**Low-Level Architecture and Data Models**

**<P04>:<Manzil>**

**<team member names & ids>**

|  |  |
| --- | --- |
| **Student ID** | **Name** |
| **25100320** | **Muhammad Usman Arshid** |
| **25100313** | **Muhammad Mehdi** |
| **25100235** | **Shahrez Faisal** |
| **24100199** | **Umer inayat** |
| **25100015** | **Omar Ibne Sajjad** |

|  |  |  |
| --- | --- | --- |
| **Content** | **Totals** | **Obtained** |
| Architecture diagram | 20 | 20 |
| Architecture justification | 20 | 20 |
| E/R diagram | 20 | 20 |
| E/R diagram description | 20 | 20 |
| Tools and Technologies | 10 | 10 |
| Who did what | 3 | 3 |
| Review checklist | 2 | 2 |
| Overall formatting/template | 5 | 5 |
| GitHub folder structure penalty | -15 | - |
| Late submission penalty | -20 | - |
| **Grand Total** | **100** | **100** |
| **General Comments/Individual Grading:** Update as the high level architecture. Make MVC, no need for sequence diagrams, your architecture should show your own system. | | |

**Table of Contents**

[1. Introduction 3](#_Toc87859470)

[2. System Architecture 4](#_Toc87859471)

[2.1 Architecture Diagram—As it is in the prototype code 4](#_Toc87859472)

[2.2 Architecture Diagram—As it should-be 4](#_Toc87859473)

[3. Data Models 5](#_Toc87859474)

[4. Tools and Technologies 6](#_Toc87859475)

[5. Who Did What? 7](#_Toc87859476)

[6. Review checklist 7](#_Toc87859477)

