

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

**“Fit-Track-Pro Personal Health and Fitness Tracker”**

**AGILE SOFTWARE ENGINEERING & DEVOPS**

(CS2004)

2024-2025

****

**Submitted by**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | **Abhay Aditya N** | **1RVU23CSE012** | | **Aneesh Adithya S R**  **Atyul V Bhaskar** | **1RVU23CSE054**  **1RVU23CSE090** | | **Amogh M Kashyap** | **1RVU23CSE045** |   Vaibhav M Chittakki 1RVUA24CSE7019 |

**Under the guidance of**

Prof. Manjul Krishna Gupta

Faculty of Agile SW and DevOps

School of Computer Science and Engineering

RV University-560059

**RV UNIVERSITY, BENGALURU-59**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

****

**CERTIFICATE**

Certified that the project work titled **Fit-Track-Pro**:**Personal Health and Fitness Tracker** is carried out by **Abhay Aditya N (1RVU23CSE012), Atyul V Bhaskar(1RVU23CSE090), Aneesh Adithya S R(1RVU23CSE054), Amogh Kashyap(1RVU23CSE045), Vaibhav M Chittakki(1RVUA24CSE7019)** in partial fulfillment of the completion of the course **Agile SW and DevOps** of the IV Semester Computer Science Engineering program, during the academic year 2024-2025. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the project report and duly approved by the faculty.

**Signature of Faculty Signature of Head of the Department**

**ABSTRACT**

**Fit-Track Pro** is a comprehensive, user-friendly health and fitness tracking application designed to support individuals in achieving and maintaining their wellness goals. Developed as part of an Agile Software Engineering project, the application integrates various health management components — including physical activity tracking, meal logging, workout calorie burner, and progress visualization — into a single, seamless platform.

The system architecture incorporates a responsive **ReactJS** frontend, a secure and scalable **Python (Flask)** backend, and a **MySQL** database for robust data management. Users can create personal profiles, log their fitness activities, monitor their daily nutritional intake, and visualize progress through intuitive dashboards. Additional features such as weekly/monthly health summaries can enhance the app's usability and effectiveness.

Unlike traditional fitness apps that focus on isolated features, Fit-Track Pro emphasizes holistic health management by combining multiple aspects of personal fitness into one synchronized tool. The application’s agile development methodology ensures iterative improvement, user feedback integration, and modular design. This approach not only enhances the system’s scalability and maintainability but also ensures a high degree of user engagement.

With privacy-focused design and future-ready capabilities such as wearable integration and AI-driven suggestions, Fit-Track Pro is well-positioned as a modern fitness companion, tailored to meet the dynamic needs of university students and general users alike. Our further plan includes adding a LLM by connecting google/gemma-2b-it downloaded from hugging face and running it locally using “transformers accelerate” to our website for smart feedback and responses.

**Table of Contents**

**Abstract** 3

1. **Introduction** 5

1.1 Project Domain and Problem Addressing

1.2 Issues and Challenges

1.3 Problem Statement

1.4 Requirements and Feasibility Study

1.5 Project objectives

1. **Literature Study** 7
2. **Design Details** 9

3.1 UML Diagrams

3.2 Methodology

3.3 Hardware and Software requirements

1. **Implementation details of the Project** 14

4.1 Language/Tools/APIs used

1. **Testing and Results** 15

5.1 Types of testing and validations done

5.2 Test Cases

* 1. Deployment

1. **Conclusion** 20

**References** 21

**Appendices** 21

**Appendix A**: Screenshots

1. **Introduction**

**1.1 Project Domain and Problem Addressing**

The project domain lies in the intersection of fitness technology and personal healthcare. In a time where sedentary lifestyles and poor diet choices are leading to increased chronic conditions, digital health applications can play a major role in prevention and self-management. Fit-Track Pro addresses the increasing demand for a holistic digital fitness companion that goes beyond simple step counters. It provides users with tools to manage not only their workouts but also their diet, workout and sleep and overall progress — all in one responsive, user-centric platform.

