WanderPlan

By:

Abdullah Ahtsham

45679

Muhammad Nabeel Awais

43944

Muhammad Bilal Butt

44712

Supervised by:

Shehzad Ahmad Khan

Faculty of Computing

Riphah International University, Islamabad

Fall 2025

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfilment of the Requirement for the Award of the Degree of

Bachelor of Science in Software Engineering

Faculty of Computing

Riphah International University, Islamabad

Date: 05 January, 2026

Final Approval

This is to certify that we have read the report submitted by ***Abdullah Ahtsham (45679)***, ***Muhammad Nabeel Awais (43944)*** and ***Muhammad Bilal Butt (44712)*** for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Software Engineering (BSSE). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Software Engineering (BSSE).

Committee:

1

_____

Shehzad Ahmad Khan

Supervisor

2

_____

Dr. Musharraf Ahmad

Head of Department/chairman

Declaration

We hereby declare that this document "**WanderPlan**" neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers especially our supervisor **Shehzad Ahmad Khan**. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

_____

Abdullah Ahtsham

45679

_____

Muhammad Nabeel Awais

43944

_____

Muhammad Bilal Butt

44712

Dedication

This work is dedicated first and foremost to our Creator,
the source of all knowledge, strength, and blessings—
the One who commands “Be,” and it is.
(Qur’an, 36:82).

We also want to express our sincere gratitude to our parents, whose sacrifices and unwavering support have been the cornerstone of our success.
We sincerely dedicate this milestone to our supervisor, Shehzad Ahmad Khan, and faculty members who supported, encouraged, and believed in us.

Acknowledgement

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We extend our deepest gratitude to our respected supervisor, Sir Shahzad Ahmad Khan, for his continuous guidance, valuable insights, and motivation throughout this journey. His mentorship played a vital role in shaping thisproject into its final form.

_____

Abdullah Ahtsham

45679

_____

Muhammad Nabeel Awais

43944

_____

Muhammad Bilal Butt

44712

Abstract

WanderPlan is a mobile companion app that combines real-time navigation, local recommendations, and budget planning into one cohesive platform. The aim of this project is to help tourists manage their travel expenses efficiently while discovering authentic local food, attractions, and experiences — all within their defined budget.

WanderPlan's intelligent recommendation and budget management system provides an all-in-one solution, in contrast to current fragmented solutions that force Tourists to rely on multiple applications for food discovery, lodging, map navigation as well as checking reviews. Using user-generated content (UGC) and location-based data, the system suggests the best and most genuine options while dynamically allocating a visitor's budget among lodging, travel, and dining. Additionally, the application features a community-driven review system that encourages participation and genuineness by allowing contributors to share their experiences and receive rewards. By enabling users to explore local restaurants, attractions, and bargains with clear cost and distance estimation, real-time mapping and navigation features improve usability.

WanderPlan's ultimate goal is to revolutionise traveler planning and experience by providing a comprehensive, effective, and user-focused travel management solution. The project uses technology for convenience, personalisation, and engagement while promoting accessible and culturally immersive travel.

Table Of Contents

List Of Figures

List Of Tables

Chapter: 1

INTRODUCTION

Chapter 1

INTRODUCTION

WanderPlan is an intelligent, user-centric travel assistance platform designed to make travel easier for tourists, particularly when visiting new places. The system assists users in making secure travel choices that fit their taste and budget by integrating a contributor-driven review ecosystem, location based recommendations, an interactive map interface as well as a dynamic budget engine. WanderPlan offers a dedicated web-based business dashboard in addition to it's mobile application for the tourists and contributors. This dashboard allows local attractions, hotels, and restaurants to post deals, manage listings, create profiles, and track customer engagement. In order to synchronise menus, pricing, availability, and promotional data for more precise app recommendations, businesses with existing point-of-sale systems can choose to integrate them. WanderPlan makes travel easier, boosts local visibility, and fosters a more informed, pleasurable, and customised travel experience by bringing together travellers, local contributors, and businesses under a single digital ecosystem.

1.1. Opportunity and Stakeholders

The global travel and tourism industry continues to grow rapidly, driven by advancements in digital technologies and increased accessibility of information. The modern traveler seeks convenience, personalization, and efficiency. Travellers anticipate smart, integrated tools that simplify their trip, whether it is - selecting the ideal location, controlling travel costs, or identifying the greatest dining establishments. Despite this need, the majority of applications in the travel ecosystem are still dispersed; some concentrate only on budgeting, while others concentrate on reviews and navigation. Users are forced to juggle multiple applications during a single trip due to this lack of integration, which results in inefficiencies.

By providing a unified mobile platform that combines trip budgeting, food and lodging suggestions, maps, reviews, and gamified user interaction, our project, ***WanderPlan***, fills this gap. By introducing a data-driven and AI-powered approach to trip management, it lessens users' cognitive load and makes smarter travel planning possible.

1.1.1. Stakeholders

A wide range of stakeholders are involved in WanderPlan:

- Tourist.
- Business Owners.
- Admin.

1.2. Motivations and Challenges

1.2.1. Motivation

The growing demand for more intelligent, individualised, and easily accessible travel planning tools led to the creation of WanderPlan. Travellers still have to deal with disorganised information, shaky reviews, inconsistent prices, and an overwhelming range of options in a world where tourism is growing and digital platforms are changing how people discover new places. Our goal is to make travel planning easier by developing a centralised platform that uses real business data and community contributions to intelligently create trip plans, manage budgets, and offer reliable recommendations.

1.2.1.1. Problem Relevance

Based on our interviews with Mr. Ahmad and several other people who travel primarily on their own, tourism is currently in the spotlight. Due to unexpected expenditures and a lack of local knowledge, we discovered that managing a budget is the most challenging aspect of travelling. This is a waste of money and results in a poor tour experience.

1.2.1.2. Team Motivation

As software engineering students, we were motivated by the challenge of building a real-world system that merges multiple technical layers - budget algorithms,

recommendation engines, dynamic mapping, user-generated content, and a business-facing dashboard.

We can put the principles of system design, mobile development, API integrations, security, and user experience design into practice with this project. We chose this domain because it is both technically diverse and socially impactful, offering significant learning in problem-solving and innovation.

1.2.1.3. Community Impact

WanderPlan could have a beneficial impact on numerous aspects of society. Better planning, cost transparency, and real peer-driven insights are all beneficial to tourists.

Local contributors gain a platform to share their experiences and earn rewards, fostering a sense of community ownership. Owners of businesses, particularly independent and small ones, benefit from increased visibility, improved outreach, and analytics that help them comprehend consumer behaviour.

Long-term, such a platform can help local tourism economies, advertise lesser-known destinations, and facilitate data-driven enhancements to travel and hospitality offerings.

1.2.2. Challenges

We had to overcome a number of operational, technical, and team-building obstacles in order to develop WanderPlan. These challenges put our ability to solve problems to the test and influenced how we approached the project.

1.2.2.1. Technical Complexity

WanderPlan combines a number of complex components, including business dashboards, backend APIs, mapping services, recommendation logic, mobile development, and user-generated content workflows. During early development, it was difficult to handle real-time data, ensure seamless communication between these layers, and preserve performance on low-end devices.

1.2.2.2. Resource Limitations

There was limited access to reliable internet, high-performance systems, and paid APIs (like map services or Google Distance Matrix). This slowed down integration cycles and limited how often build pipelines could be tested. Maintaining usage limits while working with free tiers also required careful optimisation.

1.2.2.3. Time Management

One of the biggest challenges was juggling project deadlines, semester projects, coursework and job. Consistent planning and prioritisation were necessary to prevent delays in coordinating development timelines, particularly between frontend, backend, and documentation processes.

1.2.2.4. Learning and Skill Gaps

A number of new technologies were introduced by the project, including data scraping, mapping services, backend API integration, mobile UI/UX design, and business dashboard design. A steep learning curve was introduced by comprehending these technologies as well as best practices for database structuring, clean architecture, and authentication.

1.2.2.5. Lessons Learned

We came to understand that successful completion of a large-scale project requires clear module ownership, early planning, and effective communication. In software development, technical difficulties are inevitable, but progress is fuelled by constant coordination and a readiness to modify the plan.

1.2.2.6. Future Recommendations

Future teams should plan frequent check-ins, adhere to stringent version control, and complete the core architecture as soon as possible. Setting reasonable goals and conducting early prototype testing can greatly lower stress levels and avoid rework in later phases.

1.2.3. Conclusion

WanderPlan was created in response to the increasing demand for a single, dependable, and easy-to-use platform that assists tourists in organising their travels, controlling their spending, and finding genuine experiences. However, the logistical, technical, and teamwork challenges we encountered changed our strategy and improved our problem-solving skills. These insights clarified things, enhanced our

performance, and equipped us to create a system that is useful, scalable, and beneficial to all users.

1.3. Goals and Objectives

Our objective is to offer a solution that will significantly help those who are worried about the financial difficulties that arise when travelling throughout Pakistan.

1.3.1. Introduction

WanderPlan aims to address the common difficulties faced by modern travelers who struggle to manage budgets, discover trustworthy recommendations, and plan trips efficiently. Without incorporating real-time prices, user-generated insights, or local recommendations, the majority of current platforms either concentrate on high-level travel inspiration or offer limited budget management. Similar to this, local businesses and contributors do not have a single online presence where they can advertise services, share updates, and interact meaningfully with visitors.

1.3.2. Overall Goal

WanderPlan's main objective is to develop an intelligent, cost-effective travel companion that, through a single mobile application, assists travellers in creating the best possible travel plans, finding suggested destinations, navigating quickly, and accessing real, community-driven insights.
Furthermore, the system offers:

- A single user platform that brings together the functionalities for travelers to plan their trips, control their expenses, find new places, and post genuine reviews or contributions in exchange for rewards and profile levels.
- A cloud-based business dashboard for restaurants, hotels, and sightseeing spot that helps them in managing verified profiles, announcing deals, replying to reviews.
- An admin control panel that checks business registrations, tracks platform activity, moderates content, and guarantees data quality, security, and smooth system operations.

1.3.3. Rationale

WanderPlan's justification is based on multiple factors:

- Unpredictable local prices, erroneous internet advice, and dispersed information frequently cause tourists to overspend or experience inconvenience.
- Despite having the most genuine knowledge of locations, cuisine, activities, and hidden treasures, local contributors are not as well-known.
- Without investing in extensive marketing channels, businesses find it difficult to engage tourists, advertise deals, and maintain an online presence.

WanderPlan fills these gaps by creating a system that distributes budgets intelligently, gathers reliable local data, and enables companies to communicate with Tourists directly. This guarantees a travel experience that is more knowledgeable, open, and community-driven.

1.3.4. Timeline

In the Part I of our FYP, we are determined to complete the documentation of the first 5 chapters of our report and make a functional mobile application that has the base functionality implemented necessary for the further development of our application.

All the frontend of the application as well as of the admin panel and the Business Panel would be completed as well as 2 of the modules such as Authentication and Reviews would be completed in Part I.

By the grace of Allah (S.W.T.), we hope to finish the remaining documentation and incorporate all of the key features that make up the entire WanderPlan experience in the second phase. This comprises the dynamic budget engine, the recommendation system, the reviews and user-generated content module, the maps and navigation features, the profile and rewards system. To guarantee system dependability and seamless operation, we will also conduct thorough testing, including unit, integration, and end-to-end. Our goals for this phase are to successfully launch the mobile

application's beta version and backend services, complete all documentation, and get the system ready for the last demonstration and assessment.

1.3.5. Success Criteria

Stakeholders will establish the success criteria if the entire solution is developed weeks prior to our part two presentation. Otherwise, by concentrating on the issues that have been identified and evaluating the effectiveness of the implemented solution, we can assess our own success criteria.

1.3.6. Conclusion

As a result, WanderPlan's main goal is to make travel planning and experience easier, better, and more modern by offering a centralised mobile platform backed by community-driven insights and intelligent automation.

1.4. Solution Overview

When considering the difficulties travellers encounter in finding genuine experiences, controlling spending, and locating reliable local information, there are a number of crucial areas where enhancements can greatly improve their travel experience. WanderPlan uses the following focused solutions to close these gaps:

1.4.1. Personalized Trip Budgeting

Travel budgeting frequently becomes too much to handle, particularly when costs differ for lodging, food, and transportation. By providing a dynamic budgeting system that allocates spending intelligently and transparently, WanderPlan streamlines this process and enables travellers to plan within their constraints without sacrificing their experience.

1.4.2. Intelligent Recommendations and Local Discovery

Tourists frequently rely on scattered sources for food, attractions, and activities. WanderPlan brings these elements together by providing tailored recommendations

based on real-time trends, Tourist budgets, and authentic local insights. This helps travelers explore both popular spots and hidden gems with confidence and ease.

1.4.3. Integrated Maps and Real-Time Cost Estimation

Without precise cost and route information, navigating a new city can be challenging. With WanderPlan, travellers can quickly make well-informed decisions by combining map navigation with travel-cost and distance estimation. This reduces unforeseen costs and guarantees more seamless movement on the ground.

1.4.4. Community-Driven Reviews

Because of inaccurate or out-of-date information, it can be challenging to find trustworthy reviews. In order to address this, WanderPlan developed an ecosystem driven by contributors in which residents and visitors exchange current reviews, images, and experiences. The interactive feed guarantees that Tourists receive authentic, context-rich insights and increases transparency.

1.4.5. Business Engagement and Deals Visibility

Travellers may lose out on promotions, discounts, or relevant products just because companies aren't properly visible online. Restaurants, cafés, hotels, and other local vendors can display menus, promotions, and real-time updates on WanderPlan's web-based business dashboard. Travellers can instantly obtain reliable information through targeted deals and business profiles.

1.4.6. Gamification and Contributor Rewards

Although it is frequently overlooked, encouraging Tourists to share reviews and insights is crucial for a healthy ecosystem. Through gamification—reward points, contributor badges, and discount coupons—WanderPlan increases user engagement and encourages active participation and ongoing platform enhancement.

1.4.7. Enhanced Experience Through Unified Ecosystem

Instead of juggling multiple apps for budgeting, food, travel, and reviews, WanderPlan brings every essential feature under one intelligent, easy-to-navigate

platform. This unified experience ensures seamless planning, informed choices, authentic interactions, and better resource utilization throughout the trip.

1.5. Report Outline

1.5.1. Chapter 1: Introduction

The WanderPlan project's foundation is presented in this chapter, along with opportunities that have been identified, key stakeholders, the idea's motivations, and the main obstacles encountered during development. Along with outlining the project's goals and objectives, it provides a high-level summary of the suggested solution, emphasising the key components of the business dashboard and mobile application.

