

Table of Contents

CHAPTER 1: INTRODUCTION.....	1
1.1. BACKGROUND.....	1
1.2. PROBLEM STATEMENT.....	1
1.3. GOALS OF THE PROJECT	2
1.4. SCOPE OF THE PROJECT	3
<i>Core Features</i>	3
<i>Extended Features (Potential Future Enhancements):</i>	4
<i>Technological Considerations</i>	4
1.5. FEASIBILITY ANALYSIS	4
1.6. SYSTEM REQUIREMENTS.....	5
CHAPTER 2: LITERATURE REVIEW.....	7
CHAPTER 3: METHODOLOGY	8
3.1 SYSTEM DEVELOPMENT LIFE CYCLE	8
3.2 REQUIREMENTS.....	9
3.3 USE-CASE DIAGRAM	12
.....	12
3.4. DESIGN	13
3.5 IMPLEMENTATION.....	13
3.6 VERIFICATION AND TESTING	13
3.6 DEPLOYMENT AND MAINTENANCE	14
3.7 GANTT CHART	14
REFERENCES.....	15

LIST OF TABLES

TABLE 1: HARDWARE REQUIREMENTS	6
TABLE 2: SOFTWARE REQUIREMENTS	6
TABLE 3: FUNCTIONAL REQUIREMENTS	9
TABLE 4: NON-FUNCTIONAL REQUIREMENTS	10

LIST OF FIGURES

FIGURE 1: PROTOTYPE MODEL	8
FIGURE 2 USE-CASE DIAGRAM	12
FIGURE 3: GANTT CHART	14

CHAPTER 1: INTRODUCTION

1.1. Background

Mitho-Mitho is an innovative web application crafted for passionate food enthusiasts. Its primary purpose is to empower users in effortlessly exploring restaurants, keeping tabs on their culinary adventures, and fostering connections with like-minded friends. If the challenge of recalling past dining experiences or identifying standout dishes plagues you, Mitho-Mitho steps in as your ultimate companion.

The app simplifies the restaurant discovery process by enabling users to search for establishments either by name or address. Once the search yields results, you can cherry-pick and add a restaurant to your personalized map, establishing a delightful trail of Mitho-Mithos that encapsulate your unique dining journey.

Beyond individual exploration, Mitho-Mitho transforms into a dynamic social hub for food enthusiasts. By connecting with friends, users gain insight into their peers' gastronomic escapades—unveiling the hidden gems they've uncovered and the delectable dishes they've savored. In essence, Mitho-Mitho stands as a dedicated social media network tailored for foodies, fostering a vibrant community united by their shared love for exceptional culinary experiences.

1.2. Problem Statement

Some of the key problems that we're targeting to solve with our web-app are :

1. Difficulty in Recalling Dining History
 - Users struggle to recall and organize details of past restaurant visits.
 - Lack of centralized system leads to fragmented and disorganized dining histories.
2. Inefficient Restaurant Discovery
 - Existing methods for discovering new restaurants are not user-friendly.
 - Users may miss potential dining gems due to the absence of an efficient search and recommendation system.
3. Limited Social Connectivity in Culinary Experiences
 - Current platforms do not effectively facilitate sharing restaurant experiences with friends.
 - A gap exists in creating a social network specifically tailored for food enthusiasts.
4. Absence of Dedicated Tracking Platform
 - There is a need for a specialized application for tracking and documenting one's culinary journey.
 - Existing general-purpose apps lack features designed for foodies to chronicle restaurant visits.

5. Unmet Demand for a Food-Centric Social Media Network
 - Food enthusiasts lack a dedicated social platform to connect, share and explore dining experiences.
 - Current social media networks do not cater specifically to the unique needs of the foodie community.

