Hamdard University

Department of Computing

Final Year Project

AI Powered Tourism App
(FYP-002/FL24)

Software Design Specifications

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

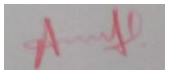
Supervisor(s)
Mr. Iqbal-Ud-Din
Dr. Khurram Iqbal

Fall 2024

Document Sign off Sheet

1.1.1 Document Information

Project Title	AI Powered Tourism App
Project Code	FYP-002/FL24
Document Name	Software Design Specifications
Document Version	1.0
Document Identifier	FYP-002/FL24-SDS
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		16/01/2025
Ali Ahmed	Team Member 2		16/01/2025
Ahmed Ali	Team Member 3		16/01/2025
Mr. Iqbal-Ud-Din	Supervisor		
Dr. Khurram Iqbal	Co-Supervisor		16/01/2025
Mr. Mohsin Raza Khan	Project Coordinator		

Revision History

Date	Version	Description	Author
15/01/2025	1.0	Initial	Ali Ahmed

Definition of Terms, Acronyms, and Abbreviations

[This section should provide the definitions of all terms, acronyms, and abbreviations required to interpret the terms used in the document properly.]

Term	Description
AI	Artifical Intelligence
UI	User Experience
NLP	Natural Language Processing

2 Table of Contents

3 Introduction

3.1 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

3.2 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.

3.3 Document Convention

• Headings: Arial, Bold, Size 16
• Subheadings: Arial, Bold, Size 14
• Body Text/Descriptions: Arial, Italic, Size 12
• Table Fields: Arial, Regular, Size 12, 10

3.4 Project Overview

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 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.

3.5 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.

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.*

4 Design Considerations

User-Centric Design:

User-Friendly UI: An intuitive app UI is also crucial for everyone to use easily based on their age group and technical proficiency.

Individualization: The Application should give personalized suggestions depending at the user choices, area, and ancient records.

Recommendation System: Approach that suggests destinations, activities, and accommodations based on the user profile and preferences.

Scalability and Performance:

Utilize a Cloud-Based Architecture: Implement a cloud architecture that can handle large amounts of user data and requests

4.1 Assumptions and 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.

4.2 Risks and Volatile Areas

Technical Risks:
System Integration Challenges: The app's functionality could be affected by difficulties in integrating third-party APIs, such as maps, weather, or ticket booking services.
Cloud Services and Firebase: While using Firebase and other cloud-based services, there may be periods of downtime if these services become unresponsive.
Data Accuracy: Inaccurate or outdated real-time data (e.g., traffic, weather, or availability for bookings) may trust.

User Experience Challenges:
Misinformation in AI: AI chatbots/recommender system may provide wrong or materialistic information to user, which cause user to be unsatisfied.
Too complex for non-tech users: The app may be too complicated for users who are less tech-savvy, leading to a lower adoption rate.
Offline Capabilities: The app may not offer full offline functionality, which could be an annoyance for users in low-connectivity regions.

Market and Competition Risks:
High Competitive analysis Focus on a specific demographic may limit appeal: The service's focus on millennial travelers may exclude older travelers or families who also travel.
Changing Trends: Travel trends and user behavior can change rapidly and may make some features less relevant over time.

Operational Risks:
Coordination between Teams: Different teams (developers, designers, and stakeholders) not being on the same page could delay delivery.
Testing & Quality Assurance: Lack of thorough testing can leave you with buggy, difficult-to-use software.
Scalability Issues: A sudden influx of users can lead to infrastructure overloads, thus hindering performance

User Adoption Risks:
Trust Building: Users might be reluctant to use the app because of concerns about data privacy or reliability.
Aspects that might cause trouble: Insufficient or unsuccessful marketing programs can result in bad user acquisition.
Handling of Feedback: Not responding to users feedback in time can hurt the reputation of the app.

5 System Architecture

The AI-powered tourism app follows a modular architecture for flexibility, scalability, and maintainability. The architecture comprises:

- Client Layer:
- Mobile: Provides a user-friendly platform for interactions via mobile apps.
- User Interaction: Handles inputs, displays recommendations, and offers AI chatbot functionality.

- Application Layer:
- Backend API: Manages communication between the client layer and backend services.
- Business Logic: Processes user inputs and applies application rules.
- Chatbot & Recommendation Engine: Delivers personalized responses and suggestions based on user data.

- Data Layer:
- Firebase Integration: Provides authentication, real-time database, and cloud storage.
- Third-Party APIs: Enables access to external services like maps, weather, and booking platforms.
- Data Analytics: Analyzes user behavior and trends to improve service.

5.1 System Level Architecture

1. User Interface Layer:

Mobile Application: Cross-platform app (React Native) with features like recommendations, trip planning, and chatbot support.

2. Application Logic Layer:

Backend Server: Manages business logic, API routing, and database communication.

3. AI Modules:

Chatbot: NLP-powered for user queries.
Recommendation Engine: Offers personalized suggestions.
Authentication: Secure login via Firebase Authentication.

4. Data Management Layer:

Firebase: Real-time database for user data and cloud storage for media files.
Analytics: Tracks user behavior for improvement.
Third-Party APIs: Google Maps, weather, and booking platforms.
API Gateway: Ensures secure communication between components.

5. Workflow:

Users interact via the UI layer.
Requests are processed by the backend server.
Data is retrieved/updated from Firebase.
AI modules generate responses or recommendations.
Responses are sent back to the user.
System performance is continuously monitored.

5.2 Software Architecture

The software architecture for the AI-powered tourism app is designed to ensure modularity, scalability, and maintainability. It adopts a layered architecture pattern, enabling separation of concerns and efficient development.

1. Presentation Layer:
Mobile Application: Developed using cross-platform frameworks like React Native.
Provides features like personalized recommendations, trip planning, and chatbot interactions.

2. Application Layer:
Chatbot: Powered by NLP to respond to user queries.
Recommendation Engine: Delivers personalized suggestions based on user data.
Authentication: Handles user login and security using Firebase Authentication.

3. Data Layer:
Database: Firebase Real-Time Database stores user data, preferences, and dynamic content.

4. Integration Layer:
Third-Party APIs: Integrates with Google Maps, weather services, and booking platforms.
API Gateway: Manages and secures API requests.