**1.2 Issues and Challenges**

During the design and implementation phases, several challenges were identified:

* Integration of multiple modules (workouts, meals, water intake, BMI, reports) without overwhelming the user.
* Providing real-time, responsive UI with accurate health visualizations.
* Ensuring secure user data handling and providing offline support.
* Achieving consistent data syncing across devices.
* Managing complexity using Agile practices without feature bloat.

**1.3 Problem statement**

While numerous fitness apps exist, many fail to combine all essential health-tracking components into a single platform. This forces users to switch between apps, disrupting continuity and reducing motivation. There is also a lack of apps with customizable goals, motivational feedback, and AI-driven suggestions. Fit-Track Pro aims to fill this gap by offering an integrated and engaging platform that supports all-round wellness.

**1.4 Requirements and Feasibility Study**

1.4 Requirements and Feasibility Study The application requires:

* Frontend: ReactJS-compatible device+HTML/CSS (modern browser)
* Backend: Flask along with SQL Integration, LLM Integration
* Database: SQL-compatible engine (MySQL)

Feasibility analysis confirmed that all technical requirements are within team capability and timeline. Tools and frameworks used are open-source and widely supported. Project management with Agile (Scrum) was deemed feasible for collaborative, incremental progress.

**1.5 Project objectives**

1.5 Project Objectives

* To provide a centralized, responsive dashboard summarizing fitness activity.
* To allow users to track and manage daily workouts, food intake, and sleep.
* To use modern technologies (React, Flask, SQL) in a modular system.
* To ensure security, scalability, and ease of use across all devices. The project follows the Agile development lifecycle with the Scrum framework. Tasks were divided into sprints and evaluated in retrospectives.

**2. Literature study**

Several studies and existing applications underscore the growing importance of health-tracking systems and their impact on wellness and chronic disease prevention. Fit-Track Pro builds upon these foundations by integrating modular components into a single, unified system, addressing the limitations observed in both commercial apps and academic projects.

**[1] Integration of Wearable Sensor Data in Health Applications**

Recent developments in wearable health tracking have enabled real-time monitoring of key physiological metrics such as heart rate, step count, and sleep duration. Studies demonstrate that users who receive regular health feedback are more consistent in physical activity and hydration routines. Fit-Track Pro incorporates similar feedback loops through customizable dashboards and reminders.

**[2] Impact of Data Visualization on Fitness Engagement**

Research has shown that visualizing health data (like calories burned or hydration levels) improves user retention and motivation. Applications such as Google Fit and Fitbit use charts and progress bars to help users interpret their habits. Fit-Track Pro employs **Chart.js** to render user-friendly graphs for workouts, meals, and sleep patterns, allowing for better decision-making.

**[3] Personalized Goal-Setting and Its Effectiveness**

According to multiple studies, users are more likely to achieve health goals when apps offer personalized, adaptive goals rather than static ones. Fit-Track Pro features a dynamic goal module where users can set custom calorie targets, workout durations, and hydration goals. Progress is tracked against these metrics and adjusted based on weekly performance.

**[4] Security and Privacy in Health Platforms**

User trust in fitness applications is closely linked to data privacy. Research recommends secure authentication, encrypted databases, and transparent data policies. Fit-Track Pro implements **JWT-based authentication**, stores user data securely in **MySQL**, and ensures GDPR-compliant data management for privacy and safety.

**[5] AI-Driven Fitness Recommendation Systems(LLM)**

Emerging platforms now integrate machine learning to analyze user behavior and suggest workouts or meals accordingly. While Fit-Track Pro currently offers rule-based feedback, its modular architecture supports future integration of **AI-driven recommendation engines** that will tailor advice based on usage trends and goals.