1.5.2. Chapter 2: Literature / Market Survey

This chapter examines existing travel applications and digital tourism solutions, analyzing the functionalities they offer and identifying the core gaps in budget planning, real-time information accuracy, personalized recommendations, and business digitalization. It also explains the tools, technologies, and research methodologies that guide the development of WanderPlan.

1.5.3. Chapter 3: Requirement Engineering

Chapter 3 details the functional and non-functional requirements of the WanderPlan system. It describes the requirement elicitation techniques used—such as interviews, market study, and competitor analysis—and includes identified system constraints. Test case–based defect detection is also incorporated to validate requirement completeness and correctness.

1.5.4. Chapter 4: System Design

This chapter presents the architectural framework of WanderPlan, covering system architecture diagrams, detailed use cases, activity diagrams, sequence diagrams, ERD, and the database schema. It also includes component diagrams and discusses how the design supports scalability, modularity, and integration between the mobile application and business dashboard.

1.5.5. Chapter 5: Implementation

This chapter explains the implementation phase, including work breakdown structure (WBS), roles and responsibilities of team members, development tools and technologies, and code structuring standards. It also includes backend API development, frontend implementation details (mobile + dashboard), and deployment environment setup. White-box testing and defect detection practices are documented as well.

1.5.6. Chapter 6: Testing & Evaluation

The system's validation is the main topic of Chapter 6. It contains thorough test cases covering both functional and non-functional requirements for every feature of the WanderPlan business dashboard and mobile application. To illustrate dependability and performance, the findings, assessments, and defect fixes are also covered.

1.5.7. Chapter 7: Conclusion & Outlook

This chapter highlights WanderPlan's accomplishments while considering how the project resolves the issues that have been identified. Along with highlighting its weaknesses, it offers suggestions for the future, such as the incorporation of halal/haram food filtering, AI-driven dynamic schedules, expanded Plugin features, and additional improvements to boost platform utility and user experience.

Chapter: 2

LITERATURE / MARKET SURVEY

Chapter 2

MARKET SURVEY

2.1. Introduction

These days, there are numerous apps for travel planning, including ones for budgeting, dining, locations, and navigation. Because of this, tourists find the process confusing, time-consuming, and challenging to handle. Current budget, food, and travel apps function well on their own, but they don't provide a cohesive and customised experience.

In order to fill this gap, WanderPlan provides travellers with an all-in-one platform that allows them to plan trips, manage budgets, look through recommendations, navigate locations, read real reviews, and interact with local businesses and contributors. In order to comprehend current solutions and pinpoint the issues that WanderPlan seeks to address, this chapter examines apps that are currently available in these categories.

2.2. Literature Review / Technologies Overview

2.2.1. Historical Context

Over the past ten years, digital travel solutions have changed dramatically. Travel agencies, manual research, and guidebooks were the first resources used by tourists. These manual processes were eventually supplanted by programs like Google Maps, TripAdvisor, and budgeting tools. However, rather than developing as an unified system, budgeting, suggestions, and discovery tools continued to develop independently.

SplitWise and TrabeePocket are budgeting apps that only track expenses. While food delivery services like FoodPanda made dining more accessible, they lacked travel-related insights. Yelp and TripAdvisor are examples of review sites that prioritise community feedback over a traveler's budget or specific trip requirements. Although

navigation apps offered precise routes, they did not offer recommendations for contextual travel.

The fragmentation of the digital travel industry creates a significant need for a single system, which WanderPlan seeks to fill.

2.2.2. Key Concepts & Theories

Travel planning in the modern era is largely dependent on digital support, where ideas like location-based services, user-generated content, personalised recommendations, and budget optimisation are crucial. The goal of budget optimisation is to assist travellers in effectively allocating their spending according to constraints and preferences. Recommendation systems make recommendations for restaurants, attractions, and lodging based on Tourist behaviour and data patterns. By offering actual traveler experiences, opinions, and insights, user-generated content enhances authenticity. Furthermore, real-time navigation and context-aware recommendations are made possible by location-based services, which are backed by mapping technologies and distance estimation algorithms. Together, these ideas inform WanderPlan's design, guaranteeing a smooth and wise travel experience.

2.2.3. Related Work

We looked at applications in a variety of categories to assess WanderPlan's applicability and necessity:

1. Budget Apps:

- SplitWise does not help with planning or destination discovery, but it does manage group-based expense splitting.
- TrabeePocket is good at keeping track of travel expenses, but it doesn't have review or recommendation systems.
- Wallet prioritises tracking finances over budgeting for individual trips.

2. Food Apps:

- FoodPanda is great for placing food orders, but it's not made for organising or exploring travel.

- Yelp and Restaurant Finder provide ratings and reviews, but they don't take budget preferences or travel routes into account.

3. Travel & Trip Apps:

- TripAdvisor is good at reviews but poor at personalised planning and budgeting.
- Google Travel is useful for general planning, but it doesn't offer personalised regional recommendations or social interaction.
- Airbnb Experiences does not plan the entire trip; it only concentrates on activities.

They Lack:

i. No single app for both trip planning and budgeting

ii. No intelligent suggestions powered by AI

iii. Restricted customised navigation-based exploration

iv. Absence of a reward-based, integrated contributor ecosystem

v. No business dashboard

vi. Absence of regional, local insights specific to Pakistan's tourism culture

WanderPlan addresses these missing components.

2.2.4. Technologies Overview

WanderPlan uses a variety of contemporary technologies to enable real-time travel assistance. Cross-platform frameworks for mobile development, like React Native, make it possible for visitors and contributors to have a seamless and responsive user interface. Node.js and Express power the backend, handling RESTAPIs for business dashboards, recommendations, budget processing, and authentication. Database like MongoDB are used to manage and store structured and media-rich data.

2.3. Existing Systems

2.3.1. SplitWise

- A well-known app for splitting expenses.
- Excellent for travelling in groups.
- lacks navigation, suggestions, and trip planning.

2.3.2. Trabee Pocket

- Designed with travel budgeting in mind.
- Excellent user interface for cost breakdowns.
- Does not create plans or link users to dining establishments or events.

2.3.3. Wallet

- An app for tracking finances.
- Not related to travel.
- Absence of map integrations or discovery.

2.3.4. FoodPanda

- Strong food exploration in large cities.
- No suggestions based on travel.
- Unrelated to travel arrangements.

2.3.5. Restaurant Finder / Yelp

- Platforms that rely on reviews.
- Wide-ranging but not specifically tailored to Pakistan.
- Absence of integrated trip flows or budget estimation.

2.3.6. Trip Advisor

- Excellent for reviews.

- Inadequate for creating a budget.
- Pakistan's limited local insights.

2.3.7. Airbnb Experiences

- Offers carefully chosen experiences.
- Not related to navigation or budget planning.
- Not accessible everywhere in Pakistan.

2.3.8. Google Travel

- Excellent automated recommendations.
- Does not provide engagement with local contributors.
- No vendor business dashboard.

2.3.9. Survey Table

A simplified comparison of the studied systems is shown in **Table 2.1. :**

Application	Budgeting	Recommendations	Navigation	Reviews	Business Dashboard
Split Wise	Partially Available	Not Available	Not Available	Not Available	Not Available
Trabee Pocket	Available	Not Available	Not Available	Not Available	Not Available
Wallet	Available	Not Available	Not Available	Not Available	Not Available
FoodPanda	Not Available	Available	Not Available	Available	Not Available
Restaurant Finder/ Yelp	Not Available	Available	Not Available	Available	Not Available

TripAdvisor	Not Available	Available	Not Available	Available	Not Available
Google Travel	Not Available	Available	Available	Available	Not Available
Airbnb Exp.	Not Available	Available	Not Available	Available	Not Available

Table 2.1. Survey Table.

2.4. Summary

This chapter examined the state of travel planning solutions today and emphasised the shortcomings in local business integration, recommendations, reviews, budgeting, and navigation. Although a number of apps cover different facets of the travel experience, none offer a single, cost-effective solution. These results lay the groundwork for the discussion of the system's specific functional and non-functional requirements in the following chapter.

Chapter: 3

REQUIREMENT ENGINEERING

Chapter 3

REQUIREMENT ENGINEERING

3.1. Introduction

This chapter will discuss the stakeholders' current issues, how we collected the requirements, and what those requirements were (both functional and nonfunctional).

3.2. Problem Scenarios

The main problem statements found for WanderPlan are detailed in the following tables (Table 3.1 to Table 3.5). Stakeholder interviews, surveys, competitor analysis, and gap identification in tourism, navigation, budgeting, and local food discovery applications were used to create these problem scenarios.

3.2.1. Problem Statement 1: Budget Planning Difficulty	
The Problem	Travellers are unable to properly plan and manage their trip budgets.
Affects	Tourists
The Result of which	• The challenge of creating practical routes. • Time lost flipping between several apps for activity costs, accomodation fares and food prices. • Inaccurate budget expectations led to a bad trip. • Unintentionally, tourists overspend or underspend.

Benefits of	• Automatically created spending plan for lodging, travel, and meals. • Realistic travel schedules derived from local and real-time data. • A clear breakdown of costs to prevent unforeseen costs. • Enhanced confidence in planning and financial management.

Table 3.1. Problem Statement 01

3.2.2. Problem Statement 2: Unreliable Recommendations	
The Problem	The majority of apps display generic or sponsored listings, making it challenging for travellers to obtain reliable, local, and reasonably priced recommendations.
Affects	Tourists, Contributors
The Result of which	• Tourists end up at overprices or overrated places. • Finding undiscovered local treasures is challenging. • There is no platform that allows contributors to highlight their local expertise. • Travellers rely on inaccurate or out-of-date internet information.
Benefits of	• Travellers receive trustworthy, neighborhood-based recommendations. • Local contributors are recognised for their genuine recommendations. • Genuine quality businesses gain from genuine reviews rather than sponsored placements. • A more budget-friendly discovery experience.

Table 3.2. Problem Statement 02

3.2.3. Problem Statement 3: Navigation And Travel Cost	
The Problem	Tourists cannot accurately estimate transportation costs, distances, or directions, especially in unfamiliar cities where prices vary widely.
Affects	Tourists
The Result of which	• Transport services overcharge tourists. • Unknown distances lead to poor trip planning. • Navigating between locations wastes time. • Fear and confusion in strange places.
Benefits of	• Precise fare and distance estimates prior to travel. • Clearly defined routes and options. • A visual map of nearby recommendations. • Tourists will feel more confident and have safer mobility.

Table 3.3. Problem Statement 03

3.2.4. Problem Statement 4: Untrustworthy Reviews	
The Problem	Existing review platforms often contain outdated, fake, or paid reviews, making it difficult to trust community feedback.
Affects	Tourists, Business Owners
The Result of which	• Inaccurate reviews cause tourists to make bad choices. • It is difficult for businesses to gain the trust of real customers. • There are few options for reporting, filtering, and engaging with reviews.

Benefits of	• An open and interactive review feed. • Local donors who make significant reviews are rewarded. • Gamification promotes involvement. • Companies can improve trust by responding to reviews.

Table 3.4. Problem Statement 04

3.2.5. Problem Statement 5: Lack of Exposure & Tools to Reach Tourists	
The Problem	Many restaurants, hotels, and nearby attractions don't have the digital presence or resources necessary to advertise sales, reply to reviews, or track consumer interaction.
Affects	Business Owners.
The Result of which	• Businesses rely solely on walk-in clients. • There are no tools to advertise sales to specific audiences. • No analytics to identify patterns or visitor behaviour. • Insufficient customer communication.
Benefits of	• A specific business dashboard for managing profiles. • The capacity to post sponsored listings, deals, and advertisements. • Perceptive data derived from user visits and review patterns. • Increased visibility and interaction with visitors.

Table 3.5. Problem Statement 05

3.3. Elicitation Techniques

To comprehend the necessities, anticipations, and difficulties of the participants in the WanderPlan project, elicitation techniques were used. These techniques helped in verifying the system's practicability, revealing actual Tourist's issues, and establishing the requirements of the system through genuine feedback rather than guesswork. The undermentioned means were utilized in the course of the requirement elicitation period:

3.3.1. Interview

Structured interviews were organized with various stakeholders, comprising of regular travelers, local entrepreneurs, and tourist lovers in the city. The main issues pointed out by the interviews can be acknowledged through the below given Table.

MEETING RECORDS							
Sr #	**Date**	**Name**	**Role**	**Location**	**Purpose**	**Outcomes**	**Aligns with Goal**
01	25-09-2025	Bilal Ahmed	Local Food Business Owner	Saddar, Rawalpindi	Understand business onboarding needs	Wants visibility to nearby travelers, Can manage a simple dashboard.	Yes
02	09-10-2025	Ruman Bashir	Student Traveler	I-14 Markaz, Islamabad	Identify pain-points in planning trips & budgeting	Uses multiple apps for food, maps,booking, and reviews.	Yes

03	11-10-2025	Mr. Imran	Travel Agency Owner	Kohinoor Mills, Rawalpindi	Identification of pain points, requirement gathering from travel agency perspective	Suggested that WanderPlan should work as a complete ticketing and tour-booking platform.	No, Because WanderPlan concerns itself with aspects such as trip planning, budgeting, recommend ations, and user-generated content and is non-operational in this case.
04	15-10-2025	Abdul rehman	Travel Blogger	Chour Chowk, Rawalpindi	Gather some insights on the review and content contributions	Wants reward-based contribution, profile levels, and protected moderation against fake reviews.	Yes
05	19-10-2025	Mehroz Bilal	Frequent Traveler	Kohinoor Mills, Rawalpindi	Validate AI-based budget module usefulness	Likes the idea of AI-generated budget plans, but insists estimates should be transparent & editable.	Yes

Table 3.6. Interview Meeting Records.

3.3.2. Observation

The observation of travelers in an informal way and their interaction with modern travel platforms revealed numerous problems related to usability and trust. Moreover, we saw how local businesses marketed themselves and handled their customers. These observations revealed shortcomings like biased or promotional rankings, a lack of genuine reviews, and the difficulty in obtaining localized recommendations.

3.3.3. Brainstorming

The team held several brainstorming sessions to merge the feedback from stakeholders and turn it into organized system requirements. Besides, such discussions pointed us to the ways out, helped us to check features against actual needs, and led us to the decision on the top functionalities like AI-based trip budgeting, trustworthy recommendations, and Tourist rewards based on contributions.

3.3.4. Justification for Elicitation Technique

Interviews combined with observations and brainstorming led to the user-centered, validated, practical, and technically feasible requirements. The techniques adopted made it possible for us to gain a clear insight into stakeholder expectations and thereby convert them into the measurable and relevant system requirements for WanderPlan.

3.4. Functional Requirement

Definition: Functional requirements outline the precise tasks that the WanderPlan system must complete, including inputs, behaviours, and outputs. These specifications outline the key components, including computations, data processing, user interactions, and system reactions.

