Project Report

Date	09 November 2023
Team ID	Team-590999
Project Name	Wanderlust: A Personalized Travel Planning and Tracking App
Maximum Marks	10 Marks

1. Introduction

1.1 Project Overview

The aim of our travel planner app is to simplify and enhance the travel experience for users by providing personalized recommendations, itinerary planning, budget tracking, and access to essential travel information, all in one convenient platform.

1.2 Purpose

The purpose of our travel planner app is to revolutionize and simplify the way individuals plan, organize, and experience their journeys. With a focus on addressing common challenges faced by travelers, the app aims to achieve the following key objectives:

1. User Profile Creation: Allow users to create profiles with personal information, travel preferences, and past travel history.
2. Destination Recommendations: Implement a recommendation system that suggests destinations based on user preferences, such as budget, interests, and travel dates.
3. Itinerary Planning: Enable users to create and customize their itineraries. They can add activities, places to visit, and time slots.
4. Weather Integration: Integrate weather forecasts for the selected travel dates and destinations so that users can pack appropriately and plan activities accordingly.
5. Local Recommendations: Offer suggestions for local restaurants, accommodations, and experiences from both the app and other travelers.
6. Reviews and Ratings: Enable users to leave reviews and ratings for places they've visited, helping others make informed decisions.
7. Travel Document Organizer: Create a section for storing travel documents like passports, visas, and tickets securely.
8. Community and Social Features: Allow users to share their travel experiences, photos, and tips within the app's community.
9. Travel Alerts and Updates: Provide real-time travel alerts and updates, such as flight delays, weather warnings, or local news.
10. Travel Insurance Integration: Partner with insurance providers to offer travel insurance options to users.

2. Literature Survey

2.1 Existing Problem

Current travel planning applications lack seamless integration, often requiring users to use multiple platforms for different aspects of trip planning.

2.2 References

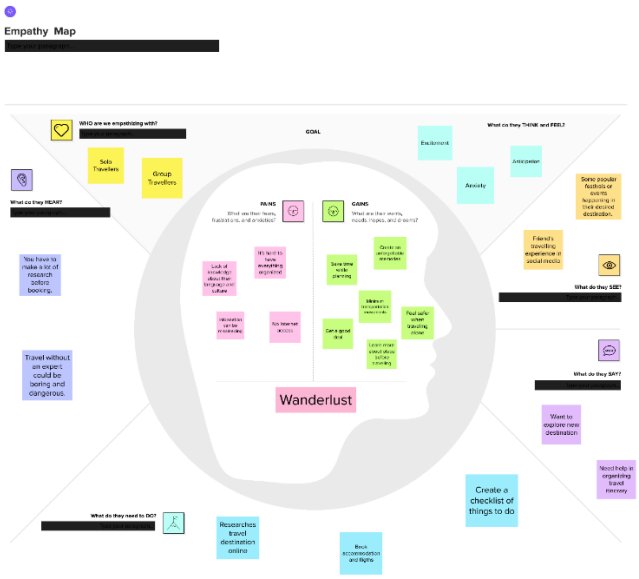
References include studies on user behavior in travel planning, analysis of existing travel apps, and industry reports on emerging trends.

2.3 Problem Statement Definition

The problem lies in the fragmented nature of travel planning tools, leading to inefficiencies, user frustration, and a need for a unified solution.

