirements define the basic system behaviour. Essentially, they are what the system does or must not do, and can be thought of in terms of how the system responds to inputs.As Mentioned in the Table 1.1 Functional requirements usually define if/then behaviours and include calculations, data input, and business processes. Functional requirements are product features and focus on user requirements.

Table 1.1: Functional Requirements

ID	Name	Description	Priority
FR-01	User Registration	The system shall allow users to register using their credentials.	High

ID	Name	Description	Priority
FR-02	User Authentication	The system shall authenticate users during login using their credentials.	High
FR-03	User Profile Management	The system shall allow users (tourists and guides) to create and edit their profiles and personal information.	High
FR-04	Admin Profile Management	The system shall allow the admin to edit or delete a user's profile.	Medium
FR-05	Guide Authentication	The system shall allow the admin to review and accept the application request of users for tourist guide.	Medium
FR-06	Tour Generation	The system shall allow users to generate tours by entering desired destinations and customizations.	Medium
FR-07	Cost Estimation	The system shall provide travel cost estimation based on selected destinations.	High
FR-08	Route Generation	The system shall generate a travel route based on user input and tour preferences.	High
FR-09	Guide Interaction	The system shall allow tourists to contact local guides and utilize their services.	Medium
FR-10	Guide Feedback	The system shall allow users to provide feedback on the guide's services.	Medium
FR-11	Chatbot Interaction	The system shall offer a chatbot for answering travel-related queries.	High
FR-12	Query Authentication	The system shall authenticate user queries and provide real-time responses for valid queries.	Medium
FR-13	Language Translation	The system shall allow user to translate sentences from multiple languages to Urdu and vice versa during communication.	High
FR-14	Logout Functionality	The system shall allow users to log out of their accounts.	Medium

1.2 Non-functional Requirement

Non-functional requirements specify how the system should operate. As mentioned in Table 1.2 Non-functional requirements do not affect the basic functionality of the system report.

Table 1.2: Non-Functional Requirements

ID	**Name**	**Description**	**Priority**
NR-01	Usability	The system should have an intuitive and user-friendly interface accessible to users of all technical levels.	High
NR-02	Availability	The system should be available 99.9% of the time to ensure continuous access to travel information and services.	High
NR-03	Performance	The system should load user queries and travel suggestions within 3 seconds.	High
NR-04	Concurrent User Handling	The system should be able to handle up to many concurrent users without performance degradation.	High
NR-05	Security	The system should encrypt all sensitive user data to protect against unauthorized access.	High
NR-06	Access Control	The system should implement role-based access control for different user types (e.g., admin, user).	Medium
NR-07	Scalability	The system should be scalable to accommodate additional features and increased user load in the future.	Medium
NR-08	Supportability	The system should be maintainable with regular updates and patches to fix bugs and add new features.	Medium

Chapter 2

Design specification

2.1 System Behavioral Design

Behavioral diagrams portray a dynamic view of a system, illustrating how it behaves and functions over time. They describe the interactions and processes within the system, providing insight into its operational aspects.

2.1.1 Use case Diagram

As shown in the Figure 2.1 Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact.Use case diagrams are typically developed in the early stage of development and people often apply use case modeling for the following purposes:

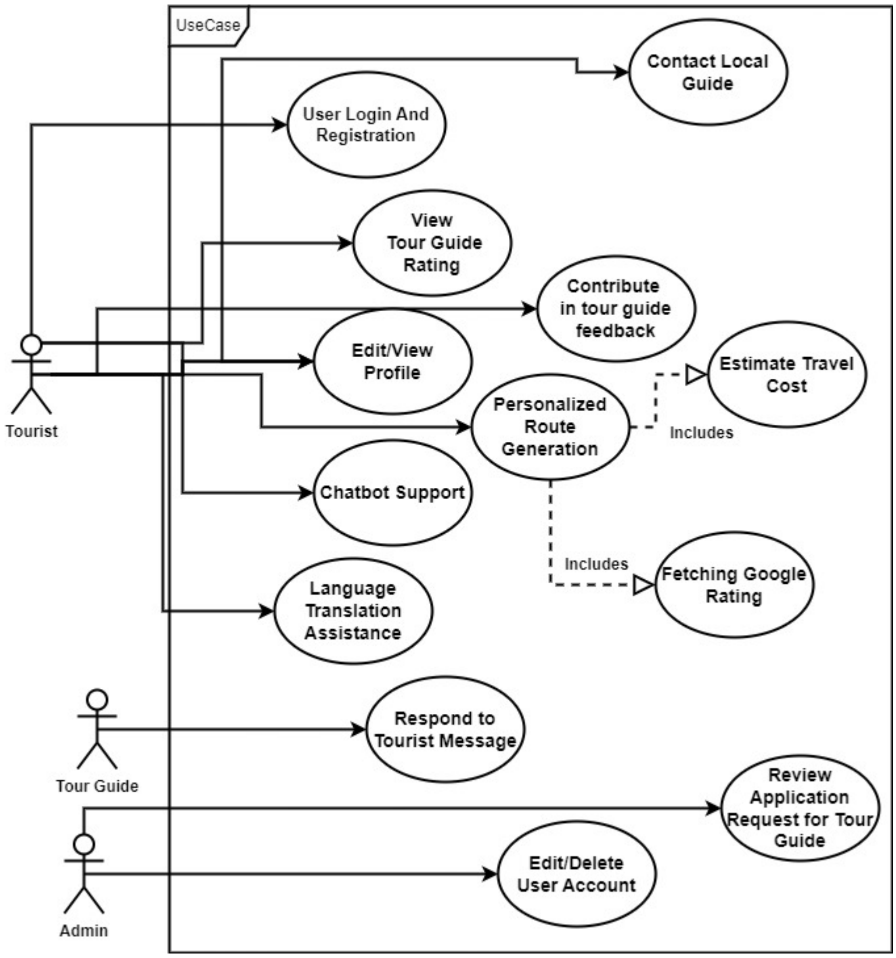

UseCase
Contact Local Guide
User Login And Registration
View Tour Guide Rating
Contribute in tour guide feedback
Edit/View Profile
Estimate Travel Cost
Personalized Route Generation
Includes
Tourist
Chatbot Support
Includes
Fetching Google Rating
Language Translation Assistance
Respond to Tourist Message
Tour Guide
Review Application Request for Tour Guide
Edit/Delete User Account
Admin

FIGURE 2.1: Use Case

The different Use casses of our system are presented from Table 2.1 - Table 2.10

TABLE 2.1: Use Case: User Registration and Login

Use Case	**User Registration and Login**
Use Case ID	UC-01
Primary Actor	User
Goal	To create a new account and log in to the system.

Use Case	**User Registration and Login**
Preconditions	• The user has access to the application. • The user has a valid email address. • An internet connection is available.
Postconditions	• **Success:** The user is logged in (registered or authenticated) and redirected to the dashboard. • **Failure:** The user is shown an appropriate error message.
Main Flow	1. The user opens the application. 2. The user selects the registration option. 3. The user enters required details (email, password, etc.). 4. The system verifies the information and creates an account. 5. The user logs in using their credentials. 6. The system authenticates the user and redirects them to the dashboard.
Alternate Flow	• If the user enters incorrect credentials, the system prompts them to re-enter or reset their password.

Use Case	**User Registration and Login**
Exceptions	- **E1:** Invalid Email Format – The system detects that the email provided is not in a valid format and prompts the user to correct it. - **E2:** Email Already in Use – The system identifies that the email is already registered and asks the user to log in or reset their password. - **E3:** Weak Password – The system detects that the password does not meet security requirements (e.g., too short, no special characters) and prompts the user to choose a stronger password. - **E4:** System Error – If a system error occurs during registration or login, the user is shown a generic error message and asked to try again later.

Table 2.2: Use Case: User Profile Management

Use Case	**User Profile Management**
Use Case ID	UC-02
Primary Actor	User
Goal	To allow users to edit their profile.
Preconditions	- The user is logged in.
Post-conditions	- **Success:** The user's profile is updated. - **Failure:** The user is shown an appropriate error message.

Use Case	**User Profile Management**
Main Flow	1. The user navigates to the profile section. 2. The user clicks on ”Edit Profile.” 3. The user changes their credentials. 4. The user saves the changes. 5. The system updates the user’s profile and preferences.
Alternate Flow	Not Available
Exceptions	• **E1:** Invalid Input – If the user enters invalid data (e.g., incorrect budget format), the system prompts for correction. • **E2:** Save Failure – If there is an issue saving the profile, the system notifies the user and suggests retrying. • **E3:** System Error – The system detects that the internet connection is not secure and notifies the user.

Table 2.3: Use Case: Create a Personalized Travel Itinerary

Use Case	**Create a Personalized Travel Itinerary**
Use Case ID	UC-03
Primary Actor	User
Goal	To create a customized travel itinerary based on user preferences.

Use Case	**Create a Personalized Travel Itinerary**
Preconditions	• The user is logged in. • The user has clicked on the ”Generate Tour” option.
Post-conditions	• **Success:** A personalized travel itinerary is generated for the user. • **Failure:** The user is shown an appropriate error message.
Main Flow	1. The user selects the ”Generate tour” option. 2. The user inputs preferences such as source, destination, shopping, etc. 3. The system suggests tours along with estimated costs based on input. 4. The user selects preferred suggestions. 5. The user saves the itinerary for future reference.
Alternate Flow	No

Use Case	**Create a Personalized Travel Itinerary**
Exceptions	• **E1:** Incomplete Preferences – If the user fails to input all required preferences (e.g., source, destination), the system prompts for the missing information. • **E2:** No Matching Suggestions – If the system cannot find attractions or activities that match the user's preferences, a message is displayed asking the user to broaden their criteria. • **E3:** System Error – If a system error occurs while generating the itinerary, the system informs the user of the issue and suggests trying again later. • **E4:** Network Failure – If the user's connection drops during itinerary creation, the system saves progress and allows the user to resume once the connection is restored.

