



## **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 I worked 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 .....	6
1.1	Goals .....	7
1.2	Motivation .....	9
1.3	Method .....	10
2	Background and Problem Statement .....	12
2.1	Introduction .....	12
2.2	Literature Review .....	13
2.3	Problem Statement .....	14
3	Project management .....	14
3.1	Approach .....	14
3.2	Initial Project Plan .....	16
3.3	Problems and Changes to the Plan .....	17
3.4	Final Project Record .....	19
4	Feasibility study .....	20
4.1	Time feasibility .....	20
4.2	Cost feasibility. ....	21
4.3	Scope feasibility .....	21
4.4	Technical feasibility. ....	23
4.5	Economic feasibility .....	23
5	Design .....	24
5.1	Introduction your choice of proposed network system .....	24
5.2	Hardware and software requirements .....	24
5.3	Evaluating of solutions .....	25
6	implementation .....	26
7	Reference .....	36

## 1 Introduction

In today's business world, managing tasks well is very important. This is especially true in industries like garment factories, where many departments must work together. To do this properly, tasks need to be clear, on time, and with the right person.

Even though there are many digital tools today, most companies still use old methods. These include Excel sheets, paper notes, emails, or even just talking. These ways may feel easy, but they have many problems. Tasks are not clear, people forget deadlines, no one knows who is in charge, and it's hard to check the work.

The FollowMe 3W Management System was made to fix these problems. It uses a simple idea called the 3W model – What, When, and Who. This means each task clearly says what needs to be done, when it should be done, and who is doing it. This helps everyone understand their job. It also makes people responsible and builds trust and discipline in the workplace.

FollowMe is a web system. It can grow with your business. It is built using modern software tools. The backend is made with Laravel, a popular PHP framework. Laravel is clean, safe, and well-organized. MySQL is used to store task data. It is fast and easy to search.

FollowMe also connects with Active Directory (LDAP). This allows companies to use their current user login system. It makes it easy to control who can see or do what.

A special part of FollowMe is the auto reminder system. It sends emails before deadlines or when something is late. This is done using PHP Mailer. So, no need for people to keep reminding others. It saves time and reduces mistakes.

We also plan to link FollowMe with Power BI in the future. This will let managers see dashboards and reports easily.

We are building FollowMe using good software practices. It will be reliable, easy to maintain, and ready for future updates.

- 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.

Right now, about 60% of the planned work for the FollowMe system is finished. This includes the

backend services, database setup, main features, and early design of the user screens.

The remaining work includes:

- Finishing the frontend development
- Adding analytics and reporting
- Improving the notification system
- Doing full testing and fixing bugs
- Setting up Docker containerization
- Getting the system ready to go live

In short, FollowMe aims to help organizations by giving them a single platform where tasks are clearly created, tracked, and completed. It helps connect high-level planning with actual day-to-day work.

This report explains the design decisions, progress made, challenges faced, and the next steps to complete and launch the FollowMe system.

## **1.1 Goals**

The main purpose of the FollowMe – 3W Management System is to build a smart and easy-to-use task management platform. It is made for large organizations to help them assign, track, and complete tasks in a better way.

FollowMe uses the 3W method – What, When, and Who – to make sure every task has a clear goal, a deadline, and a responsible person. This helps to work in more organized way, specially in companies who contains many departments.

The system is designed to be:

- Easy to use in real work situations
- Built in parts (modular), so it's easier to manage and update
- Ready to connect with other company systems
- Based on modern software rules like SOLID, DRY, and KISS, so it's clean, simple, and



ready for future changes.

The The FollowMe – 3W Management System has both main goals and specific goals. These goals guide how the system is planned, built, and used.

- Primary Objective

To build a web-based system that helps companies create, assign, track, and report tasks clearly using the 3W method –

What (task), When (deadline), and Who (responsible person).

- Functional Goals

1. Task Creation and Assignment

Users can make tasks and say clearly what to do, when to finish, and who will do it.

Tasks can be grouped by department and priority.

2. Task Tracking and Monitoring

Users can see task progress in real-time.

Dashboards will show how many tasks are completed, pending, or late.

3. Automated Email Notifications

The system sends emails when a task is due, late, or finished.

This helps people not miss deadlines.

4. Secure Login and Role-Based Access

The system connects with Active Directory (LDAP) for safe login.

People will only see and do what their job role allows.

5. File Upload and Document Handling

Users can upload files or documents related to tasks.

Files are kept safe and can only be seen by allowed users.

6. Reports and Analytics

Users can create and download reports (like task completion rates).

In the future, it will work with Power BI for powerful charts and dashboards.

- Non-Functional Goals

### 1. Scalability

The system should work well even if more users or data are added.

### 2. Easy to Maintain

The code will be clean and well-organized, so future changes are easy.

New features can be added without breaking the system.

### 3. Easy to Use

The system will look nice and work well on all devices.

It will be easy to learn with little or no training.

### 4. Security

The system will check all inputs and use safe login methods.

Sensitive data will be protected with access control and session security.

### 5. Portability and Deployment

The system will use Docker to make it easy to install anywhere (office or cloud).

Settings will be managed using “.env files” for flexibility.

- Learning and Professional Goals

Use what was learned in software engineering: system planning, database design, APIs, and project steps.

Get real experience with popular tools like Laravel, HTML, CSS, MySQL, Docker, and LDAP.

Show the ability to build a full system from start to finish, alone.

## 1.2 Motivation

In today’s busy workplaces, especially in big industries like garment factories, logistics, and service companies, managing tasks properly is very important. But still, many companies in Sri Lanka and other developing countries use old methods like Excel sheets, printed checklists, or just telling things verbally to manage work.