Note: The list does not include implied constraints like input validation. The requirements listed below are ranked by stakeholder value and priority; the use case ID is indicated by the numbers in square brackets.

3.4.1. Tourist

A single unified Tourist role. The same account can be used both for traveling and for contributing reviews.

3.4.1.1 Dashboard & Home

- The Tourist shall be displayed with the unified home page after login.
- The dashboard shall show personalized recommendations.
- The Tourist shall be able to navigate to Trips, Reviews, Maps, Profile.
- The Tourist shall be able to view notifications.

3.4.1.2. Trip & Budget Management

- The Tourist shall be able to create a trip budget.
- The Tourist shall be able to input budget preferences (travel mode, group size, destination, stay duration).
- The Tourist shall generate a suggested trip plan based on Tourist inputs.
- The Tourist shall be able to edit or customize a generated plan.
- The Tourist shall be able to manually create a custom trip.
- The Tourist shall be able to save trip plans.
- The Tourist shall be able to track previous trip histories.

3.4.1.3. Recommendations & Discovery

- The Tourist shall be able to view recommended restaurants, hotels, and attractions.
- The Tourist shall be able to filter recommendations by distance, rating, and price.
- The Tourist shall be able to open detailed pages for places.
- The Tourist shall be able to view deals and sponsored ads submitted by businesses.

3.4.1.4. Maps & Navigation

- The Tourist shall be able to search for destinations via map.
- The Tourist shall be able to view routes and directions.
- The system shall display estimated travel cost and distance.
- The Tourist shall be able to view nearby recommendations.

3.4.1.5. Reviews

- The Tourist shall be able to create a review.
- The Tourist shall be able to upload media (images) in a review.
- The Tourist shall be able to edit or delete their existing reviews.

- The Tourist shall be able to like, comment and react to reviews.
- The Tourist shall be able to report inappropriate content.

3.4.1.6. Profile & Leveling System

- The Tourist's profile shall show personal details and activity statistics.
- The system shall automatically assign Tourist Level based on the trip activity.
- The system shall automatically assign Reviewer Level based on the contribution.
- The Tourist shall be able to edit their personal details.
- The Tourist shall be able to view earned rewards or coupons.

3.4.2. Business Owner

A separate web dashboard for restaurants, hotels, and local service providers.

3.4.2.1. Business Registration & Setup

- The business owner shall be able to register a business account.
- The business owner shall be able to create a business profile.
- The business owner shall be able to update business details, such as:

  Name, Category, Menu/services, Price range, Location, Operating hours.

3.4.2.2. Deals & Ads Management

- The business owner shall be able to create deals/promotions.
- The business owner shall be able to manage active/inactive deals.
- The business owner shall be able to promote ads within the app.

3.4.2.3. Review Interaction

- The business owners shall be able to view reviews posted about their business.
- The business owners shall be able to reply to reviews.
- The business owner shall be able to report fake or harmful reviews.

3.4.2.4. Analytics Dashboard

- The business owner shall be able to view analytics such as:

  i. Customer visits.

  ii. Engagement.

  iii. Review Trends.

  iv. Popular Menu items/Services.

3.4.3. Admin

A seperate admin dashboard for management and verification of businesses as well as the content.

3.4.3.1. Business Management & Verification

- The Admin shall be able to view newly registered business profiles.
- The Admin shall be able to verify bussiness documents and details.
- The Admin shall be able to approve or reject business accounts.
- The Admin shall be able to Suspend or Reactivate the business account.
- The Admin shall be able to resolve-business-related quiries.

3.4.3.2. Content Moderation

- The Admin shall be able to review Tourist reviews & ratings.
- The Admin shall be able to remove inappropriate or false content.
- The Admin shall be able to monitor reported reviews.
- The Admin shall be able to handle spam or abusive content.
- The Admin shall be able to track repeat offenders.
- The Admin shall be able to maintain moderation logs.

3.4.3.3. Reports & Complaints Handling

- The Admin shall be able to view all Tourist-submitted reports.
- The Admin shall be able to categorize complaints (Business / Tourist).
- The Admin shall be able to assign priority level.
- The Admin shall be able to take corrective action (warn / suspend / delete).
- The Admin shall be able to maintain resolutions history.

3.4.3.4. System & Access Control

- The Admin shall be able to authenticate login .
- The Admin shall be able to manage admin roles & permissions.
- The Admin shall be able to log admin activities (audit trail).
- The Admin shall be able to session monitoring.
- The Admin shall be able to view security & risk alerts.

3.4.3.5. Analytics & Monitoring

- The Admin shall be able to view review engagement statistics.
- The Admin shall be able to monitor business onboarding.
- The Admin shall be able to view review trends & patterns.

3.5. Non-Functional Requirements

Non-functional requirements set forth the quality benchmarks and limitations within which WanderPlan is supposed to function. Among the mentioned characteristics are the performance, reliability, security, usability, maintainability, and so on, which affect the overall quality of the system rather than its functionality. Thus, such requirements make the application steady, seamless, and trustworthy for travelers, contributors, businesses, and the administration.

3.5.1. Performance Requirements (Quantitative)

WanderPlan must provide a smooth and responsive experience, especially since Tourists frequently switch between budget tools, maps, recommendations, and reviews.

- **App Response Time:**
  Normal screen transitions and operations should load within 5-10 seconds under a stable internet connection.
- **API Response Time:**
  Requests made to the backend should generally return a response within 5 seconds.
- **Search & Recommendation Time:**
  AI-assisted recommendations and place searches should ideally complete within 12 seconds.
- **Scalability Expectation:**
  The system should handle growth in Tourists, searches, and reviews without degrading performance.

3.6. SQA Activities: Defect Detection

3.6.1 Functional Test Cases

3.6.1.1 Tourist Test Case:

3.6.1.1.1 Sign In:

3.6.1.1.1.1 (Positive Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-T-SI-01	Tourist logs in with valid email and password	Tourist account exists and is verified	1. Enter valid email & password 2. Tap Sign In	Tourist is logged in and redirected to Home screen
TC-T-SI-02	Tourist remains logged	Tourist already	1. Login	Session persists and

	in after reopening app	logged in.	2. Close app 3. Reopen app	user stays logged in
TC-T-SI-03	Tourist successfully logs out.	Tourist logged in.	1. Open Settings 2. Press Logout	Tourist is logged out and redirected to Login screen

Table 8. Sign In Postive Test Case

3.6.1.1.1.2 (Negative Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-T-SI-01	Login attempt with incorrect password	Tourist account exists	1. Enter valid email & wrong password 2. Tap Sign In.	Error message displayed: "Invalid credentials"
TC-T-SI-02	Login attempt with unregistered email.	No account exists	1. Enter unregistered email 2. Tap Sign In.	System displays "Email not found".
TC-T-SI-03	Login attempt with empty fields.	App Installed.	1. Leave fields blank 2. Tap Sign In.	System prompts to fill required fields.

Table 9. Sign In Negative Test Case

3.6.1.1.2 Sign Up:

3.6.1.1.2.1 (Positive Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-T-SU-01	Successful Tourist registration.	App Installed.	1. Open Sign-Up 2. Enter valid details 3. Tap Sign Up	Account is created successfully.
TC-T-SU-02	Tourist receives verification.	Valid data provided.	1. Register 2. Check email.	Tourist receives account verification notification

Table 10. Sign Up Positive Test Case

3.6.1.1.2.2 (Negative Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-T-SU-01	Registration using already registered email.	Email exists.	1. Enter existing email 2. Tap Sign Up	System displays "Email already registered".
TC-T-SU-02	Registration with incomplete data.	App Installed.	1. Leave a field blank 2. Tap Sign Up	System prompts to fill required fields.
TC-T-SU-03	Registration with invalid email format.	App Installed.	1. Enter invalid email format 2. Tap Sign Up	System rejects email format.

Table 11. Sign Up Negative Test Case

3.6.1.2 Admin Test Case:

3.6.1.2.1 Sign In:

3.6.1.2.1.1 (Positive Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-A-SI-01	Admin logs in with valid credentials.	Admin account exists.	1. Open Admin Panel. 2. Enter valid credentials. 3. Sign In.	Admin Dashboard is displayed.
TC-A-SI-02	Admin remains logged in after page refresh.	Admin logged In.	1. Login 2. Refresh page	Session remains active.

Table 12. Admin Sign In Positive Test Case

3.6.1.2.1.2 (Negative Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-A-SI-01	Admin logs in with wrong password.	Admin account exists.	1. Enter wrong password 2. Press Sign In.	System displays "Invalid password".
TC-A-SI-02	Login with empty fields.	Admin panel open.	1. Leave both fields blank. 2. Press Sign In.	System prompt to fill all fields.
TC-A-SI-03	Non-Admin tries to access Admin Panel.	Non-admin account exists.	1. Try logging in via Admin Panel. 2. Enter normal user credentials.	Access Denied.

Table 13. Admin Sign In Negative Test Case

3.6.1.3 Reviews Test Case:

3.6.1.3.1 Submit Review:

3.6.1.3.1.1 (Positive Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-SR-01	Tourist posts a valid review.	User logged in.	1. Open review page. 2. Add Review 3. Submit	Review is successfully posted.
TC-SR-02	Review contains both text, rating and image.	Logged-in user.	1. Enter comments, ratings and image(s). 2. Submit.	Review appears in review list.

Table 14. Submit Review Positive Test Case.

3.6.1.3.1.2 (Negative Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-SR-01	Submit review without rating, text or image.	User logged in.	1. Leave review section blank. 2. Submit.	System shows the prompt to fill out all fields.
TC-SR-02	Image size and format is invalid.	Logged-in user.	3. Enter image(s) with a greater size and an invalid extension. 4. Submit.	System shows the prompt to submit image with less size and .jpg / .png extension.

Table 15. Submit Review Negative Test Case.

3.6.1.3.2 Manage Review:

3.6.1.3.2.1 (Positive Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-MR-01	User edits their review.	Review exists.	1. Go to 'My Reviews'. 2. Edit review. 3. Save.	Review updates succesfully.
TC-MR-02	User deletes their review.	Review exists.	1. Opens 'My Reviews'. 2. Deletes review.	Review is removed.

Table 16. Manage Review Positive Test Case.

3.6.1.3.2.2 (Negative Test Case) Valid Scenarios:

T.C ID	Test Case Description	Pre-Conditions	Test Steps	Expected Result
TC-MR-01	Comments something inappropriate.	Review exists.	1. Opens reviews. 2. Comment on someone else review.	Comment deleted.

Table 17. Manage Review Negative Test Case.

Chapter: 4

SYSTEM DESIGN

Chapter 4

SYSTEM DESIGN

4.1. Introduction

In this chapter, we present all the essential system design artifacts for WanderPlan, including its overall architecture, key design diagrams, detailed use cases, activity diagrams, and the database structure. These elements collectively outline how the system functions, how different modules interact, and how the application supports the needs of tourists, contributors, and business owners.

4.2. Architectural Design

The architectural design of WanderPlan follows a layered (n-tier) architecture to ensure modularity, scalability, and efficient data flow between its mobile application, business dashboard, backend, and database.

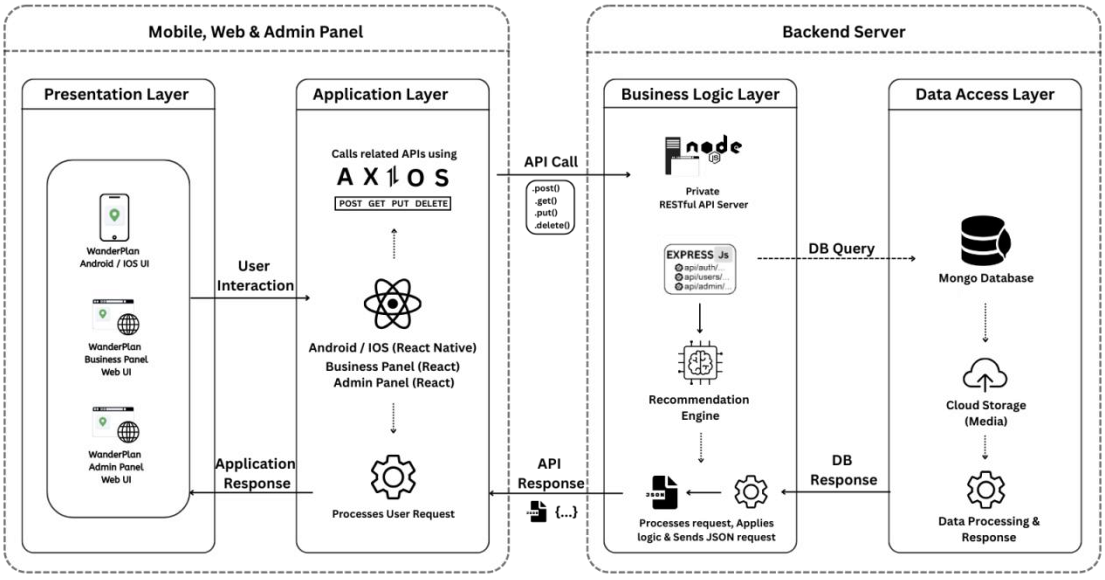

Figure 1. Architecture Diagram of WanderPlan.

4.3. Detailed Design

The WanderPlan system's detailed design is shown in this section. It concentrates on how various users engage with the system via specified features. Use case diagrams are used to clearly depict system behaviour from the viewpoint of the user in the detailed design.