Table 2.4: Use Case: View and Select Local Guides

Use Case	**View and Select Local Guides**
Use Case ID	UC-04
Primary Actor	User
Goal	To allow users to view and select local guides based on profiles, experience, and reviews.
Preconditions	• The user is logged in.

Use Case	**View and Select Local Guides**
Post-conditions	• **Success:** The user selects a local guide and contacts them for service. • **Failure:** The user is shown an error message or alternative options.
Main Flow	1. The user navigates to the local guide section. 2. The system displays a list of local guides with profiles, experience, and reviews. 3. The user browses the profiles and selects a guide. 4. The user contacts the guide for a service.
Alternate Flow	• If no guides are available, the system suggests other means of local assistance.
Exceptions	• **E1:** Guide Unavailable – If the selected guide is unavailable, the user is informed and prompted to choose another guide. • **E2:** Profile Loading Error – If a guide's profile fails to load, the system notifies the user and offers alternative guides.

Table 2.5: Use Case: Local Guide Feedback

Use Case	**Local Guide Feedback**
Use Case ID	UC-05

Use Case	**Local Guide Feedback**
Primary Actor	User
Goal	To give feedback on the recent experience with a local guide.
Preconditions	• The user is logged in. • The user has received a service from a local guide available in the app.
Post-conditions	• **Success:** The user's review or rating is published and visible to other users. • **Failure:** The user is shown an appropriate error message or asked to retry the submission.
Main Flow	1. The user selects the option to give feedback on the local guide. 2. The system prompts the user to enter their review and rating. 3. The user submits the review. 4. The system displays the review alongside other user-generated content.
Alternate Flow	Not Available

Use Case	**Local Guide Feedback**
Exceptions	• **E1:** Invalid Review Content – If the review contains prohibited content (e.g., offensive language), the system rejects the submission and prompts the user to revise the review. • **E2:** Duplicate Review – If the user attempts to submit a review for a service they have already reviewed, the system notifies them and prevents duplicate submissions. • **E3:** Review Submission Failure – If there is an issue submitting the review (e.g., network failure), the system saves the review locally and attempts to submit it once the connection is restored. • **E4:** System Error – If a system error occurs during review submission, the user is informed that the review could not be posted and is encouraged to try again later.

Table 2.6: Use Case: Chat-bot Interaction for Real-Time Support

Use Case	**Chat-bot Interaction for Real-Time Support**
Use Case ID	UC-06
Primary Actor	User
Goal	To get real-time answers and travel advice from the chat-bot.
Preconditions	• The user is logged in. • The chat-bot feature is active.

Use Case	**Chat-bot Interaction for Real-Time Support**
Post-conditions	• **Success:** The user receives relevant information or advice from the chat-bot. • **Failure:** The user is informed of an issue with the chat-bot.
Main Flow	1. The user accesses the chat-bot feature. 2. The user asks a question or requests travel advice. 3. The chat-bot processes the query and provides an appropriate response. 4. The user receives the response and may ask follow-up questions or end the interaction.
Alternate Flow	• If the chat-bot does not recognize a query, it asks the user to rephrase the question.
Exceptions	• **E1:** Unrecognized Query – If the chat-bot does not understand the query, it asks the user to rephrase the question. • **E2:** Network Failure – If the chat-bot cannot connect to the server, it informs the user that it's currently unavailable and suggests trying again later.

Table 2.7: Use Case: Translation Assistance

Use Case	**Translation Assistance**
Use Case ID	UC-07
Primary Actor	User
Goal	To assist users with real-time language translation during their travel.
Preconditions	• The user is logged in. • The language translation feature is active.
Post-conditions	• **Success:** The user receives translated text or speech to aid in communication. • **Failure:** The user is notified if translation cannot be completed.
Main Flow	1. The user selects the "Language Translation" option from the menu. 2. The user inputs or speaks the text they need translated. 3. The system processes the input and provides a translation in the URDU language. 4. The user views or listens to the translated text.
Alternate Flow	• Not Available.

Use Case	**Translation Assistance**
Exceptions	• **E1:** Unrecognized Language – If the system does not recognize the input language, it prompts the user to select the language manually. • **E2:** Translation Error – If the system cannot accurately translate the input, it suggests the user rephrase or use simpler text. • **E3:** Network Failure – If the translation service is unavailable due to network issues, the system notifies the user and offers to save the input for later translation.

Table 2.8: Use Case: Create Tourist Guide Profile

Use Case	**Create Tourist Guide Profile**
Use Case ID	UC-08
Primary Actor	User
Goal	To create a profile for a tourist guide.
Preconditions	• The user is logged in.
Post-conditions	• **Success:** The user successfully creates a tourist guide profile and can provide services. • **Failure:** The system notifies the user if the profile could not be created.

Use Case	**Create Tourist Guide Profile**
Main Flow	1. The user selects the option to register as a tourist guide. 2. The system prompts the user to enter credentials and identity information. 3. The user submits the data. 4. The system displays the user as a tourist guide in the tourist guides list.
Alternate Flow	• Not Available.
Exceptions	• **E1:** Invalid Credentials – If the credentials contain invalid content, the system rejects the submission and prompts the user to re-enter the data. • **E2:** System Error – If a system error occurs during profile creation, the user is informed and encouraged to try again later.

Table 2.9: Use Case: Edit User Account

Use Case	**Edit User Account**
Use Case ID	UC-09
Primary Actor	Admin
Goal	To allow Admin to edit the user's profile.
Preconditions	• The admin is logged in.

Use Case	**Edit User Account**
Post-conditions	• **Success:** The user's account is successfully edited. • **Failure:** The system notifies the admin if the profile could not be updated.
Main Flow	1. The admin navigates to the user's account section. 2. The admin clicks on "Edit Profile." 3. The admin modifies the user's credentials. 4. The admin saves the changes. 5. The system updates the user's profile.
Alternate Flow	• Not Available.
Exceptions	• **E1:** Invalid Input – If the admin enters invalid data, the system prompts for correction. • **E2:** Save Failure – If there's an issue saving the profile, the system notifies the admin and suggests retrying. • **E3:** System Error – If the internet connection is insecure, the system warns the admin.

Table 2.10: Use Case: Delete User Account

Use Case	**Delete User Account**
Use Case ID	UC-10

Use Case	**Delete User Account**
Primary Actor	Admin
Goal	To allow the Admin to delete the user's profile.
Preconditions	• The admin is logged in.
Post-conditions	• **Success:** The user's account is successfully deleted. • **Failure:** The system notifies the admin if the account could not be deleted.
Main Flow	1. The admin navigates to the user's account section. 2. The admin clicks on "Delete Account." 3. The system deletes the user's profile.
Alternate Flow	• Not Available.
Exceptions	• **E1:** Save Failure – If there's an issue saving the profile, the system notifies the admin and suggests retrying. • **E2:** System Error – If the internet connection is insecure, the system warns the admin.

2.1.2 Activity Diagram

Activity diagrams represent workflows in a graphical way as shown in Figure 2.2. They can be used to describe the business workflow or the operational workflow of any component in a system.Draw an activity diagram of your project.

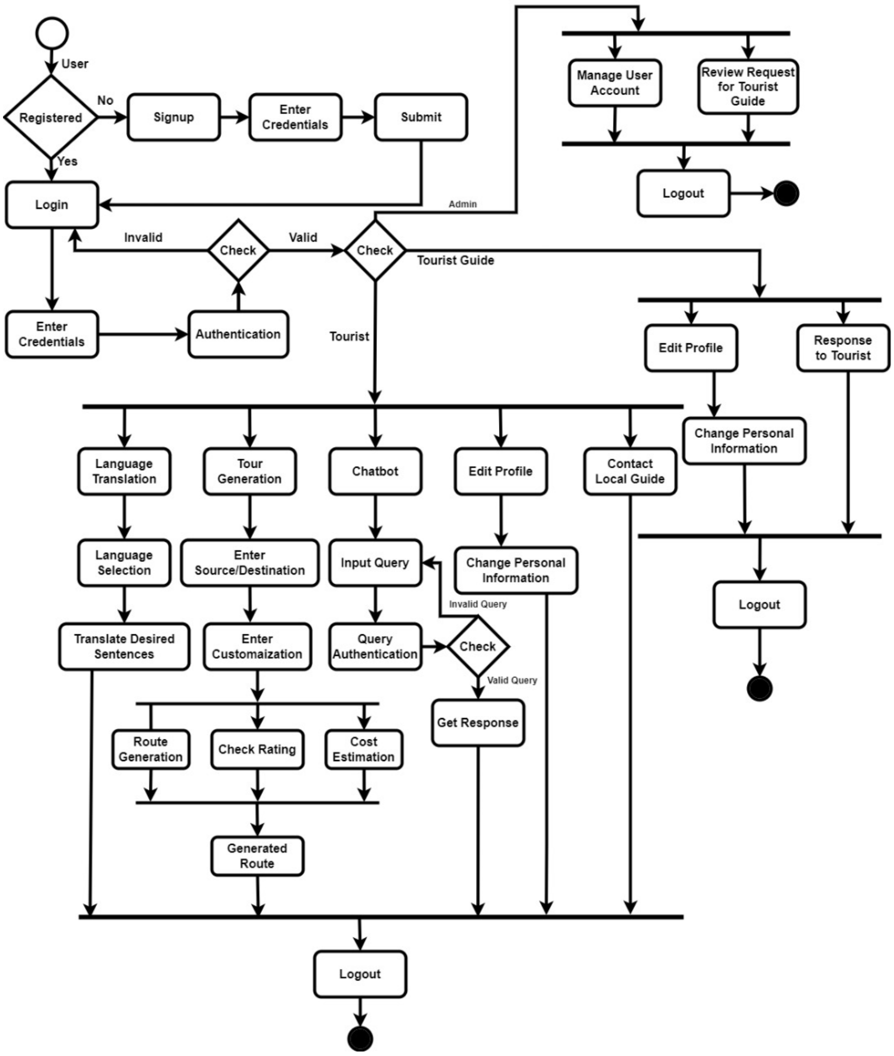


Figure 2.2: Activity Diagram

2.1.3 State Diagram

State machine diagrams are similar to activity diagrams, although notations and usage change a bit. They are sometimes known state chart diagrams as well. As shown in Figure 2.3, Figure 2.4 and Figure 2.5 These are very useful to describe the behavior of objects that act differently according to the state they are in at the moment. The State diagram shows the basic states and actions.Draw a state diagram of your project.

