HIKERNET - COMMUNITY TREKKING COMPANION

A PROJECT REPORT

submitted by

ANSHIDA SHIRIN P (WYD20CS023)
JERIN THOMAS (WYD20CS036)
LEZIN SAJID (WYD20CS037)

to

The APJ Abdul Kalam Technological University
in partial fulfilment of the requirements for the award of the Degree

of

Bachelor of Technology
in
Computer Science and Engineering

Department of Computer Science and Engineering

Government Engineering College, Wayanad
Thalappuzha – 670644

OCTOBER, 2025

DECLARATION

I, on behalf of authors of the report: Anshida Shirin, Jerin Thomas, Lezin Sajid, hereby declare that the project report HIKERNET - COMMUNITY TREKKING COMPANION submitted for partial fulfilment of the requirements for the award of degree of Bachelor of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by us under supervision of Prof. Hasna M (Assistant Professor, CSE Department). This submission represents our ideas in our own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also invoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Thalappuzha Lezin Sajid

Date: 23-10-2025

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
GOVERNMENT ENGINEERING COLLEGE, WAYANAD
THALAPPUZHA – 670644

CERTIFICATE

This is to certify that the report entitled **HIKERNET - COMMUNITY TREKKING COMPANION** submitted by **Anshida Shirin P, Jerin Thomas, Lezin Sajid** to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the Degree of Bachelor Technology in Computer Science and Engineering is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Project Supervisor(s)

Department Seal

Project Coordinator(s) Head of the Department

ACKNOWLEDGEMENTS

We greatly indebted to God Almighty for being the guiding light throughout with his abundant grace and blessings that strengthened us to do this endeavour with confidence. We would like to acknowledge , Prof. Hasna M, (Assistant Professor, Dept. of Computer Science and Engineering), Govt. Engineering College, Wayanad, for inspiring and providing sincere guidance throughout the project. We also express our sincere thanks to the project coordinators Prof. Neevan R (Assistant Professor, Dept. of Computer Science and Engineering), Prof. Manu Ratheesh (Assistant Professor, Dept. of Computer Science and Engineering) for their valuable suggestions and constant encouragements. We extremely grateful to Dr. Shibily Joseph, Head of the Department, Computer Science and Engineering, Govt. Engineering College, Wayanad for her wise guidance and supportive attitude. We would also like to express our sincere gratitude to each member of Dept. of Computer Science and Engineering, Govt. Engineering College, Wayanad, for their kind co-operation and encouragement that helped us in completing the project. We would like to acknowledge Dr. Pradeep V, Principal of Govt. Engineering College, Wayanad for providing the facilities. We would also like to extend my gratitude to all wellwishers and friends who supported us to present the project.

ABSTRACT

HikerNet is a community-driven trekking companion application designed to integrate essential trail utilities with interactive social features. The platform enables trekkers to safely discover, record, and share adventure experiences through secure GPS trail tracking, dynamic group coordination, and automated rendezvous point suggestions. HikerNet supports both solo and group treks, offering modules for real-time location sharing, offline SOS alerts, and weather integration to enhance safety and preparedness. User profiles, photo-tagging, posts, and activity logs contribute to a vibrant and engaging community environment. By bridging the gap between navigation tools and social networking, HikerNet fosters coordinated, safe, and memorable outdoor exploration while encouraging meaningful connections among adventurers.

CONTENTS

Contents Page No.

ACKNOWLEDGEMENTS i
ABSTRACT ii
LIST OF FIGURES v
CHAPTER 1. INTRODUCTION 1
1.1 Problem Description 1
CHAPTER 2. LITERATURE SURVEY 3
2.1 AllTrails 3
2.1.1 Intended Users 3
2.1.2 Functionalities 3
2.1.3 Techniques Used 3
2.1.4 Merits 4
2.1.5 Drawbacks 4
2.2 Strava 4
2.2.1 Intended Users 4
2.2.2 Functionalities 4
2.2.3 Techniques Used 5
2.2.4 Merits 5
2.2.5 Drawbacks 5
2.3 Komoot 6
2.3.1 Intended Users 6
2.3.2 Functionalities 6
2.3.3 Techniques Used 6
2.3.4 Merits 6
2.3.5 Drawbacks 7
2.4 Lessons from A Large Language Model-based Outdoor Trail Recommendation Chatbot with Retrieval Augmented Generation (2024) 7
2.4.1 Intended Users 7
2.4.2 Functionalities 7
2.4.3 Techniques Used 7
2.4.4 Merits 8
2.4.5 Drawbacks 8
2.5 Emergency Response System: Design and Development of an SOS Mobile Application (IJSREM, 2025) 8
2.5.1 Intended Users 8
2.5.2 Functionalities 8
2.5.3 Techniques Used 9

2.5.4 Merits 9
2.5.5 Drawbacks 9
2.6 Summary of Existing Works 10
2.7 Motivation for the project proposal 11
2.8 Problem Statement 11
CHAPTER 3. REQUIREMENT ANALYSIS 12
3.1 Introduction 12
3.1.1 Purpose 12
3.1.2 Definitions and Acronyms 12
3.2 Overall Description 13
3.2.1 User Needs 13
3.2.2 Assumptions and Dependencies 15
3.3 Method of Requirement Elicitation 16
3.3.1 Brainstorming 16
3.3.2 Survey and Analysis 17
3.4 System Features and Requirements 17
3.4.1 Functional Requirements 17
3.4.2 External Interface Requirements 19
3.4.3 Nonfunctional Requirements 20
3.4.4 Hardware Requirements 20
3.4.5 Software Requirements 20
3.5 Process Model 21
CHAPTER 4. FEASIBILITY STUDY 23
4.1 Economic Feasibility 23
4.1.1 Estimated Hardware Cost 23
4.1.2 Estimated Software Cost 23
4.1.3 Conclusion 23
4.2 Technical Feasibility 23
4.2.1 Product Requirements 23
4.2.2 Software Requirements 24
4.2.3 Development Requirements 24
4.2.4 Skill Requirements 24
4.3 Operational Feasibility 25
4.3.1 Throughput and Response Time 25
4.3.2 Timeliness and Data Accuracy 25
4.3.3 Resource Utilization 25
4.3.4 Reliability 26
4.3.5 Scalability and Flexibility 26
4.3.6 Compatibility With New System 26
4.3.7 User Adoption 26

4.4 Scheduling Feasibility 26
CHAPTER 5. ARCHITECTURAL DESIGN 27
5.1 Module Descriptions 28
5.1.1 User Management 28
5.1.2 Trek Management 28
5.1.3 Rendezvous Engine 28
5.1.4 Live Location Sharing 29
5.1.5 Safety and Alerts (SOS) 29
5.1.6 Community Interaction (Social Feed) 29
5.1.7 Reports and Analytics 30
5.1.8 Admin Management 30
5.1.9 Notification System 30
5.1.10 Leaderboards and Achievements 31
5.1.11 Supporting Services 31
CHAPTER 6. DESIGN AND IMPLEMENTATION 32
6.1 Introduction 32
6.2 Data Flow Diagrams 32
6.2.1 Level 0 DFD 32
6.2.2 Level 1 DFD 33
6.2.3 Level 2 DFD 34
6.2.4 Trek Features DFD 35
6.2.5 Community Features DFD 36
6.3 Work Plan and Task Allocation 37
REFERENCES 38

LIST OF FIGURES

No.	Title	Page No.
3.1	Agile method	22
5.1	System Architecture	27
6.1	Level 0 Data Flow Diagram	32
6.2	Level 1 Data Flow Diagram	33
6.3	Level 2 Data Flow Diagram - Trek Features	35
6.4	Level 2 Data Flow Diagram - Community Features	36
6.5	Workplan and Task Allocation	37

CHAPTER 1
INTRODUCTION

Trekking has surged in popularity as an outdoor recreational activity that combines physical exercise with nature exploration and social engagement. However, trekkers often face challenges related to navigation, safety, and coordination during their outings. The absence of a unified platform that integrates real-time location sharing, reliable trail recording, dynamic group management, and community interaction limits optimal trekking experiences. HikerNet aims to bridge this gap by offering a comprehensive mobile application that serves as both a trekking companion and a social network.