These methods often cause problems like:

- No clear responsibility
- Missed deadlines
- Confused communication
- Hard to handle when company grows

In industries where work is fast and there are many tasks and people involved (like supervisors, managers, and office staff), these problems become worse. If there is no system to track tasks, check who is responsible, or see progress, people often repeat work, face delays, and waste time. Even top managers don't have a clear picture of what's going on, which makes decision-making harder.

The FollowMe – 3W Management System was created to solve these real problems. It helps companies move from paper or half-digital work to a modern, easy-to-track system. The 3W method (What, When, Who) is used to clearly show what the task is, when it must be done, and who will do it. This helps with better planning and smoother teamwork.

Also, many modern companies now use systems like Active Directory (AD) for logging in safely and giving the right access to staff. But most task tools don't support this. FollowMe fills this gap by connecting smoothly with AD, so companies can manage who sees what based on job roles.

Other reasons that motivated the project are:

- Fewer human errors – because of automatic reminders and real-time dashboards
- More transparency – everyone can see what's going on, so less confusion and blame
- Smarter decisions – Power BI support will help managers understand data and fix delays

In short, this project was started to fix a common and serious problem in many companies. By automating the full task process – from start to finish – FollowMe helps organizations work faster, clearer, and together, giving them a strong advantage in their work.

### **1.3 Method**

The FollowMe system was built step by step using an incremental method, following Agile principles. This helped to develop one feature at a time, get early feedback, and make improvements along the way.

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.

### 1.3 Overview

FollowMe is an online tool that helps to manage companies with their tasks using the simple idea of What to do, When to finish, and Who will do it. It makes it easy for different teams. Ant it gives clear tasks, set deadlines, and know who is responsible. This helps everyone work better together and keeps things clear.

The system is made with Laravel (a programming tool) for the back part that works behind the scenes. For the part users see and click, Vue.js will be used soon. All task data is saved in a MySQL database. For security, it connects to the company's login system , which is called Active Directory. It also sends automatic email reminders using PHP Mailer to keep people updated.

The system is designed to grow easily and can run in special software boxes called Docker (planned). Soon, it will also have tools to show easy reports and graphs with Power BI.

In short, FollowMe makes managing work simple, helps avoid mistakes, and gives managers clear information to make good decisions.

## **2 Background and Problem Statement**

Many companies still use old ways to manage tasks. They use spreadsheets, emails, or talk to each other. These ways cause many problems. People don't know who is responsible. Deadlines get missed. Communication is weak. Teams in different departments find it hard to work together.

Popular project management tools like Trello or Asana don't work well for companies that need strong control, secure user access, and connection with company login systems like Active Directory. Because of this, these tools are not the best fit for big organizations.

Problem Statement:

There is no easy-to-use, web-based system that helps companies assign tasks clearly, track progress in real time, and let departments work together securely using Active Directory. Companies need a system that shows What needs to be done, When it should be finished, and Who is responsible, so tasks are clear and transparent.

FollowMe was made to fix this problem using the simple 3W idea and built with modern technology and good software practices.

### **2.1 Introduction**

In many organizations, especially those with many departments, managing tasks is still a big problem. People often use spreadsheets, emails, or just talk to share tasks. These ways are not very organized. They cause poor communication, no clear responsibility, and missed deadlines. This is a big issue in industries like garment manufacturing where teams must work closely on tight schedules.

The main problems are that manual task systems are slow and unclear. They do not show who is responsible or how tasks are going in real time. Also, big companies use systems like Active

Directory to manage users and control access. But many task tools do not work well with these systems. Without a good system, tasks may not be assigned properly or finished on time.

FollowMe solves these problems. It is a web-based system that brings all tasks into one place. Every task is clear about What to do, When to finish, and Who will do it. This stops confusion and makes tasks easy to track.

## **2.2 Literature Review**

### **1. Task Clarity and Accountability Models**

Good task management needs clear responsibility. Smith et al. (2021) found that systems which say exactly what to do, when to finish, and who will do it improve on-time completion by over 30%. The 3W model (What, When, Who) follows this idea and makes sure no task is unclear.

### **2. Limits of Common Tools**

Popular tools like Trello and Asana have flexible task boards. But they don't show tasks by department or support role-based access and strong security like LDAP. These tools work well for small teams but not for big companies with many departments. This causes problems and mixed use.

### **3. Enterprise Integration and Authentication**

Big companies need systems that fit their IT setup. Microsoft (2023) says LDAP authentication is important for security and control. Most popular tools do not connect directly with Active Directory, causing extra work and security risks.

### **4. Automation and Notifications**

Research (Lee & Kumar, 2020) shows automatic reminders help finish tasks on time. Alerts that warn users before deadlines reduce manual follow-ups and improve discipline. FollowMe uses this idea with scheduled emails via PHP Mailer.

### **5. Analytics and Decision Support**

Managers need clear reports to see how work is going. Tools like Power BI help show data with real-time charts and dashboards. These help find delays and improve work. FollowMe plans to support this feature too.

## 2.3 Problem Statement

Many companies with several departments experience hardships when manage tasks well. They often use tools like spreadsheets or basic project software. These tools don't clearly show who is responsible for what, or when things are due. They also don't track progress well or give proper access control.

Right now, there is no system that:

- Clearly says What the task is, When it should be done, and Who should do it.
- Works smoothly with **LDAP-based Active Directory** for secure, role-based login.
- Shows task progress in **real-time, sends reminders, and gives reports by department.**

The FollowMe system is being built to fix these problems. It is a web-based tool that uses the 3W model to bring better task clarity, stronger accountability, and smoother teamwork across departments.