6 Design Strategy

User-Centered Design:
Focus on the needs, behaviors, and preferences of the target users (tourists).
Prioritize features like ease of navigation, local recommendations, and personalization.

AI-Powered Personalization:
Use personalized recommendations based on user behavior, preferences, and historical data.
Implement machine learning models to suggest destinations, activities, and accommodations that suit individual users.
Offer dynamic itineraries that adapt to changes in preferences or schedules.

Scalable Architecture:
Design the app to be scalable to accommodate growing user demand, large datasets (e.g., travel locations, reviews), and AI model improvements.
Use cloud-based infrastructure for flexible resource allocation and better performance.

Intuitive User Interface (UI):
Design a clean and simple interface with easy-to-navigate sections, clear typography, and visual elements that enhance user experience.
Integrate interactive maps, real-time location tracking, and instant travel advice.

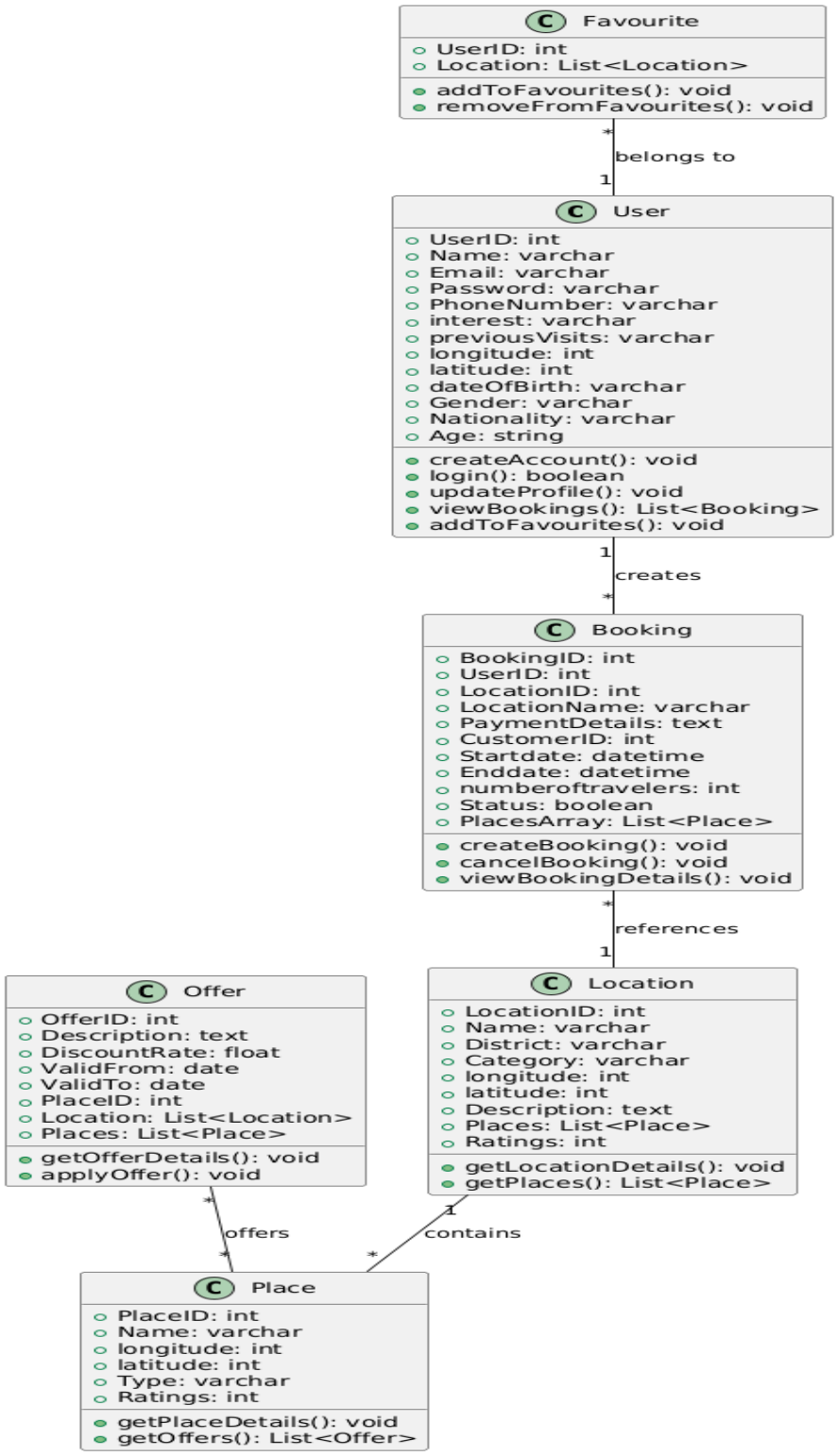
Multi-Platform Support:
Ensure the app is responsive across devices such as smartphones and tablets.
Optimize for Android platforms only.

Data Security and Privacy:
Use Firebase Authentication for secure user sign-up and login. Users will fill out a registration form, and then verify their identity via a multi-step authentication process (such as email verification, SMS verification, etc.) before gaining full access to the app.
Firebase Authentication provides a robust, secure authentication system with options for email/password, Google Sign-In, and other identity providers.

Performance and Speed:
Initially, the app will use Firebase's free plan to handle up to 20,000 user records. Firebase provides reliable backend services, but the free plan may have limitations in terms of storage and concurrent users.
As the user base grows, the app will gradually transition to Firebase's paid plan to ensure optimal performance, greater data handling capabilities, and scalability to support increasing demands without compromising speed.
Monitor app performance regularly to assess data usage and identify any bottlenecks, ensuring the app can scale as needed.