The platform provides trekkers with tools to record GPS trails, suggest rendezvous points dynamically, and share live location data with fellow adventurers. Beyond navigation, HikerNet fosters a supportive community environment where users can create profiles, share posts and photos, participate in leaderboards, and receive timely notifications and safety alerts. Integration with weather and map APIs enhances preparedness while offline SOS functionality ensures critical help is available when needed. This holistic approach motivates safer, more coordinated, and socially enriching trekking activities.

1.1 PROBLEM DESCRIPTION

Current trekking applications primarily focus on either navigation or social features in isolation, rarely offering an integrated experience. Trekking safely in unfamiliar locations requires coordinating group movements, tracking trails accurately, and maintaining real-time communication—all of which remain cumbersome with existing solutions.

Furthermore, emergency situations demand immediate alerts and quick response mechanisms, but few apps provide offline SOS features or seamless emergency notifications.

Additionally, the lack of a dedicated community platform for trekkers slows down experience sharing, reduces motivation, and impedes information exchange about trail conditions or safety. Many online communities for outdoor enthusiasts are dispersed across generic social media, lacking trekking-specific tools and data integration. This fragmentation not only compromises user safety but also diminishes the overall enjoyment of trekking activities.

HikerNet addresses these issues by integrating essential trekking functionalities with a vibrant and tailored social networking environment. It aims to improve safety, coordination, and user engagement through real-time data sharing, group rendezvous computation, and proactive alerting services. The platform thus enhances trekking experiences for individuals and groups alike, creating a reliable digital companion for every trekker.

CHAPTER 2
LITERATURE SURVEY

2.1 ALLTRAILS [1]

2.1.1 Intended Users

- AllTrails is designed for hikers, trekkers, cyclists, and outdoor enthusiasts seeking robust trail navigation and information.

2.1.2 Functionalities

- The app provides trail maps, elevation data, and difficulty ratings for planning outdoor activities.
- It features GPS-based tracking with offline map support, ensuring usability even in areas without network coverage.
- Users can share their reviews and trail experiences on a community platform, building a comprehensive trail database.

2.1.3 Techniques Used

- AllTrails uses GPS-based tracking to monitor trail progress and user location.
- The app offers offline map capabilities to maintain navigation when connectivity is lost.
- User-generated content is leveraged for navigation and for improving trail recommendations.

2.1.4 Merits

- The large trail database covers diverse difficulty levels and regions.
- Real-time GPS tracking and offline map support enhance navigation reliability.
- User reviews and photos provide valuable insights for trail evaluation.
- The app supports multiple outdoor activities beyond just hiking.

2.1.5 Drawbacks

- Social interaction features are limited compared to other platforms.
- Some advanced features require a paid subscription.
- The platform lacks AI-based personalization in its trail recommendation engine.

2.2 STRAVA [2]

2.2.1 Intended Users

- Strava targets runners, cyclists, hikers, and fitness enthusiasts interested in performance tracking and community engagement.

2.2.2 Functionalities

- The application tracks GPS-based activity across multiple sports, supporting route planning and exploration.
- It offers detailed performance analytics including pace, elevation gain, and heart rate.
- Social networking features allow sharing activities, joining leaderboards, and participating in challenges.

2.2.3 Techniques Used

- Strava utilizes GPS tracking, mobile sensors, and integration with wearable devices for comprehensive data capture.
- Data analytics and AI are employed for route recommendations and performance insights.
- Community-driven datasets enhance route popularity scores and leaderboard rankings.

2.2.4 Merits

- The platform has strong social engagement features, including activity sharing and interaction.
- Leaderboards and competitive challenges motivate users to stay active.
- Integration with a wide range of fitness devices offers flexibility.
- Detailed analytics support comprehensive training and goal-setting.

2.2.5 Drawbacks

- Some advanced features, such as detailed analytics and route planning, require a paid subscription.
- Privacy concerns arise from public sharing of routes and personal performance data.
- There is less emphasis on hiking-specific community features compared to dedicated hiking apps.

2.3 KOMOOT [3]

2.3.1 Intended Users

- Komoot is aimed at hikers, cyclists, mountain bikers, and outdoor explorers desiring personalized routes and offline navigation.

2.3.2 Functionalities

- It provides detailed route planning by activity type, terrain, and difficulty.
- Turn-by-turn voice navigation and offline maps support outdoor adventures in remote areas.
- Users can explore, save, and share community-recommended highlights and routes.

2.3.3 Techniques Used

- The app employs GPS data and topographic mapping for accurate route calculations.
- Community content, reviews, and highlights enrich the user experience and discovery process.
- Komoot integrates with wearable devices and GPS trackers for seamless recording.

2.3.4 Merits

- Komoot offers advanced route customization to match user preferences like surface type, fitness level, and activity style.
- Strong offline navigation ensures guidance when connectivity drops.
- A rich database of highlights and scenic points improves adventure discovery.

- The platform supports hiking, cycling, mountain biking, running, and more.

2.3.5 Drawbacks

- Multi-region offline maps are only available via paid subscription.
- Its community engagement is less interactive than Strava's social approach.
- Relying on user-generated content can sometimes make trail details inconsistent.

2.4 LESSONS FROM A LARGE LANGUAGE MODEL-BASED OUTDOOR TRAIL RECOMMENDATION CHATBOT WITH RETRIEVAL AUGMENTED GENERATION (2024) [4]

2.4.1 Intended Users

- The system is built for hikers, outdoor enthusiasts, park authorities, and platform developers seeking personalized guidance.

2.4.2 Functionalities

- The chatbot provides tailored trail recommendations through conversational interaction.
- It utilizes a large language model and a database of trails and user reviews for personalized suggestions.

2.4.3 Techniques Used

- Data scraping and review processing fuel the recommendation system.
- Semantic search is achieved through Retrieval Augmented Generation (RAG).

- The system implements Llama3, LangChain, and FAISS for managing AI embeddings.

2.4.4 Merits

- Recommendations are accurate and rich in context, improving user trust.
- Fast response times are enabled by optimized embedding search.
- Interactive conversations distinguish it from static recommendation platforms.

2.4.5 Drawbacks

- Coverage is presently restricted to specific regions.
- The recommendation quality is dependent on the volume and quality of user reviews.
- High computational demands are required for the AI search components.

2.5 EMERGENCY RESPONSE SYSTEM: DESIGN AND DEVELOPMENT OF AN SOS MOBILE APPLICATION (IJSREM, 2025) [5]

2.5.1 Intended Users

- Target users include the general public, vulnerable groups, emergency responders, and individuals with medical conditions.