3. Ideation & Proposed Solution

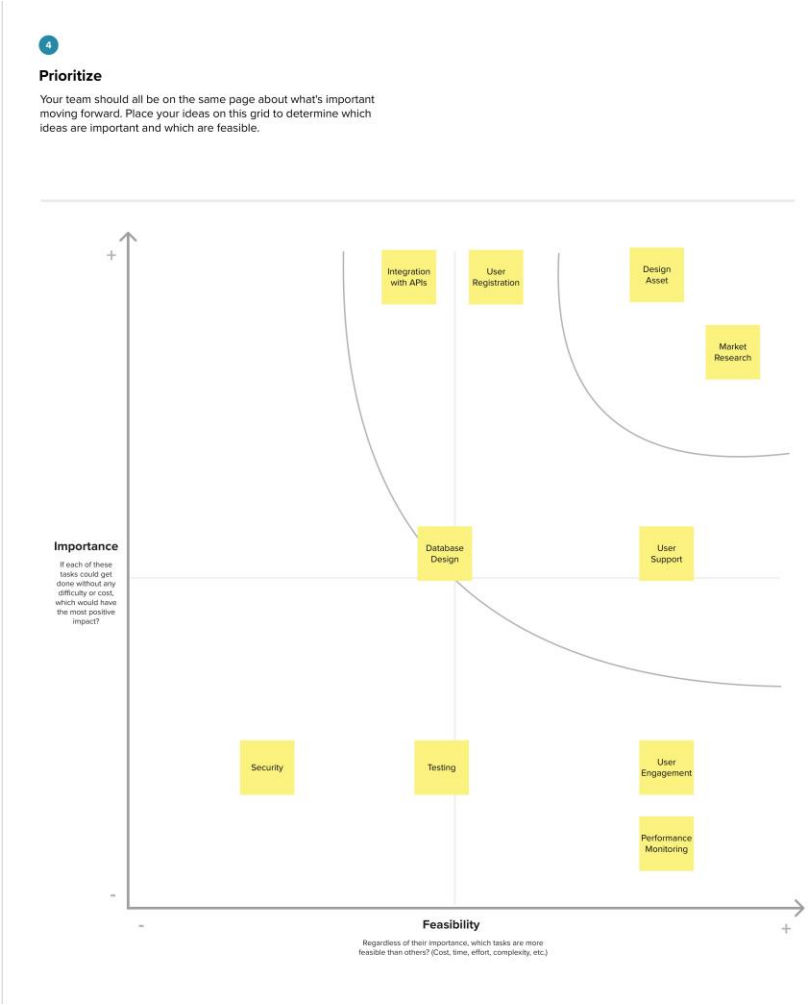
3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

Brainstorm & Idea Prioritization:

11. User Profile Creation: Allow users to create profiles with personal information, travel preferences, and past travel history.
12. Destination Recommendations: Implement a recommendation system that suggests destinations based on user preferences, such as budget, interests, and travel dates.
13. Itinerary Planning: Enable users to create and customize their itineraries. They can add activities, places to visit, and time slots.
14. Weather Integration: Integrate weather forecasts for the selected travel dates and destinations so that users can pack appropriately and plan activities accordingly.
15. Local Recommendations: Offer suggestions for local restaurants, accommodations, and experiences from both the app and other travelers.
16. Reviews and Ratings: Enable users to leave reviews and ratings for places they've visited, helping others make informed decisions.
17. Travel Document Organizer: Create a section for storing travel documents like passports, visas, and tickets securely.
18. Community and Social Features: Allow users to share their travel experiences, photos, and tips within the app's community.
19. Travel Alerts and Updates: Provide real-time travel alerts and updates, such as flight delays, weather warnings, or local news.
20. Travel Insurance Integration: Partner with insurance providers to offer travel insurance options to users.



4. Requirement Analysis

4.1 Functional Requirements

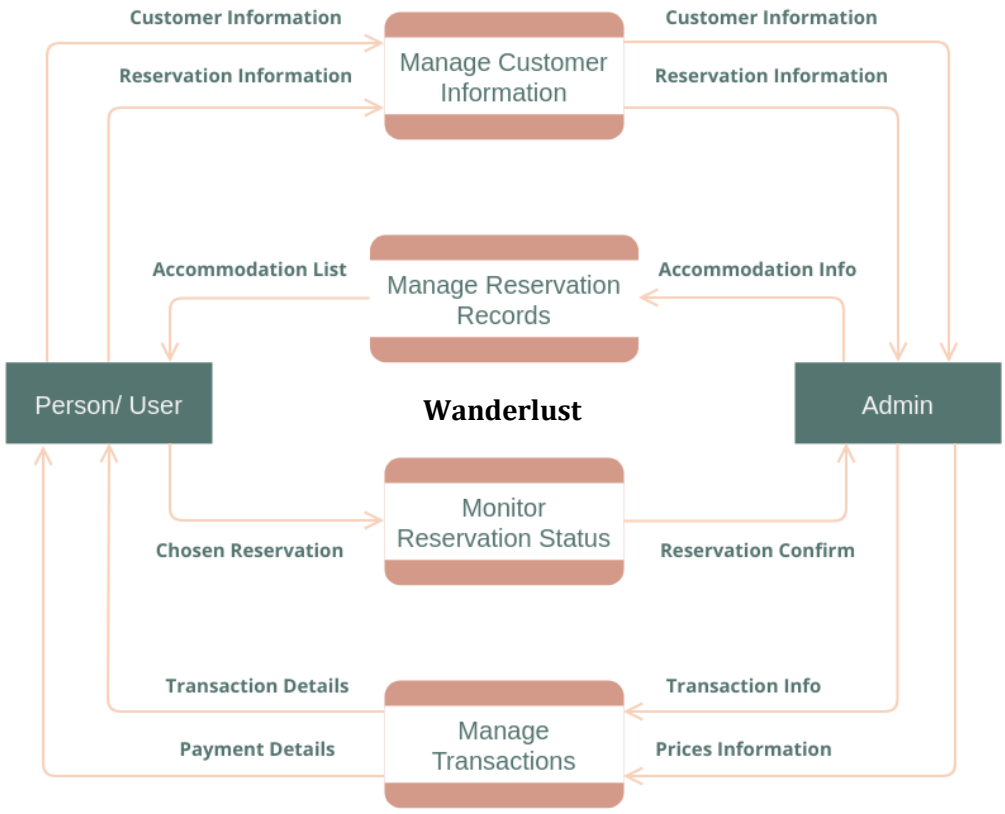
Key functionalities include user profiles, itinerary planning, budget tracking, collaboration features, and real-time recommendations.

