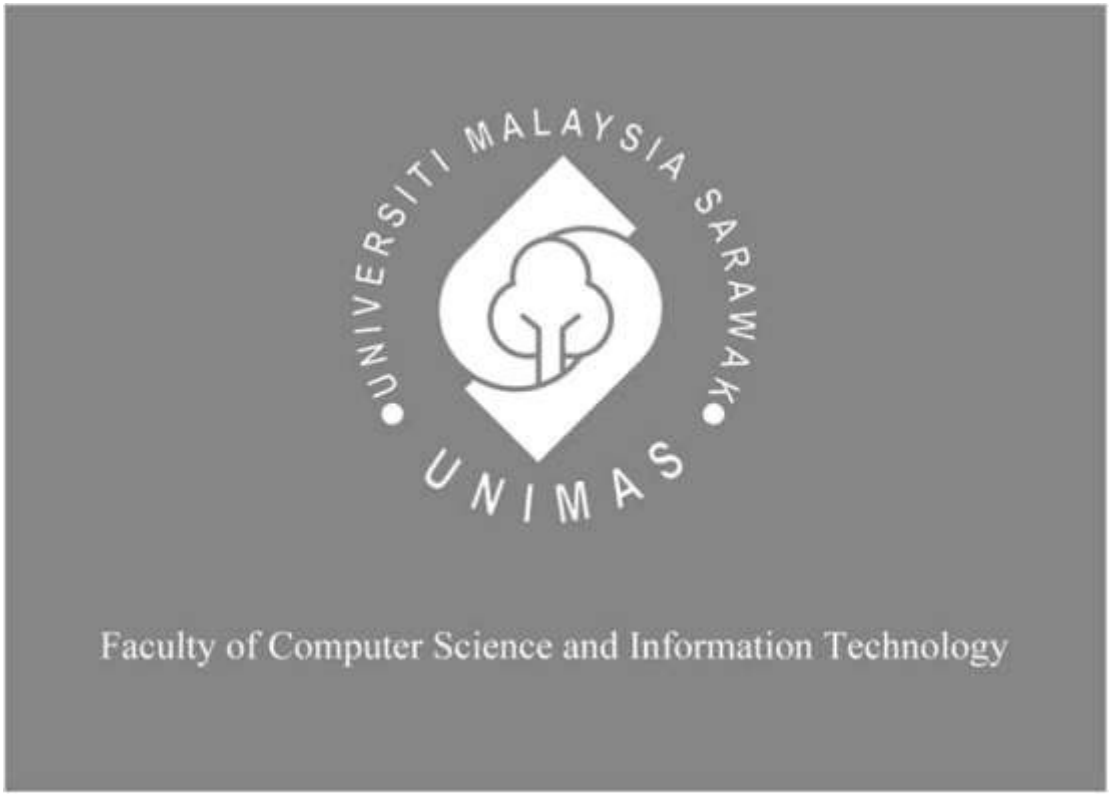


Smart Task Management System for UNIMAS Students

ELYCE PETAR

77954

BACHELOR OF COMPUTER SCIENCE WITH HONOURS

(INFORMATION SYSTEMS)

2025

SMART TASK MANAGEMENT SYSTEM FOR UNIMAS STUDENT

ELYCE PETAR

This project is submitted in partial fulfilment of

the requirements for the degree of Bachelor of Computer Science with Honours
(Information Systems)

Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2025

SISTEM PENGURUSAN TUGAS PINTAR UNTUK PELAJAR UNIMAS

ELYCE PETAR

Projek ini merupakan salah satu keperluan untuk
Ijazah Sarjana Muda Sains Komputer dengan Kepujian
(Sistem Maklumat)

Fakulti Sains Komputer dan Teknologi Maklumat
UNIVERSITI MALAYSIA SARAWAK
2020

DECLARATION

I declare that the work in this thesis was carried out in accordance with the regulations of Universiti Malaysia Sarawak. Except where due acknowledgements have been made, the work is that of the author alone. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

……

Signature

Name: Elyce Petar

Matric No.: 77954

Faculty of Computer Science and Information Technology

Universiti Malaysia Sarawak

Date: 18/06/2025

ACKNOWLEDGMENT

First and foremost, I would like to thank Almighty God for His endless blessings, guidance, and strength, which have made this project possible. Without His grace, this work would not have come to fruition. I would like to express my sincere gratitude to my final year project supervisor, Dr. Fatihah binti Ramli, for her invaluable guidance, support, and encouragement throughout the project. Her expertise and constructive feedback have been instrumental in shaping the direction and success of this work. I would also like to extend my appreciation to my FYP coordinator, Prof. Dr Wang Yin Chai, for his insights and guidance during this project. His expertise has provided me with the necessary foundation to complete the project successfully. My heartfelt thanks go to Professor Dr. Jane Labadin, my project examiner, for her valuable time, feedback, and professional evaluation of my work. Her insightful comments and suggestions have helped enhance the quality and depth of this project. A special thank you goes to my friend, Iqmah, for her continuous support, motivation, and collaboration throughout the journey of completing this project. Her willingness to share ideas and help, especially during challenging moments, has truly made a difference. I am also grateful to the participants who took the time to complete the surveys and provide valuable responses. Their feedback has provided crucial insights into the challenges faced by university students in managing tasks and deadlines, helping to refine the features and functionality of the system. Lastly, I would like to acknowledge the University Malaysia Sarawak (UNIMAS) for providing the opportunity to conduct this project, as well as for the academic environment that fosters innovation and research. This project would not have been possible without the support and resources provided by the university. Thank you to everyone who has supported me along the way and to all who have contributed to making the Smart Task Management System (STMS) a tool that aims to enhance productivity and collaboration for university students.

ABSTRACT

The Smart Task Management System (STMS) is a web-based platform designed to enhance productivity, collaboration, and task management among University Malaysia Sarawak (UNIMAS) students. Managing academic responsibilities, group projects, and extracurricular commitments can be challenging without an efficient system. STMS addresses these challenges by integrating key features such as task prioritization and team collaboration tools to help students efficiently manage their academic workload. This study presents a comprehensive analysis of the design, development, and evaluation of STMS, following the Agile methodology to ensure flexibility and continuous improvement based on user feedback. The system is developed with a user-centric approach, focusing on an intuitive interface and robust backend functionality to support scalability and security. The evaluation phase involved usability testing and student feedback, demonstrating the system's effectiveness in improving task organization, time management, and group collaboration. The findings indicate that STMS significantly enhances academic efficiency, with students reporting improved task tracking, deadline management, and teamwork. By providing a structured yet adaptable framework for academic task management, STMS serves as a valuable tool for students in higher education. Future enhancements will focus on expanded mobile accessibility and integration with university learning management systems to further improve user experience and system capabilities.

ABSTRAK

Sistem Pengurusan Tugas Pintar (STMS) ialah platform berasaskan web yang direka untuk meningkatkan produktiviti, kerjasama dan pengurusan tugas dalam kalangan pelajar Universiti Malaysia Sarawak (UNIMAS). Menguruskan tanggungjawab akademik, projek kumpulan, dan komitmen ekstrakurikuler boleh menjadi mencabar tanpa sistem yang cekap. STMS menangani cabaran ini dengan menyepadukan ciri utama seperti keutamaan tugas dan alatan kerjasama pasukan untuk membantu pelajar mengurus beban kerja akademik mereka dengan cekap. Kajian ini membentangkan analisis komprehensif reka bentuk, pembangunan dan penilaian STMS, mengikut metodologi Agile untuk memastikan fleksibiliti dan penambahbaikan berterusan berdasarkan maklum balas pengguna. Sistem ini dibangunkan dengan pendekatan berpusatkan pengguna, memfokuskan pada antara muka intuitif dan kefungsian bahagian belakang yang mantap untuk menyokong kebolehskalaan dan keselamatan. Fasa penilaian melibatkan ujian kebolehgunaan dan maklum balas pelajar, menunjukkan keberkesanan sistem dalam meningkatkan organisasi tugas, pengurusan masa dan kerjasama kumpulan. Penemuan menunjukkan bahawa STMS meningkatkan kecekapan akademik dengan ketara, dengan pelajar melaporkan penjejakan tugas yang lebih baik, pengurusan tarikh akhir dan kerja berpasukan. Dengan menyediakan rangka kerja yang tersusun tetapi boleh disesuaikan untuk pengurusan tugas akademik, STMS berfungsi sebagai alat yang berharga untuk pelajar di pendidikan tinggi. Penambahbaikan pada masa hadapan akan memberi tumpuan kepada kebolehcapaian mudah alih yang diperluaskan dan integrasi dengan sistem pengurusan pembelajaran universiti untuk meningkatkan lagi pengalaman pengguna dan keupayaan sistem.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations	Descriptions
CSS	Cascading Style Sheets

FYP	Final Year Project
HTML	Hypertext Markup Language
STMS	Smart Task Management System
TDD	Test-Driven Development
UAT	User Acceptance Testing
UI	User Interface
UI/UX	User Interface / User Experience
UNIMAS	University Malaysia Sarawak

Chapter 1: Introduction

1.1 Title

Smart Task Management System for UNIMAS Student

1.2 Introduction

In an increasingly fast-paced academic and professional environment, efficient task management has become a critical factor for productivity, accountability, and effective communication within teams. For university students, particularly those in project-driven fields like computer science and information technology, balancing various academic and extracurricular responsibilities can be challenging. Many institutions, including universities like University Malaysia Sarawak (UNIMAS), face issues in managing student tasks effectively, leading to missed deadlines, inefficiencies, and limited team collaboration. Current manual or semi-automated systems often fall short, lacking the flexibility and functionality required to optimize task delegation and tracking, particularly in dynamic, collaborative environments.

This final year project aims to address these challenges by developing a Smart Task Management System for UNIMAS students, incorporating essential task management features tailored to the needs of students and academic teams. This web-based system will support automate task prioritization, task creation, assignment, scheduling, and progress tracking, enabling users to manage their responsibilities more effectively. Designed with intuitive interfaces and real-time notifications, the system also offers performance analytics and reporting tools to support better planning and decision-making.

Built using robust backend technologies such as PHP and MySQL, the system will include core functionalities like task categorization, deadline setting, status updates, and user collaboration. It will also feature modules for role-based access control, ensuring that tasks and projects are managed in an organized and secure environment.

The project focuses on three core objectives: facilitating streamlined task management, improving time management and team communication, and enabling performance evaluation through insightful reports. The outcome will be a practical and organized platform that helps UNIMAS students manage and track their tasks more effectively, ultimately enhancing productivity and collaboration.

1.3 Problem Statement

In academic environments, especially among university students, managing multiple tasks and deadlines can be overwhelming, leading to common issues such as missed deadlines, inefficient time management, and fragmented team communication. Without an effective system for task management, students often rely on manual methods like spreadsheets or communication through email and messaging platforms. These approaches lack real-time updates, automated prioritization, and clear task visibility, which can result in miscommunication, task duplication, and ineffective resource allocation.

At University Malaysia Sarawak (UNIMAS), there is a need for a structured task management solution that can cater to the unique requirements of students working on individual and group projects. The absence of an organized and responsive task management system limits students' ability to effectively prioritize tasks, forecast deadlines, and track progress. Furthermore, without automated notifications and smart task tracking, students face challenges in identifying critical tasks and optimizing workflows, which impacts overall productivity and collaboration.

This project seeks to address these challenges by developing a smart task management system tailored for UNIMAS students. The system will aim to streamline task delegation, provide real-time updates, and enable structured task prioritization to enhance efficiency, accountability, and communication within academic project teams.

1.4 Scope

This project aims to develop a practical, task management system specifically for UNIMAS students, focusing on essential features to improve task organization and tracking. The system will allow users to create tasks, assign them to team members, and set deadlines within an easy-to-use interface. Basic tracking features, like status updates ("Not Started," "In Progress," "Completed"), will give users a clear view of progress, along with automated reminders for upcoming or overdue tasks to keep everyone on track.

A straightforward feature will help users focus on urgent tasks based on deadlines and dependencies, using a simple algorithm. Basic reporting tools will also be included to provide summaries of task completion and upcoming deadlines, offering an overview of progress without complex data analysis.

1.5 Aim of Project

The aim of the Smart Task Management System (STMS) is to develop an intelligent and user-friendly platform that enhances productivity, improves time management, and fosters collaboration among UNIMAS students. By integrating task prioritization, deadline tracking, and real-time notifications, the system seeks to streamline academic and group task management. The project aims to provide a structured, efficient, and accessible solution tailored to students' dynamic academic needs, ensuring better organization, accountability, and seamless communication within teams.

1.6 Objectives

1. To develop a user-friendly task management system that simplifies task assignment and tracking.
2. Integrating task prioritization, deadline management, real-time status updates, and notifications to enhance time management and team communication.
3. To implement reporting tools and task analysis and performance evaluation.
4. To evaluate the performance of proposed system

1.7 Brief Methodology

This project follows a clear, step-by-step approach to develop and evaluate a task management system for UNIMAS students. The system will be created in phases using the Agile methodology, allowing for continuous feedback and improvement. We'll start by defining essential features such as task creation, tracking, notifications, and reporting and then develop a user-friendly interface using PHP, MySQL, HTML, CSS, and JavaScript. After coding the core features, we'll deploy a prototype for UNIMAS students to test, which will help us identify improvements based on real user feedback.

To measure the system's success, we'll track several key outputs. For example, we'll monitor the number of tasks created and assigned, aiming for at least 100 tasks during testing. We'll also assess the accuracy of task prioritization (aiming for 85% alignment with user expectations) and measure user engagement by tracking how often students use the system, with a goal of 50% weekly active users. These metrics help ensure that the system is user-friendly, efficient, and relevant to student needs.

Data will be collected from two main sources: system logs (to track user activity) and user surveys (to gather feedback on features like prioritization and ease of use). Data collection will

be done weekly for system logs and every two weeks for user surveys. This data will be analysed through simple statistical methods to summarize usage and test the system's effectiveness. We'll then create a report detailing user satisfaction and system performance, highlighting any needed improvements.

Agile Approach in the Development of the Smart Task Management System for UNIMAS student:

The Agile methodology, known for its flexibility and iterative process, is well-suited for developing the Smart Task Management System (STMS) for UNIMAS students. This approach emphasizes collaboration, continuous feedback, and incremental progress to ensure the system meets both user needs and project goals. The development will be divided into multiple sprints, each lasting 2–3 weeks, with each sprint focusing on specific tasks such as designing the user interface or setting up notifications. This breakdown allows the team to make steady progress while quickly addressing any challenges.

Active collaboration with stakeholders, including supervisors and student testers, will be maintained throughout the project. Regular sprint meetings will be held to review progress, gather feedback, and refine requirements, ensuring the system remains aligned with expectations. Agile also allows for early delivery of working prototypes. For instance, a basic version of the system with task creation features may be released early for user testing, helping guide future improvements.

Testing is integrated into every sprint. Unit testing will ensure individual components function correctly, while integration and user acceptance testing (UAT) will check how the entire system performs and how users interact with it. Data will drive adjustments; metrics like task completion rates, user satisfaction scores, and task prioritization accuracy will help the team make informed improvements. Agile's adaptability also allows the development team to respond to changing needs, such as adding new features based on user feedback.

Each sprint will produce deliverables. Early sprints will cover foundational elements like database setup and front-end design. Mid-phase sprints will focus on implementing AI features and notifications, while the final stages will address optimization, bug fixes, and preparing user documentation. This Agile approach ensures the system is built progressively, with quality and user satisfaction at its core.

1.8 Significance of Project

This study holds significant potential for advancing both theoretical understanding and practical applications in educational task management. By developing a task management system specifically tailored for UNIMAS students, the study will contribute to the broader theoretical framework of educational technology and digital task organization. The insights gained from this project will enrich the literature on how digital tools, such as task prioritization features and automated notifications, can enhance time management, improve organization, and foster accountability among students in a higher education setting.

The findings may suggest areas for further research in educational tools, particularly regarding task prioritization models and how these can support diverse learning environments. Future studies could explore additional capabilities, such as user input interpretation or adaptive features that align with individual student behaviours and preferences. Such research could deepen understanding of how technology can support personalized learning and collaborative skills in educational contexts.

For educators, the results offer practical implications for how task management tools can be integrated into academic settings to support student productivity and engagement. The system's user-friendly design, combined with its notification and prioritization features, may serve as a valuable model for improving educational methods, potentially inspiring new approaches to project-based learning and collaborative work. As a result, educators may be more equipped to incorporate digital task management systems into their instructional strategies, thereby helping students develop better organizational skills.

This study's results could also inform educational policies by demonstrating the benefits of implementing digital task management systems at an institutional level. Should the findings show a positive impact on student productivity and task accountability, UNIMAS and other educational institutions might consider adopting similar digital tools more broadly. This could encourage policies that prioritize digital skill-building in student curricula, emphasizing practical tools for organization and time management.

1.9 Project Schedule

This Final Year Project (FYP), scheduled for the 2024/2025 academic session, will be completed over two semesters, divided into Final Year Project I (FYP I) and Final Year Project II (FYP II). FYP I will cover Chapter 1 (Introduction), Chapter 2 (Literature Review), and Chapter 3 (Methodology), focusing on the project's background, objectives, theoretical framework, and research design. FYP II will consist of Chapter 4 (Implementation and Testing), which includes the development, execution, and evaluation of the project, and Chapter 5 (Conclusion), summarizing findings, limitations, and recommendations. A detailed project schedule with milestones and deadlines is provided in Appendix A.

1.10 Expected Outcomes

The expected outcomes of this project focus on providing UNIMAS students with a straightforward task management system to help organize, assign, and track tasks easily.

One key outcome is better teamwork. By centralizing all task assignments and progress tracking in one place, every team member can clearly see their responsibilities and stay updated on the team's progress. This reduces confusion, overlap, and miscommunication, helping students work toward shared goals more effectively.

The system is also expected to boost productivity. Features like task prioritization and automated reminders keep students focused on completing their most important tasks first, reducing the time spent on manual tracking. With a structured approach to task management, students can handle their workload better, making project work faster and more efficient.

Another outcome is improved accountability. Each task is assigned to a specific person with a deadline, and progress updates are visible to the whole team. This transparency motivates team members to stay on top of their work and makes individual contributions clear to everyone.

Finally, the system will reduce miscommunication by giving all team members access to the same task details, deadlines, and status updates. With everyone on the same page, teams can avoid unnecessary follow-ups and work together more smoothly, with a clear understanding of project goals and timelines.

In summary, this task management system is expected to help students stay organized, accountable, and productive. By improving task tracking and team collaboration, it aims to enhance both individual time management and overall project success.

1.11 Project Outline

This project is outlined into 5 significant chapters as below:

Chapter 1: Introduction

This chapter provides an overview of the project, including its background, problem statement, objectives, scope, and significance. It explains the need for the Smart Task Management System (STMS) and how it aims to enhance productivity and collaboration among UNIMAS students.

Chapter 2: Literature Review

This chapter explores existing research, frameworks, and related task management systems. It analyses their strengths and limitations to identify gaps that the STMS aims to address. Additionally, it discusses key technologies, methodologies, and best practices relevant to the system's development.

Chapter 3: Methodology

This section outlines the Agile development methodology used for STMS, detailing the planning, design, development, and testing phases. It explains the tools, techniques, and frameworks applied in system development, along with stakeholder involvement and requirement gathering processes.

Chapter 4: Implementation and Testing

This chapter details the technical implementation of the system, including the development of front-end and back-end components, database integration, and security measures. It also covers system testing, user acceptance testing (UAT), and performance evaluation based on user feedback and test results.

Chapter 5: Conclusion and Future Works

The final chapter summarizes the project’s achievements, highlighting its contributions and effectiveness in addressing the identified problems. It also discusses limitations and proposes future improvements, such as additional features, scalability, and AI-driven enhancements for better task management.

Chapter 2: Literature Review

2.1 Introduction

The development of a smart task management system requires a thorough understanding of the existing body of knowledge, technologies, and systems in the domain of task management. This chapter reviews the relevant literature to establish a foundation for the design and implementation of the proposed system. It explores the strengths and limitations of current methodologies, tools, and technologies to identify gaps and opportunities for innovation.

The chapter begins with a current scenario analysis, examining the challenges and trends in task management, particularly in academic and team-based environments. This is followed by a review of similar systems, focusing on existing task management solutions, their features, and their usability. A comparison of these systems highlights the differences, strengths, and weaknesses of the reviewed systems, providing insights into how the proposed system can improve upon them.

Next, a discussion on the proposed system outlines the envisioned improvements and unique contributions of the smart task management system. This section bridges the gap between the existing systems and the specific needs of UNIMAS students.

By synthesizing insights from the current scenario, existing systems, and technical needs, this chapter lays a robust foundation for the smart task management system, ensuring that it addresses the identified challenges and aligns with project objectives.

2.2 Current Scenario Analysis

Effective task management plays a crucial role in improving productivity and achieving goals, especially in collaborative environments such as universities. However, many students and teams still rely on traditional methods like paper-based lists, email threads, or unorganized spreadsheets to manage their tasks. These outdated approaches often lead to issues such as missed deadlines, lack of accountability, and communication breakdowns, which hinder overall efficiency.

In academic settings, where students frequently work on multiple assignments, projects, and group tasks, the absence of a centralized system for tracking progress and deadlines exacerbates the challenges. Task management becomes even more complicated when working in teams, as coordination and communication issues arise due to unclear responsibilities and poor task visibility.

While some existing digital tools and platforms aim to address these challenges, many fall short in catering to the specific needs of students. Commonly used tools either focus heavily on professional or corporate environments or lack features like task prioritization, deadline tracking, or collaborative functions tailored for academic purposes.

This scenario underscores the pressing need for a system designed specifically for students, which combines the simplicity of traditional tools with the functionality of modern digital task management. The proposed smart task management system seeks to bridge this gap, offering features like task prioritization, progress tracking, notifications, and collaboration tools, all tailored to meet the unique requirements of UNIMAS students.

2.3 Review on Similar System

2.3.1 TaPro (TaskProgress); A System to Monitor Progress of Students' Task

The developer of this task management system aims to address significant challenges students face when working on group assignments or projects. These challenges often include a lack of accountability, difficulty tracking individual contributions, and inefficiencies caused by task redundancy. The system is designed specifically for students, focusing on creating a transparent and collaborative environment that enhances teamwork and ensures fair distribution of responsibilities (Madang, 2020).

A core goal of the developer is to improve accountability within student teams (Madang, 2020). By implementing features such as progress bars and evidence uploads, the system ensures that team members can monitor each other's contributions (Madang, 2020). This addresses a common issue where some team members fail to contribute, causing frustration and an uneven workload (Madang, 2020). The visibility of each member's progress fosters a sense of responsibility and accountability (Madang, 2020).

Another key objective is to reduce redundancy in tasks (Madang, 2020). By allowing tasks to be assigned and tracked clearly, the system prevents situations where multiple members unknowingly work on the same task (Madang, 2020). This not only saves time but also ensures that the team's efforts are focused and productive, leading to more efficient project completion.

The developer also seeks to encourage team collaboration and productivity. By providing tools that clearly define tasks and responsibilities, the system ensures that all team members are aligned in their objectives (Madang, 2020). This fosters a systematic approach to completing

assignments, helping teams manage their workload effectively and avoid last-minute rushes (Madang, 2020).

Additionally, the system simplifies progress tracking through visual tools such as progress bars (Madang, 2020). These tools offer real-time updates on task completion, allowing teams to identify delays or bottlenecks and take corrective action (Madang, 2020). This feature enhances the team's ability to meet deadlines and deliver high-quality work (Madang, 2020).

The system is designed with simplicity and accessibility in mind (Madang, 2020). Recognizing that students may not be familiar with complex professional tools, the developer focuses on essential functionalities such as adding, deleting, editing, and searching tasks (Madang, 2020). This ensures that the system is easy to use, even for those with minimal technical knowledge (Madang, 2020).

In conclusion, the developer's approach is thoughtful and student-centric, aiming to create a task management system that directly addresses the unique needs of student teams. By prioritizing accountability, collaboration, and ease of use, the system has the potential to significantly improve group project experiences for students. Additional features such as AI for task prioritization or integration with mobile platforms could further enhance its utility and impact.

2.3.2 FYP Online Management System (FYPOS)

The developer aims to modernize the management of Final Year Projects (FYP) by transitioning from a manual, paper-based system to an efficient web-based platform called the FYP Online Management System (FYPOS). This system is designed to address the inefficiencies of the current process, such as time-consuming administrative tasks, communication barriers, and challenges with data archiving (Binti et al., 2012). The proposed solution introduces a streamlined and systematic approach to managing FYPs, benefiting students, supervisors, and coordinators alike (Binti et al., 2012).

One of the key objectives of the developer is to create a platform that improves communication and ensures smooth information exchange between all stakeholders (Binti et al., 2012). In the manual system, the lack of a centralized communication medium often led to delays and misunderstandings (Binti et al., 2012). The proposed system resolves this by providing a unified platform where updates, task allocations, and notifications can be shared efficiently (Binti et al., 2012). This fosters better collaboration and ensures timely progress in project completion (Binti et al., 2012).

The developer also aims to eliminate the inefficiency of managing physical documents and forms (Binti et al., 2012). The traditional system requires significant manpower and time for registration, task allocation, and assessment (Binti et al., 2012). Additionally, archiving physical documents poses challenges, as data retrieval can be cumbersome, and there is a risk of loss or misplacement (Binti et al., 2012). By digitizing these processes, the FYPOS enables secure, efficient data storage and retrieval, reducing the reliance on physical documentation (Binti et al., 2012).

The proposed system incorporates several innovative features to ensure its effectiveness (Binti et al., 2012). It is designed to be scalable, accommodating a growing number of users and large datasets without compromising performance (Binti et al., 2012). The system is also flexible, allowing dynamic updates to project requirements or data. To ensure organized operations, administrative isolation is implemented, creating a structured hierarchy for managing access and permissions (Binti et al., 2012). Furthermore, the system is robust, capable of adapting to reconfigurations and recovering from potential failures (Binti et al., 2012).

The developer adopts a systematic and well-structured methodology for the project. In the initial phase, extensive research and planning were conducted to define system requirements and functionalities (Binti et al., 2012). The project is divided into two key areas: web and server design, which is the primary developer's focus, and database design, managed by a collaborator (Binti et al., 2012). The development process is phased, with milestones for database setup, web application development, and system testing, ensuring timely progress and reliability (Binti et al., 2012).

In conclusion, the FYP Online Management System is a practical and innovative solution to the challenges faced in traditional FYP management. By addressing key inefficiencies, enhancing communication, and incorporating advanced features, the system has the potential to significantly improve workflows, reduce administrative burdens, and provide a seamless experience for all stakeholders involved in the FYP process.

2.3.3. Web-Based Student Task Management System for students at University College of Yayasan Pahang

The developers of the Web-Based Student Task Management System have undertaken a commendable project aimed at addressing a significant challenge faced by students, particularly in the context of remote learning due to the COVID-19 pandemic. The system is designed to help students at University College of Yayasan Pahang manage their academic

tasks more effectively (Nurzi & Ab Wahab, 2022). Given the constraints of virtual learning environments, where students struggle with communication and time management, the developers focused on providing a comprehensive tool that assists with task tracking, prioritization, and deadline management (Nurzi & Ab Wahab, 2022).

The system's core features include task tracking, note management, event calendaring, and group collaboration tools (Nurzi & Ab Wahab, 2022). These functionalities are essential for students who need to manage both individual assignments and collaborative projects (Nurzi & Ab Wahab, 2022). The system's task tracking feature allows students to monitor their progress, while the recommendation tool helps prioritize tasks based on urgency and importance, ensuring that students stay on track with their work (Nurzi & Ab Wahab, 2022). Additionally, the platform's ability to support group work via task pages and chat rooms enhances collaboration among peers, which is often a challenge when working remotely (Nurzi & Ab Wahab, 2022).

The developers chose a modern technology stack to ensure scalability, performance, and a seamless user experience. The front-end of the system is built using Next.js and TypeScript, while the back-end is powered by Node.js and GraphQL, with data stored in MongoDB Atlas on the cloud (Nurzi & Ab Wahab, 2022). Firebase authentication is employed to streamline user login via Google or Facebook accounts (Nurzi & Ab Wahab, 2022). This choice of technologies ensures that the system is not only robust but also user-friendly, making it easier for students to adopt and use the platform without unnecessary barriers (Nurzi & Ab Wahab, 2022).

Adopting the Agile development methodology, specifically the Scrumban framework, demonstrates a strategic approach to software development (Nurzi & Ab Wahab, 2022). The Scrumban methodology blends the best of Scrum and Kanban, enabling the developers to work iteratively through sprints, managing the development process in smaller, manageable stages (Nurzi & Ab Wahab, 2022). This approach ensures that the system is built incrementally, with regular testing and feedback, reducing the likelihood of major errors or inefficiencies (Nurzi & Ab Wahab, 2022). The use of Test-Driven Development (TDD) and continuous integration further highlights the developers' commitment to maintaining high-quality standards throughout the process (Nurzi & Ab Wahab, 2022).

Overall, the Web-Based Student Task Management System represents a thoughtful and well-executed solution to a critical problem. The developers have used a robust technological

framework, adhered to best practices in development, and designed the system with the needs of students in mind. With continuous user feedback and potential improvements, the system could become an indispensable tool for students navigating the complexities of remote learning and academic life.

2.3.4 Todoist

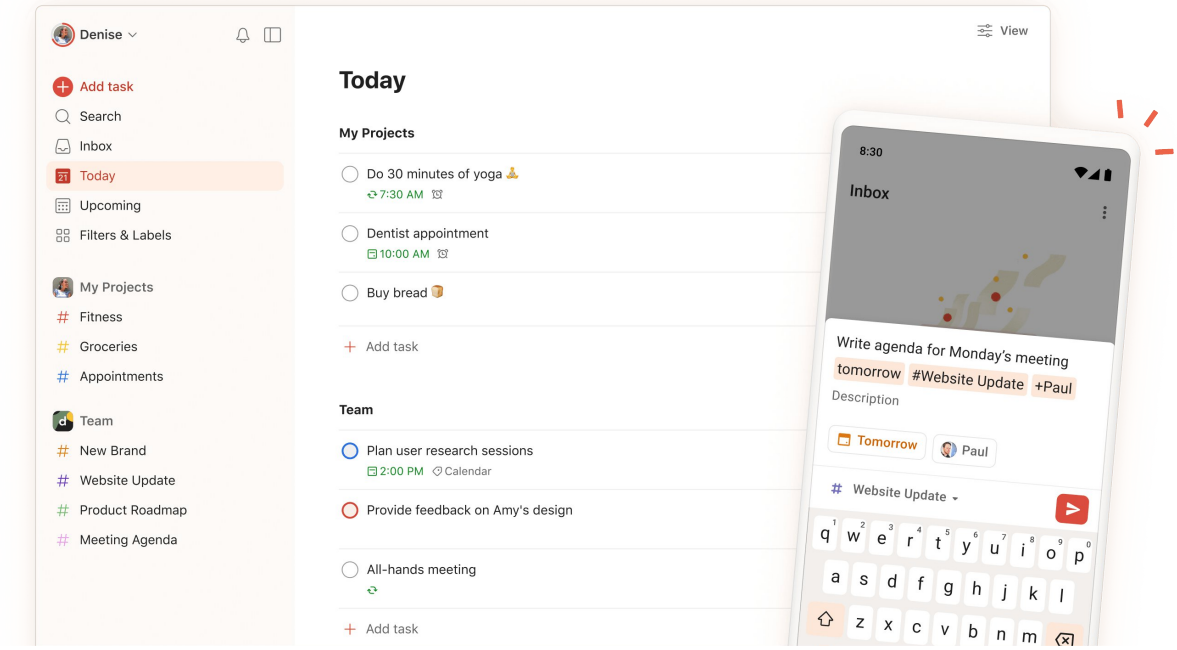


Figure 2.1 Todoist Overview

Todoist is a versatile task management application designed to help users organize and prioritize their personal and professional responsibilities efficiently (Doist, 2007). The app allows users to create tasks with natural language input for deadlines and reminders (Doist, 2007). Tasks can be organized into projects and sections, which help categorize and manage them systematically (Doist, 2007). Subtasks, priority levels, and due dates add further granularity to planning, enabling users to focus on what matters most (Doist, 2007).

One standout feature of Todoist is its collaborative capabilities (Doist, 2007). Shared projects allow teams or families to assign tasks, share comments, and attach relevant files, making it a great tool for group organization (Doist, 2007). Todoist's integrations with calendars like Google Calendar provide real-time syncing, ensuring tasks and schedules remain updated (Doist, 2007). Recurring tasks and customizable templates are also available, streamlining the setup for repetitive or similar projects (Doist, 2007).

The app's design encourages daily use through views like "Today" and "Upcoming," which show tasks in order of priority or schedule (Doist, 2007). Regular reviews can be set up as

recurring tasks, helping users adjust plans and maintain progress (Doist, 2007). For teams, Todoist offers features like shared workspaces and permissions for public or private projects, making it adaptable to different working environments (Doist, 2007).