4.3.1. Use Case Diagrams

4.3.1.1. Authentication Use Case Diagram

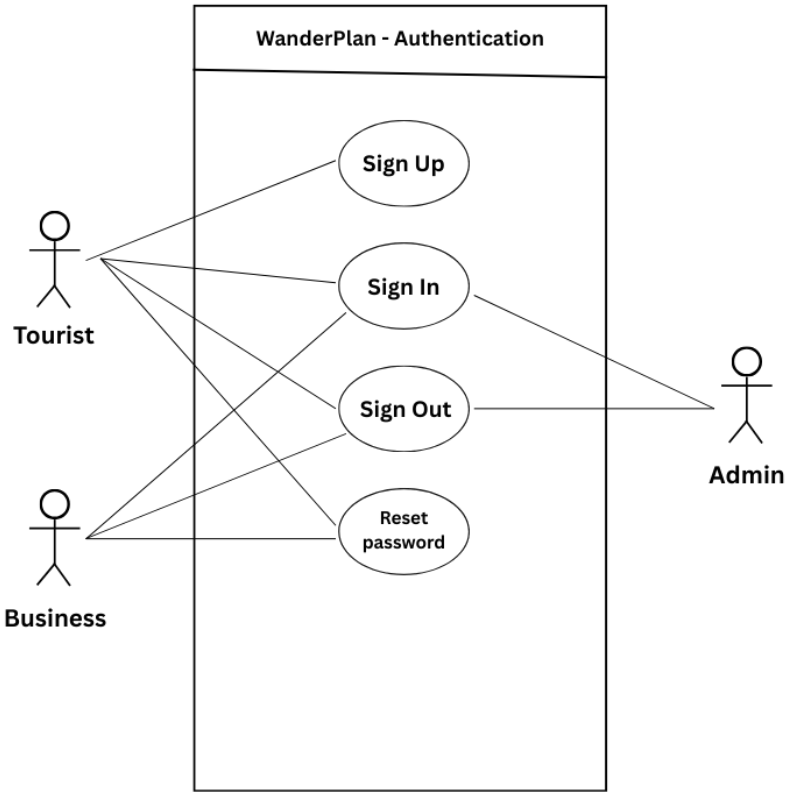

Figure 2. Authentication Use Case Diagram

4.3.1.2. Dashboard Use Case Diagram

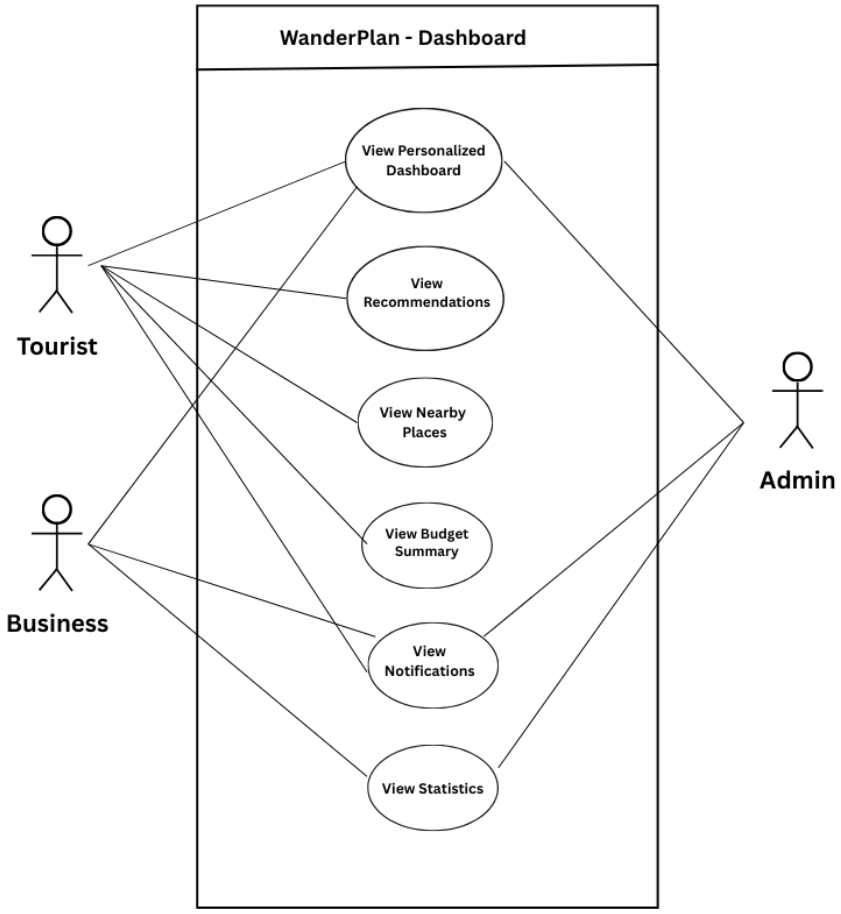

Figure 3. Dashboard Use Case Diagram

4.3.1.3. Reviews & UGC Use Case Diagram

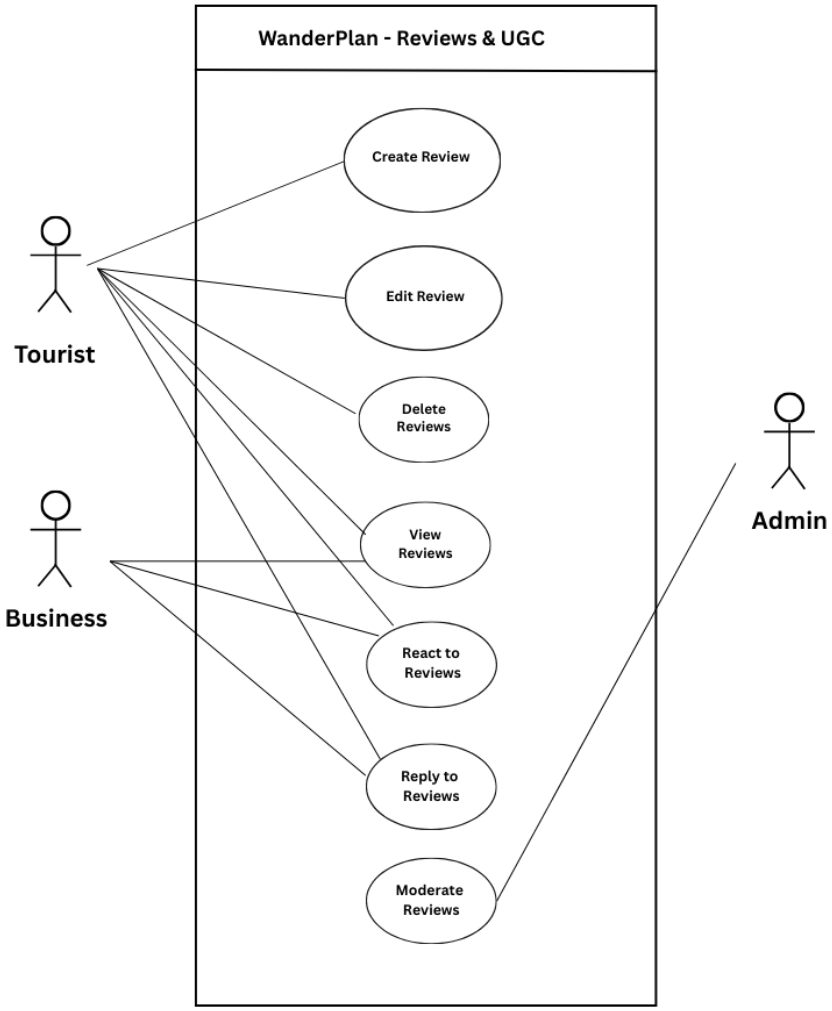

Figure 4. Reviews & UGC Use Case Diagram

4.3.1.4. Trip & Budget Planning Use Case Diagram

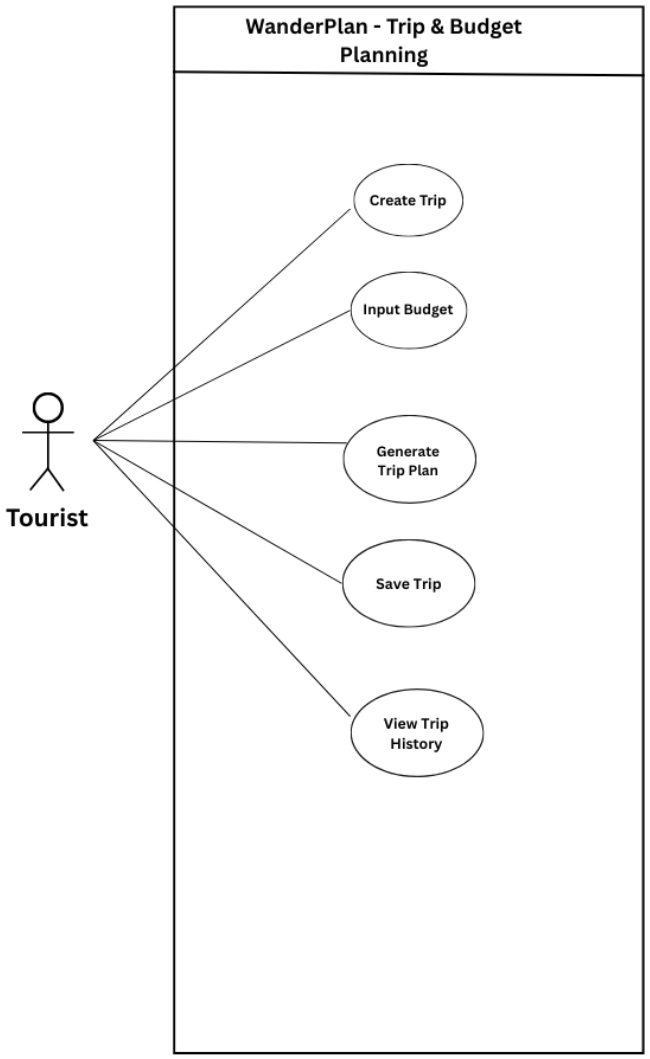

Figure 5. Trip & Budget Planning Use Case Diagram

4.3.1.5. Profile Use Case Diagram

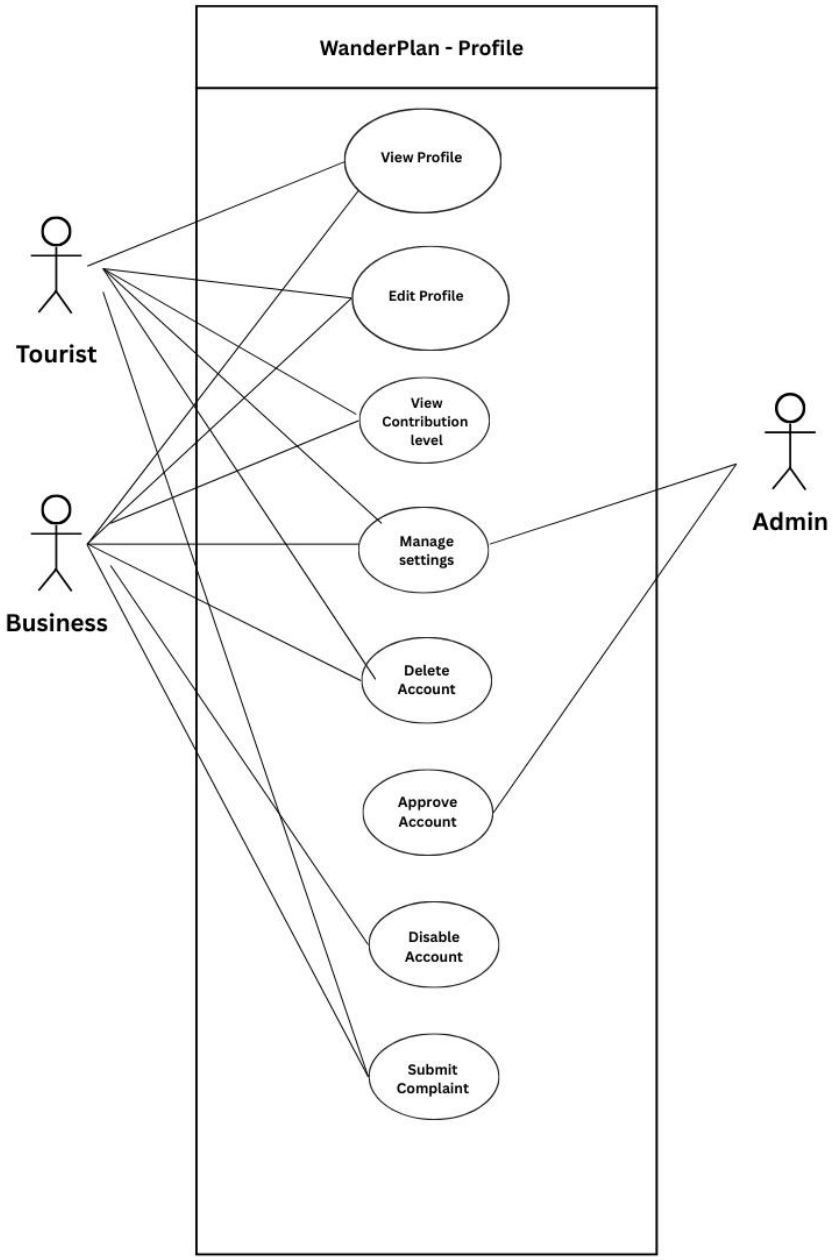

Figure 6. Profile Use Case Diagram

4.3.1.6. Business Use Case Diagram

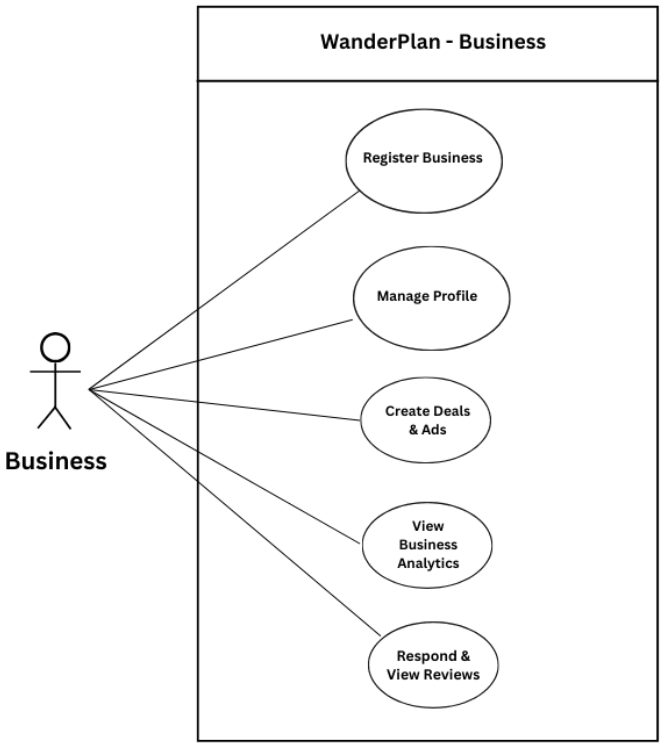

Figure 7. Business Use Case Diagram

4.3.1.7. Admin Use Case Diagram

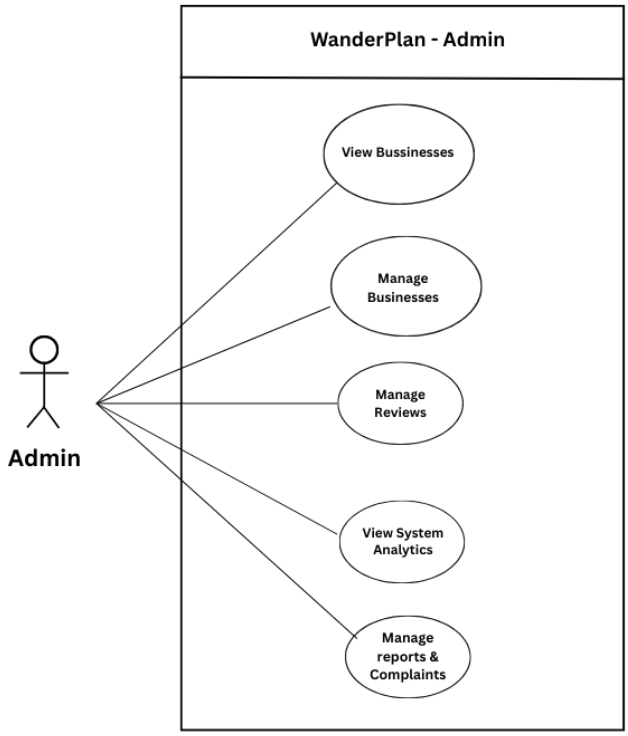

Figure 8. Admin Use Case Diagram

4.3.2. Fully Dressed Use Cases

4.3.2.1. Authentication

4.3.2.1.1. Sign Up

Use Case ID	001	
Use Case	Sign Up	
Actors	Tourist	
Trigger	Tourist wants to create an account	
Pre Conditions	Tourist has launched into the application	
Post Conditions	Tourist has successfully created account and redirected to home page.	
Basic Flow	**Actor**	**System**
	1. Tourist will press the button "Sign Up" on the login screen. 2. Tourist will enter the full name & email. 3 . Tourist will enter their password. 4 . Tourist will enter the code sent to their email.	1.1. Tourist will be directed to the sign up form. 2.1. System will validate the fields. 3.1 System will verify password strength. 3.2 System will redirect the Tourist to Email verification screen. 4.1. System validates the code and redirects it to Home Page.
Alternate Flow	**Actor**	**System**
	2a. Email is already in use. 1. Tourist shall return to the login screen. 2. Tourist shall enter their email along with the password to log into their existing account. **2b.** Email is already registered, but Tourist does not remember password. 1. Tourist shall return to login screen. 2. Tourist will click "Forgot Password"	2.1 System will verify credentials. 2.2 System will redirect Tourist to the home page. 3.1 System Validates the Email. 4.1 System will send a reset link to their email.