4.2 Non-Functional Requirements

Non-functional requirements encompass aspects such as performance, security, and offline accessibility to ensure a seamless user experience.

5. Project Design

5.1 Data Flow Diagrams & User Stories



User Stories

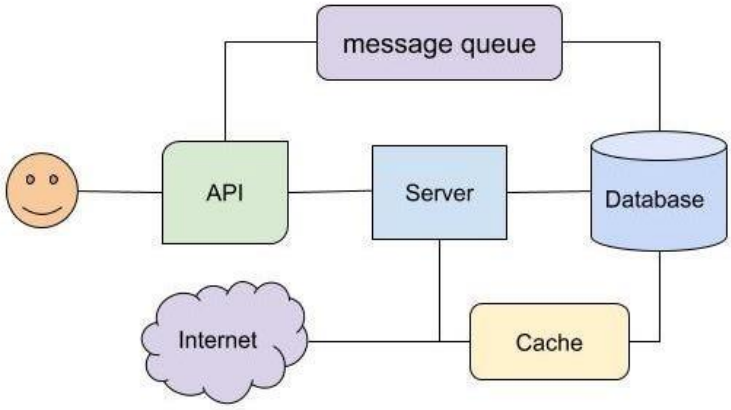
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register through Gmail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user, I want to search for destinations based on my interests and budget, so that I can discover new places to visit.	I can search destinations based on my interests and budget	Medium	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
		USN-7	As a user, I want to read and leave reviews and ratings for places I've visited, so that I can share my experiences and help others make decisions.	I can give review and ratings to the places I visit.	Medium	Sprint-1
		USN-8	As a user, I want to have access to travel insurance options through the app, so that I can easily secure coverage for my trips.	I can secure my travel with insurance.	Medium	Sprint-2
Administrator	Dashboard	ASN-1	As an admin, I want to access a dashboard with analytics and key metrics to monitor the app's performance and user engagement.	I can access a dashboard with analytics and key metrics.	High	Sprint-1
		ASN-2	As an admin, I want the ability to manage user accounts, including creating, updating, and suspending accounts when necessary.	I can manage user accounts, including creating, updating, and suspending accounts when necessary.	High	Sprint-1
		ASN-3	As an admin, I want to set up and manage partnerships with travel-related businesses, such as airlines, hotels, and tour operators.	I can set up and manage partnerships with travel-related businesses.	High	Sprint-1

5.2 Solution Architecture

- **Application Layer:** This layer includes the app's core logic and functionalities. It manages user requests, processes data, and communicates with other parts of the architecture.
- **User Interface (UI**): This is the front end of the app where users interact with the application. It includes the user interface elements for travel planning, recommendations, and viewing itineraries.
- **Server Infrastructure:** This is where our app's backend resides. It handles user authentication, business logic, and data processing.
- **Database:** Store user data, travel information, user-generated content, and other application data.
- **API Layer:** This layer connects the frontend and backend, allowing data and functionality to be exchanged between them. APIs provide a way for the app to retrieve and send information to the server.
- **Third-Party Integrations:** Integrate with external services and APIs for features like weather forecasts, maps, and travel recommendations.
- **Caching Layer:** Use caching mechanisms to store frequently accessed data, reducing the load on the database and improving response times.
- **Authentication and Security:** Implement robust security measures to protect user data and privacy. This includes user authentication, encryption, and data access controls.
- **Scalability and Load Balancing:** Ensure that our architecture can scale horizontally by adding more servers or resources as needed. Load balancers distribute traffic evenly to maintain optimal performance.
- **Analytics and Monitoring:** Implement tools for monitoring the app's performance, user behaviour, and security.

