Hamdard University
Department of Computing
Final Year Project

AI Powered Tourism App
(FYP-002/FL24)

Software Requirements Specifications

Submitted by
Daniyal Ali (2460-2021)
Ali Ahmed (2280-2021)
Ahmed Ali (1844-2021)

Supervisor(s)
Mr. Iqbal-Ud-Din
Dr. Khurram Iqbal

Fall 2024

Document Sign off Sheet

Document Information

Project Title	AI Powered Tourism App
Project Code	FYP-002/FL24
Document Name	Software Requirements Specifications
Document Version	1.0
Document Identifier	FYP-002/FL24-SRS
Document Status	Final
Author(s)	Ali Ahmed
Approver(s)	Mr. Iqbal-Ud-Din
Issue Date	16/January/2025

Name	Role	Signature	Date
S. Daniyal Ali	Team Lead		16/Jan/2025
Ali Ahmed	Team Member 2		16/Jan/2025
Ahmed Ali	Team Member 3		16/Jan/2025
Mr. Iqbal-Ud-Din	Supervisor		
Dr. Khurram Iqbal	Co-Supervisor		15/Jan/2025
Mr. Mohsin Raza Khan	Project Coordinator		

Revision History

Date	Version	Description	Author
16/01/2025	1.0	Initial	Ali Ahmed

Definition of Terms, Acronyms, and Abbreviations

Term	Description
AI	Artificial Intelligence
DBMS	Database Management System
IDA	App Design Document
APIs	Application programming interface
LAN	Local Area Network
APP	Application

Table of Contents

1. Introduction

Smart Travel Companion AI-Powered Tourism App Discover the Future of Tourism in Pakistan Leverage AI to provide custom itineraries, user based recommendations system, interactive guides and so on. Our app gives your travel moments a seamless, effective, and improved experience based on your preference whether you are planning a trip or exploring on the go.

1.1 Purpose of Document

This Software Requirements Specification (SRS) document outlines the functional and non-functional requirements that guide the development of the AI-Powered Tourism App. It serves as a guide for the development and testing of the system.

1.2 Intended Audience

IDA App Design Document this document is useful for the development team, end users (travelers and tourists), project supervisors, Tester, marketing and sales teams With this, it makes sure that everyone is on the same page regarding the goals, features, and requirements of the AI-Powered Tourism App.

2. Overall System Description

2.1 Project Background

With this booming tourist industry in Pakistan, there is a demand for better solutions around traveling. The old-school process of trip planning and routing is rarely personalized or efficient. This is where the AI-Powered Tourism App comes in, utilizing the power of art to offer travelers user based recommendations, real- time information, and smooth navigation to their destination. In order to simplify travel planning and improve user engagement — and offer a modern solution for discovering and exploring. The app will change the way users travel, and will make it beyond easy.

2.2 Problem Statement

Travelers exploring Pakistan encounter diverse challenges, from navigating unfamiliar regions to customizing their experiences a wealth of options. An AI-powered tourism app tackles these complexities by harnessing sophisticated algorithms to deliver personalized recommendations, streamline itinerary planning, and offer real-time insights. By enhancing accessibility, improving convenience, and fostering cultural appreciation, this technology not only enriches travel experiences but also contributes to a more inclusive and vibrant tourism landscape in Pakistan.

2.3 Project Scope

AI-powered tourism application focused on enhancing user travel experiences through user based recommendations, travel planning, and real-time updates. The scope of this project explicitly defines what aspects will and will not be considered, along with key assumptions guiding system evolution.

2.4 Not In Scope

1. E-commerce – the app will not feature direct booking or purchasing for flights, accommodation, or other travel-based services.
2. High level social networking: Our app won't allow using profile to create content, send messages, etc.
3. No live chat or customer service integration.
4. While the app will support offline access for some tools (such as cached maps and guides), full offline capabilities for all features are not in scope currently.

2.5 Project Objectives

User-Friendly Interface: Design an intuitive and responsive interface that enhances user interaction and accessibility, making travel planning and navigation seamless and enjoyable.

Chatbot Integration: Implement a conversational chatbot capable of understanding user queries, providing personalized recommendations, assisting with bookings, and offering real-time support throughout the journey.

Trip Recommendations by User Interest: Utilize Recommendation algorithms to analyze user preferences, past travel behavior, and demographic data to suggest personalized trip itineraries, including attractions, activities, and dining options aligned with user interests.

2.6 Stakeholders & Affected Groups

Project Stakeholders:

Development Team: Designers/developers/testers/managers

Marketing and Sales Team: User acquisition and promotion.

Affected Groups:

End Users i.e. Travelers gaining the most out of personalized itineraries and real-time recommendations.