	3. Tourist will enter their Email. 4. Tourist clicks “Send Reset Link” 5. Tourist shall login to the system with new password.	5.1 System will verify credentials. 5.2 System will redirect Tourist to home page.

Table 18. Sign Up Use Case

4.3.2.1.2. Sign In

Use Case ID	002	
Use Case	Sign In.	
Actors	Tourist, Business, Admin.	
Trigger	Tourist wants to log into their account.	
Pre Conditions	Tourist has launched into the application.	
Post Conditions	Tourist has successfully logged in and redirected to home page.	
Basic Flow	**Actor**	**System**
	1. User will enter their login credentials. 2. User will enter the verification code.	1.1 System will verify the credentials. 1.2 System will send a verification code on the Email of Tourist. 2.1 System will validate the code. 2.2 System will redirect Tourist to home page.
Alternate Flow	**Actor**	**System**
	1a. User enters wrong credentials 1. User will be prompted to enter correct credentials. 2. User will enter login credentials again. **1b.** User with forgotten password 1. User will click “Forgot Password” 2. User will enter their Email. 3. User clicks “Send Reset Link” 4. User shall login to the system with new password.	2.1 System will verify credentials. 2.2 System will redirect user to the home page. 2.1 System Validates the Email. 3.1 System will send a reset link to their email.

		4.1 System will verify credentials. 4.2 System will redirect user to home page.
	2a. User doesnot receive email. 1. User will click “Resend” 2. User will enter the verification code.	 1.1 System will send the verification code again. 2.1 System validates the code. 2.2 System will redirect user to the home page.

Table 19. Sign In Use Case

4.3.2.1.3. Sign Out

Use Case ID	003	
Use Case	Sign Out	
Actors	Tourist, Business, Admin	
Trigger	User wants to log out of their accounts.	
Pre Conditions	Users are signed into their accounts.	
Post Conditions	User has successfully logged out of their accounts.	
Basic Flow	**Actor**	**System**
	1. User clicks “Sign Out”.	1.1 System clears session.
Alternate Flow	**Actor**	**System**
	1a. User faces network problem while logging out. 1. User will have network problem.	 1.1 System will show ‘You will be logged out automatically once the network stabilize’.

Table 20. Sign Out Use Case.

4.3.2.1.4. Reset Password

Use Case ID	004
Use Case	Reset Password.
Actors	Tourist, Business, Admin

Trigger	Tourist wants to reset their forgotten password.	
Pre Conditions	User knows the email which was used to create the account.	
Post Conditions	User will have successfully reset their password.	
Basic Flow	**Actor**	**System**
	1. User will click “Forgot Password” 2. User will enter their Email. 3. User clicks “Send Reset Link” 4. User shall login to the system with new password.	2.1 System Validates the Email. 3.1 System will send a reset link to their email. 4.1 System will verify credentials. 4.2 System will redirect user to home page.
Alternate Flow	**Actor**	**System**
	3a. User doesnot receive email. 1. User will click “Resend” 2. User shall login to the system with new password.	1.1 System will send the reset link again. 2.1 System will verify credentials. 2.2 System will redirect user to the home page.

Table 21. Reset Password Use Case.

4.3.2.2. Dashboard

4.3.2.2.1. View Personalized Dashboard.

Use Case ID	005	
Use Case	View Personalized Dashboard.	
Actors	Tourist, Business, Admin	
Trigger	User wants a personalized space.	
Pre Conditions	User is logged in the system.	
Post Conditions	User is shown the dashboard.	
Basic Flow	**Actor**	**System**
	1. User will verify the email. 2. Users will be shown their	1.1 System loads specified dashboards for each user.

	specified dashboards (Home Page for the mobile application).	
Alternate Flow	**Actor**	**System**
	1a. Service Unavailable 1. User will stabilize their connection.	1.1 System will show cached data until network is stabilized.

Table 22. View Personalized Dashboard Use Case.

4.3.2.2.2. View Recommendations.

Use Case ID	006	
Use Case	View Recommendations	
Actors	Tourist.	
Trigger	Users wants to explore some places.	
Pre Conditions	User is logged in the system.	
Post Conditions	User is shown all the recommended places.	
Basic Flow	**Actor**	**System**
	1. User will be redirected to the home page. 2. User clicks on one of the recommendations. 3. User clicks the 'Bookmark' sign.	1.1 System will show all the attractions, hotels and restaurants which the user can visit. 2.1 System redirect it on the page with that recommendation's details. 3.1 System saves that attraction to 'Saved' in the profile.
Alternate Flow	**Actor**	**System**
	1a. No data	1. System will show the placeholders.

Table 23. View Recommendations Use Case.

4.3.2.2.3. View Nearby Places.

Use Case ID	007
Use Case	View Nearby Places
Actors	Tourist.
Trigger	User wants to visit someplace nearby.

Pre Conditions	User has enabled location permission.	
Post Conditions	All the nearby recommendations are shown on the map.	
Basic Flow	**Actor**	**System**
	1. User clicks “Maps” bar. 2. User navigates on the map. 3. User searches any place nearby with the filters.	1.1 System gets user location. 1.2 System fetches places nearby. 3.1 System fetches the location and shows it to user.
Alternate Flow	**Actor**	**System**

Table 24. View Nearby Places Use Case.

4.3.2.2.4. View Budget Summary.

Use Case ID	008	
Use Case	View Budget Summary	
Actors	Tourist.	
Trigger	User wants to see all the spendings till now.	
Pre Conditions	User has atleast one trip.	
Post Conditions	Budget summary displayed.	
Basic Flow	**Actor**	**System**
	1. User opens their profile. 2. User clicks on “budget summary”.	 2.1 System retrieves budget data. 2.2 System shows total budget and usage.
Alternate Flow	**Actor**	**System**
	2a. No budget.	 1. System asks the user to create budget.

Table 25. View Budget Summary Use Case.

4.3.2.2.5. View Notifications.

Use Case ID	009	
Use Case	View Notifications.	
Actors	Tourist, Business, Admin.	
Trigger	User wants to stay alert.	
Pre Conditions	User is logged into the application.	
Post Conditions	All notifications shown.	
Basic Flow	**Actor**	**System**
	1. User will navigate to "My Notifications".	1.1 System shows the user their notifications.
Alternate Flow	**Actor**	**System**
	1a. No notifications received.	

Table 26. View Notifications Use Case.

4.3.2.2.6. View Statistics

Use Case ID	010	
Use Case	View Statistics.	
Actors	Business, Admin	
Trigger	User wants to see broad view of their data.	
Pre Conditions	Users are logged into their account.	
Post Conditions	Users are shown all the statistics.	
Basic Flow	**Actor**	**System**
	1. Business Owner or Admin will go to their dashboards. 2. Business Owner or Admin will view the statistics charts there.	2.1 System fetches the statistics data. 2.2 System renders the charts.
Alternate Flow	**Actor**	**System**
	1a. Insufficient data.	

Table 27. View Statistics.

4.3.2.3. Trip & Budget Planning

4.3.2.3.1. Create Trip.

Use Case ID	011	
Use Case	Create Trip.	
Actors	Tourist	
Trigger	User wants to make a trip.	
Pre Conditions	User is logged into their account.	
Post Conditions	User has successfully made a trip.	
Basic Flow	**Actor**	**System**
	1. User goes to trip planning section. 2. User fills the budget. 3. User fills up the preferences. 4. User edits the trip plan.	1.1 System asks for budget. 2.1 System asks for preferences on the trip. 3.1 System starts to make the budget and trip plan. 3.3 System shows the trip plan. 4.1 System edits the changing and provides the plan.
Alternate Flow	**Actor**	**System**
	2a. Invalid budget. 1. User Enters the budget again. **4a.** Edits which makes the increase in budget. 1. User adds something which makes the budget count more. 2. User keeps the edit in the budget.	1.1 System provides increase in budget error. 2.1 System edits the changing and provides the plan.

Table 28. Create Trip Use Case.

4.3.2.3.2. Input Budget.

Use Case ID	012
Use Case	Input Budget.
Actors	Tourist.
Trigger	User has to input their budget for the trip.
Pre Conditions	User is creating a trip.

Post Conditions	User's budget is distributed amongst the expenses of the trip.	
Basic Flow	**Actor**	**System**
	1. User goes to trip planning section. 2. User inputs their budget.	1.1 System asks for budget.
Alternate Flow	**Actor**	**System**
	2a. Invalid budget. 1. User enters the budget again.	

Table 29. Input Budget Use Case.

4.3.2.3.3. Generate Trip Plan.

Use Case ID	013	
Use Case	Generate Trip Plan.	
Actors	Tourist.	
Trigger	User wants to see their trip plan.	
Pre Conditions	User enters the trip details & preferences.	
Post Conditions	User is shown their trip plan.	
Basic Flow	**Actor**	**System**
	1. User enters their budget. 2. User enters their preferences. 3. User is displayed with the trip plan.	2.1 System fetches the budget and preferences and provides all the plan within budget.
Alternate Flow	**Actor**	**System**
	1a. Invalid Budget 1. User enters the budget again.	

Table 30. Generate Trip Plan Use Case.

4.3.2.3.4. Save Trip.

Use Case ID	014
Use Case	Save Trip.
Actors	Tourist.
Trigger	User wants to save the trip.
Pre Conditions	User has generated the trip plan.

Post Conditions	The trip plan is saved.	
Basic Flow	**Actor**	**System**
	1. User is shown the trip plan. 2. User has done all the edits they want. 3. User adds the trip plan to favorite.	2.1 System finalizes the trip plan. 3.1 System adds the trip to saved plans in profile.
Alternate Flow	**Actor**	**System**

Table 31. Save Trip Use Case.

4.3.2.3.5. View Trip History.

Use Case ID	015	
Use Case	View Trip History.	
Actors	Tourist.	
Trigger	User wants to view trip history.	
Pre Conditions	User has atleast one trip plan.	
Post Conditions	User is displayed their all trips.	
Basic Flow	**Actor**	**System**
	1. User opens the profile and looks for saved trips.	1.1 System fetches the saved trips. 1.2 System displays all the saved trips.
Alternate Flow	**Actor**	**System**
	1a. No saved trips. 1. User has no saved trips.	1.1 System shows empty state.

Table 32. View Trip History Use Case.

4.3.2.4. Reviews & User Generated Content

4.3.2.4.1. Create Review

Use Case ID	016
Use Case	Create Review.
Actors	Tourist.
Trigger	User wants to share their experience.

Pre Conditions	User is logged into the account.	
Post Conditions	Review is published.	
Basic Flow	**Actor**	**System**
	1. User opens the review tab. 2. User clicks on the '+' sign. 3. User enters the place, category, rating, their review, adds picture of the place. 4. User clicks 'Post Review'.	1.1 System shows all the reviews to the user. 3.1 System validates all the information. 4.1 System saves all information and posts the review.
Alternate Flow	**Actor**	**System**

Table 33. Create Review Use Case.

4.3.2.4.2. Edit Review

Use Case ID	017	
Use Case	Edit Review	
Actors	Tourist.	
Trigger	User wants to edit their review.	
Pre Conditions	Review the user wants to edit is already published.	
Post Conditions	Review is updated.	
Basic Flow	**Actor**	**System**
	1. User opens the profile tab. 2. User goes to 'My Reviews'. 3. User clicks on the three dots of the review they want to edit. 4. User clicks 'Edit'. 5. User edits the review.	2.1 System shows the user their reviews. 4.1 System allows the user to edit. 5.1 System saves the changing.
Alternate Flow	**Actor**	**System**

Table 34. Edit Review Use Case.

4.3.2.4.3. Delete Review

Use Case ID	018	
Use Case	Delete Review	
Actors	Tourist	
Trigger	User wants to delete their review.	
Pre Conditions	Review the user wants to delete is already published.	
Post Conditions	Review is removed from the feed.	
Basic Flow	**Actor**	**System**
	1. User opens the profile tab. 2. User goes to 'My Reviews'. 3. User clicks on the three dots of the review they want to edit. 4. User clicks 'Delete'. 5. User deletes the review.	2.1 System shows the user their reviews. 4.1 System confirms to delete. 5.1 System saves the changing.
Alternate Flow	**Actor**	**System**

Table 35. Delete Review Use Case.

4.3.2.4.4. View Reviews

Use Case ID	019	
Use Case	View Reviews	
Actors	Tourist, Business.	
Trigger	User wants to view reviews.	
Pre Conditions	User is logged into their account.	
Post Conditions	User is viewing the reviews.	
Basic Flow	**Actor**	**System**
	1. User Opens their profile. 2. User goes to 'My Reviews'.	2.1 System shows the user their reviews.
Alternate Flow	**Actor**	**System**

Table 36. View Reviews Use Case.

