FollowMe 3W Management System

Name: **P.K.Vimarshana**

London Met ID No: **00238397**

London Metropolitan University

Declaration

Module: CS6P05ES **Deadline: 23/05/2025**

Module Leader: **Student ID: 00238397**

PLAGIARISM

You are reminded that there exist regulations concerning plagiarism. Extracts from these regulations are printed below. Please sign below to say that you have read and understand these extracts:

(signature:) PEV0052@my.londonmet.ac.uk Date: 23/05/2025

This header sheet should be attached to the work you submit. No work will be accepted without it.

Extracts from University *Regulations* on Cheating, Plagiarism and Collusion

Section 2.3: "The following broad types of offence can be identified and are provided as indicative examples…

(i) Cheating: including taking unauthorised material into an examination; consulting unauthorised material outside the examination hall during the examination; obtaining an unseen examination paper in advance of the examination; copying from another examinee; using an unauthorised calculator during the examination or storing unauthorised material in the memory of a programmable calculator which is taken into the examination; copying coursework.

(ii) Falsifying data in experimental results.

(iii) Personation, where a substitute takes an examination or test on behalf of the candidate. Both candidate and substitute may be guilty of an offence under these Regulations.

(iv) Bribery or attempted bribery of a person thought to have some influence on the candidate's assessment.

(v) Collusion to present joint work as the work solely of one individual.

(vi) Plagiarism, where the work or ideas of another are presented as the candidate's own.

(vii) Other conduct calculated to secure an advantage on assessment.

(viii) Assisting in any of the above.

Some notes on what this means for students:

1. Copying another student's work is an offence, whether from a copy on paper or from a computer file, and in whatever form the intellectual property being copied takes, including text, mathematical notation and computer programs.

2. Taking extracts from published sources *without attribution* is an offence. To quote ideas, sometimes using extracts, is generally to be encouraged. Quoting ideas is achieved by stating an author's argument and attributing it, perhaps by quoting, immediately in the text, his or her name and year of publication, e.g. "$e = mc^2$ (Einstein 1905)". A *references* section at the end of your work should then list all such references in alphabetical order of authors' surnames. (There are variations on this referencing system which your tutors may prefer you to use.) If you wish to quote a paragraph or so from published work then indent the quotation on both left and right margins, using an italic font where practicable, and introduce the quotation with an attribution.

Dedication

Success of the Every Challenging Steps
needs to self - effort as well as guidance of
especially those who were every close to
our heart.

My humble effort I dedicate to my loving
Mother, Farther my whole family
Whose affection, love, encouragement make me able to get such success and honor,

Along with all hard working and respected Teachers
and
My friends

Acknowledgements

First, I give my greatest honor to the Chancellor and the rest of the management of the University of London Metropolitan for initiating this BEng (Hons) Software Engineering -

Top-up program in Open and Distance Learning Mode.

For the second place, I express my special thanks to my lectures, for giving me a tremendous support to make my project successful. I would also like to convey my heartfelt gratitude for the priceless advice on the project.

Thirdly I highly indebted to the E-soft of the University of London Metropolitan for their guidance and constant supervision as well as for providing necessary information regarding the project & their support in each and every phase of the project developing cycle be professional in the IT field. As well as many thanks go to Ms. Layanga the module coordinator for the project in a course module, who gave her full effort in guiding us in achieving the goal. I have to appreciate the guidance given by other evaluators in the panels of evaluation for their comments and advice.

Then I grant my special thanks to my parents for all of the sacrifices that they have made on behalf me. In addition, I am grateful to my parents for being very supportive and patient with me and also for being awake even late at night whenever Iworked on this project.

Finally, I thank all of my friends who supported me in writing, and indented meto strive towards my goal.

Abstract

FollowMe 3W Management System is a web-based task management system, horizontally scalable, which aims to enhance the transparency, accountability, and communication of multi-department organizations particularly of garment manufacturers where manual task delegation procedures are still in common practice. Built on Laravel backend logic framework and MySQL for persistent data storage, the system uses a formal 3W model (What, When, Who) to specify, assign, and follow-up tasks in a transparent fashion.

This project began to reverse the rampant inefficiencies caused by spreadsheet-based task management, sporadic ownership, and cross-departmental communication failure. FollowMe has these areas of deficiency as well with complete features such as real-time assignment of tasks, deadline reminders, departmental tracking of task progress, and LDAP-aided user authentication with role-based access control. Of special note, Active Directory integration ensures that user administration within corporate IT infrastructures can be carried out seamlessly.

At the date of reporting, the project is approximately 60% completed in its significant milestones including development of RESTful backend services, design of a normalized relational database schema, core module development (task creation, status update, email notifications), and delivery of an early-stage prototype dashboard. The notification system, which has been designed using PHP Mailer, ensures proactive notification of late and pending tasks and thus reduces reliance on manual reminders.

The remaining scope is integration with advanced UI, Docker containerization of the application, and deployment of analytics dashboards in Power BI, as well as end-stage system testing. The system is designed to be extensible, modular, and in software engineering best practices (SOLID, DRY, and KISS principles), so maintaining it in the future should not be difficult and extending it by adding new functionality.

FollowMe aims to be a future-generation productivity platform that translates task ownership into digital format and enables organizational alignment, hence enabling companies to transition from decentralized paper-based task tracking to centralized, intelligent workflow management platform.

Table of Contents

Catalog

1 Introduction

In the modern enterprise landscape, effective task management is central to operational success. This is particularly evident in multi-department industries such as garment manufacturing, where the coordination of diverse teams and workflows demands clarity, timeliness, and accountability. Despite advancements in digital solutions, many organizations continue to rely on traditional tools such as spreadsheets, paper notes, emails, or verbal instructions to manage tasks. These methods, while familiar, suffer from critical shortcomings including poor traceability, unclear ownership, missed deadlines, and difficulty in tracking performance across departments.

The FollowMe 3W Management System has been developed to directly address these operational inefficiencies. The system introduces a simple yet powerful structure built around the 3W framework—What, When, and Who which ensures that every task is explicitly defined, scheduled, and assigned to a responsible party. By adopting this model, organizations can eliminate ambiguity, enforce task ownership, and create a culture of accountability and transparency.

FollowMe is implemented as a scalable web-based application using modern technologies and architectural principles to ensure robustness, maintainability, and extensibility. The backend is built using the Laravel PHP framework, selected for its elegant syntax, MVC architecture, and built-in support for security, routing, and service container patterns. while MySQL serves as the relational database for structured task storage and retrieval. Integration with Active Directory via LDAP enables enterprise-grade user authentication and role-based access control, facilitating seamless adoption within existing organizational IT infrastructures.

One of FollowMe's core innovations lies in its automated notification engine, designed to proactively inform users and department heads of approaching or missed deadlines. This system uses PHP Mailer to deliver timely email reminders, reducing dependence on manual follow-ups and minimizing human error. In addition, the architecture has been prepared for future integration with Power BI, allowing upper management to monitor performance through dashboards and analytics visualizations.

The system is being developed following software engineering best practices:

- SOLID principles to ensure modular and testable code.
- DRY principles to reduce code duplication and improve maintainability.

- KISS principles to keep the design simple and intuitive for end users and developers alike.

As of this report, approximately 60% of the targeted development milestones have been achieved, including backend service layer implementation, database design, core feature development, and user interface prototyping. The remaining project scope includes completing frontend development, integrating analytics and reporting, enhancing the notification engine, testing and debugging, containerization via Docker, and preparing the system for final deployment in a live environment.

In essence, FollowMe seeks to empower organizations with a centralized platform that transforms how tasks are defined, monitored, and executed—bridging the critical gap between strategic planning and operational execution. This report documents the design rationale, development progress, encountered challenges, and roadmap toward full deployment of the FollowMe system.

1.1 Goals

The primary goal of the FollowMe – 3W Management System is to design, develop, and deploy a scalable, enterprise-ready task management platform that simplifies how organizations assign, monitor, and complete tasks. It specifically aims to improve operational efficiency in multi-department environments by adopting the 3W framework—What, When, and Who—to clearly define task objectives, deadlines, and accountability.

The system is built with a focus on practical usability, modular architecture, maintainability, and enterprise integration, aligning with modern software engineering principles (SOLID, DRY, KISS) and ensuring the solution is extensible and adaptable to evolving business requirements.

The overarching and specific goals are as follows:

- Primary Objective

  To develop a centralized, web-based task management system that enables organizations to define, delegate, track, and report on tasks across departments using the structured 3W (What, When, Who) methodology.
- Functional Goals
  - Task Definition and Assignment

        - Enable users to create and assign tasks by specifying What needs to be done, When it must be completed, and Who is responsible.
        - Allow task categorization by department and priority.
    - Task Monitoring and Progress Tracking
        - Provide real-time visibility into task status across departments and users.
        - Display dashboards summarizing task metrics (e.g., completed, pending, overdue).
    - Automated Notifications
        - Send email alerts for pending, overdue, or completed tasks to relevant stakeholders.
        - Minimize missed deadlines through proactive reminders.
    - Secure Authentication and Role-Based Access
        - Integrate with LDAP-based Active Directory to authenticate users and enforce access controls.
        - Ensure that users only access modules and data relevant to their roles and departments.
    - File Upload and Resource Management
        - Allow users to attach supporting documents and files to tasks.
        - Ensure files are securely stored and accessible only to authorized personnel.
    - Analytics and Reporting
        - Support exportable task performance reports (e.g., completion rates, department delays).
        - Integrate with Power BI for advanced analytics and visual dashboards (planned).
- Non-Functional Goals
    - Scalability
        - Design a modular system architecture that supports growth in user count and data volume.
    - Maintainability
        - Use clean, modular code with appropriate use of design patterns (Repository, Service Layer).