**[6] Mobile Health App Retention Patterns**

A 2023 study on mHealth apps revealed that apps with more than three active features (e.g., step tracking, diet, water, BMI) had significantly higher user retention. Fit-Track Pro’s holistic offering aligns with this insight, featuring five interconnected modules: **Workout**, **Diet**, **Water**, **BMI**, and **Dashboard Reports**.

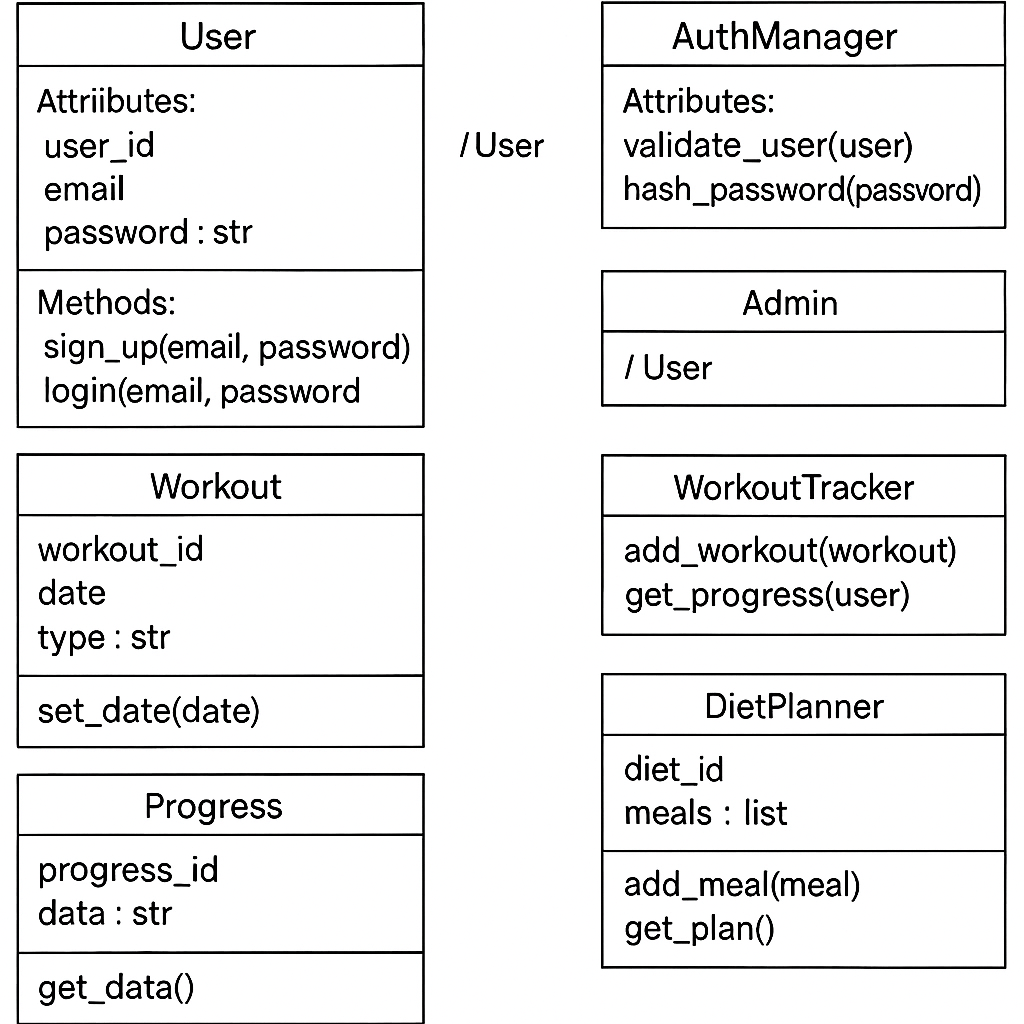
**[7] UX and Accessibility in Health Platforms**

Apps that fail to adapt to different screen sizes or ignore usability design often report poor engagement. React-based platforms are particularly well-suited for mobile-first, responsive applications. Fit-Track Pro, built in **ReactJS**, uses a modular design and accessible layout to ensure consistency across devices and for differently-abled users.