## 3 Project management

Effective planning was very important to build the FollowMe system. We did the work in small parts, step by step. This helped us check each part and improve it before moving to the next. We used a simple method like Agile. It gave us the chance to change things when needed. We also got feedback often and made changes quickly. We made a basic plan at the start. We used a timeline (Gantt chart) to see how the work was going. We worked in short time periods called sprints. Sometimes we faced problems. But we fixed them by working together. This way, we finished the project on time and met our 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

While building the FollowMe system, we faced some problems. These problems were both technical and practical. Some parts took more time than we expected. Sometimes, the system didn't work the way we planned. We also had to change a few things after talking to users and getting their feedback. So, we changed the plan a bit. We fixed issues step by step and made sure the system stayed strong and easy to use.

These changes helped us create a better system that fits real company needs and works well in the real world.

- **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 Analytics
  - 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 Mitigation
  - 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.

## Conclusion

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.

## Conclusion

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

### 4.5 Economic feasibility

#### Cost Benefit Analysis

- 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.

## Conclusion

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.

### Conclusion

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

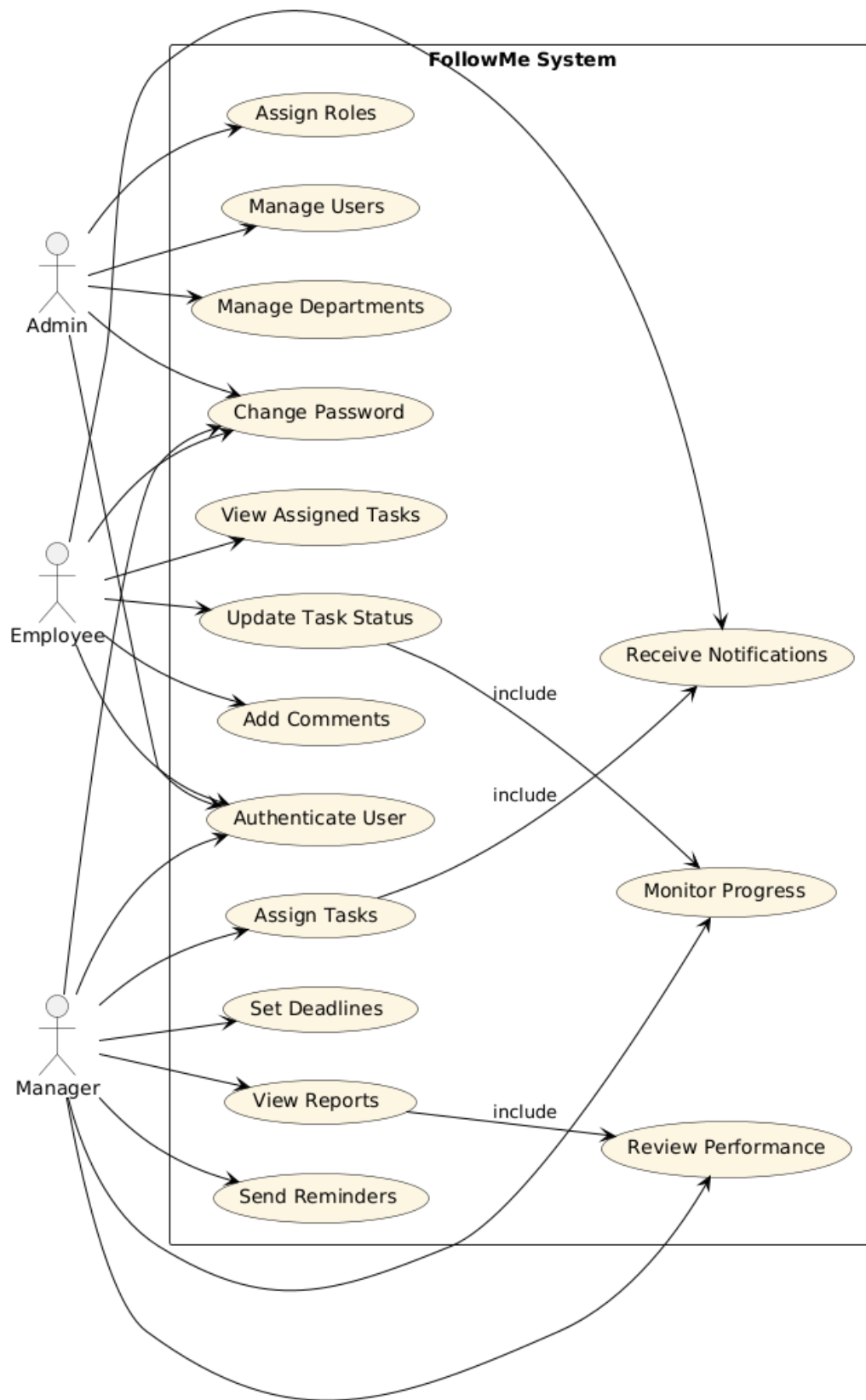


Figure 3: Use Case

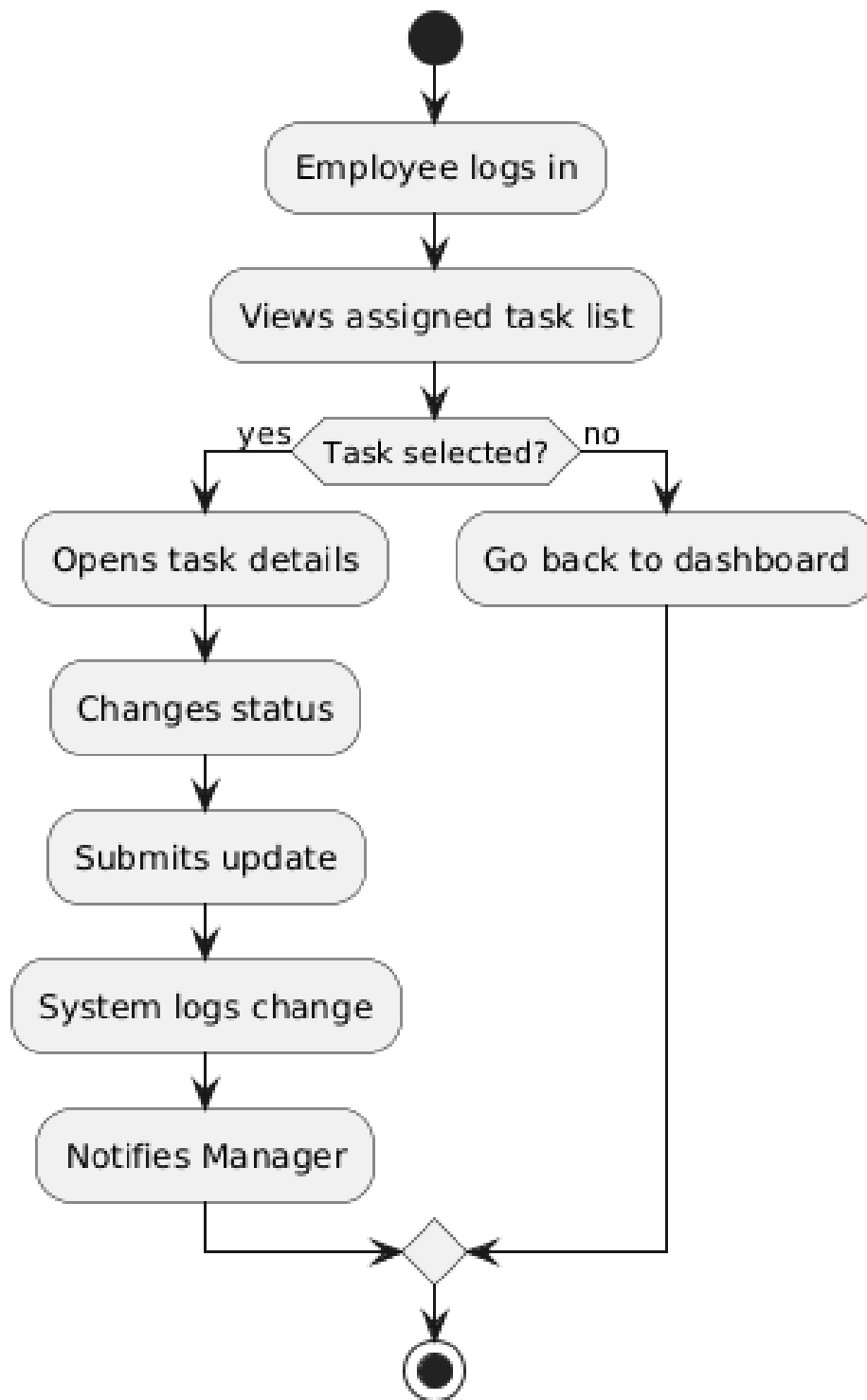


Figure 4: Activity Diagram – Update Task Status

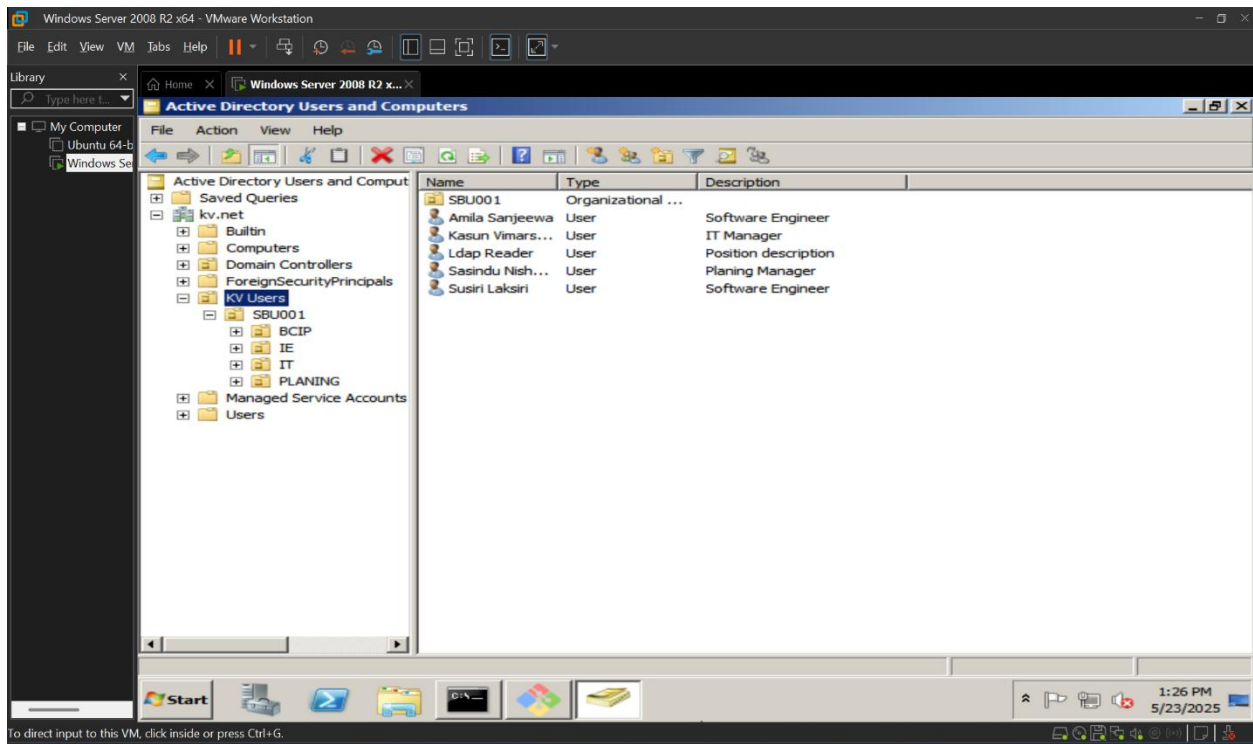


Figure 5: AD Users And Computers

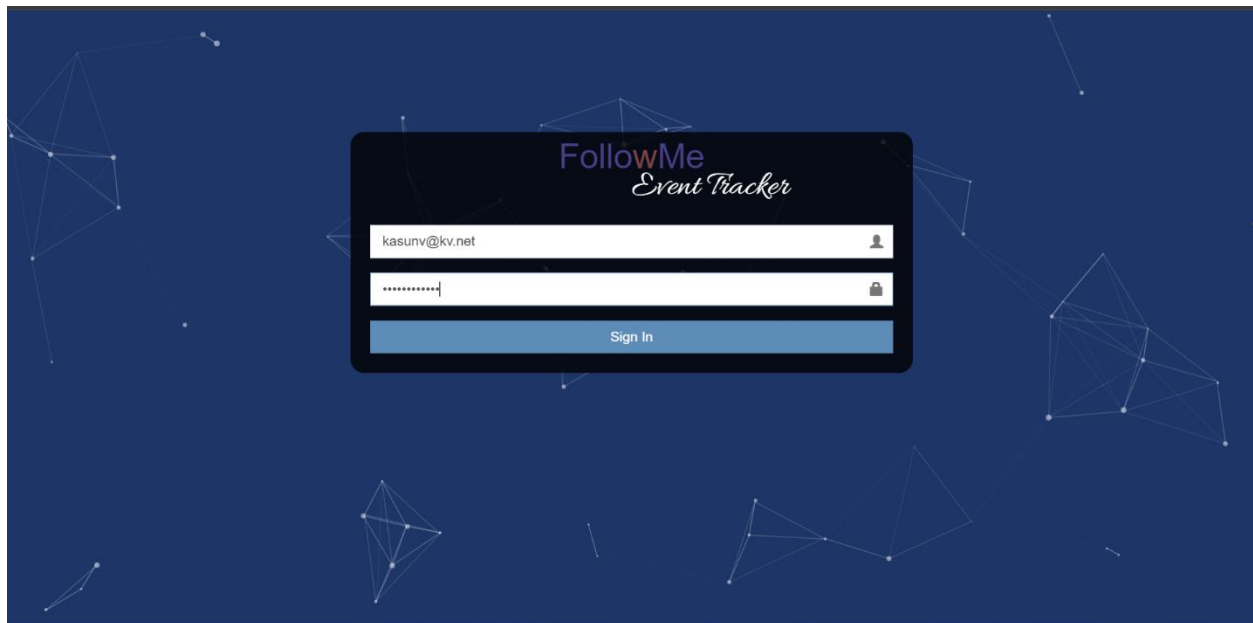


Figure 6: Login Screen





**Follow ME**

**ACTIVITIES**

- 3W
- Create New**
- Assigned To Me
- Assigned To Others
- My Team
- My Department
- My SBU
- Backstage

**Create New**

**What**

Category:

Subject:

**Who**

Owner:

**When**

Start Date:  Due Date:

Description:

File: 

Drag & drop files here ...

Select files ... Remove Upload Done Submit

[List](#)