Tour Operators and Travel Agencies: Working together to deliver packaged experiences.

2.7 Operating Environment

Smartphone device: At least android version 10 or above.

The app aims to access any external source by integrating with its respective APIs — such as weather, flight, and local attraction APIs — to provide users with real-time data.

Network Requirements: App will require stable internet connection, which will work well over Wi-Fi and Cellular data (4G/5G).

Databases: We will use Firebase (for real-time database and authentication services) in our app; it provides a NoSQL scalable database and integrates with other Google Cloud services.

2.8 System Constraints

Performance: The application must handle at least 10,000 concurrent users without noticeable latency (low execution time).

Firebase data storage must optimize for speed in retrieval and storage management under free and pay plans.

Security : The application will use basic password-based authentication to ensure secure access to user accounts.

AI Powered tourism App will operate on Android devices with internet access.

2.9 Assumptions & Dependencies

User Devices: We assume that some users with smartphones or tablets (running Android 10 or above).

Internet Connectivity: Users should have a stable internet connection (Wi-Fi, 4G, 5G) for the app functionalities.

APIs: If the application relies on APIs for real-time data (weather, flight information, local attractions, etc), those need to be reliable and always up.

Quality: Testing and evaluating features to ensure the highest user experience quality.

Firebase: Usage of Firebase to handle real-time database, user authentication services and other backend functionalities.

3. External Interface Requirements

3.1 Hardware Interfaces

Android Devices:

Logical Structure: The application will be installed on smartphones and tablets running Android.

Physical Addresses: Compatible with devices that have a minimum of 4GB RAM and At least android version 10 or above.

Expected Behavior: The application should utilize device sensors (e.g., GPS for location tracking).

Cloud Servers:

Logical Structure: Firebase cloud services used for storage, processing and real time database.

Physical Addresses: The application will interact with Firebase endpoints for database access.

Expected Behaviour: Firebase allows quick response times for data retrieval and storage, so tourism data is immediately available to users.

3.2 Software Interfaces

Database Management System:

Name of Application: Firebase

Application External Owner: Google (Friendly to open-source)

How the Interface Works: Firebase will handle user data travel records and preferences using its real time database and Firestore to store and retrieve data.

Backend Framework:

Application Name: Firebase Functions

Google (open-source friendly) application external owner

Interfaces specifics: The app will utilize Firebase Functions to implement the backend logic, take care of requests, and communicate between frontend and backend using serverless APIs.

3.3 Communications Interfaces

Network Connectivity: User needs to have an internet connection to sync the data in real time with Firebase and travel updates with user preferences.

APIs Local Area Networks (LAN) — APIs to support local servers used by organizations to share localized travel data.

APIs: firebase APIs, helping to connect react native frontend with backend services

4. System Functions / Functional Requirements

4.1 System Functions

Ref #	Functions	Category	Attribute	Details & Boundary Constraints
R1.1	Record user travel preferences and inputs	Evident	System Response Time	Data entry and saving must occur within 5 seconds of submission.
R1.2	Provide-travel recommendations based on preferences	Hidden	Recommendation Accuracy	Recommendations must be at least 85% relevant based on user preferences and behavior analysis.
R1.3	Offer personalized itineraries and follow-ups	Evident	Usability	Itineraries and suggestions should be clear, visually appealing, and easy to understand.
R1.4	Adjust recommendations dynamically based on inputs	Hidden	Adaptive Algorithm	Recommendations must update dynamically in real-time as users adjust preferences or provide new inputs.
R1.5	Display travel history and saved itineraries	Evident	Data Retrieval	Historical travel data should be accessible within 3 seconds upon request.
R1.6	Secure user login and data storage	Evident	Security	User authentication must utilize secure passwords and data encryption.

Attribute	Details and Boundary Constraints	Category
Response Time	When recording user travel preferences, the entry should be saved and visible within 8 seconds.	Optional
Concurrent User Load	The application must support a minimum of 50 users connected simultaneously under the Firebase free tier, with scalability for higher loads in paid plans.	Mandatory
Data Availability	Historical travel data and itineraries must be available for retrieval 24/7.	Mandatory
User Interface	The interface should be simple and easy to use for people with different levels of technical knowledge.	Mandatory
Language Support	The application must support English Language only.	Optional
Security Measures	The application will use basic password-based authentication to ensure secure access to user accounts.	Optional
Database Scope	The system will use Firebase for storing user preferences and travel data.	Mandatory
User Training	The system will provide basic training through documentation and in-app tutorials. Comprehensive in-person training is not in scope.	Mandatory
Expansion Flexibility	The application should be designed to easily incorporate new features, such as group travel planning	Optional

4.2 Use Cases

- ***Actors & use cases***

Actors:

1. Tourist: A traveller who is about to go on then or is already on a trip and wants recommendations and information.
2. Tour Guide : The chatbot driven assistant which recommends, guide and answer user question.
3. Admin: A system supervisor who maintains app content and settings, e.g. refreshing destinations and local affairs.