2.3.5 Trello

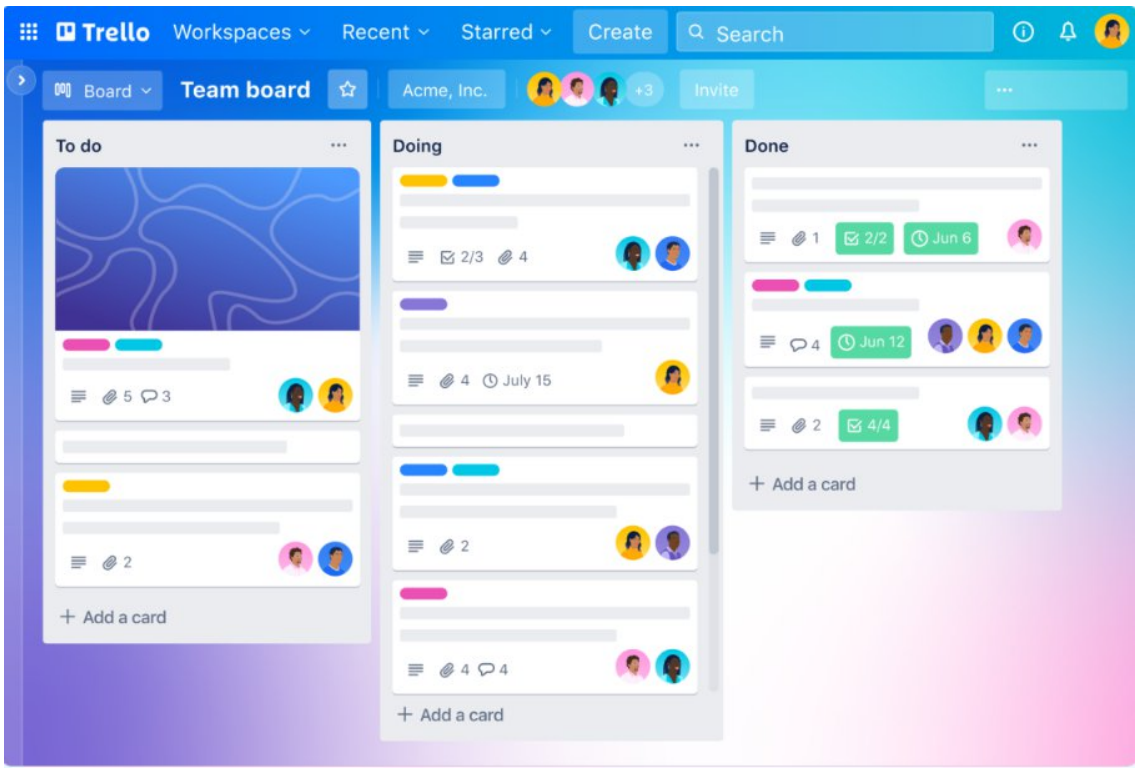


Figure 2.2 Trello Overview

Trello is a versatile project management tool designed to streamline workflows and enhance team collaboration through a visually organized system (Moon, 2022). The platform operates using "boards," which represent projects or initiatives, broken down into "lists" that structure phases or categories of work. Each list contains "cards," representing tasks or individual work items (Moon, 2022). Users can drag and drop cards between lists to signify progress, such as moving a task from "To Do" to "In Progress" and eventually to "Done" (Moon, 2022).

Trello supports customization to suit various needs, such as personal productivity, team projects, or departmental workflows (Moon, 2022). The cards can be enriched with details like descriptions, attachments, labels, due dates, and checklists (Moon, 2022). It also integrates with various third-party apps and services through "Power-Ups," which expand its functionality, including calendar views, file sharing, and task automation (Moon, 2022). Trello's flexibility extends to its pricing options, offering a free version for individuals and

premium plans for businesses requiring advanced features and collaboration tools (Moon, 2022).

The platform excels in offering multiple views to track work progress, including Kanban boards, timelines, and calendars. These features make Trello adaptable for different project sizes, from personal task management to large-scale team collaboration (Moon, 2022). Automation features like Butler allow users to set rules and actions, such as automatic task assignments or notifications, saving time and reducing manual efforts (Moon, 2022).

Trello is particularly popular for its user-friendly interface and adaptability across various industries, including marketing, product management, and engineering (Moon, 2022). Its visual organization system and integrations with tools like Slack and Google Drive make it a preferred choice for enhancing productivity (Moon, 2022).

2.3.6 Asana

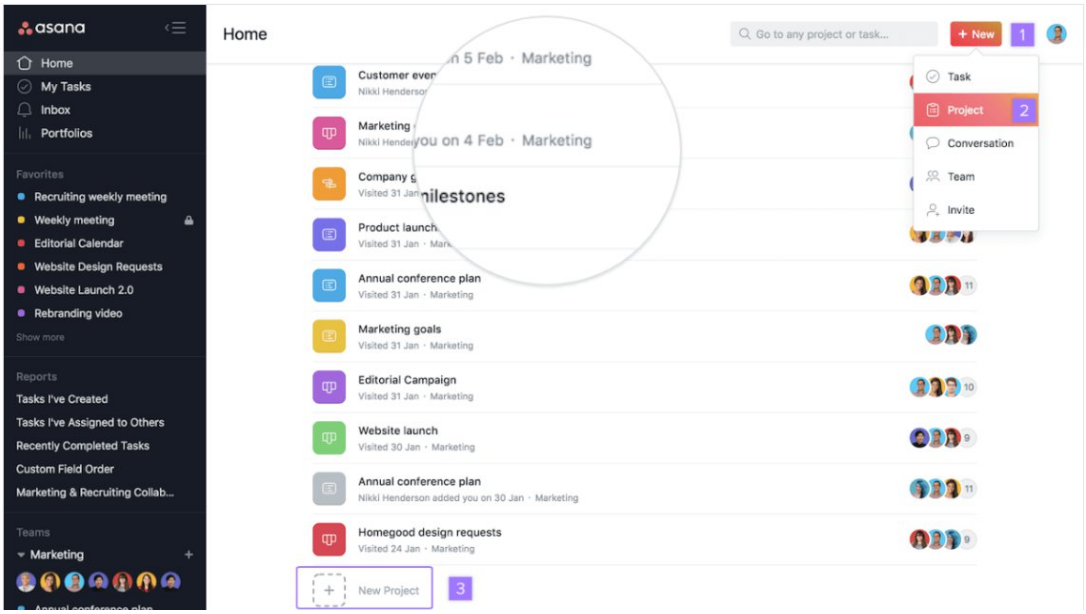


Figure 2.3 Asana Overview

Asana is a task and project management tool designed to enhance individual and team productivity (Help Center Asana, 2008). It works by centralizing tasks, projects, and communications in one place, ensuring clarity and collaboration among team members (Help Center Asana, 2008).

At its core, Asana organizes work through tasks, which are the fundamental units of productivity. Tasks can be grouped into projects, representing broader initiatives. Projects can be viewed in multiple formats such as lists, Kanban boards, timelines, or calendars, catering to

diverse workflows (Help Center Asana, 2008). Each task can include details like descriptions, due dates, assignees, attachments, and subtasks, enabling users to manage complex responsibilities systematically (Help Center Asana, 2008).

Asana also supports advanced features like workflow automation through rules and triggers, allowing teams to reduce repetitive tasks (Help Center Asana, 2008). Its Portfolios feature lets managers oversee multiple projects and track progress across teams, providing insights into performance and bottlenecks (Help Center Asana, 2008). With integration capabilities for tools like Slack, Zoom, and Google Drive, Asana seamlessly fits into existing work ecosystem (Help Center Asana, 2008).

One standout aspect of Asana is its focus on real-time collaboration (Help Center Asana, 2008). Team members can leave comments, share updates, and track changes directly within tasks, reducing dependency on email threads (Help Center Asana, 2008). Notifications ensure that users stay informed about task updates and deadlines, fostering accountability and coordination (Help Center Asana, 2008).

Overall, Asana is a versatile platform suitable for diverse organizational needs, from small teams managing simple workflows to large enterprises coordinating complex projects. It is accessible on desktop and mobile, making it a reliable tool for modern, distributed teams.

2.4 Features of Existing System

Table 2.1 Key Features of Existing System

System	Key Features
TaPro (TaskProgress)	- Progress tracking with visual bars and evidence uploads.
	- Promotes accountability by monitoring individual contributions.
	- Task assignment to avoid redundancy.
	- Collaboration tools for teamwork and task definition.
	- Simple and accessible features like adding, deleting, and searching tasks.
FYP Online Management	- Centralized platform for communication and

System	notifications.
	- Digitized document management replacing manual processes.
	- Scalable design for growing user bases and datasets.
	- Flexible system for adapting to updates and changes.
	- Organized data management with structured hierarchies.
Web-Based Student Task Management System	- Task management for tracking, prioritizing, and monitoring deadlines.
	- Group collaboration through task pages and chat rooms.
	- Event calendaring for schedules and deadlines.
	- Built on a modern tech stack (Next.js, TypeScript, Node.js, GraphQL, Firebase).
	- Agile development methodology (Scrumban) and Test-Driven Development (TDD).
Todoist	- Task organization with deadlines, reminders, and priority levels.
	- Project management with categorized sections and subtasks.
	- Collaboration features like shared projects and task assignments.
	- Integrations with Google Calendar and recurring tasks support.
	- “Today” and “Upcoming” views for task prioritization.
Trello	- Visual workflow with boards, lists, and cards for task

	tracking.
	- Customizable cards with labels, attachments, and checklists.
	- Automation via Butler for repetitive tasks.
	- Integrations with Slack, Google Drive, and other third-party apps.
	- Multiple views: Kanban boards, timelines, and calendars.
Asana	- Task grouping into projects with list, Kanban, timeline, and calendar views.
	- Workflow automation using rules and triggers.
	- Portfolio management for multi-project oversight.
	- Real-time collaboration with comments, updates, and shared communications.
	- Integrations with Slack, Zoom, Google Drive, and more.

2.5 Technology Used of The Existing System

The existing systems employ a variety of technologies and methodologies tailored to their specific purposes and user needs. TaPro (TaskProgress), while not explicitly detailed in its technological stack, is likely based on straightforward, user-friendly web or desktop applications designed to streamline task tracking for students. Its emphasis on accessibility suggests reliance on intuitive and lightweight technologies.

The FYP Online Management System (FYPOS) incorporates robust web and server design, emphasizing scalability and adaptability. The system uses a digitized database to manage documents efficiently, with structured hierarchical permissions for secure access control. Its design is both flexible and resilient, ensuring the platform can adapt to changes and recover from potential failures, which is crucial for academic environments managing large volumes of data.

The Web-Based Student Task Management System for students at University College of Yayasan Pahang uses a modern technology stack to ensure scalability and high performance. The front-end is developed using Next.js and TypeScript, providing a dynamic and user-friendly interface. The back-end, powered by Node.js and GraphQL, ensures efficient data querying and processing. Data storage is managed via MongoDB Atlas, a cloud-based database solution. Additionally, the system uses Firebase Authentication for secure and streamlined user logins via Google or Facebook accounts. The adoption of Agile methodology, specifically Scrumban, coupled with Test-Driven Development (TDD), ensures iterative progress with continuous testing and quality assurance.

Todoist leverages integration technologies such as Google Calendar synchronization to offer real-time updates on tasks and schedules. Its focus on personal and collaborative task management includes features like customizable templates, shared projects, and team task assignments, all designed for seamless daily usage. Similarly, Trello adopts a Kanban-based task management system, enabling users to visually organize workflows. With integrations like Slack and Google Drive and automation via Butler, Trello enhances productivity and reduces manual effort for repetitive tasks.

Finally, Asana integrates advanced task and project management tools, offering multiple viewing formats such as lists, Kanban boards, and timelines. The system's workflow automation capabilities, powered by rules and triggers, streamline repetitive processes. Asana also supports extensive integrations with tools like Slack, Zoom, and Google Drive, ensuring compatibility with existing workflows. Its focus on real-time collaboration, combined with accessible desktop and mobile platforms, makes it versatile for teams of all sizes.

In summary, the technologies used across these systems highlight a blend of modern development frameworks, scalable back-end solutions, and user-centric interfaces. These choices reflect the developers' focus on enhancing usability, efficiency, and collaboration for their target audiences.

2.6 Discussion of Proposed System

The proposed Smart Task Management System for UNIMAS students is a response to the challenges faced by students in managing academic tasks, deadlines, and collaborative projects. It aims to create a structured, efficient, and user-friendly platform tailored to the unique needs of UNIMAS students. By integrating key features such as task prioritization, automated

notifications, and progress tracking, the system aspires to enhance time management, improve task visibility, and facilitate seamless team collaboration.

At the core of the system is its task management capability, enabling users to create, assign, and track tasks with ease. Students will be able to categorize their activities, set deadlines, and monitor task statuses such as "Not Started," "In Progress," or "Completed." The task prioritization feature will help students determine which tasks should be tackled first based on urgency and importance. This functionality will guide students to manage their workloads effectively, preventing missed deadlines and reducing stress during peak academic periods.

The platform will also emphasize improving communication and collaboration, especially for group assignments. Shared task boards and real-time updates will provide all team members with a clear view of progress and responsibilities. Notifications and reminders for approaching deadlines or overdue tasks will ensure that everyone stays on track, minimizing the risk of miscommunication and task duplication.

Accessibility and usability are central to the design of this system. Developed as a web-based platform, it will be accessible through modern browsers on desktops and laptops. The intuitive interface will ensure that students of varying technical abilities can navigate the system effortlessly. Features such as a simple layout, clear task categorization, and easy-to-understand visuals will enhance user experience and engagement.

The proposed system will also include basic reporting tools to evaluate individual and team performance. Summarized insights into task completion rates, overdue tasks, and upcoming deadlines will help students reflect on their productivity and identify areas for improvement. These reports can also support educators in monitoring student engagement and project progress.

By implementing this Smart Task Management System, UNIMAS students will gain access to a robust tool that promotes better organization, reduces inefficiencies, and fosters accountability. The system’s focus on task prioritization, real-time updates, and ease of use will make it a valuable resource for enhancing academic performance and collaboration within the university community.

2.7 Summary Chapter 2

The literature review highlights the critical role of task management systems in addressing challenges associated with organizing and tracking tasks in dynamic academic and professional

environments. These systems aim to enhance productivity, time management, and collaboration by providing structured solutions for task delegation, tracking, and communication.

Research indicates that conventional task management methods, such as spreadsheets and email communication, often fall short due to their lack of automation, real-time updates, and prioritization capabilities. These limitations can lead to inefficiencies, such as missed deadlines, miscommunication, and low accountability, particularly in collaborative academic settings like universities.

For university students, particularly those engaged in project-based learning or group assignments, task management systems offer specific benefits. These include streamlined task assignment, real-time notifications, and performance analytics, which collectively improve team communication and individual accountability. Moreover, this system predicts task urgency or recommend schedules demonstrate the potential to transform traditional task management approaches into more proactive and data-driven processes.

The findings from the literature emphasize the need for customized task management solutions that cater to the unique needs of academic environments. By focusing on user-friendly designs, intelligent features, and reliable reporting tools, such systems can significantly improve productivity, foster collaboration, and support better decision-making. The insights also highlight the value of continuous feedback and iterative development to ensure the systems remain relevant and effective for their intended users.

Chapter 3: Methodology

3.1 Introduction

Methodology refers to the analysis of the methods used appropriate to a field of study. It is a systematic way of accomplishing certain tasks and is defined as a collection of procedures, techniques, tools and documentation aids that helps a software developer to speed up and simplify the software development process.

In this research, a combination of methodologies was used. This chapter describes the methodology for this dissertation and the methodology that is used for system development.

3.2 Research Methodology

The purpose of the research methodology is to provide a comprehensive view of the methods applied in this research. Additionally, it defines the Agile methodology as the framework for developing the Smart Task Management System for UNIMAS students.

The research methodology outlined here is based on the methods used to collect information about the real-world problems faced by UNIMAS students in managing their tasks and team projects. Information was gathered through analysis studies, literature reviews, and feedback from questionnaires, which formed the foundation for identifying the initial system requirements.

The Agile methodology is employed as the development approach for this project. Unlike traditional methodologies, Agile emphasizes flexibility, iterative progress, and continuous feedback. This approach is particularly suited for the dynamic needs of task management systems, allowing for the incorporation of user feedback throughout the development process. The methodology begins with defining the initial requirements, which are then used to plan and execute development in iterative sprints. Each sprint focuses on delivering functional components of the system, starting from basic features such as task creation and tracking to more advanced functionalities like task prioritization and notifications.

The Agile process involves four main phases: planning, designing, development (coding), and testing. Each phase is revisited and refined based on stakeholder input and evaluation of the working prototype. This iterative process ensures that the system evolves to meet user needs

effectively, aligning with the overarching goal of improving task management and collaboration for UNIMAS students.

By adopting Agile methodology, this research aims to deliver a robust, user-centered solution, ensuring the system is functional, reliable, and adaptable to the unique requirements of its users.

3.2.1 Research Questions and Methodology

In this research, the researcher seeks to address the following questions to provide a comprehensive solution for the Smart Task Management System for UNIMAS Students:

1. How can a task management system improve the organization and tracking of academic and extracurricular tasks for UNIMAS students?
2. What features are essential for enhancing team communication and collaboration among UNIMAS students working on group projects?
3. How can task prioritization improve task management for students?
4. What are the challenges and limitations in developing a user-friendly, web-based task management system for UNIMAS students?
5. How effective is the proposed system in improving task management, time management, and productivity for UNIMAS students?

By addressing these questions, the research aims to deliver a well-rounded, practical, and effective task management solution tailored to the needs of UNIMAS students.

3.3 Research Technique

There are many techniques used to collect data. The Techniques used to collect this research are as follows:

- Literature review
- Survey (Google Form)
- Brainstorming

3.3.1 Literature Review

As outlined in Chapter 2, six task management systems were examined to identify common features and functionalities. The key attributes of these systems have been systematically

analyzed and are presented in Table 2.0 for a comprehensive comparison. This evaluation provides a foundation for understanding the capabilities of existing solutions and identifying potential enhancements for the proposed system.

3.3.2 Research Survey (Google Form)

The research survey conducted for the Smart Task Management System for UNIMAS Students provided valuable insights into current task management practices, challenges, and expectations among students. The survey, distributed via Google Forms, gathered responses from a diverse group of students across various faculties, with the majority from the Faculty of Computer Science and Information Technology. Key findings revealed that while many students rely on digital tools like Google Calendar, Excel, and task management apps, significant challenges persist, including forgetting deadlines, difficulty prioritizing tasks, and ineffective team communication. The most desired features for the proposed system include automated reminders, task prioritization, and tools for team collaboration, with a strong preference for real-time notifications and a user-friendly interface. Additionally, respondents emphasized the need for a mobile-compatible platform and visual dashboards for task tracking. Notably, 70% of students expressed a high likelihood of adopting the system, indicating strong demand for a tailored solution. These findings underscore the importance of incorporating these features into the system's design to effectively address students' needs and improve their task management capabilities.

3.4 Agile Methodology

Agile is a dynamic and flexible software development methodology designed to accommodate changing requirements and foster collaboration among team members and stakeholders. Unlike traditional linear approaches, Agile emphasizes iterative progress, breaking the development process into small, manageable cycles known as sprints. Each sprint delivers functional increments of the system, allowing for continuous testing, feedback, and improvement. This iterative nature ensures that the final product aligns closely with user needs and expectations. Agile values customer collaboration, encouraging active involvement of stakeholders throughout the development process to ensure the system evolves effectively. By focusing on adaptability, teamwork, and regular delivery of working solutions, Agile minimizes risks, enhances efficiency, and produces user-centered software that meets real-world demands.

3.5 Agile Methodology for Smart Task Management System for UNIMAS Student

For the development of the Smart Task Management System for UNIMAS students, the Agile methodology will be adopted to ensure flexibility, iterative progress, and continuous user feedback throughout the process. The system will be developed through a series of sprints, each lasting 2-3 weeks. Each sprint will focus on specific features and functionalities, ensuring that the system is continuously evolving to meet the needs of the users.

The phases within the Agile methodology for this system development include:

1. **Planning**: During this phase, the initial requirements for the Smart Task Management System will be gathered. This includes understanding user needs through surveys, prioritizing features, and creating a product backlog. The features will be organized based on importance and user feedback.
2. **Designing**: In this phase, the system's architecture, user interface (UI), and overall design will be outlined. The design will prioritize user experience, ensuring the system is intuitive and accessible for students. Initial wireframes and UI designs will be created, focusing on ease of use and efficiency.
3. **Coding**: The development team will implement the system's features according to the design. In the first few sprints, core functionalities such as task creation, assignment, and basic task tracking will be implemented. Following sprints will introduce advanced features, such as AI-based task prioritization and real-time notifications.
4. **Testing**: Testing will be an ongoing process throughout the Agile lifecycle. Each feature will undergo unit testing to ensure it works as intended. Additionally, integration testing will ensure that different components of the system work cohesively. User Acceptance Testing (UAT) will be conducted at the end of each sprint to gather feedback from students at UNIMAS, ensuring that the system meets their needs.
5. **Reviewing and Refining**: After each sprint, the development team will hold review meetings to gather feedback and identify areas for improvement. The system will be refined based on user feedback and any issues found during testing. This iterative process ensures that the system evolves to meet the expectations of users and continuously improves.

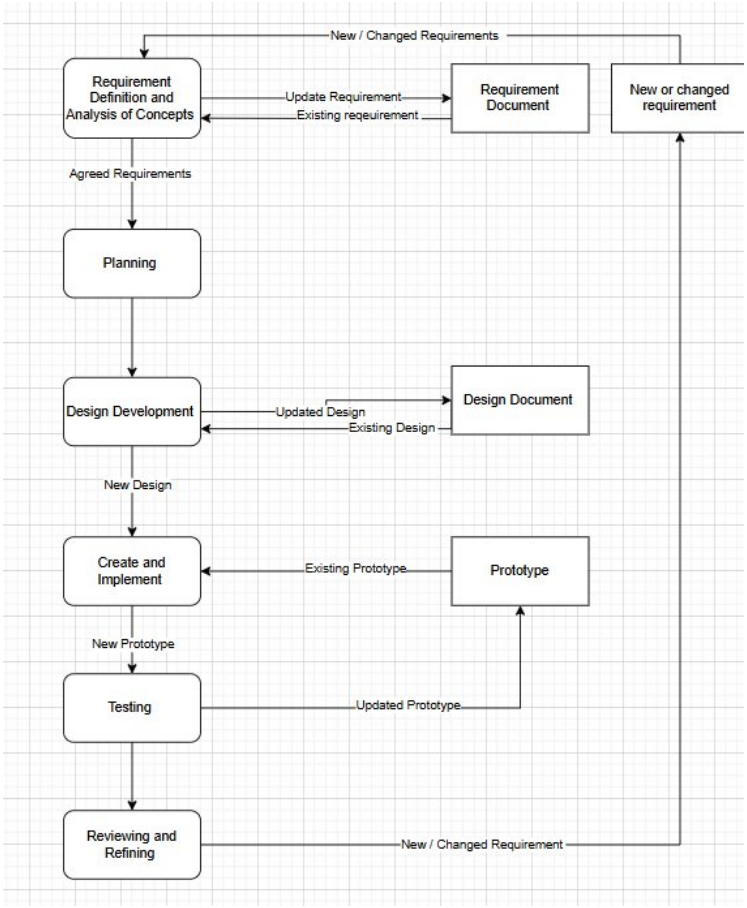


Figure 3.1 Flowchart for Agile Methodology STMS

By adopting the Agile methodology, the Smart Task Management System will be developed in a flexible, adaptive manner that allows for regular updates, user feedback, and refinement, ensuring that the final product is user-centered and effectively meets the task management needs of UNIMAS students.

3.5.1 Planning Phase for Smart Task Management System for UNIMAS Student

The Planning Phase in the Agile methodology for developing the Smart Task Management System (STMS) for UNIMAS students will establish a clear roadmap for the project, define the project scope, identify key stakeholders, and outline specific deliverables. This phase focuses on gathering requirements, defining user stories, setting priorities, and preparing for the iterative development process.

3.5.1.1 Project Scope

The goal of the Smart Task Management System (STMS) is to provide UNIMAS students with a comprehensive solution for task management, aimed at enhancing productivity, collaboration, and accountability in academic settings. The system will offer tools for task tracking, prioritization, deadline prediction, and seamless communication among team members, specifically tailored for student needs.

- Provide task prioritization.
- Enable seamless collaboration for group tasks.
- Enhance task visibility and accountability with real-time updates.
- Offer simple, intuitive UI/UX for students with varying technical skills.
- Ensure data security and privacy for student-related information.

3.5.1.2 Stakeholder Identification

In the Agile methodology, stakeholders play an important role in the planning phase. They provide the necessary feedback and collaborate throughout the development cycles. For the Smart Task Management System, the primary stakeholders are:

- Students (End Users): They will use the system to manage individual tasks and collaborate on group projects.
- Development Team: Includes developers, testers, and designers who will work on building and refining the system.
- System Administrators: Handle user access control and maintenance of the system.

3.5.1.3 Requirements Gathering

During this phase, detailed discussions will occur with stakeholders to gather functional and non-functional requirements. This process involves:

- Literature Review: Review existing systems and research to understand current challenges and solutions in task management for students.
- Survey: Conduct surveys using Google Form with UNIMAS students to gather insights on their task management needs, preferences, and pain points.
- Brainstorming: Organize brainstorming sessions with key stakeholders (student)

- Ideas from Supervisor: Gather feedback and suggestions from the project supervisor, incorporating expert opinions into the system design.

3.5.1.4 Defining the Product Backlog

Once the requirements are gathered, they will be translated into a Product Backlog then a prioritized list of features, enhancements, and fixes. Each backlog item is a user story that will be worked on in different sprints. Example user stories:

- Task Creation: As a student, I want to create tasks with due dates and priorities so that I can organize my assignments.
- Task Assignment: As a group member, I want to assign tasks to team members so that we can divide responsibilities clearly.
- Task Collaboration: As a student, I want to leave comments and attach files to tasks so that we can collaborate effectively.
- Task Suggestions: As a student, I want the system to suggest tasks to prioritize based on deadlines and importance.

The product backlog will be continuously refined based on stakeholder feedback, and some features might be split or modified based on the iteration progress.

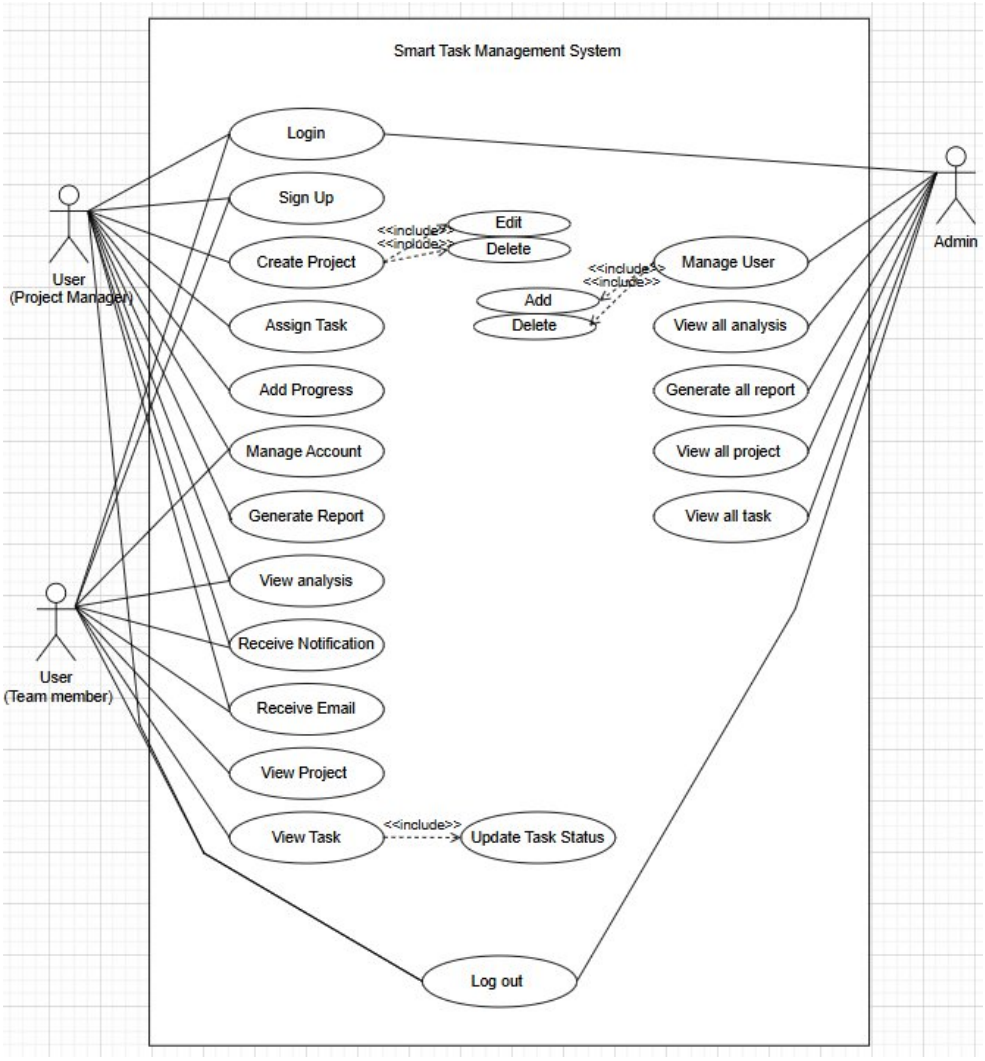


Figure 3.2 Use Case Diagram for STMS

3.5.1.5 Use Case Diagram

The Figure 3.2 illustrates the Use Case Diagram for the Smart Task Management System. This diagram serves to represent the functional requirements of the system by showing the interactions between various user roles (actors) and the system's functionalities (use cases).

3.5.1.5.1 Actors

There are three primary actors involved in the system:

1. Admin

   The Admin is responsible for overseeing user management and system monitoring activities. The Admin has access to all analytical and reporting functionalities and can view all tasks and projects in the system.

2. User (Project Manager)

The Project Manager is responsible for creating and managing projects. This includes assigning tasks to team members, adding progress updates, generating project-specific reports, and managing personal account settings.

3. User (Team Member)

   The Team Member is responsible for updating task statuses, viewing assigned tasks and projects, receiving notifications and emails, and viewing analysis relevant to their contributions.

3.5.1.5.2 Use Cases

The system supports the following main functionalities:

- Login / Sign Up: Allows users to authenticate or register in the system.
- Create Project: Enables Project Managers to create new projects. This use case includes the ability to edit or delete project information.
- Assign Task: Permits Project Managers to allocate tasks to team members.
- Add Progress: Allows Project Managers to input progress updates for tasks or projects.
- Manage Account: All users can update their personal account information.
- Generate Report: Enables Project Managers to produce task or project-specific reports.
- View Analysis: Allows users to visualize performance or task progress analysis.
- Receive Notification / Receive Email: Notifies users about task updates, changes, or reminders.
- View Project / View Task: Users can browse their assigned projects and tasks. Task viewing includes the ability to update the task status.
- Manage User: Admins can add or delete users from the system.
- View All Analysis / Tasks / Projects: Admins can access and monitor all records in the system.
- Generate All Report: Enables Admins to generate global reports across all projects.

3.5.1.5.3 Use Case Relationships

- The Create Project use case includes the Edit and Delete functionalities to ensure comprehensive project management.
- The Manage User use case includes Add and Delete to allow the Admin to handle user accounts.
- The View Task use case includes Update Task Status to facilitate progress tracking.

This use case diagram provides a high-level overview of the system's behavioral requirements and serves as a foundation for further development, including the design of sequence diagrams and class structures.

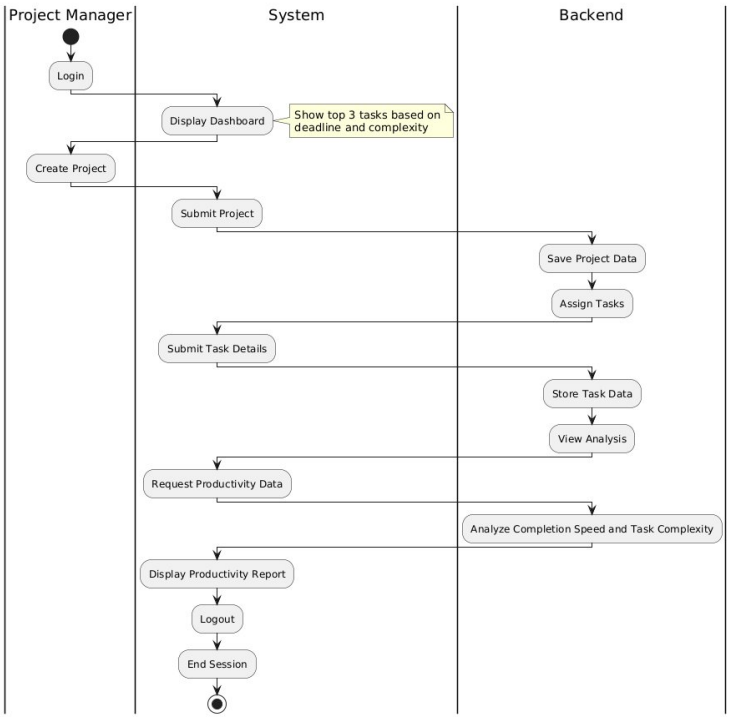


Figure 3.3 Activity Diagram for Project Manager

3.5.1.6 Activity Diagram

The activity diagram in Figure 3.3 for the Project Manager outlines the workflow involved in managing projects and monitoring performance within the Smart Task Management System. After logging in, the dashboard displays the top three tasks that require attention, selected based on the nearest deadlines and task complexity to help prioritize effectively. The Project Manager can then proceed to create new projects and assign tasks to team members, with each action interacting with the backend for data storage. Additionally, the Project Manager can access a productivity analysis report, which is generated based on the completion speed and complexity of tasks. The session concludes with the logout process, ensuring secure access control.