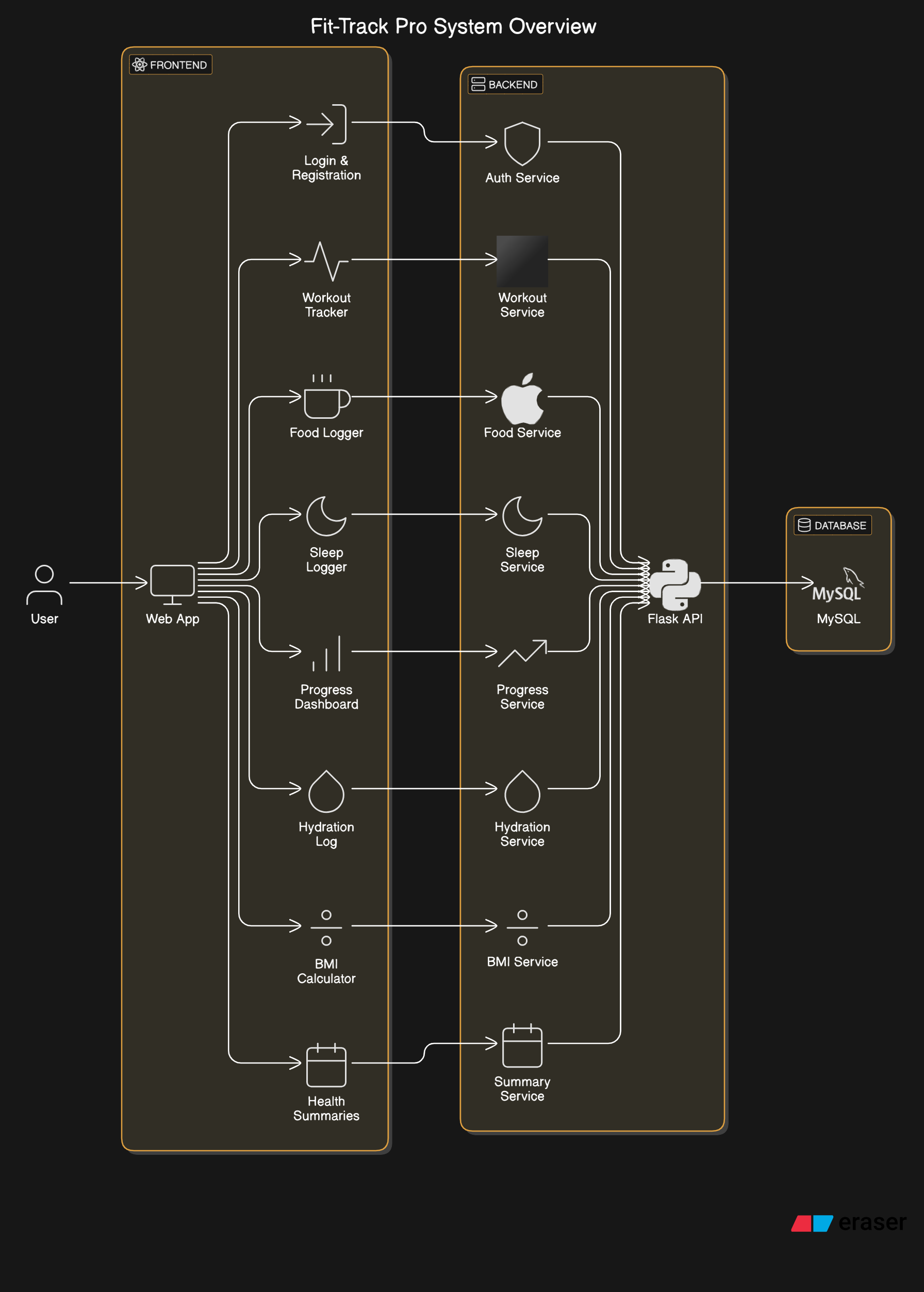
**3. Design details**

**3.1 UML Diagrams**

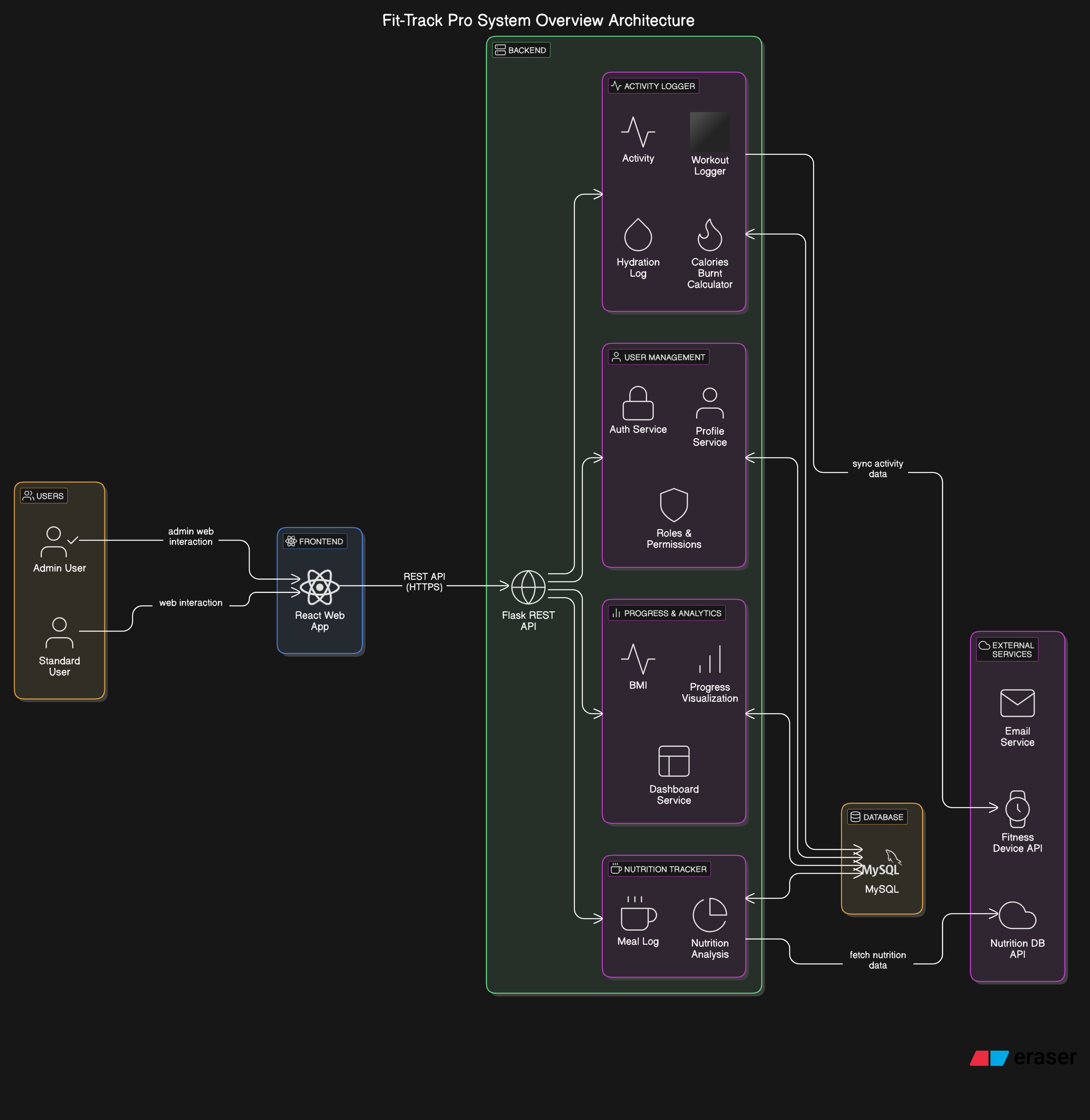
# *1.Class UML Diagram*



### *2. Use Case Diagram*



1. ***State Diagram***



**3.2 Methodology**

The *Fit-Track Pro* system involves multiple user roles interacting seamlessly through a health and fitness management platform. Primary users include individual users (students, fitness enthusiasts, or health-conscious individuals) who use the system to track their workouts, monitor calorie intake and burn, measure diet,workout and visualize overall progress. Users interact via a responsive front-end built using **ReactJS**, while the backend logic and API endpoints are powered by **Flask**.

The system is backed by a **MySQL** database for storing user credentials, fitness logs, health metrics, and feedback data. The application follows an **Agile methodology**, with iterative development cycles and continuous user feedback to improve features such as physical activity monitoring, daily goals, and personalized insights.

Administrators and developers manage the application via secure backend access to ensure data integrity, system reliability, and periodic updates. The system supports modular enhancements, allowing easy integration of wearable devices or machine learning-based health suggestions in future versions. Progress tracking and feedback are visualized using interactive charts powered by **Chart.js** or **Plotly**, enabling users to make informed health decisions and maintain a consistent fitness routine.