Use Cases:

1. View Destination Information

   Actor: Tourist
   Description: Tourist views contents of destination such as history, attractions and events

2. Get user based Recommendations

   Actor: Tourist
   Description: This program shifts through user preferences and offers personalized suggestions for places to visit, dine, and things to do.

3. Book a Tour or Activity

   Actor: Tourist
   Description: Based on the suggestions by the recommendation, or simply by viewing the available tours and activities, the tourist books a tour or activity.

4. Ask for Local Assistance

   Actor: Tourist
   Description: The tourist approaches a person and asks for directions, need a taxi, or looking for a medicine delivery.

5. Manage App Content

   Actor: Admin
   Description: Admin can update destinations, events, and activities listed in the app.

- *Use case diagrams*

Actors:

Tourist is going to be attached with all different types of use cases (View Destination Information, Get user base Recommendations, etc.).

Tour Guide: This will be connected to the use cases of providing information, recommendations, and responding to questions.

Admin would be that will indeed be connected to the use case of admin managing the app content.

- *High level, essential use cases*

Get user based Recommendations:

Actor: Tourist
Description: In a nutshell, this use case provides travels suggestions based on their preferences, current location, and historical data.
Precondition: Tourist is logged in and has given relevant preferences (i.e. activities type, budget).
Postcondition: A list of recommendations is returned by the system.

View Destination Information:

Actor: Tourist
Description: This allows tourists to show other details related to different travel spots.
Precondition: The tourist chooses a destination.
Postcondition: Displays information about the destination, its history, attractions and current events.

Book a Tour or Activity:

Actor: Tourist
Description: Tourist can be able to book activities / tour through the app directly.
Precondition: Another tourist has decided on a destination or activity to check out.
Postcondition: A booking confirmation is sent to the user.

Ask for Local Assistance:

Actor: Tourist
Description: Tourists can consult the chatbot guide for local help (directions, emergency assistance).
Precondition: The tourist requests assistance.
Postcondition: The system provides relevant local information or actions.

4.2.1 List of Actors

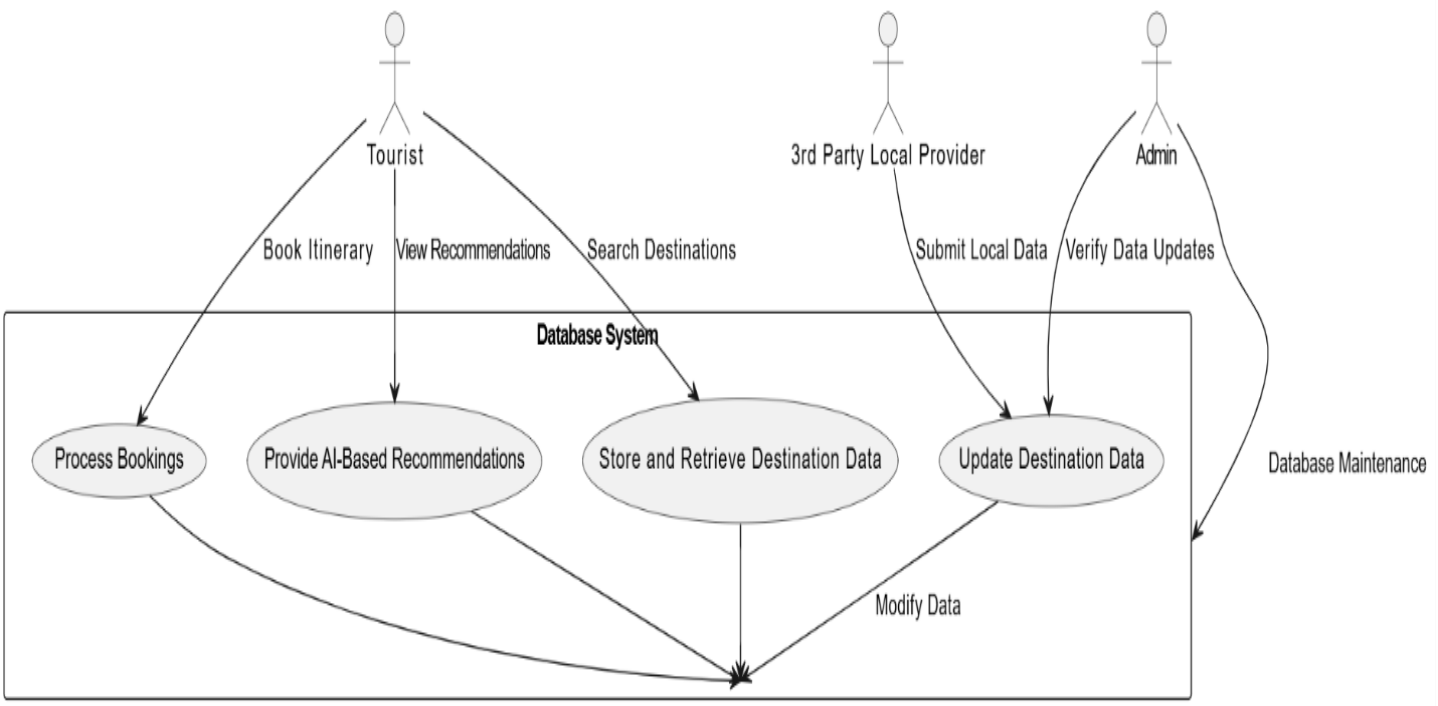
Actor	Role and Use Cases
Tourist/User	Searches for destinations, sets preferences, views recommendations, books itineraries, and provides feedback.
Administrator	Manages user accounts, monitors system performance, and updates travel information in the app.
Tour Operator	Provides details on tours, offers, and packages; collaborates with the app to update real-time availability.
Travel Guide	Shares detailed information about destinations, reviews itineraries, and provides recommendations.
Local Vendor	Lists services such as transportation, accommodations, or activities for tourists to book through the app.
Payment Gateway	Processes payments securely and confirms booking transactions.
Firebase System	Manages backend functionalities like database storage, user authentication, and real-time updates.