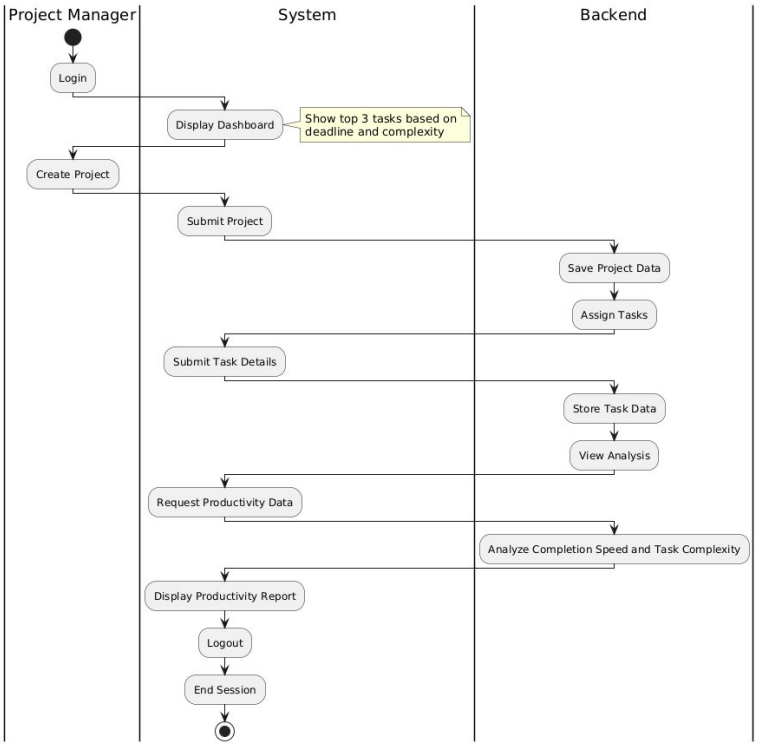


Figure 3.4 Activity Diagram for Team Member

The activity diagram for the Team Member illustrates in Figure 3.4 how individual users interact with the task management system to manage and report their work progress. After logging in, the system dashboard presents the top three priority tasks, selected according to upcoming deadlines and the complexity level of each task. Team Members can view their assigned tasks, select a specific task to update its progress, and optionally update the task's status. The system records this information in the backend. Additionally, Team Members can view a personalized productivity analysis that is calculated based on the speed at which they complete tasks relative to the tasks' complexity. The session ends with a secure logout operation.

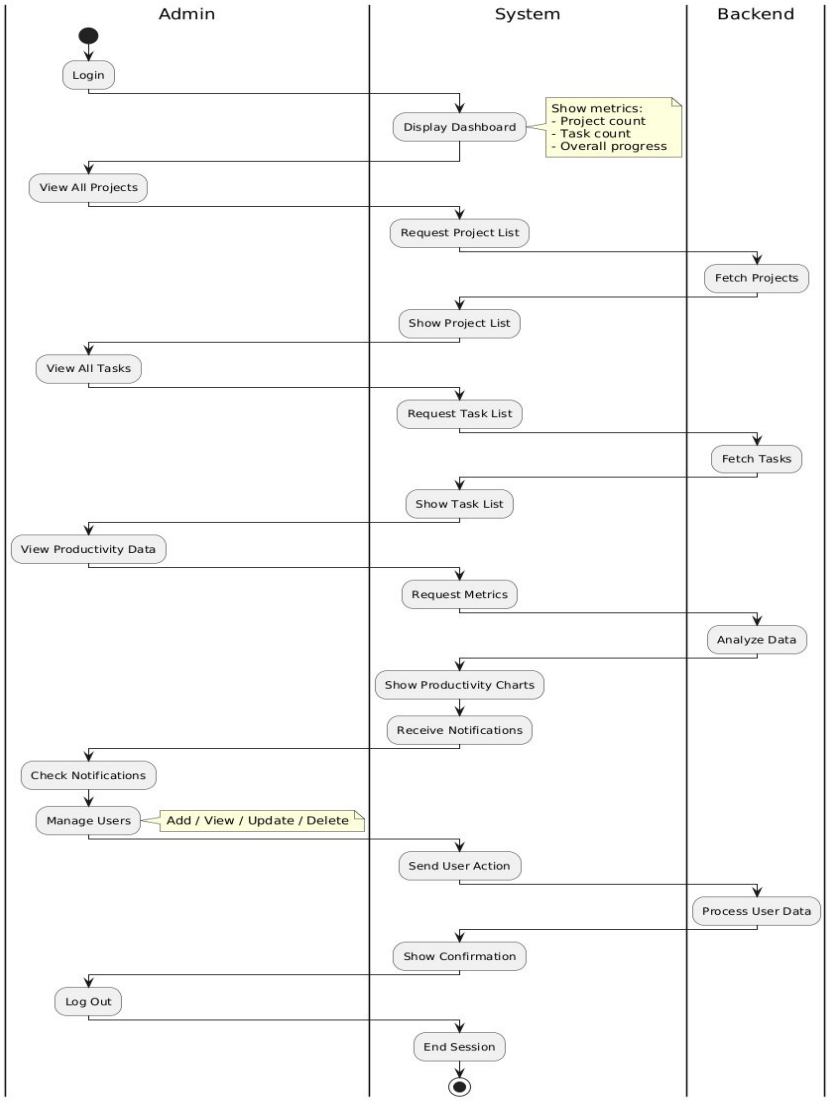


Figure 3.5 Activity Diagram for Admin

Figure 3.5 shows that the activity diagram for the Admin illustrates the administrative functions within the Smart Task Management System. After logging in, the Admin accesses a dashboard that presents key system metrics such as total project count, task volume, and overall progress. The Admin can view all projects and tasks, monitor user productivity, and check system-wide notifications. The Admin also has full user management capabilities, including adding new users, viewing the user list, updating details, and removing users. Each operation involves coordination between the system interface and the backend to process and retrieve data. The session ends with a standard logout process.

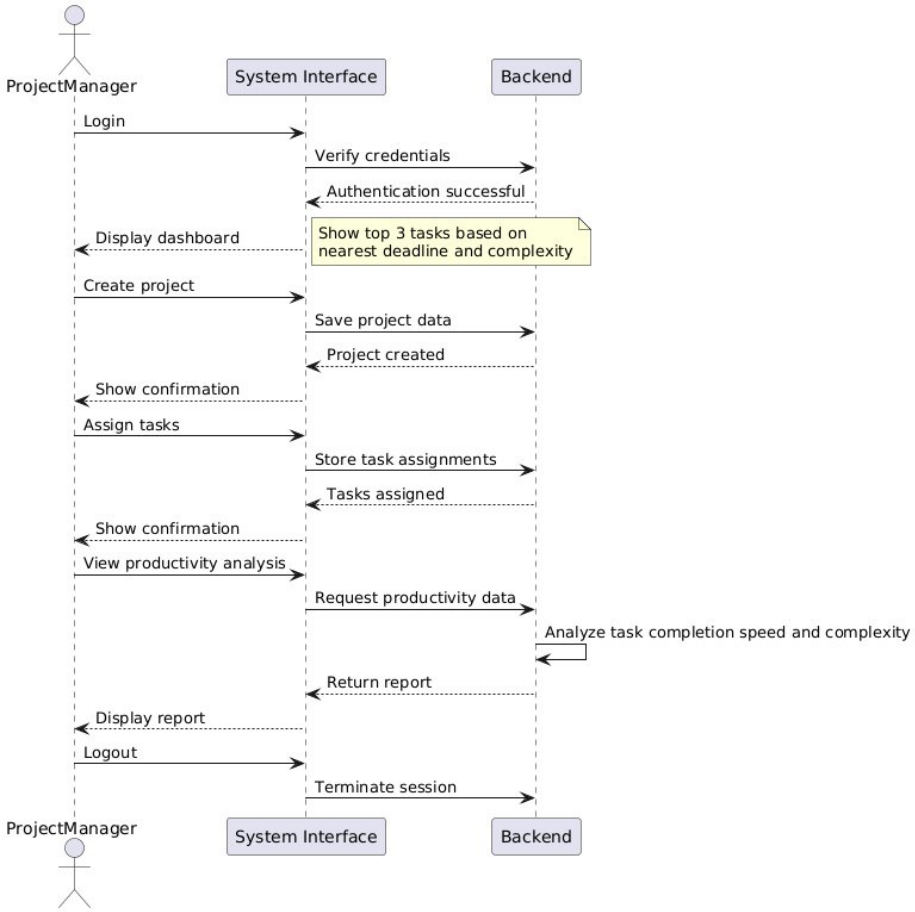


Figure 3.6 Sequence Diagram for Project Manager

3.5.1.7 Sequence Diagram

Figure 3.6 depicts the interaction sequence for the Project Manager. After successful login and dashboard access, the Project Manager creates new projects and assigns tasks to team members, with the system confirming each operation. The Project Manager can also request detailed productivity analysis reports to assess task completion efficiency and overall project progress. The sequence ends with the Project Manager logging out securely.

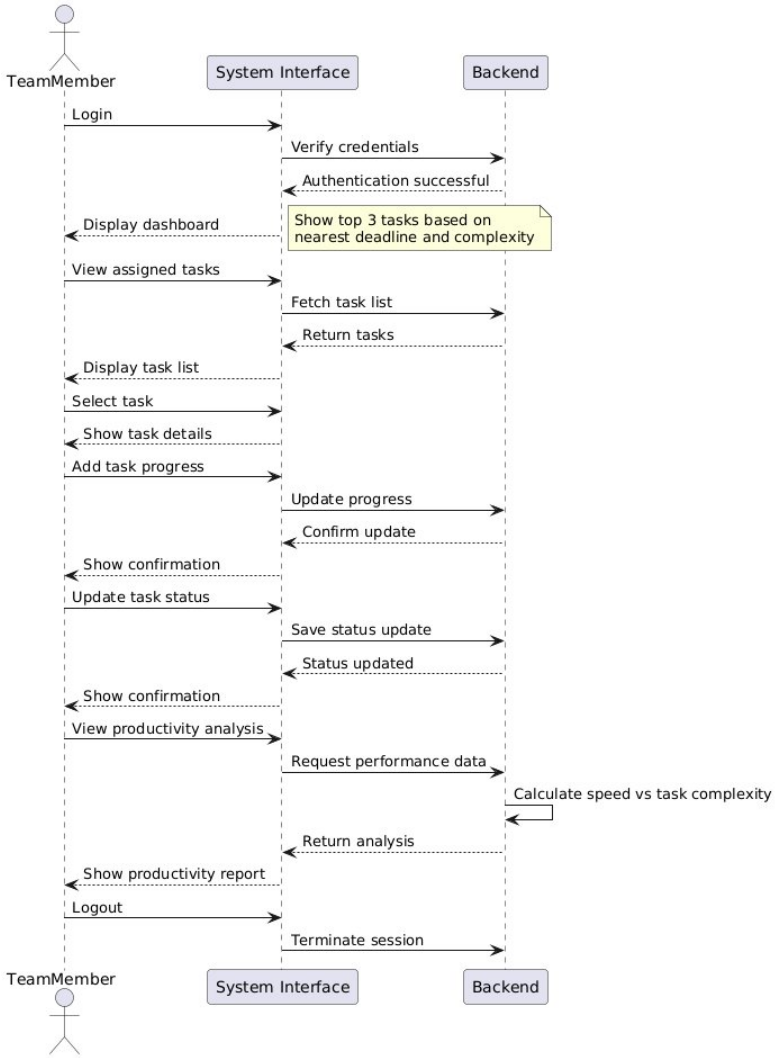


Figure 3.7 Sequence Diagram for Team Member

As illustrated in Figure 3.7, the Team Member begins by logging into the system, where their credentials are verified before granting access to a personalized dashboard featuring prioritized tasks. The Team Member can view assigned tasks, select specific tasks to see details, update task progress and status, and receive confirmation for each action. Additionally, they can request productivity reports that analyze task completion speed relative to complexity. The session concludes with a secure logout process.

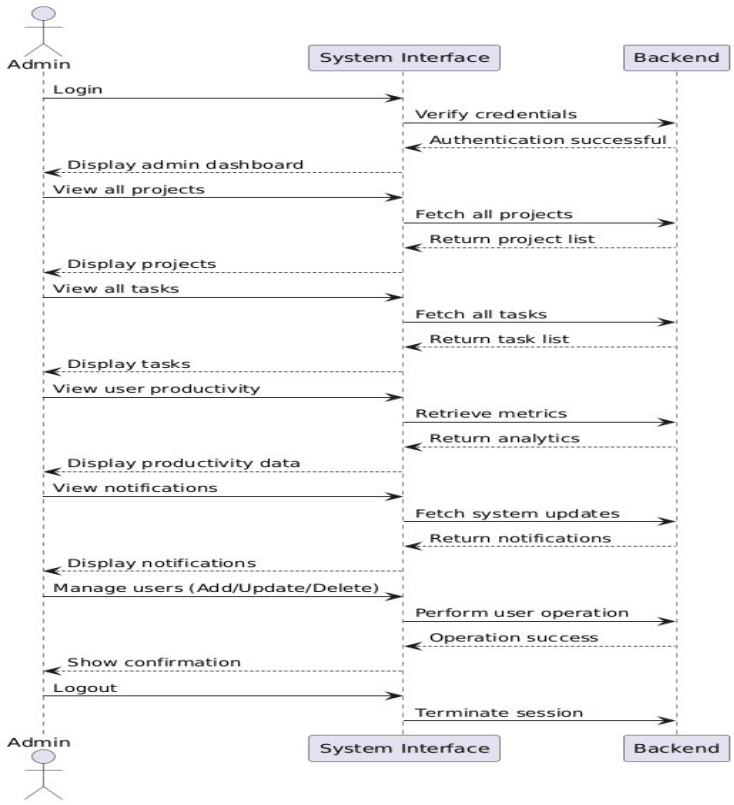


Figure 3.8 Sequence Diagram for Admin

As shown in Figure 3.8, the Admin logs into the system and accesses the administrative dashboard, which provides comprehensive views of all projects, tasks, and user productivity metrics. The Admin manages system notifications and performs user administration tasks such as adding, updating, or deleting users. Each management action is confirmed by the system. The session is terminated through a secure logout.

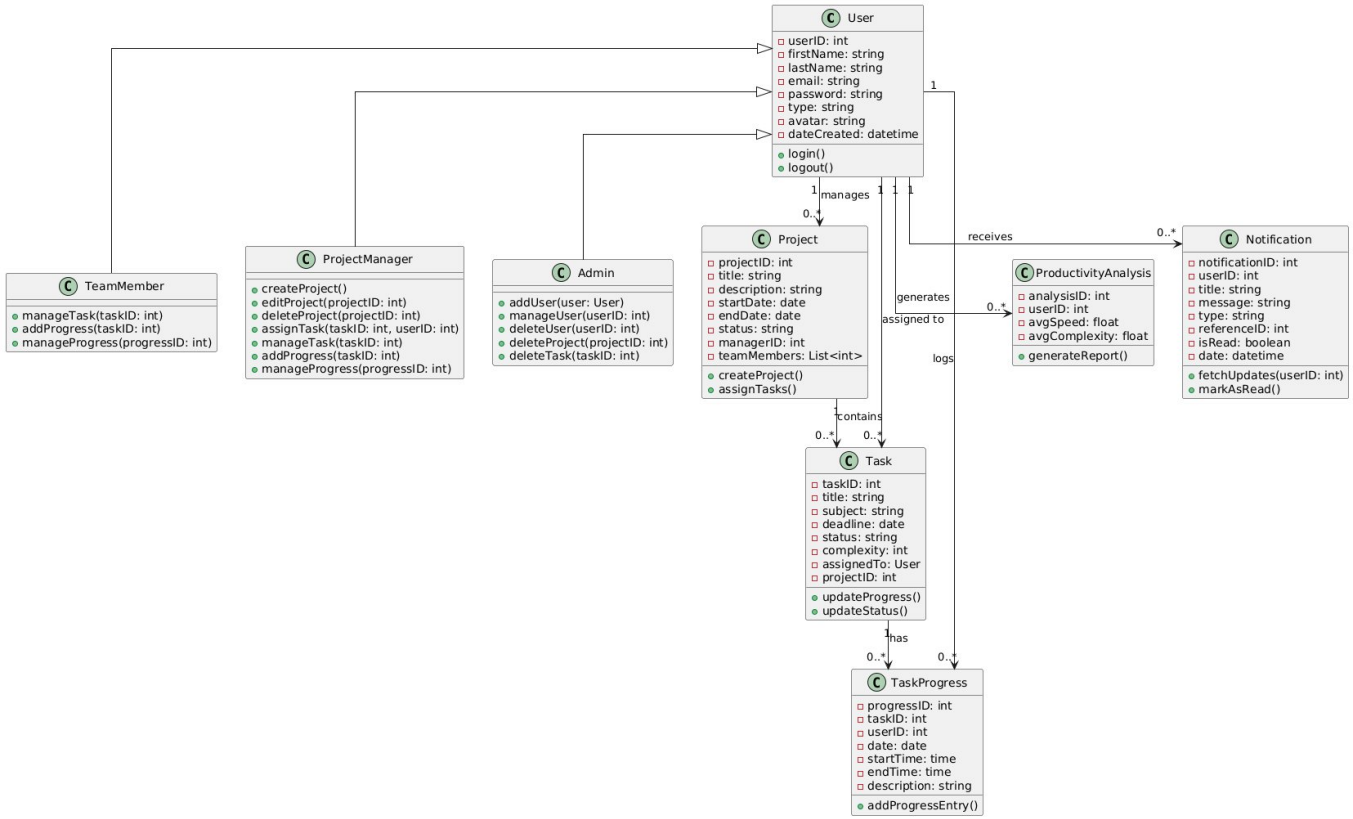


Figure 3.9 Class Diagram for STMS

3.5.1.8 Class Diagram

Figure 3.9 shows the class diagram of the task management system, where the User class is the base for Project Manager, TeamMember, and Admin roles, each with distinct functions. Project Managers can create, edit, delete projects, assign and manage tasks, and add progress updates, while Team Members manage their assigned tasks and update progress. Admins oversee user management and have rights to delete projects and tasks.

Projects contain multiple tasks, which are linked to users and progress tracked via the TaskProgress class. Additional classes like Productivity Analysis and Notification support performance reporting and system alerts, enabling clear role-based access and efficient management.

3.5.2 Designing Phase for Smart Task Management System for UNIMAS Student

The Designing Phase is a critical step in the development of the Smart Task Management System (STMS). It bridges the gap between planning and development, ensuring that the system's design aligns with the requirements gathered during the planning phase.

The primary goals of the Designing Phase are to establish a clear system blueprint, ensuring that the architecture and database design effectively guide development while supporting scalability and future growth. This phase emphasizes creating a user-centric design with an intuitive interface tailored to the academic needs of UNIMAS students. Additionally, it focuses on guaranteeing the system's scalability and security to handle user data safely and accommodate increasing demands over time.

The Use Case Diagram for the Smart Task Management System (STMS) illustrates the key functionalities and user interactions within the system. There are three primary actors: Students (Users), Group Members, and the System Administrator. Students can register, log in, create and manage tasks, assign tasks to group members, track task status, view deadlines, receive notifications, and edit their accounts. Group members can comment on tasks and update task progress, with the ability to attach files when commenting. The System Administrator is responsible for managing user accounts and maintaining system functionality. The diagram also includes relationships between use cases, such as the "Assign Tasks to Group Member" use case including the "Add/Delete" functionality, and the "Comment Task" feature. These relationships ensure a seamless workflow within the system. Overall, this diagram highlights the collaborative and task-tracking capabilities of STMS, ensuring that UNIMAS students can efficiently manage academic tasks, improve productivity, and enhance team collaboration.

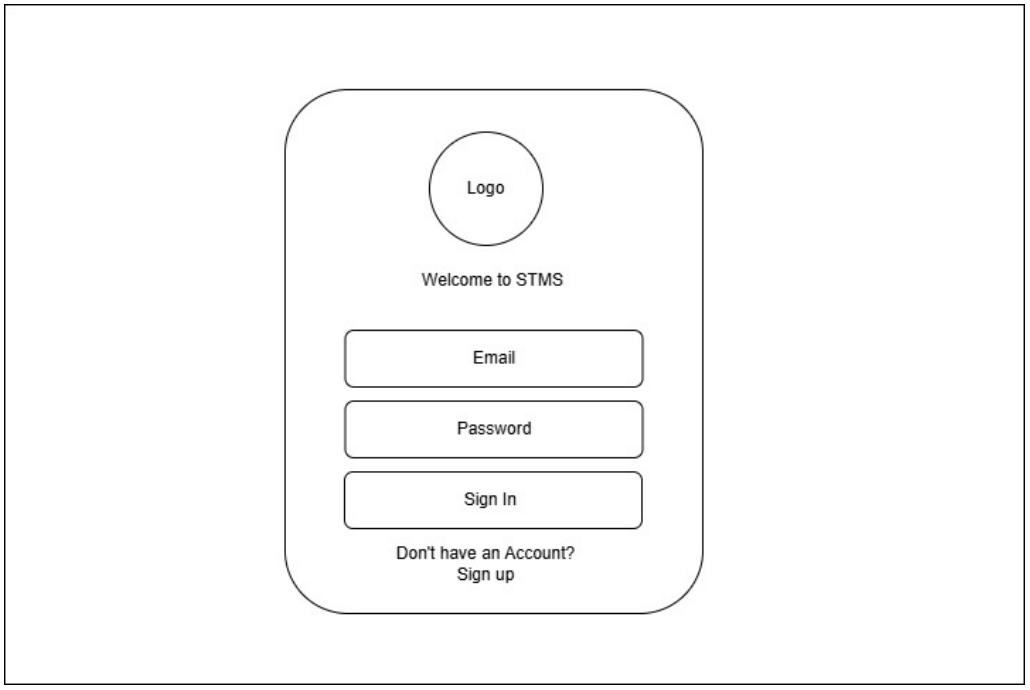


Figure 3.10 STMS Login

3.5.2.1 STMS Login Design

Figure 3.10 shows that the login function allows registered users which are Project Managers, Team Members, and Admins to securely access the system by entering their email and password. Upon successful authentication, users are redirected to their respective dashboards based on their roles.

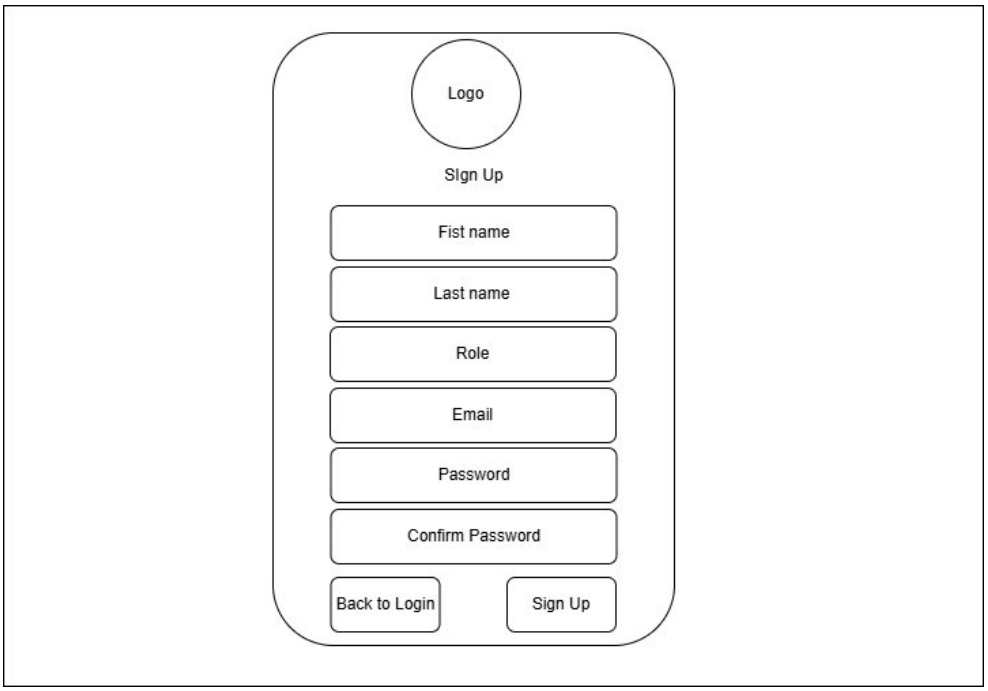


Figure 3.11 STMS Sign Up

3.5.2.2 STMS Sign Up Design

The sign-up function enables new users (excluding Admins) to register by providing their name, avatar, role, email, and password as shown in Figure 3.11. Once registered, they can log in to access their personalized dashboard and assigned projects.

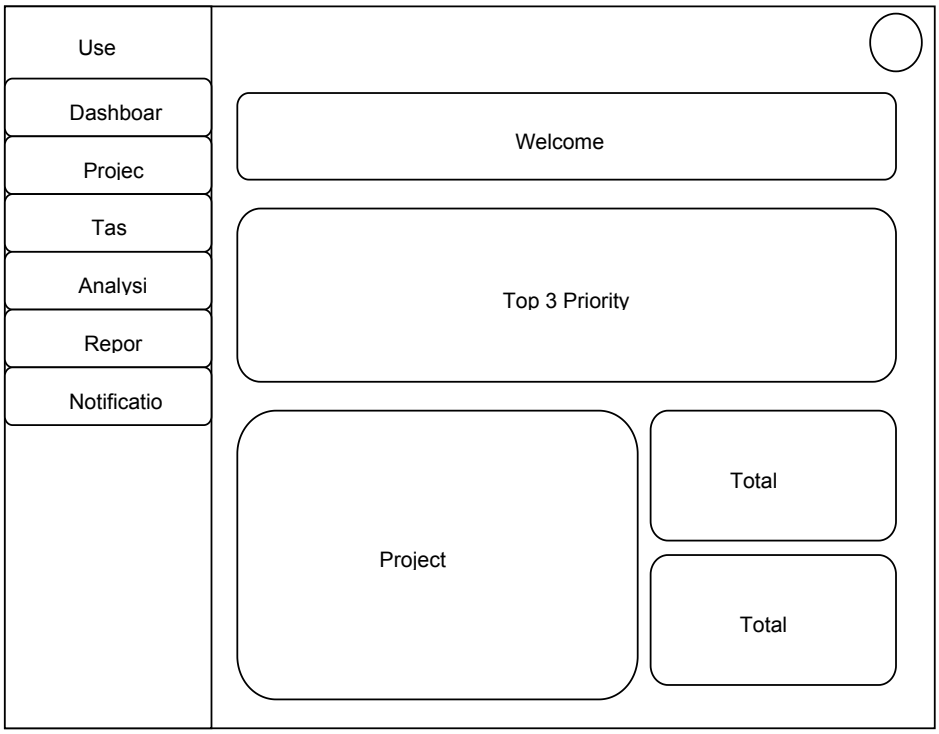


Figure 3.12 STMS Dashboard

3.5.2.3 STMS Dashboard Design

The Project Manager and Team Member dashboard displays a summary of current projects, top three tasks, and project progress bars as illustrate in Figure 3.12. This allows managers to monitor task distribution and ongoing work efficiently.

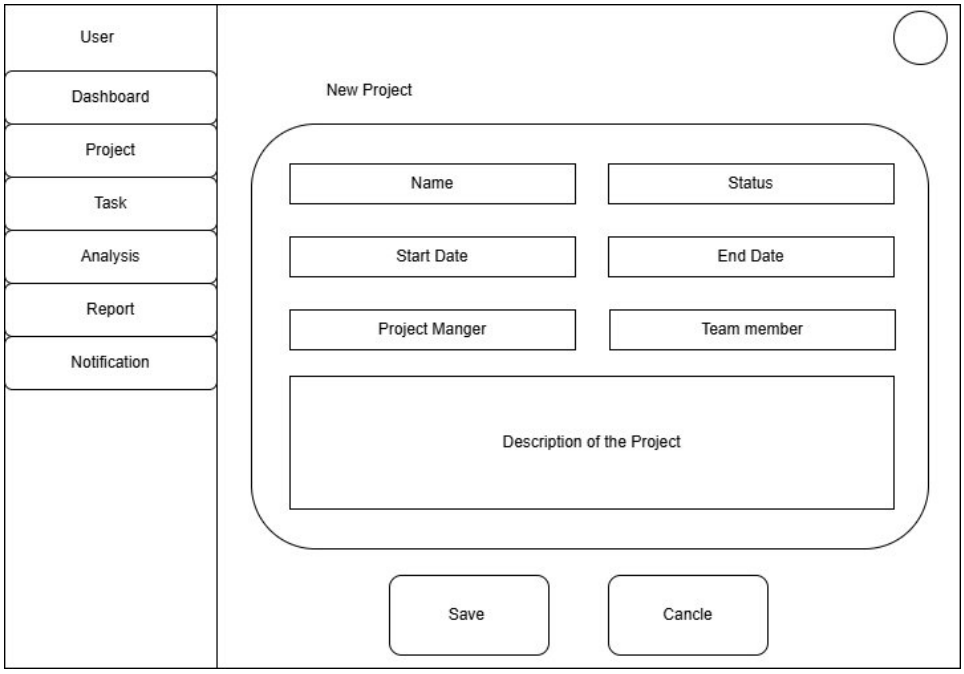


Figure 3.13 STMS Add New Project

3.5.2.4 STMS Add New Project Design

Project Managers can create new projects by entering a title, description, start and end dates, and assigning team members as illustrate in Figure 3.13. The project is then listed in the project list and made accessible to assigned users.

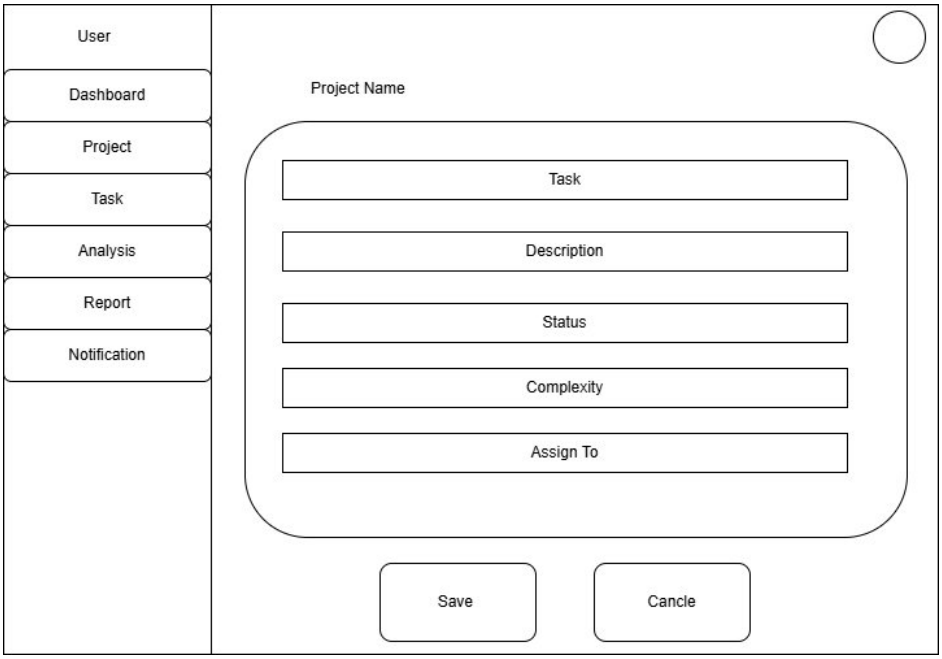


Figure 3.14 STMS Add New Task

3.5.2.5 STMS Add New Task Design

Tasks are added under specific projects by Project Managers as illustrate in Figure 3.14. Each task includes a title, description, complexity, status, and assigned user, ensuring proper delegation and tracking.