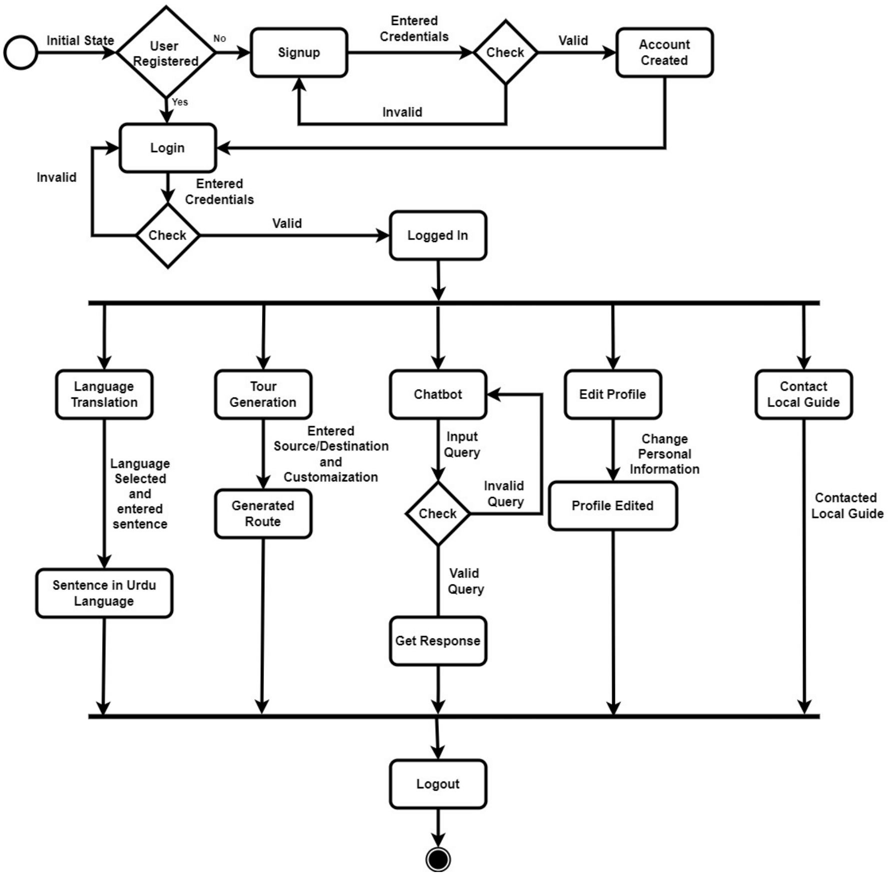


FIGURE 2.3: Tourist State Diagram

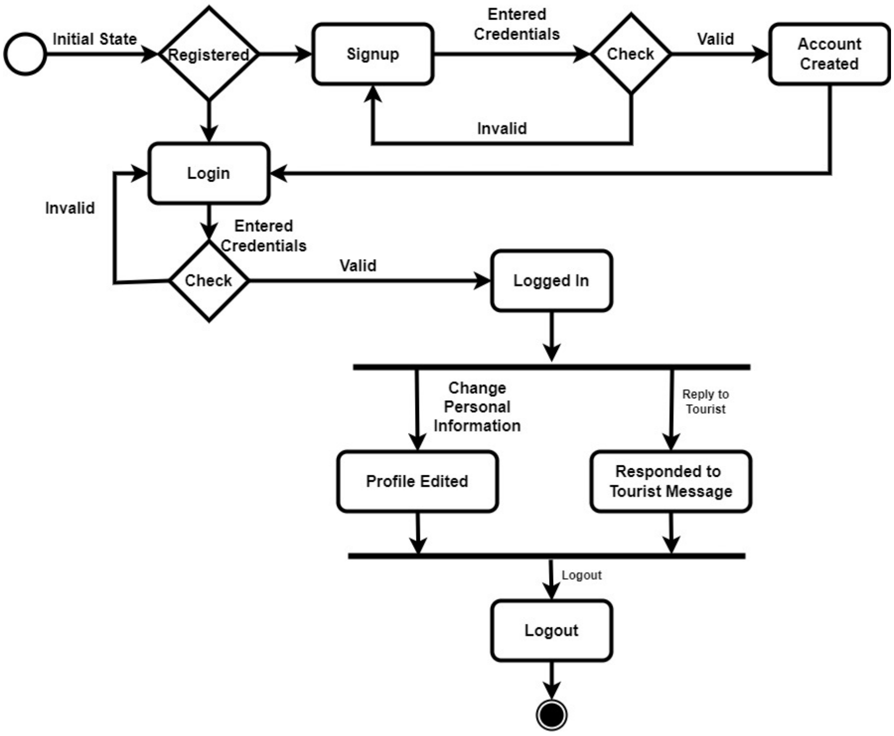


FIGURE 2.4: Tour Guide State Diagram

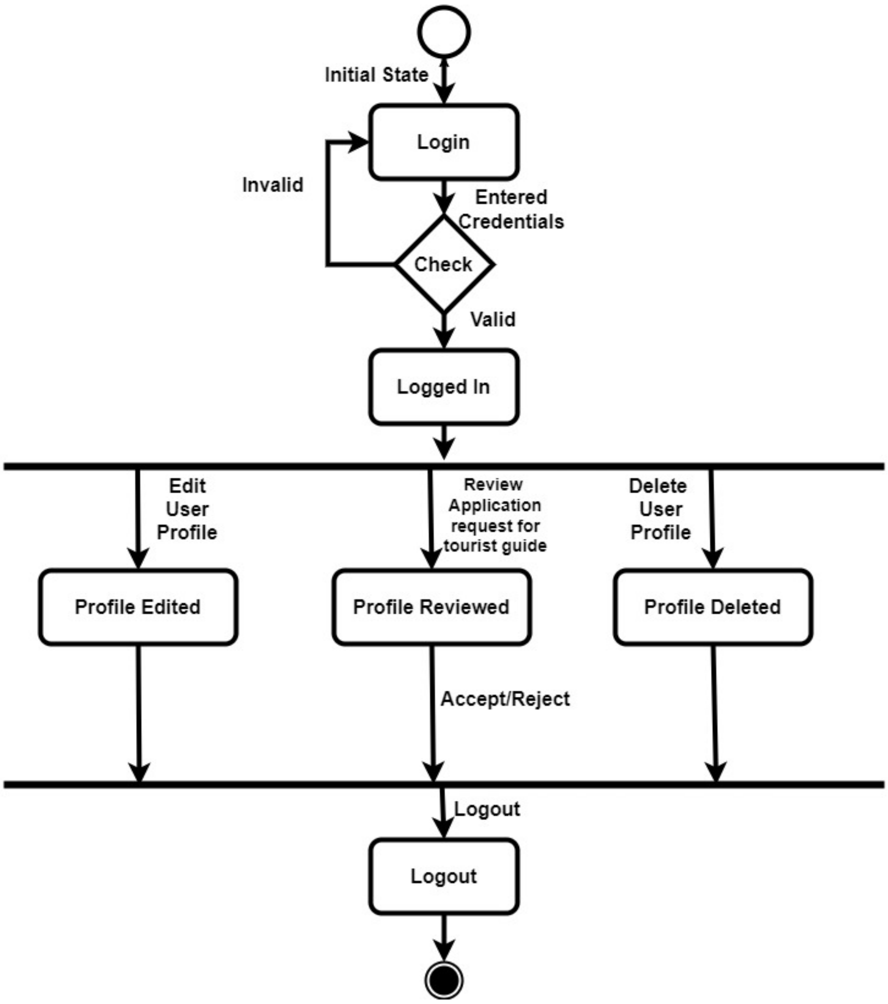


Figure 2.5: Admin State Diagram

2.1.4 Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur as shown in Figure 2.6 - Figure 2.12. It's important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are shown as arrows.Draw a sequence diagram of your project [**?**].