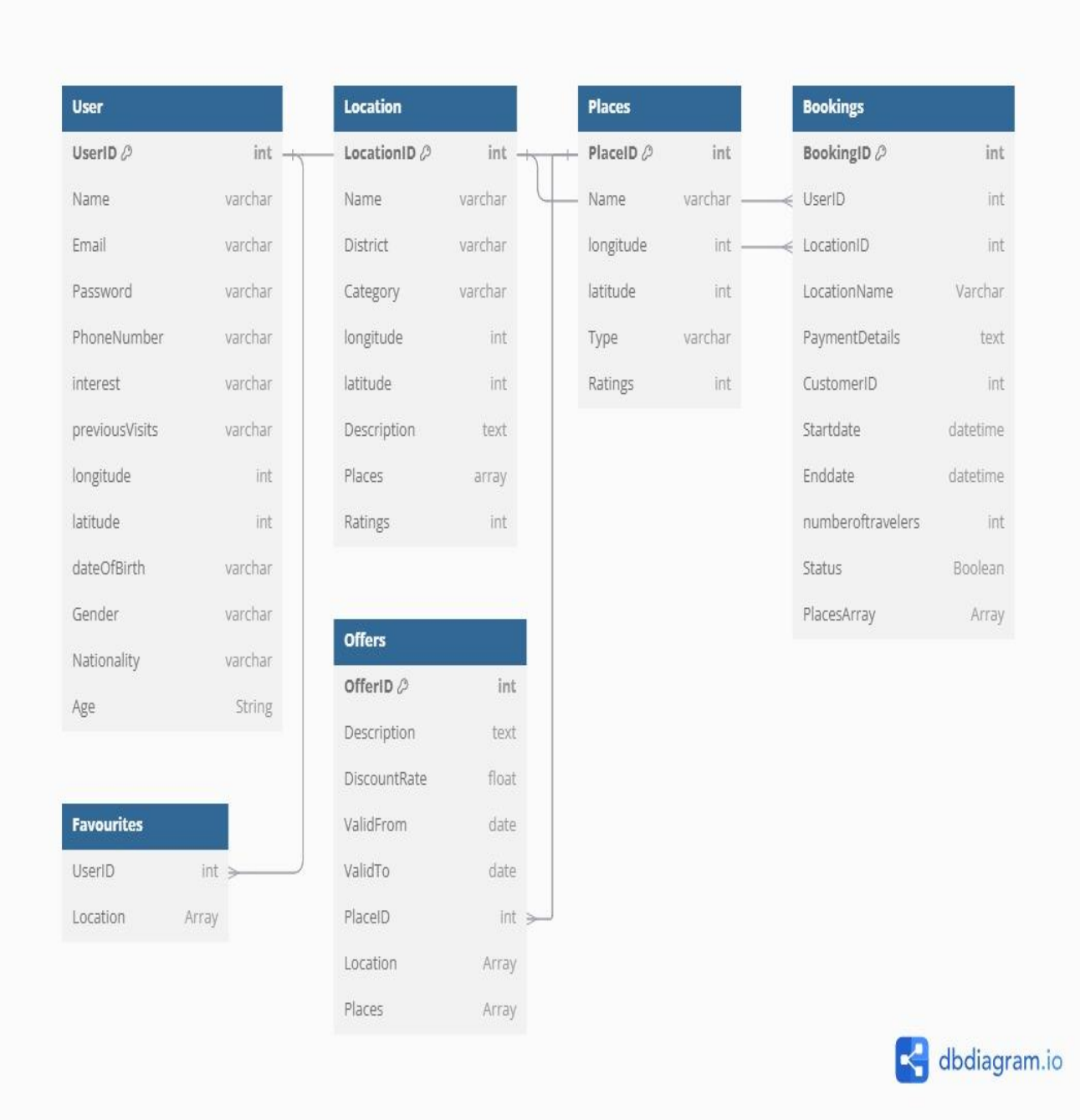
7 Detailed System Design

7.1 Design Class Diagram



7.2 Database Design

7.2.1 ER Diagram



7.2.2 Data Dictionary

7.2.2.1 Data 1 User

User	
Name	User
Alias	
Where-used/how-used	The **user** attribute stores profile details and preferences to personalize recommendations. It is used to track booking history and manage user-specific data like reviews and itineraries.
Content description	The **user** attribute in the AI-powered tourism app captures essential profile information, including preferences, booking history, and personal details. It enables personalized travel recommendations, creating customized itineraries and activity suggestions. Additionally, it tracks user interactions, such as reviews, ratings, and loyalty rewards.

Column Name	Description	Type	Length	Null able	Default Value	Key Type
Name	Name of the location	VARCHAR	255	NO	-	-
Age	Age of the user	String	255	YES	-	-
DOB	DOB of User	INT	255	YES	-	-
Longitude	Longitude value	INT	-	YES	-	-
Latitude	Latitude value	INT	-	YES	-	-
Email	Email of user	VARCHAR	-	YES	-	-
Gender	Gender of user	VARCHAR	-	YES	-	-
Nationality	Nationality of user	VARCHAR	-	YES	-	-
Phone	Phone of user	INT	-	YES	-	-

Interests	Type of places that user wanted to visit	ARRAY	-	YES	-	-
Previous Visited	Previous Visited place of user	ARRAY	-	YES	-	-

7.2.2.2 Data 2 Location

Location	
Name	Location
Alias	
Where-used/how-used	It is used to recommend nearby tourist attractions and filter results based on geographical proximity. Additionally, it helps personalize itineraries and suggest travel options based on the user's location.
Content description	In the AI-powered tourism app, the **location** attribute provides personalized travel recommendations based on the user's current or desired destination. It helps in filtering nearby attractions and activities for a more customized experience. Additionally, it plays a role in optimizing itineraries and travel routes for seamless navigation.

Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
Place ID	Unique Identifier	INT		NO	-	Primary Key
Name	Name of the place	VARCHAR	255	YES	-	-
Location ID	ID of the location	INT		YES	-	Foreign Key
Longitude	Longitude Coordinate	INT	-	YES	-	-
Latitude	Latitude Coordinate	INT	-	YES	-	-

District	On which district in country	VARCHAR	50	YES	-	-
Ratings	Rating of the place	INT	-	YES	-	-
Category	Place category	VARCHAR	-	YES	-	-
Description	Place Description	VARCHAR	-	YES	-	-

7.2.2.3 Data 3 Places

Places						
Name	Places					
Alias						
Where-used/how-used	**Where-Used**: The Places attribute is used to store related or nearby locations in the form of an array or JSON structure. **How-Used**: It helps in linking and displaying associated places for better navigation or recommendations.					
Content description	The Places attribute stores an array or JSON structure representing related or nearby locations. It provides additional contextual information, such as associated landmarks or points of interest.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**
Place ID	Unique Identifier	INT		NO	-	Primary Key
Name	Name of the place	VARCHAR	255	YES	-	-
Longitude	Longitude Coordinate	INT	-	YES	-	-
Latitude	Latitude Coordinate	INT	-	YES	-	-
Type	Type of the place	VARCHAR	50	YES	-	-
Ratings	Rating of the place	INT	-	YES	-	-

7.2.2.4 Data 4 Bookings

Bookings	
Name	Bookings
Alias	
Where-used/how-used	The "Bookings" attribute is typically used to record and track reservations or appointments made at a place, such as a hotel, restaurant, or event venue. It is used to store details like booking date, guest information, and the associated place.
Content description	The "Bookings" attribute stores information about reservations made at a place, including the date, time, and customer details. It helps manage availability, track reservations, and optimize service operations.