4.3.2.4.5. React to Reviews

Use Case ID	020	
Use Case	React to Reviews.	
Actors	Tourist, Business.	
Trigger	User wants to react to review.	
Pre Conditions	User is logged into their account.	
Post Conditions	Reaction is recorded.	
Basic Flow	**Actor**	**System**
	1. User goes to 'Reviews' tab. 2. User views all reviews. 3. User reacts to the review they like.	3.1 System saves all changing.
Alternate Flow	**Actor**	**System**
	2a. No reviews 1. User does not see any review.	1.1 System shows a message.

Table 37. React to Reviews Use Case.

4.3.2.4.6. Reply to Reviews

Use Case ID	021	
Use Case	Reply to Reviews.	
Actors	Tourist, Business	
Trigger	User wants to comment on the reviews.	
Pre Conditions	The review to reply on exists.	
Post Conditions	Reply is recorded.	
Basic Flow	**Actor**	**System**
	1. User goes to 'Reviews' tab. 2. User views all reviews. 3. User replies to the review.	3.1 System posts and saves the reply.
Alternate Flow	**Actor**	**System**
	3a. Inappropriate Content	

Table 38. Reply to Reviews Use Case.

4.3.2.4.7. Moderate Reviews

Use Case ID	022	
Use Case	Moderate Reviews.	
Actors	Admin.	
Trigger	Admin wants to moderate the reviews.	
Pre Conditions	Admin is logged into their account.	
Post Conditions	Review is moderated.	
Basic Flow	**Actor**	**System**
	1. Admin opens the 'Review & Content' tab. 2. Admin views the amount of reviews flagged on a review. 3. Admin approves / removes / Flags a review.	1.1 System fetches all the reviews. 3.1 System saves the changing.
Alternate Flow	**Actor**	**System**

Table 39. Moderate Reviews Use Case.

4.3.2.5. Profile

4.3.2.5.1. View Profile

Use Case ID	023	
Use Case	View Profile.	
Actors	Tourist, Business.	
Trigger	Users wants to open their profiles.	
Pre Conditions	User is logged into their accounts.	
Post Conditions	User is shown their profiles.	
Basic Flow	**Actor**	**System**
	1. User opens their dashboards. 2. User clicks on profile.	2.1 System shows the profile data.
Alternate Flow	**Actor**	**System**

Table 40. View Profile Use Case.

4.3.2.5.2. Edit Profile

Use Case ID	024	
Use Case	Edit Profile.	
Actors	Tourist, Business.	
Trigger	User wants to edit their profile.	
Pre Conditions	User is logged into their account.	
Post Conditions	The profile of the user is updated.	
Basic Flow	**Actor**	**System**
	1. User opens their dashboards. 2. User clicks on profile. 3. User make some changing in profile.	 2.1 System shows profile data. 3. System updates the profile & saves data.
Alternate Flow	**Actor**	**System**

Table 41. Edit Profile Use Case.

4.3.2.5.3. View Contribution Level

Use Case ID	025	
Use Case	View Contribution Level.	
Actors	Tourist, Business.	
Trigger	User wants to edit their profile.	
Pre Conditions	User is logged into their account.	
Post Conditions	The profile of the user is updated.	
Basic Flow	**Actor**	**System**
	1. User opens the profile section.	1.1 System displays the contribution level based on the number of reviews.
Alternate Flow	**Actor**	**System**

Table 42. View Contribution Level Use Case.

4.3.2.5.4. Manage Settings

Use Case ID	026	
Use Case	Manage Settings.	
Actors	Tourist, Business, Admin.	
Trigger	User wants to manage their account settings.	
Pre Conditions	User is logged into their account.	
Post Conditions	The changes are applied.	
Basic Flow	**Actor**	**System**
	1. User opens their dashboards. 2. User goes to the settings. 3. User makes the changing in the settings.	2.2 System allows the user to change the different type of information and permissions. 3.1 System updates the changes made and saves it.
Alternate Flow	**Actor**	**System**

Table 43. Manage Settings Use Case.

4.3.2.5.5. Delete Account

Use Case ID	027	
Use Case	Delete Account.	
Actors	Tourist, Business.	
Trigger	User wants to delete their account.	
Pre Conditions	User is logged into their account.	
Post Conditions	The account of the user is deleted.	
Basic Flow	**Actor**	**System**
	1. User opens their settings. 2. User clicks 'Delete Account'. 3. User enters the email and password of their account. 4. User verified the email by providing OTP.	1.1 System displays all the settings 2.1 System asks for login credentials. 3.1 System verifies the credentials. 3.2 System send the verification code to email. 4.1 System verifies the OTP.

	5. User confirms to delete. 6. User account is deleted.	4.2 System asks to confirm the deletion of profile. 6.1 System removes all the data of the user.
Alternate Flow	**Actor**	**System**
	3a. User enters invalid credentials. 1. The user enters invalid credentials of their account. **4a.** User enters wrong OTP. 1. User enter an invalid OTP. 2. User exceeds the 3 invalid tries. **5a.** User denies to delete. 1. User is taken back to the settings.	1.1 System allows the user to re-enter their credentials. 1.1 System allows to resend the mail 3 times. 2.1 System shows the prompt 'Email Verification failed. Please try again later after 5 hours.'.

Table 44. Delete Account Use Case.

4.3.2.5.6. Approve Account

Use Case ID	028	
Use Case	Approve Account.	
Actors	Admin.	
Trigger	Admin needs to verify the business accounts.	
Pre Conditions	Business registration request is received.	
Post Conditions	The business is approved/disapproved.	
Basic Flow	**Actor**	**System**
Alternate Flow	**Actor**	**System**

Table 45. Approve Account Use Case.

4.3.2.5.7. Disable Account

Use Case ID	029	
Use Case	Disable Account.	
Actors	Business.	
Trigger	User wants to disable their accounts.	
Pre Conditions	User is logged into their account.	
Post Conditions	The account of the user is disabled.	
Basic Flow	**Actor**	**System**
	1. User opens their settings.	1.1 System displays all the settings
	2. User clicks 'Disable Account'.	2.1 System asks for login credentials.
	3. User enters the email and password of their account.	3.1 System verifies the credentials. 3.2 System send the verification code to email.
	4. User verified the email by providing OTP.	4.1 System verifies the OTP. 4.2 System asks the confirmation of disabling the profile.
	5. User confirms to disable.	
	6. User account is disabled.	6.1 System disables the account of user temporarily.
Alternate Flow	**Actor**	**System**
	3a. User enters invalid credentials. 1. The user enters invalid credentials of their account.	1.1 System allows the user to re-enter their credentials.
	4a. User enters wrong OTP. 1. User enter an invalid OTP. 2. User exceeds the 3 invalid tries.	1.1 System allows to resend the mail 3 times. 2.1 System shows the prompt 'Email Verification failed. Please try again later after 5 hours.'.
	5a. User denies to disable. 2. User is taken back to the settings.	

Table 46. Disable Account Use Case.

4.3.2.5.8. Submit Complaint

Use Case ID	030	
Use Case	Submit Complaint.	
Actors	Tourist, Business.	
Trigger	User wants to send complaint.	
Pre Conditions	User is logged into their account.	
Post Conditions	The complaint of the user is submitted.	
Basic Flow	**Actor**	**System**
	1. User goes to profile. 2. User enters settings. 3. User clicks 'Help Center'. 4. User enters the complaint and sends it.	2.1 System displays all the settings. 3.1 System allows the user to enter complaint. 4.1 System takes the complaint to admin.
Alternate Flow	**Actor**	**System**

Table 47. Submit Complaint Use Case.

4.3.2.6. Bussiness

4.3.2.6.1. Register Business

Use Case ID	031	
Use Case	Register Business.	
Actors	Business.	
Trigger	User wants to register their business.	
Pre Conditions	User has launched the web application.	
Post Conditions	User has successfully sent their account for verification.	
Basic Flow	**Actor**	**System**
	1. User will press the button 'Create Account' or 'Get Started' on the welcome screen. 2. User will enter their first name, last name, Business Name, Email	1.1 User will be directed to the next screen to fill the form. 2.1 System will validate the fields. 2.2 System will send the user to

	and Password. 3. User enters the business information. 4. User enters all the business images. 5. User selects their business category. 6. User enters all the business location details. 7. User enters their business license image and proof of address.	the next screen where business information will be taken. 3.1 System validates the fields. 3.3 System will take the user to the next screen where business images will be taken i.e, logo etc. 4.1 System validates the image sizes and format. 4.2 System will take the user to next screen where business selects their category. 5.1 System will take the user to next screen where business location is entered. 6.1 System validated all the fields. 6.2 System will then take the user to another screen where business verification details would be taken. 7.1 System verifies all the image sizes and format. 7.2 System will then show the user that they can login their account after the approval of admin.
Alternate Flow	**Actor**	**System**
	2a. Business Name or Email is already in use. 1. User shall return to the welcome screen. 2. User shall press 'Login' button. 3. User shall enter their email or phone number along with their password to log into their existing account. **4b.** Email or phone is in use, but user does not remember password. 1. User shall return to welcome screen.	3.1 System will verify the credentials. 3.2 System will redirect user to dashboard

	2. User shall press “Login” button. 3. User will use the forgot / reset password functionality. 4. User shall login to the system with new password.	 4.1 System will verify credentials. 4.2 System will redirect user to the dashboard.

Table 48. Register Business Use Case.

4.3.2.6.2. Manage Profile

Use Case ID	032	
Use Case	Manage Profile.	
Actors	Business.	
Trigger	User wants to manage their profile.	
Pre Conditions	Business is verified and is logged into their account.	
Post Conditions	The business profile is updated.	
Basic Flow	**Actor**	**System**
	1. User opens their dashboards. 2. User goes to the business profile. 3. User edits the details in the basic info, gallery, menu/services, location pages. 4. User clicks the ‘save changes’ to save all the changes made.	 2.1 System allows the user to change the different type of information and permissions. 3.1 System validates the changings made. 4.1 System updates the changes made and saves data.
Alternate Flow	**Actor**	**System**

Table 49. Manage Profile Use Case.

4.3.2.6.3. Create Deals & Ads

Use Case ID	033
Use Case	Create deals & Ads.
Actors	Business.
Trigger	User wants to create an ad for their business.

Pre Conditions	Business is verified and is logged into their account.	
Post Conditions	The ad is running on the application.	
Basic Flow	**Actor**	**System**
	1. User opens the dashboard. 2. User goes to Deals & Ads. 3. User can view their total deals, active campaigns, pending approvals and total revenue generated from ads. 4. User searches the deals with filter and Status.	2.1 System displays all the information regarding the ads. 3.1 System fetches all data from the app. 4.1 System fetches the data and shows user deal they want to find.
Alternate Flow	**Actor**	**System**

Table 50. Create deals & Ads Use Case.

4.3.2.6.4. View Business Analytics

Use Case ID	034	
Use Case	View business analytics.	
Actors	Business.	
Trigger	User wants to see their sales, engagement etc.	
Pre Conditions	Business is verified and is logged into their account.	
Post Conditions	User is displayed with the charts and other data.	
Basic Flow	**Actor**	**System**
	1. User opens the dashboard. 2. User goes to analytics.	2.1 System shows all the details on monthly, weekly, yearly basis. 2.2 System shows the performance charts.
Alternate Flow	**Actor**	**System**

Table 51. View business analytics Use Case.

4.3.2.6.5. Respond & View Reviews

Use Case ID	035	
Use Case	Respond & View Reviews.	
Actors	Business.	
Trigger	User wants to view reviews of their business.	
Pre Conditions	Business is verified and is logged into their account.	
Post Conditions	The user is displayed with all the reviews of their business.	
Basic Flow	**Actor**	**System**
	1. User opens the Reviews & Reputation page. 2. User replies on one of the reviews. 3. User opens the analytics page of the reviews.	1.1 System shows all reviews for their business only. 2.1 System saved the reply. 3.1 System shows the charts for different trends and top keywords.
Alternate Flow	**Actor**	**System**

Table 52. Respond & View Reviews Use Case.

4.3.2.7. Admin

4.3.2.7.1. View Businesses

Use Case ID	036	
Use Case	View businesses.	
Actors	Admin.	
Trigger	Admin wants to view all users and businesses registered.	
Pre Conditions	Admin is logged into their account.	
Post Conditions	Admin is shown all the users and businesses registered.	
Basic Flow	**Actor**	**System**
	1. Admin navigates to the 'Business owner' page. 2. User views all businesses successfully.	1.1 System fetches the data of businesses registered.
Alternate Flow	**Actor**	**System**

	2a. Admin applies filters 1. Admin applies filters at the search bar of businesses. 2. System views the businesses successfully.	 1.1 System refine the results according to the filter.

Table 53. View Businesses Use Case.

4.3.2.7.2. Manage Businesses

Use Case ID	037	
Use Case	Manage Businesses.	
Actors	Admin.	
Trigger	Admin have to verify all the business registration requests.	
Pre Conditions	Business verification request is notified to admin.	
Post Conditions	The business request is approved/disapproved.	
Basic Flow	**Actor**	**System**
	1. Admin navigates to the 'Business owner' page. 2. Admin searches business by pending verification category. 3. Admin finds their information and all other documents completes, admin approves the business.	 2.1 System refines the search and provides the businesses with pending verification. 3.1 System sends the approval mail to the business.
Alternate Flow	**Actor**	**System**
	3a. Admin disapprove the request. 1. Admin view all the details of the business. 2. The registration documents of the business are fake and other details are also not seems good. Admin rejects the request.	 2.1 System sends the rejection mail to the business.

Table 54. Manage Businesses Use Case.

4.3.2.7.3. Manage reviews

Use Case ID	038	
Use Case	Manage reviews.	
Actors	Admin.	
Trigger	Admin wants to moderate reviews.	
Pre Conditions	Admin is logged into their account.	
Post Conditions	The reviews are moderated.	
Basic Flow	**Actor**	**System**
	1. Admin navigates to the 'Reviews & Content' page. 2. Admin searches review using filter. 3. Admin approves the review.	1.1 System shows all the details and reviews. 2.1 System refines the search. 3.1 System saves the changing and review is live on the application.
Alternate Flow	**Actor**	**System**
	3a. Admin flagged a review. 1. A review gets 3-4 reports for which admin flags it. **3b.** Admin removes a review. 1. Admin finds a review with a lot of reports as well as the content is against policy, Admin removes the review.	1.1 System takes that review under supervision. 1.1 System sends the user community guideline warning.

Table 55. Manage Reviews Use Case.

4.3.2.7.4. View system analytics

Use Case ID	039	
Use Case	View system analytics.	
Actors	Admin.	
Trigger	Admin wants to view analytics of the system.	
Pre Conditions	Admin is logged into their account.	
Post Conditions	The system analytics are displayed.	
Basic Flow	**Actor**	**System**
	1. Admin navigates to 'Analytics'	1.1 System loads all information &

	page. 2. Admin views analytics in form of charts succesfully.	metrics.
Alternate Flow	**Actor**	**System**

Table 56. View System Analytics Use Case.