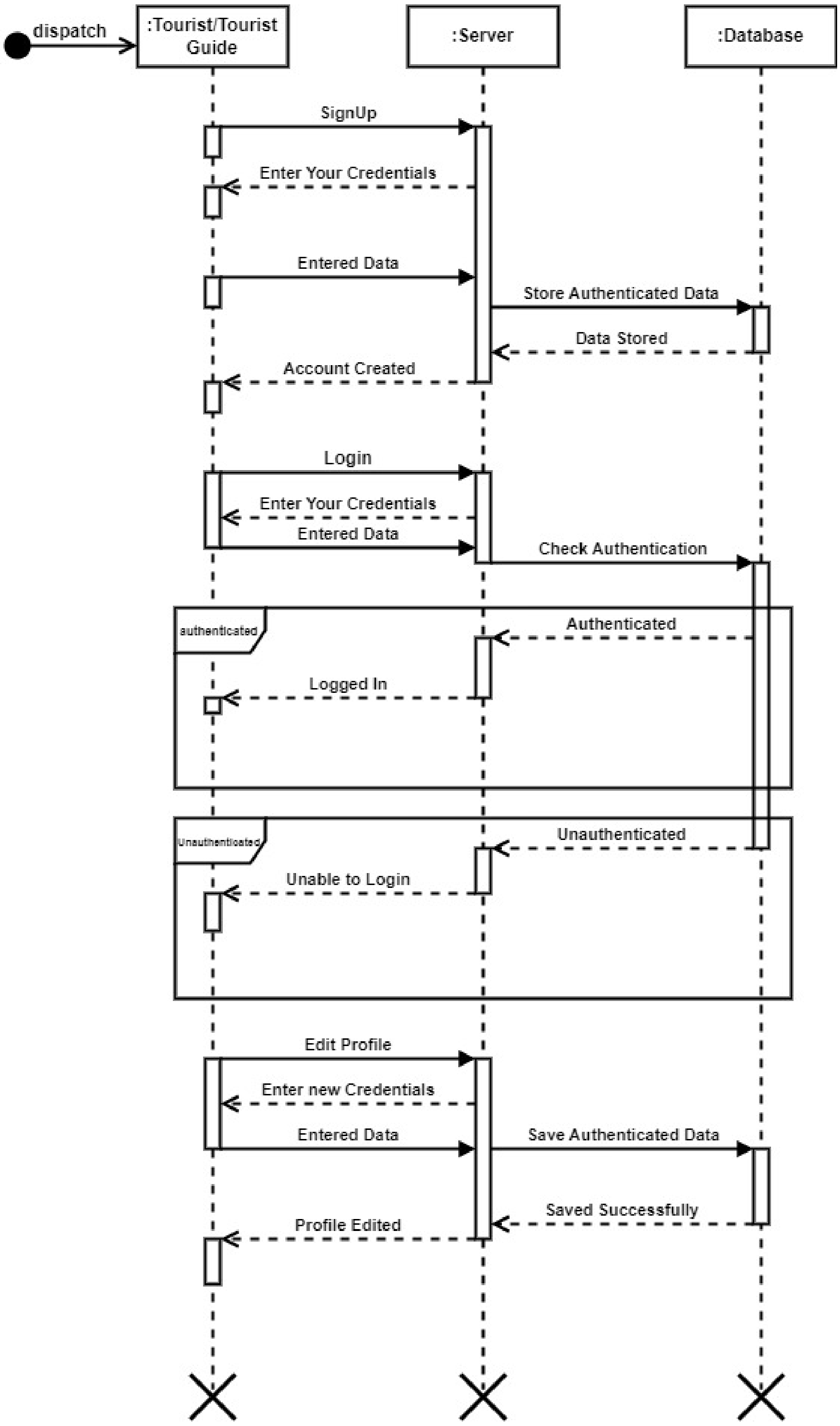


Figure 2.6: Sign up Login and Manage Profile

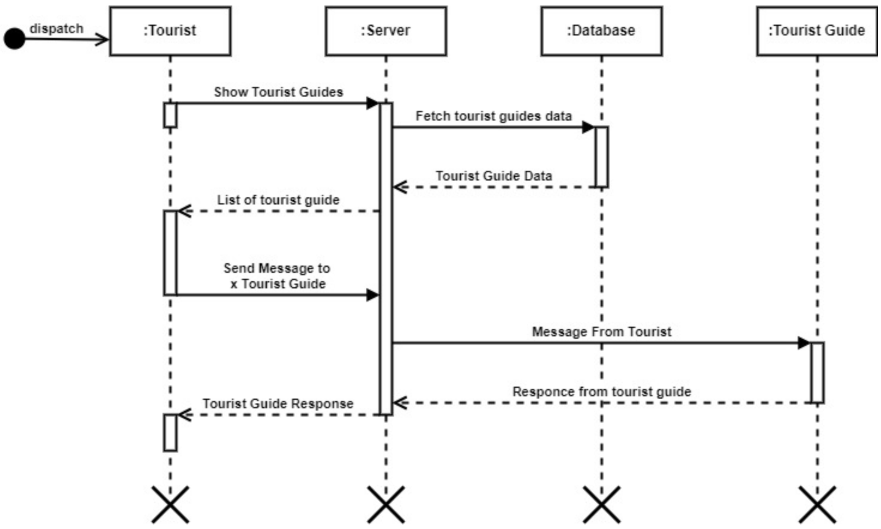


Figure 2.7: Chatting with Tour Guide

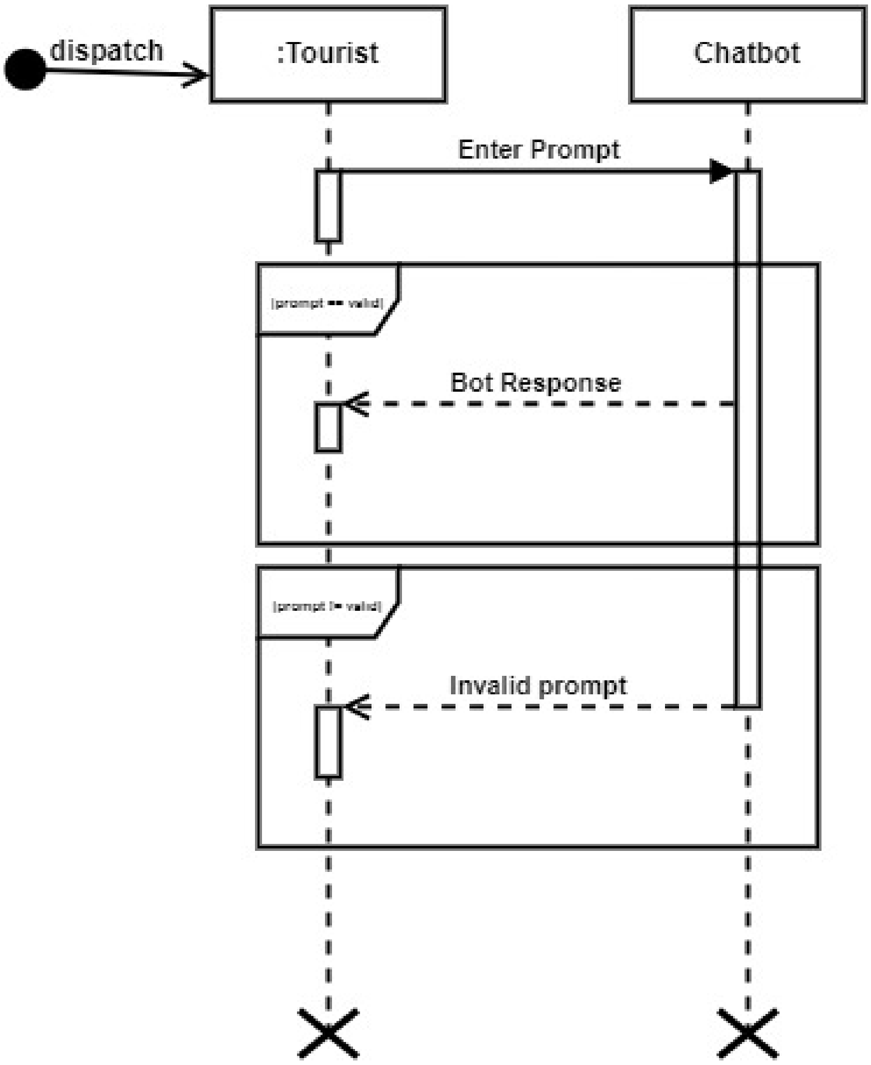


Figure 2.8: Interaction with Chatbot

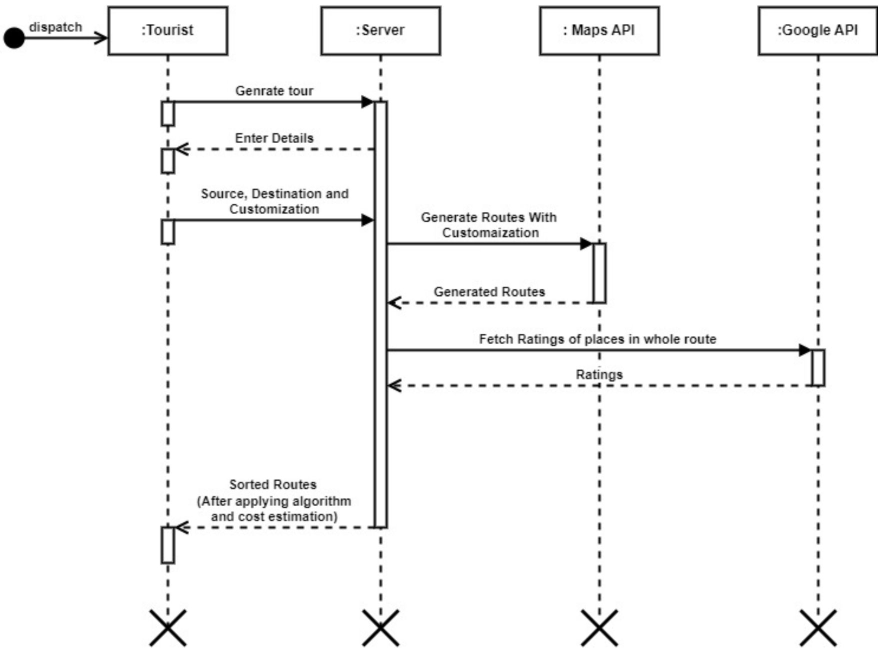

dispatch
:Tourist
:Server
: Maps API
:Google API
Genrate tour
Enter Details
Source, Destination and Customization
Generate Routes With Customaization
Generated Routes
Fetch Ratings of places in whole route
Ratings
Sorted Routes (After applying algorithm and cost estimation)