Column Name	Description	Type	Length	Null able	Default Value	Key Type
Booking ID	Unique Identifier for the booking	INT	-	NO	-	Primary Key
Location ID	ID of the place being booked	INT	-	NO	-	Foreign Key
Customer ID	ID of the customer making the booking	INT	-	NO	-	Foreign Key
Start Date	Start date of the booking period	DATETIME	-	NO	-	-
End Date	End date of the booking period	DATETIME	-	NO	-	-
Number of Travelers	Number of people for the booking	INT	-	YES	-	-
Status	Current status of the booking	Boolean	50	YES	Pending Time	-

Places Array	Places ID, Name, Start Date, End Date, Price	Array	50	YES	Pending Time	-
Payment Details	Payment Object contain total cost, Payment method, transaction id, Payment status	Object	50	YES	Pending Time	-
User ID	ID of the user.	int	50	YES	Pending Time	-
Location Name	The **location name** identifies a specific geographical destination or place.	Varchar	50	YES	Pending Time	-

7.2.2.5 Data 5 Favourites

Favourites						
Name	Favourites					
Alias						
Where-used/how-used	The "Favourites" attribute is used to track places or items that a user has marked as preferred or liked. It helps personalize recommendations and enhances user experience by saving their preferences for quick access.					
Content description	The "Favourites" attribute stores a list of places or items that a user has marked as preferred. It allows quick access to these selections for personalized recommendations and easier future interactions.					
Column Name	**Description**	**Type**	**Length**	**Null able**	**Default Value**	**Key Type**

User ID	ID of the user who marked the favourite	INT	-	NO	-	Foreign Key
Location Array	ID and name of the Locations marked as a favourite	ARRAY	-	NO	-	Foreign Key

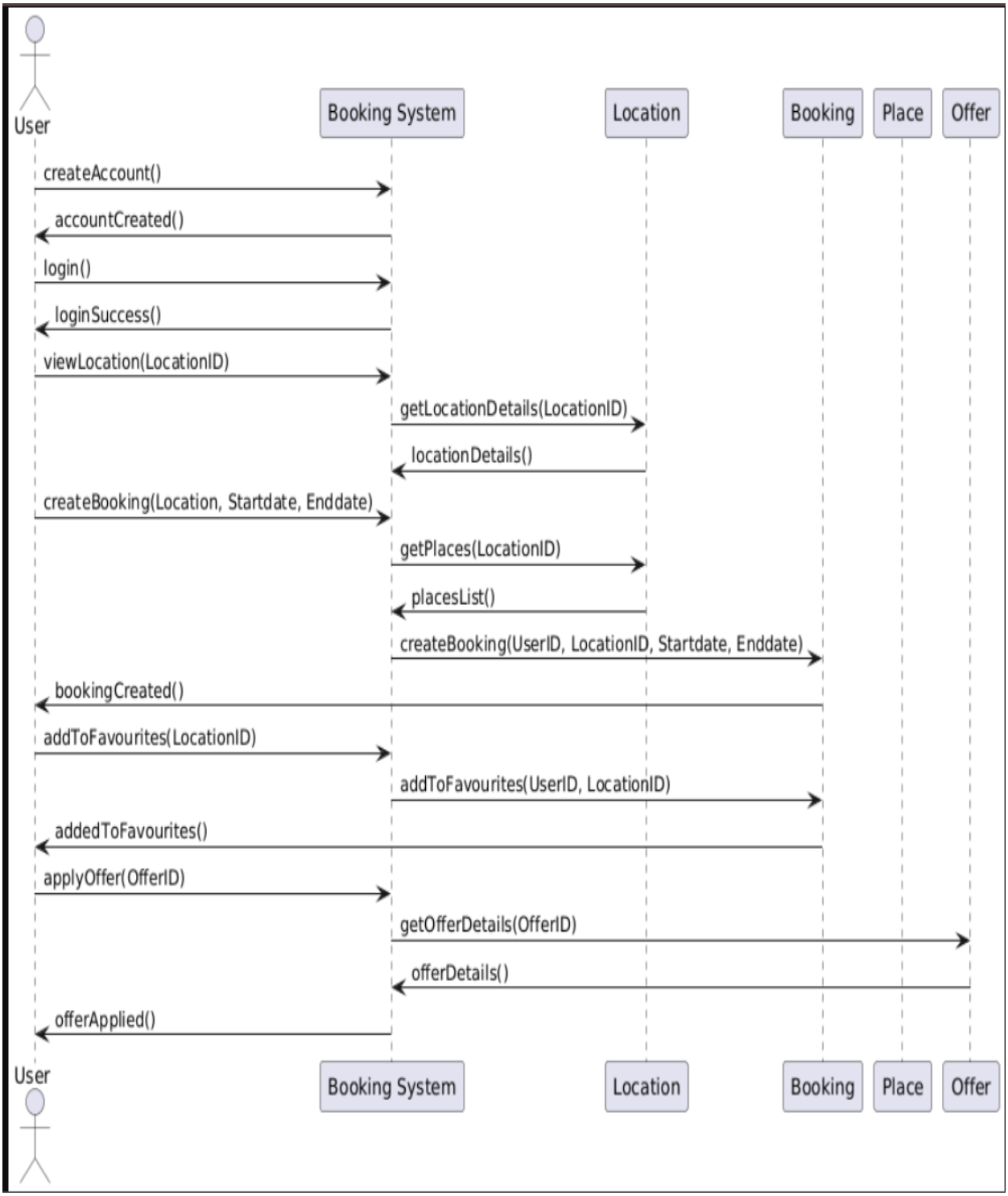
7.2.2.6 Data 6 Offers

Offers	
Name	Offers
Alias	
Where-used/how-used	The "Offers" attribute is used to store special promotions or discounts available for places or services. It helps attract customers by providing them with exclusive deals, improving customer engagement and increasing bookings or sales.
Content description	The "Offers" attribute represents special deals or discounts available at a place or service. It helps highlight promotions to customers, encouraging bookings or purchases with attractive pricing or benefits.