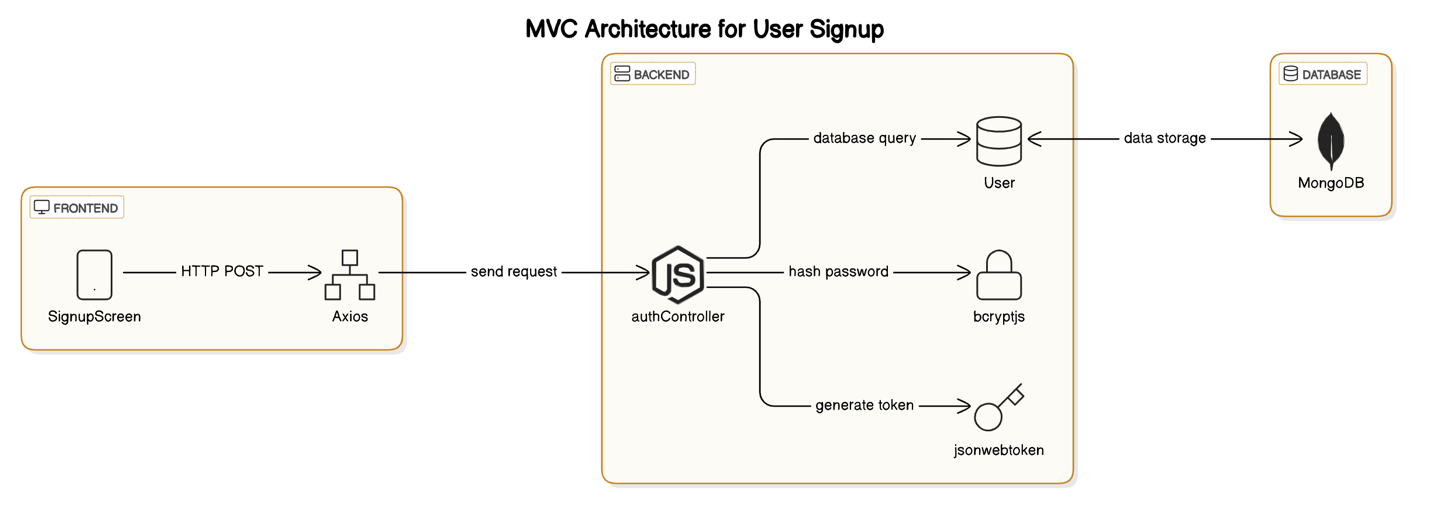
# Introduction

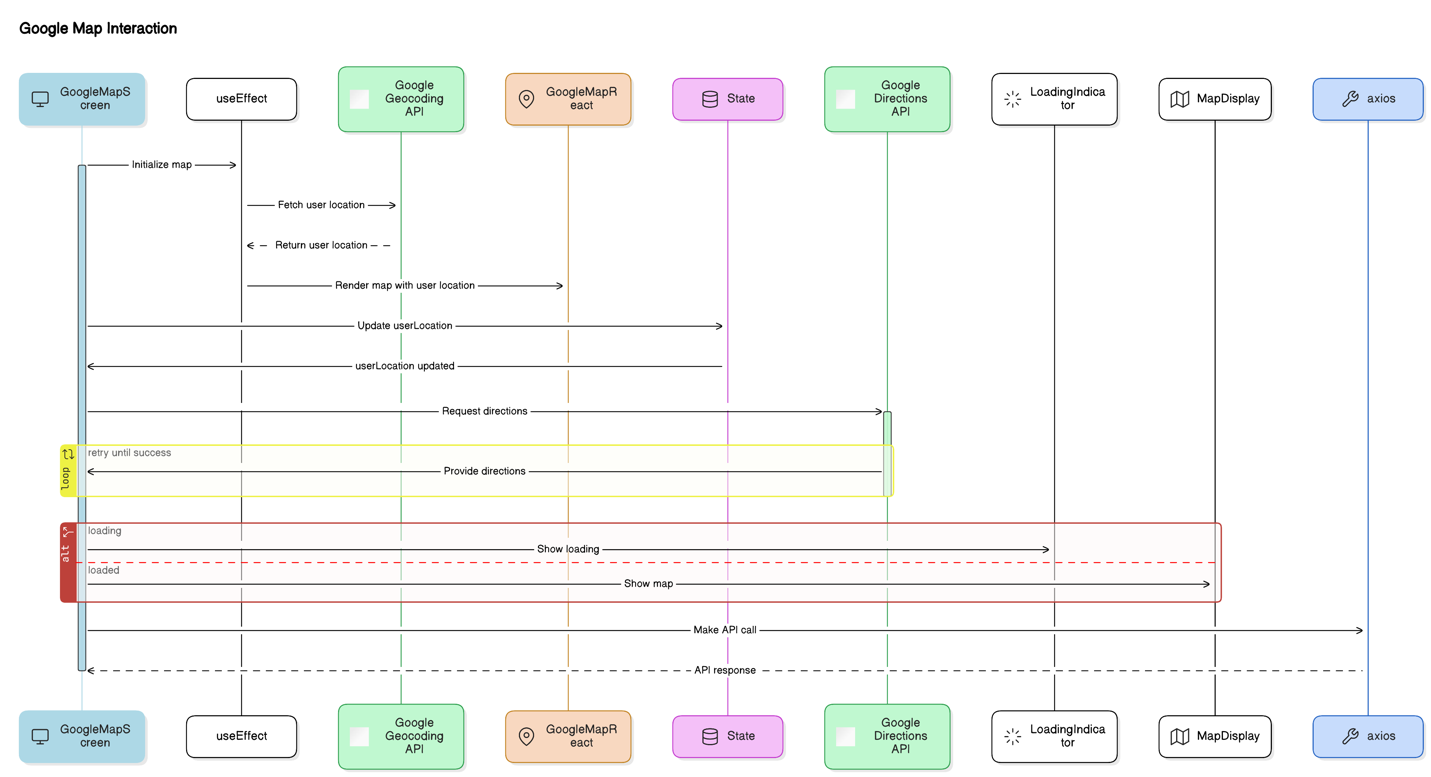
**Manzil** is a mobile application designed to assist users traveling to northern destinations and major cities in Pakistan. It aims to streamline the process by offering a one-stop solution for booking hotels, renting vehicles, checking weather updates, and finding local services and their reviews for better choice of places, such as restaurants, schools, and hospitals.

The app leverages machine learning (ML) to provide personalized recommendations and supports real-time services like live weather updates and itinerary management. The primary users include travelers, hotel management, and app administrators.