2.5.2 Functionalities

- The app enables real-time location sharing and SOS alerts via SMS and email, even offline.
- It provides multilingual accessibility and usable interface for a wide demographic.

- Medical data is stored for quick access, with the system able to compute the fastest emergency routes.

2.5.3 Techniques Used

- Built using Flutter and Firebase technology stack.
- Utilizes mapping and route computation algorithms for emergency navigation.
- Supports voice and gesture activation for sending alerts, including offline support.

2.5.4 Merits

- SOS alerts are functional in offline and low-connectivity scenarios.
- Multilingual support broadens the scope of application.
- Fast route optimization assists emergency responders in timely intervention.

2.5.5 Drawbacks

- Performance is affected by GPS and network limitations when offline.
- Medical data may be missing or incomplete for some users.
- Activation of alerts via voice or gesture may be unreliable depending on circumstances.

2.6 SUMMARY OF EXISTING WORKS

Platform	Merits	Drawbacks
AllTrails [1]	• Large trail database with diverse levels. • Real-time GPS tracking. • User reviews and photos for trail evaluation. • Supports multiple outdoor sports.	• Limited social features. • Some features are paid only. • Trail recommendations lack AI-based personalization.
Strava [2]	• Strong social engagement features. • Leaderboards and competitive challenges. • Integrates many wearables. • Detailed performance insights.	• Some advanced features are paid. • Privacy concerns (public routes and data). • Focus on performance, not hiking-specific features.
Komoot [3]	• Extensive route customization. • Strong offline navigation. • Rich database of scenic points. • Supports multiple sports.	• Multi-region offline maps are paid. • Less interactive community engagement. • Relies on user content, which may be inconsistent.
Lessons from A Large Language Model-based Outdoor Trail Recommendation Chatbot with Retrieval Augmented Generation (2024) [4]	• Accurate, context-rich recommendations. • Fast response through optimized embeddings. • Interactive conversation versus static platforms.	• Currently covers only specific regions (e.g., Connecticut). • Quality depends on review data. • Requires computational resources for AI search.
Emergency Response System: Design and Development of an SOS Mobile Application (IJSREM, 2025) [5]	• SOS alerts when offline or during emergencies. • Accessible through multiple languages. • Route optimization for emergencies.	• GPS/network limitations offline. • Medical data may be incomplete. • Voice/gesture activation reliability may vary.

Table 2.1: Summary of Existing Works

2.7 MOTIVATION FOR THE PROJECT PROPOSAL

Applications such as AllTrails, Strava, Komoot, Life360, and Wikiloc demonstrate clear user demand for trail discovery, activity tracking, and community sharing. However, these systems tend to optimize for one dimension at a time (navigation, fitness analytics, or social features) and seldom provide an integrated experience that jointly addresses navigation, safety, personalization, and community support. In practice, hikers still face fragmented workflows—planning in one app, safety in another, and community elsewhere—along with limited offline reliability and generic recommendations that do not reflect skill level, context, or risk.

The proposed system, HikerNet, aims to bridge this gap by unifying dependable navigation with offline readiness, real-time safety tracking, AI-driven personalized trail recommendations, and lightweight community features. This integrated approach is motivated by the need to move beyond siloed tooling and deliver a single, trustworthy application that enhances outdoor decision-making, reduces risk, and strengthens community engagement for hikers.

2.8 PROBLEM STATEMENT

To design and develop a community trekking application that enables trekkers to safely discover, record, and share adventure experiences by integrating social networking features, intelligent trail tracking, dynamic group coordination, dynamic rendezvous path finding and safety utilities—creating a reliable and engaging platform for empowered outdoor exploration.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 INTRODUCTION

HikerNet seeks to transform trekking experiences by providing a digital companion that seamlessly integrates route tracking, safety utilities, and interactive social features for outdoor lovers. This chapter details its requirement analysis, from user needs and system functionalities to technical and operational feasibility, ensuring that every aspect of the platform is designed for effective, safe, and engaging trekking.

3.1.1 Purpose

The primary purpose of this requirement analysis is to outline the functionality, interfaces, and constraints of the proposed HikerNet application. It sets the foundation for development by identifying what trekkers and community members expect from a comprehensive outdoor platform.

3.1.2 Definitions and Acronyms

1. SOS: Emergency alert feature for distress signaling.
2. Rendezvous Engine: Module for dynamic group meeting point calculation.
3. DB: Database (e.g., User DB, Trek DB).
4. UI/UX: User Interface/User Experience.
5. API: Application Programming Interface.

6. Leaderboards: Feature for tracking user scores and trek achievements.

3.2 OVERALL DESCRIPTION

3.2.1 User Needs

HikerNet is built to meet the diverse requirements of trekkers, group coordinators, and outdoor communities who rely on robust navigation, group safety, experience sharing, and engagement with the broader trekking ecosystem. The platform addresses core needs using reliable GPS tracking, offline support, weather and map integration, and a suite of community-driven communication tools.

1. Navigation, Safety, and Trail Management:

- Users need reliable trail tracking and navigation that works even in remote or offline settings.
- Group treks require dynamic rendezvous recommendations and live location sharing for effective coordination.
- Quick SOS alerts, emergency notifications, and weather integration are of high importance for trekking safety.

2. Social Connection and Community Engagement:

- Trekkers seek platforms to connect with fellow adventurers, friends, and local guide communities.

- Sharing trek updates, experiences, posts, and photos fosters both storytelling and belonging.
- In-app messaging, comments, and group formation are essential for real-time connection and support.

3. Information, Discovery, and Learning:

- Users value access to updated trail conditions, local news, and community event details relevant to their trekking interests.
- Content such as route tips, packing guides, and environmental education empowers users to make better decisions.
- The platform should promote eco-friendly practices and outdoor know-how.

4. Recognition, Motivation, and Gamification:

- Features like leaderboards, badges, and achievement tracking encourage healthy competition and sustained engagement.
- Trekkers want recognition for milestones such as treks completed, distances covered, and group activities organized.

5. Personalization and Control:

- Users expect profile customization and control over who can view their shared routes, photos, and activity logs.

- Notification settings and location sharing preferences need to be user-configurable.

6. Localized and Offline Features:

- Access to local businesses, recommended trails, events, and highlights tailored to user location.
- Mapping, SOS alerts, and basic functions should work offline for reliability in remote terrain.

3.2.2 Assumptions and Dependencies

Assumptions:

1. Users will actively participate by recording and sharing treks, photos, and contributing to the community.
2. The server infrastructure can scale to accommodate user growth and peak usage.
3. Security mechanisms are in place to safeguard user location, profile, and trek data.
4. Users adhere to community guidelines and respectful conduct in group and community spaces.
5. The interface is mobile-optimized as a majority of users trek with smartphones.
6. External services and APIs (maps, weather) will remain available and reliable.
7. Adoption will reach a critical mass, sustaining content and interactions across regions.

8. Users will trust the platform with basic personal and trek-related information, given transparent privacy policies.

Dependencies:

1. Reliable internet connectivity for social and map updates, though core features offer offline capability.
2. Third-party APIs and services for maps, weather, authentication, and notifications.
3. Compliance with regional and international data privacy and outdoor safety regulations.
4. Stable and scalable cloud/server backend for hosting user data and enabling real-time group communication.
5. Active community moderation to prevent spam and maintain safety.
6. Availability of development resources for support and feature evolution.