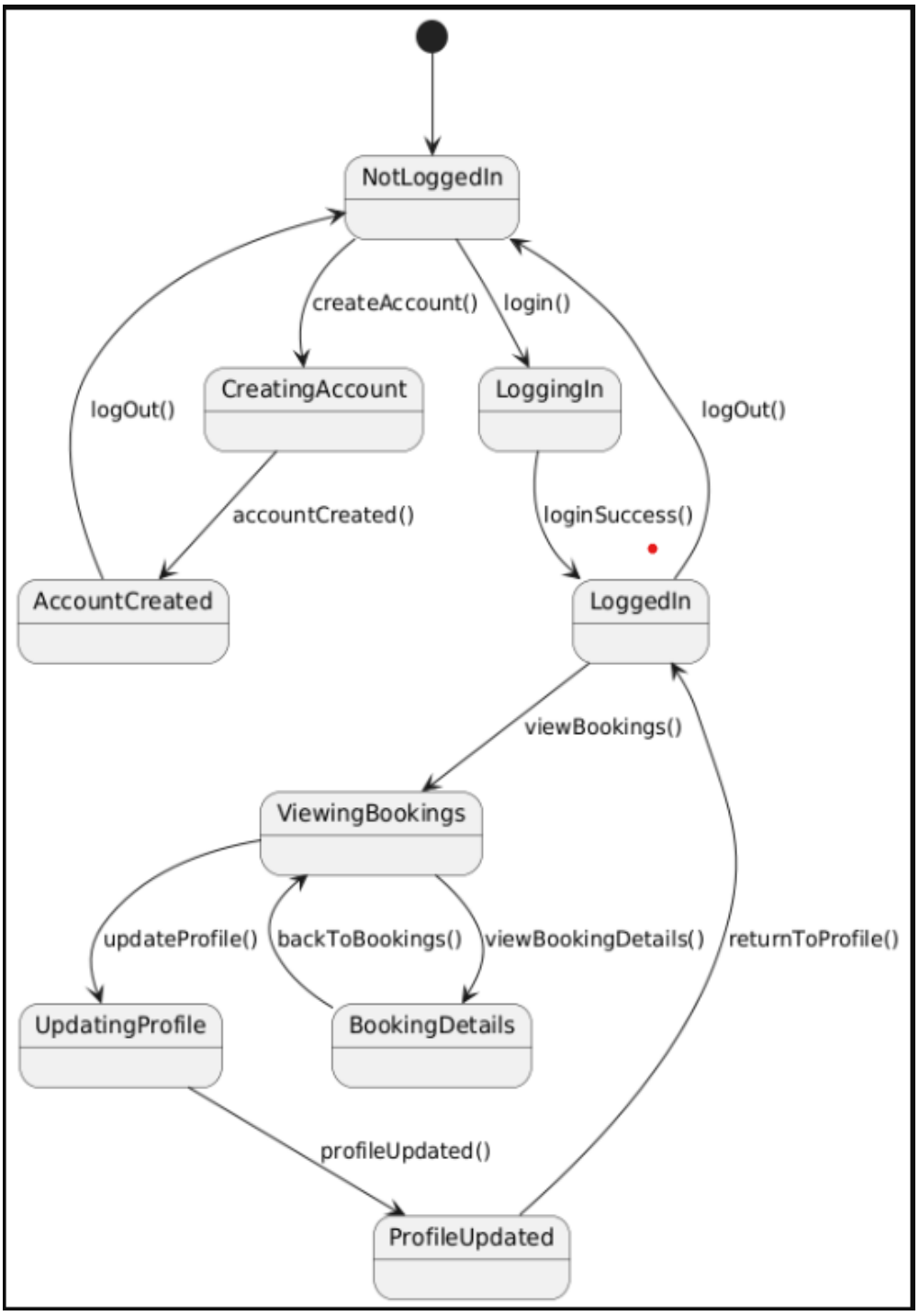
Column Name	Description	**Type**	**Length**	Nullable	Default Value	Key Type
Offer ID	Unique identifier for the offer	int		No		Primary Key
Description	Details of the offer or promotion	text		Yes		
Discount Rate	The discount percentage or amount offered	float		Yes		
Valid From	Start date when the offer is valid	date		No		
Valid To	End date when the offer is valid	date		No		
Locations Array	ID and name of locations in the offer	Array		No		Foreign Key
Places Array	ID and name of places in the offer	Array		No		