Figure 2.9: Tour Generation

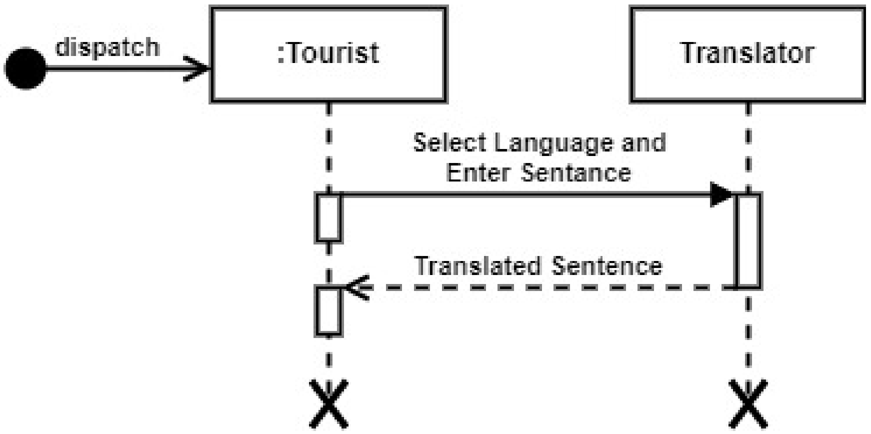


Figure 2.10: Language Translation

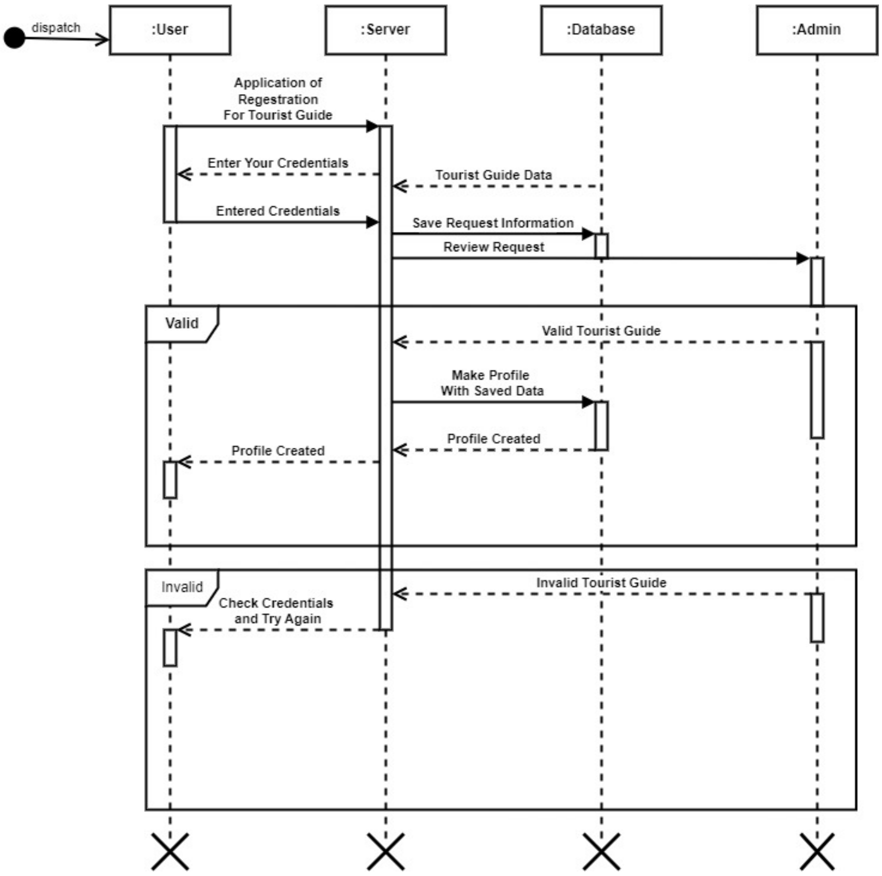


Figure 2.11: Tour Guide Validation by Admin

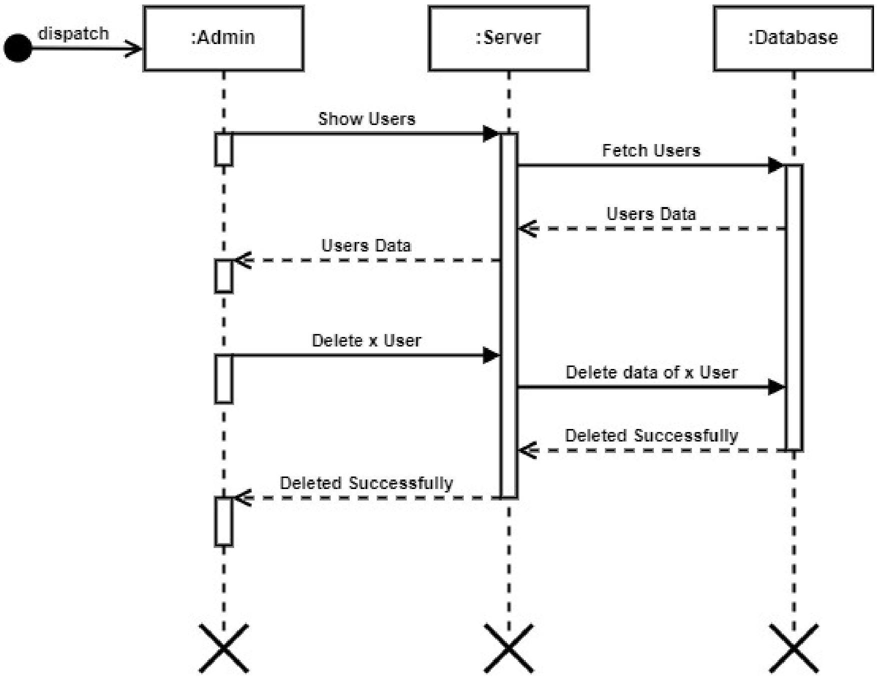


Figure 2.12: Delete Profile by Admin

2.1.5 Collaboration Diagram

Collaboration diagrams (known as Communication Diagram in UML 2.x) are used to show how objects interact to perform the behavior of a particular use case, or a part of a use case as represented in Figure 2.13 - Figure 2.21. Along with sequence diagrams, collaboration are used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.Draw a collaboration diagram of your project.