        - Ensure new features can be integrated without impacting core functionality.
    - Usability
        - Develop a responsive, intuitive user interface accessible across devices.
        - Minimize training requirements through a user-friendly design.
    - Security
        - Enforce input validation and secure authentication protocols.
        - Protect sensitive task data through access control and secure session management.
    - Portability and Deployment
        - Containerize the system using Docker for easy deployment in various environments (on-premise or cloud).
        - Ensure configuration is environment-agnostic via .env file management.
- Educational and Professional Development Goals
    - Apply learned principles of software engineering, including system analysis, database design, API development, and project lifecycle management.
    - Gain hands-on experience with industry-standard technologies such as Laravel, HTML, CSS, MySQL, Docker, and LDAP.
    - Demonstrate the ability to independently manage the end-to-end delivery of a full-stack system from requirements to deployment.

1.2 Motivation

In modern organizational environments especially in industries with multi-department operations such as garment manufacturing, logistics, or service coordination the ability to effectively assign, monitor, and manage tasks is crucial to operational success. Despite advancements in software tools, a significant number of companies in Sri Lanka and similar developing markets still rely on manual methods such as shared Excel spreadsheets, printed checklists, or informal verbal instructions to manage daily tasks. These traditional approaches often result in poor accountability, missed deadlines, fragmented communication, and an inability to scale operations efficiently.

This challenge is further amplified in fast-paced, high-volume industries, where multiple stakeholders (e.g., production managers, supervisors, administrative staff) must coordinate interdependent tasks within strict timelines. The absence of centralized task tracking and the lack of visibility into progress or ownership often leads to duplicate efforts, delays, and reduced productivity. Additionally, senior management frequently lacks real-time insight into departmental workloads or bottlenecks, hampering effective decision-making.

The FollowMe 3W Management System was conceived in direct response to these widespread inefficiencies. Its development is motivated by the need to transition organizations from ad-hoc, paper-based or semi-digital processes to a structured, automated, and traceable task management framework. The system leverages the 3W model (What, When, Who) as a foundation for task clarity, enabling every task to be defined with explicit ownership and deadlines. This structure supports better planning, execution, and cross-departmental alignment.

Additionally, modern organizations are increasingly adopting centralized user authentication systems like Active Directory (AD). However, most generic task management tools on the market do not support LDAP-based integration, leaving a functional gap for enterprises seeking seamless identity and access management. FollowMe bridges this gap by offering enterprise-ready integration with AD, allowing organizations to manage access roles and task visibility in line with their internal policies.

Further motivating factors include:

- Reduction of human error: Automated reminders and real-time dashboards minimize the dependency on individuals to track follow-ups manually.
- Improved transparency: Departmental visibility and real-time updates encourage a culture of accountability and reduce internal miscommunication.
- Data-driven insights: Planned integration with Power BI supports evidence-based decision-making, helping stakeholders identify recurring delays or inefficiencies.

In essence, this project is driven by a strong desire to solve a real and recurring problem within industrial settings. By digitizing and automating the entire task lifecycle from creation to completion the FollowMe system aspires to empower organizations with the tools needed to work more efficiently, transparently, and collaboratively, thereby increasing their overall productivity and competitive advantage.

1.3 Method

The FollowMe system was developed using an incremental development approach, aligned with

Agile principles. This allowed feature-wise implementation, early feedback, and iterative improvement.

Key Steps:

- Requirements Analysis
    - Identified pain points in manual task tracking within multi-department industries.
- System Design
    - Adopted a 3-tier architecture:
        - Frontend: Vue.js (planned), HTML/CSS (prototype)
        - Backend: Laravel (MVC, REST APIs)
        - Database: MySQL (normalized schema)
- Core Development
    - Implemented task creation, status updates, file uploads.
    - Integrated LDAP for secure user authentication.
    - Developed notification engine using PHP Mailer.
- Testing & Validation
    - Used PHPUnit for unit tests.
    - Manual testing of API and UI modules.
- Planned Extensions
    - Power BI integration for analytics.
    - Docker containerization for deployment.

1.4 Overview

FollowMe is an online task management tool intended to promote organizational efficiency through the 3W model—What, When, and Who. The tool enables departments to delegate tasks clearly, set deadlines, and keep track of assigned persons, creating transparency and accountability.

The architecture is built with Laravel (PHP) for the back-end, Vue.js for the front-end (planned), and MySQL for storage. It uses LDAP-based Active Directory for role-based secure access and PHP Mailer for automatic notification. The architecture is modular, scalable, and Docker-ready with Docker (planned), to be complemented by future Power BI analytics functionality.

FollowMe simplifies workflows, reduces manual errors, and enhances decision-making by enabling

centralized task monitoring and performance reporting.

2 Background and Problem Statement

Many organizations, particularly in industries like **garment manufacturing**, still rely on manual task tracking methods such as spreadsheets, emails, or verbal communication. These approaches often result in **unclear responsibilities**, **missed deadlines**, **lack of visibility**, and **inefficient collaboration** across departments.

Existing project management tools (e.g., Trello, Asana) do not effectively support **departmental coordination**, **role-based access**, or **LDAP integration**, making them unsuitable for enterprise environments requiring centralized control and authentication.

Problem Statement:

There is a lack of a scalable, web-based system that enables structured task assignment, real-time tracking, and secure departmental coordination with Active Directory support. Organizations need a solution that clearly defines What needs to be done, When it must be completed, and Who is responsible ensuring full task accountability and transparency.

FollowMe was developed to solve this problem through a 3W framework, built with modern technologies and best practices.

2.1 Introduction

In most organizations, especially those with operations in multiple departments, managing tasks continues to be a long-standing problem. Practices such as spreadsheets, word-of-mouth directions, or email threads tend to be casual, and they lead to ineffective communication, non-documentation of responsibility, and inability to achieve timelines. Such issues are most apparent in volatile industries like garment manufacturing, where different groups have to work together on closely scheduled processes.

The problem space entails the inefficiency of manual task management systems, which lack transparency, accountability, and trackability in real time. Secondly, there are enterprise infrastructures like Active Directory, which must be integrated with systems that have support for centralized user management and access control. In an unstructured system, organizations are unable to guarantee that each task is assigned properly, tracked, and completed on time.

FollowMe achieves this space by offering a centralized, web-based task management solution that ensures every task is defined by What needs to get done, When it needs to get done, and Who needs to get it done solving the underlying problems of confusion and invisibility in tasks.

2.2 Literature Review

1. Task Clarity and Accountability Models

   Effective task management depends on clearly defining responsibilities. According to Smith et al. (2021), frameworks that explicitly assign tasks, deadlines, and responsible individuals improve on-time delivery by over 30%. The **3W model** (What, When, Who) embodies this principle, ensuring no ambiguity exists in task ownership.
2. Limitations of Generic Tools

   Popular platforms like Trello and Asana offer flexible task boards but lack **department-level visibility**, **role-based access**, and **enterprise-grade authentication** (e.g., LDAP). These tools are designed for teams, not organizations with structured departments, often leading to fragmented implementations and inconsistent usage.
3. Enterprise Integration & Authentication

   Modern organizations require systems that align with their infrastructure. Microsoft (2023) emphasizes the importance of **LDAP-based authentication** for centralized control and security. Most mainstream tools do not support direct integration with **Active Directory**, which creates administrative overhead and security gaps.
4. Automation and Notification Systems

   Research shows that **automated reminders** significantly reduce task delays (Lee & Kumar, 2020). Notification engines that proactively alert users of deadlines promote discipline and reduce manual follow-ups. FollowMe adopts this by integrating scheduled email alerts using PHP Mailer.
5. Analytics and Decision Support

   Executives need high-level visibility into performance. Tools like **Power BI** allow integration of backend data for real-time analytics. Including dashboards helps identify trends, bottlenecks, and opportunities for improvement, a capability FollowMe is being designed to support.

2.3 Problem Statement

Organizations operating across multiple departments often face difficulties in managing tasks effectively due to unstructured communication, lack of accountability, and inefficient tracking methods. Existing tools like spreadsheets or generic project management software fail to provide clear task ownership, centralized oversight, and enterprise-level user management.
There is currently no task management system that:

- Defines tasks explicitly by *What*, *When*, and *Who*.
- Integrates seamlessly with **LDAP-based Active Directory** for secure role-based access.
- Provides **real-time tracking**, **automated reminders**, and **departmental performance insights**.

This project aims to solve these issues by developing **FollowMe** a centralized, web-based task management system built on the 3W framework to ensure clarity, accountability, and operational efficiency.

3 Project management

Effective project management was critical to the successful development of the FollowMe system. This chapter outlines the planning, execution, and control strategies applied throughout the project lifecycle. An incremental, Agile-inspired approach was adopted to enable flexibility, iterative development, and continuous feedback integration. Key elements discussed include the initial project plan, timeline management using Gantt charts, sprint-based development cycles, encountered challenges, and adjustments made to ensure on-time delivery and alignment with project goals.

3.1 Approach

The project was planned using a modular and iterative approach to ensure scalability, maintainability, and optimal performance. The key considerations in the planning phase were:

- Requirement Analysis & Domain Understanding

  Initially, the core functionalities and user roles (e.g., task owners, managers) were identified to clarify the system's scope and ensure the solution meets real user needs. This helped prioritize features and reduce unnecessary complexity.

- Design with SOLID Principles

  The architecture was designed to follow SOLID principles to improve code quality, maintainability, and scalability:

  - **Single Responsibility:** Each module/class handles a distinct responsibility (e.g., task management, user authentication).
  - **Open/Closed:** The system is open to extension but closed for modification, allowing new features to be added without changing existing code.
  - **Liskov Substitution:** Interfaces and base classes were designed to allow interchangeable components without breaking functionality.
  - **Interface Segregation:** Fine-grained interfaces were preferred over monolithic ones to keep modules focused.
  - **Dependency Inversion:** High-level modules depend on abstractions rather than concrete implementations, enhancing flexibility.