7.3 Application Design

7.3.1 Sequence Diagram



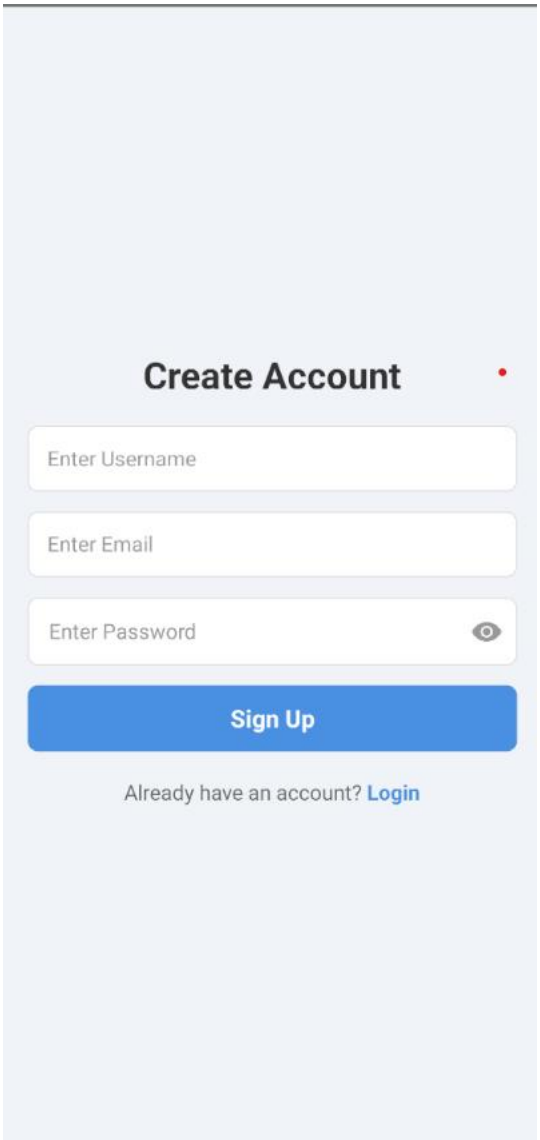
7.3.2 State Diagram



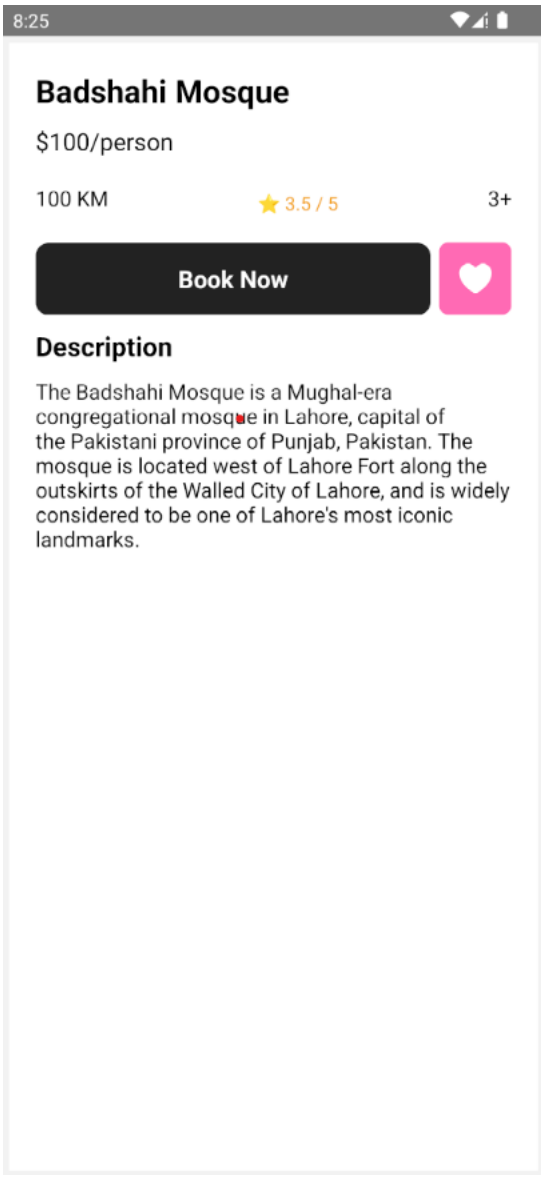
7.4 GUI Design

7.4.1 Use Case Name - Mock Screen 1

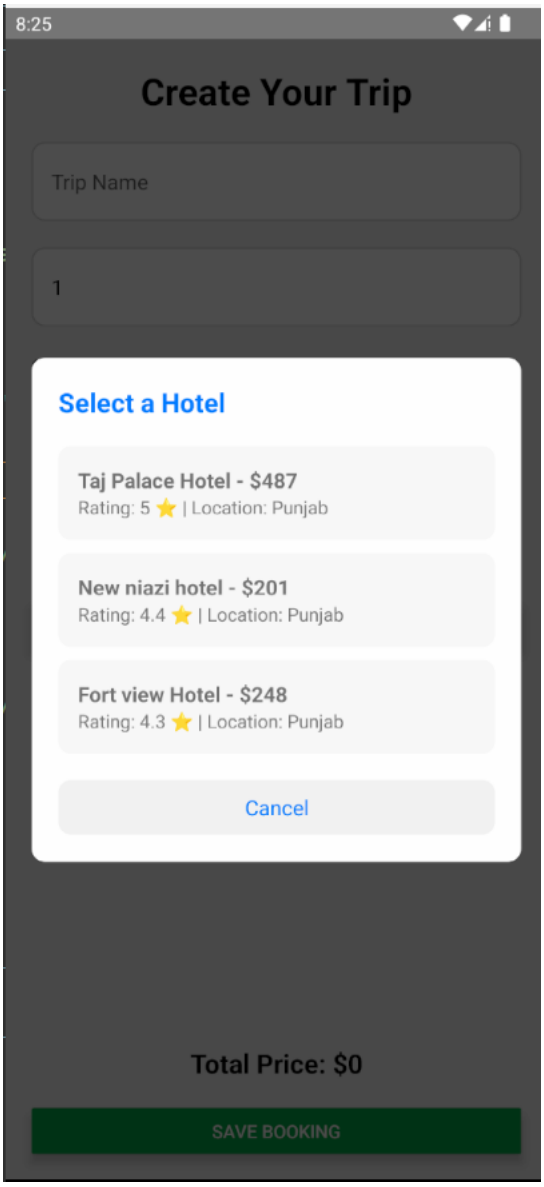
7.4.2 Use Case Name - Mock Screen 2



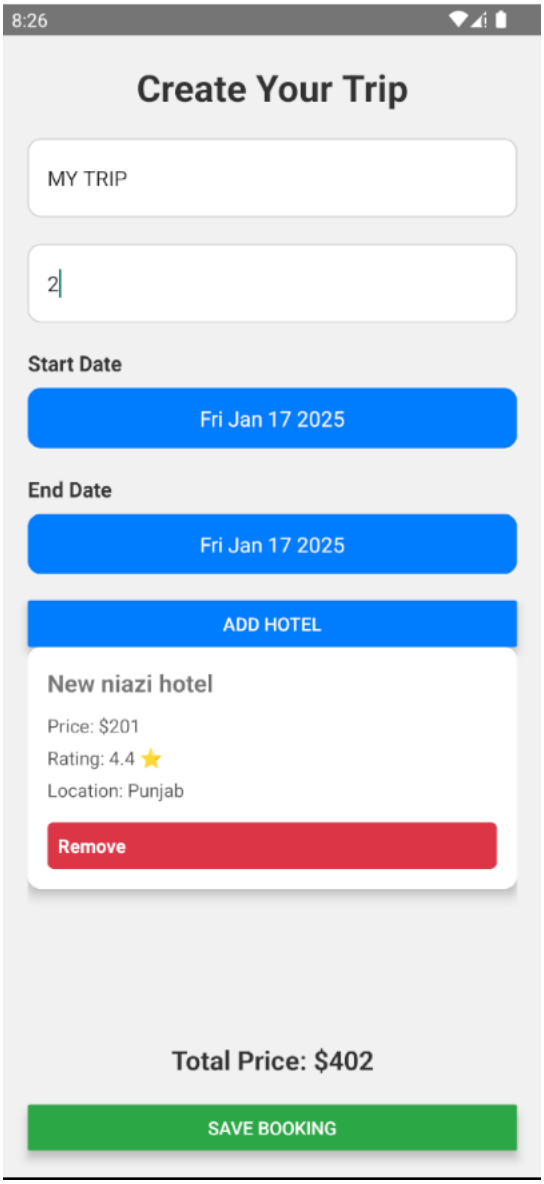