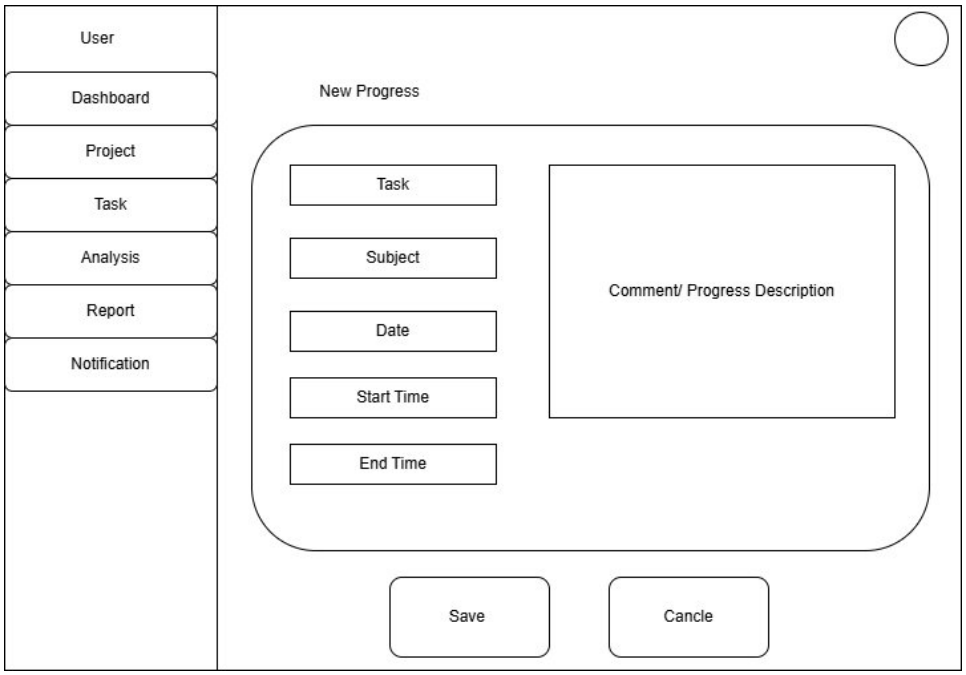


Figure 3.15 STMS Add New Activity

3.5.2.6 STMS Add New Activity Design

This feature allows Manager and Team Member to log the duration and progress of specific tasks by recording activity dates, times, and comments as illustrate in Figure 3.15. It supports time tracking and task progress monitoring.

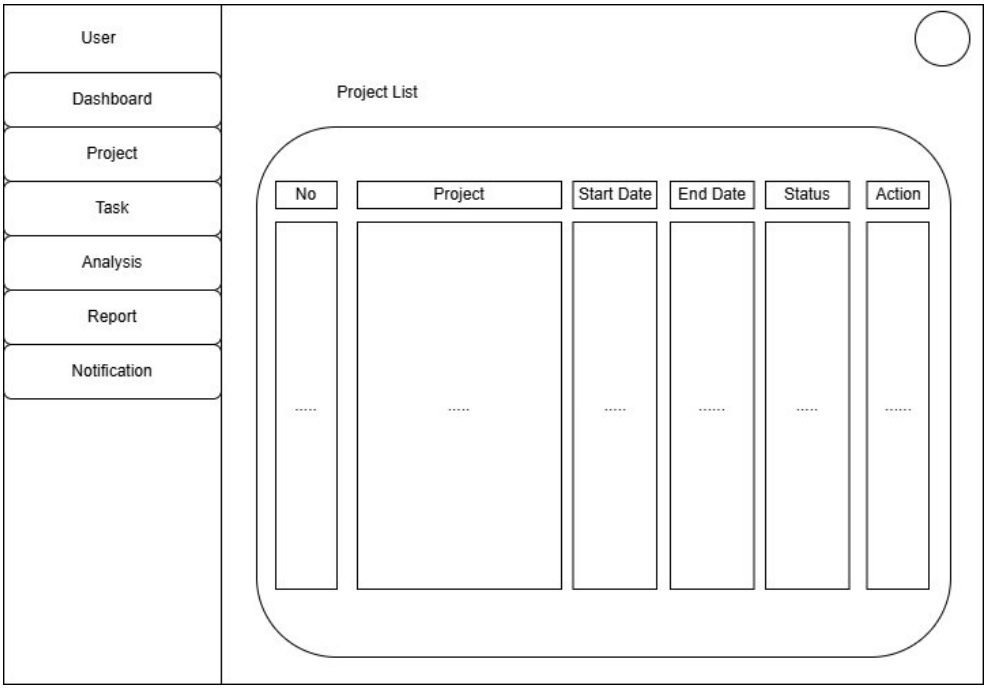


Figure 3.16 STMS Project List

3.5.2.7 STMS Project List Design

Project Manager and Team Member can view a list of all their projects, including details such as title, description, assigned users, and progress status as illustrate in Figure 3.16. Project Manager can also update or delete projects from this view.

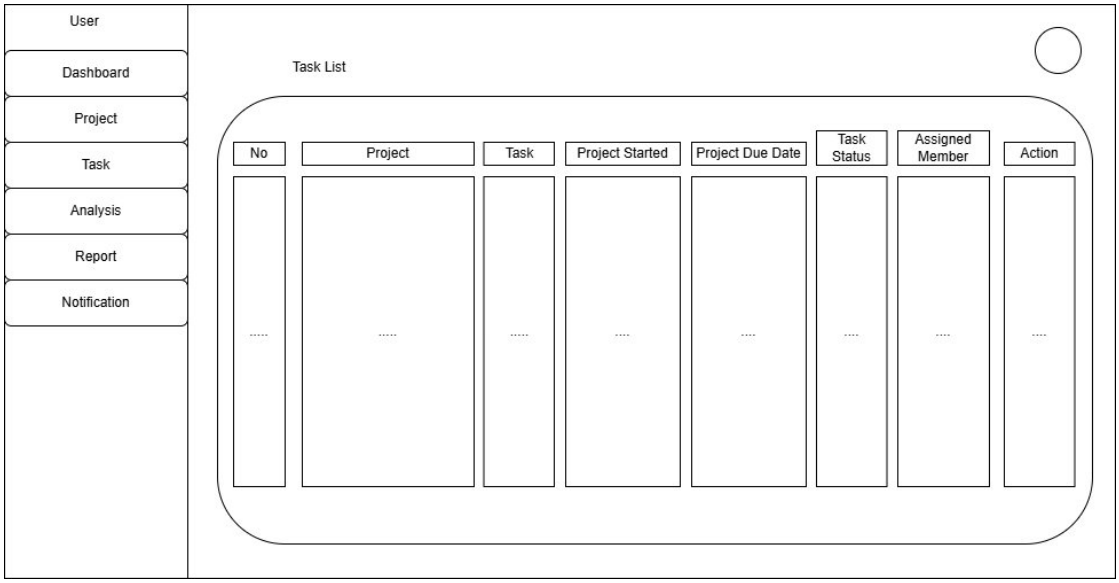


Figure 3.17 STMS Task List

3.5.2.8 STMS Task List Design

This function shows all tasks within the Manager's projects and Team Member, with status indicators and complexity levels as illustrate in Figure 3.17. Managers can edit or update task progress and track workload distribution.

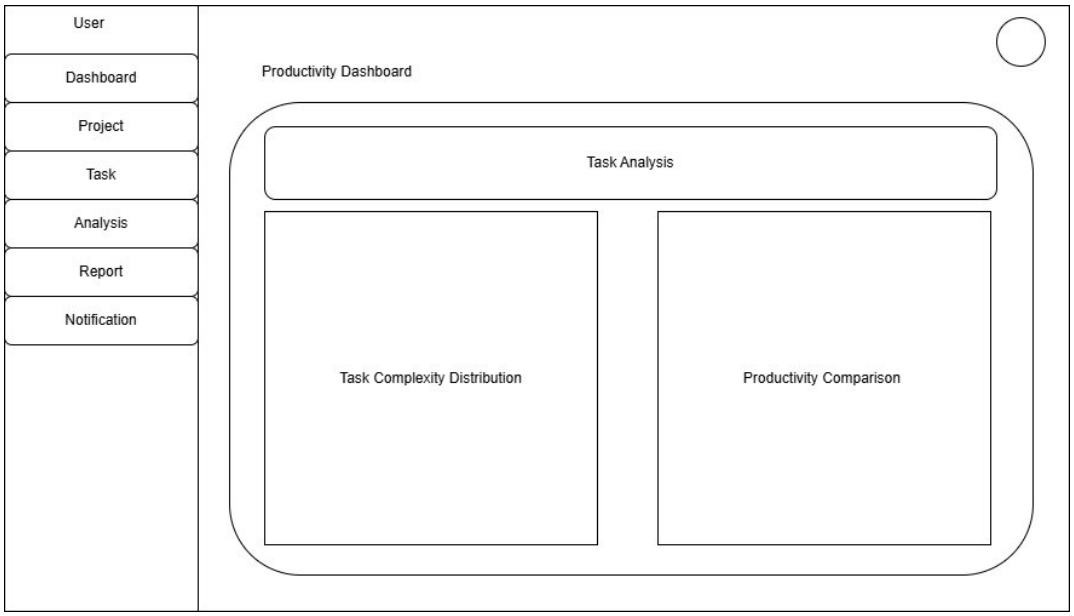


Figure 3.18 STMS User Task Analysis

3.5.2.9 STMS User Task Analysis Design

Project Managers and Tean Member can visualize task performance via graphs and charts showing task completion by complexity, as well as task duration and efficiency as illustrate in Figure 3.18. This helps in identifying team productivity levels.

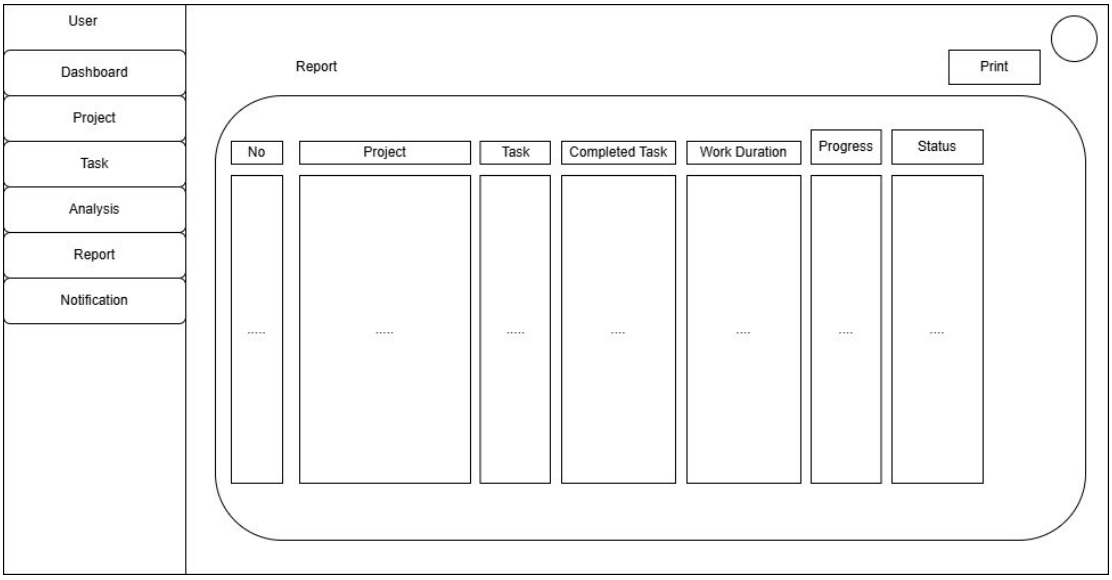


Figure 3.19 STMS Report

3.5.2.10 STMS Report Design

Project Managers can generate printable reports summarizing project details, task counts, assigned users, and task progress as illustrate in Figure 3.19. This aids in reporting and formal documentation.

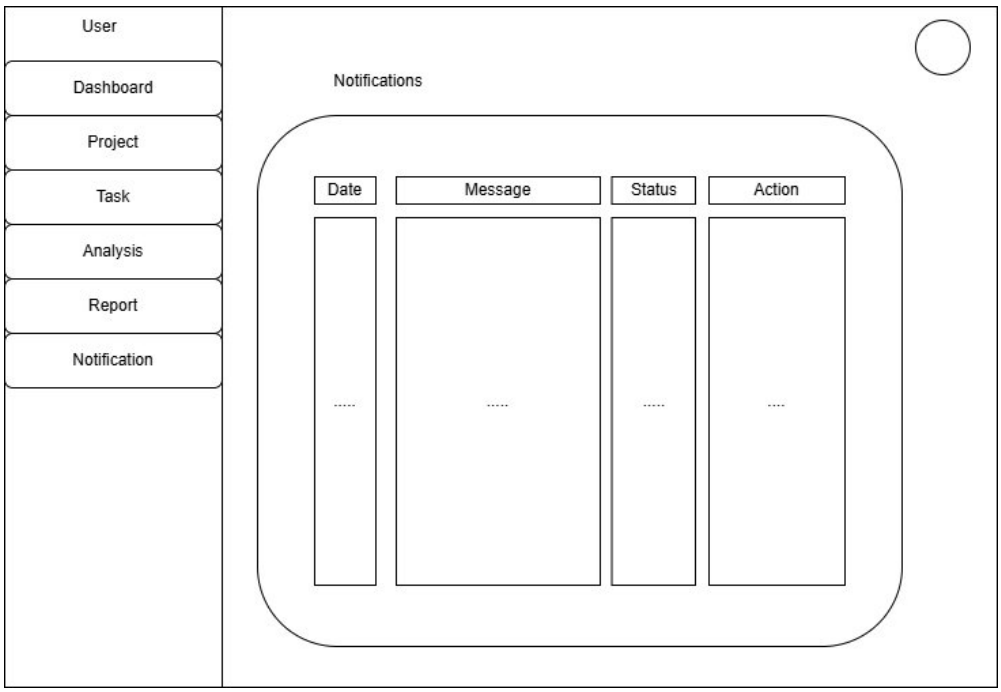


Figure 3.20 STMS Notification

3.5.2.11 STMS Notification Design

This function notifies Project Managers and Team Member when users are added to projects or when updates occur, ensuring real-time communication and system awareness. as illustrate in Figure 3.20.

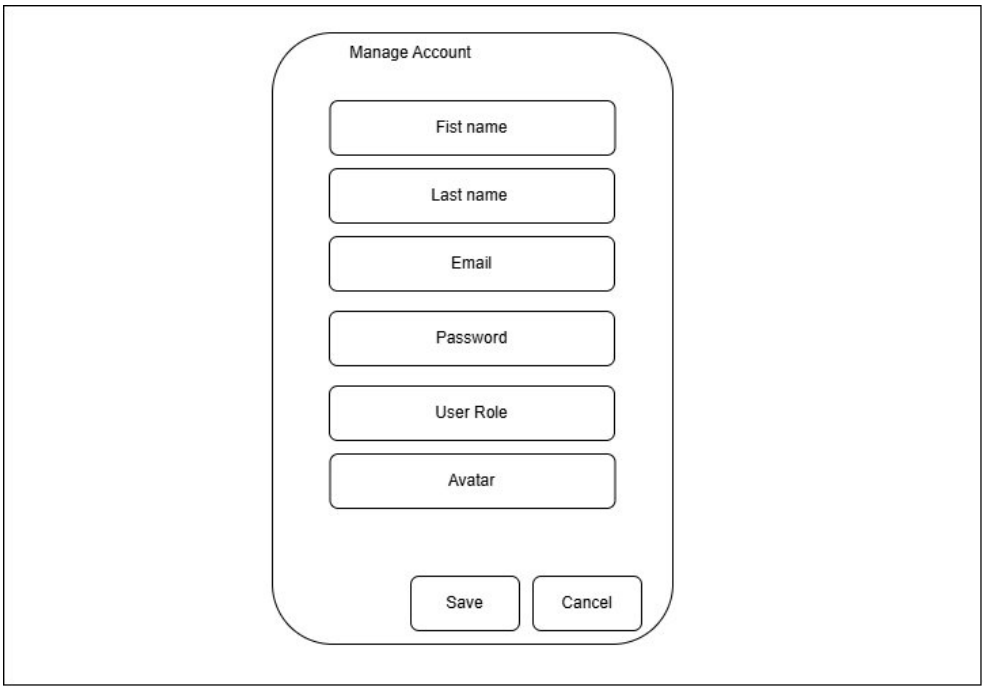


Figure 3.21 STMS Manage User

3.5.2.12 STMS Manage User Design

Users can update their own profile information, such as name, avatar, email, and password as illustrate in Figure 3.21. This promotes self-service and data accuracy.

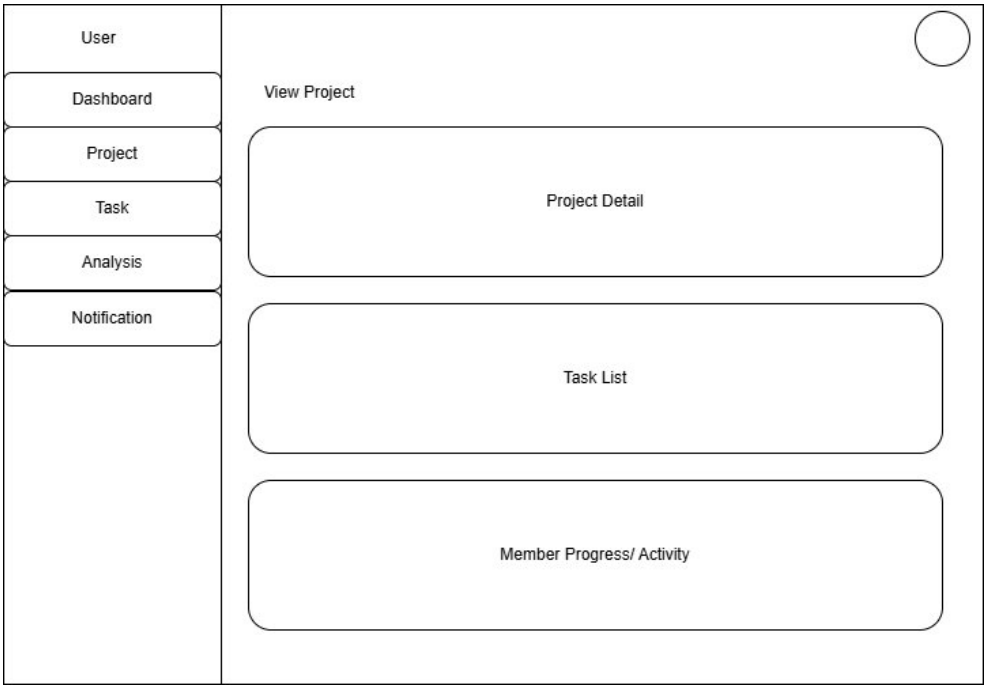


Figure 3.22 STMS Project Detail

3.5.2.13 STMS Project Detail Design

Team Members can view only the projects they are assigned to as illustrate in Figure 3.22. Each project entry shows details like title, dates, and progress, allowing members to focus on relevant tasks.

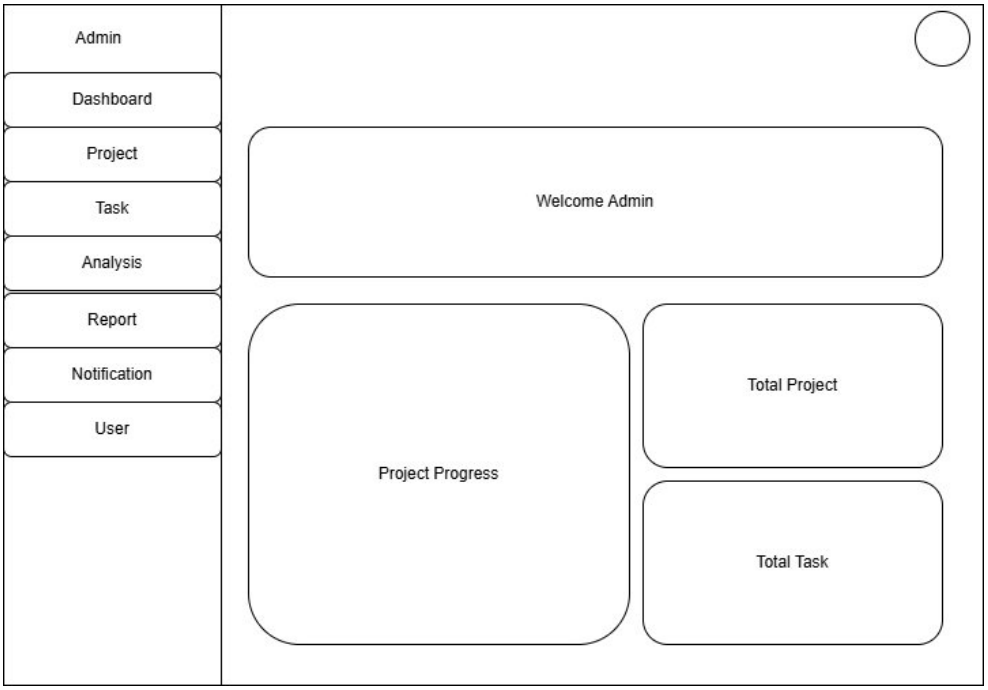


Figure 3.23 STMS Dashboard Admin

3.5.2.14 STMS Dashboard Admin Design

Admins are presented with a system-wide dashboard showing user count, project statistics, task statuses, and overall progress percentages as illustrate in Figure 3.23. This supports high-level monitoring.

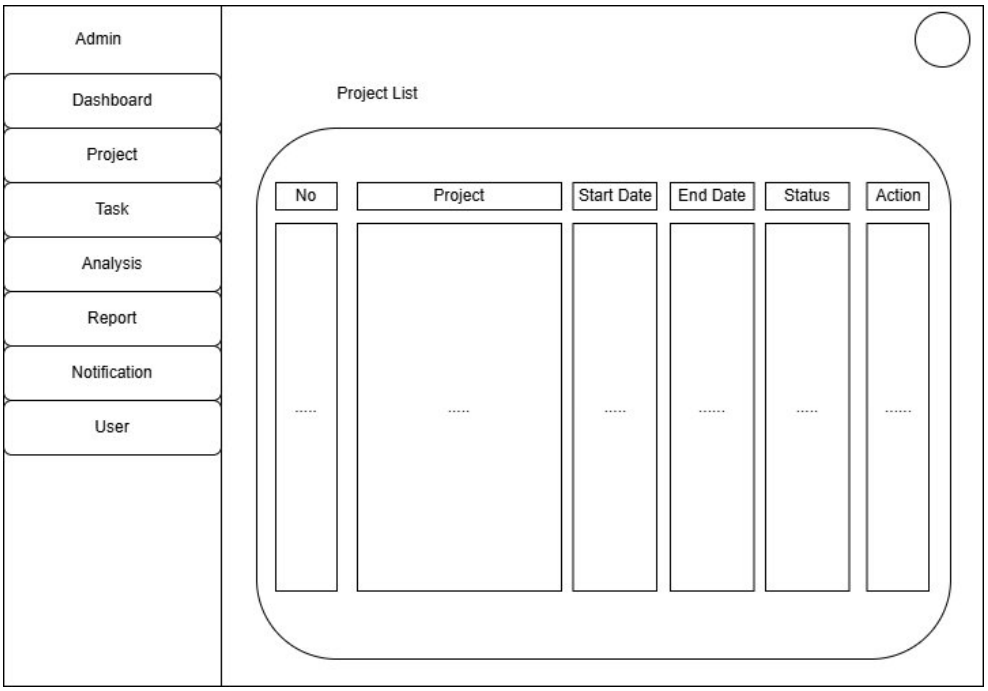


Figure 3.24 STMS Project List Admin

3.5.2.15 STMS Project List Admin Design

Admins can view and manage all system projects, regardless of owner as illustrate in Figure 3.24. They have the authority to delete or review project records for administrative oversight.

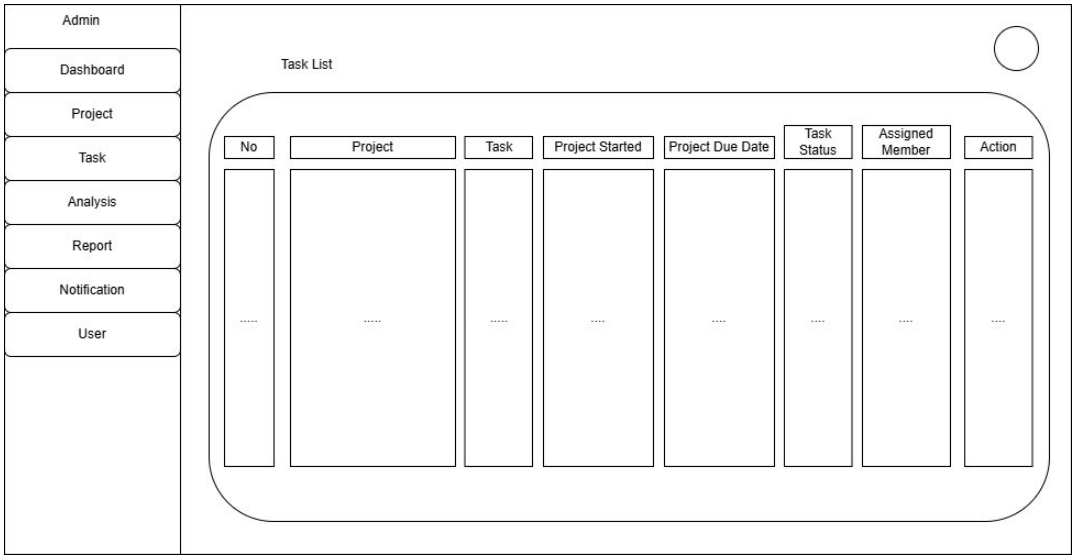


Figure 3.25 STMS Task List Admin

3.5.2.16 STMS Task List Admin Design

Admins can access all tasks across the system, view complexity levels, and delete or update records as needed to maintain system integrity as illustrate in Figure 3.25.

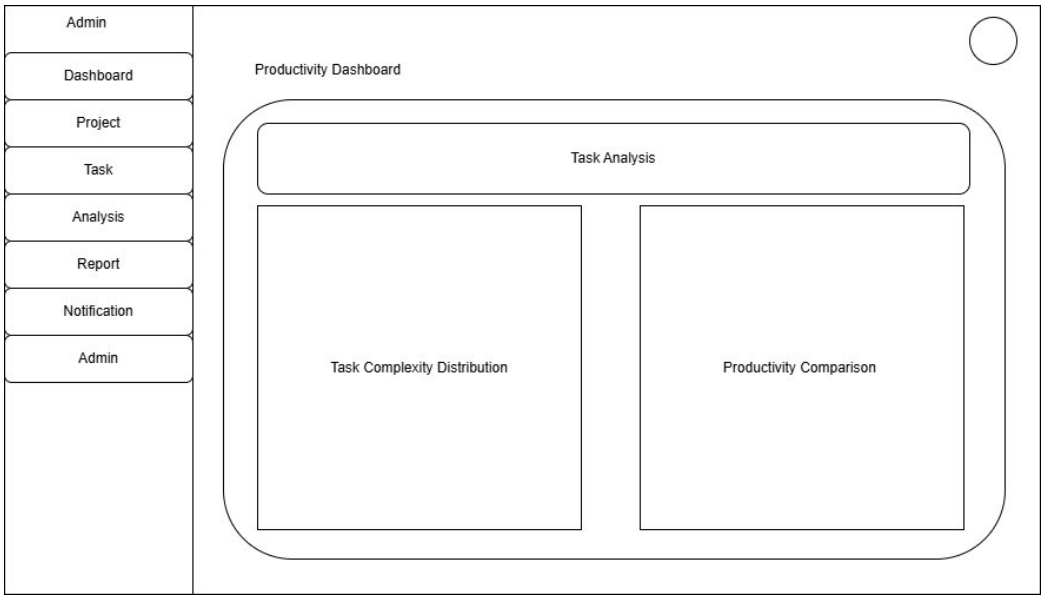


Figure 3.26 STMS User Task Analysis Admin

3.5.2.17 STMS User Task Analysis Design Admin Design

Admins are equipped with comprehensive analytics on user productivity, including comparative task complexity and efficiency graphs for evaluation and reporting as illustrate in Figure 3.26.

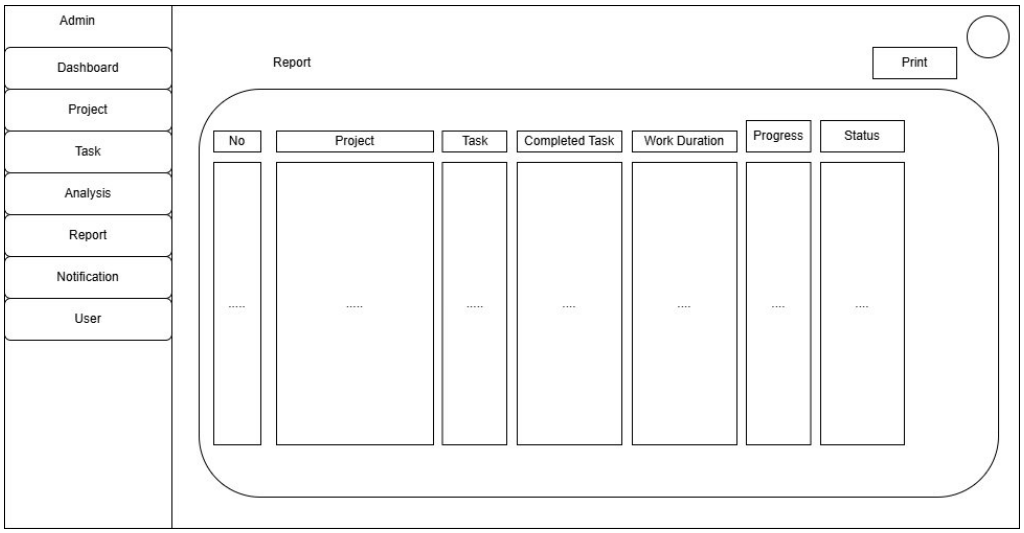


Figure 3.27 STMS Report Admin

3.5.2.18 STMS Report Admin Design

Admins can generate structured reports for all projects and tasks, including summary statistics and detailed performance metrics as illustrate in Figure 3.27. These reports support auditing and system evaluations.

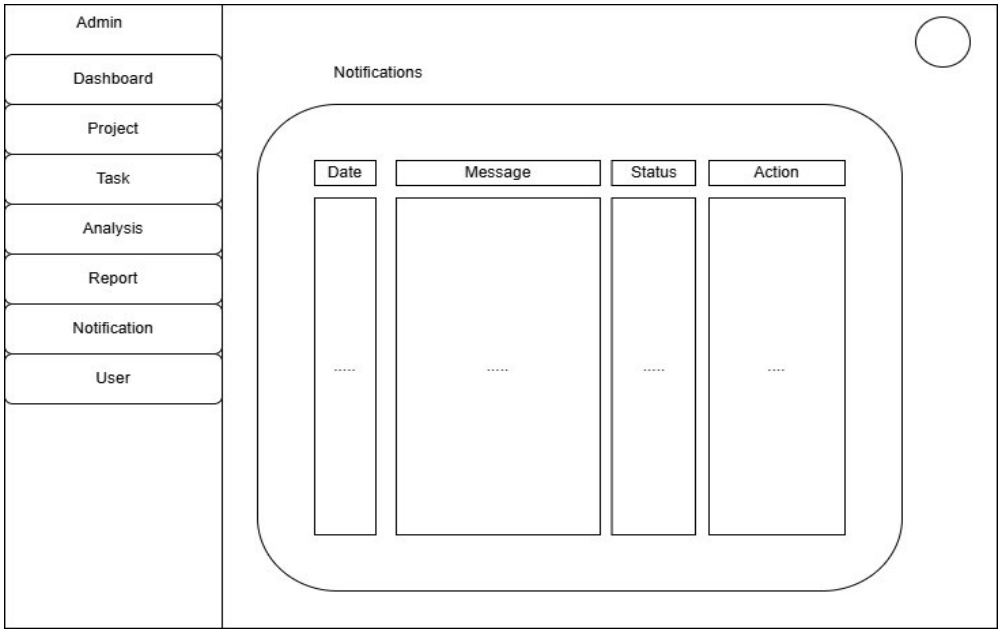


Figure 3.28 STMS Notification Admin

3.5.2.19 STMS Notification Admin Design

Admins receive system-wide notifications for any significant action such as project creation, or task updates, enabling full operational oversight as illustrate in Figure 3.28.