3.3 METHOD OF REQUIREMENT ELICITATION

3.3.1 Brainstorming

The HikerNet project team organized collaborative brainstorming sessions to identify and prioritize essential features for trekking and community engagement. Team members drew on personal trekking experience, competitor app reviews, and group safety needs to generate concepts for GPS tracking, live location sharing, rendezvous point selection, offline SOS alerts, and in-app social tools. Ideas were evaluated for technical feasibility, practicality in outdoor scenarios, and contribution to a secure, connected trekking environment. The strongest feature concepts formed the basis for further system design and module development.

3.3.2 Survey and Analysis

To ensure HikerNet aligns with real user needs, the team conducted surveys and informal interviews with active trekkers, hiking group leaders, and outdoor community members. This analysis provided insight into common pain points and desired functionality, such as offline reliability, route recording, coordinated group movement, emergency support, and community interaction (posts, comments, leaderboards). The team also analyzed existing trekking and social platforms to benchmark strengths and weaknesses. The findings from user feedback and comparative analysis directly influenced the formulation of functional and nonfunctional requirements for HikerNet.

3.4 SYSTEM FEATURES AND REQUIREMENTS

3.4.1 Functional Requirements

1. Use Case: User Registration and Login
    - Precondition: The user has a stable internet connection.
    - Trigger: The user taps the "Register" or "Login" button.
    - Post-condition: The user successfully accesses their profile and homepage.
2. Use Case: GPS Trail Recording and Navigation

- Precondition: The user has enabled location services.
- Trigger: The user starts a trek session.
- Post-condition: The trail is recorded and saved locally/offline.

3. Use Case: Group and Solo Trek Management

   - Precondition: The user creates or joins a trek group.
   - Trigger: The user initiates or joins a trek.
   - Post-condition: Group members receive rendezvous point suggestions and live location updates.

4. Use Case: Emergency SOS Alerting

   - Precondition: Users register emergency contacts.
   - Trigger: The user triggers an SOS alert manually or automatically detects emergency.
   - Post-condition: SOS alert with location is sent to contacts even if offline.

5. Use Case: Social Feed and Content Sharing

   - Precondition: The user is logged in.
   - Trigger: The user posts updates, photos, or comments.
   - Post-condition: The content is shared to connected users and visible in feeds.

6. Use Case: Live Messaging and Notifications

   - Precondition: The user is connected to other users.

- Trigger: The user sends a text, voice message, or receives system notifications.
- Post-condition: Messages are delivered in real time; notifications alert users to activity.

7. Use Case: Leaderboard and Achievement Tracking
   - Precondition: The user has completed trekking activities.
   - Trigger: System computes scores and ranks user on leaderboards.
   - Post-condition: User can view achievements and compare with peers.
8. Use Case: Map and Weather API Integration
   - Precondition: The device is online or has preloaded map data.
   - Trigger: The user opens the navigation or weather section.
   - Post-condition: User views up-to-date maps and weather forecasts.

3.4.2 External Interface Requirements

(i) Mobile Device Sensor Integration

HikerNet interfaces with GPS, compass, and barometer sensors on smartphones to collect accurate geospatial data for route tracking and environmental context.

(ii) Third-party API Integration

The app integrates with OpenStreetMap and weather APIs to provide essential mapping, route planning, and weather updates, enhancing safety and navigation.

(iii) Notification Services

Push notifications and in-app alerts are integrated to provide real-time updates on SOS alerts, trek progress, group messages, and social interactions.

3.4.3 Nonfunctional Requirements

1. User-friendly interface with intuitive navigation and clear action prompts.
2. Secure authentication with encrypted user data and session management.
3. High performance and responsiveness even with large datasets and multiple simultaneous users.
4. Offline functionality for trail recording and emergency alerts.
5. Scalable database architecture to support a growing user base.
6. Compliance with data privacy laws and secure storage protocols.

3.4.4 Hardware Requirements

- Smartphones with GPS, accelerometer, compass, and internet connectivity.
- Cloud servers for backend services and database management.

3.4.5 Software Requirements

- Frontend development using React Native for cross-platform mobile apps.
- Backend APIs built with Node.js and Express.
- MongoDB for NoSQL database storage.
- Integration with OpenStreetMap, weather APIs, and notification services.

3.5 PROCESS MODEL

We use the Agile model for the development of our project. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.

We implement our project by using the following steps:

1. Requirements gathering: In this phase, we find the requirements. We find the requirements and plan the time and effort needed to build the project by implementing the requirements.

2. Designing the system: Once the requirements are gathered we can design the system based on the requirements. The designed system must be able to satisfy all the requirements found in previous step.

3. Construction/ iteration: Once we define the requirements and designs are made we can start working on the system. The system will undergo various stages of improvement, so

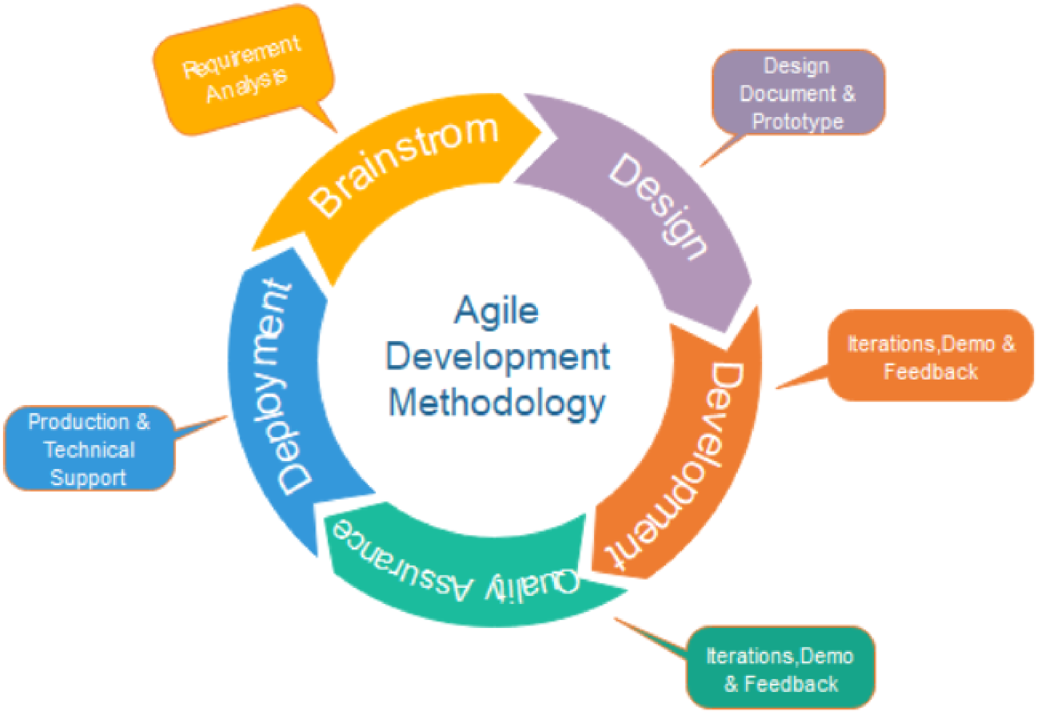

Figure 3.1: Agile method

it includes simple, minimal functionality at each iteration.

4. Testing: In this phase, the team examines the system's performance and looks for the bug.

5. Deployment: In this phase, the finished system is deployed with functionalities implemented.

6. Feedback: After the product is deployed the team receives feedback about the system, such as performance improvements or better functionalities.