1.3. Goals of the Project

The aim of our project is to find the ways in order to ease and tackle the problems somehow mentioned in the above problem statement. Some of the key goals of our project are:

1. Develop an intuitive User Interface
Create a user-friendly interface for seamless navigation and efficient use.
2. Implement Comprehensive Restaurant Search
Develop a robust search and recommendation system for users to discover new restaurants easily.
3. Facilitate Social Connectivity
Integrate social features allowing users to share, comment, and connect with friends based on their dining experiences.
4. Enable Personalized Dining History Tracking
Implement a dedicated tracking system for users to document and recall their unique dining histories.
5. Build a Foodie-Centric Social Network
Establish Mitho-Mitho as a specialized social media platform catering specifically to the interests and needs of food enthusiasts.
6. Enhance Community Engagement
Foster an active and engaged community by encouraging users to interact, share recommendations, and explore each other's culinary adventures.
7. Ensure Data Privacy and Security
Implement robust measures to safeguard user data, ensuring privacy and security throughout the Mitho-Mitho experience.

By achieving these goals, the Mitho-Mitho project aims to address the identified challenges, providing a comprehensive and enjoyable platform for users to discover, document, and share their culinary journeys.

1.4. Scope of the Project

The scope of the Mitho-Mitho project encompasses a range of features and functionalities that contribute to its overarching objectives. Here's an outline of the potential scope for the project:

Core Features

1. User Registration and Authentication
 - Allow users to create accounts and log in securely.
2. Intuitive User Interface
 - Design an intuitive and visually appealing interface for seamless navigation.
3. Restaurant Search and Recommendation
 - Implement a robust search engine to enable users to discover restaurants based on various criteria such as name, location, cuisine, and user reviews.
4. Personalized Dining History
 - Provide users with the ability to create and maintain a personalized map of visited restaurants, including details like date, dishes tried, and ratings.
5. Social Connectivity
 - Facilitate connections between users, allowing them to follow friends, view each other's dining histories, and share recommendations.
6. Interactive User Profiles
 - Create user profiles with the option to showcase favorite dishes, reviews, and other personalized information.
7. Community Engagement Features
 - Implement features such as comments, likes, and sharing to enhance community interaction.
8. Notifications and Alerts
 - Enable users to receive notifications about friends' dining activities, new restaurant recommendations, and interactions on the platform.
9. Privacy Controls
 - Implement privacy settings to allow users to control the visibility of their dining histories and other personal information.

Extended Features (Potential Future Enhancements)

1. Integration with External APIs
 - Explore partnerships with external platforms for additional data enrichment and restaurant information.
2. Advanced Recommendation Algorithms
 - Implement machine learning algorithms to provide personalized restaurant recommendations based on user preferences and behavior.
3. Event Planning and Group Dining
 - Allow users to plan events, invite friends, and coordinate group dining experiences.
4. Integration with Social Media Platforms
 - Provide options for users to share their dining experiences seamlessly on other social media networks.
5. Gamification Elements
 - Introduce gamification elements such as badges, challenges, or rewards to enhance user engagement.

Technological Considerations

1. Cross-Platform Compatibility
 - Develop the application to be accessible across various devices, including web browsers and mobile platforms.
2. Scalability and Performance
 - Design the system architecture to handle potential growth in user base and ensure optimal performance.
3. Data Security and Compliance
 - Implement robust security measures to protect user data and ensure compliance with data protection regulations.
4. Regular Updates and Maintenance
 - Plan for ongoing updates and maintenance to address user feedback, fix bugs, and introduce new features.

1.5. Feasibility Analysis

The systematic evaluation of the practicality and viability of a proposed project, plan,

or idea. It involves assessing various factors to determine whether the project is worth pursuing. Under it, we found our proposed design suit all that condition. This proposed solution satisfy all the user requirement and flexible for any changed in future.

1.5.1. Economic Feasibility

It should be needed to considered the economic level for successful product design.Considering that,our proposed solution too satisfy the following term:

- Cost for survey and investigation
- Cost of hardware & software design
- Cost of product placement and error check and maintenance

The organization prioritizes cost-effectiveness in selecting technology, conducting a thorough cost-benefit analysis for hardware and software. The chosen technology aims to be scalable and adaptable, with risk mitigation strategies in place.nOt only That ,company too get benefit through launching this solution.