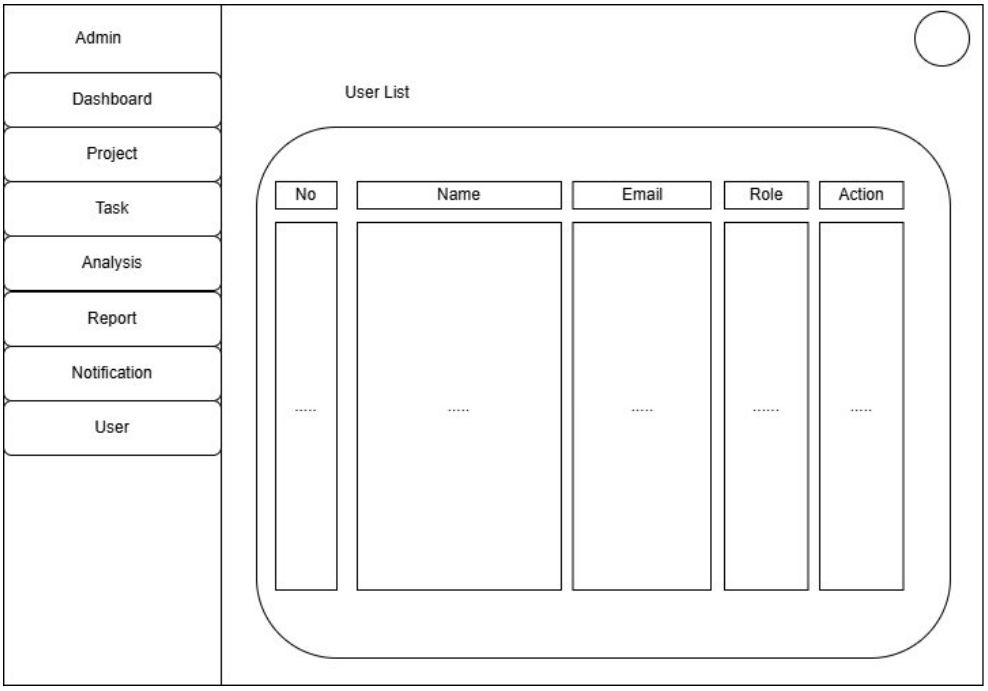


Figure 3.29 STMS User List Admin

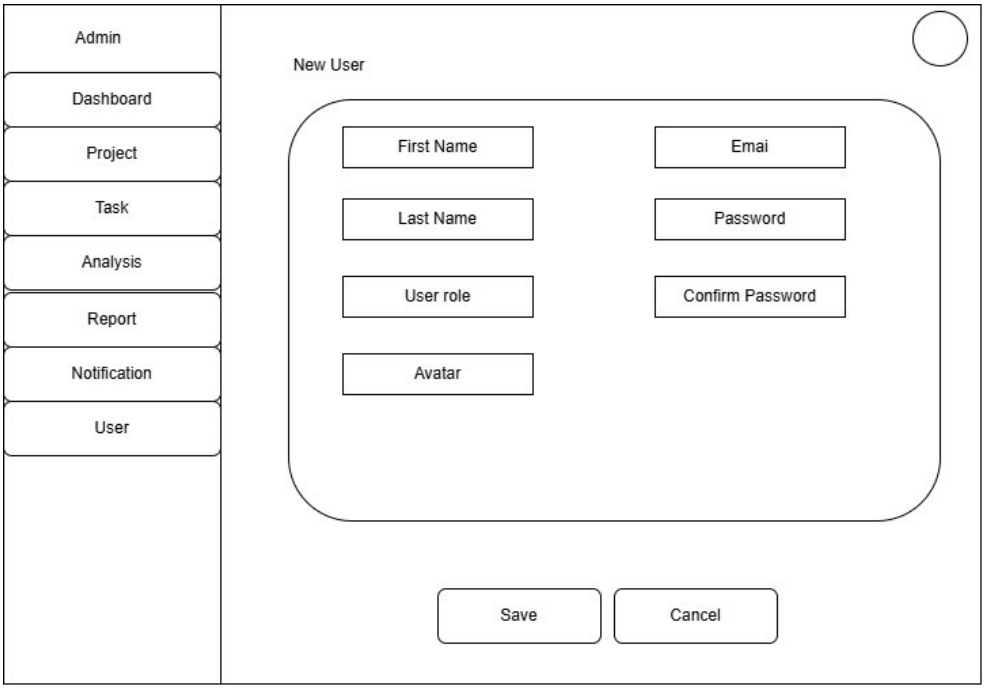


Figure 3.30 STMS Add User Admin

3.5.2.19 STMS User Management Admin Design

Admins can view, edit, and delete any user account as illustrate in Figure 3.29 and Figure 3.30. This includes managing access rights, updating user roles, and removing inactive accounts to ensure data integrity.

3.5.3 Coding Phase for Smart Task Management System for UNIMAS Student

The Coding Phase is a crucial stage in developing the Smart Task Management System (STMS) for UNIMAS students, where the design and requirements are transformed into a fully functional application. During this phase, developers focus on building the system features based on predefined user stories. Key features include user registration and secure login using email, task management to allow students to create, assign, and track assignments or goals, and team management for creating teams and assigning roles. The system will also support task prioritization and deadline tracking, notifications for important events, profile management, and a personalized dashboard to display all key information.

The development involves both front-end and back-end implementation. The front-end, created using HTML, CSS, JavaScript, and frameworks like React or Angular, ensures an intuitive and responsive interface. The back-end, built with technologies such as PHP, Node.js, or Python, manages the core application logic and database operations. A relational database like MySQL will be used to store user and task data.

Version control is handled through Git, where developers work on separate branches for each feature to allow collaboration without conflicts. Code changes are reviewed through pull requests before being merged into the main branch. A task prioritization algorithm will be implemented to help students focus on high-priority tasks based on deadlines and complexity, and this system will improve over time using collected task data.

Notifications will be integrated using a mailing system to provide real-time alerts via email, ensuring that students are promptly informed about new tasks, approaching deadlines, or team updates. Security is also emphasized, with user authentication and data protection supported by MD5 hashing for password security. Although MD5 offers basic encryption, it will be implemented with additional measures to help safeguard students' personal information during storage and access.

To ensure accessibility, the system will be designed with responsive layouts using CSS frameworks like Bootstrap or Material UI, making it usable across desktops, tablets, and

smartphones. Developers will implement continuous integration tools such as GitHub Actions to automatically test the code on every update. Unit tests will be created for critical functions like authentication and task creation to maintain software quality.

The coding process will follow Agile methodology with 1–2 week sprint cycles. Each sprint focuses on delivering specific features, and completed work is reviewed with stakeholders such as product owners or university representatives to gather feedback and ensure the project stays aligned with user needs.

3.5.4 Testing Phase for Smart Task Management System for UNIMAS Student

The Testing Phase ensures that the Smart Task Management System (STMS) functions correctly, meets all requirements, and is free from major defects. Since the system is developed using Agile methodology, testing is continuous and carried out throughout the development cycle. This phase involves verifying that the application is reliable, user-friendly, and performs well under real-world conditions.

Unit testing is the first step, focusing on individual components such as user registration, login, task creation, team management, and notification triggers. These tests help confirm that each part of the system works properly in isolation and are typically automated to catch issues early. Integration testing then checks whether different parts of the system work together seamlessly. It ensures that the front-end communicates correctly with the back-end, data flows smoothly between features, and APIs respond accurately. For example, when a new task is created, it should appear instantly in the task list and database.

Functional testing validates that the system performs all required tasks as expected. It checks user flows such as registration, task assignment, team collaboration, and profile updates to confirm they work without issues. Notifications are also tested to ensure they are sent at the right times. Following this, User Acceptance Testing (UAT) is carried out by actual UNIMAS students. They interact with the system to ensure it meets their needs, and their feedback is used to improve usability, such as suggesting enhancements to notification settings or task prioritization features.

Performance testing is also conducted to evaluate the system's speed and stability. Load testing simulates many users accessing the platform at once, while stress testing pushes the system to

its limits to check how it behaves under pressure. Response time testing measures how quickly the system performs key actions, like updating tasks or sending notifications.

Finally, continuous testing is implemented using automated tools that run tests with every code update. This helps maintain the system's quality by quickly identifying bugs or performance issues as new features are added. Overall, the Testing Phase ensures that STMS remains stable, secure, and efficient for all UNIMAS students.

3.5.5 Reviewing and Refining Phase for Smart Task Management System for UNIMAS Student

The Reviewing and Refining Phase is a crucial part of the Agile development cycle, taking place after testing to evaluate the system's performance and make necessary improvements based on feedback and real-world usage. For the Smart Task Management System (STMS) designed for UNIMAS students, this phase ensures that the system aligns with student needs and delivers a smooth and efficient user experience. At the end of each sprint, a Sprint Review is held where the development team showcases new features like task management and notifications to stakeholders, including students and university representatives. Feedback is gathered to identify usability issues or areas that need enhancement, such as refining task prioritization or improving navigation.

Based on this feedback, the team focuses on improving usability, fixing bugs, enhancing features, and optimizing performance. Adjustments are made to the user interface to make it more intuitive, and functionality is refined to ensure all features work correctly. If performance issues are identified, developers improve response time and system speed, especially under heavy usage. Security aspects, such as data protection and secure authentication, are also reviewed and strengthened.

During this phase, critical bugs are fixed first, especially those that affect essential functions like task assignment. Developers also optimize the code to ensure the system scales well as more users access it. UI/UX improvements are made to enhance design and make interactions more comfortable and enjoyable for students. The system is refined to work smoothly across various devices, ensuring accessibility.

All changes are documented thoroughly. User manuals are updated to help students understand new features, and technical documentation is prepared for developers and administrators. Knowledge transfer sessions are conducted to ensure future maintainers can easily manage and

update the system. Performance tuning is done to handle peak usage efficiently, including optimizing databases, servers, and implementing caching strategies.

Finally, a last round of validation is conducted to ensure all system functions are working as expected. This includes core tasks like creating, assigning, and viewing tasks. Final user testing with selected students confirms that changes made are effective and the system is ready for launch. The development team then plans for the system's release, prepares deployment strategies, and sets up support channels to address post-launch issues.

3.6 Summary Chapter 3

Chapter 3 outlines the Research Methodology used in developing the Smart Task Management System (STMS) for UNIMAS students. The methodology follows an Agile approach, emphasizing iterative development, flexibility, and continuous collaboration with stakeholders. This ensures that the system is regularly improved based on user feedback and evolving requirements. The Agile framework supports the development of features through sprints, where each cycle includes planning, coding, testing, reviewing, and refining, allowing for incremental progress.

The methodology begins with requirements gathering, focusing on understanding students' needs for task management and collaboration. These insights drive the design and development of the system, which includes features such as task creation, team management, and notifications. The system's interface is designed to be user-friendly, and the development follows iterative processes to improve functionality.

The testing phase ensures the system's reliability, security, and user-friendliness through unit, integration, and user acceptance testing (UAT). After identifying and addressing any issues, the system enters the reviewing and refining phase, where feedback from users is integrated to optimize performance, fix bugs, and enhance the interface. Finally, the system undergoes final validation before deployment, with proper documentation and a release plan in place, ensuring the system is ready for UNIMAS students.

By utilizing this Agile methodology, the STMS is developed to meet the needs of students, ensuring it is both functional and user-friendly, with ongoing improvements based on feedback and real-world use.

Chapter 4: Implementation and Testing

4.1 Introduction

This chapter presents the implementation and testing of the Smart Task Management System developed for UNIMAS students. It outlines the tools, technologies, and methods used during the system development process, including both front-end and back-end components. The chapter also explains how the system was deployed and the key features that were implemented.

In addition, this chapter describes the testing procedures carried out to ensure the functionality, reliability, and usability of the system. Both functional and non-functional testing were conducted to evaluate how well the system meets its intended objectives. The results of these tests are discussed to demonstrate the system's performance and effectiveness in supporting task management among students.

4.2 Installation and System Configuration

This section outlines the steps taken to install and configure the development environment for the Smart Task Management System (STMS). The system was built using a combination of web development technologies including PHP, MySQL, HTML, CSS, and JavaScript. It also utilizes external tools such as XAMPP for setting up a local server environment, Visual Studio Code as the code editor, and PHPMailer for handling email functionality within the system.

The development and testing of the system were conducted on a computer with sufficient hardware capabilities to support web application development and local server operations. This includes adequate processing power, memory, and storage to run the required software and tools efficiently.

4.2.1 System Requirements

To ensure the Smart Task Management System (STMS) runs smoothly and efficiently, certain hardware and software requirements must be met. On the hardware side, the system requires a computer with at least an Intel Core i3 processor or higher, a minimum of 4 GB of RAM, and at least 1 GB of free disk space to store project files and system data.

In terms of software, the system is compatible with Windows 10 or later, macOS, or Linux operating systems. For web server functionality, Apache is required, which is provided through the XAMPP package. The backend is developed using PHP version 7.4 or later, while MySQL is used as the relational database management system. The codebase is managed and edited using Visual Studio Code, a lightweight and efficient source-code editor.

PHPMailer is used to manage email-related functionalities such as sending notifications. Additionally, a modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge is recommended for accessing and testing the system's web interface.

4.2.2 Software Installation Procedures

This section describes the software tools used and the procedures followed to install them in preparation for developing the Smart Task Management System (STMS). Each software plays an important role in enabling different functionalities of the system, from local server hosting to coding, database management, and email communication.

4.2.2.1 XAMPP

XAMPP is an open-source web server package that includes Apache, MySQL, and PHP, which are essential for running the STMS locally. To install XAMPP, the installer was downloaded from the Apache Friends official website. During installation, Apache, MySQL, and phpMyAdmin were selected as components. Once installed, the XAMPP Control Panel was used to start the Apache and MySQL services, which enabled the local web server and database environment. XAMPP played a critical role in hosting the STMS project locally and managing the backend and database interactions.

4.2.2.2 Visual Studio Code

Visual Studio Code, a source-code editor developed by Microsoft, was used as the primary coding environment for the STMS project. The installer was downloaded from the official Visual Studio Code website and the setup process was completed by following the on-screen instructions. Additional extensions such as PHP and MySQL support were installed from the built-in extension marketplace. This tool provided an efficient and lightweight environment to write, edit, and manage HTML, CSS, JavaScript, and PHP code, and included useful features such as syntax highlighting, integrated Git control, and debugging tools.

4.2.2.3 PHP

PHP, a server-side scripting language, was used to implement the backend functionality of the STMS. Since PHP is included as part of the XAMPP package, no additional installation was required. The version bundled with XAMPP was compatible with the project requirements and supported server-side scripting tasks such as user authentication, task creation, and form handling.

4.2.2.4 MySQL

MySQL, a relational database management system, was used to store and manage all project-related data. Like PHP, MySQL was already included in the XAMPP package and was accessible through phpMyAdmin. This eliminated the need for separate installation and allowed for easy creation and management of databases required by the STMS, including tables for users, tasks, and notifications.

4.2.2.5 PHPMailer

PHPMailer is a PHP library that enables the sending of email through SMTP servers. To integrate PHPMailer into the STMS, the library was downloaded from its official GitHub repository and included in the project directory. The necessary configuration was performed in the PHP scripts to set up SMTP settings for email delivery. PHPMailer was used to send system-generated emails such as registration confirmations and task assignment notifications**.**

4.2.2.6 Web Browser

Modern web browsers such as Google Chrome, Mozilla Firefox, or Microsoft Edge were used to test and interact with the Smart Task Management System during development. These browsers provided the interface through which the system was accessed and tested. Additionally, built-in developer tools within the browsers, such as the Inspect Element and Console features, were used to debug layout issues, monitor network activity, and troubleshoot script errors.

4.3 Function of STMS

This section describes the key functions and features available in the Smart Task Management System (STMS), detailing the roles and capabilities assigned to different types of users including project managers, team members, and administrators.

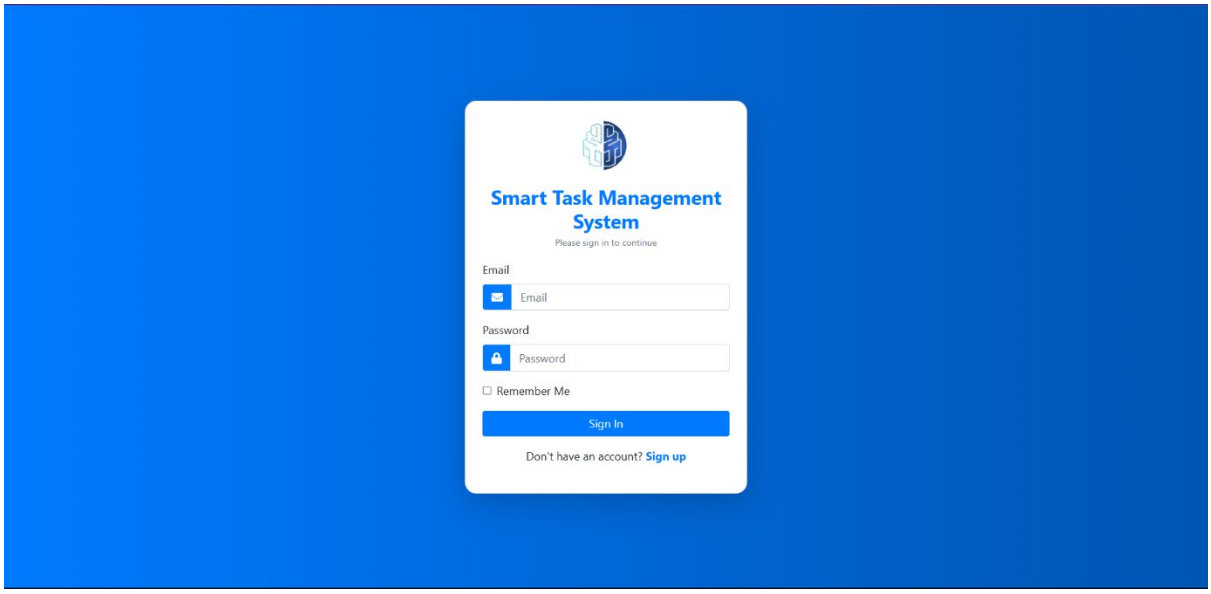


Figure 4.1 Login

4.3.1 Login (Project Manager / Team Members / Admin)

Figure 4.1 showing the login feature allows users to securely access the system by providing their registered email and password. Different user roles (project manager, team member, admin) are authenticated to access relevant modules within the system.

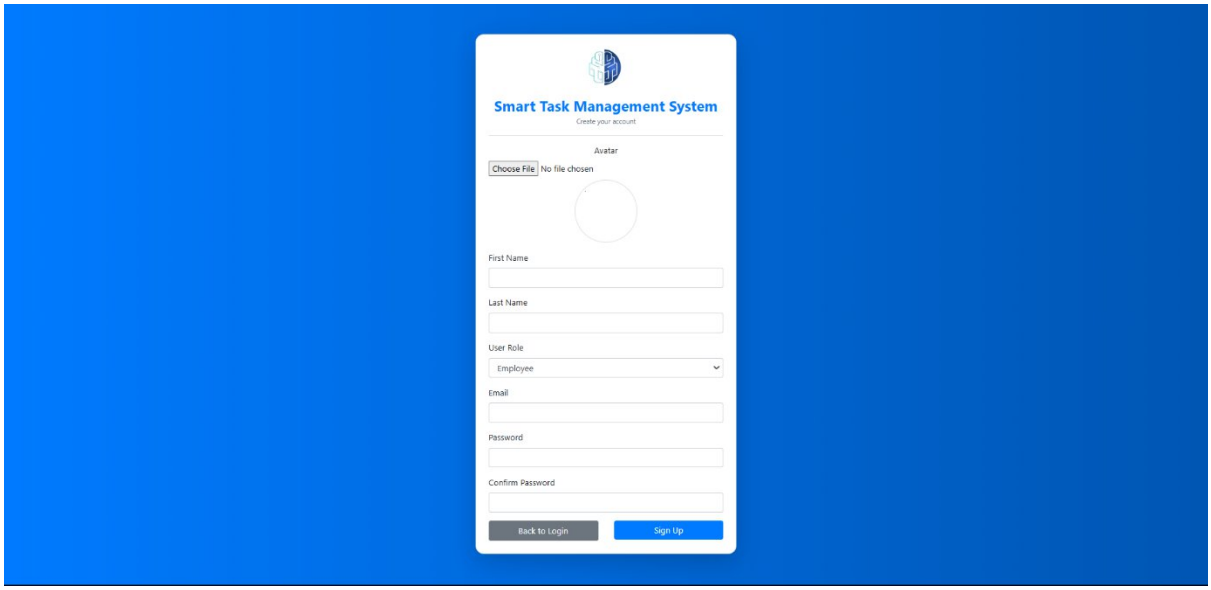


Figure 4.2 Sign Up

4.3.2 Sign Up (Project Manager / Team Member)

New users can register by signing up with their details, including choosing an avatar, first name, last name, user role (either project manager or team member), email, password, and confirming the password. This process, illustrated in Figure 4.2, enables the creation of personalized user accounts.

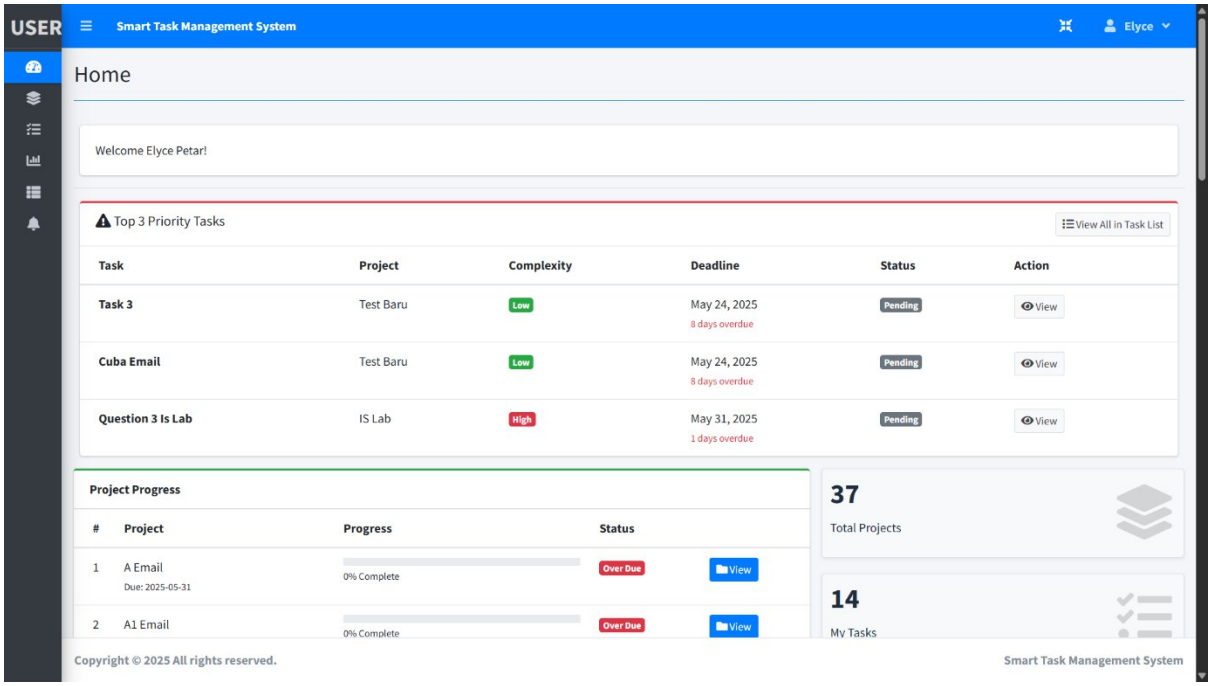


Figure 4.3 Dashboard

4.3.3 Dashboard (Project Manager / Team Member)

The dashboard in Figure 4.3 provides users with a summary view of their tasks and projects. It highlights the top three task priorities, overall project progress, total number of projects, and total tasks assigned, helping users quickly gauge their workload and status. The top three tasks are selected based on their priority level (e.g., High, Medium, Low) and approaching deadlines. Tasks marked as "High Priority" and those with the nearest due dates are given precedence. This ensures that users are immediately alerted to the most urgent and important tasks that require prompt attention, improving time management and task efficiency.

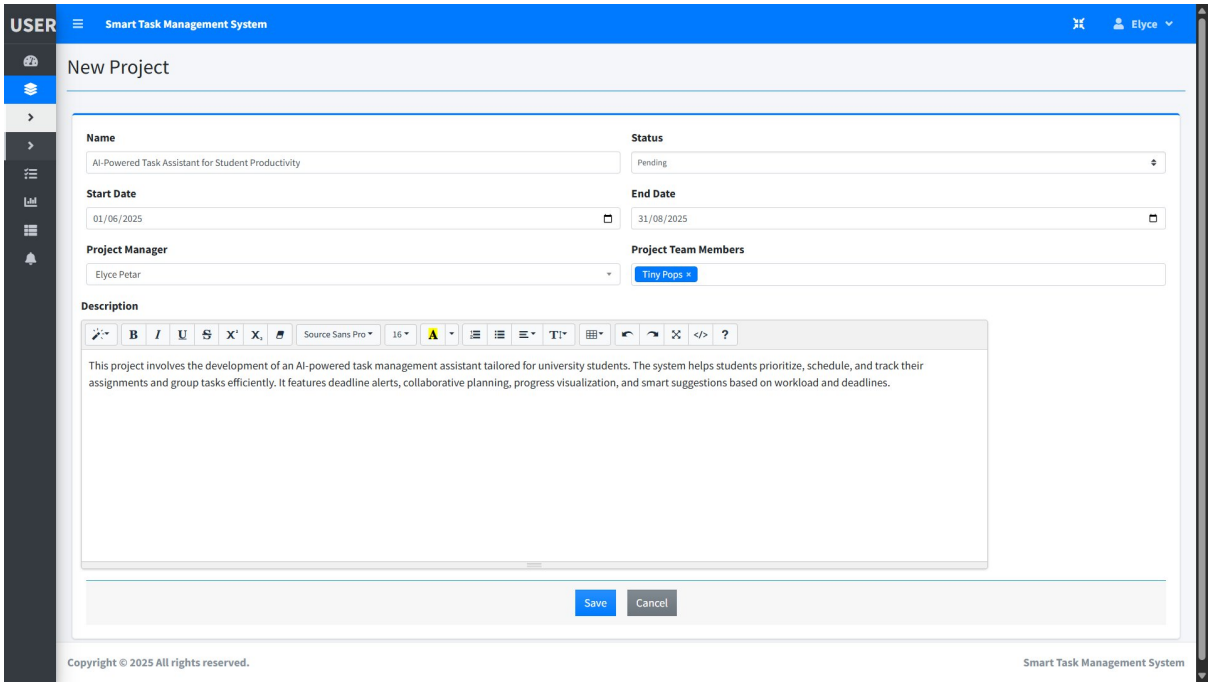


Figure 4.4 Add New Project

4.3.4 Add New Project (Project Manager)

Project managers can create new projects by entering essential information such as the project title, status, start and end dates, assigned project manager, team members, and a detailed description like Figure 4.4. This function initiates project tracking and management within the system.

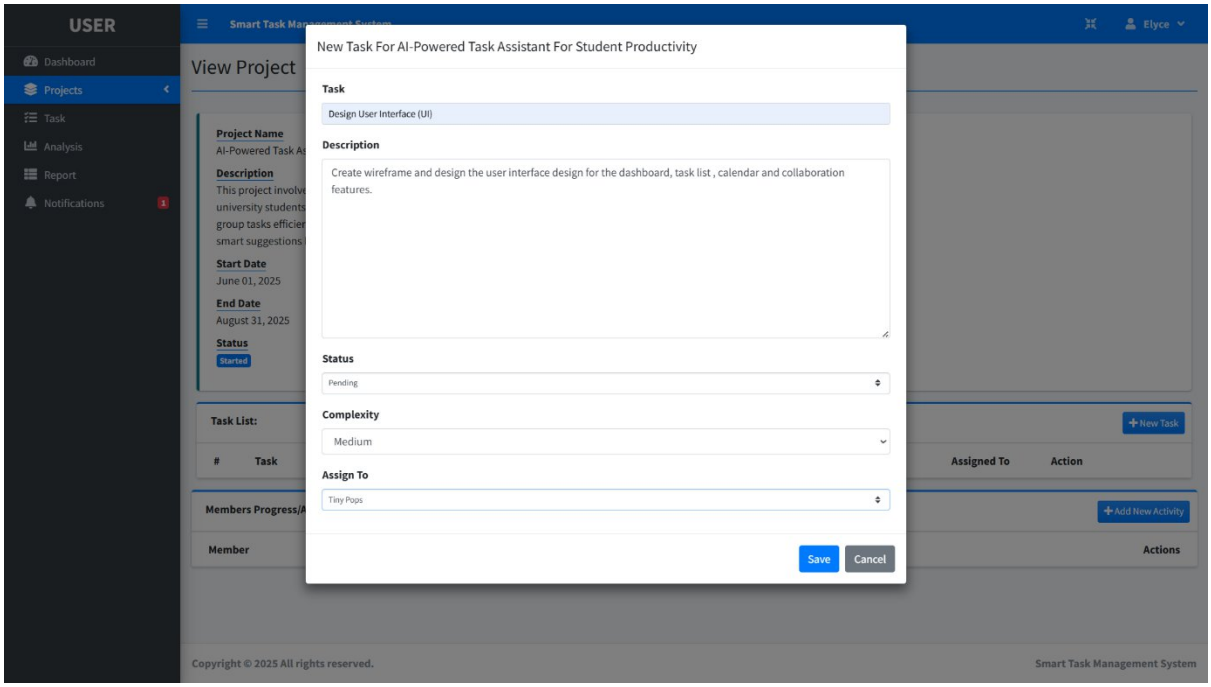


Figure 4.5 Add New Task

4.3.5 Adding New Task (Project Manager)

Within a project, project managers can add tasks by specifying the task title, description, status, complexity level, and assigning it to a team member. This process, shown in Figure 4.5, allows clear delegation and detailed tracking of individual responsibilities.

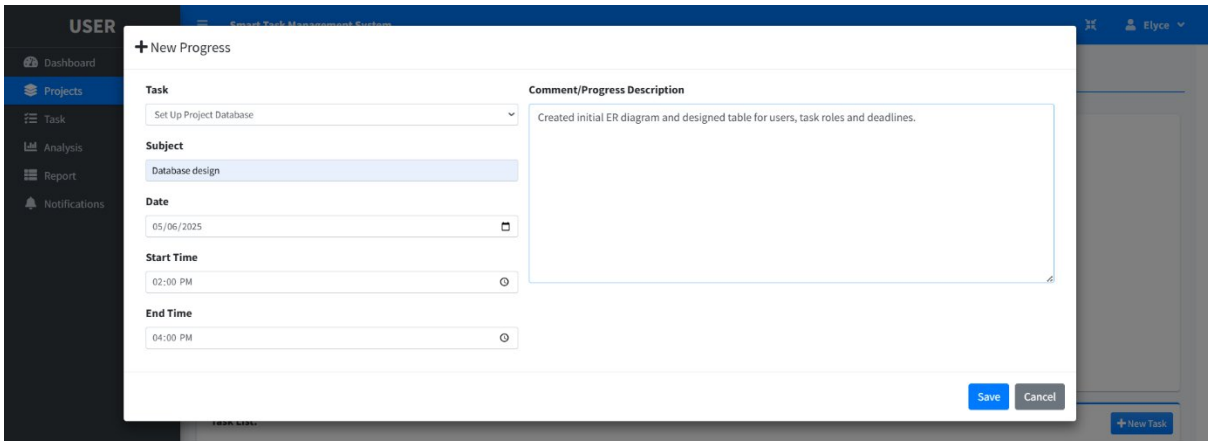


Figure 4.6 Add New Activity / Progress

4.3.6 Adding New Activity (Project Manager / Team Member)

Figure 4.6 shows that project managers can log activities or productivity details for tasks, including task title, subject, date, start time, end time, and progress comments like. This helps in monitoring task progress and recording work efforts.

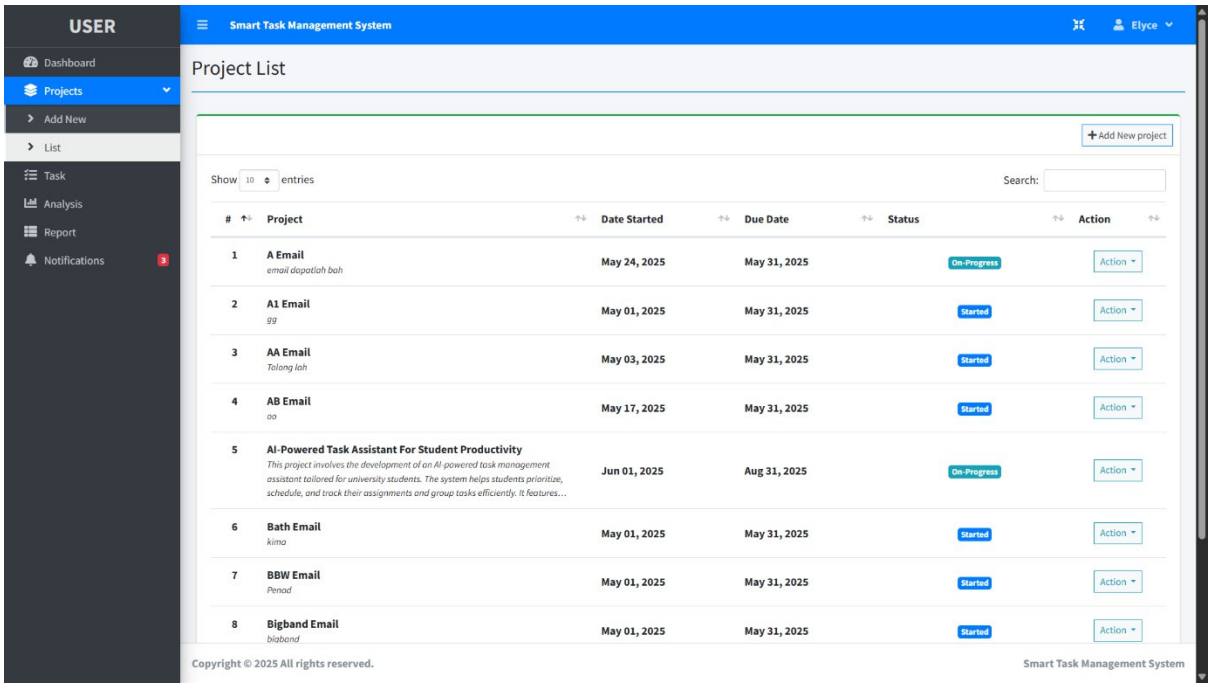


Figure 4.7 Project List

4.3.7 Project List (Project Manager)

Based on Figure 4.7, project managers have access to a comprehensive list of projects they oversee, displaying key details like project title, start date, due date, status, and available actions such as viewing, editing, or deleting a project. This function facilitates project organization and updates.