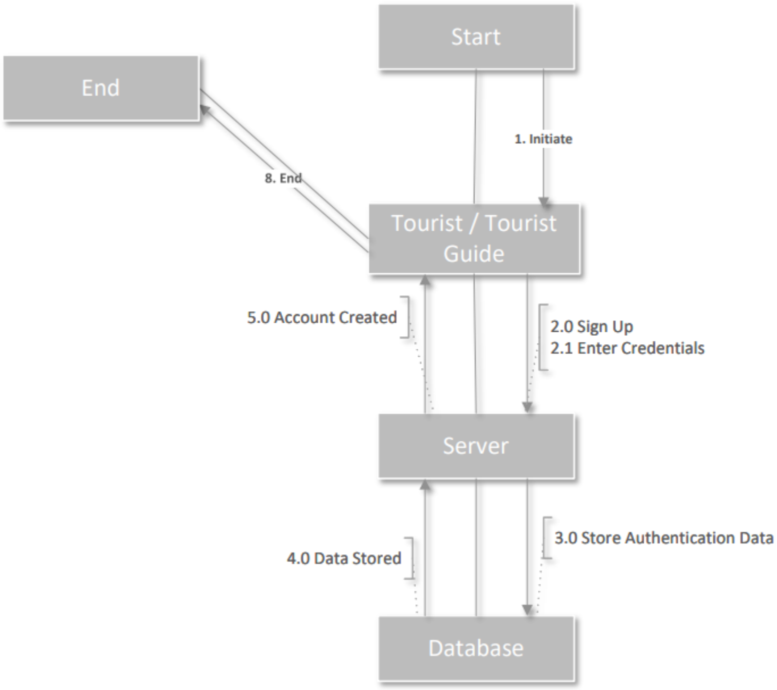


Figure 2.13: Signup

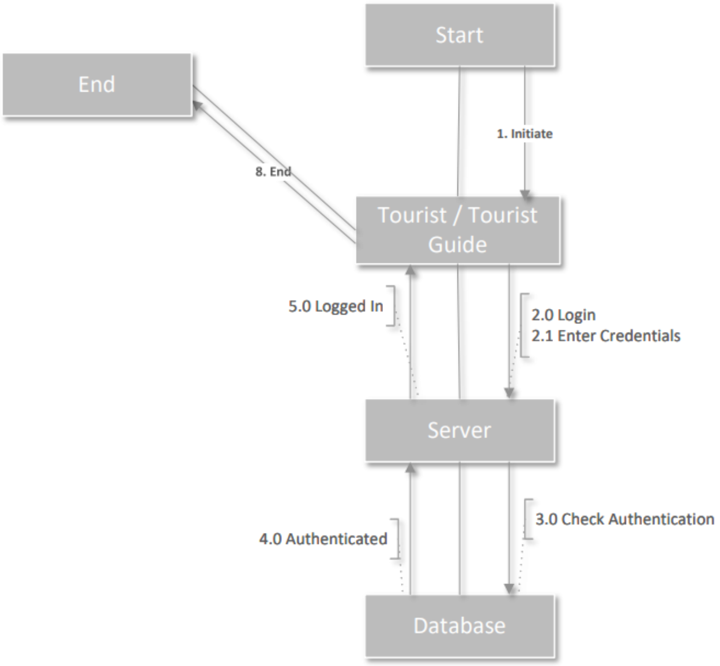


Figure 2.14: Login

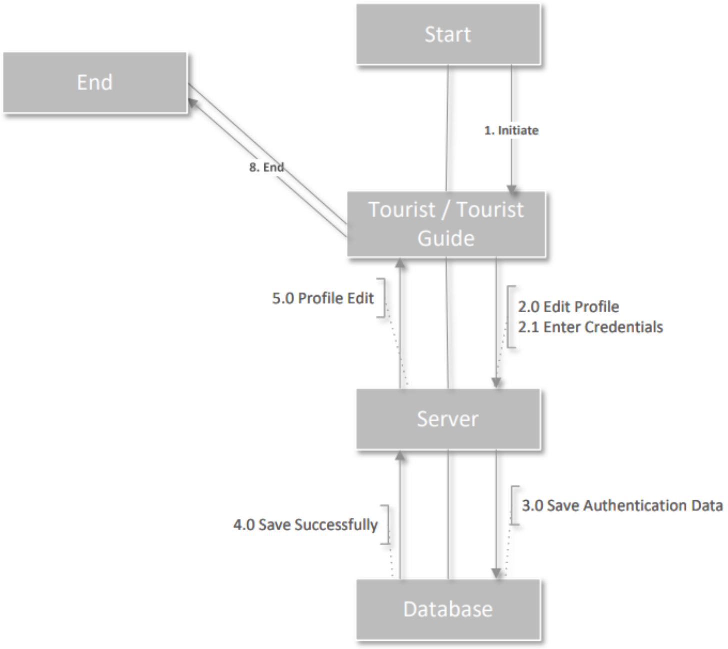


Figure 2.15: Edit Profile

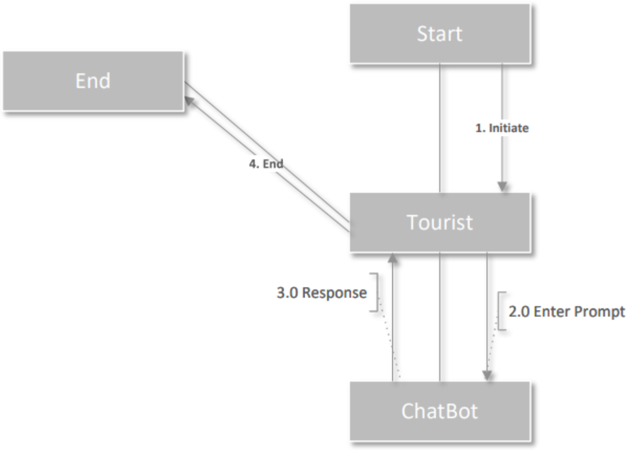


FIGURE 2.16: Chatbot

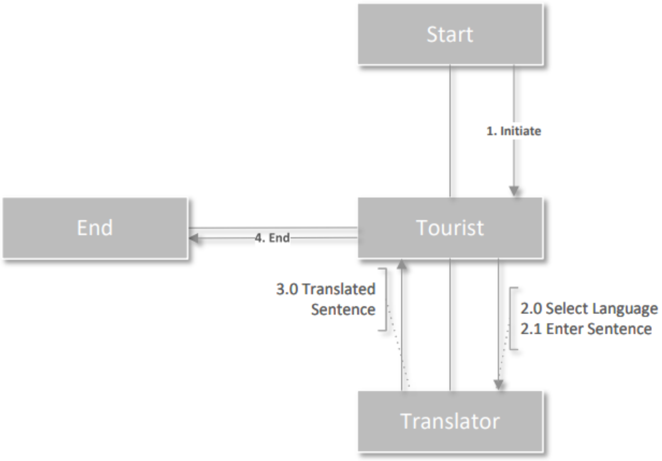


FIGURE 2.17: Language Translation

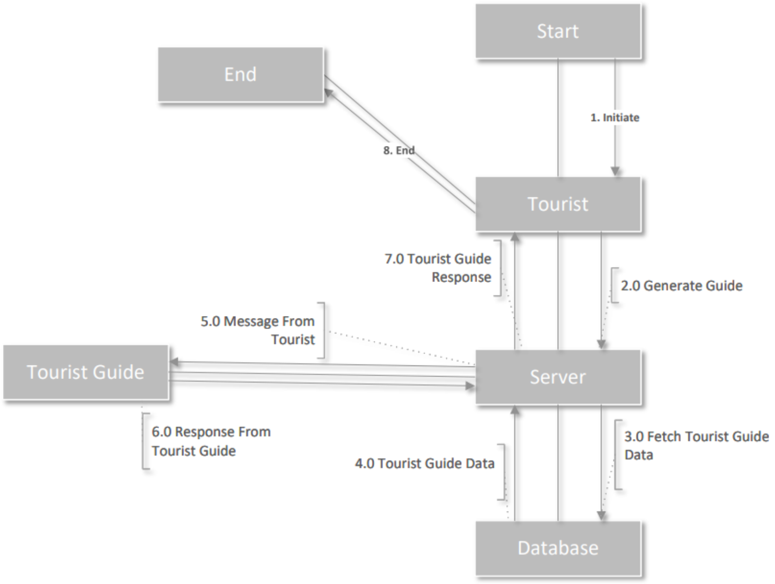


Figure 2.18: Chatting with Tour Guide

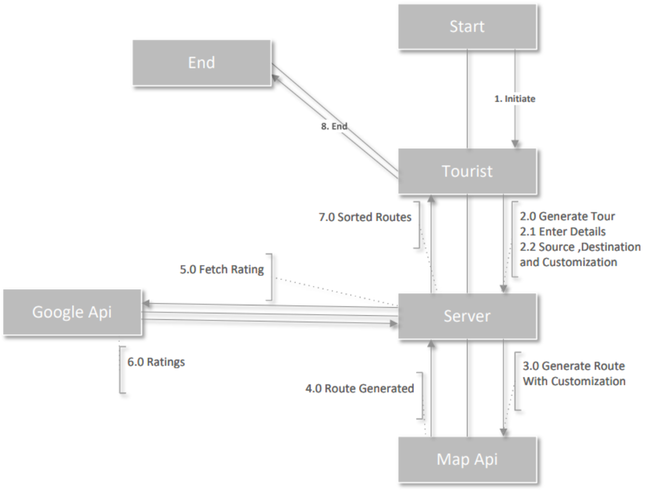


Figure 2.19: Tour Generation

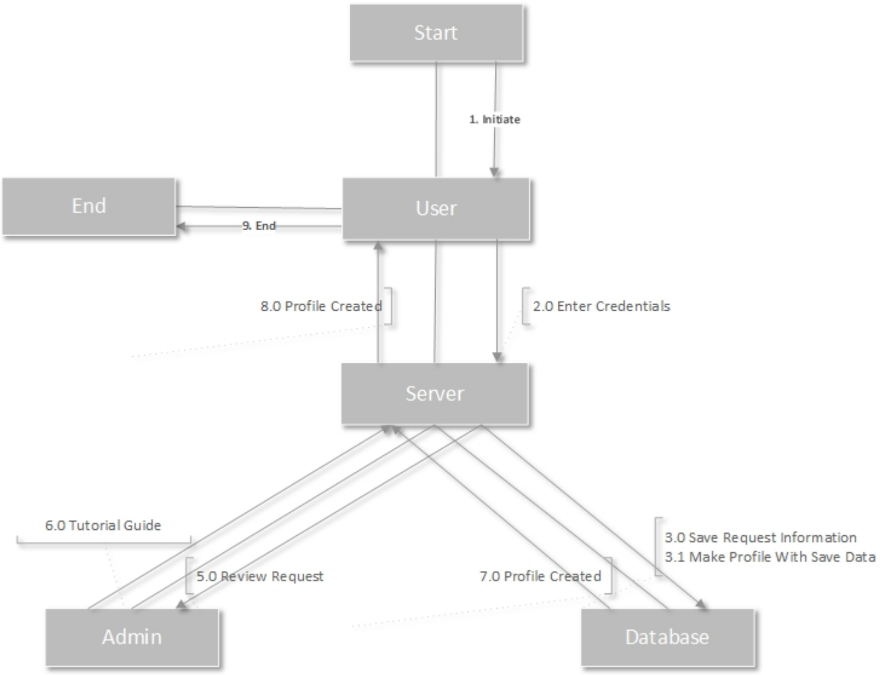


Figure 2.20: Validation of Tour Guide Profile by Admin

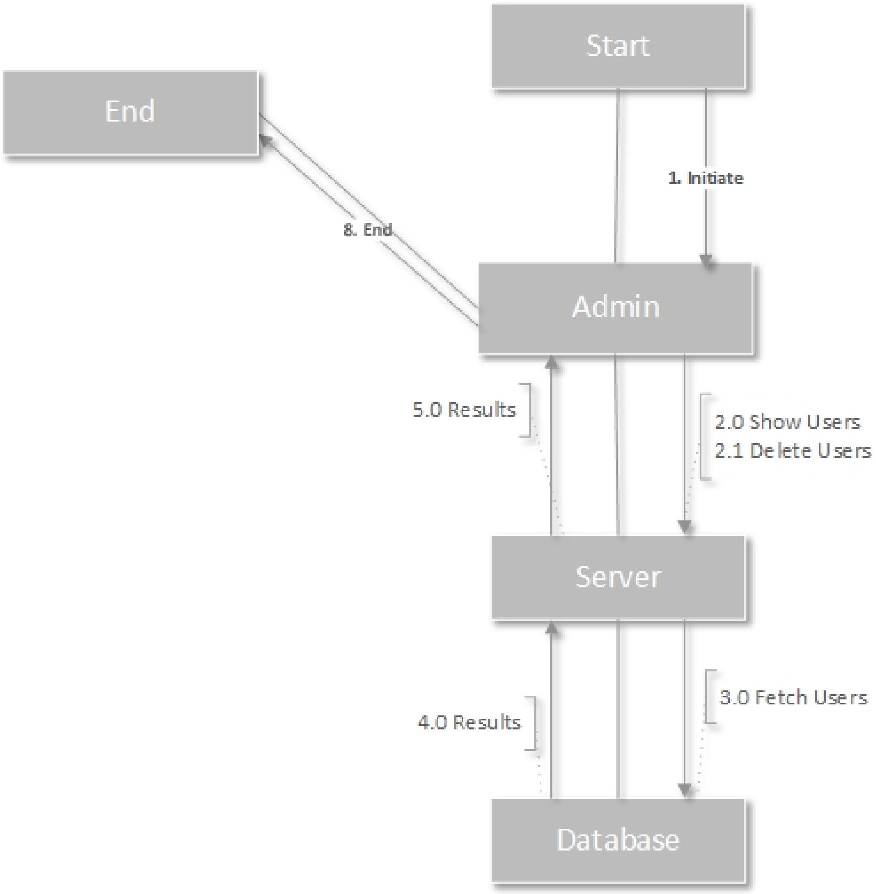


Figure 2.21: Delete Profile by Admin

2.2 System Structure Design

Structural diagrams depict the static aspects or structure of a system, providing a detailed outline of the system's architecture and its components. These diagrams are essential for documenting and understanding the software architecture, as they define the components and their relationships without focusing on the dynamic behavior.