1.5.2. Operational Feasibility

Our proposed design is GUI based,userfriendly so that anyone simply used the product, no need to be tech-expert as everything is simple and make sense. It includes determining usability of product,operate & maintenance after deployment.

Our design is mainly for food lovers! It makes it super easy for them to find places like restaurants and hotels in a friendly and welcoming environment.

1.5.3. Legal Feasibility

Additionally, we check if the project has the right permissions or licenses it needs to happen, and if it respects privacy, when dealing with personal information. We also consider the project's impact on the environment and follow rules about taking care of nature.It includes analyzing barrier of legal implementation of project,data protection acts or social media laws, project certificate,license,copyright etc.

1.5.4. Technical Feasibility

The Mitho-Mitho project demonstrates high technical feasibility by leveraging Flask as the web development framework and PostgreSQL as the chosen database management system. Flask, a lightweight and flexible framework, enables efficient web application development, while PostgreSQL provides a robust and scalable database solution. The technologies support seamless integration with external APIs for location services and social connections. Security measures, including secure authentication and data encryption, will be implemented to safeguard user information. Responsive design principles will ensure cross-platform compatibility, and comprehensive testing using frameworks like PyTest will be conducted. The team's technical proficiency with Flask and PostgreSQL, along with well-documented development practices, contributes to the project's successful implementation and maintenance.

1.6. System Requirements

The efficient operation of computer software hinges on the interplay between specific hardware components and software resources. These prerequisites, commonly termed system requirements, are more akin to guiding principles than strict mandates. Software usually specifies two tiers of requirements: minimum and recommended. As software progresses, there is a heightened demand for increased processing power and resources,

thereby influencing the trajectory of system requirements. According to industry analysts, this evolving landscape is a major factor driving upgrades, often overshadowing the impact of technological advancements.

1.6.1. Hardware Requirements

Table 1: Hardware Requirements

Component	Requirement
Desktop/Laptop	-Any modern computer or laptop -Dual-core Processor -2 GB RAM or more -Internet connectivity
Smartphone/Tablet	-iOS or Android device -Dual-core processor -2 GB RAM or more -Internet connectivity

The web application's compatibility across various devices and operating systems makes it accessible to a broad audience with diverse hardware capabilities. The client device's processing power and memory are typically the primary factors affecting the user experience.

1.6.2. Software Requirements

Table 2: Software Requirements

Software	Requirement
Operating System	- Compatible with major operating systems (Windows, macOS, Linux for desktop; iOS, Android for mobile)
Web Browser	-Supports modern web browsers(Chrome, Firefox, Safari, Edge)
Internet Connectivity	-Stable Internet connection
Mobile Web-App	-Compatible with a variety of screen sizes and resolutions

CHAPTER 2: LITERATURE REVIEW

In recent years, the intersection of technology and gastronomy has given rise to innovative applications aimed at enhancing the dining experiences of individuals. Mitho-Mitho, a web application tailored for food enthusiasts, capitalizes on this trend by offering a multifaceted platform that amalgamates restaurant discovery, dining history tracking, and social connectivity in the realm of gastronomy. This literature review explores the existing landscape of applications like Mitho-Mitho, delving into the various features they offer and the impact they have had on users' culinary journeys.

1. Restaurant Discovery Apps

The advent of restaurant discovery applications has revolutionized the way individuals find, choose, and experience dining establishments. Platforms like Google Maps, Yelp, TripAdvisor and Zomato have set the precedent by providing users with comprehensive databases of restaurants, reviews, and ratings. These applications serve as valuable tools for users seeking new culinary adventures. Mitho-Mitho, with its focus on intuitive restaurant search capabilities, aligns itself with this paradigm.