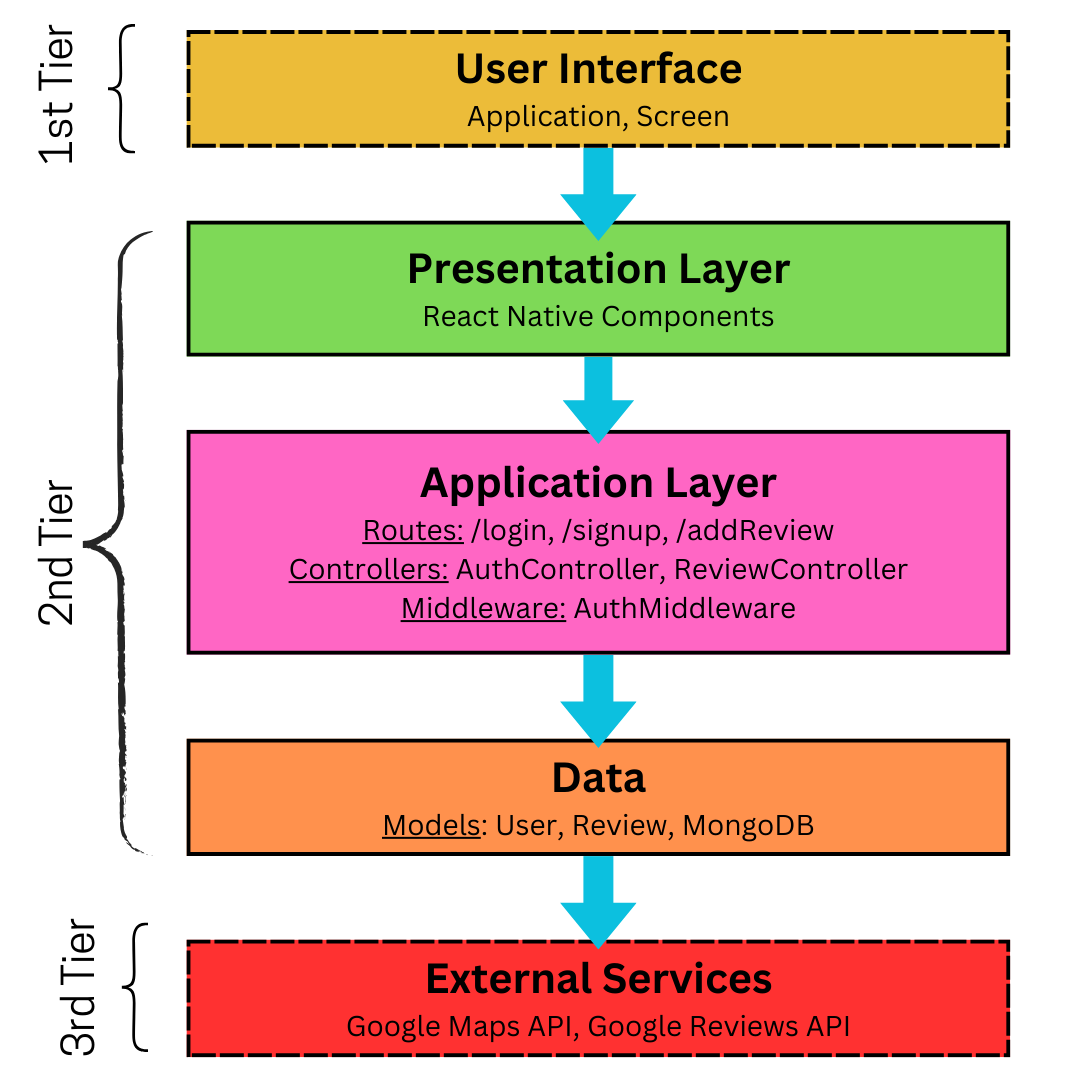
# System Architecture

## Architecture Diagram—As it is in the prototype code





## Architecture Diagram—As it should-be



#### **Frontend (React Native Components)**

1. **CityScreen**: Allows users to view all the famous places.
2. **ReviewScreen**: Displays and allows users to add reviews for destinations.
3. **MapScreen**: Shows the map and route details using Google Maps API.

**Backend (Express.js Components)**

1. **Routes**: Define API endpoints for communication with the frontend.

* /signup: User Signup
* /login: User login.
* /getReviews
* /postReviews
* /GetGoogleNavigation

1. **Controllers**:

Contains business logic.

* 1. **AuthController**: Handles user authentication and authorization.
  2. **ReviewController**: Handles reviews creation and retrieval.

1. **Middleware**:

Intercepts requests for validation and security.

* 1. **AuthMiddleware**: Verifies authentication tokens.