The above steps are repeated until all necessary functionalities are identified and implemented.

CHAPTER 4
FEASIBILITY STUDY

4.1 ECONOMIC FEASIBILITY

4.1.1 Estimated Hardware Cost

The project does not require any specialized hardware for end users as it is designed to run on smartphones equipped with GPS and other necessary sensors. The backend cloud infrastructure leverages scalable cloud servers minimizing the need for costly proprietary hardware.

4.1.2 Estimated Software Cost

HikerNet is developed predominantly with open-source tools and cloud services with free tier options such as React Native, Node.js, MongoDB, and Firebase. There are minimal costs associated with paid API subscriptions for weather and maps, and these will be managed within the project budget.

4.1.3 Conclusion

The economic requirement for HikerNet's development and deployment is manageable within academic or startup funding constraints, making it an economically feasible project.

4.2 TECHNICAL FEASIBILITY

4.2.1 Product Requirements

- Smartphones with GPS, barometer, compass sensors for accurate location and environmental data.

- Cloud servers capable of handling backend processing, live location streaming, and data storage.
- Integration with OpenStreetMap and Weather APIs for mapping and environmental updates.
- Offline data caching and SOS alert features to function in low connectivity environments.

4.2.2 Software Requirements

- React Native framework for cross-platform mobile interface.
- Node.js and Express for backend API and real-time data management.
- MongoDB or Firebase for scalable and flexible data storage.
- Libraries for geospatial computations for dynamic rendezvous point calculation and group coordination.

4.2.3 Development Requirements

- Development IDEs such as Visual Studio Code and Android Studio.
- Team expertise in JavaScript, React Native, Node.js, cloud deployment, and API integration.
- Device labs for testing on multiple smartphone platforms.

4.2.4 Skill Requirements

- Proficiency in mobile app development with React Native.

- Backend development experience with Node.js and NoSQL databases.
- GIS and map API integration capabilities.
- Knowledge in cloud infrastructure and scaling.

4.3 OPERATIONAL FEASIBILITY

4.3.1 Throughput and Response Time

- The React Native client app ensures responsive UI across device types.
- Backend API based on Node.js allows concurrency and efficient real-time communication.
- Cloud databases and caching methods facilitate rapid data retrieval.

4.3.2 Timeliness and Data Accuracy

- GPS sensors provide precise and timely location information for coordination.
- Data validation and indexing ensure accurate info delivery.
- Weather API updates help ensure timely environmental awareness.

4.3.3 Resource Utilization

- Efficient use of React Native and Node.js technology stacks minimizes overhead.
- Cloud scaling adapts resource allocation according to demand.
- Team is organized for optimized collaboration and workflow.

4.3.4 Reliability

The system employs fault-tolerant offline functionality, robust cloud infrastructure, and modular design ensuring high reliability in diverse field conditions.

4.3.5 Scalability and Flexibility

Scalable cloud solutions and flexible backend allow HikerNet to accommodate growing user base and feature expansion seamlessly.

4.3.6 Compatibility With New System

The chosen technologies and libraries are widely supported, ensuring smooth integration and future-proofing of the platform.

4.3.7 User Adoption

User-centric design combined with engaging social and safety features aims to provide a compelling experience that attracts and retains trekking communities.

4.4 SCHEDULING FEASIBILITY

- Frontend mobile application development with React Native targeted to complete within 2 months.
- Backend API and database system development planned across 2 months.
- Final testing, debugging, and deployment phase expected in 1 month.

The schedule aligns with resource availability and academic timelines, enabling a feasible project rollout within 5 months.

CHAPTER 5
ARCHITECTURAL DESIGN

Architectural design focuses on the design of system architecture. It describes the structure and behavior of the system. It defines the structure and relationship between various modules of system development process. The diagrammatic representation of the system architecture is called the system architecture diagram. This diagram gives us the abstract view of the components and their relationship with the system that makes the system work.

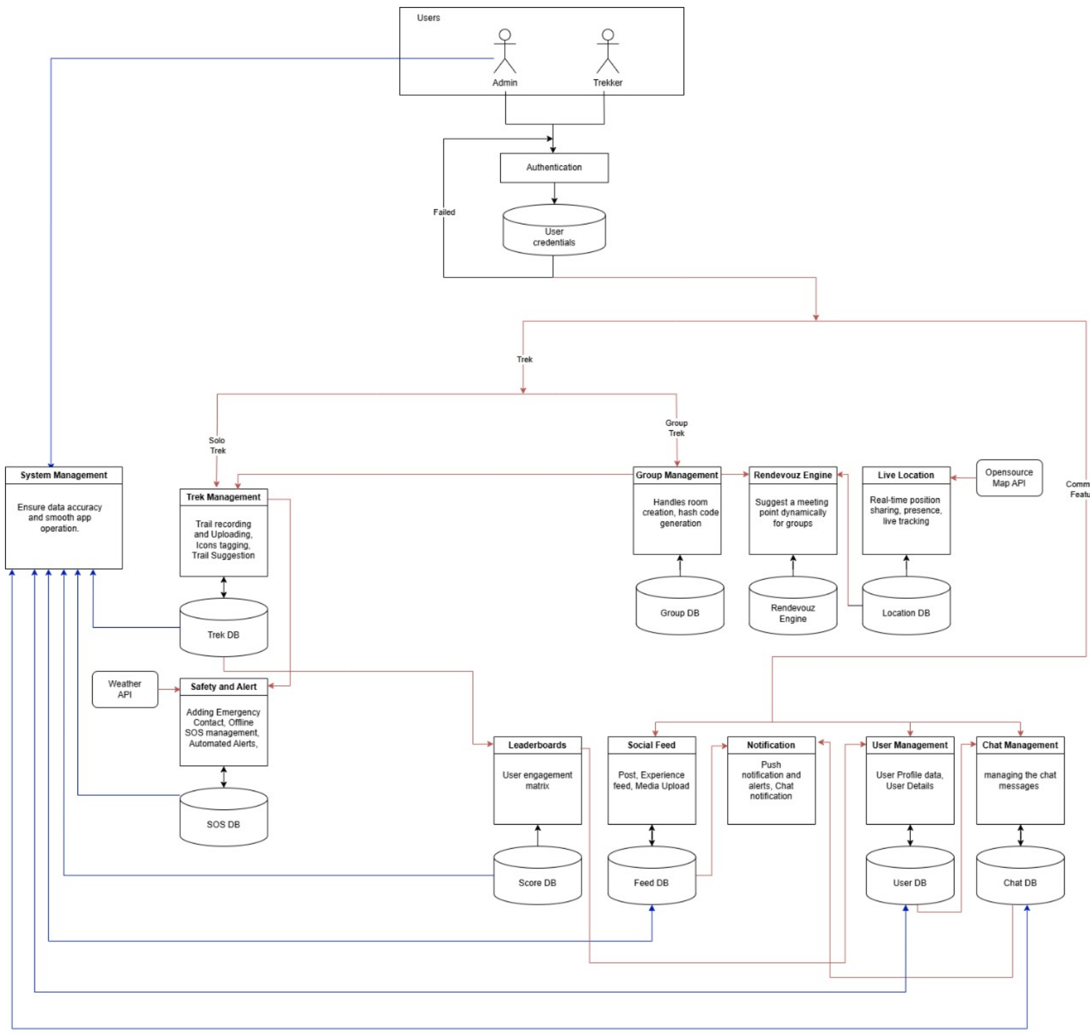

Figure 5.1: System Architecture

5.1 MODULE DESCRIPTIONS

5.1.1 User Management

This module manages user registration, authentication, profile creation, and social connections. It validates and encrypts user credentials, stores user details in a secure cloud database, and controls session management. Features include following users, updating profiles, and viewing user activity logs. Security and privacy are emphasized with role-based access for both trekkers and admins.