4.2.2 List of Use Cases

Use Case	Description
Search Destination	Allows users to search for travel destinations based on their preferences and filters.
View Recommendations	Provides travel suggestions using chatbot.
Book Itinerary	Enables users to plan and book travel itineraries, including accommodations, transport, and activities.
Manage User Profile	Allows users to create, edit, and manage their personal profiles and travel preferences.
Real-Time Updates	Provides real-time updates on bookings, travel alerts, or changes to itineraries.
Payment Processing	Facilitates secure online payment for bookings and purchases within the app.
Rate and Review Services	Allows users to leave feedback and reviews for destinations, services, or travel experiences.
Local Guide Integration	Connects users with local guides for detailed information or assistance during trips.
View Travel History	Displays a summary of past trips and saved itineraries for user reference.
Customer Support Access	Provides a feature for users to contact support for assistance or queries about their trips.

4.2.3 Use Case Diagram

Create the system level use case diagram



4.2.4 Description of Use Cases

Use Case 1: Search Destination

Attribute	Description
Use Case ID	UC1
Use Case Name	Search Destination
Actor	Tourist/User
Preconditions	User must have access to the app with an active internet connection.
Trigger	The user initiates a search for a travel destination.
Basic Flow	1. User enters preferences (e.g., budget, destination type). 2. App processes the input using AI. 3. App displays a list of destinations.
Alternative Flow	If no results are found, the app provides suggestions based on nearby locations or adjusted criteria.
Postconditions	The user views a list of destinations matching their preferences.
Exceptions	Internet connection loss or invalid search criteria results in an error message.

Use Case 2: Book Itinerary

Attribute	Description
Use Case ID	UC2
Use Case Name	Book Itinerary
Actor	Tourist/User
Preconditions	User must be logged into the app with a valid payment method configured
Trigger	User selects an itinerary and proceeds to book.
Basic Flow	1. User selects itinerary details (e.g., dates, preferences). 2. App confirms availability. 3. User makes a payment. 4. App confirms the booking.
Alternative Flow	If selected itinerary is unavailable, the app suggests alternatives.
Postconditions	Booking is completed, and confirmation details are sent to the user.
Exceptions	Payment failure or server timeout results in a failed booking attempt.