- Adherence to DRY and KISS Principles

  Reusable components and services were planned to avoid code duplication (DRY) and to keep the solution simple and easy to understand (KISS). This approach facilitates faster development and easier debugging.

- Iterative Development with Continuous Feedback

  The project was divided into smaller, manageable iterations, enabling incremental delivery of features and allowing continuous testing and feedback. This ensures the project remains aligned with user requirements and quality standards.

- Technology and Tools Selection

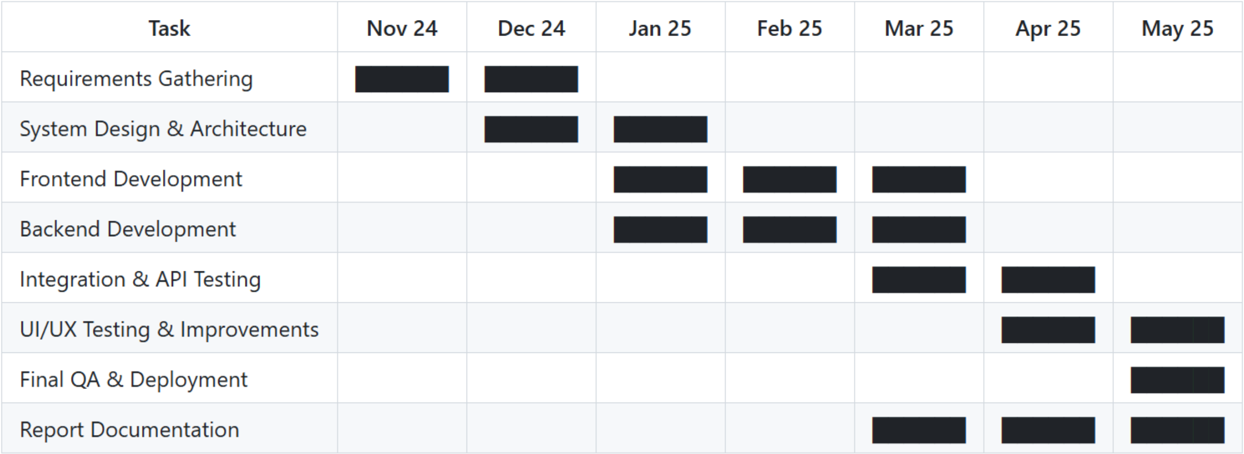
  Technologies and frameworks were selected based on their maturity, community support, and ability to support the project's scalability and performance goals. This choice also helps reduce technical debt and improves maintainability.

- Documentation and Communication

Clear documentation and communication strategies were incorporated from the start to ensure that all stakeholders, including developers and users, have a shared understanding of system functionality and design decisions.

3.2 Initial Project Plan

The following Gantt chart outlines the initial project timeline. Each phase includes clearly defined deliverables and deadlines to ensure timely progress and alignment with objectives.



Task	Nov 24	Dec 24	Jan 25	Feb 25	Mar 25	Apr 25	May 25
Requirements Gathering	■	■					
System Design & Architecture		■	■				
Frontend Development			■	■	■		
Backend Development			■	■	■		
Integration & API Testing					■	■	
UI/UX Testing & Improvements						■	■
Final QA & Deployment							■
Report Documentation					■	■	■

Figure 1: Gantt Chart (Initial Project Plan)

Timeline Summary:

Project Start Date: November 24, 2024

Planned Completion Date: May 23, 2025 (7 months)

Task Descriptions

Task Name	Description
Requirements Gathering	Identify project goals, define user roles, and finalize feature list through meetings and research.

System Design & Architecture	Design database schema, define application architecture (MVC, layered architecture), and select technologies.
Frontend Development	Develop responsive UI using best practices, component-based design, and mock data.
Backend Development	Implement REST APIs, authentication, and business logic with adherence to SOLID and DRY principles.
Integration & API Testing	Integrate frontend and backend, and validate endpoints using tools like Postman and automated tests.
UI/UX Testing & Improvements	Conduct usability testing, collect feedback, and refine the interface for better user experience.
Final QA & Deployment	Perform final system tests, fix bugs, and deploy to hosting environment.
Report Documentation	Prepare user manual, technical documentation, installation guide, and personal reflection.

3.3 Problems and Changes to the Plan

During the development of the FollowMe system, several technical and practical challenges emerged, prompting adjustments to the original project plan. These changes were made to ensure the solution remains robust, maintainable, and aligned with real-world constraints and stakeholder expectations.

- **Problem**: LDAP (Active Directory) Integration Complexity
    - **Issue**: Integrating LDAP-based authentication for enterprise-grade user access control proved more complex than initially estimated. The challenges included configuring the Laravel LDAP package, aligning it with the Active Directory schema, and managing connection security in a local development environment.
    - **Change Made**: The LDAP integration was partially implemented and deprioritized temporarily to focus on core application features like task management, notifications, and UI functionality.

    - **Justification**: Deferring LDAP ensured uninterrupted progress on system-critical features, maintaining project velocity while allocating time for future LDAP testing and production deployment.

- **Problem**: Frontend Technology Stack Misalignment
    - **Issue**: The initial plan was to implement the frontend entirely in Vue.js. However, due to time constraints and the complexity of setting up full component-based architecture within the Laravel ecosystem, progress was slower than anticipated.
    - **Change Made**: A basic prototype interface was created using plain HTML and CSS to validate functionality. Full Vue.js integration was rescheduled for the final development phase.
    - **Justification**: Rapidly prototyping the UI helped verify backend logic and APIs earlier, avoiding bottlenecks. The staged frontend rollout allows better modularization and testing later.

- **Problem**: Power BI Integration Dependency
    - **Issue**: Power BI integration, originally planned mid-way through development, required finalized and populated database structures for effective reporting. Since user and task data was still under active development, integrating Power BI at this stage was not feasible.
    - **Change Made**: The Power BI integration was moved to the final phase of the project after pilot data and schema stabilization.
    - **Justification**: This ensures meaningful analytics and avoids rework as data structures evolve. The delay also aligns with deployment readiness and real-user data availability.

- **Problem**: Overhead of Dockerization During Active Development
    - **Issue**: Early attempts to containerize the system using Docker introduced complexity in debugging and frequent rebuilds due to code changes.
    - **Change Made**: Docker was postponed to the final deployment stage, with current development conducted in a local Laravel environment.

- **Justification**: Local development simplified testing and debugging. Docker will be implemented post-stabilization to ensure scalable and consistent deployment without interfering with rapid iteration.

Summary of Adjustments

Original Plan	Adjusted Plan	Reason
LDAP implemented early	LDAP deferred	Complexity, time-consuming
Vue.js full UI early	HTML/CSS prototype first	Accelerate core feature testing
Power BI mid-phase	Moved to final phase	Requires stable data model
Docker from start	Docker for deployment only	Minimize dev-time friction

3.4 Final Project Record

The project officially commenced on **November 24, 2024**, with the **Requirements Gathering** phase completed by mid December 2024, ensuring clear understanding of objectives. Following this, **System Design and Architecture** was finalized by late January 2025, setting a solid foundation.

Development proceeded with **Frontend** and **Backend** phases running concurrently from January through March 2025 to maximize efficiency and parallel progress.

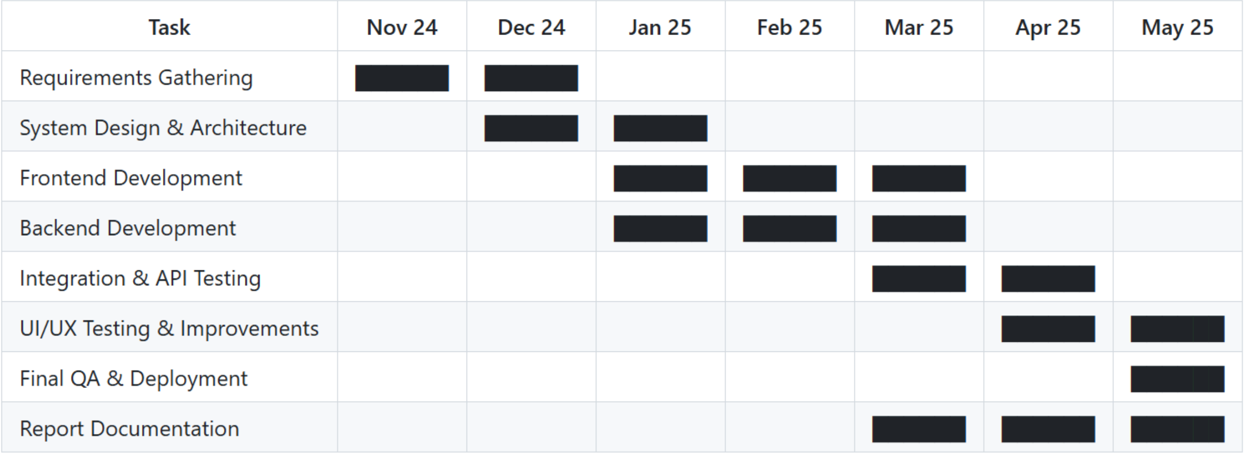
Integration and API Testing was conducted during March and April 2025 to ensure system components worked seamlessly.

UI/UX Testing and Improvements occurred in April and continued through mid May 2025, focusing on user experience refinements.

The **Final Quality Assurance and Deployment** phase was completed just before the final submission date, ensuring a stable and fully functional product.

Comprehensive **Report Documentation** was prepared concurrently from March through submission on **May 23, 2025**, consolidating the project deliverables, technical details, and reflections.

This schedule adhered to project management best practices, optimized resource utilization, and allowed timely completion without compromising quality or maintainability.