2.2.1 Class Diagram

As you can see in Figure 2.22 Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom. In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.Draw a class diagram of your project.

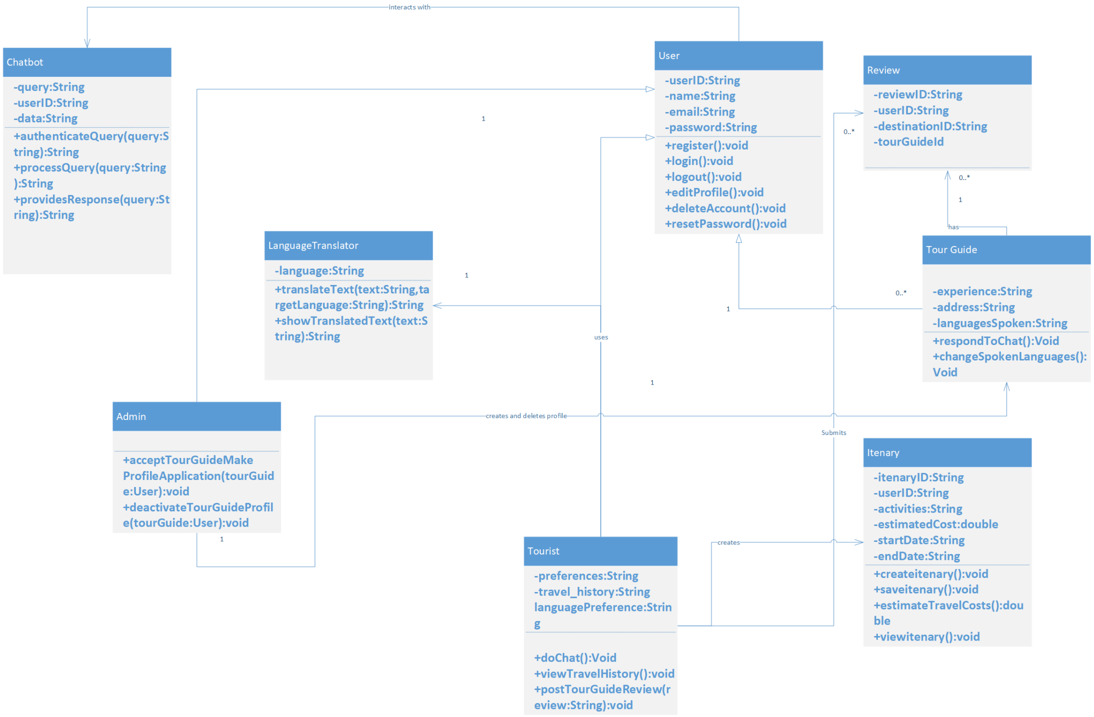


Figure 2.22: Class Diagram

2.2.2 Component Diagram

A component diagram displays the structural relationship of components of a software system. As shown in Figure 2.23 These are mostly used when working with complex systems with many components. Components communicate with each other using interfaces. The interfaces are linked using connectors.

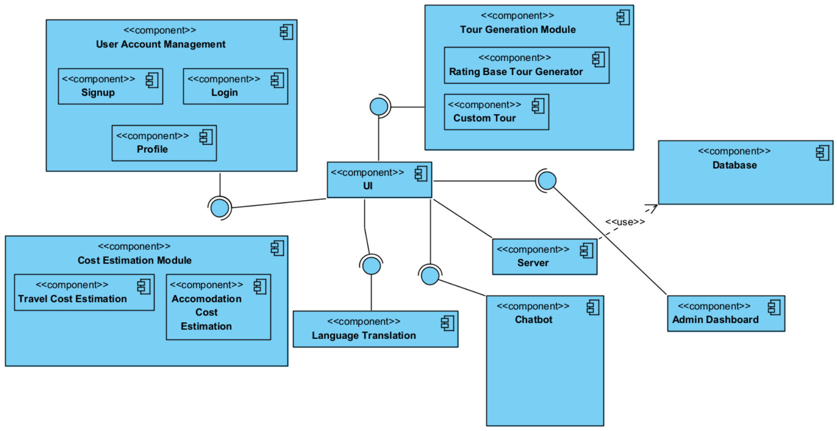


FIGURE 2.23: Component Diagram

2.2.3 Deployment Diagram

As shown in Figure 2.24 a deployment diagram shows the hardware of your system and the software in that hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration.

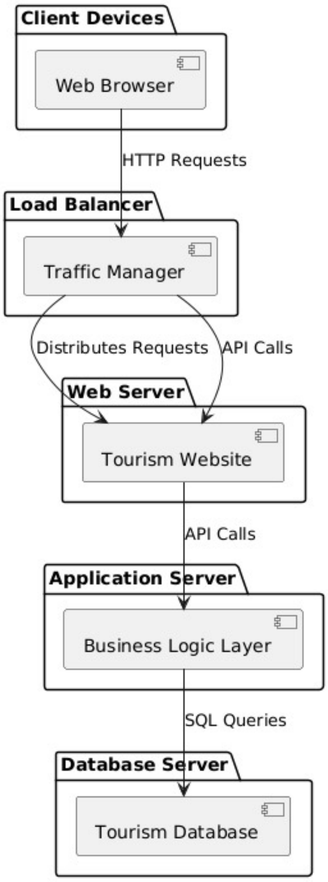


FIGURE 2.24: Deployment Diagram

2.3 User Interface Design

User interface (UI) design plays a crucial role in shaping a product's appearance, interactivity, usability, behavior, and overall user experience. A well-designed UI can significantly enhance a user's interaction with a product, making it intuitive, efficient, and enjoyable to use. Conversely, poor UI design can lead to user frustration and disengagement.

2.3.1 Wireframes

As illustrated in Figure 2.25 - Figure 2.30 wireframe is a basic visual interface guide that suggests the structure of an interface and the relationships between its pages. They serve as a blue print that defines each Web page's structure, content and functionality. Wireframes are created before any design work is started so that the focus is on layout without the distraction of color and visual elements.