**3.3 Hardware and Software Requirements**

**Hardware Requirements**

**Server Hardware (for deployment):**

* **Processor**: Multi-core CPU (e.g., Intel i5/i7 or AMD Ryzen 5/7)
* **RAM**: Minimum 8 GB; Recommended 16 GB for better performance during concurrent usage
* **Storage**: SSD with at least 256 GB capacity to host the database, app files, and user media (profile photos, etc.)
* **GPU (Optional)**: Not mandatory unless future ML models are integrated

**User Devices:**

* **Computers**: Laptops/PCs with internet browsers (Chrome, Firefox, Edge) for accessing the web portal
* **Mobile Devices**: Smartphones/tablets for using a responsive mobile interface (PWA or web view)

**Software Requirements**

**1. Operating System**

* **Server OS**: Ubuntu (preferred) or Windows Server for Flask hosting
* **Client OS**: Windows, macOS, Android, or iOS

1. **Development and Deployment**

* **Frontend Framework**: ReactJS (JavaScript) for dynamic UI
* **Backend Framework**: Flask (Python) for handling backend operations and RESTful APIs
* **Database**: MySQL for user and fitness data storage
* **Version Control**: Git and GitHub for collaborative development
* **IDE**: VSCode or PyCharm

1. **Data Handling & Visualization**

* **Chart Libraries**: Chart.js or Plotly for graphical data representation
* **Data Processing Libraries** (future use): Pandas, NumPy for handling user metrics
* **Form Validation & UI Libraries**: React Hook Form, Bootstrap, or Tailwind CSS

1. **Security & Authentication**

* **Authentication**: Session-based login for secure user sessions
* **Encryption**: HTTPS for data transmission, bcrypt for password hashing
* **Access Control**: Admin panel for managing users and feedback

**4. Implementation details of the Project**

**4.1 Languages/tools/APIs used**

**Languages**

**1. Python**

**Usage**: Backend development, handling server-side logic, and database communication along with LLM.  
**Libraries**:

* **Flask**: Used to build RESTful APIs and manage server routes.
* **SQLAlchemy**: For ORM (Object Relational Mapping) between Flask and MySQL.
* **Pandas/Numpy** *(Future Use)*: For handling fitness data analysis or ML feature expansion.

**2. JavaScript**

**Usage**: Frontend development, creating a responsive and interactive user interface.  
**Libraries/Frameworks**:

* **ReactJS**: Core frontend library used to build a modular, single-page web application.
* **Axios**: Used for sending asynchronous HTTP requests to the Flask backend.
* **Chart.js / Plotly**: For rendering visual graphs and progress charts.