4.3.2.7.5. Manage reports & complaints

Use Case ID	040	
Use Case	Manage reports & complaints.	
Actors	Admin.	
Trigger	Admin have to manage all the reports and complaints.	
Pre Conditions	All the reports and complaints are notified to admin.	
Post Conditions	Admin resolves the reports and complaints.	
Basic Flow	**Actor**	**System**
	1. Admin opens the 'Reports & Moderation' page. 2. Admin clicks on the 'Complaints' section. 3. Admin approves the request and provide them the solution by mailing them.	2.1 System shows all the complaints and reports. 3.1 System saves the changing.
Alternate Flow	**Actor**	**System**
	3a. Admin rejects the request.	

Table 57. Manage reports & complaints Use Case.

4.3.3. Activity Diagrams

4.3.3.1. Business Registration

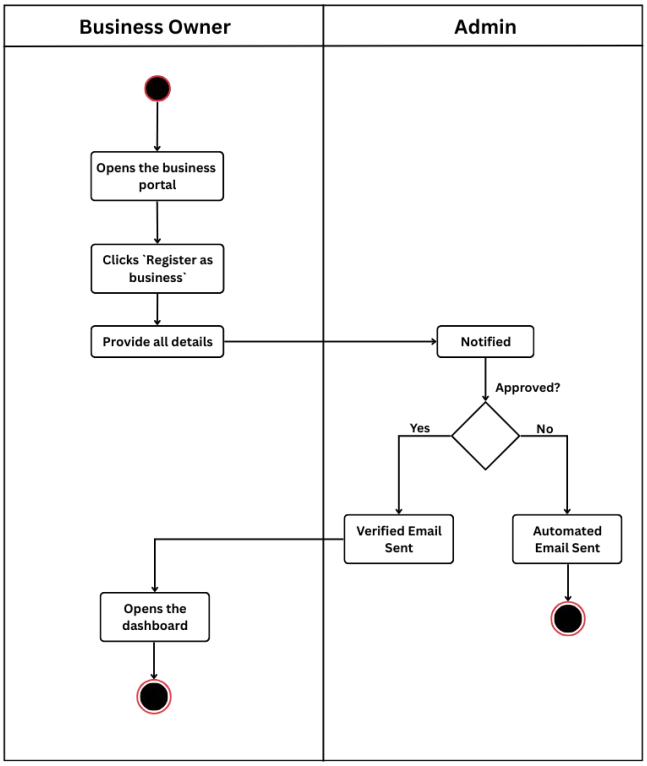

Figure 9. Business Registration Activity Diagram

4.3.3.2. Recommendation & Discovery

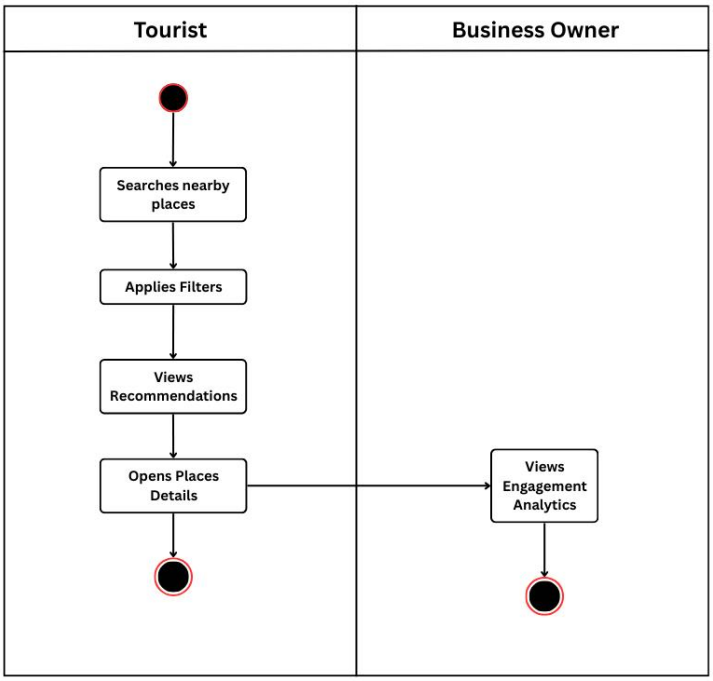

Figure 10. Recommendation & Discovery Activity Diagram

4.3.3.3. Reviews

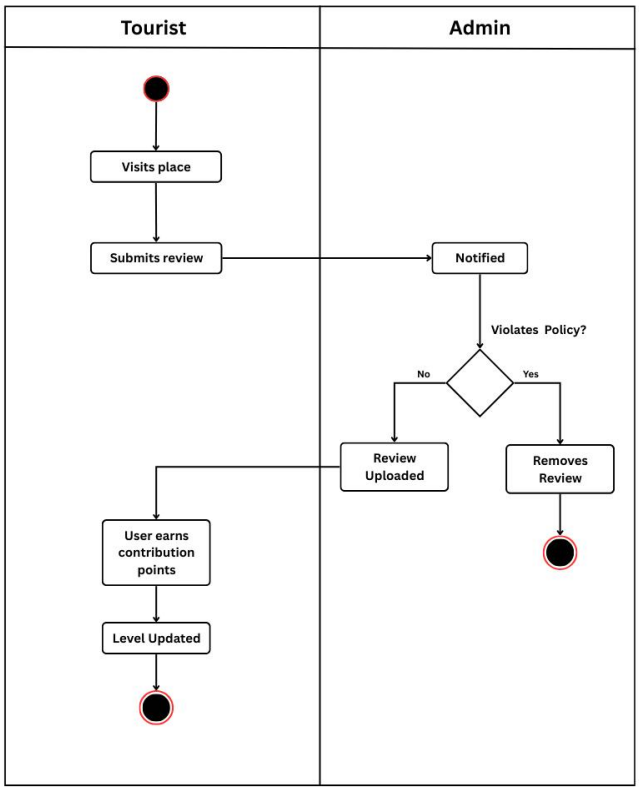

Figure 11. Reviews Activity Diagram

4.3.3.4. Trip Planning & Budget

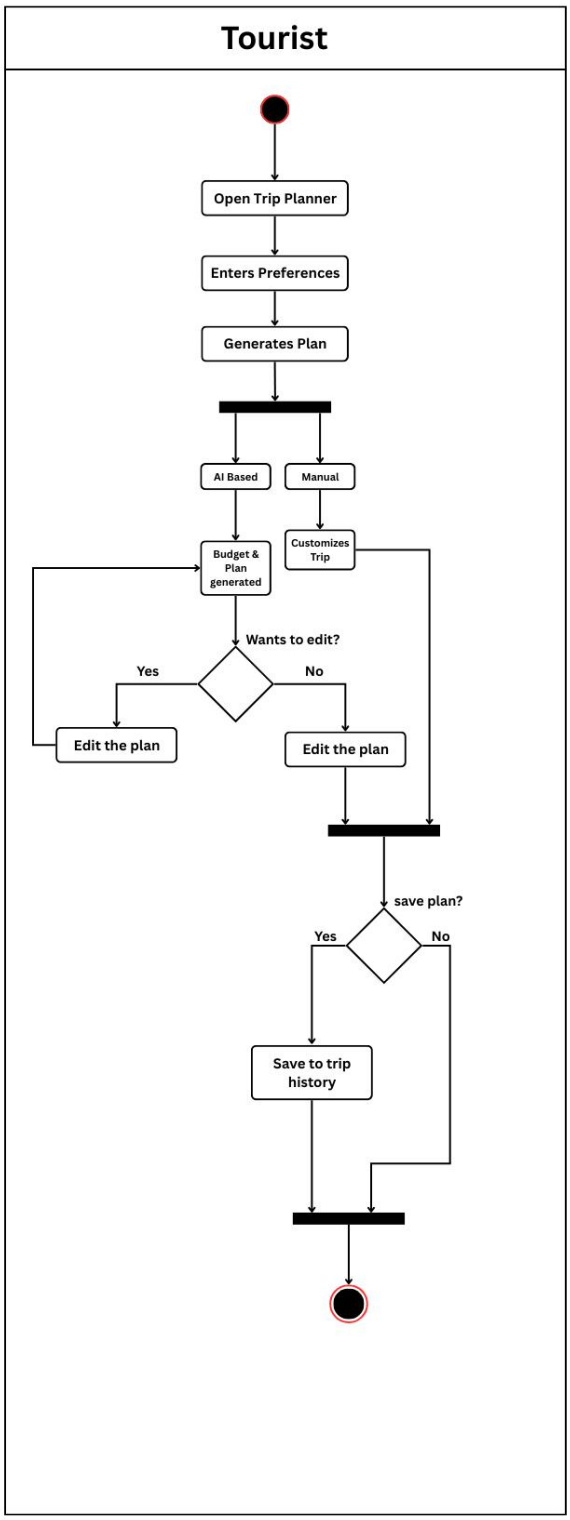

Figure 12. Trip Planning & Budget Activity Diagram

4.3.3.5. Ads & Deals

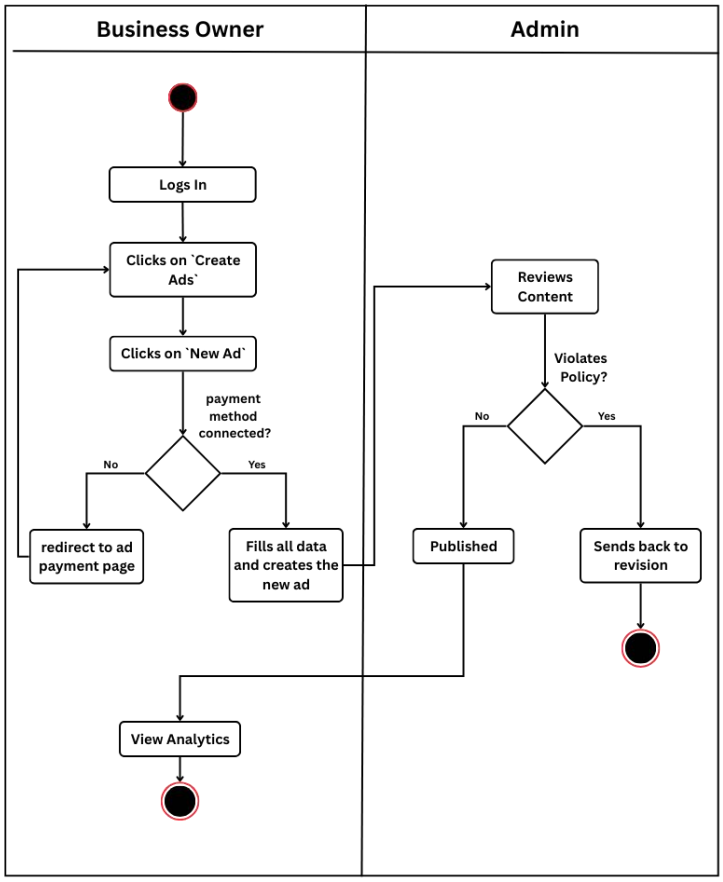

Figure 13. Ads & Deals Activity Diagram

4.4. Database Design

4.4.1 Overview

In considering that WanderPlan is being built as a modern, scalable web and mobile platform a NoSQL document-based database (MongoDB) is selected to follow where a relational SQL database absolutely leads. With MongoDB the data is not kept in tables as in the case of SQL but in collections and every entry is saved as a document similar to JSON. This gives flexibility, speed, and the capacity to process easily semi-structured data such as user-generated reviews, media, and business profiles, which are the major types of data MongoDB is mostly used for.

Node.js + Express.js work perfectly together with MongoDB, which is the backend technology of WanderPlan. The models are built with Mongoose, which does data

structure enforcement, rules validation, and relationships among collections in NoSQL-like.

There is a one-to-many relationship between WanderPlan modules and collections. Right now the following modules are active:

- Tourists Accounts.
- Business Profiles & Verification.
- Reviews & Ratings.
- Admin Panel.
- Email Verification & Security Tokens

Considering the implementation method used, these collections are indirectly connected via ObjectId references. In other words, instead of performing joins, the documents point to one another directly.

4.4.2 Detailed Collection Schemas

4.4.2.1 Tourist Collection

Field	Type	Description	Constraint
userId	ObjectId	Unique identifier of the user account	Auto-generated
fullName	String	User's full name	Required
email	String	Login email	Required; Unique
password	String	Hashed password	Required
phoneNumber	String	Optional phone	Required
country	String	Country of residence	Required
city	String	City	Required
gender	String	Gender Value	Optional
otp	Number	OTP during verification	Required
createdAt	Date	Account creation time	Auto
updatedAt	Date	Modification timestamp	Auto

Table 58. Tourist Collection Schema.

4.4.2.2 Business Collection

Field	Type	Description	Constraint
businessId	ObjectId	Unique Identifier of the business account	Auto-generated
businessName	String	Registered business name	Required
email	String	Business login email	Required; Unique
password	String	Hashed password	Required
phone	String	Contact number	Required
businessType	String	E.g. Restaurant, Hotel, Travel.	Required
Address	String	Business Location	Required
country	String	Country	Required
city	String	City	Required
description	String	Business information	Optional
verified	Boolean	Approval status	Default; false
Status	String	Active / Disabled	Enum
createdAt	Date	Account creation time	Auto
documents	Array[]	Uploaded verification files	Required

Table 59. Business Collection Schema.

4.4.2.3 Reviews Collection

Field	Type	Description	Constraint
userId	ObjectId	Who posted review	Required
businessId	ObjectId	Business reviewed	Required
rating	Number	Rating 1 - 5	Required
comment	String	Written feedback	Optional
images	Array[]	Review images	Optional
Likes	Array[]	Users who liked review	Optional
createdAt	Date	Timestamp	Auto

Table 60. Reviews Collection Schema.

4.4.2.4 Admin Collection

Field	Type	Description	Constraint
_id	ObjectId	Unique identifier of the user account	Auto-generated
email	String	Admin login email	Required
password	String	Encrypted password	Required

Table 61. Admin Collection Schema.

4.4.2.5 Email verification tokens collection

Field	Type	Description
owner	ObjectId	Linked account
token	String (hashed)	Verification token
createdAt	Date	Expiry logic applied

Table 62. Email Verification tokens collection Schema.

4.4.3 Justification for choosing MongoDB

The decision was made to go with MongoDB for the following reasons:

- User Generated Content that can be supported scaling.
- No constraints on schema.
- Storage in JSON format natively.
- Faster development cycle with iterations.
- Compatibility with Node.js.

NoSQL doesn't enforce strict relationships as Rational Database Management System do, thus traditional ERD diagrams were not used. Schema models and relationships are handled through Mongoose instead.