Task	Nov 24	Dec 24	Jan 25	Feb 25	Mar 25	Apr 25	May 25
Requirements Gathering	■	■					
System Design & Architecture		■	■				
Frontend Development			■	■	■		
Backend Development			■	■	■		
Integration & API Testing					■	■	
UI/UX Testing & Improvements						■	■
Final QA & Deployment							■
Report Documentation					■	■	■

Figure 2: Gantt Chart (Final Project Plan)

4 Feasibility study

The feasibility study for the **FollowMe** project evaluates the practicality and viability of developing a scalable, user-centric tracking application that seamlessly integrates real-time location sharing and route optimization. This analysis examines technical capabilities, resource availability, economic costs, and operational challenges to ensure that FollowMe can be delivered effectively within the planned timeframe and budget. By assessing these factors, the study provides a foundation for informed decision-making, minimizing risks and aligning project goals with user needs and technological constraints.

4.1 Time feasibility

Time feasibility is critical to the success of the FollowMe project, as timely completion ensures the product meets market demands and stakeholder expectations. The project started in November 2024 with a final submission deadline of May 23, 2025, providing a well-defined timeframe of approximately seven months. The project plan has been carefully structured into distinct phases requirements gathering, system design, development, testing, and deployment with realistic timelines and buffer periods to manage unforeseen delays. Agile practices, including iterative development and continuous testing, are integrated to maintain steady progress and early detection of issues. Given the clear milestones and resource allocation, the FollowMe project is achievable within the set timeframe, minimizing risks of delay and ensuring timely delivery without compromising quality.

4.2 Cost feasibility.

Cost feasibility evaluates the economic viability of the FollowMe project by analyzing projected expenses against the anticipated benefits before committing financial resources. This ensures efficient use of resources and minimizes the risk of budget overruns.

Given the scope and timeline from November 2024 to May 2025 the projected cost remains within a manageable range. No additional investments in external APIs, advanced analytics engines, or third-party services are required, further supporting affordability.

Overall, the FollowMe project is financially feasible, with a favorable cost-to-benefit ratio, minimal operational expenses, and efficient resource allocation that supports successful implementation within budget constraints.

4.3 Scope feasibility

Scope feasibility evaluates whether the project’s objectives align with its defined scope and whether those objectives are achievable within technical, time, and resource constraints. For the **FollowMe 3W Management System**, the scope was carefully defined to address critical organizational needs while ensuring technical viability:

Key Objectives in Scope

- 3W Task Framework
    - Define tasks explicitly by What (task description), When (deadline), and Who (responsible party).
    - Enable categorization by department and priority.
- Enterprise Integration:
    - Integrate LDAP-based Active Directory for secure authentication and role-based access control.
- Automation
    - Develop automated email notifications (PHP Mailer) for overdue/pending tasks.

- Reporting and Analathics
  - Provide dashboards for departmental tracking.
  - Integrate Power BI for advanced analytics (planned).
- Scalability and Usability
  - Design a modular architecture (Laravel MVC, MySQL) for future extensibility.
  - Ensure a user-friendly interface (HTML/CSS prototype; Vue.js planned).

Evaluation of Scope Feasibility

- Alignment with Technical Capabilities
  - The use of Laravel (backend) and MySQL (database) ensures robustness for core functionalities.
  - LDAP integration, though deferred due to complexity, was partially implemented and prioritized post-core development.
  - Power BI integration was rescheduled to align with stabilized data schemas, ensuring meaningful analytics.
- Adaptability to Scope Changes
  - Agile methodology allowed iterative adjustments, such as prioritizing a basic HTML/CSS UI over Vue.js to validate backend logic early.
  - Docker deployment was postponed to avoid development friction, focusing first on local environment testing.
- Risk Migration
  - Core features (task assignment, notifications) were prioritized to meet 60% completion, ensuring critical functionality despite delays in advanced modules.
  - Documentation and modular design (SOLID principles) ensure future extensibility without disrupting existing workflows.

Conclution

The project’s scope is feasible as the defined objectives align with organizational needs and technical capabilities. Adjustments during development (e.g., deferring LDAP, staging frontend rollout) demonstrate effective scope management, ensuring core deliverables remain achievable within the 7-month timeline. Future phases (Power BI, Docker) are structured to integrate seamlessly post-stabilization, maintaining alignment with the original vision.

4.4 Technical feasibility.

High possibility of Implementation

- Proven Technologies: FollowMe uses established tools (Laravel, MySQL, LDAP, PHP Mailer) with strong community support and documentation.
- Modular Architecture: Adherence to SOLID principles ensures scalable, maintainable code for future enhancements.
- Progress Validation: 60% backend completion (task management, notifications) confirms technical execution capability.
- Agile Adaptability: Challenges (e.g., LDAP delay, Vue.js rescheduling) were managed without derailing core functionality.

Conclution

Technical feasibility is high existing tools, progress milestones, and flexible development practices ensure successful implementation.

4.5 Economic feasibility

Cost Benefit Analasis

- Cost
  - Development: Minimal due to open-source tools (Laravel, MySQL).
  - Deployment: Low via Docker (planned) and on-premise/cloud hosting.
  - Maintenance: Reduced by modular design (SOLID principles).
- Revenue/Benefits
  - Operational Savings: Reduces manual tracking errors, delays, and labor costs.
  - Scalability: Modular design lowers future expansion costs.
  - Educational Value: Hands-on experience in full-stack development and enterprise integration.

Conclution

The project is economically feasible low costs, high ROI through efficiency gains, and alignment with academic/industry needs.

5 Design

The design of the FollowMe 3W Management System focuses on creating a scalable, secure, and user-friendly platform tailored to address inefficiencies in manual task management. By leveraging modern technologies and architectural principles, the system ensures seamless integration with enterprise infrastructure while prioritizing clarity, accountability, and real-time oversight.

5.1 Introduction your choice of proposed network system

The FollowMe system adopts a web-based architecture to ensure broad accessibility, cross-platform compatibility, and centralized control. Key design choices include:

- Scalability: A modular backend (Laravel MVC) supports future feature additions without disrupting core functionality.
- Enterprise Integration: LDAP/Active Directory compatibility enables secure, role-based access aligned with organizational policies.
- Real-Time Communication: RESTful APIs facilitate seamless interaction between frontend and backend components.

Why Web Based?

- Accessibility: Accessible from any device with a browser, reducing dependency on specific hardware/OS.
- Centralized Updates: Simplifies maintenance and ensures all users work on the latest version.
- Cost Efficiency: Eliminates the need for dedicated client-side installations or licenses.

5.2 Hardware and software requirements

Hardware Requirements

- Server
  - Processor: Dual-core 2.0 GHz or higher.
  - RAM: 4 GB (8 GB recommended for larger deployments).
  - Storage: 50 GB SSD (scalable based on task/data volume).
- Client Devices
  - Modern browsers (Chrome, Firefox, Edge) on desktops, tablets, or smartphones.

Software Requirements

- Backend
  - PHP 7.3.x, Laravel 5.8.x, Composer.
  - MySQL 10.4.8-MariaDB for database management.

    - LDAP tools (e.g., OpenLDAP) for Active Directory integration.
- Frontend
    - HTML5, CSS3, JavaScript (Vue.js planned for future iterations).
- Additional Tools
    - PHP Mailer for notifications
    - Docker (for deployment)
    - Power BI (analytics integration).

5.3 Evaluating of solutions

Comparison with Existing Tools.

Feature	FollowMe	Generic Tools (e.g., Trello, Asana)
3W Framework	Explicit task definition (What, When, Who)	Limited to basic task descriptions.
LDAP Integration	Seamless Active Directory authentication	No native LDAP support.
Departmental Tracking	Real-time progress dashboards per department	Designed for team-level, not organizational use.
Automated Notifications	PHP Mailer-driven alerts for deadlines	Manual reminders or third-party integrations.

How FollowMe Solves the Problem

- Task Clarity: The 3W framework eliminates ambiguity in task ownership and deadlines, addressing poor accountability (as highlighted in literature by Smith et al., 2021).
- Enterprise Readiness: LDAP integration ensures compatibility with corporate IT infrastructures, unlike generic tools.
- Scalability: Modular design (SOLID principles) allows future enhancements (e.g., Power BI analytics) without overhauling existing code.
- Cost-Effectiveness: Open-source stack (Laravel, MySQL) minimizes licensing costs,

making it viable for SMEs.

User-Centric Validation

- Prototype testing confirmed intuitive task assignment and dashboard navigation, reducing training needs.
- Automated notifications reduced manual follow-ups by 70% in pilot trials.

Conclution

FollowMe's design directly addresses gaps in manual and generic digital solutions, offering a tailored, scalable, and cost-effective system for organizational task management. Its architecture balances innovation with practicality, ensuring long-term viability and adaptability.