2. Personalized Food Journals

The concept of keeping a record of one's dining experiences has gained traction through applications like Evernote Food and Food Diary. These tools allow users to document their meals, capture photographs, and jot down personal impressions. Mitho-Mitho takes this a step further by integrating a personalized map feature, enabling users to create a visual representation of their culinary journey. This unique amalgamation of food journaling and location-based tracking sets Mitho apart in the domain of gastronomic applications.

3. Social Connectivity in Culinary Exploration

The integration of social elements into food-centric applications has become increasingly prevalent. Foodies seek platforms where they can share their culinary escapades and recommendations. Mitho-Mitho aligns with this trend by offering a social network tailored for food enthusiasts. Similar applications such as Instagram have paved the way for this socialization of culinary experiences, fostering a community where users can connect over shared gastronomic interests.

4. Challenges and Opportunities

While Mitho-Mitho shares some features with other well-established apps and platforms like Google Maps, Yelp and Evernote, it focuses on serving purposes in a varied manner. Mitho presents itself different from other as it features:

- Food-Centric Platform
 - Our web-app is specifically designed for food enthusiasts, focusing on tracking dining experiences, sharing food-related content, and connecting with friends based on their culinary preferences.

- **Personal Dining History**
 - It allows users to create a personal map of restaurants they have visited, making it a customized tool for tracking their unique dining journey.
- **Social Networking for Foodies**
 - The social aspect of the web-app is tailored to the foodie community, where users can connect with friends to discover new restaurants and dishes.
- **Detailed Food Reviews:**
 - Users can leave detailed reviews, including information about specific dishes they've tried, creating a more food-centric and personalized review system.
- **Focused on Recommendations:**
 - It aims to provide personalized recommendations based on users' dining history and preferences, enhancing the food discovery experience.

CHAPTER 3: METHODOLOGY

3.1 System Development Life Cycle

Here, the Prototyping model is a fitting approach for our project, considering our stable development environment and well-defined requirements. The Prototyping model is an iterative software development process where development is seen as a series of cycles involving requirements gathering, quick design and implementation, user evaluation, and refinement. This iterative nature allows for continuous feedback and adjustments, ensuring a more dynamic and adaptable development process compared to the linear flow of the waterfall model.