Figure 9: Create New 3W

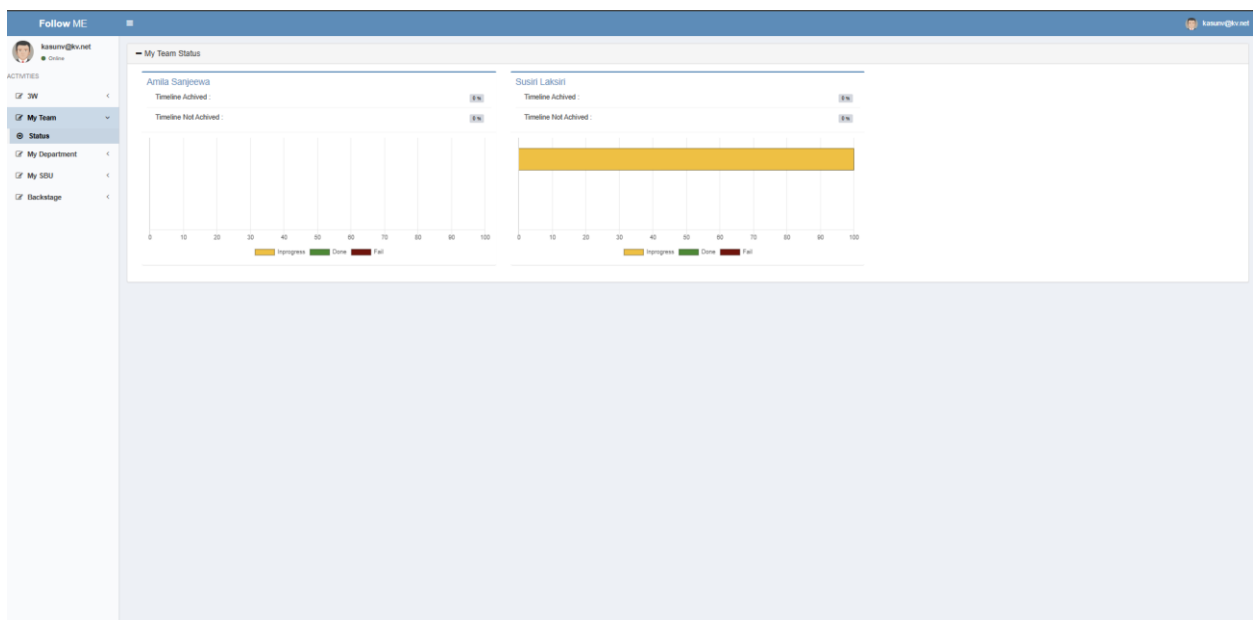


Figure 10: My Team Status

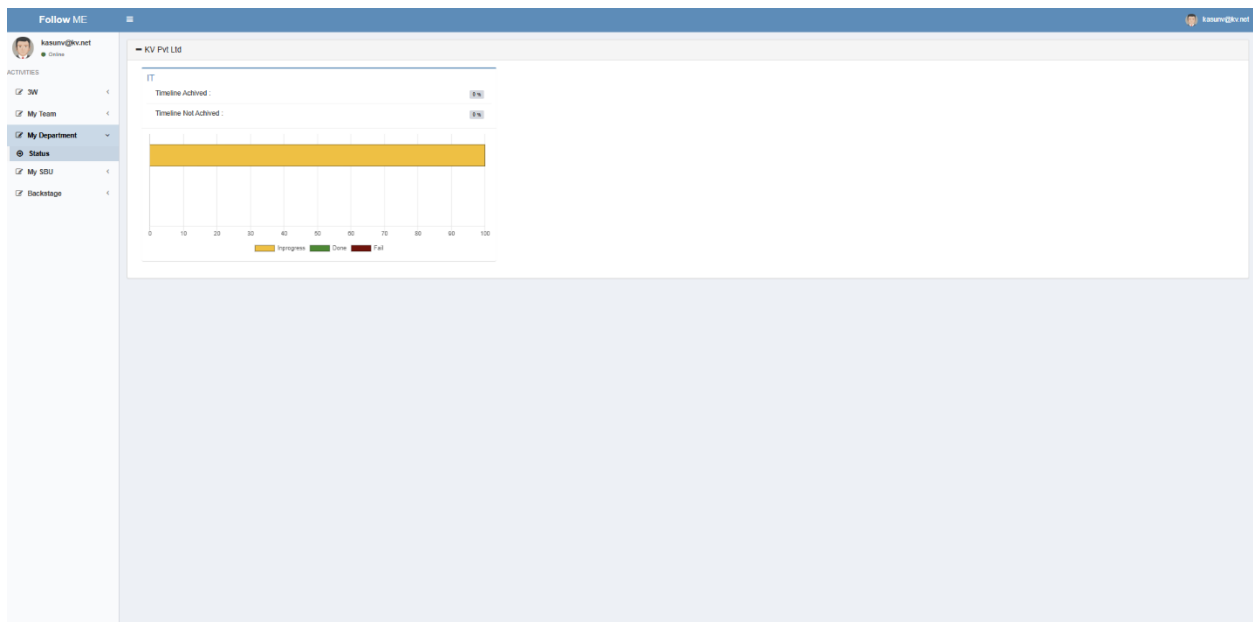


Figure 11: My Department Status

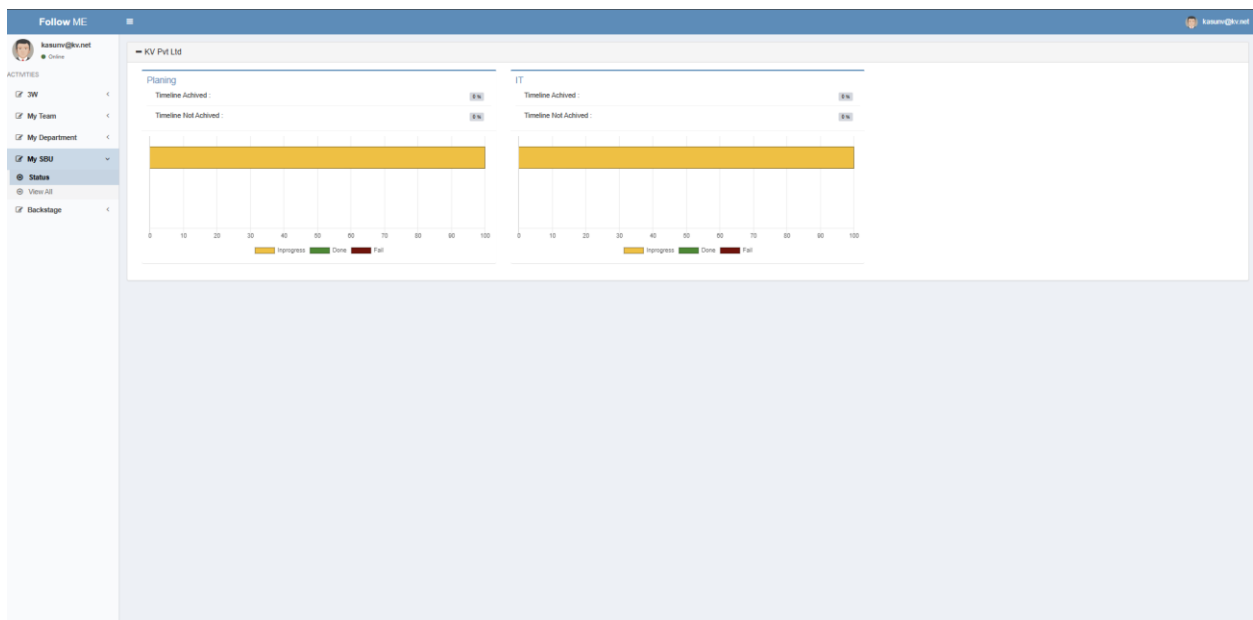


Figure 12: My SBU Status

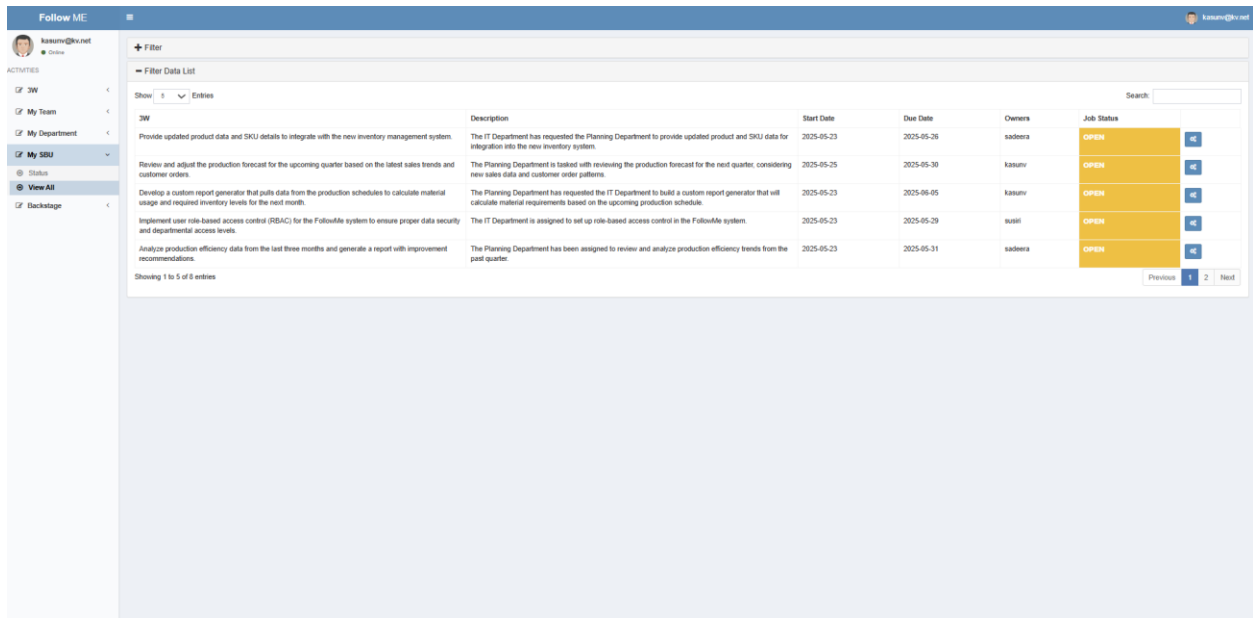


Figure 13: My SBU All

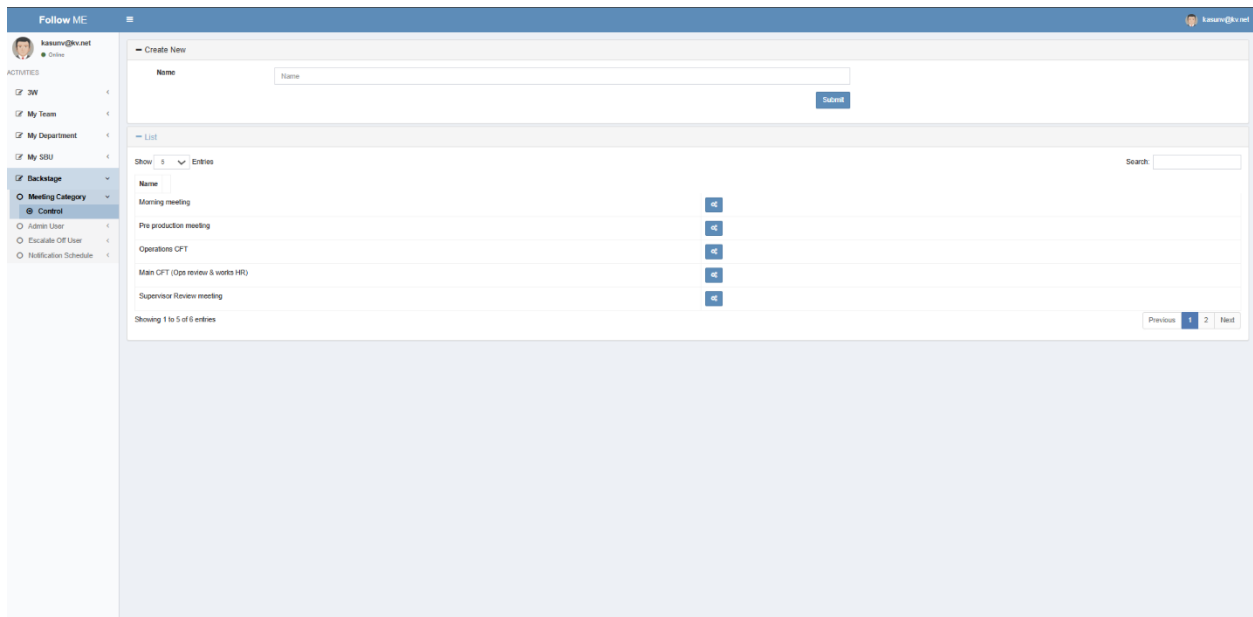


Figure 14: Meeting Category Control

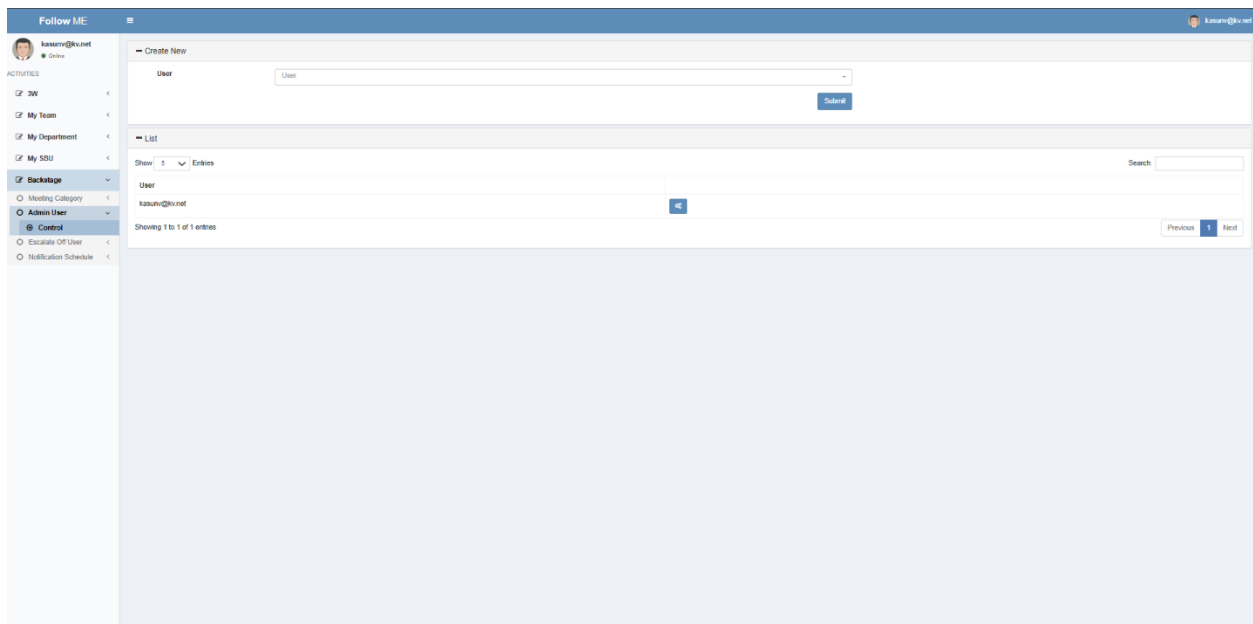


Figure 15: Admin User Control