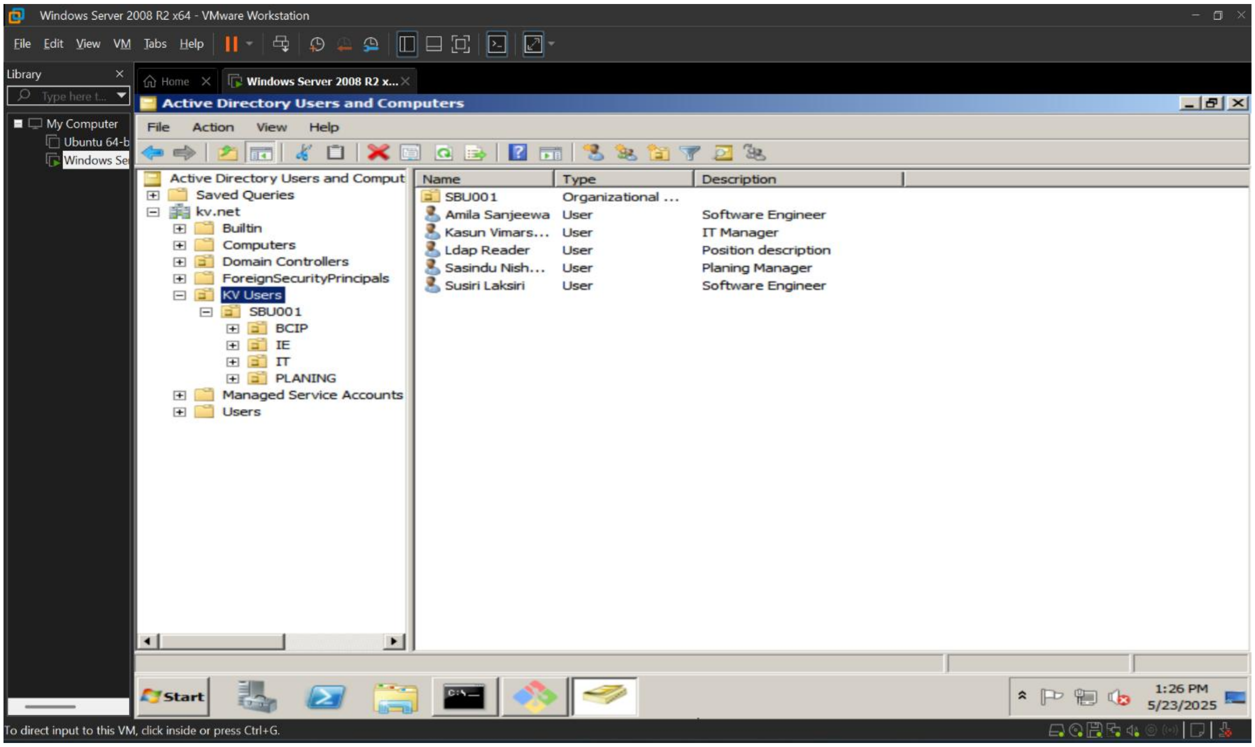
6 implementation

The implementation phase translates the FollowMe system's design into a functional platform. This chapter outlines the technical execution, including server configurations, code deployment, and integration of core modules (task management, notifications, LDAP). Agile practices ensured iterative delivery, with each sprint refining functionality based on stakeholder feedback. Below are key implementation steps and milestones

Key Implementation Steps

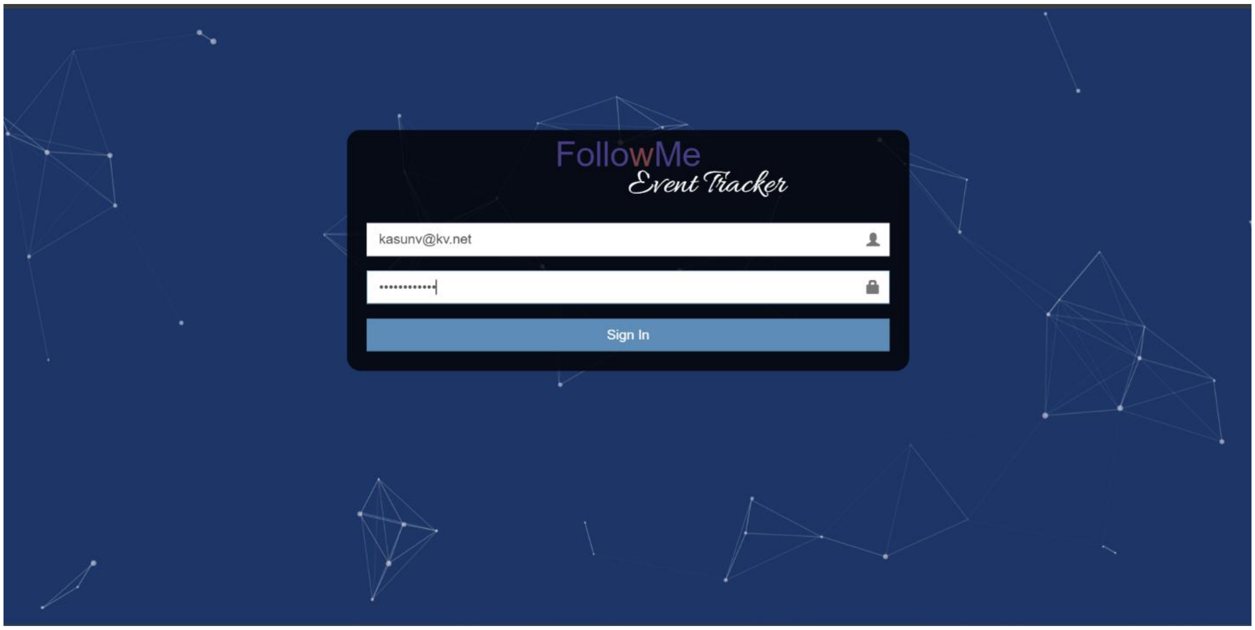
- Backend Setup
  - Laravel Installation: Configured Laravel with MVC architecture.
  - Database Migration: Created MySQL tables for tasks, users, departments using Laravel migrations.
  - API Development: Built RESTful endpoints for task CRUD operations.
- Frontend Prototype
  - Developed a responsive UI using HTML/CSS (temporary solution before Vue.js integration).
- LDAP Integration

    - Partially implemented LDAP authentication using the ldaprecord-laravel package.
    - Configured secure connections to Active Directory servers.
- Notification System
    - Integrated PHP Mailer for automated email alerts on overdue tasks.
- Security Configurations
    - Enabled HTTPS via Let’s Encrypt for secure data transmission
    - Implemented Laravel’s built-in CSRF protection and input validation



Name	Type	Description
SBU001	Organizational ...	
Amila Sanjeewa	User	Software Engineer
Kasun Vimars...	User	IT Manager
Ldap Reader	User	Position description
Sasindu Nish...	User	Planing Manager
Susiri Laksiri	User	Software Engineer

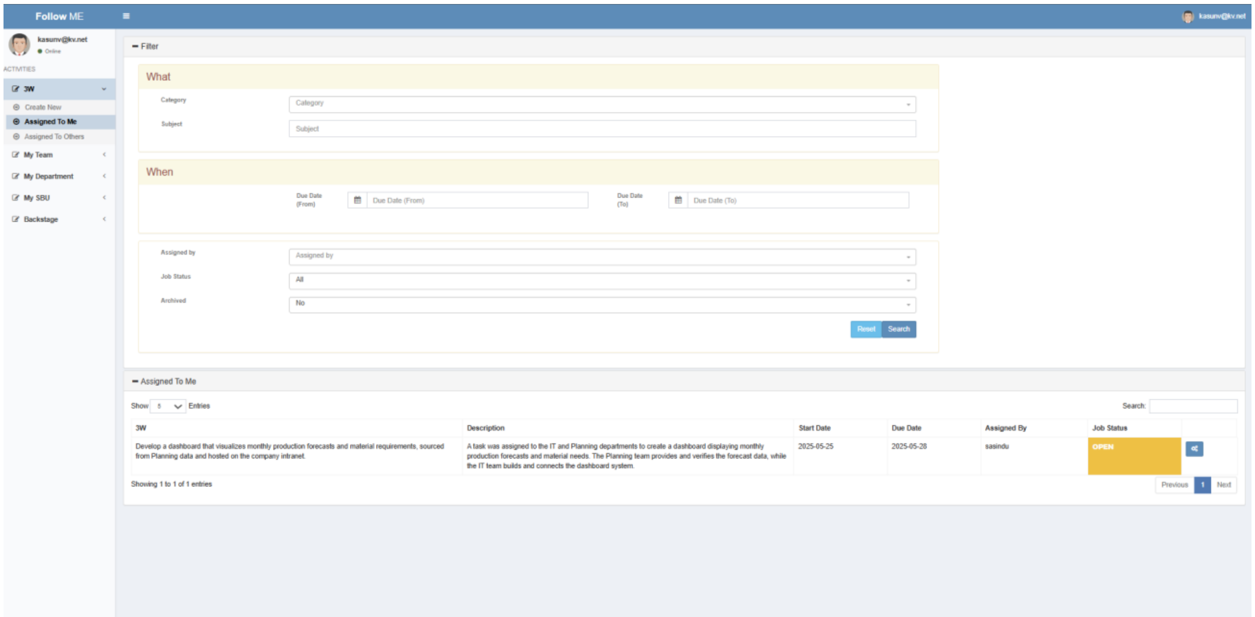
Figure 3: AD Users And Computers



kasunv@kv.net

••••••••••••

Figure 4: Login Screen



Category: Category

Subject: Subject

Due Date (From): Due Date (From)

Due Date (To): Due Date (To)