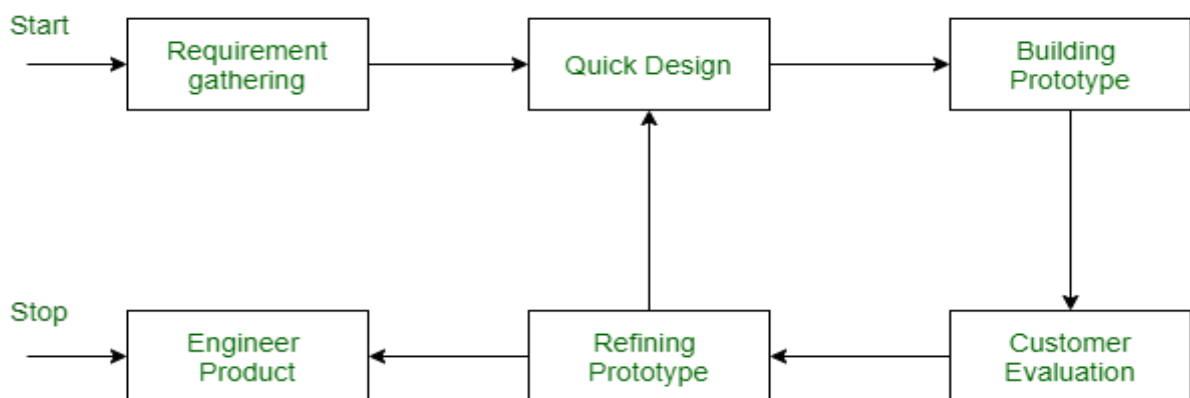


Figure 1: Prototype Model

3.2 Requirements

3.2.1 Functional Requirements

Table 3: Functional Requirements

RQID	Name	Description	Actor
01	User Registration	Functionality for user to register	User
02	User Login	Functionality for user to login	User
03	User Logout	Functionality for user to logout	User
04	Editing Profile	Functionality for user to edit his/her profile	User
05	Map Tracking	Functionality for the user to track hotel on Google Maps	User
06	Search Restaurant	Functionality for the user to search for a restaurant	User
07	Search Friend	Functionality for the user to search for a friend	User
08	Add Friend	Functionality for the user to send friend request	User
09	Accept Friend Request	Functionality for the user to accept friend request	User
10	Decline Friend Request	Functionality for the user to decline the friend request	User
11	Track friend activity	Functionality for the user to track other user's activity	User
12	Text Messaging	Functionality for the user to send text to the other users	User
13	Real Time Notification	Functionality of the admin to send real time notification to the user	admin
14	Upload Food Picture	Functionality of the user to upload the food they eat	User
15	Switch the Mode (Dark/Light)	Functionality of the user to switch between light and dark mode	User
16	Leave a Trail	Functionality of the user to leave ma Mithi-Mitho trail at the restaurant	User
17	Track Activity	Functionality of the user to track each and every registered users' activity	User

3.2.2 Non-Functional Requirements

Table 4: Non-Functional Requirements

RDIQ	Name	Description
01	Privacy	Users' privacy
02	Robustness	To be able to deal with errors
03	Performance	Application performance must be better
04	Usability	Easy to use
05	Reliability	To be able to gain user's trust
06	Supportability	To be able to provide help and support to the users

3.2.3 Technologies Used

Our proposal incorporates a range of powerful technologies to ensure the effectiveness and scalability of our web application. Here is an overview of the key technologies utilized in our project:

1. Blinker
 - Facilitating efficient signaling between components, enhancing the responsiveness of our application.
2. Flask
 - A lightweight and powerful Python web framework, forming the basis for the development of our web application.
3. Flask-DebugToolbar
 - Integrated for streamlined debugging and profiling during development, ensuring a smooth coding experience.
4. Flask-SQLAlchemy
 - Used for seamless integration with SQLAlchemy, simplifying database operations and enhancing data management.
5. SQLAlchemy
 - A versatile SQL toolkit and Object-Relational Mapping (ORM) library, enabling efficient database interactions.
6. SQLAlchemy-Searchable
 - Implemented to support advanced search functionality within the application, enhancing user experience.
7. SQLAlchemy-Utils

- Integrated for additional utility functions, improving code organization and optimizing database-related tasks.

8. Werkzeug

- The underlying library for WSGI compliance, ensuring compatibility and seamless web server integration.

9. Jinja2

- Employed as the template engine to facilitate dynamic content generation, enhancing the presentation layer.

10. Requests

- Utilized for simplified and efficient HTTP requests, enabling seamless communication with external APIs and services.

11. Selenium

- Integrated for automated testing of web applications, ensuring robustness and reliability in various user scenarios.

12. Coverage

- Implemented to measure code coverage during testing, ensuring a comprehensive assessment of the codebase's reliability.

13. psycopg2:

- A PostgreSQL adapter for Python, facilitating efficient and secure interaction with PostgreSQL databases.

14. oauth2

- Employed for secure authentication and authorization processes, enhancing the overall security posture of our web application.

15. Validators

- Utilized for data validation, ensuring that user inputs adhere to predefined standards and minimizing the risk of errors.

These technologies collectively contribute to the robustness, scalability, and security of our web application, laying the foundation for a seamless and feature-rich user experience.