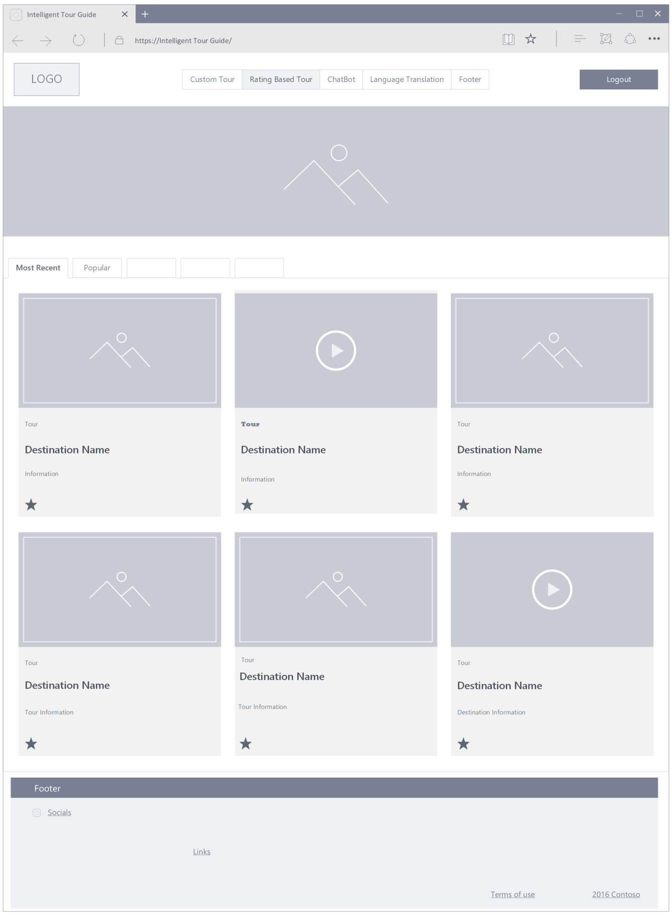


FIGURE 2.25: Home Page

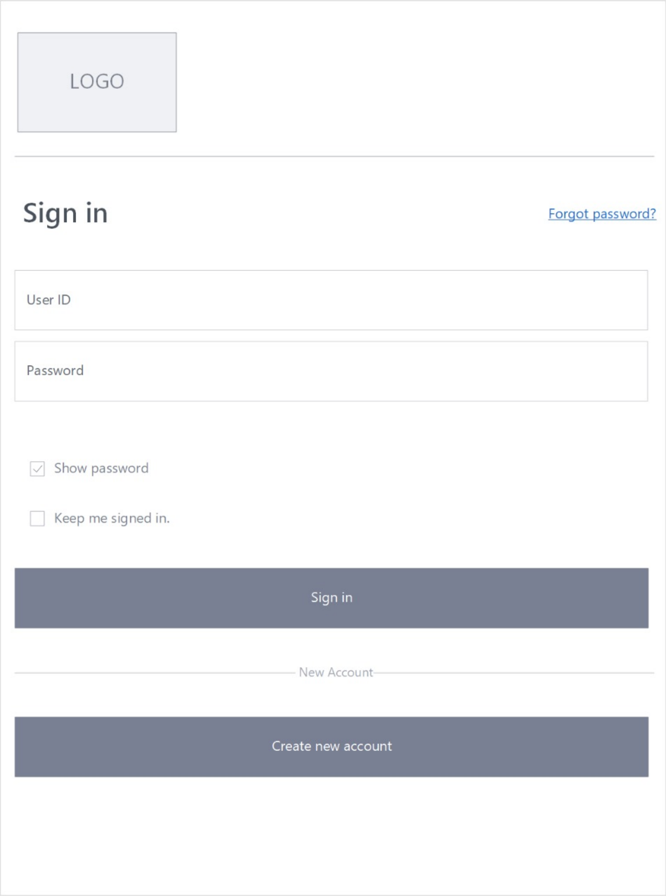


FIGURE 2.26: Login And Sign up

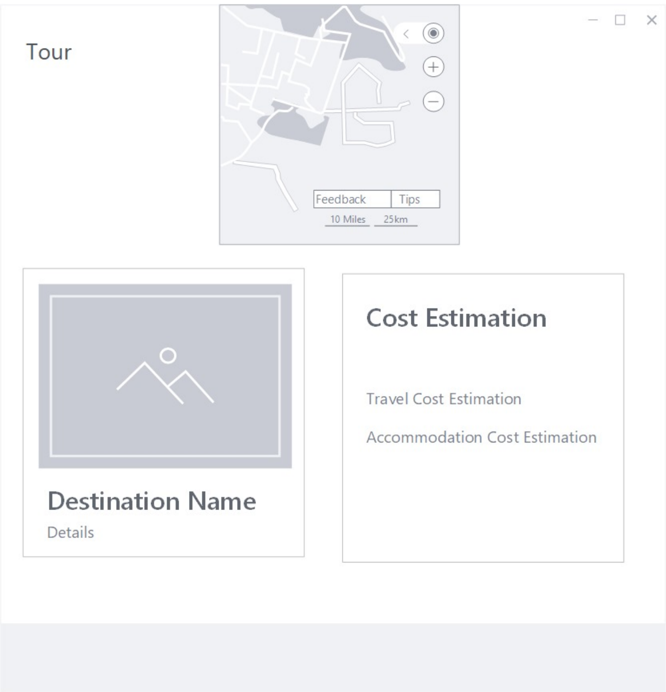
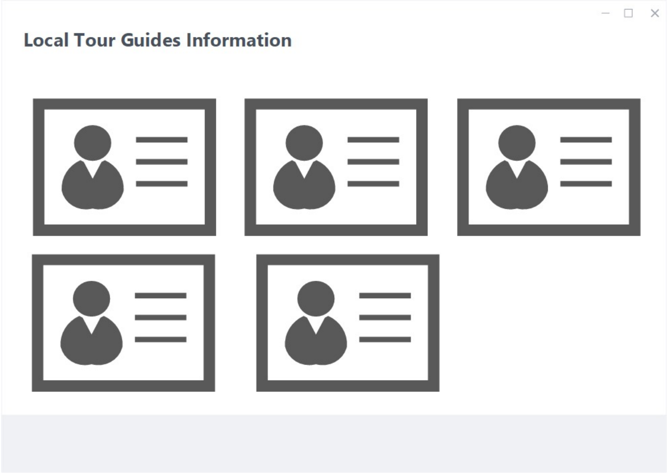
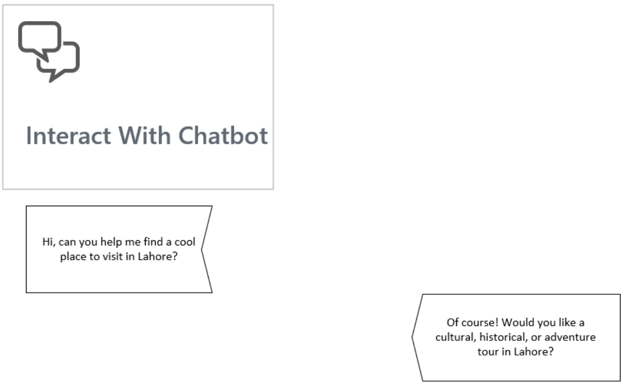


FIGURE 2.27: Individual Itenary



Local Tour Guides Information

FIGURE 2.28: Tour Guides



Interact With Chatbot

Hi, can you help me find a cool place to visit in Lahore?

Of course! Would you like a cultural, historical, or adventure tour in Lahore?

FIGURE 2.29: Chatbot Interaction

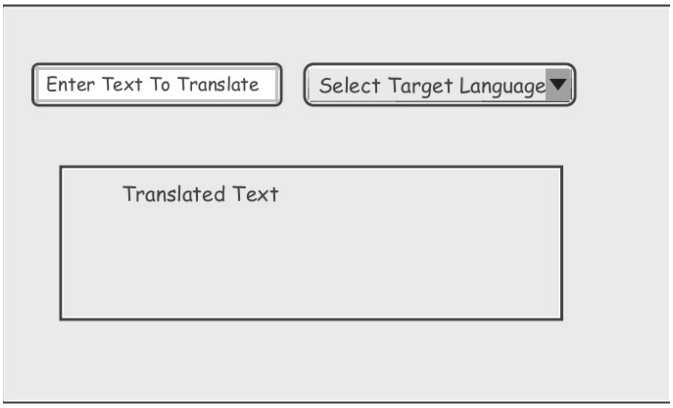


FIGURE 2.30: Text Translation

2.4 Database Design

As illustrated in Figure 2.31 a well-designed database gives you access to current and accurate information. A proper design is crucial for meeting your objectives when working with a database.

2.4.1 ER Diagram

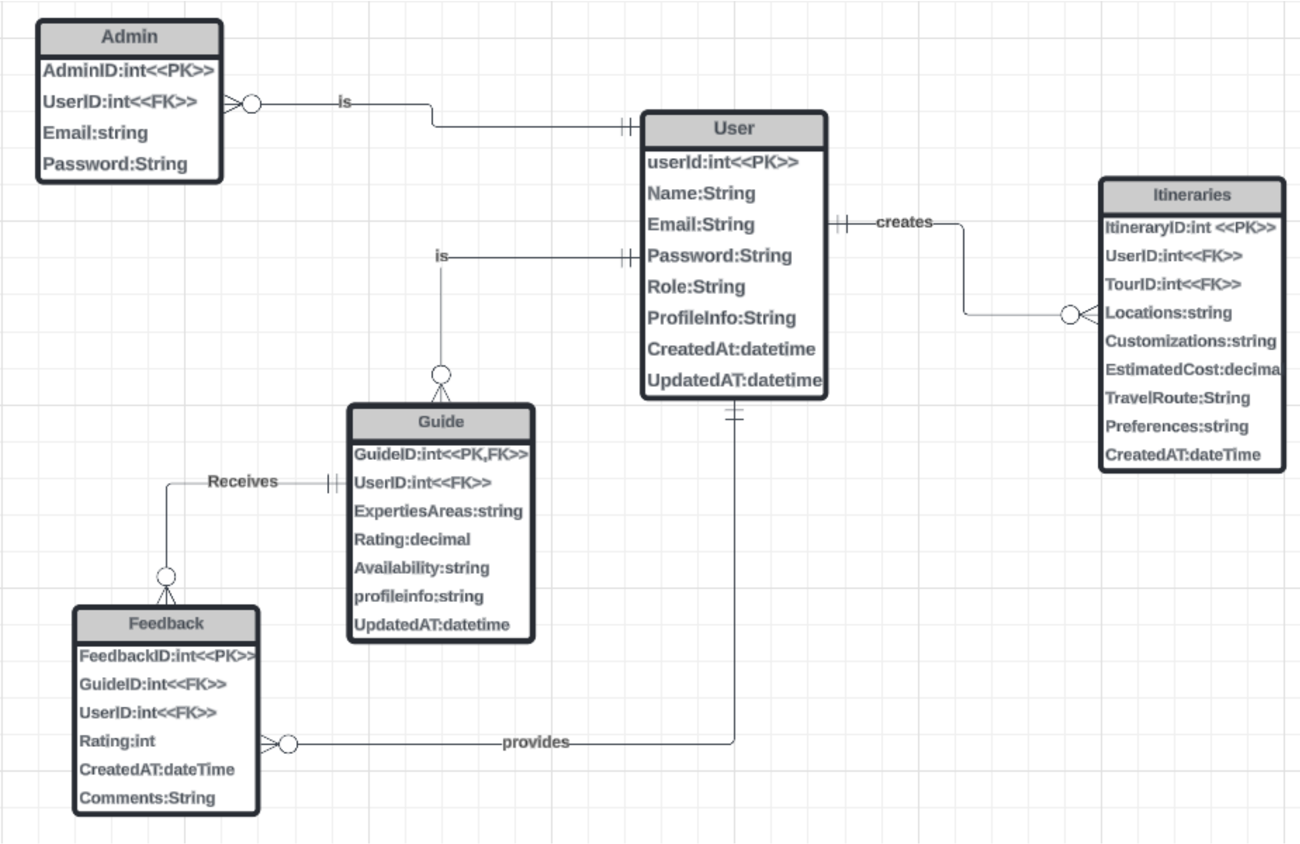


Figure 2.31: ER Diagram

References