4.4.4 Summary

WanderPlan's database design is organized in such a way that it can easily grow, perform fast read/write operations and be expanded in a modular way. Mongoose's schema-based validation guarantees data consistency and retains NoSQL flexibility at the same time.

Chapter: 5

IMPLEMENTATION

Chapter 5

IMPLEMENTATION

5.1. Endeavor (Team Work + Work + Way of Working)

5.1.1. Team

- Abdullah Ahtsham 45679
- M. Nabeel Awais 43944
- M. Bilal Butt 44712

5.1.2. Work Breakdown Structure

5.1.2.1 Outline Form

5.1.2.1.1 FYP Report

5.1.2.1.1.1 Introduction

5.1.2.1.1.2 Literature / Market Survey

5.1.2.1.1.3 Requirement Engineering

5.1.2.1.1.4 System Design

5.1.2.1.1.5 Implementation

5.1.2.1.1.6 Testing & Evaluation

5.1.2.1.1.7 Conclusion & Outlook

5.1.2.1.2 Development

5.1.2.1.2.1 Database Implementation

5.1.2.1.2.2 Admin Web Portal

5.1.2.1.2.3 Business Web Portal

5.1.2.1.2.4 Mobile App

5.1.2.1.2.5 Server

5.1.2.1.2.5.1 Mobile App APIs

5.1.2.1.2.5.2 Business Web APIs

5.1.2.1.2.5.3 Admin Web APIs

5.1.2.1.2.6 SQA

5.1.2.1.2.6.1 Black Box

5.1.2.1.2.6.2 White Box

5.1.2.1.3 Presentations

5.1.2.1.3.1 Proposal Presentation

5.1.2.1.3.2 Progress Report Presentation

5.1.2.1.3.3 Final Demo Presentation

5.1.2.1.4 Project Management

5.1.2.1.4.1 WBS (Work Breakdown Structure)

5.1.2.1.4.2 Roles & Responsibility Matrix

5.1.2.1.4.3 GitHub Repositories

5.1.2.1.5 Training

5.1.2.1.5.1 JavaScript

5.1.2.1.5.2 Node

5.1.2.1.5.3 Express

5.1.2.1.5.4 React

5.1.2.1.5.5 React Native

5.1.2.1.5.6 MongoDB

5.1.2.1.5.7 GitHub

5.1.2.1.5.8 Documentation Writing

5.1.2.2 Diagram Form

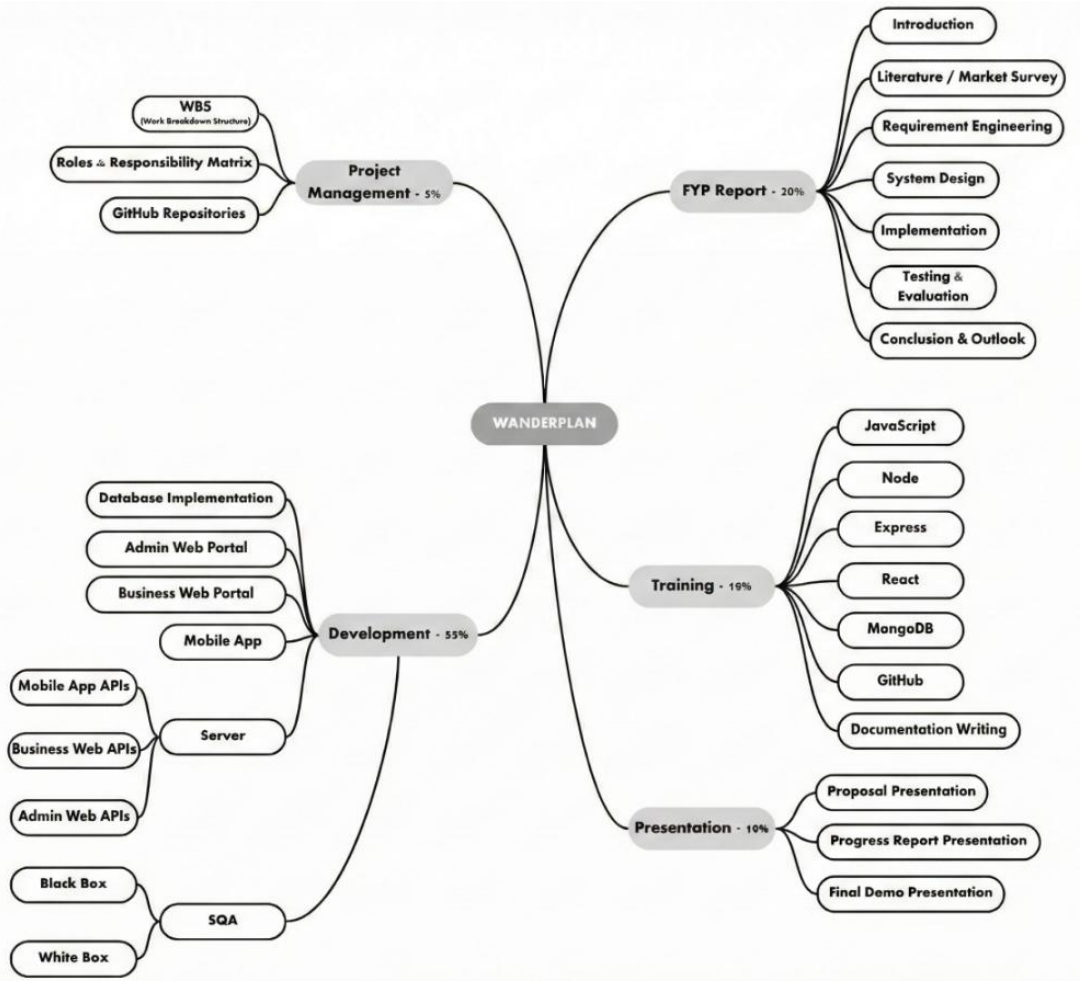

Figure 14. Work Breakdown Structure

5.1.3. Roles & Responsibility Matrix

WBS Name	Activity	**Roles** (**R**esponsible, **A**ccountable, **C**onsulted, **I**nformed)		
		Abdullah Ahtsham	**M. Bilal Butt**	**M.Nabeel Awais**
Development	Maintain GitHub Repositories	**R**	**R, A**	**R**
	Develop wanderplan server.	**C**	**R, A**	**R**
	Develop wanderplan admin.	**R**	**C**	**R, A**
	Develop wanderplan mobile	**C**	**R, A**	**C**
	Develop wanderplan business.	**R**	**C**	**R, A**
	Testing Code (Whitebox + Blackbox)	**R, A**	**R**	**R**
	Implement Database	**C**	**R, A**	**C**
	Make wireframes	**R, A**	**C**	**C**
FYP Reports / Documentation	Write Introduction, Market Survey	**R, A**	**I**	**I**
	Maintain Functional Requirements.	**R, A**	**C**	**C**
	Maintain Non-Functional Requirements.	**R, A**	**C**	**C**
	Make Use Cases, Fully Dressed Use Cases.	**R, A**	**C**	**R**
	Make Architecture, Deployment Diagrams.	**R, A**	**R**	**I**
	Maintain database design.	**R**	**R, A**	**I**
	Make Activity Diagram.	**R, A**	**C**	**C**
	Make Test Case Designs.	**R, A**	**C**	**C**
	Maintain WBS, Roles & Responsibility, Meeting Records.	**R, A**	**R**	**I**
Training	Learn MERN STACK.	**R**	**R, A**	**R**

	Learn how to use Tools & Technologies (Github, Version Control, Android Studio)	**R**	**R, A**	**R**
Presentations	Make Project Proposal Presentation	**A**	**C**	**R**
	Make Progress Report Presentation	**R, A**	**C**	**C**
	Make Final Demo Presentation	**R, A**	**R**	**C**

Table 63. Roles & Responsibility Matrix

5.1.4. Way of Working

5.1.4.1. Abdullah Ahtsham - Documentation, UI/UX & Wireframes, Testing.

5.1.4.2. M. Bilal Butt - Mobile Application, Backend Server & Database.

5.1.4.3. M. Nabeel Awais - Business Panel, Admin Panel.

5.2 IDE, Tools & Technologies

5.2.1 Methodology

For WanderPlan, we followed an iterative and incremental development approach, inspired by Agile practices. Since FYP projects evolve over time and requirements become clearer as development progresses, this approach suited us well. Work was divided into short development cycles where we planned, implemented, tested, and reviewed progress continuously.

This helped us:

- Break the project into manageable milestones
- Prioritize core features first (Authentication, Reviews, Business & Admin Panels)
- Adjust designs based on feedback from the supervisor
- Reduce risk of major re-work later
- Ensure working modules existed at every stage

Although our team did not formally run Scrum sprints, our process was light-weight Agile, allowing flexibility while maintaining structure and progress visibility.

5.2.2 Integrated Development Environment (IDE)

Different components of WanderPlan were developed using different IDEs suited to the platform:

5.2.2.1 Mobile App Development

We used Visual Studio Code (VS Code) for developing the mobile application using React-Native. It provided:

Lightweight and fast environment

- Built-in terminal
- Strong plugin ecosystem
- Auto-code formatting
- Syntax highlighting & debugging tools

5.2.2.2 Web/Admin Panel Development

The Business Web Dashboard and Admin Panel were also developed in VS Code, since it works exceptionally well with JavaScript and React-based applications.

5.2.2.3 Backend Development

Backend APIs were built using Node.js & Express inside VS Code with tools like:

- REST Client / Thunder Client / Postman
- Nodemon for automatic restart
- ESLint formatter

5.2.2.4 Database Management

We used MongoDB Atlas along with MongoDB Compass (GUI Viewer & Query Tool)

This helped us visualize collections, indexes, documents, and relations.

5.2.3 Tools

The following tools supported development and project management:

- Visual Studio Code
- Android Studio
- Figma
- MongoDB Atlas
- Canva
- WPS Office
- Expo
- Github (Desktop + Web)

5.2.4 Technologies

Following table shows the technologies used in our project till date.

Layer	Technology Used
Mobile Application	React Native
Business Panel (Web)	React.js
Admin Panel (Web)	React.js
Backend	Node.js + Express.js
Database	MongoDB (NoSQL, Cloud-Hosted)
API Protocol	RESTful APIs
Authentication	JWT (JSON Web Tokens)
Package Manager	npm
Version Control	GitHub

Table 64. Technologies Used.

5.2.5 Stack

The WanderPlan project uses a mobile-oriented modified MERN stack:

5.2.5.1 Frontend

- React-Native Mobile Application
- React.js Web Business Dashboard

- React.js Admin Panel

5.2.5.2 Backend

- Node.js
- Express.js REST APIs

5.2.5.3 Database

- MongoDB (NoSQL Database)

5.2.5.4 Authentication

- Secure login with JWT

Hence, the system can be easily made scalable, modular, and extended.

5.2.6 Justifications

The technologies were chosen considering these factors:

5.2.6.1 Scalability

MongoDB is the best option to keep unstructured travel data since it can store it very well

5.2.6.2 Cross-Platform Development

React-Native is the most efficient approach that we can use to develop Android and iOS apps at the same time with one codebase

5.2.6.3 Fast Development

JavaScript becomes the major language not only for frontend but also for backend development

5.2.6.4 Large Community Support

There is an abundance of learning resources and support for troubleshooting

5.2.6.5 Cloud-Based Database

MongoDB Atlas is a provider that guarantees the safety and trustworthiness of data

5.2.6.6 REST APIs Enable Integration

The system is open for the future like AI modules, payment systems and mapping services etc.

In summary, the stack is a big win for rapid development & future scaling (Recommendation engine, business automation).

5.3 Best Practices / Coding Stardards

To maintain code quality and consistency across the team, we followed these standards:

5.3.1 General Coding Practices

- Meaningful and readable variable names
- Code indentation and formatting
- Reusable components and functions
- Separation of concerns (UI, Logic, API calls kept modular)

5.3.2 JavaScript / React Standards

- Functional components preferred over class components
- Hooks used where applicable
- API logic kept separate from UI logic
- Use of constants for routes and configs

5.4 Deployment Environment

5.4.1 Deployment Diagram

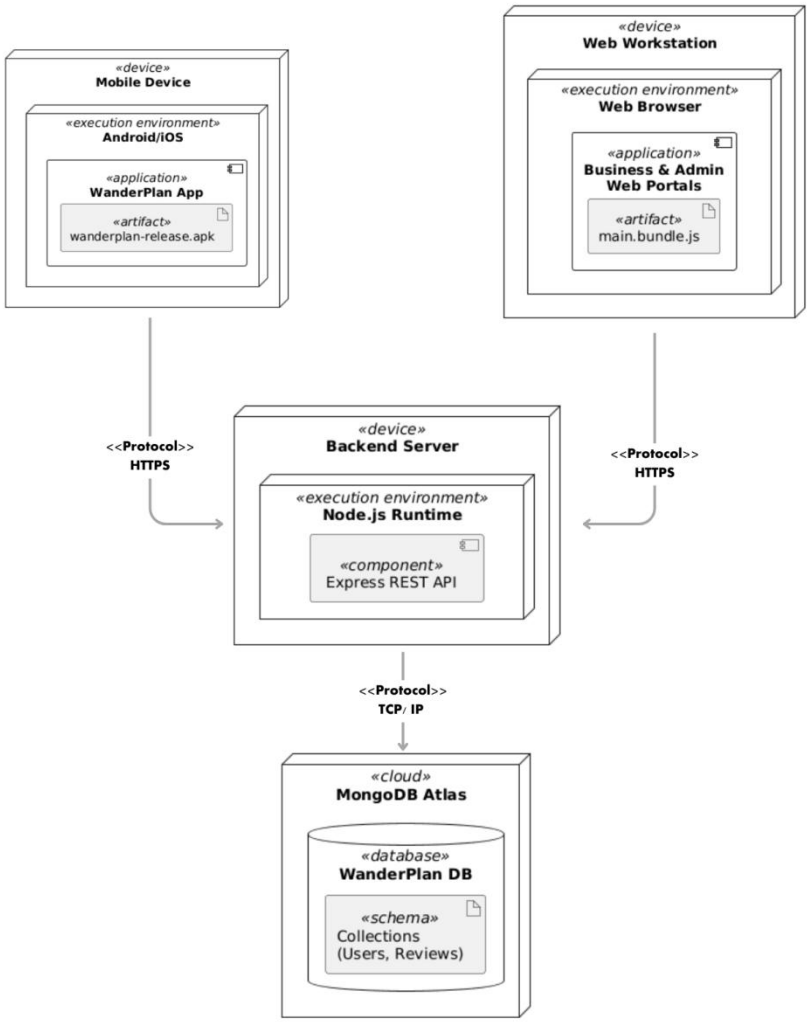

Figure 15. Deployment Diagram

Chapter: 6

TESTING & EVALUATION

Chapter: 7

CONCLUSION & OUTLOOK