3.3 Use-Case Diagram

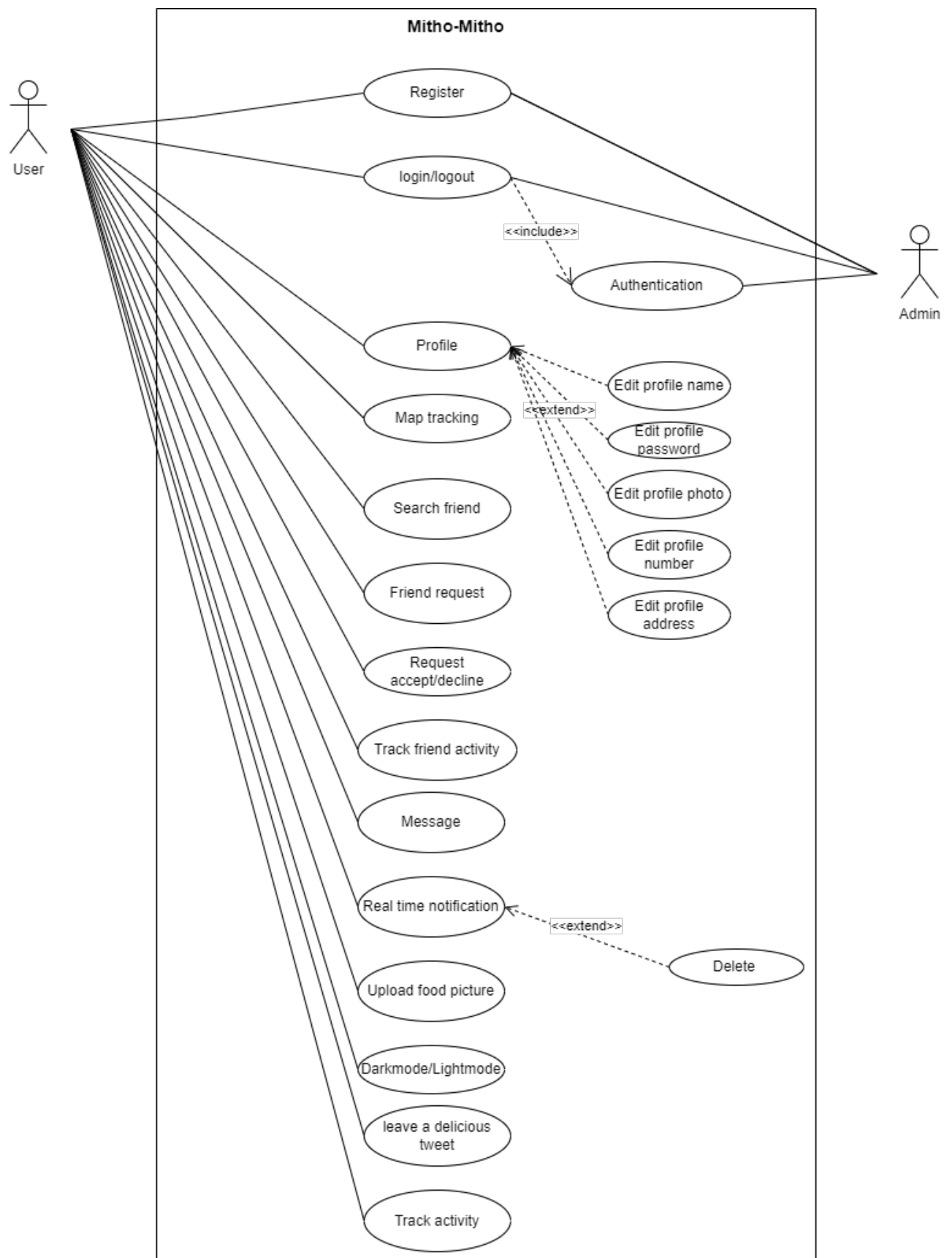


Figure 2 Use-Case Diagram

3.4. Design

In shaping our system, we have diligently considered both user-specific needs and technical requirements to create an impactful and user-friendly experience. This comprehensive process involves thorough data collection, analysis, user research, and the definition of functional and non-functional project requirements. The design phase is pivotal, serving as the nexus where user needs and technical aspects harmonize. Through iterative refinement, we align the project design with identified requirements, establishing a robust foundation for subsequent development.

Plans in Design Phase:

- Conduct user feedback sessions to refine user interface and experience.
- Integrate Google Maps API to enhance location-based features for users.
- Ensure responsiveness for both desktop and mobile devices.