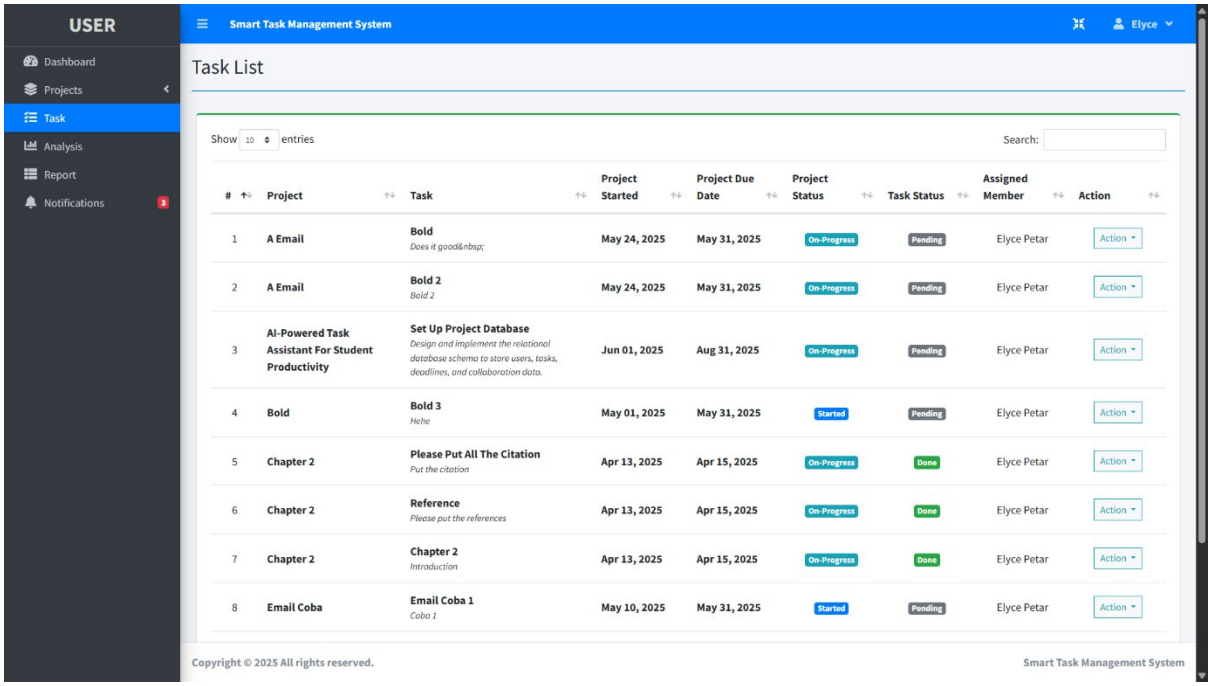


Figure 4.8 Task List

4.3.8 Task List (Project Manager / Team Member)

Both project managers and team members can view a list of tasks associated with projects as shown in Figure 4.8. The list includes project title, task name, project start and due dates, project status, assigned member, and actions such as adding productivity notes or editing tasks.

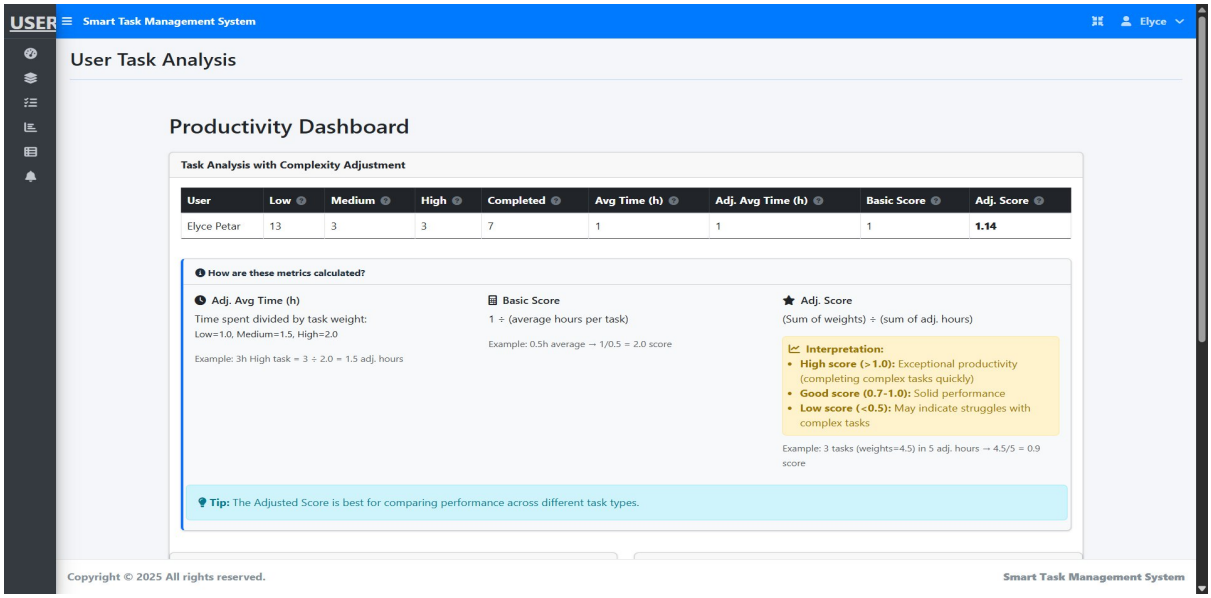


Figure 4.9 User Task Analysis 1

4.3.9 User Task Analysis (Project Manager / Team Member)

Figure 4.9 shows that the feature offers individual users a productivity dashboard analyzing their completed tasks based on complexity and time taken.

- **Complexity Weights**

  Each task is assigned a weight based on its complexity:

Complexity Level	Weight
Low	1.0
Medium	1.5
High	2.0

  These weights are used in all subsequent calculations.

- **Adjusted Average Time (Adj. Avg Time)**

  This metric calculates the time spent per task, adjusted for its complexity.

  Formula:

  Adj. Avg Time (h) = Time Spent ÷ Task Weight

- **Basic Score**

The Basic Score measures how efficiently a user completes tasks, regardless of complexity.

Formula:

Basic Score = 1 ÷ (Average Time per Task)

- **Adjusted Score**

  The Adjusted Score combines both complexity and time into a single performance metric.

  Formula:

  Adjusted Score = Total Task Weights ÷ Total Adjusted Hours

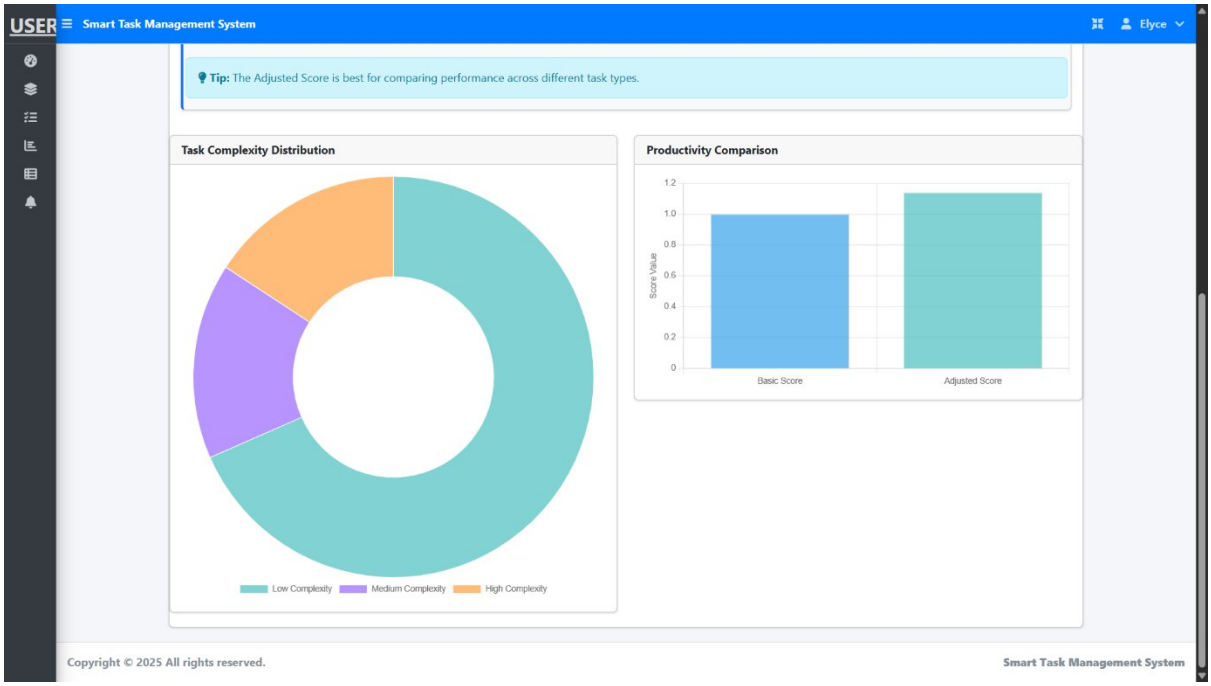


Figure 4.10 User Task Analysis 2

Visual data such as pie charts for task complexity distribution and bar graphs for productivity comparisons are displayed, as illustrated in Figure 4.10, supporting self-assessment and improvement.

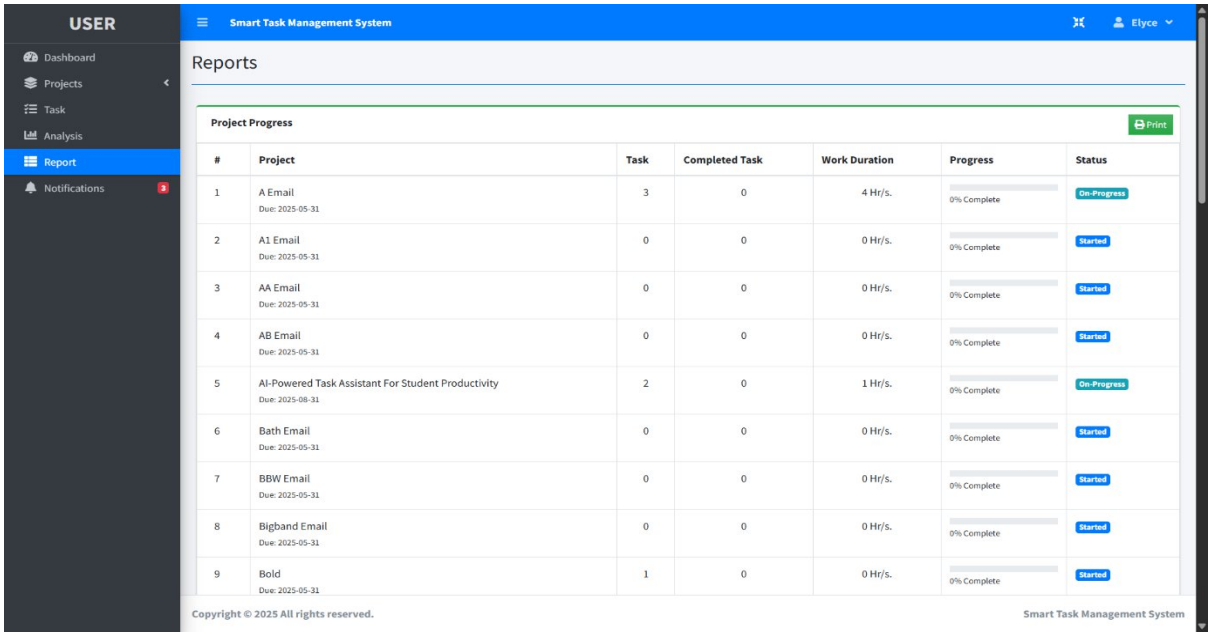


Figure 4.11 Report

4.3.10 Report (Project Manager)

Project managers can generate detailed reports listing their projects, including total tasks, completed tasks, work duration, project progress, and status as shown in Figure 4.11. These reports can be printed to provide documented evidence of project performance.

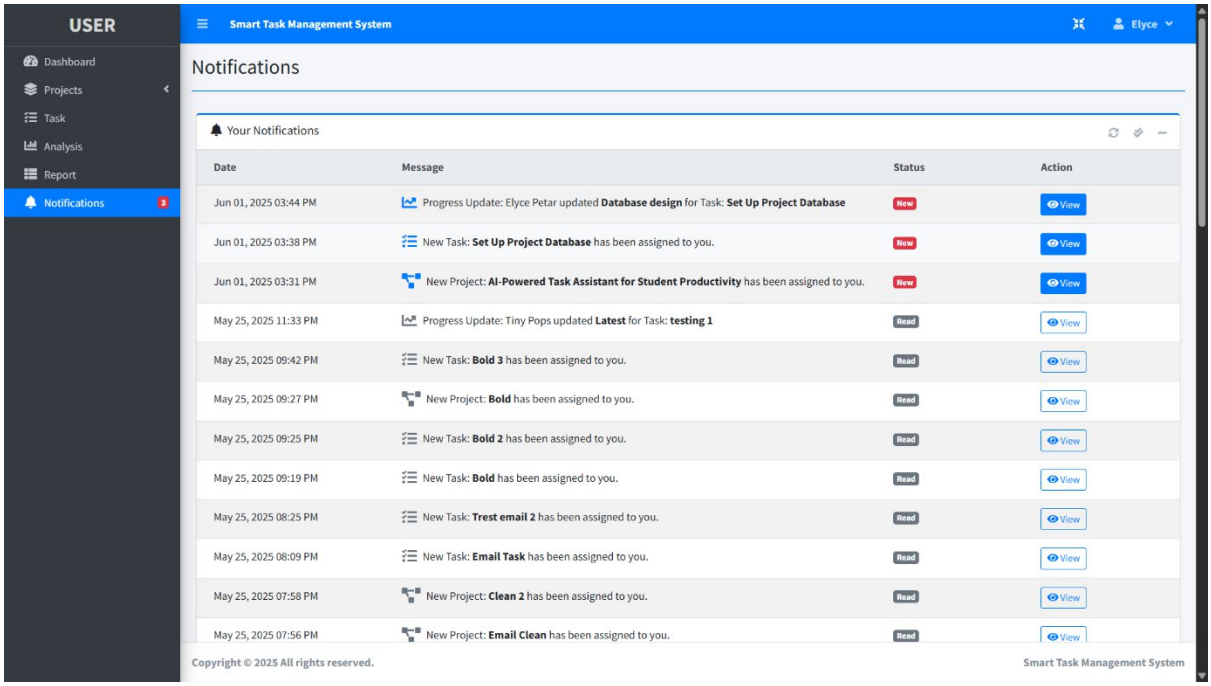


Figure 4.12 Notification

4.3.11 Notification and Email (Project Manager)

Project managers receive notifications related to system events, such as the creation of new projects, assignment of new tasks, and progress updates from team members. Action buttons within notifications allow quick navigation to relevant sections as shown in Figure 4.12.

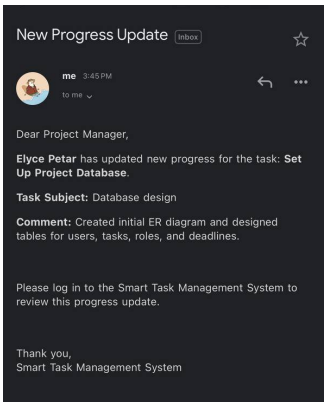


Figure 4.13 Email

In addition to in-system notifications, project managers also receive email notifications simultaneously to ensure they are promptly informed even when not actively using the system as shown in Figure 4.13.

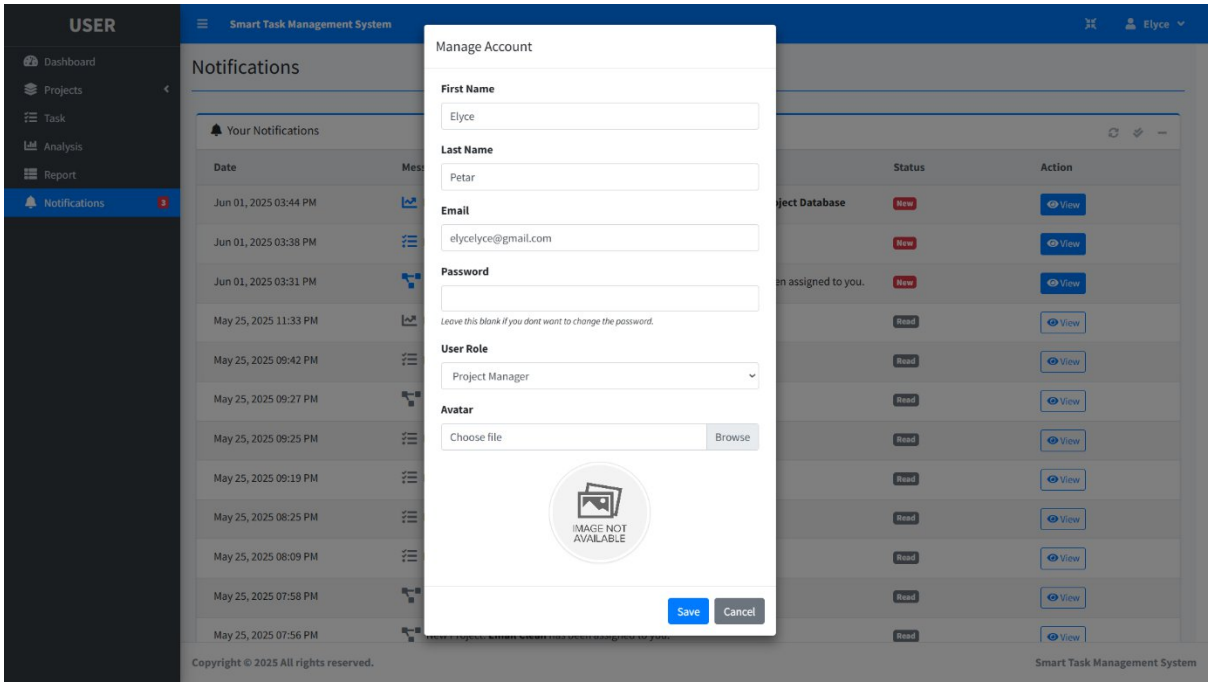


Figure 4.14 Manage User

4.3.12 Manage User (Project Manager / Team Member)

Users can update their profile information, including first name, last name, email, password, user role (project manager or team member), and avatar, ensuring that their personal and account details remain current as shown in Figure 4.14.

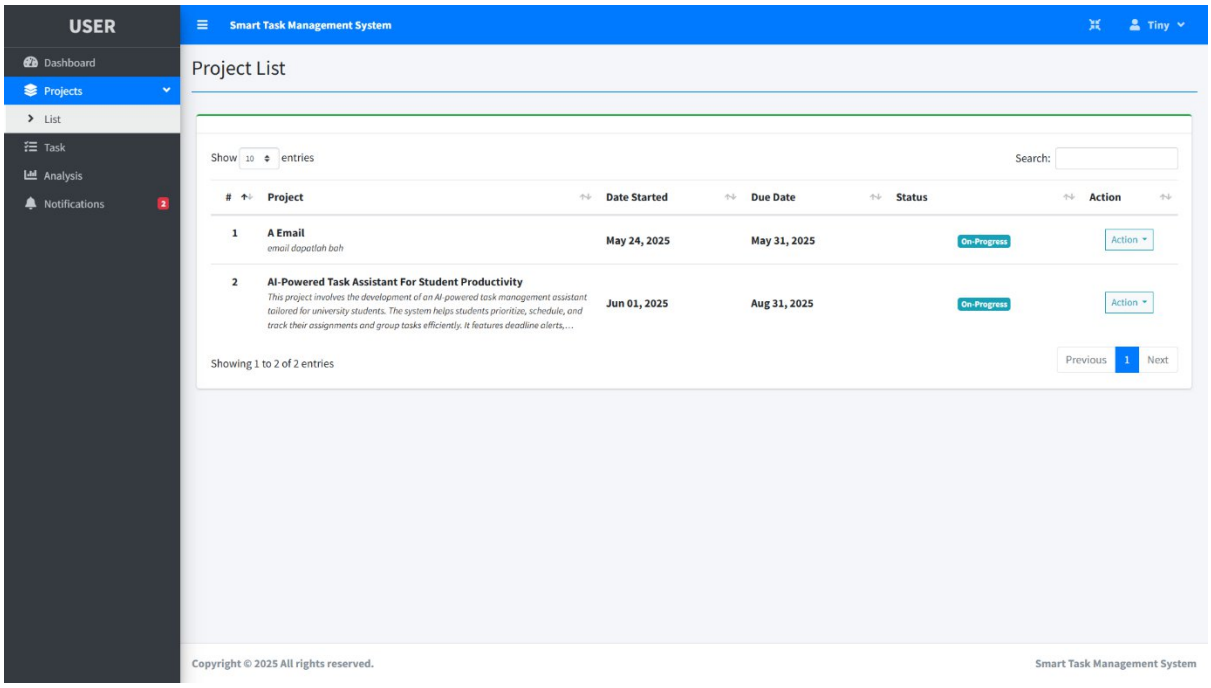


Figure 4.15 Project List Team Member

4.3.13 Project List (Team Member)

Team members can view a list of projects they are involved in, including details such as project name, start date, due date, status, and actions like viewing project details as shown in Figure 4.15.

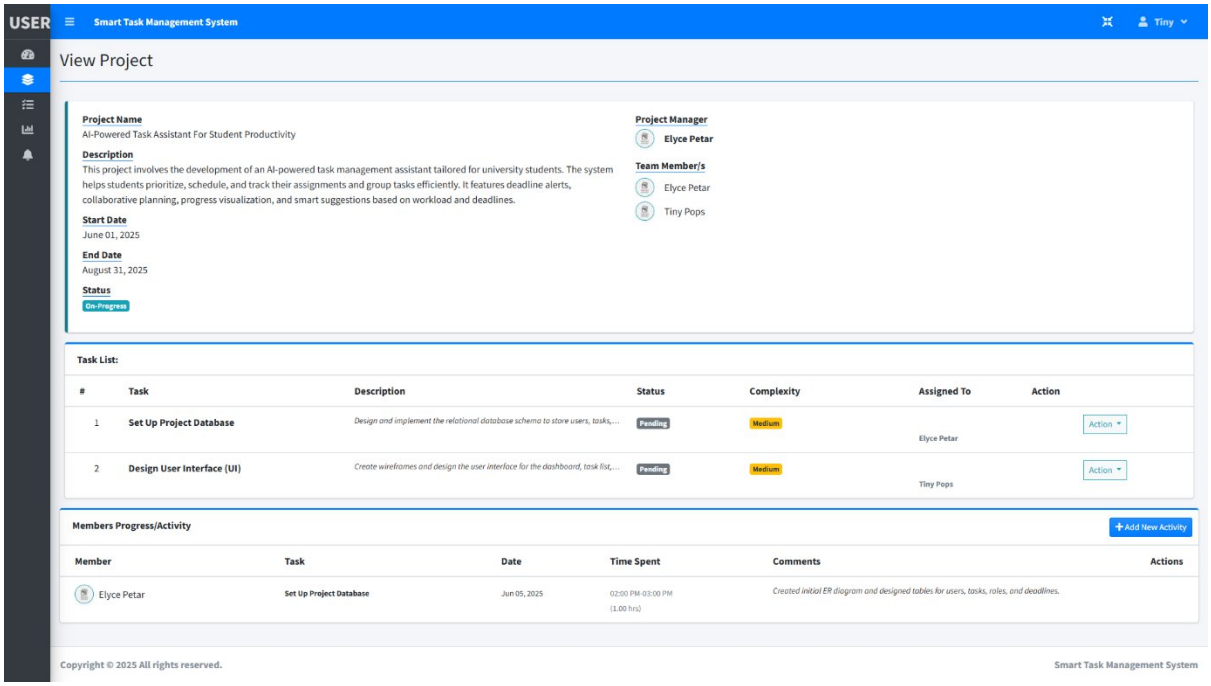


Figure 4.16 Project Detail

Upon viewing, team members can access task lists and monitor progress/activity as shown in Figure 4.16.

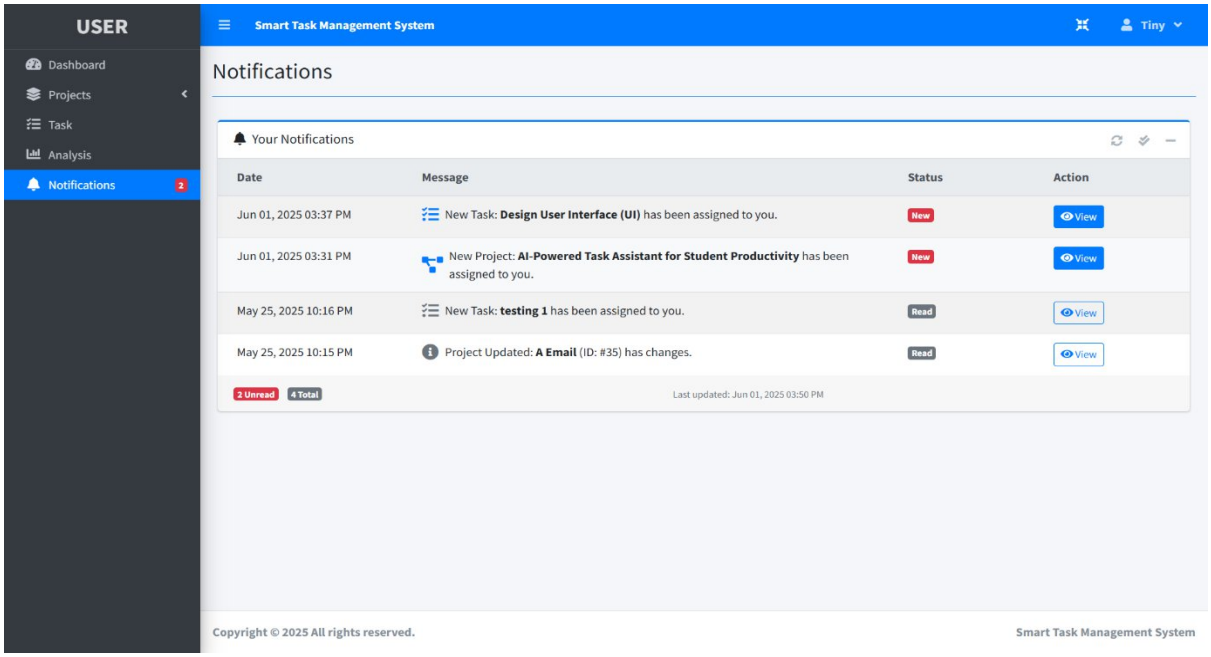


Figure 4.17 Notification Team Member

4.3.14 Notification (Team Member)

Team members receive notifications when they are added to new projects or assigned tasks as shown in Figure 4.17. These notifications include action buttons to directly access the relevant project or task for timely responses and updates.

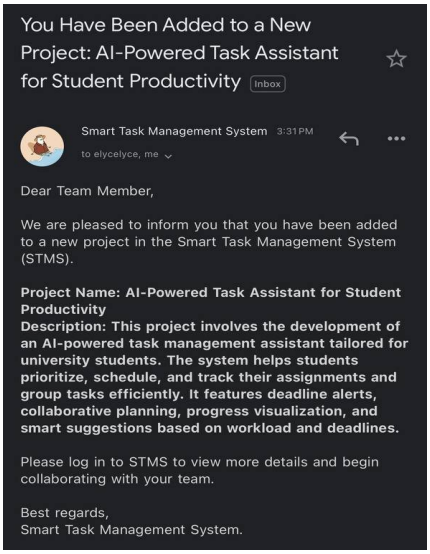


Figure 4.18 Email Team Member

These notifications are also sent via email simultaneously, ensuring that team members are promptly informed even if they are not logged into the system as shown in Figure 4.18.

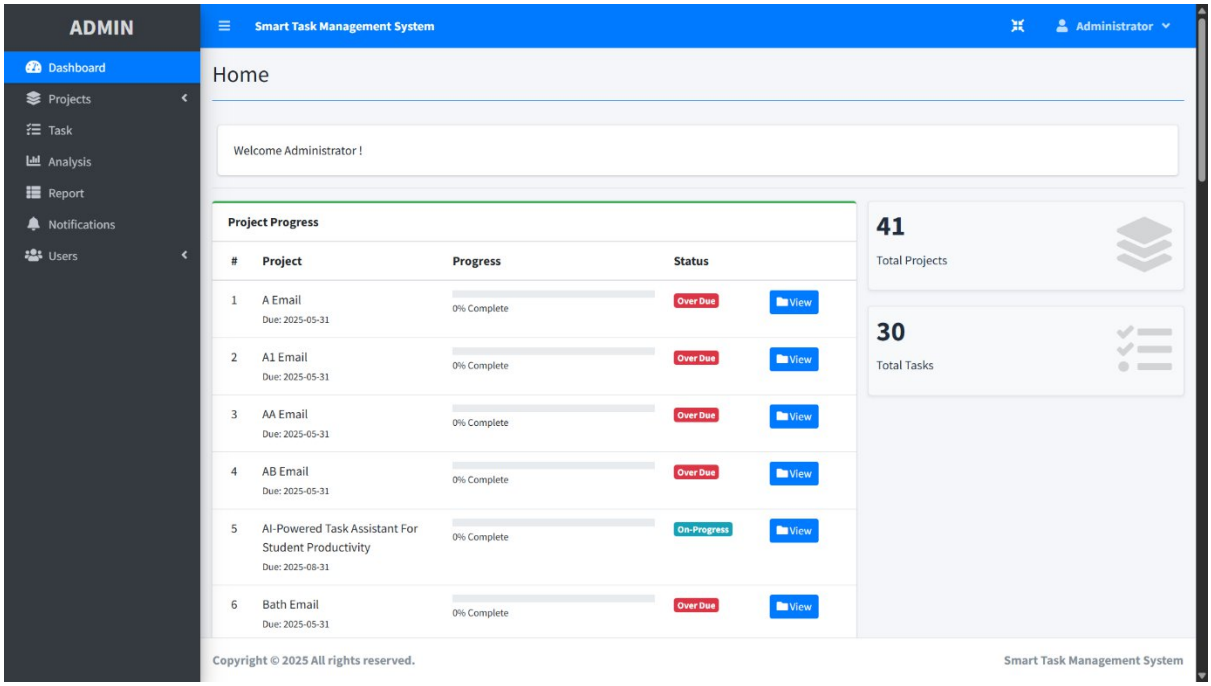


Figure 4.19 Dashboard Admin

4.3.15 Dashboard (Admin)

Administrator has an overview dashboard that displays the overall system metrics, including total project progress, number of projects, and total tasks across the platform. This provides a high-level perspective of system activity like in Figure 4.19.

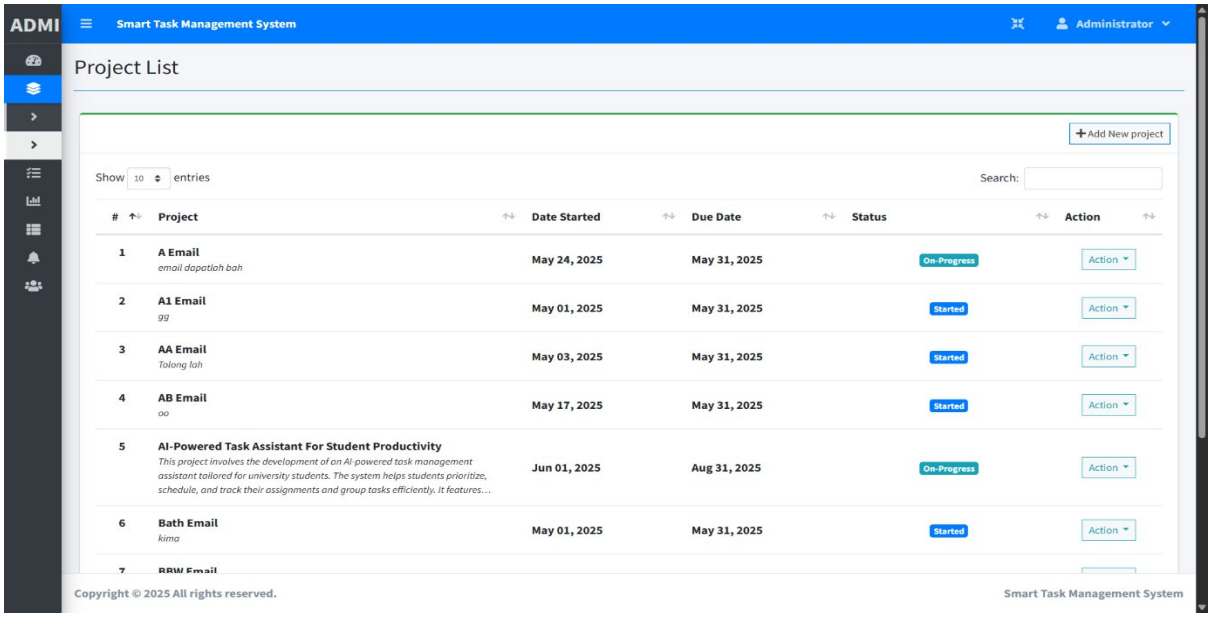


Figure 4.20 Project List Admin

4.3.16 Project List (Admin)

Admin can view a complete list of all projects in the system, with details such as project title, start date, due date, status, and have the authority to delete projects if necessary, enabling system-wide project management as shown in Figure 4.20.