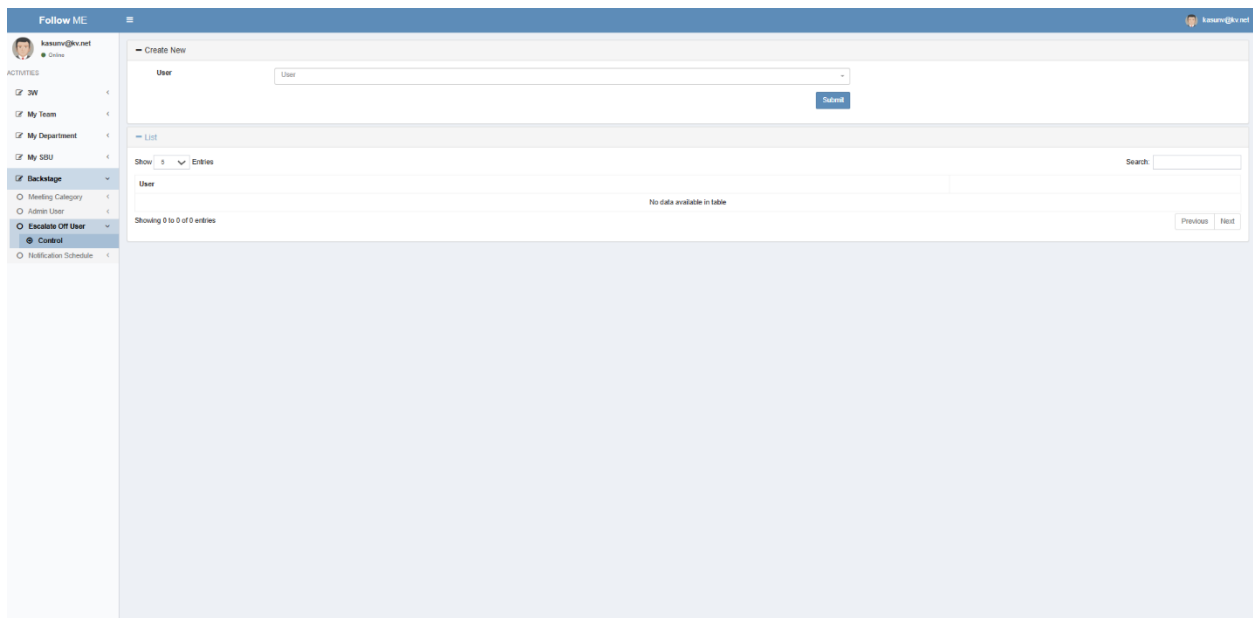


Figure 16: Escalate Off User Control

The screenshot displays the 'Follow ME' application interface. On the left is a sidebar with a user profile and a list of activities: 3W, My Team, My Department, My SBU, Backstage, Meeting Category, Admin User, Escalate Off User, Notification Schedule, and Control. The main content area is titled 'Follow ME' and contains two sections for scheduling notifications. The first section, 'Schedule for 3W Owners', and the second, 'Schedule for HOD', both feature a 'Feature' toggle (set to 'On'), a 'Recurrent' toggle (set to 'Off'), and three horizontal timelines for 'Day', 'DOW (1-7)(7-Sunday)', and 'M (1-12)'. Each timeline has a series of dots representing days, with numbers 0 through 12 below them. A 'Submit' button is positioned at the bottom right of each section.

Figure 17: Notification Schedule Control

## 7 Reference

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