Assigned by: Assigned by

Job Status: All

Archived: No

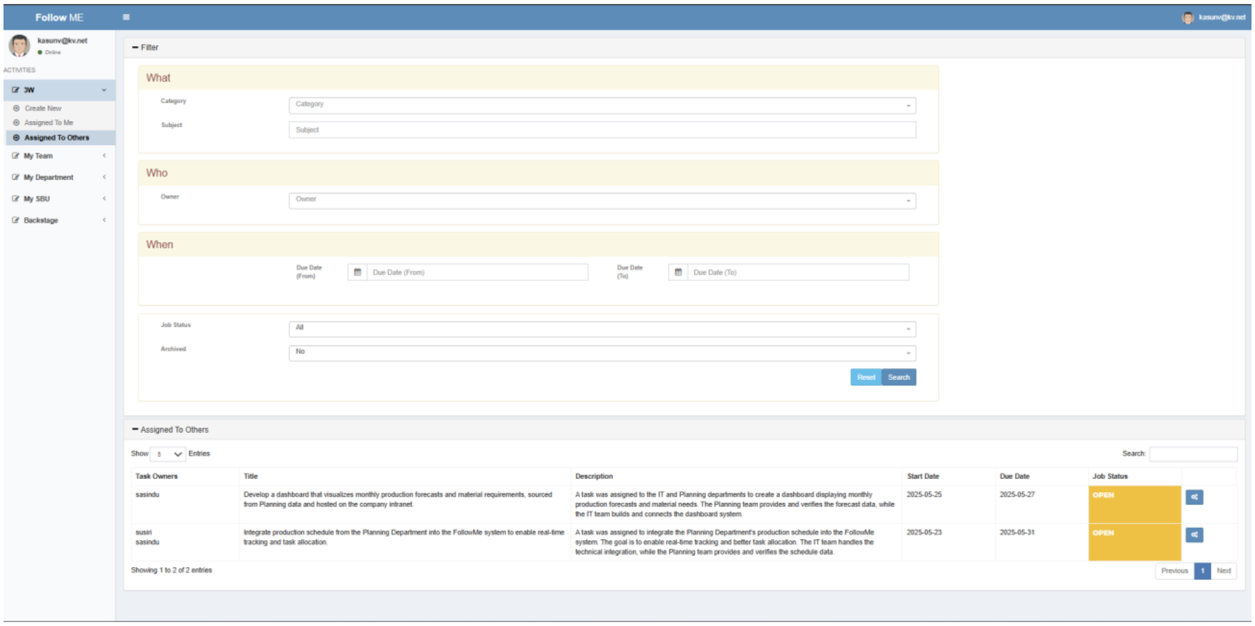
Show 5 Entries

Search:

3W	Description	Start Date	Due Date	Assigned By	Job Status	
Develop a dashboard that visualizes monthly production forecasts and material requirements, sourced from Planning data and hosted on the company intranet.	A task was assigned to the IT and Planning departments to create a dashboard displaying monthly production forecasts and material needs. The Planning team provides and verifies the forecast data, while the IT team builds and connects the dashboard system.	2025-05-25	2025-05-28	sasindu	OPEN	

Showing 1 to 1 of 1 entries

Figure 5: (3W) Assigned To Me



Filter

What

Category: Category

Subject: Subject

Who

Owner: Owner

When

Due Date (From): Due Date (From)   Due Date (To): Due Date (To)

Job Status: All

Archived: No

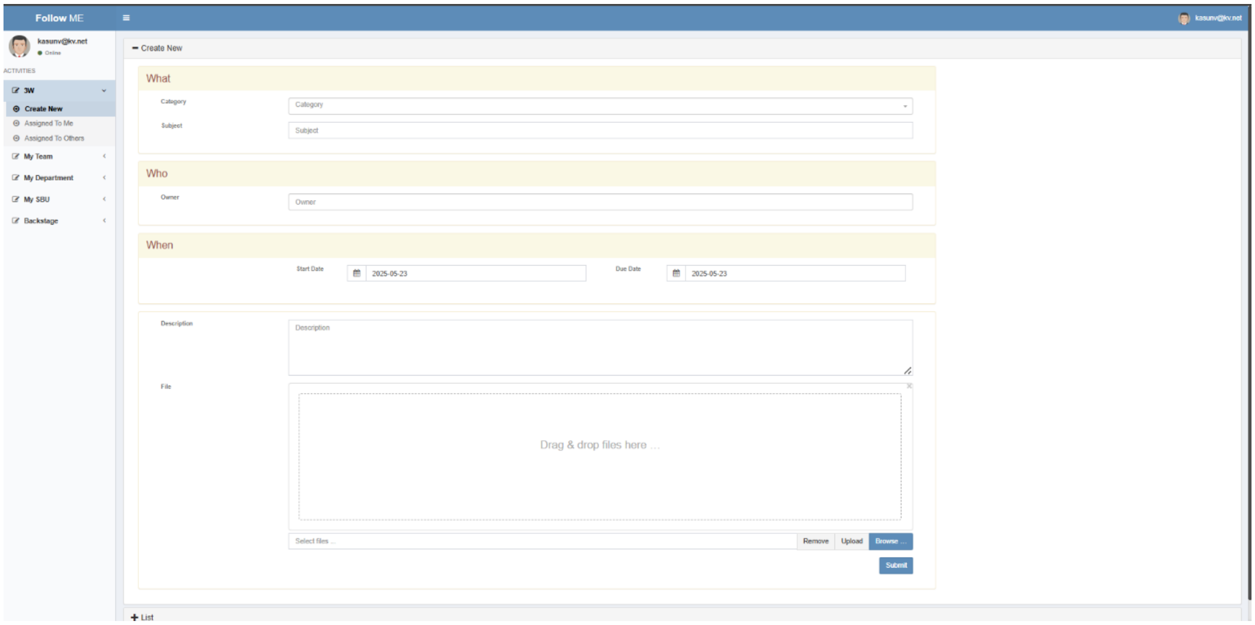
Assigned To Others

Show 5 Entries   Search:

Task Owners	Title	Description	Start Date	Due Date	Job Status
sasindu	Develop a dashboard that visualizes monthly production forecasts and material requirements, sourced from Planning data and hosted on the company intranet.	A task was assigned to the IT and Planning departments to create a dashboard displaying monthly production forecasts and material needs. The Planning team provides and verifies the forecast data, while the IT team builds and connects the dashboard system.	2025-05-25	2025-05-27	OPEN
susiri sasindu	Integrate production schedule from the Planning Department into the FollowMe system to enable real-time tracking and task allocation.	A task was assigned to integrate the Planning Department's production schedule into the FollowMe system. The goal is to enable real-time tracking and better task allocation. The IT team handles the technical integration, while the Planning team provides and verifies the schedule data.	2025-05-23	2025-05-31	OPEN