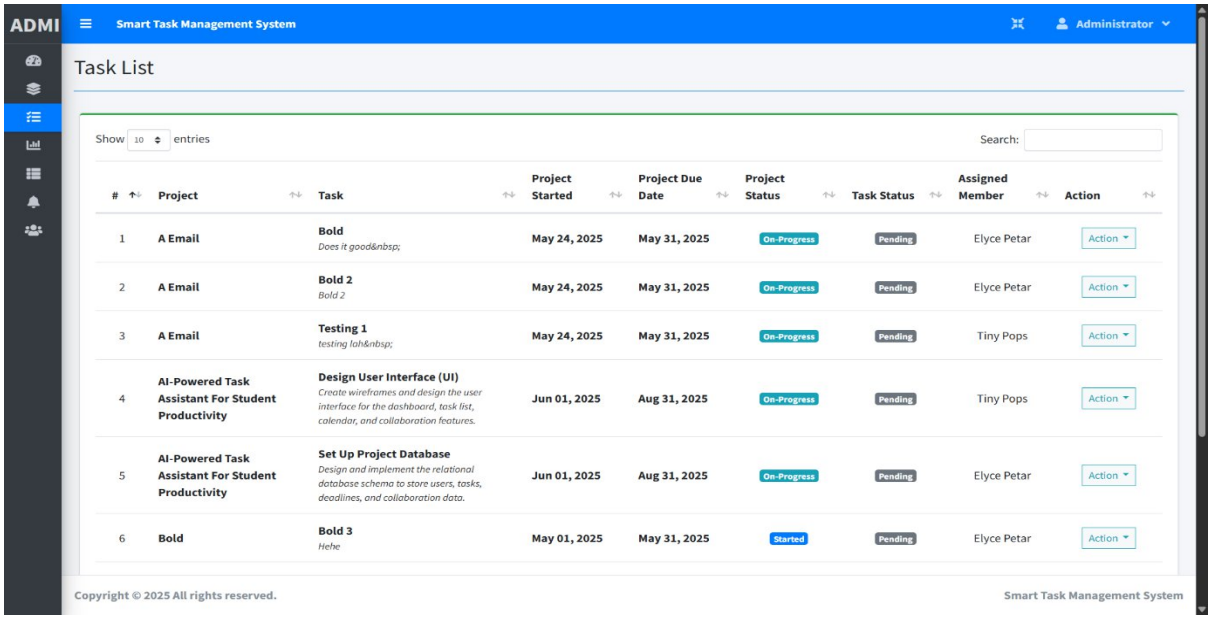


Figure 4.21 Task List Admin

4.3.17 Task List (Admin)

Based on Figure 4.21, admin can access the full task list across all projects, viewing information such as project title, task name, start and due dates, status, assigned member, and can delete tasks, providing administrative control over task data.

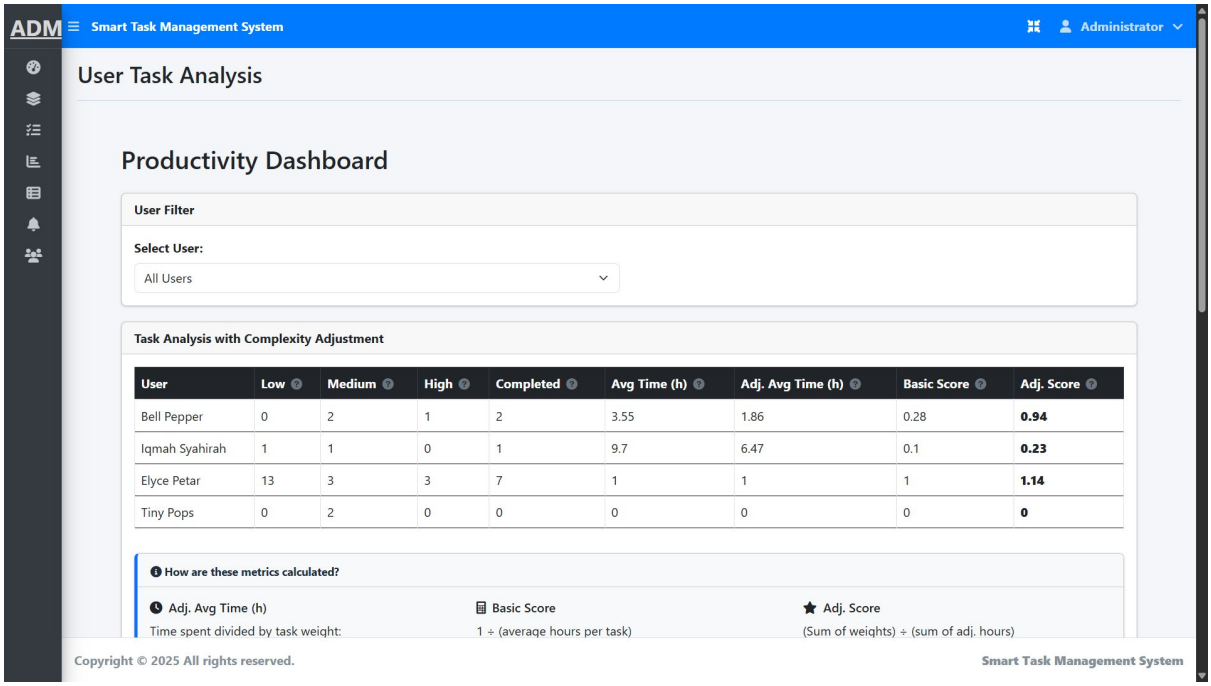


Figure 4.22 User Task Analysis Admin 1

4.3.18 User Task Analysis (Admin)

Admins can analyze productivity data for all users in the system. This includes visual reports displaying task complexity distribution and productivity comparisons, assisting in identifying trends and overall performance metrics.

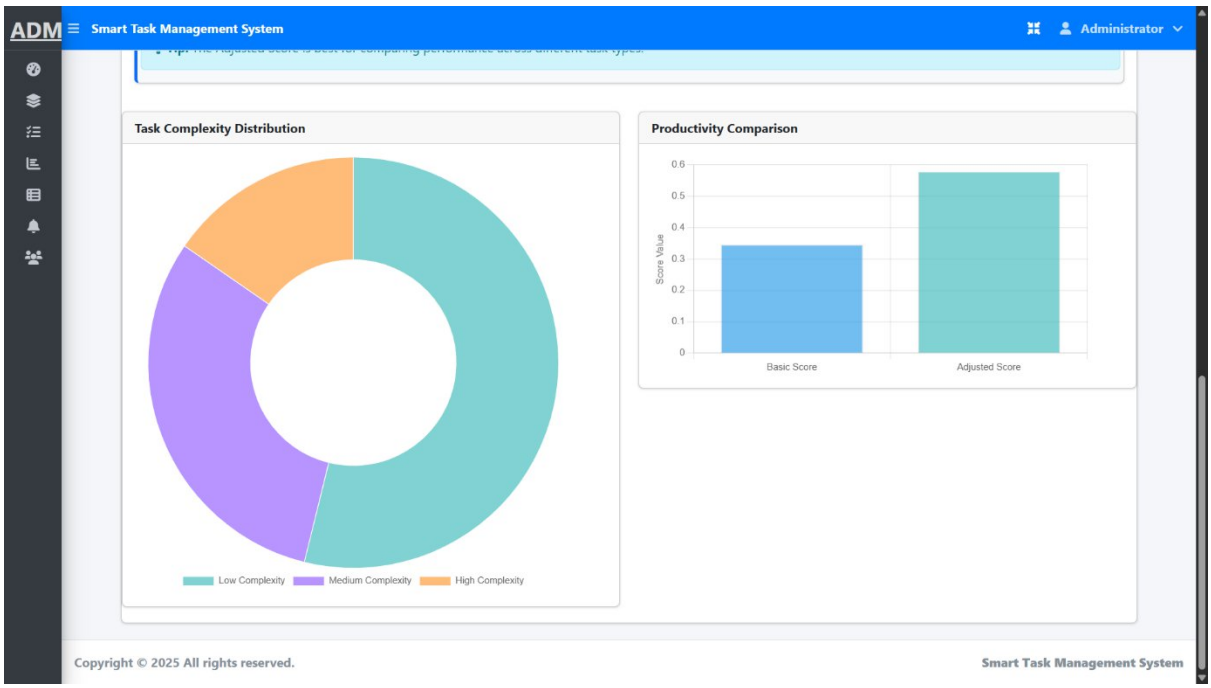


Figure 4.23 User Task Analysis Admin 2

As visual reports displaying task complexity distribution in a pie chart and productivity comparisons in a bar chart, illustrated in Figure 4.23. These tools assist in identifying trends and overall performance metrics.

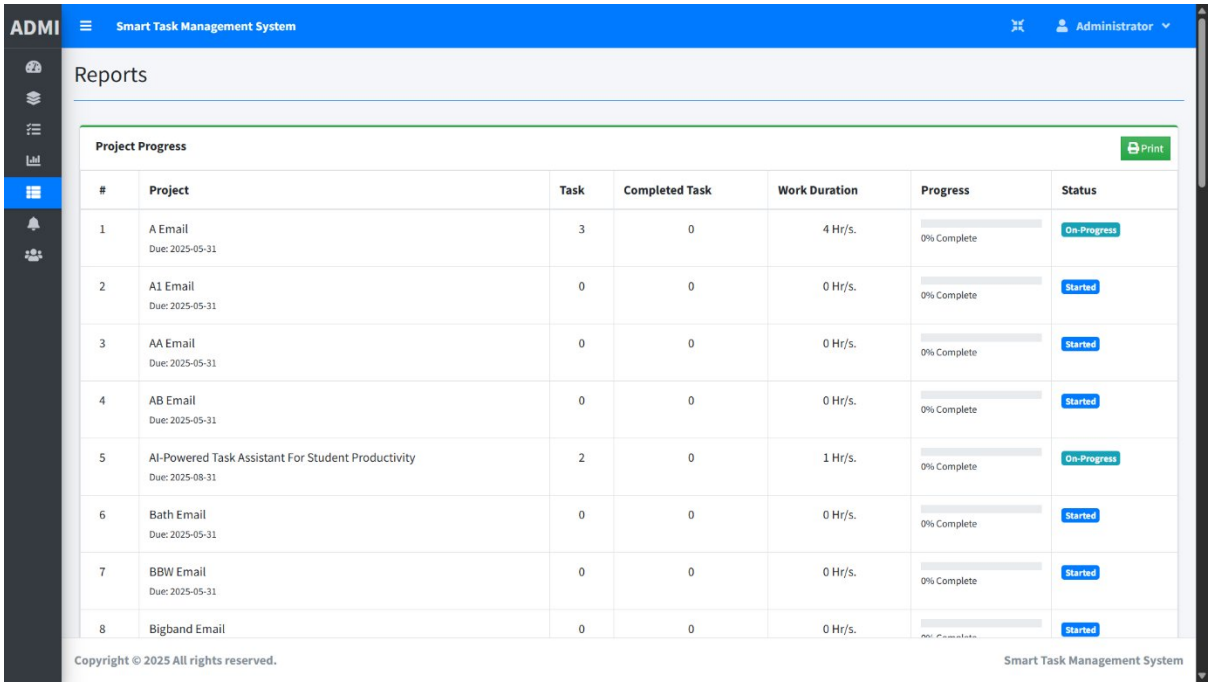


Figure 4.24 Report Admin

4.3.19 Report (Admin)

Admin can view the report title, task in the project, completed task, work duration and progress as Shown in Figure 4.24. Admin also can select button print to print or save in pdf.

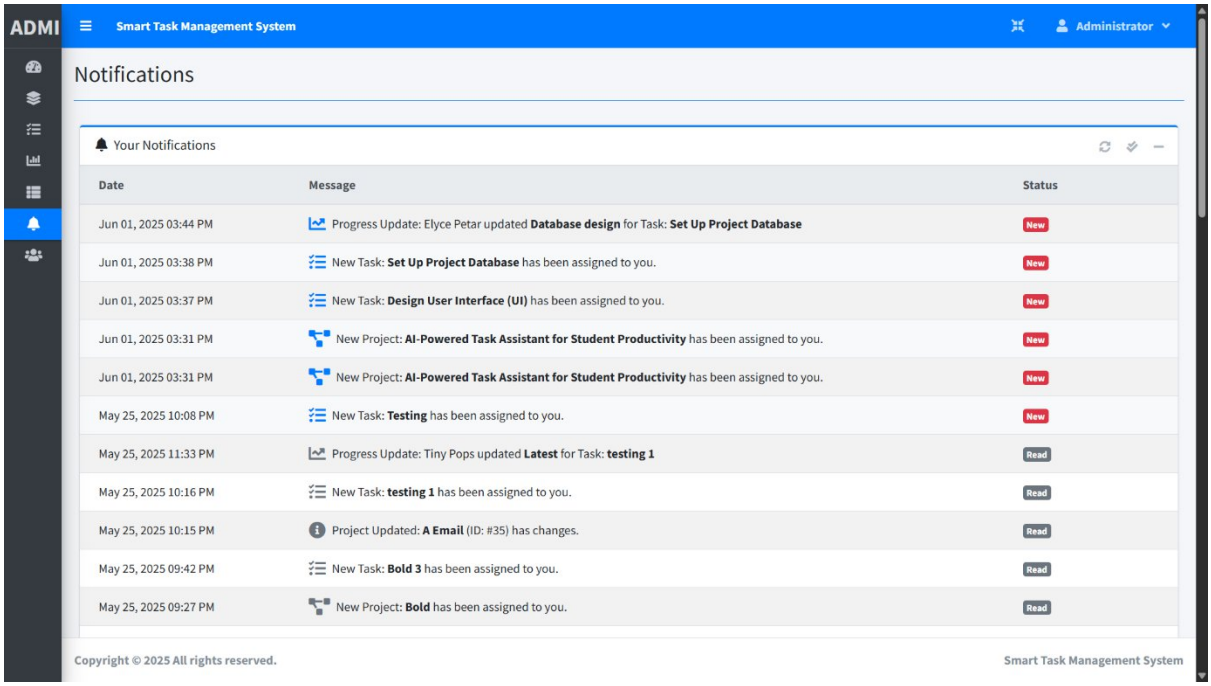


Figure 4.25 Notification Admin

4.3.20 Notification (Admin)

Admins receive notifications regarding changes and updates made by all users in the system, allowing them to stay informed about system activities and intervene, if necessary, as shown in Figure 4.25.

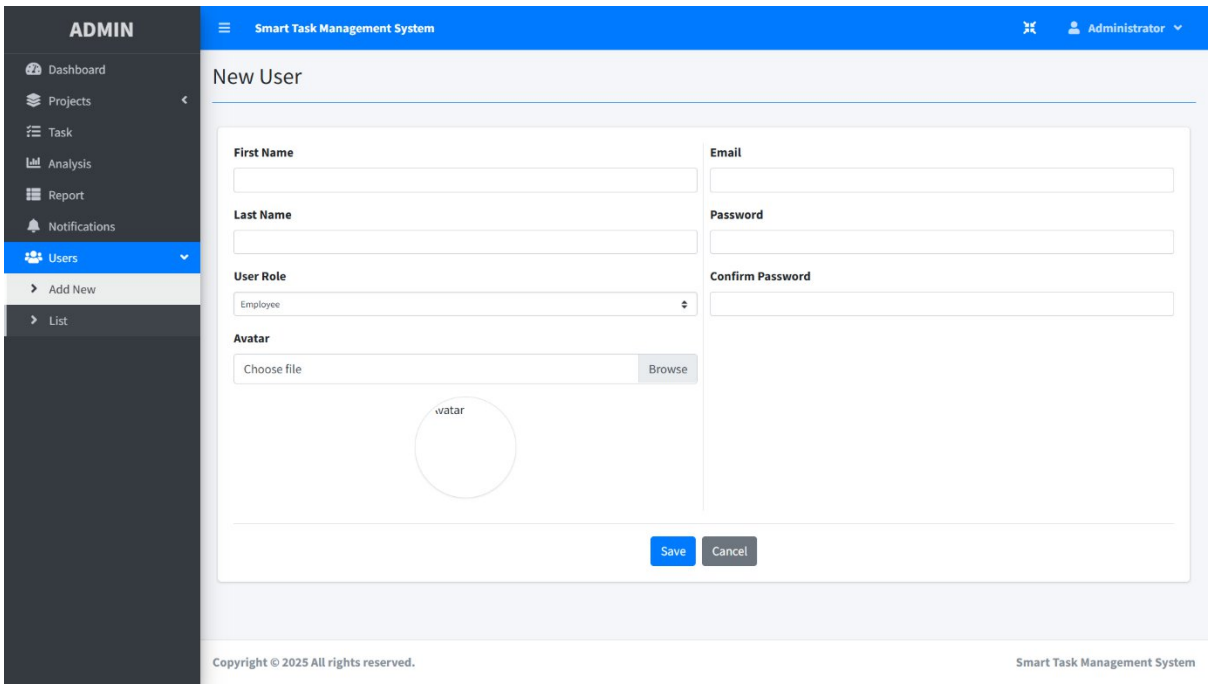


Figure 4.26 Add New User

4.3.21 User Management (Admin)

Administrators have full control over user accounts. They can add new users as shown in Figure 4.26

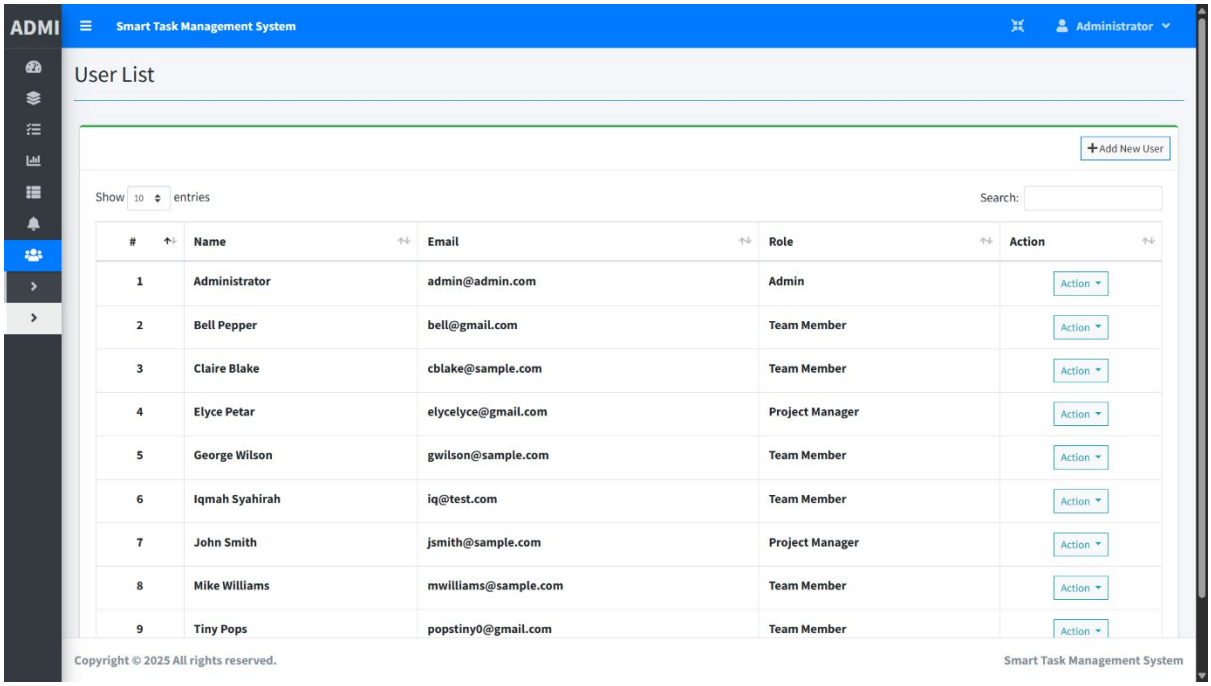


Figure 4.27 Manage User Account

Based on Figure 4.27, admin can view the list of all users, update user details, and delete users, ensuring the system maintains accurate and authorized access.

4.4 Functional Testing

Functional testing was carried out to ensure that all features and modules of the Smart Task Management System (STMS) work as intended. This type of testing validates that the system performs according to the specified functional requirements by checking each feature's output based on defined inputs.

The purpose of functional testing is to verify the functionality of components such as user authentication, project and task management, notifications, reporting, and user analysis. Each function was tested independently to ensure that it behaves correctly under both normal and edge-case scenarios.

The testing process included:

- Unit Testing: To verify the functionality of individual modules like login, sign-up, task creation, and report generation.

- Integration Testing: To ensure that different modules work together seamlessly (e.g., task updates triggering notifications).
- User Acceptance Testing (UAT): To validate that the system meets user expectations and functional requirements from a real-user perspective.

All major functionalities were tested using valid and invalid inputs to confirm the system's stability, accuracy, and reliability. The successful completion of functional testing confirms that the STMS is ready for deployment and meets the core operational needs of the intended users.

Table 4.1 Unit Test Cases for Smart Task Management System (STMS)

Module	**Test Case Description**	**Expected Output**	**Actual Result**	**Status**
Login	Enter valid email and password	User is redirected to dashboard	User successfully logged in	Passed
Login	Enter invalid email/password	Error message is displayed	Error message shown	Passed
Sign Up	Register with all valid inputs	Account is created, user redirected to login page	Account successfully created	Passed
Add Project	Submit valid project details	Project is added to database	Project appears in list	Passed
Add Task	Assign task with valid details	Task is saved and visible to assigned member	Task listed in task list	Passed
Add Activity / Progress	Add progress to task	Activity saved and reflected in task view	Activity shown	Passed
Notification System	Trigger task assignment	Notification appears for the user and an email is sent	Notification received in system and email received	Passed
Report Generation	Generate report from completed project	Report displays all task and project info	Report generated correctly	Passed
User Profile Management	Edit user details and save	Updated details appear in user profile	Changes saved successfully	Passed
Delete Project	Project manager deletes a	Project removed from system	Project successfully deleted	Passed

	project			
Top 3 Task Prioritization	System automatically prioritizes top 3 tasks	3 tasks with highest urgency/importance are shown on top	Top 3 tasks are displayed correctly	Passed
Top 3 Task Prioritization	Change task status or due date, re-check prioritization	Top 3 list updates to reflect new priority	List updated based on new task values	Passed
User Task Analysis	Generate a summary of a user's tasks by status	System displays task analysis complexity adjustment and 2 different type of graph.	Task summary generated correctly	Passed

4.4.1 Unit Testing

Unit testing involves testing individual components of the Smart Task Management System (STMS) to ensure each one functions as expected. The following table outlines the main units tested, their expected outcomes, and the results in Table 4.1.

Table 4.2 Integration Test Cases for Smart Task Management System (STMS)

Test Case	**Modules Involved**	**Expected Result**	**Status**
Login and Dashboard Loading	Login Module, Dashboard Module	Upon successful login, user is redirected to the correct dashboard.	Pass
Task Creation and Notification	Task Module, Notification Module	Creating a task sends a notification in the system and an email to the assigned user.	Pass
Project Assignment and User	Project Module, User Module	Assigned users can view and interact with their designated	Pass

Access		project.	
Task Update and Report Refresh	Task Module, Reporting Module	Updating task progress reflects accurately in the reporting dashboard.	Pass
User Update and Profile Display	User Management Module, Dashboard Module	Changes in user profile are displayed correctly across the system.	Pass
Top 3 Task Prioritization Display	Dashboard Module	The 3 highest-priority tasks are dynamically ranked and displayed at the top.	Pass

4.4.2 Integration Testing

Integration testing ensures that the interconnected modules of the Smart Task Management System (STMS) operate seamlessly as a whole. It verifies that the data flow between components functions correctly. For instance, creating a task should trigger a corresponding notification, and updating task progress should reflect accurately in reports. This testing phase is crucial for identifying issues that may not appear in isolated unit tests, such as database integration problems, mismatched user roles, or failures in UI response. Table 4.2 outlines the key integration test cases conducted to verify that components of the system interact as expected.

Table 4.3 User Acceptance Test (UAT) for Smart Task Management System (STMS)

Test Scenario	**User Role**	**Expected Outcome**	**Status**
Sign Up and Login	Project Manager, Team Member	Users can sign up, log in, and access appropriate dashboards.	Pass

Project Creation and Task Assignment	Project Manager	Project manager can create a project and assign tasks to team members.	Pass
Task Execution and Productivity Tracking	Team Member	Team member can update task progress and view productivity analytics.	Pass
Notification Visibility	All Roles	Users receive relevant notifications for projects, tasks, and updates within the system, and corresponding email notifications are also sent.	Pass
Report Generation and Download	Project Manager, Admin	Reports display correct information and are downloadable or printable.	Pass

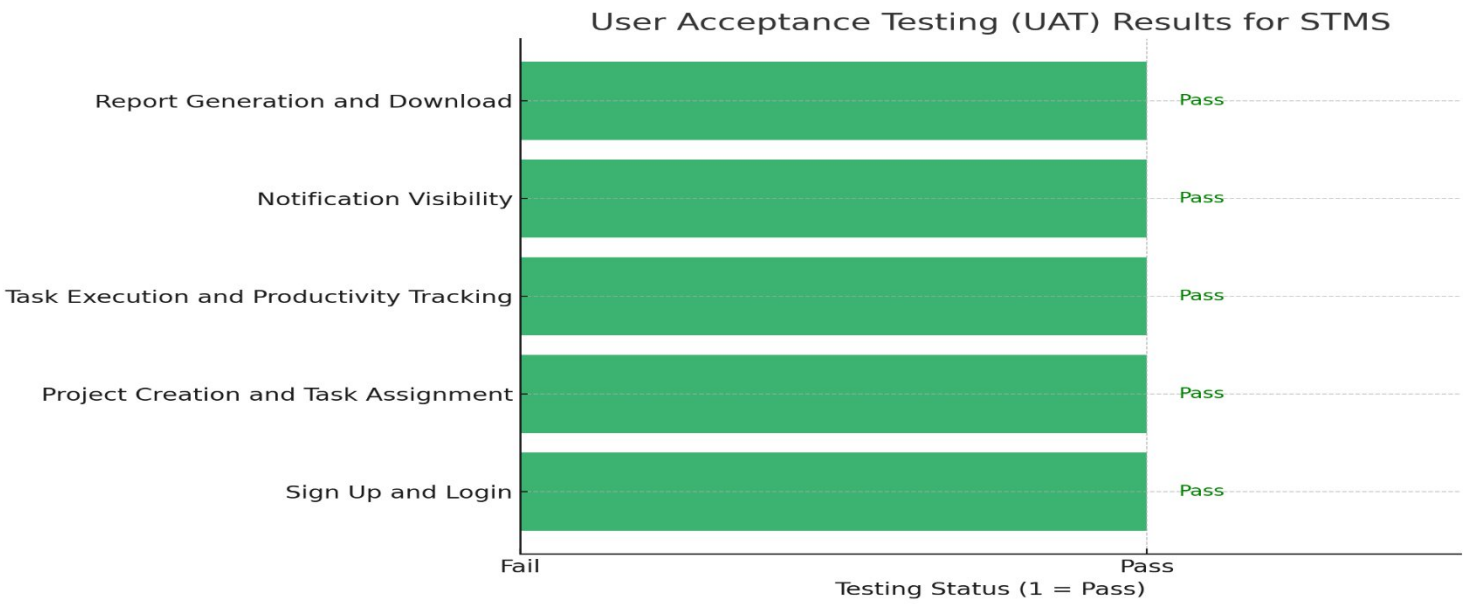


Figure 4.28 User Acceptance Testing (UAT) Result for STMS

4.4.3 User Acceptance Testing (UAT)

User Acceptance Testing (UAT) is performed to ensure the system meets user needs and functional requirements from a real-world usage perspective. In the case of the Smart Task Management System (STMS), UAT was conducted with sample users such as students acting as project managers or team members. These users were asked to complete common tasks like signing up, creating projects, and managing tasks. Their feedback was used to confirm the system's usability and functionality before final deployment.

Table 4.3 presents the User Acceptance Testing scenarios, expected outcomes, and testing status for the STMS. This table summarizes how sample users verified key functionalities of the system to ensure it meets real-world user requirements.

4.5 Non-functional Testing

Non-functional testing focuses on evaluating the qualities of the Smart Task Management System (STMS) that affect the user experience and system performance, rather than specific functions. In this project, usability testing was conducted to assess how easy and intuitive the system is for users. This testing helped identify any issues related to navigation, interface design, and overall user satisfaction, ensuring the system is accessible and efficient for its intended users.

4.5.1 Usability Testing

Usability testing was conducted to evaluate how effectively users could interact with the Smart Task Management System (STMS). The focus was on three key aspects: functionality of the system, user interface and design, and overall evaluation of the system. A total of 14 students from the Faculty of Computer Science and Information Technology participated in the testing. A Google Form was distributed to these participants to gather structured feedback based on their hands-on experience with the system.

Table 4.4 Functionality of STMS

Functionality	**1**	**2**	**3**	**4**	**5**
Log In	-	-	-	-	14
Sign Up	-	-	-	-	14
Create New Project	-	-	-	14	-
Add Task	-	-	-	-	14
Assign Task	-	-	-	-	14
Update Task Progress	-	-	-	1	13
View Dashboard	-	-	-	-	14
View Notifications (System + Email)	-	-	-	14	-
View Productivity Analysis	-	-	-	4	10
Generate Report	-	-	-	-	14
Logout	-	-	-	-	14

4.5.1.1 Functionality of STMS

The results are shown in Table 4.4, where most participants reported a positive experience across all tested functions. This subsection assesses how well users were able to perform key system tasks such as signing in, signing up, and interacting with core features like inserting task data and viewing analytics. Each task was rated on a scale from 1 to 5, where 1 indicates "Very Difficult" and 5 indicates "Very Easy."

Table 4.5 User Interface and Design

Design Aspect	**1**	**2**	**3**	**4**	**5**
Layout Clarity	0	0	0	2	12
Colour Scheme	0	0	0	6	8
Font Readability	0	0	1	3	10
Navigation Ease	0	0	3	6	5
Button Placement	0	0	4	5	5
Responsiveness (Page Speed)	0	4	2	8	0
Overall Visual Aesthetics	0	0	0	7	7

4.5.1.2 User Interface and Design

The results of this usability testing are summarized in Table 4.5, which shows that most students found the system's interface to be clear, visually appealing, and easy to navigate, with high scores especially in colour scheme, font readability, and overall visual aesthetics. This section evaluates users' perception of the system's interface layout, visual clarity, ease of navigation, and overall design aesthetics. The students rated various design aspects on a scale from 1 to 5, where 1 indicates "Very Poor" and 5 indicates "Excellent."

Table 4.6 Overall Evaluation of STMS

Evaluation Criteria	**1 (Very Poor)**	**2 (Poor)**	**3 (Average)**	**4 (Good)**	**5 (Excellent)**
Ease of Use	0	0	0	2	12
System Efficiency	0	0	2	3	9

User Satisfaction	0	0	3	2	9

4.5.1.3 Overall Evaluation of STMS

The results, shown in Table 4.6, reflect that most users found the system intuitive and satisfactory for managing tasks and projects efficiently. The overall evaluation of the Smart Task Management System (STMS) aimed to gather users' general impressions regarding the system's usability, reliability, and satisfaction. The evaluation focused on three primary aspects: ease of use, system efficiency, and user satisfaction. Participants rated their experience on a scale of 1 to 5, where 1 indicates "Very Poor" and 5 indicates "Excellent".

4.5.1.4 User Feedback and Suggestions

Following the usability testing of the Smart Task Management System (STMS), feedback was collected from 14 students via a Google Form. Overall, users found the system functional and user-friendly, but they also provided several suggestions to enhance the system further.