Implementation: User registration/authentication via Firebase Auth, profile storage in MongoDB Atlas, bcrypt password hashing, and session management.

5.1.2 Trek Management

Handles the core trekking features: recording GPS trails, saving checkpoints, calculating distance and elevation, and recommending new trek routes. Supports offline logging and syncs trek data to the cloud database. Visualizes trek maps and statistics, providing a complete log for each user.

Implementation: Device GPS and sensor APIs, Haversine formula for distances, integration with OpenStreetMap.

5.1.3 Rendezvous Engine

Dynamically suggests the most optimal meeting points for trekking groups based on real-time user positions. Uses geometric median algorithms for midpoint calculation and integrates safety checks, avoiding unsafe locations.

Implementation: Geometric median computation, visualization on map interfaces.

5.1.4 Live Location Sharing

Broadcasts a user's real-time GPS location to group members for coordination and safety in the trekking environment. Uses efficient data streaming so that group members can see each other's movements on a shared map.

Implementation: Location updates via backend built with Node.js Express, real-time display using map APIs.

5.1.5 Safety and Alerts (SOS)

Provides emergency functionality by allowing users to trigger SOS alerts, queue emergency messages, and notify designated contacts automatically. Alerts are sent even during offline scenarios, once connectivity is restored.

Implementation: Offline message queuing, background worker service for retrying sends, notification delivery.

5.1.6 Community Interaction (Social Feed)

Enables users to share posts, photos, trek logs, comment, like, and follow each other to build a shared adventure record. Supports uploading tagged images and creating stories, with content optimized for mobile viewing.

Implementation: CRUD Applications for user-generated content, profile interaction features, image storage via Firebase, metadata tagging (EXIF extraction).

5.1.7 Reports and Analytics

Aggregates trek data and user activities into summaries, generates statistics for progress tracking, map visualizations, and downloadable reports. Users can view their history, trails, and engagement rankings.

Implementation: Charting via MPAndroid-Chart (mobile), export functions with iTextPDF, real-time analytics dashboarding.

5.1.8 Admin Management

Provides a dashboard for administrators to moderate users and content, manage reports, perform system audits, and enforce security policies. Includes automated moderation tools and manual review capabilities.

Implementation: Admin dashboard built in React Native, role-based access control, content management workflows, automated flagging of inappropriate material.

5.1.9 Notification System

Manages push and in-app notifications for trek events, alerts, group messages, and community activities. Each user can configure notification preferences for personalized alerts.

Implementation: Notification queue system, event-driven triggers, and integration with multi-channel delivery platforms for cross-platform notification delivery.

5.1.10 Leaderboards and Achievements

Tracks user engagement, awards badges for milestones, and maintains community leaderboards. Encourages active participation and allows users to view their ranking and achievements.

Implementation: Rule-based algorithms analyzing activity logs, badges as vector icons, leaderboard updates with scoring functions.

5.1.11 Supporting Services

Includes weather API integration for up-to-date environmental information, external map APIs for route and location sharing, and robust cloud storage for all user and group data.

Implementation: Real-time weather via OpenWeatherMap, maps via OpenStreetMap, MongoDB Atlas for secure data management.

CHAPTER 6
DESIGN AND IMPLEMENTATION

6.1 INTRODUCTION

A data flow diagram is a graphical representation of the information flow in a business process. It demonstrates how data is transferred from the input to the file storage and reports generation. By visualizing the system flow, the flow charts will give users helpful insights into the process and open up ways to define and improve their business. A DFD can be divided into levels and layers, thus users can focus on describing a particular stage.

6.2 DATA FLOW DIAGRAMS

6.2.1 Level 0 DFD

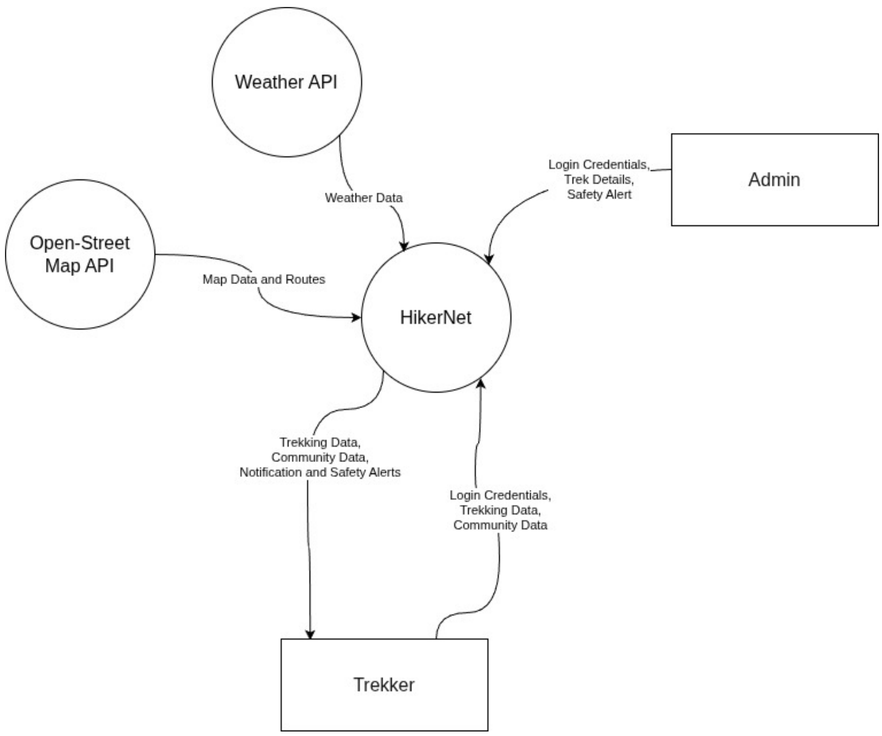

Figure 6.1: Level 0 Data Flow Diagram

The Zeroth Level Data Flow Diagram (DFD) provides a high-level overview of HikerNet, showing major interactions between external entities and the system. HikerNet collects and processes login credentials, trekking data, and community data from trekkers and admins, while integrating route and weather information from relevant APIs. The diagram highlights how key data flows support coordination, safety, and communication for all users.