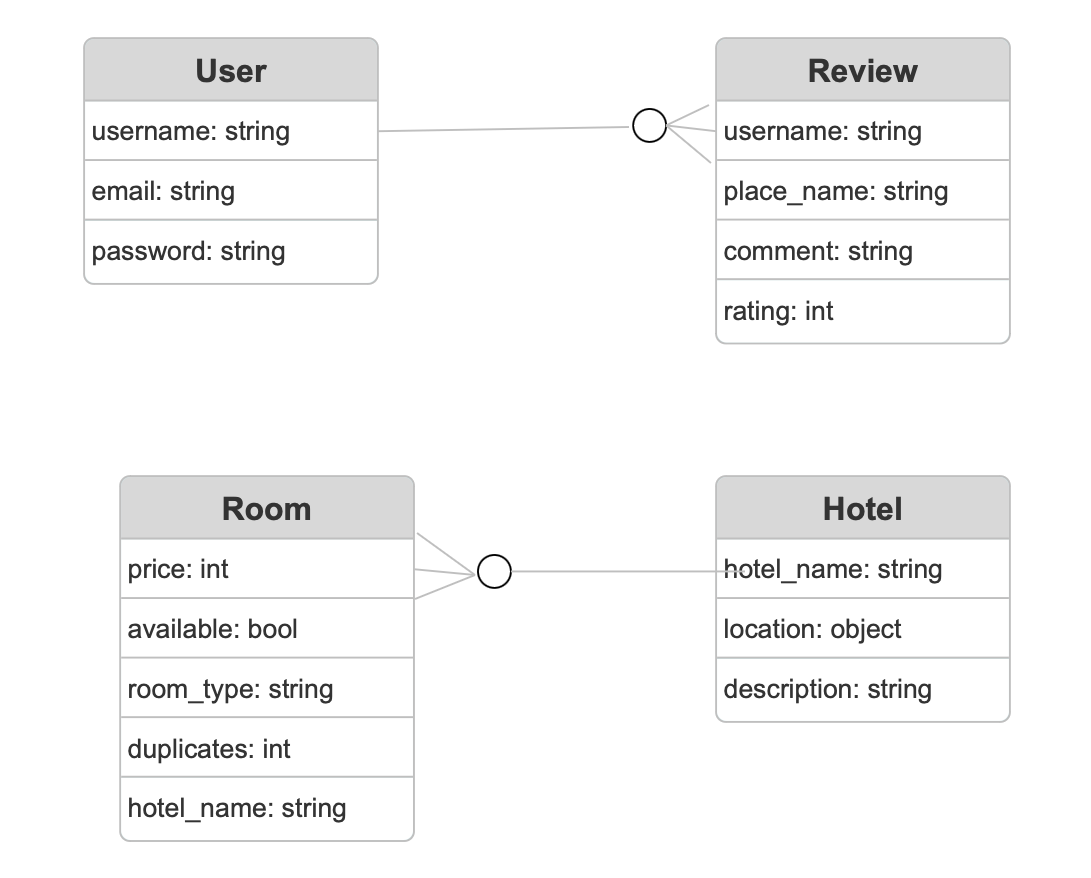
1. **Database Layer**

* **User Model**: Stores user account information (username, email, password).
* **Review Model**: Stores user reviews (username, place name, comment, rating).

**External Services**

1. **Google Maps API**: Powers navigation and location-based services.

# Data Models



User: all users interacting with the application (admins, consumers, hotel staff).

User data fields:

Username: visible name on platform for the user for reviews.

Email: to ensure uniqueness of users and valid accounts.

Password: to ensure account security.

Review: review left by user on destination.

Review data fields:

Username: to store and display the associated user that made the review.

Place name: to store and display which destination the review was left on.

Comment: comment left by user.

Rating: out of 5 ratings assigned by user.

Hotel: hotels available on the platform for booking.

Hotel data fields:

Hotel name: name of the hotel.

Location: location of the hotel. (type object to store coordinates, address etc.)

Description: description of hotel assigned by hotel staff.

Room: room to be booked by user at specified hotel.

Room data fields:

Price: price specified by hotel staff.

Available: bool to check if room is available for requested date.

Room type: type of room: deluxe, executive, presidential suite etc.

Duplicates: number of such rooms available at the hotel.

Hotel name: name of hotel room is associated with.

# Tools and Technologies

* **React Native** (v0.71): For building cross-platform mobile applications.
* **Express** (v4.x): For developing backend REST APIs.
* **Node.js** (v18.x): As the runtime environment for the backend.
* **MongoDB** (v6.x): For database management.
* **Mongoose** (v6.x): For managing MongoDB through object modeling.
* **Firebase:** For real-time database and authentication services.
* **railway:** Hosting back-end services.
* **TensorFlow** (v2.10): For implementing ML models.
* **Google Maps API**: For map services and location-based features.
* **Google Reviews API**: For review services of different places (hotels, restaurants, etc.).
* **OpenWeather API:** For live weather updates.
* **GitHub:** For version control and collaboration.
* **GitHub Boards:** For project management and issue tracking.

**5.** **Who Did What?**

|  |  |
| --- | --- |
| **Name of the Team Member** | **Tasks done** |
| Muhammad Usman Arshid | 1 – Introduction, 2.2 - Architecture Diagram |
| Muhammad Mehdi | 2.1 - Architecture Diagram |
| Shahrez Faisal | 4 - tools and techniques |
| Omar Ibne Sajjad | 2.2 - Architecture Diagram |

# 6. Review checklist

Before submission of this deliverable, the team must perform an internal review. Each team member will review one or more sections of the deliverable.

|  |  |
| --- | --- |
| **Section** **Title** | **Reviewer Name(s)** |
| 4 – Tools & Technologies | Muhammad Usman Arshid |
| 2.1 | Shahrez Faisal |
| 1 | Umer Inayat |
| 2.2 | Omar Ibne Sajjad |
| 3 | Muhammad Mehdi |