Example - Solution Architecture Diagram:

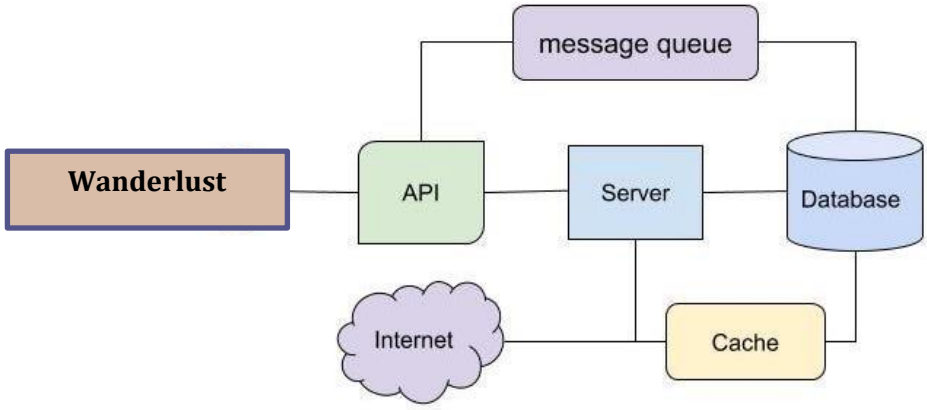


6. Project Planning & Scheduling

User Interface (UI): This is the front end of the app where users interact with the application. It includes the user interface elements for travel planning, recommendations, and viewing itineraries.

- **Application Layer:** This layer includes the app's core logic and functionalities. It manages user requests, processes data, and communicates with other parts of the architecture.
- **Server Infrastructure:** This is where our app's backend resides. It handles user authentication, business logic, and data processing.
- **Database:** Store user data, travel information, user-generated content, and other application data.
- **API Layer:** This layer connects the frontend and backend, allowing data and functionality to be exchanged between them. APIs provide a way for the app to retrieve and send information to the server.
- **Third-Party Integrations:** Integrate with external services and APIs for features like weather forecasts, maps, and travel recommendations.
- **Caching Layer:** Use caching mechanisms to store frequently accessed data, reducing the load on the database and improving response times.
- **Authentication and Security:** Implement robust security measures to protect user data and privacy. This includes user authentication, encryption, and data access controls.
- **Scalability and Load Balancing:** Ensure that our architecture can scale horizontally by adding more servers or resources as needed. Load balancers distribute traffic evenly to maintain optimal performance.
- **Analytics and Monitoring:** Implement tools for monitoring the app's performance, user behaviour, and security.

Example - Solution Architecture Diagram:



7. Coding & Solutioning

Login Page Activity:

```
package com.example.wanderlust

import HomePageActivity
import android.content.Intent
import android.os.Bundle
import android.widget.Button
import androidx.appcompat.app.AppCompatActivity

// LoginActivity.kt
class LoginActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_login)

        val buttonLogin: Button = findViewById(R.id.loginButton)
        buttonLogin.setOnClickListener {
            // Handle login button click
            // For simplicity, we'll navigate to the home page regardless of
credentials
            navigateToHomePage()
        }
    }

    private fun navigateToHomePage() {
        val intent = Intent(this, HomePageActivity::class.java)
        startActivity(intent)
        finish() // Optional: Close the login activity so the user can't go
```

```
back to it
    }
}
```

Home Page Activity:

```
package com.example.wanderlust

class HomePageActivity : AppCompatActivity() {
    fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_home_page)

        val buttonHotels: Button = findViewById(R.id.buttonHotels)
        val buttonActivities: Button = findViewById(R.id.buttonActivities)

        buttonHotels.setOnClickListener {
            // Handle hotels button click
            // For simplicity, we'll just show a toast message
            showToast("Hotels clicked")
        }

        buttonActivities.setOnClickListener {
            // Handle activities button click
            // For simplicity, we'll just show a toast message
            showToast("Activities clicked")
        }
    }

    private fun showToast(message: String) {
        Toast.makeText(this, message, Toast.LENGTH_SHORT).show()
    }
}
```