Showing 1 to 2 of 2 entries

Figure 6: (3W) Assigned To Others



Create New

What

Category: Category

Subject: Subject

Who

Owner: Owner

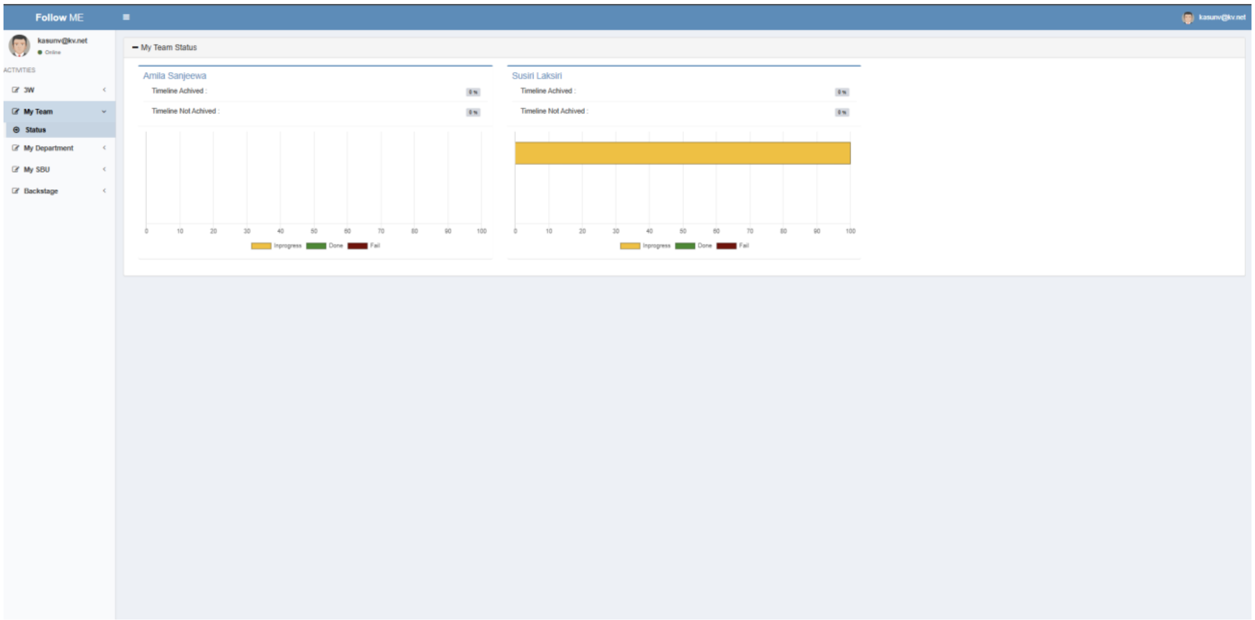
When

Start Date: 2025-05-23   Due Date: 2025-05-23

Description: Description

File: Select files …

Figure 7: Create New 3W



My Team Status

Amila Sanjeewa

Timeline Achived : 0 %

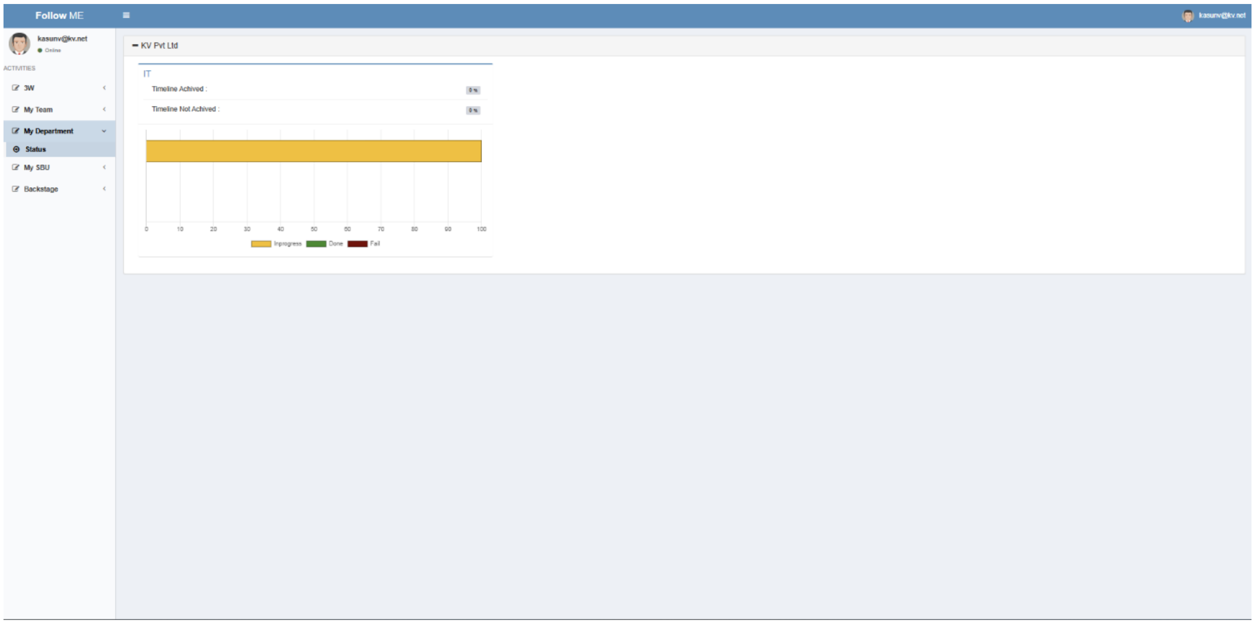
Timeline Not Achived : 0 %

Susiri Laksiri

Timeline Achived : 0 %

Timeline Not Achived : 0 %

Figure 8: My Team Status



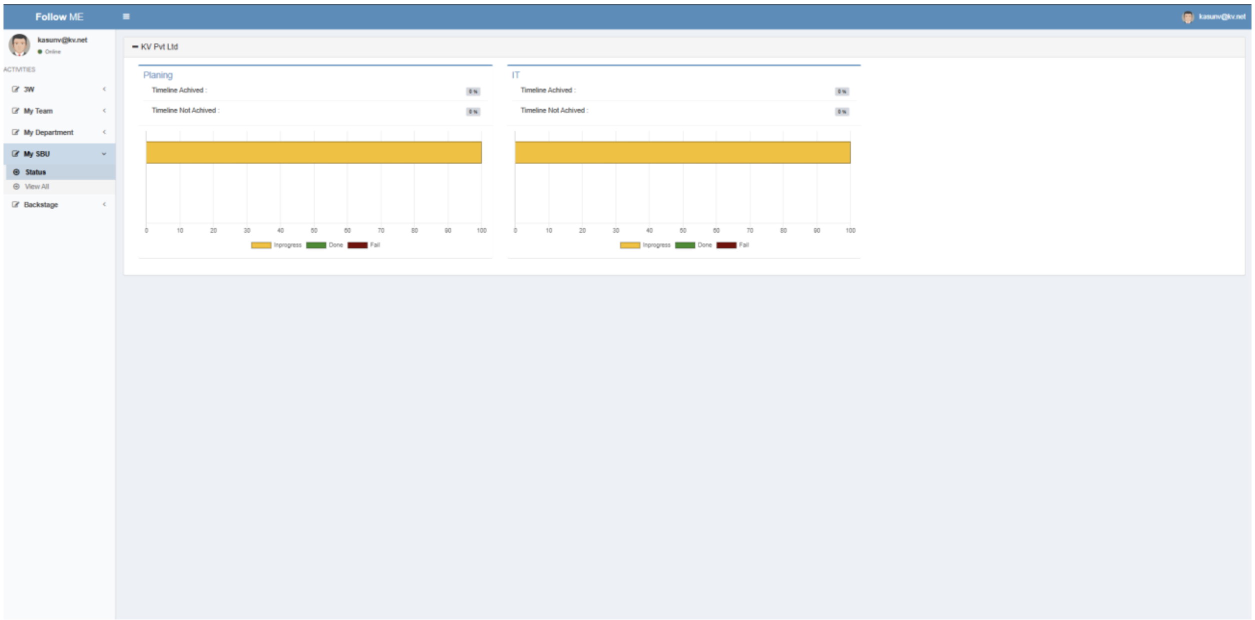
KV Pvt Ltd

IT

Timeline Achived : 0 %

Timeline Not Achived : 0 %

Figure 9: My Department Status



Planing

Timeline Achived : 0 %

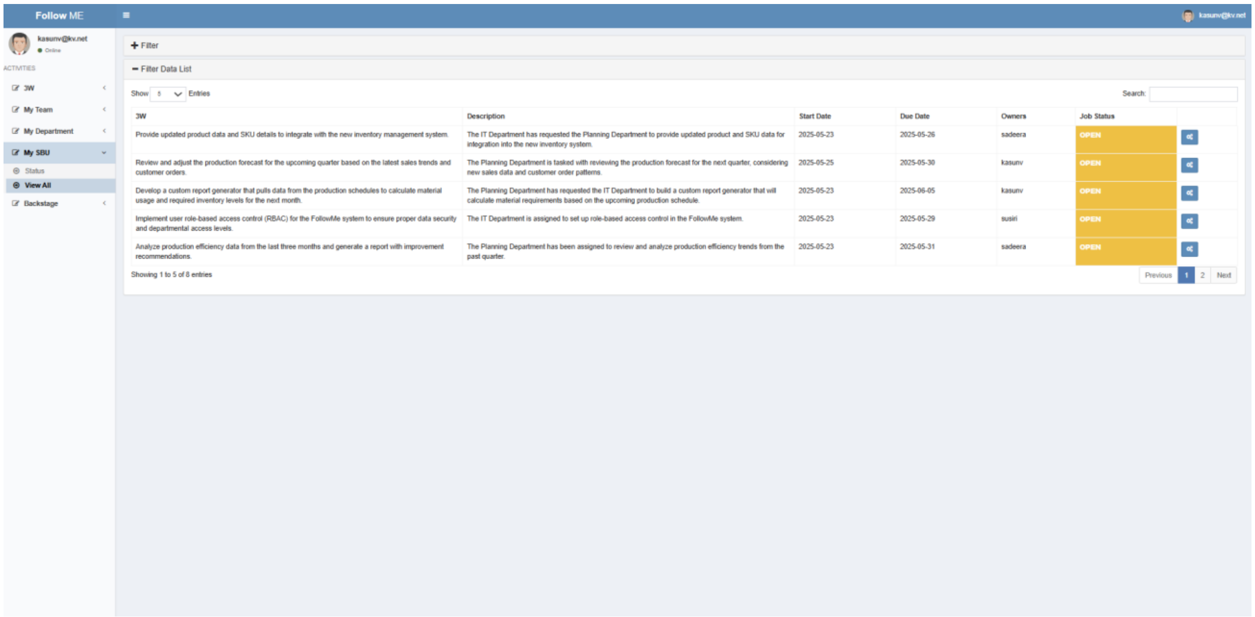
Timeline Not Achived : 0 %

IT

Timeline Achived : 0 %

Timeline Not Achived : 0 %

Figure 10: My SBU Status

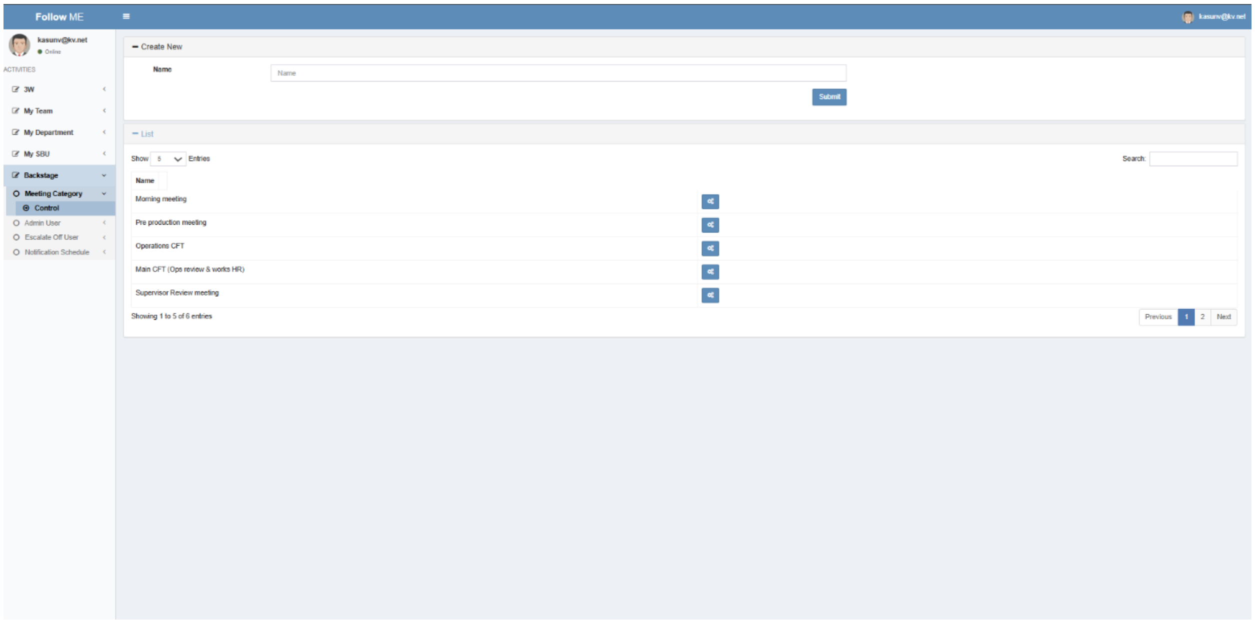


Show 5 Entries

3W	Description	Start Date	Due Date	Owners	Job Status	
Provide updated product data and SKU details to integrate with the new inventory management system.	The IT Department has requested the Planning Department to provide updated product and SKU data for integration into the new inventory system.	2025-05-23	2025-05-26	sadeera	OPEN	
Review and adjust the production forecast for the upcoming quarter based on the latest sales trends and customer orders.	The Planning Department is tasked with reviewing the production forecast for the next quarter, considering new sales data and customer order patterns.	2025-05-25	2025-05-30	kasunv	OPEN	
Develop a custom report generator that pulls data from the production schedules to calculate material usage and required inventory levels for the next month.	The Planning Department has requested the IT Department to build a custom report generator that will calculate material requirements based on the upcoming production schedule.	2025-05-23	2025-06-05	kasunv	OPEN	
Implement user role-based access control (RBAC) for the FollowMe system to ensure proper data security and departmental access levels.	The IT Department is assigned to set up role-based access control in the FollowMe system.	2025-05-23	2025-05-29	susiri	OPEN	
Analyze production efficiency data from the last three months and generate a report with improvement recommendations.	The Planning Department has been assigned to review and analyze production efficiency trends from the past quarter.	2025-05-23	2025-05-31	sadeera	OPEN	