6.2.2 Level 1 DFD

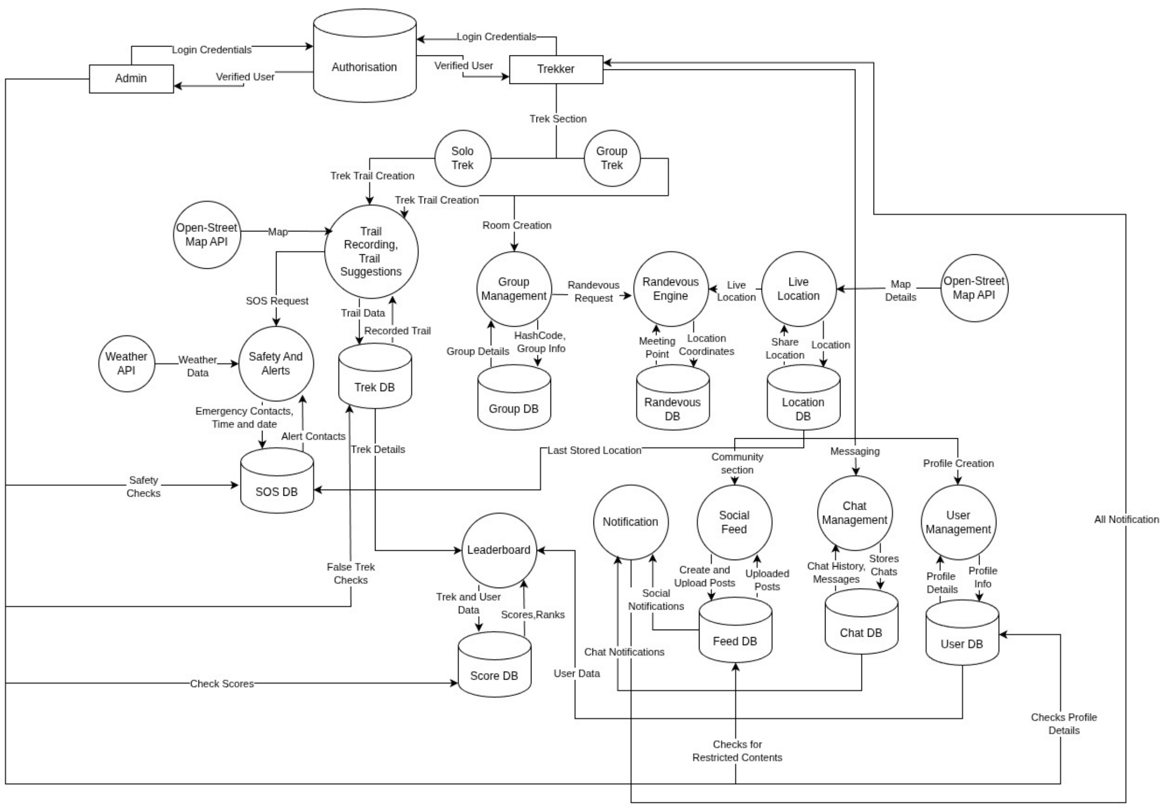

Figure 6.2: Level 1 Data Flow Diagram

The First Level Data Flow Diagram (DFD) breaks down HikerNet's core functionalities, showing how users interact with system modules such as login/authentication, trekking management, group coordination, live location, and safety features. Each module handles specific processes: user management controls profiles and connections, while notification and social feed modules enable messaging and community engagement. The platform employs proactive toxicity checks and route safety validation to maintain a secure environment before content or location updates are published. Integration with map and weather APIs supports accurate navigation and real-time environmental awareness. The overall system leverages a robust backend to ensure secure data processing, personalized user experiences, and effective filtering of inappropriate content, making for a safe and interactive trekking companion platform.

6.2.3 Level 2 DFD

The Second Level Data Flow Diagrams (DFDs) of HikeNet provide a detailed view of the system's internal processes by splitting functionality into two focused diagrams: one for Trek Features and one for Community Features. This modular approach highlights both the operational backbone and the interactive, user-facing elements of the platform.

6.2.4 Trek Features DFD

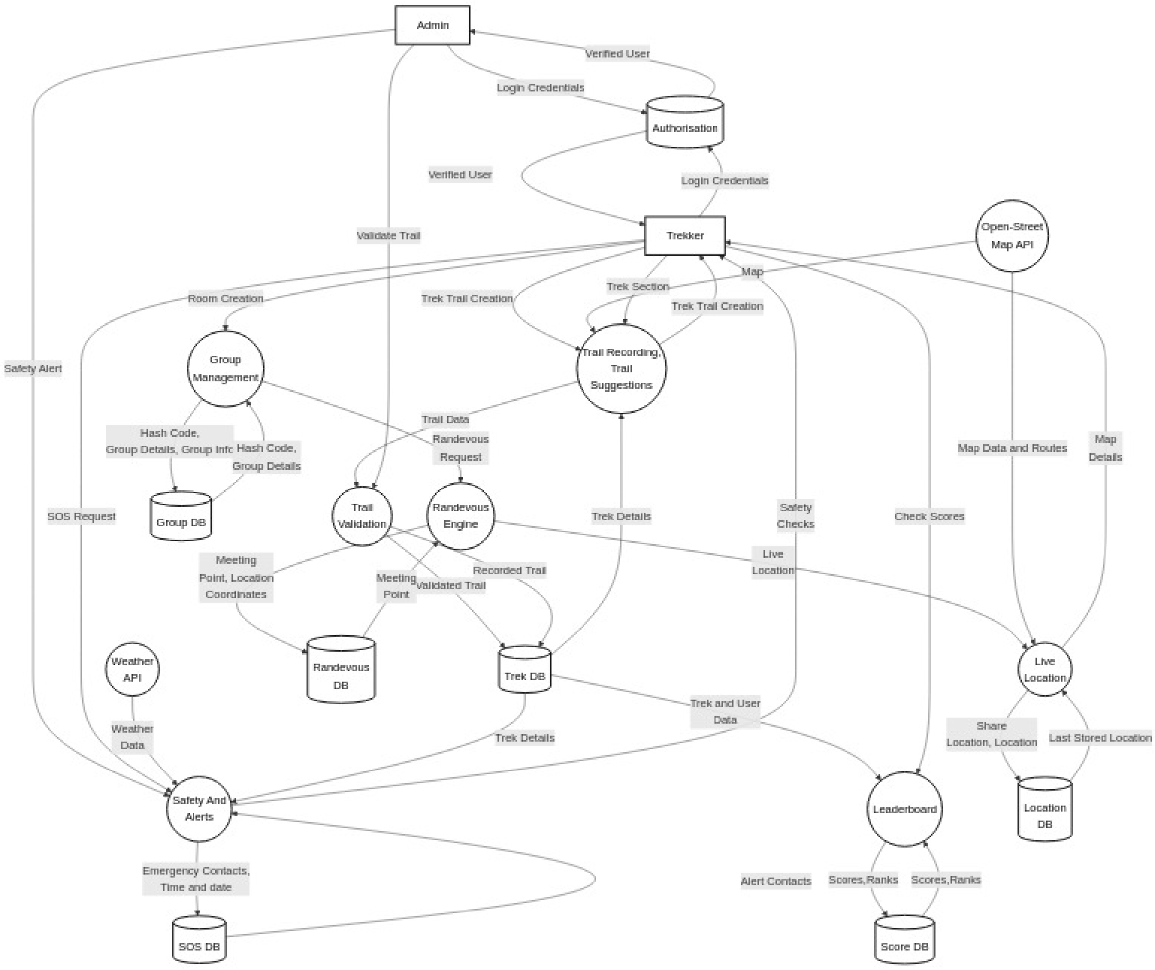

Figure 6.3: Level 2 Data Flow Diagram - Trek Features

The Second Level Data Flow Diagram (DFD) for trekking features elaborates how HikerNet manages all aspects of the trekking process, including trail recording, group and solo trek coordination, route validation, and safety management. The diagram shows how data moves between modules such as trail suggestions, group management, safety alerts, and real-time location sharing. Integration with APIs for weather and maps as well as connections with various internal databases enables seamless tracking, emergency responses, and activity validation for trekkers, ultimately ensuring safety and robust adventure planning.