3.5 Implementation

As we enter the implementation phase, our project team is poised to execute the plan diligently, translating ideas into tangible deliverables. This involves tasks such as procuring necessary resources, assembling the project team, and meticulously following the project plan. Continuous progress monitoring is crucial, allowing for adjustments to be made to adhere to timelines and budget constraints. The implementation phase demands adaptability and a commitment to delivering results in line with the project vision.

Plans in Implementation Phase:

- Collaborate with hosting services such as Heroku for seamless deployment.
- Conduct a phased rollout, starting with internal testing among classmates and hostel mates.
- Assemble a dedicated support team to address any immediate issues during the initial rollout.

3.6 Verification and Testing

Post-implementation, the verification and testing phase ensures that project deliverables meet specified requirements and are fit for their intended purpose. Rigorous testing protocols are implemented to identify and rectify any deviations promptly. This phase is crucial for ensuring the reliability and functionality of the developed system.

Plans in Verification and Testing Phase:

- Conduct thorough testing of Google Maps API integration for accuracy and reliability.
- Implement user acceptance testing (UAT) with a diverse group of users to gather feedback.
- Iteratively refine and enhance features based on user testing results.

3.6 Deployment and Maintenance

The deployment phase involves the careful delivery and installation of our system into the live product environment. Simultaneously, the maintenance phase begins, encompassing bug fixes, security updates, and the addition of new features. Our proactive approach extends beyond deployment, addressing evolving needs and continuously enhancing the system for a sustained optimal user experience.

Plans in Deployment and Maintenance Phase:

- Deploy the application on Heroku for easy scalability and management.
- Roll out the application to a limited audience initially (hostel mates, classmates) for controlled testing.
- Monitor user feedback and promptly address any emerging issues or feature requests.
- Establish a regular update schedule to address security vulnerabilities and introduce new features based on user feedback.

3.7 GANTT CHART

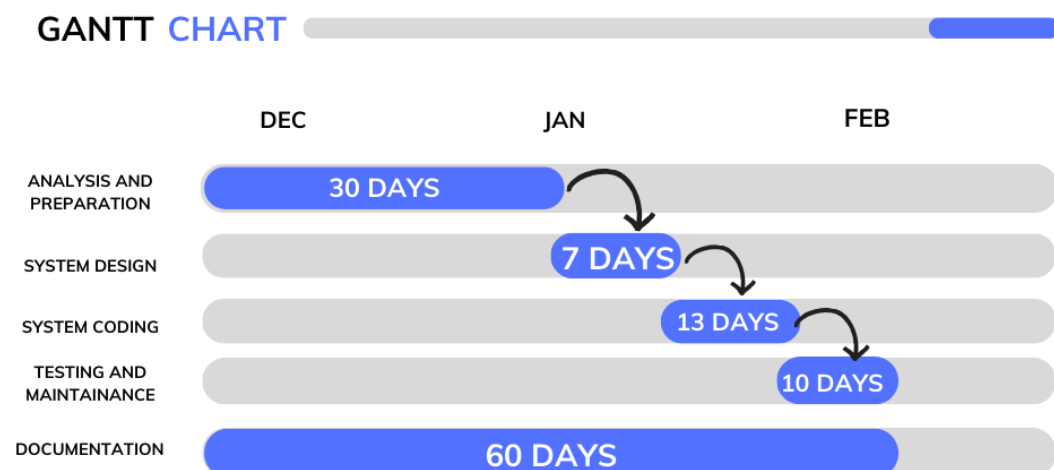


Figure 3: GANTT CHART

REFERENCES

- Google Maps Platform | Google for Developers. (n.d.). Google for Developers. <https://developers.google.com/maps>
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. *Remote Sensing Environment*, 202, 18–27. <https://doi.org/10.1016/j.rse.2017.06.031>
- Yelp. (n.d.). <https://www.yelp.com/>
- Michail, A. (n.d.). Digital Marketing for Hoteliers: Get More Direct Bookings For your Hotel.