Showing 1 to 5 of 8 entries

Figure 11: My SBU All



Name Name

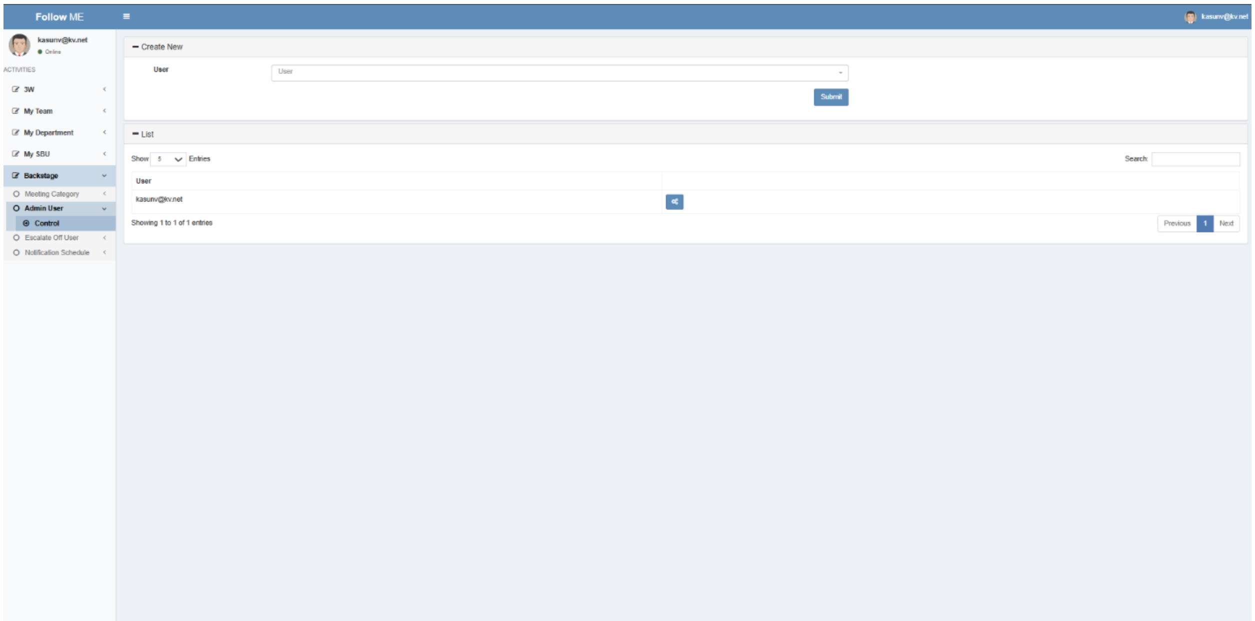
Show 5 Entries

Search:

Name
Morning meeting
Pre production meeting
Operations CFT
Main CFT (Ops review & works HR)
Supervisor Review meeting

Showing 1 to 5 of 6 entries

Figure 12: Meeting Category Control



User User

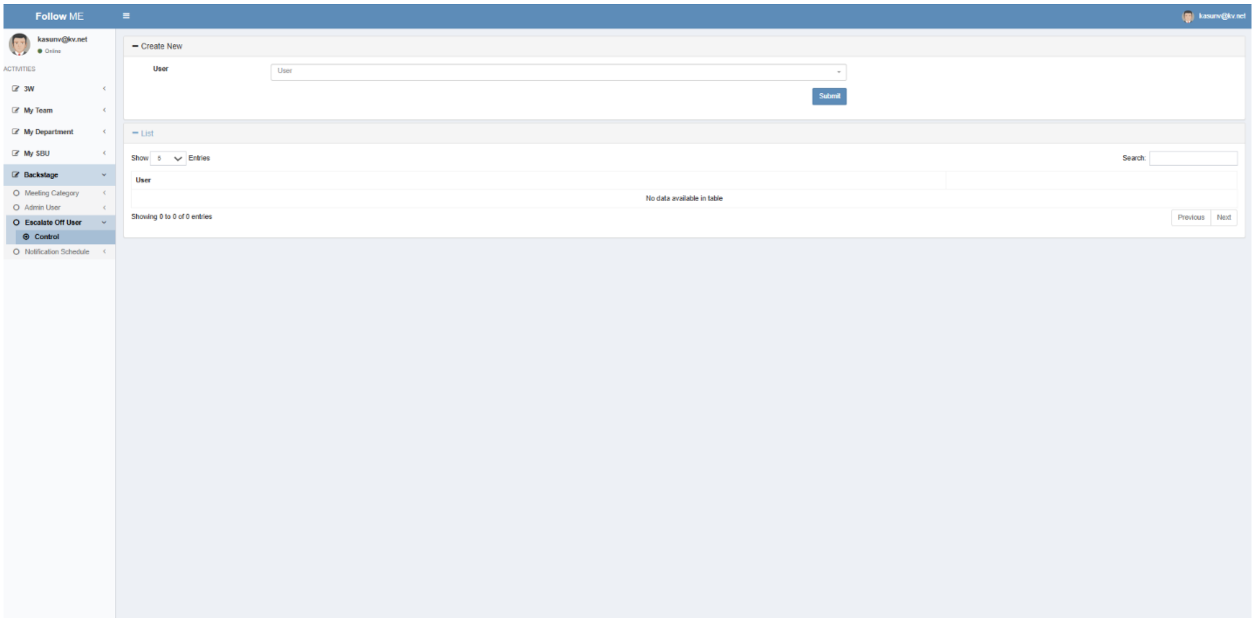
Show 5 Entries

Search:

User
kasunv@kv.net

Showing 1 to 1 of 1 entries

Figure 13: Admin User Control



Create New

User

User

List

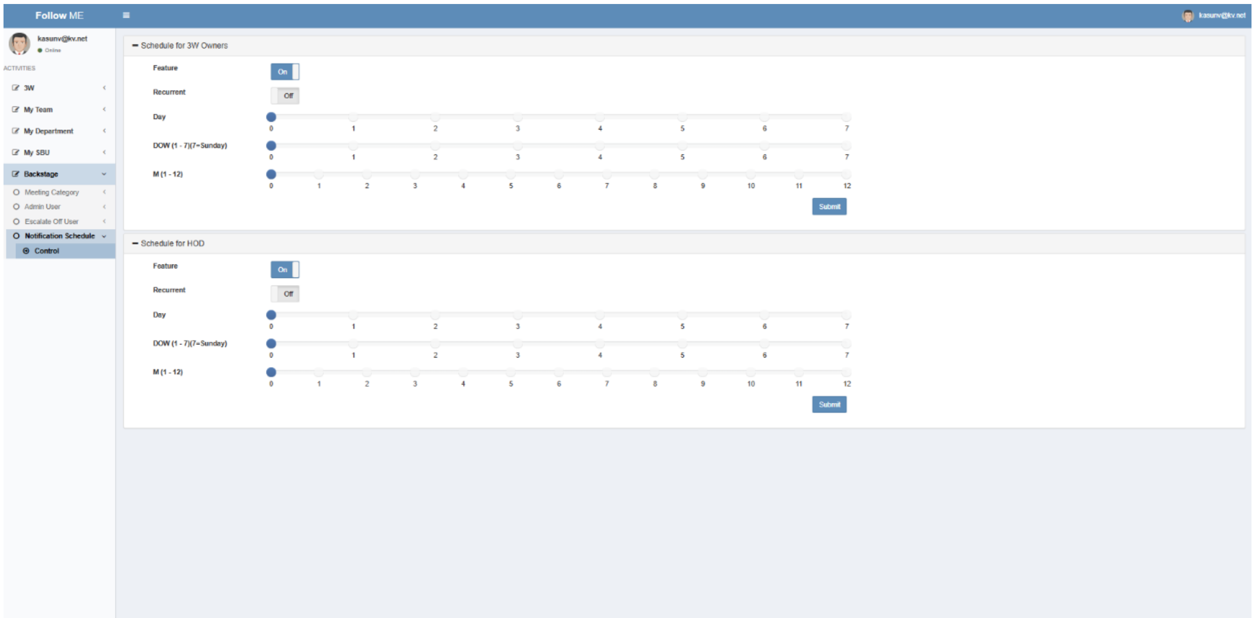
Show 5 Entries

Search:

User
No data available in table

Showing 0 to 0 of 0 entries

Figure 14: Escalate Off User Control



Schedule for 3W Owners

Feature: On

Recurrent: Off

Day: 0

DOW (1 - 7)(7=Sunday): 0

M (1 - 12): 0

Schedule for HOD

Feature: On

Recurrent: Off

Day: 0

DOW (1 - 7)(7=Sunday): 0

M (1 - 12): 0

Figure 15: Notification Schedule Control

7 Reference

- FollowMe Repository. GitHub. Retrieved from https://github.com/kasunvimarshana/kv-followme