Use Case 3: View Recommendations

Attribute	Description
Use Case ID	UC3
Use Case Name	View Recommendations
Actor	Tourist/User
Preconditions	User must have input preferences or search history in the app.
Trigger	The user navigates to the recommendations section.
Basic Flow	1. App analyzes user data (preferences and history). 2. Recommendations are displayed.
Alternative Flow	If no data is available, the app provides popular destinations as recommendations.
Postconditions	The user explores personalized travel suggestions.
Exceptions	System error or data unavailability results in default suggestions being shown

Typical Course of Events

Actor Action	System Response
1. The use case begins when a user logs into the app using valid credentials.	System authenticates the user and grants access to the personalized dashboard.
2. User searches for tourist destinations based on preferences like budget, type of attraction, and duration.	System processes the input and displays a list of matching destinations.
3. User selects a destination to view detailed information, including travel guides, reviews, and nearby attractions.	System retrieves and displays the detailed information about the selected destination.
4. User books an itinerary, including flights, accommodations, and tours.	system confirms availability, processes payment, and finalizes the booking.
5. User provides feedback or preferences for tailored recommendations.	System records the inputs and updates the user profile for personalized recommendations.
6. Travel Expert reviews and updates destination data to ensure accuracy and add new attractions.	System updates the database and makes the new information available to users.
7. System generates recommendations for destinations, activities, and experiences based on user preferences and history.	Recommendations are displayed in the user dashboard for easy access.
8. Admin manages user accounts, including creating, editing, or deactivating accounts.	System updates the user account database accordingly.
9. User logs out of the app after completing their tasks.	System ends the session and securely logs out the user.

Alternative Course Of Events

User enters incorrect credentials.
System prompts the user to try again or to start the password recovery process.

User submits incomplete search criteria (such as budget or travel length missing).
System fills in missing or incomplete fields automatically to narrow down the search results.

Searched destination has no information available.
System notifies the user and prompts them to look at popular or alternate destinations.

The itinerary booking is left unfinished by the user.
System logs the itinerary-in-progress and informs the user to finalize it later.

Travel Expert flags mistakes or gaps in destination data.
System alerts the Travel Expert to modify or correct the data.

Destination data update is unsuccessful due to incorrect input or system malfunction.
System notifies the Travel Expert of the failure and requests valid input for reprocessing.

Insufficient user profile data to generate recommendations.
System warns the user and encourages them to provide a wider variety of preferences to enable meaningful recommendations.

No itinerary recommendations are generated due to incomplete or invalid inputs.
System notifies the user to provide further or corrected information to create accurate options.

Exporting a travel plan or itinerary fails due to a system problem or incorrect file format selection.
System alerts the user about the error and advises them to retry or select another format.

Notifications (e.g., reminders, updates, promotions) fail to deliver due to system errors or disabled notifications.

5. Non - Functional Requirements

5.1 Performance Requirements

Time: App must respond to user actions in a maximum of 5 seconds.

Concurrent Users: Support 100 concurrent users without performance issue

Optimization: Search results must load in less than 5 seconds

Uptime: 99.9% uptime for uninterrupted service.

5.2 Safety Requirements

Protect user data with secure storage and compliance with privacy laws.

Ensure safe financial transactions with encryption and secure gateways.

Provide accurate and verified travel information.

Use strong authentication to prevent unauthorized access.

Include emergency contact features for user safety.

5.3 Security Requirements

User Login: A secure authentication method (username / password / 2-factor authentication) that prevents unauthorized access.

Data Encryption: Encrypt sensitive data during transfer and at rest to protect user privacy.

Role-Based Access Control: Control data visibility according to the user role (e.g., admin, traveler, travel expert).

Maintain Regular Security Updates: Make regular updates to plug vulnerabilities and enhance the system security.

Secure API Calls: Ensure you are using secure protocols and proper authentication for any third-party API calls.

5.4 Reliability Requirements

Uptime: 99.9% uptime guarantee to ensure availability.

Robustness: The system should be resilient and detect errors.

Consistency of Data: You require that all user data and preferences should be consistent across the same session and device.

Scalability: The application must handle higher levels of user demand without loss of performance or reliability.

5.5 Usability Requirements

Intuitive Interface: Easy-to-navigate design for all users.

Personalized Recommendations: AI-driven, user-specific suggestions.

Quick Response: Fast loading and minimal interaction delays.

Consistent Design: Uniform interface across the app.

Feedback: Easy user feedback for continuous improvement.

5.6 Supportability Requirements

Error Logging & Diagnostics: Track errors and allow remote troubleshooting.

User Feedback: Easy reporting of issues by users.

Regular Updates: Timely software updates for security and performance.

Scalable & Monitored: Scalable architecture with performance monitoring tools.

5.7 User Documentation

Overview: Brief app introduction and key features.

Getting Started: Installation, registration, and setup guide.

Features & FAQs: Main functionalities and common questions.

Troubleshooting & Privacy: Issue resolution and data security info.

6. References

List References

https://www.paradigmshift.com.pk/tafreego-pakistan/
https://mofa.gov.pk/meet-pakistan-app 8.
https://kodytechnolab.com/blog/travel-app-ideas/