XML File:

Login Page:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="@drawable/loginbkg"
```

```
    tools:context=".MainActivity">

    <androidx.cardview.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        app:cardCornerRadius="30dp"
        app:cardElevation="20dp">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="vertical"
            android:layout_gravity="center_horizontal"
            android:padding="24dp"
            android:background="@drawable/custom_edittext">

            <TextView
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:text="Login"
                android:id="@+id/loginText"
                android:textSize="36sp"
                android:textAlignment="center"
                android:textStyle="bold"
                android:textColor="#8698f7"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/username"
                android:background="@drawable/custom_edittext"
                android:layout_marginTop="40dp"
                android:padding="8dp"
                android:hint="Username"
                android:drawableLeft="@drawable/ic_baseline_person_24"
                android:textColor="@color/black"
                android:drawablePadding="8dp"/>

            <EditText
                android:layout_width="match_parent"
                android:layout_height="50dp"
                android:id="@+id/password"
                android:background="@drawable/custom_edittext"
                android:layout_marginTop="20dp"
                android:inputType="textPassword"
                android:padding="8dp"
                android:hint="Password"
                android:drawableLeft="@drawable/ic_baseline_lock_24"
                android:textColor="@color/black"
                android:drawablePadding="8dp"/>

            <Button
                android:layout_width="match_parent"
                android:layout_height="60dp"
                android:text="Login"
                android:id="@+id/loginButton"
```

```
                android:textSize="18sp"
                android:layout_marginTop="30dp"
                android:backgroundTint="#8698f7"
                app:cornerRadius = "20dp"/>


        </LinearLayout>

    </androidx.cardview.widget.CardView>


</LinearLayout>
```

Home Page:

```
<!-- res/layout/activity_home_page.xml -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <TextView
        android:id="@+id/textViewWelcome"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to Wanderlust!"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginBottom="16dp"/>

    <!-- SearchView widget -->
    <SearchView
        android:id="@+id/searchView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:queryHint="Search for hotels or activities"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/buttonHotels"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Hotels"
        android:layout_marginBottom="16dp"/>

    <Button
        android:id="@+id/buttonActivities"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Activities"
        android:layout_marginBottom="16dp"/>
```