7.4.3 Use Case Name - Mock Screen 3



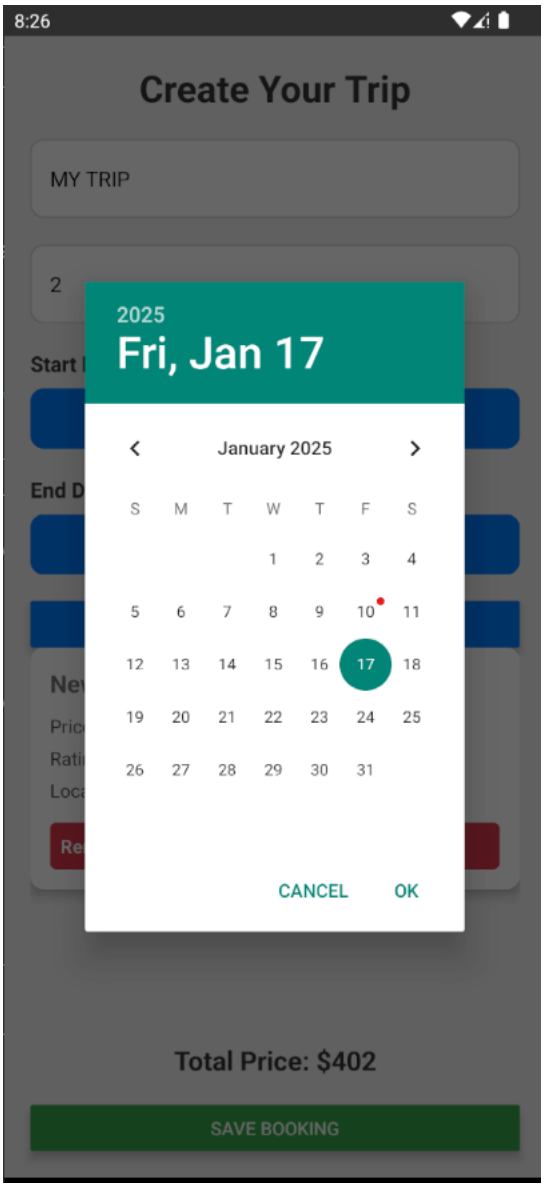
7.4.4 Use Case Name - Mock Screen 4



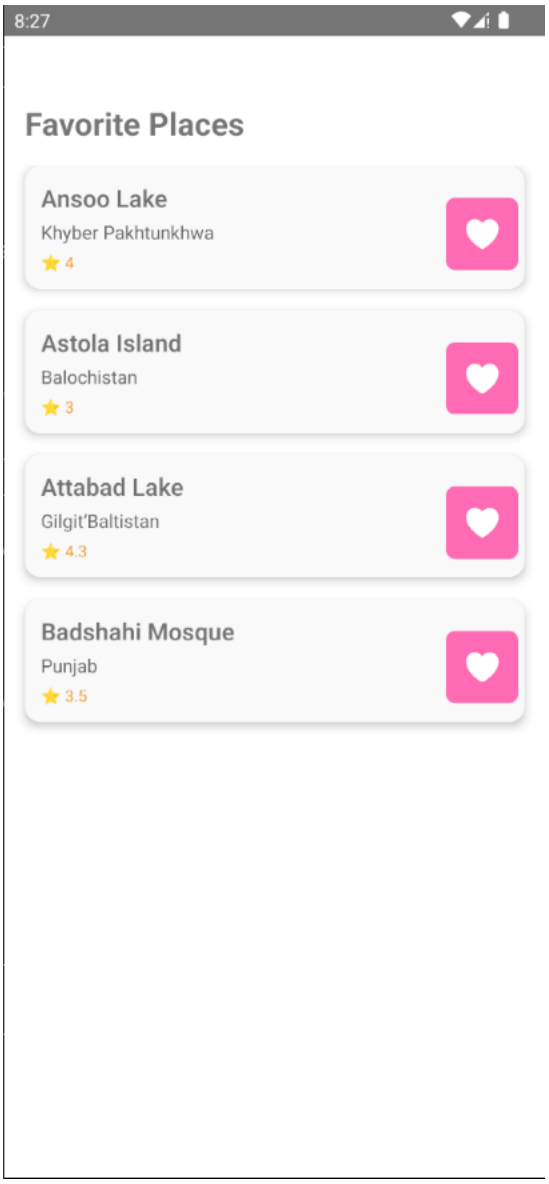
7.4.5 Use Case Name - Mock Screen 5



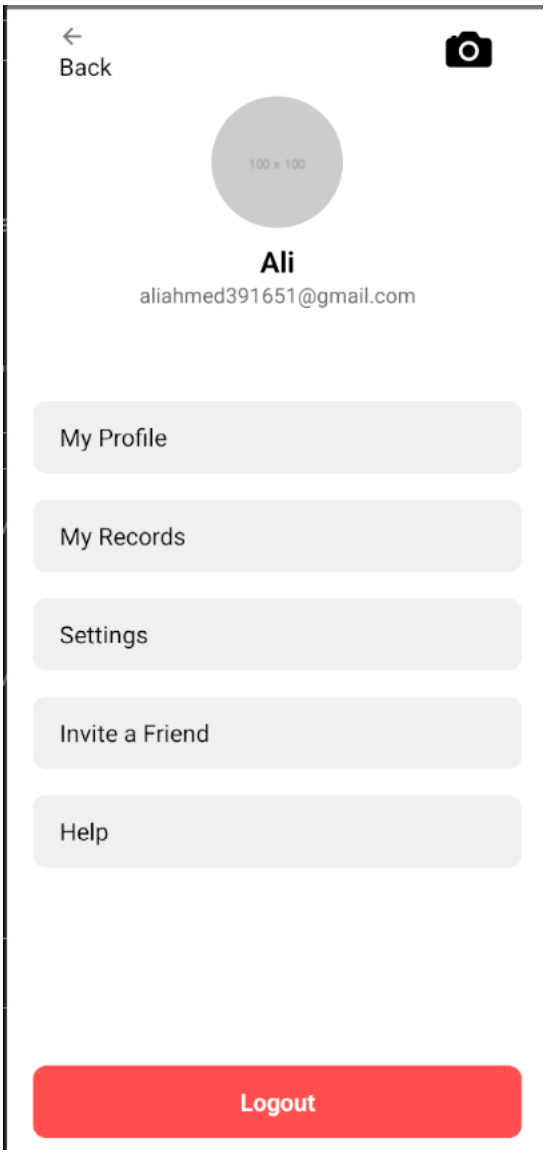
7.4.6 Use Case Name - Mock Screen 6