6.2.5 Community Features DFD

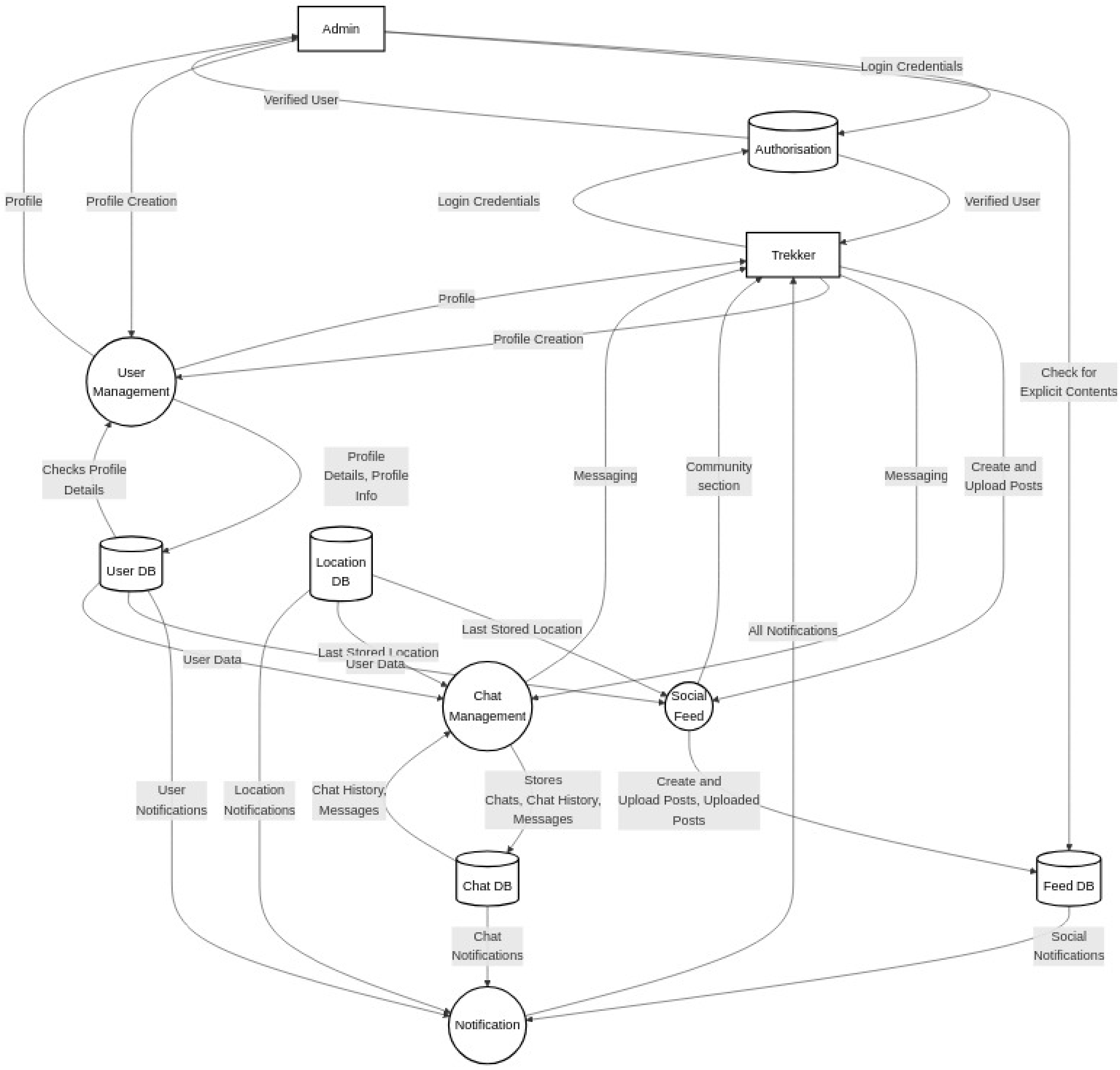

Figure 6.4: Level 2 Data Flow Diagram - Community Features

The second Second Level Data Flow Diagram (DFD) focusing on community-centric components, emphasizing user engagement and personalized interactions. Here, the homepage aggregates a dynamic feed with posts and activity from a user's network, fostering discovery and connectivity. Profile management functionalities are detailed, supporting tasks such as editing information and managing social links. The diagram also details real-time notifications—alerting users to posts, messages, and network events—and interactive messaging, which encourages community participation. Together, these processes reinforce HikerNet's commitment to a safe, engaging, and responsive user environment.

6.3 WORK PLAN AND TASK ALLOCATION

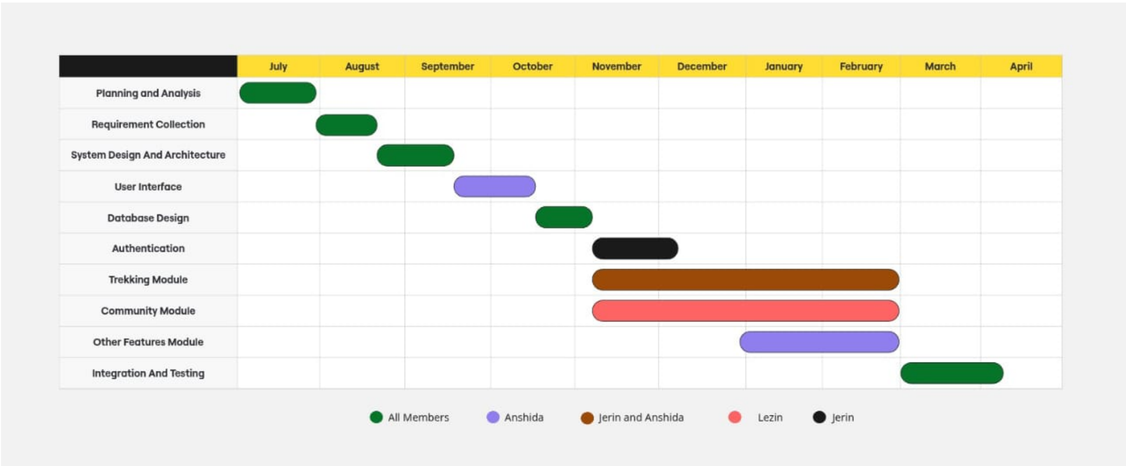

Figure 6.5: Workplan and Task Allocation

REFERENCES

1. Inc. AllTrails. Alltrails: Hiking and trail app. `https://www.alltrails.com`, 2025. Mobile and Web Application. Accessed October 2025.

2. Inc. Strava. Strava: Activity tracker. `https://www.strava.com`, 2025. Mobile and Web Application. Accessed October 2025.

3. Komoot GmbH. Komoot: Outdoor route planner. `https://www.komoot.com`, 2025. Mobile and Web Application. Accessed October 2025.

4. Julia Ann Mathew and Suining He. Lessons from a large language model-based outdoor trail recommendation chatbot with retrieval augmented generation. In *UrbComp Conference*, 2025. Retrieved from arXiv preprint arXiv:2508.05652.

5. R. Ragavendran, S. Kumar, and A. Sharma. Sos emergency alert and assistance mobile application. *ResearchGate*, 2024. Accessed October 2025.