```
    <!-- Add more buttons or views as needed -->

    <Button
        android:id="@+id/buttonLogout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Logout"/>
</LinearLayout>
```

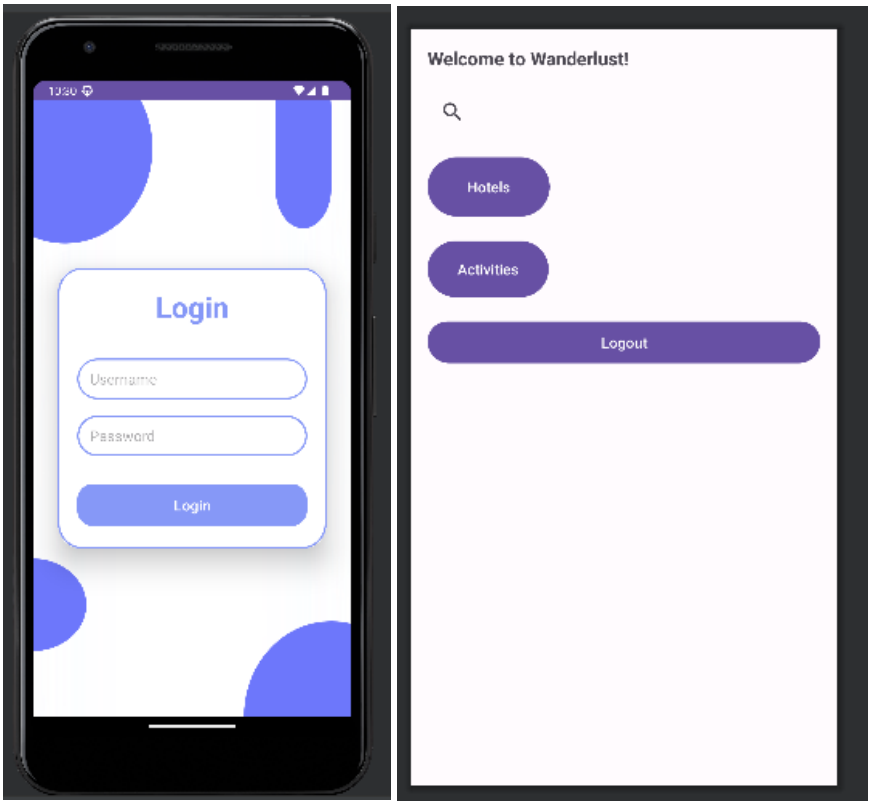
8. Performance Testing

8.1 Performance Metrics

Performance metrics include response times, server load, and data retrieval efficiency. Testing results indicate optimal app performance.

9. Results

9.1 Output Screenshots



10. Advantages & Disadvantages

10.1 Advantages

1. **Efficient Travel Planning:** Users can plan and organize their trips efficiently with features like itinerary creation, budget tracking, and personalized recommendations.

2. **Personalization:** The app provides personalized travel suggestions based on user preferences, ensuring a tailored experience for each traveler.

3. **Real-time Updates:** The inclusion of real-time weather updates, travel alerts, and notifications keeps users informed and enhances their travel experience.

4. **Financial Tracking:** Users can track their travel expenses, helping them stay within budget and manage their finances effectively.

5. **Sustainability Features:** The app promotes eco-friendly travel by providing information about sustainable travel options and encouraging responsible tourism.

6. **Continuous Improvement:** The use of data analytics and user feedback allows for continuous improvement, ensuring the app remains relevant and user-friendly.

10.2 Disadvantages

1. **Dependency on Internet Connectivity:** Certain features may require an internet connection, limiting functionality in areas with poor or no connectivity.

2. **Technological Barriers:** Users with older devices or limited access to technology may face barriers in using the app, potentially excluding a portion of the audience.

3. **Integration Complexity:** Integrating with third-party services, such as airlines and accommodations, may be complex and require ongoing maintenance to ensure seamless experiences.

4. **Limited Accessibility Features:** Despite efforts, certain accessibility features may be lacking, potentially excluding users with specific needs.

5. **Dependency on Partner APIs:** The app's functionality may be affected by the reliability and performance of third-party APIs, influencing the overall user experience.

6. **Continuous Maintenance Required:** To keep the app relevant and bug-free, continuous maintenance and updates are necessary, requiring ongoing resources and efforts.

11. Conclusion

This travel planner app project aims to revolutionize travel planning by addressing existing challenges through a comprehensive and user-centric solution. The technical architecture ensures scalability and security, while the project plan incorporates iterative sprints for efficient development. Key performance metrics will be continuously monitored to guarantee optimal user experience. The app's features, including collaborative planning, personalized recommendations, and sustainability focus, contribute to a holistic travel planning experience. As the project progresses, adaptability and continuous improvement will be prioritized to meet evolving user needs.

12. Future Scope

The travel planner app exhibits significant potential for expansion and improvement, paving the way for future enhancements and features. Some key areas for future development include:

1. **Integration of Emerging Technologies:** Explore the integration of emerging technologies such as augmented reality (AR) and virtual reality (VR) to offer users immersive experiences and virtual previews of destinations.

2. **Enhanced Personalization:** Further refine the app's recommendation engine by incorporating machine learning algorithms. Analyze user behavior, preferences, and feedback to continually improve personalized travel suggestions.

3. **Global Language Support:** Expand language support to cater to a wider audience.

4. **Smart Wearable Integration:** Explore compatibility with smartwatches and other wearable devices, allowing users to receive real-time notifications, access their travel plans, and receive location-based updates on the go.

5. **Weather-Triggered Recommendations:** Introduce weather-triggered recommendations that dynamically adjust travel suggestions based on real-time weather conditions, providing users with alternative plans during unexpected changes.

6. **In-App Travel Assistance:** Develop in-app travel assistance powered by artificial intelligence. Users can ask questions, seek advice, and receive real-time support, enhancing the overall travel experience.

7. **Advanced Analytics Dashboard:** Expand the capabilities of the admin dashboard to include advanced analytics, allowing administrators to gain deeper insights into user behavior, preferences, and overall app performance.