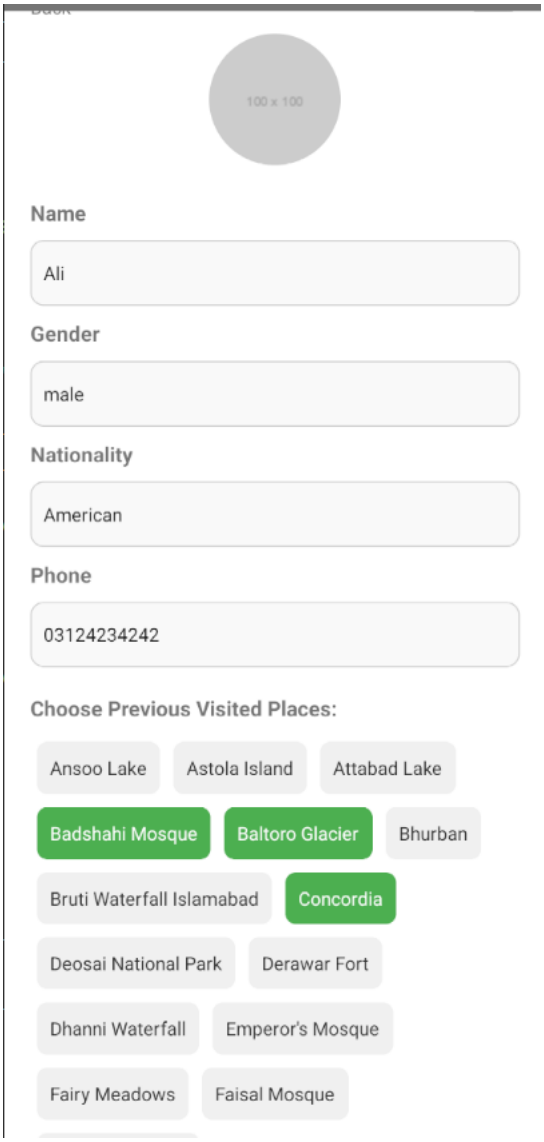
7.4.7 Use Case Name - Mock Screen 7



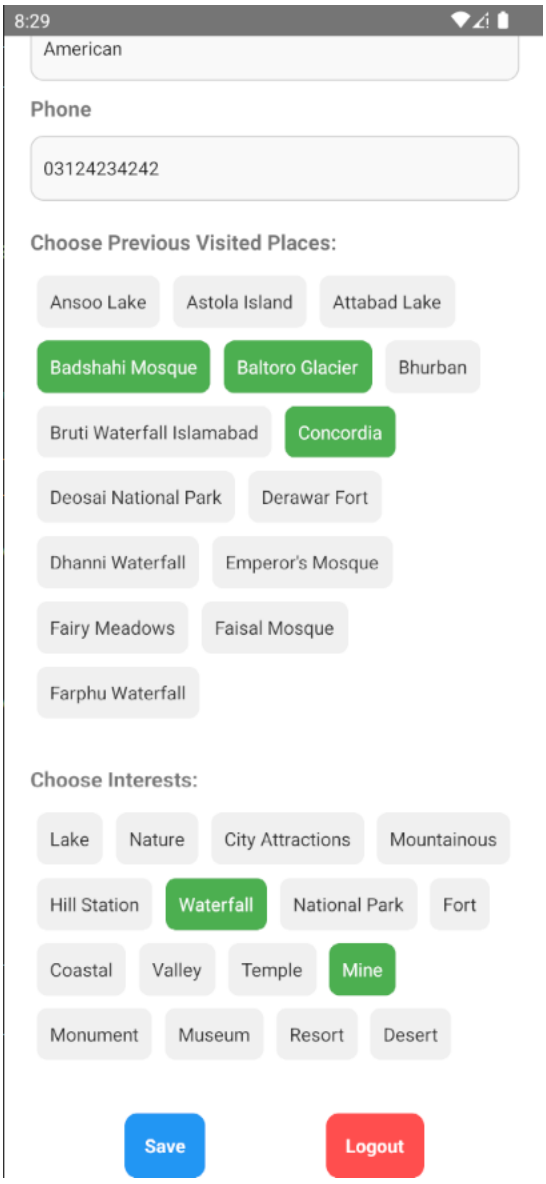
7.4.8 Use Case Name - Mock Screen 8



7.4.9 Use Case Name - Mock Screen 9



7.4.10 Use Case Name - Mock Screen 10



8 References

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

9 Appendices