**3. SQL**

**Usage**: Managing structured user and fitness data in relational format.

* **MySQL**: Used as the primary database for storing user credentials, health logs, and feedback.

**4. HTML/CSS**

**Usage**: Structuring and styling the frontend of the application.

* **CSS Libraries**: Tailwind CSS or Bootstrap for UI styling and responsive layouts.

**Tools**

**1. Development Tools**

* **IDE/Editor**: Visual Studio Code (VSCode) for full-stack development.
* **Version Control**: Git for code tracking and collaboration, with GitHub for hosting.

**2. Testing**

* **Manual Testing**: Conducted on forms, login, data entry, and visualization modules.
* **API Testing**: Postman used to test API endpoints between React frontend and Flask backend.

**3. Deployment *(for full deployment readiness)***

* **Web Server**: Gunicorn + Nginx (for Flask deployment on Linux servers).
* **Local Testing**: Flask’s development server used during project demos.

**4. Data Visualization**

* **Chart.js / Plotly.js**: For real-time rendering of health statistics like BMI, calorie logs, water intake, etc.

**APIs**

**1. Internal APIs**

* **Flask REST APIs**: Handle operations like:
  + User registration & login.
  + Logging health and activity data.
  + Fetching progress reports and stats.

**2. Database Management**

* **MySQL (via SQLAlchemy or Flask-MySQL)**: For handling fitness records, user data, and session logs.

**5. Testing and results**

**5.1 Types of testing and validation done**

**Data Testing and Validation:**

* Purpose: Ensure that user data such as workouts, diet logs, sleep hours, hydration, and personal goals are accurate, complete, and in the correct format.

**Methods:**

**Schema Validation**:

* Tool: MySQL schema enforcement.
* Focus: Ensure each table (users, activities, meals, sleep, hydration) has correct fields and data types.

**Range and Format Checks**:

* Tool: Backend validation scripts (Python/Flask).
* Focus: Ensure numerical inputs (e.g., hours slept, calories burned, diet intake) are within expected ranges.

**Consistency Checks**:

* Tool: SQL queries / backend checks.
* Focus: User-specific entries (like workouts and meals) should link to existing user accounts.

**Data Cleaning**:

* Tool: Flask middleware or form validation.
* Focus: Catch and handle missing or null fields in frontend or backend before saving to DB.

**2**. **System Testing**

* Purpose: Validate the entire system workflow from user login to data entry and visualization.

**Methods:**

**Unit Testing**:

* Tools: PyTest (for Flask), Jest (for React).
* Focus: Test isolated frontend/backend functions.

**Integration Testing**:

* Tools: Postman, SQL testing scripts.
* Focus: Ensure frontend forms interact properly with Flask APIs and DB.

**End-to-End Testing:**

* Tools: Selenium or Cypress.
* Focus: Simulate full user journey: login → add data → view graphs → get tips.

**User Acceptance Testing (UAT**):

* Implementation: Demo to classmates/mentors.
* Focus: App is user-friendly, reliable, and meets the Agile sprint goals.

**Performance Testing**:

* Tools: JMeter or simple load scripts.
* Focus: Ensure the system handles multiple concurrent users logging health metrics.

**Security Testing**:

* Focus: Basic tests for SQL injection, session expiry, role-based access.

**3. Continuous Testing and Validation:**

**Monitoring**:

* Tools: Debug logs, API error tracking.
* Focus: Monitor system response time and DB/API errors during usage.

**Feedback Loops**:

* Implementation: Collect peer feedback during demo/presentations.
* Focus: Identify UI/UX improvements, missing features, and bugs.

**Feature Iteration**:

* Implementation: Based on feedback, rework visualizations, optimize goal logic, improve tip generator.

**5.2 Traceability Matrix**

This traceability matrix links user stories with their corresponding requirements, design elements, implementation artifacts, and test cases. It is tailored for an Agile-based Fitness App developed using HTML, CSS, Python, SQL, and React.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | User Story | Requirement | Design Element | Implementation | Test Case(s) | Status |
| US01 | As a user, I want to sign up and log in securely. | User authentication system | Login/Signup forms (HTML/CSS), Backend API (Python), SQL Users table | `login.html`, `signup.html` | TC01: Valid login TC02: Invalid login | Completed |
| US02 | As a user, I want to track my workouts daily. | Workout data entry and tracking | Workout form UI, React component for input, SQL workouts table | `trackWorkout.js`, `workout.css`, `add\_workout.py` | TC03: Add new workout,diet list,sleep analysis TC04: Empty fields validation | Completed |
| US03 | As a user, I want to view my progress on a dashboard. | Progress dashboard with historical data | Dashboard layout (HTML/CSS), React for dynamic data fetch | `dashboard.html`, `progress.js`, `progress.sql` | TC05: Fetch correct data TC06: Data format check | In Progress |
| US04 | As a user, I want to get basic fitness suggestions. | Static suggestions based on goals | HTML page with tips, Python function for condition-based suggestions | `suggestions.html`, `fitness\_tips.py` | TC07: Correct tips shown TC08: Goal matching logic | Done |
| US05 | As a user, I want to ask health and fitness questions and get intelligent responses | Integration of a Large Language Model for health-related Q&A | LLM google/gemma-2b-it integration | ‘chat\_api.py’ | TC09: Query returns relevant answer  TC10: Handles unexpected questions gracefully | Done |

**5.3 Deployment**

We have deployed our project in git, which can be accessed through a GitHub Repositary also looking to deploy at vercel which is a cloud platform optimized for frontend frameworks like React.

**GitHub Link** - <https://github.com/Aneesh997/Fitness-Tracker-Website.git>

1. **Conclusion**

Fit-Track Pro stands as a successful outcome of our Agile Software Engineering project, embodying the principles of iterative development, user-centered design, and seamless technology integration. By combining a modern ReactJS frontend with a Flask backend and MySQL database, the system delivers a comprehensive and responsive platform for health and fitness tracking. The project showcases novelty not just in its technical architecture, but in its approach to personalized wellness management, offering real-time activity tracking, meal logging, LLM for smart responses and progress visualization in a single application.

While the application meets core functionality goals, limitations such as data accuracy, personalization depth, and integration complexity remain areas for future work. Nevertheless, the iterative Agile approach has laid a strong foundation for continuous improvement. Future development phases can extend Fit-Track Pro’s utility through features like telehealth integration, AI-based health recommendations, and adaptive user interfaces.

Overall, Fit-Track Pro exemplifies how Agile methodologies can drive innovation in health tech, producing a scalable and user-friendly solution that supports individuals on their fitness journeys.

**References**

* Lee, Y., Kim, J., & Park, S. (2022). Design and Development of a Mobile Fitness Tracking Application Using Agile Methodology. *International Journal of Computer Applications*, 184(23), 15–21. <https://doi.org/10.5120/ijca2022912021>
* Hossain, M. S., & Muhammad, G. (2016). Cloud-assisted Industrial Internet of Things (IIoT) – Enabled framework for health monitoring. *Computer Networks*, 101, 192–202. <https://doi.org/10.1016/j.comnet.2016.01.009>
* Patel, S., Park, H., Bonato, P., Chan, L., & Rodgers, M. (2012). A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(1), 21.

<https://doi.org/10.1186/1743-0003-9-21>

* Agarwal, P., & Aggarwal, P. (2021). An Overview of Fitness Tracking Applications: Trends and Challenges. *International Journal of Health and Fitness Technologies*, 3(1), 12–18.
* Puri, R., & Joshi, S. (2023). Data-Driven Personalized Health Monitoring Using Mobile Apps and Cloud Integration. *HealthTech Journal*, 5(3), 88–96.

**Appendix A**: Screenshots