The main suggestions from users include:

- Mobile Accessibility: Users expressed a desire to access and use the STMS conveniently from their smartphones. Improving mobile responsiveness or developing a dedicated mobile app would enhance usability on the go.
- Dark Mode Feature**:** Several users requested a dark mode option to reduce eye strain during extended use, especially in low-light environments.
- Customizable Themes: Some participants suggested allowing users to customize the system's theme and interface colors based on personal preferences to create a more personalized user experience.
- Action Links in Email Notifications**:** Users would like email notifications to contain direct action links, enabling them to quickly navigate to relevant tasks or projects from their inbox without needing to log in first.

These suggestions will serve as a valuable reference for future system improvements and feature enhancements. Implementing these changes will help further align the STMS with user needs and expectations, improving its long-term usability and effectiveness.

4.6 Conclusion

In conclusion, this chapter discussed the implementation and testing of the Smart Task Management System (STMS). It explained the tools and technologies used to develop the system, such as PHP, MySQL, and PHPMailer, and described the setup of the development environment. Key system functions were outlined for different user roles, including project managers, team members, and administrators.

Testing was carried out to ensure that all features worked correctly and met user requirements. Both unit and integration testing showed that the system was stable, functional, and reliable. The successful implementation and testing confirm that the STMS is ready to be used by UNIMAS students for effective task and project management.

Chapter 5: Conclusion

5.1 Introduction

This chapter concludes the development and implementation of the Smart Task Management System (STMS) for UNIMAS students. It summarizes the key achievements of the project, evaluates how well the initial objectives were met, discusses the limitations encountered, and suggests future enhancements. This project provides insights into the development of effective task management tools tailored for academic use.

5.2 Project Achievements

The STMS aimed to provide a structured platform to help students efficiently manage their academic tasks and collaborations. The following table presents a comparison between the project objectives and the actual outcomes:

Table 5.1 Project Objectives and Project Achievement

Objectives	**Project Achievement**
1. To develop a user-friendly task management system that simplifies task assignment and tracking.	A clean and intuitive web interface was designed, enabling students to easily assign, track, and manage their tasks.
2. Integrating task prioritization, deadline management, real-time status updates, and notifications to enhance time management and team communication.	Features such as task priority levels, due dates, real-time task status changes, and notification alerts were implemented to support productivity and collaboration.
3. To implement reporting tools and task analysis and performance evaluation.	A reporting module was introduced to allow users to generate task summaries and analyze completion status and team progress.
4. To evaluate the performance of the proposed system.	The system was tested with a sample group of users, and performance evaluations were carried out through user feedback and functionality testing.

Overall, the project met its core objectives by delivering a functional system that enhances academic task organization, group coordination, and performance visibility.

5.3 Project Limitations

Although the system achieved its major goals, certain limitations were observed:

- **No native mobile support:** The system is only available as a web application, limiting accessibility for mobile users without a mobile app.
- **Basic reporting features:** While basic reports and performance summaries were included, more detailed analytics could further benefit users.
- **Lack of real-time collaboration tools:** Instant messaging or live collaboration features were not integrated, which could enhance teamwork.
- **Limited scalability testing:** The system was tested in a controlled environment, and large-scale deployment may require further optimization and testing.

5.4 Future Works

To further improve the STMS and expand its usability, the following future enhancements are proposed:

- **Develop a mobile application:** A mobile version would increase system accessibility and user engagement.
- **Add advanced analytics and reporting tools:** Enhancing the reporting module with charts, productivity trends, and detailed evaluations could provide better insights.
- **Enable real-time communication:** Features such as chat or team discussion boards could be added to support seamless group interaction.
- **Incorporate AI-Based recommendations :** Suggestions for task prioritization, scheduling and deadline adjustment based on user habit and workload instead of manually set baseline.
- **System integration with external tools:** Allowing integration with academic calendars and learning management systems would streamline task tracking across platforms.

5.5 Summary

This chapter has summarized the Smart Task Management System's goals, outcomes, and future potential. The project successfully delivered a system that simplifies task handling,

fosters collaboration, and improves productivity for UNIMAS students. With additional features and improvements, STMS can evolve into a comprehensive academic productivity suite with broader applications.

REFERENCES

Afifah, S., & Alwi, W. (2024). *UNIMAS FCSIT FYP SUPERVISION TRACKING SYSTEM*. https://ir.unimas.my/id/eprint/34592/1/Sharifah%20Afifah%20Binti%20Wan%20Alwi%20-%2024%20pgs.pdf

Binti, R., Wahab, A., & Darul, P. (2012). *FINAL YEAR PROJECT ONLINE MANAGEMENT SYSTEM (FYPOS): CLIENT AND SERVER DESIGN*. http://utpedia.utp.edu.my/id/eprint/3814/1/RAIHANAH_11409_Dissertation.pdf

Doist. (2007, January 28). *How to Use Todoist Effectively – The Complete Guide*. Todoist Inspiration Hub. https://todoist.com/inspiration/how-to-use-todoist-effectively

Help Center Asana. (2008, December 16). *How Asana works*. Asana.com. https://help.asana.com/s/article/how-asana-works?language=en_US

Madang, A. D. A. (2020). *TaPro (TaskProgress); A System To Monitor Progress of Students' Task ANDALINA DAYANG AK MADANG 55473 BACHELOR OF COMPUTER SCIENCE WITH HONOUR (MULTIMEDIA COMPUTING)*. https://ir.unimas.my/id/eprint/33996/1/Andalina%20Dayang%20Ak%20Madang%20-%2024%20pgs.pdf

Moon, L. (2022, June 22). *How to use Trello: organization and productivity for beginners*. https://blog.trello.com/2014/10/06/beginner-tips-for-using-trello

Nurzi, F. A., & Ab Wahab, K. (2022). Web-Based Student Task Management System. *Engineering, Agriculture, Science and Technology Journal (EAST-J)*, *1*(1), 50–56. https://doi.org/10.37698/eastj.v1i1.119

APPENDICES

Appendix A: Task Schedule and Gantt Chart of the Project

Phase	Task	Start Date	End Date	Duration
Requirement Analysis	FYP Briefing	10/10/2024	11/10/2024	1
	Brief Proposal	11/10/2024	20/10/2024	9
	Feedback and Comment from Examiner	21/10/2024	28/10/2024	7
	Full Proposal	29/10/2024	14/11/2024	16
	Chapter 1	15/11/2024	21/11/2024	6
	Research on Current Existing System	22/11/2024	13/12/2024	21
	Chapter 2	14/12/2024	5/1/2025	22
	Submission of Chapter 2	5/1/2025	6/1/2025	1
System Design	Developing Flowchart for the System	26/1/2025	1/2/2025	6
	Designing the System	2/2/2025	16/2/2025	14
	Report for Chapter 3	17/2/2025	23/2/2025	6
	Submission on Chapter 3	23/2/2025	24/2/2025	1
	FYP 1 Final Report	6/1/2025	17/1/2025	11
	FYP 1 Presentation	24/1/2025	25/1/2025	1
Implementation	Front-End Development	10/3/2025	30/3/2025	20
	Backend Development	10/3/2025	20/4/2025	41
	Task Prioritization	31/3/2025	13/4/2025	13
	Notification System	14/4/2025	27/4/2025	13
	User Task Analysis	5/5/2025	11/5/2025	6
	UI Refinement	5/5/2025	11/5/2025	6
Testing & Maintenance	Integration Testing	19/5/2025	25/5/2025	6
	User Acceptance Testing	26/5/2025	1/6/2025	6
	Bug Fixing (if any)	2/6/2025	8/6/2025	6
	Report for Chapter 4	2/6/2025	8/6/2025	6
	Submission for Chapter 4	24/5/2025	25/5/2025	1
	Report for Chapter 5	26/5/2025	30/5/2025	4
	Submission on Chapter 5	30/5/2025	31/5/2025	1
	First Draft of FYP 2 Full Report	9/6/2025	10/6/2025	1
	Finalizing Final Report	11/6/2025	22/6/2025	11
	Final Report Submission	22/6/2025	23/6/2025	1

Figure i: Schedule task for Final Year Project I and Final Year Project II

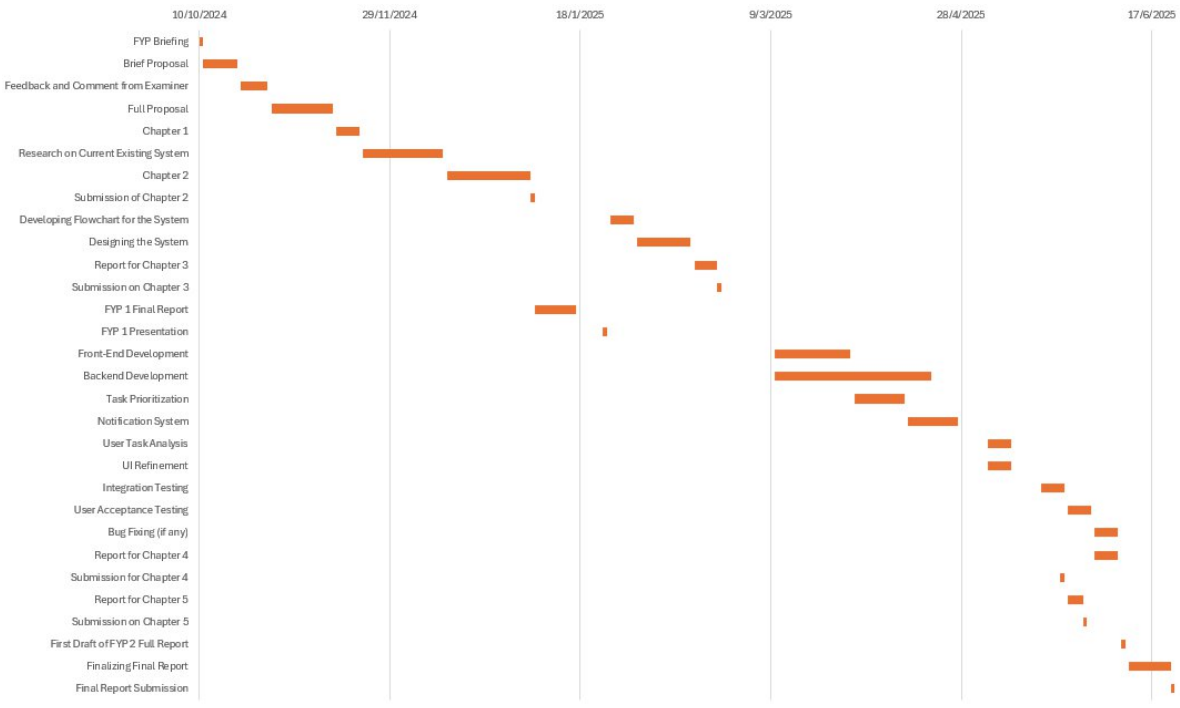


Figure ii: Gantt Chart for Project Schedule

Appendix B: Survey Findings

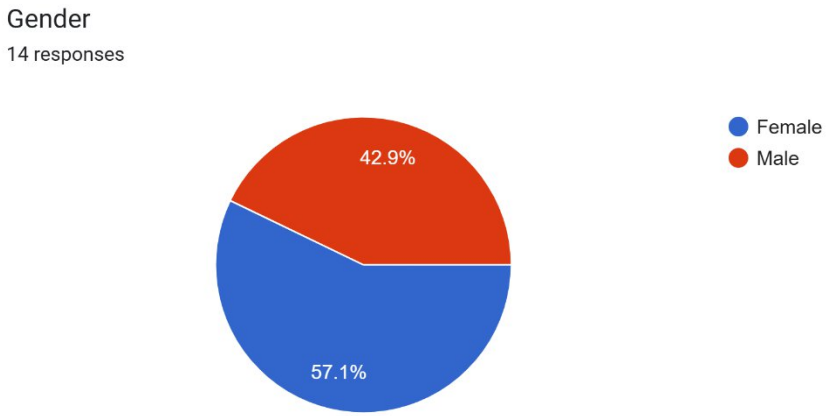


Figure iii: Question 1

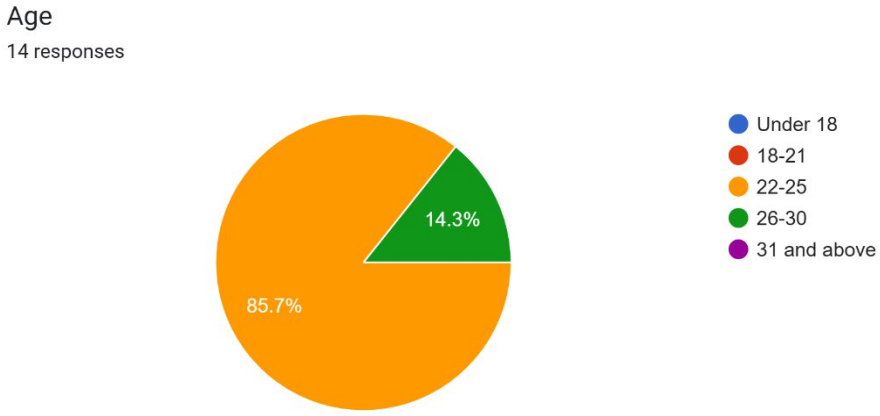


Figure iv: Question 2

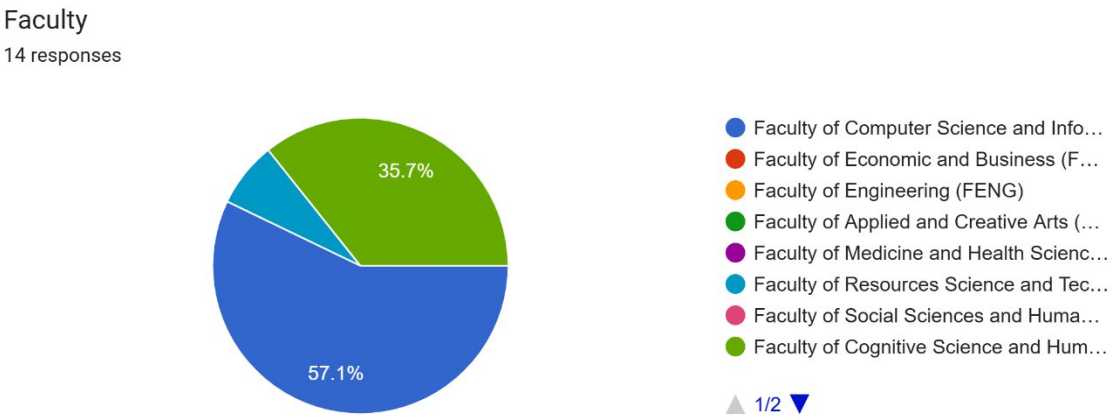


Figure v: Question 3

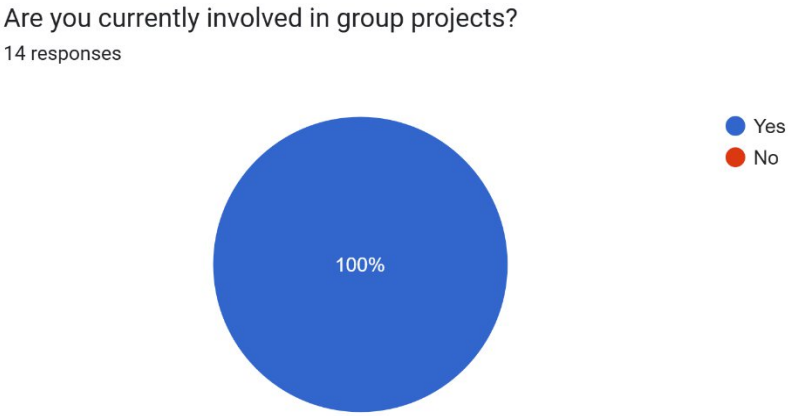


Figure vi: Question 4

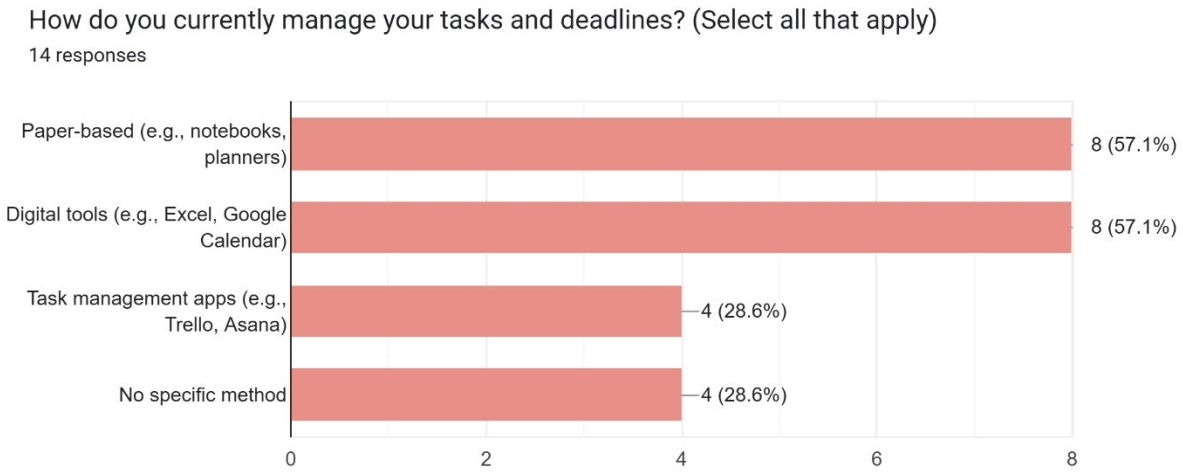


Figure vii: Question 5

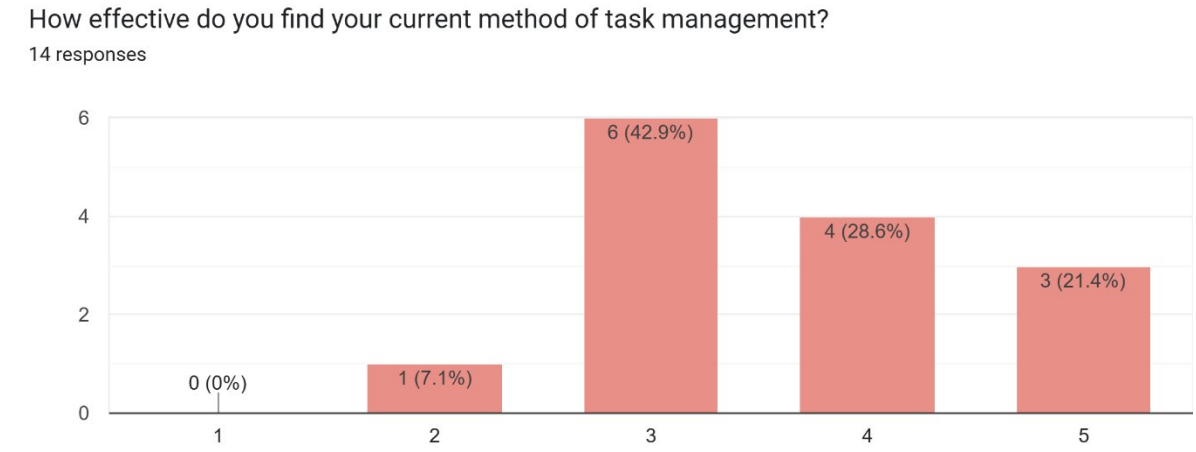


Figure viii: Question 6

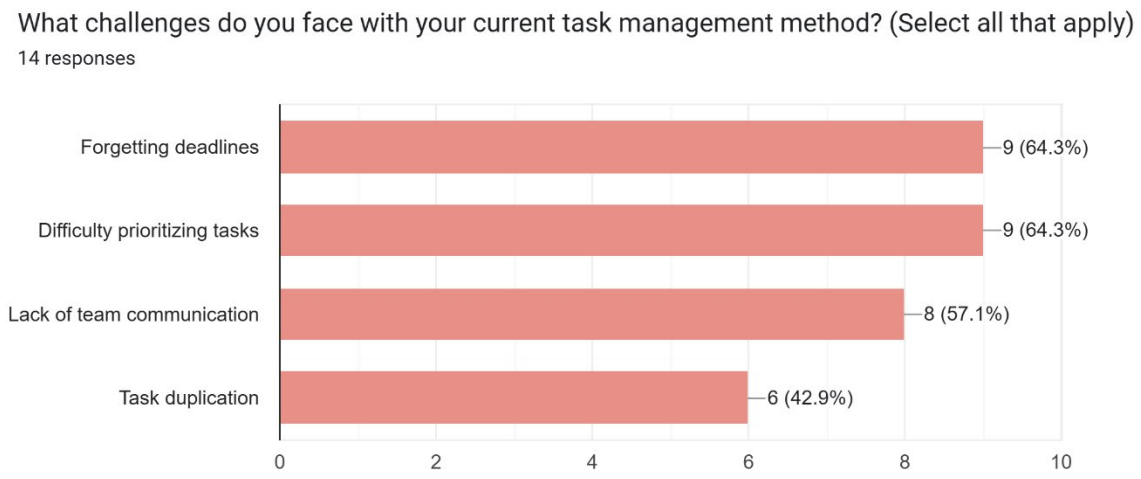


Figure ix: Question 7

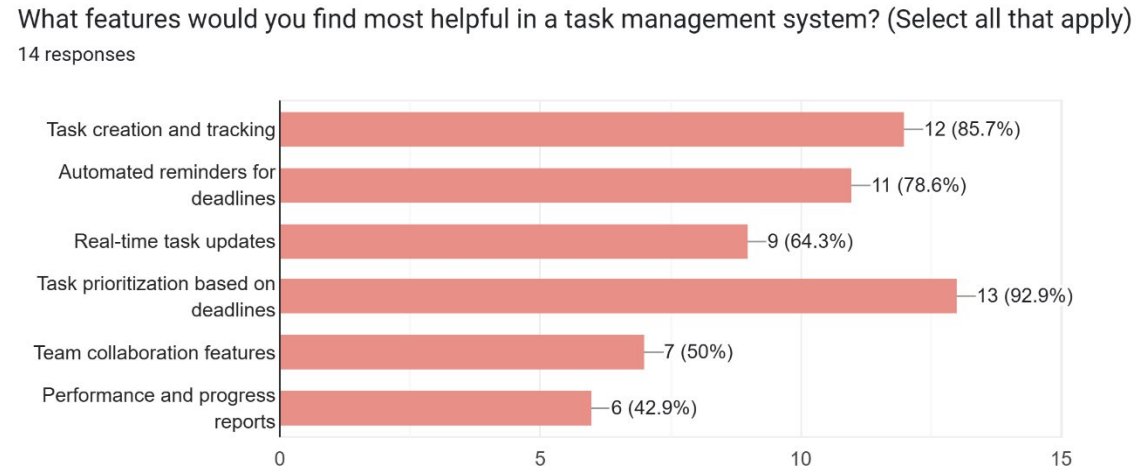


Figure x: Question 8

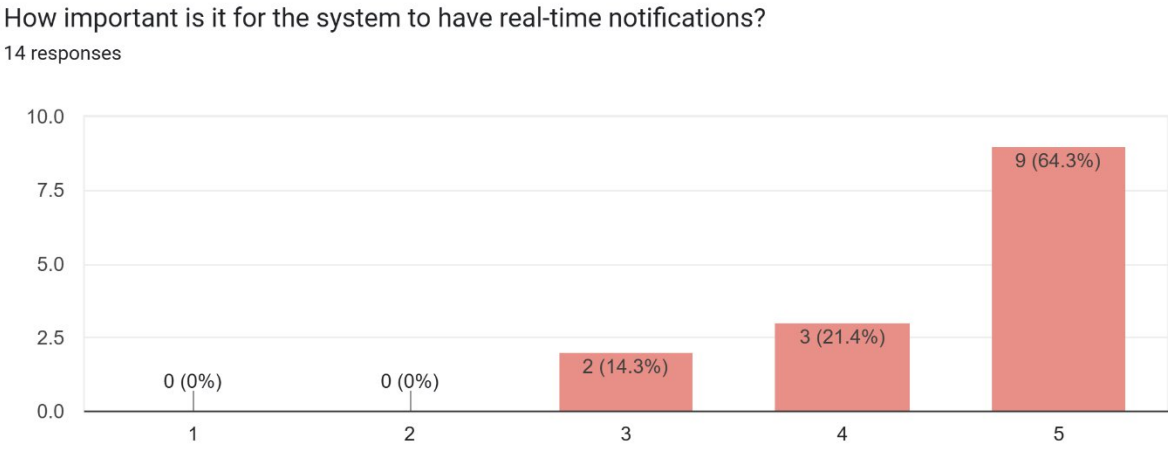


Figure xi: Question 9

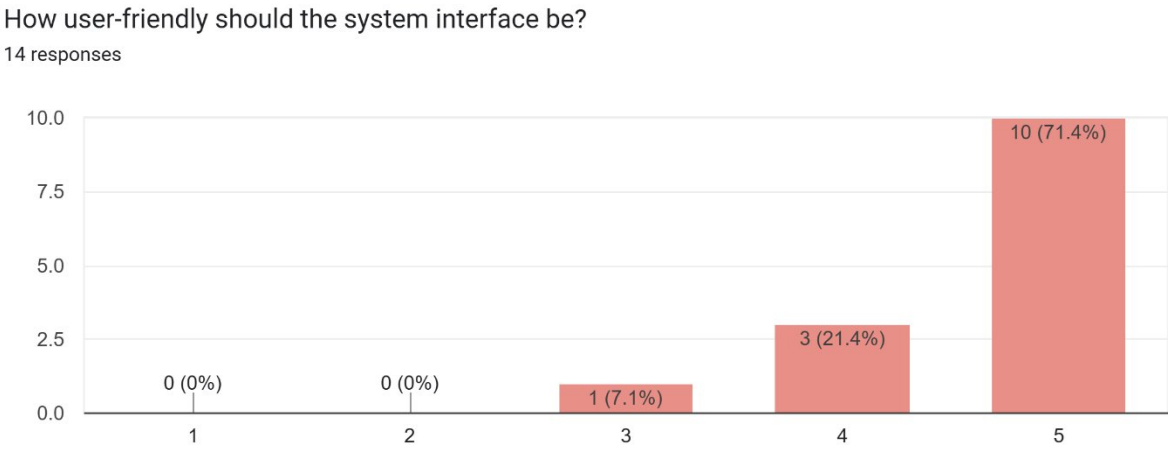


Figure xii: Question 10

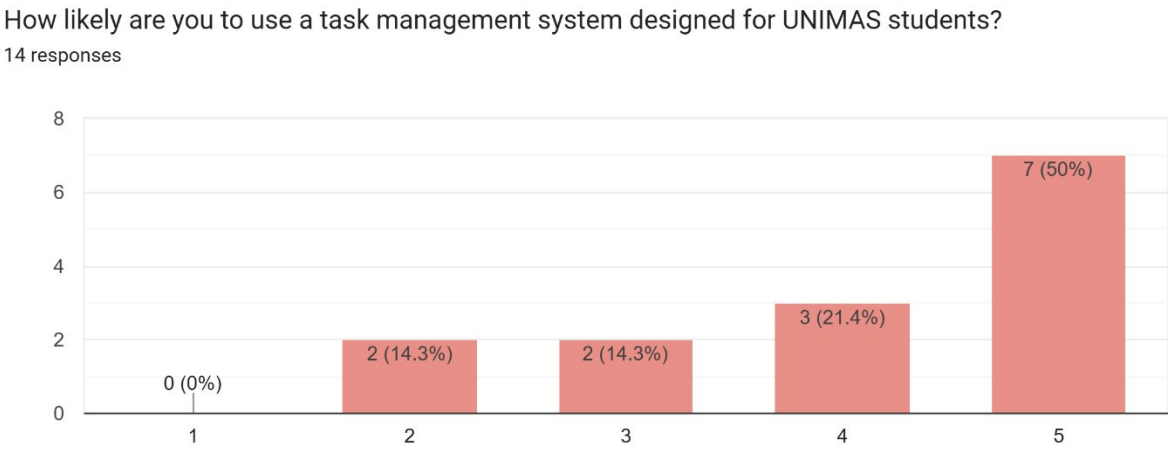
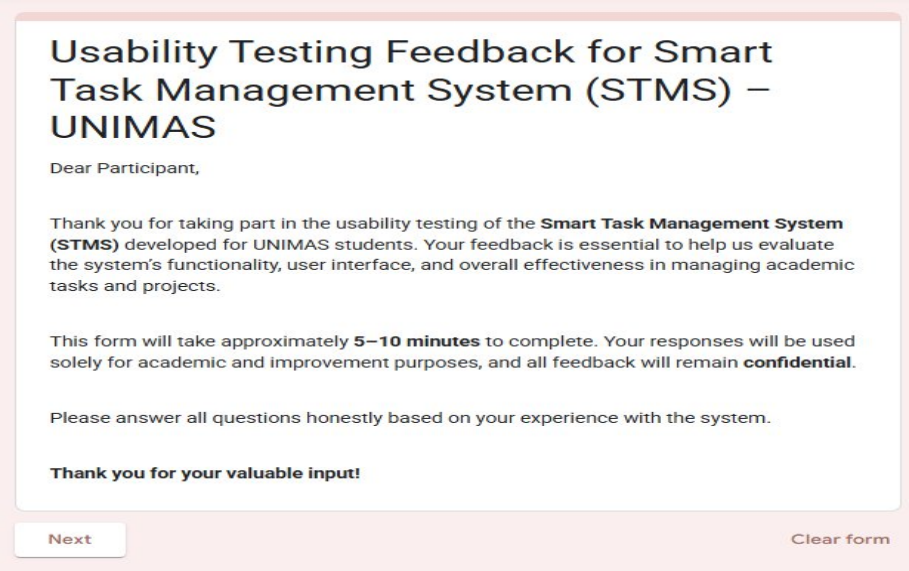


Figure xiii: Question 11

Appendix C: Questionnaire for User Acceptance Testing



Usability Testing Feedback for Smart Task Management System (STMS) – UNIMAS

Dear Participant,

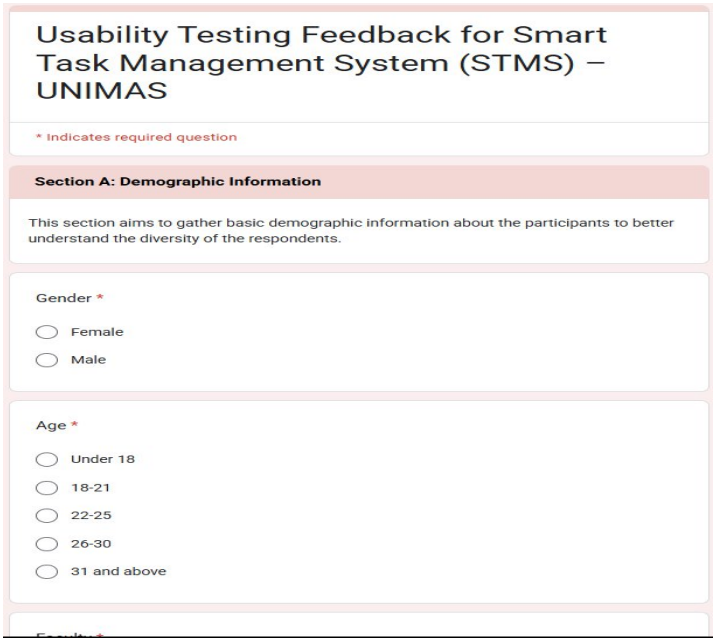
Thank you for taking part in the usability testing of the **Smart Task Management System (STMS)** developed for UNIMAS students. Your feedback is essential to help us evaluate the system's functionality, user interface, and overall effectiveness in managing academic tasks and projects.

This form will take approximately **5–10 minutes** to complete. Your responses will be used solely for academic and improvement purposes, and all feedback will remain **confidential**.

Please answer all questions honestly based on your experience with the system.

Thank you for your valuable input!

Next Clear form



Usability Testing Feedback for Smart Task Management System (STMS) – UNIMAS

* Indicates required question

Section A: Demographic Information

This section aims to gather basic demographic information about the participants to better understand the diversity of the respondents.

Gender *

- Female
- Male

Age *

- Under 18
- 18-21
- 22-25
- 26-30
- 31 and above

- ◯ 18-21
- ◯ 22-25
- ◯ 26-30
- ◯ 31 and above

Faculty *

- ◯ Faculty of Computer Science and Information Technology (FCSIT)
- ◯ Faculty of Economic and Business (FEB)
- ◯ Faculty of Engineering (FENG)
- ◯ Faculty of Applied and Creative Arts (FACA)
- ◯ Faculty of Medicine and Health Sciences (FMHS)
- ◯ Faculty of Resources Science and Technology (FRST)
- ◯ Faculty of Social Sciences and Humanities (FSSH)
- ◯ Faculty of Cognitive Science and Human Development (FCSHD)
- ◯ Faculty of Language and Communication (FLC)
- ◯ Faculty of Built Environment (FBE)
- ◯ Centre For Pre-University Studies (PPPU)

Back Next Clear form

Usability Testing Feedback for Smart Task Management System (STMS) – UNIMAS

* Indicates required question

Section B: Functionality of STMS

Rate the ease of using each function from 1 (Very Difficult) to 5 (Very Easy).

Log In *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Sign Up *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

View Dashboard *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

View Dashboard *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Create New Project *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Add Task *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Assign Task *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Assign Task *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Update Task Progress *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

View Notifications (System + Email) *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

View Productivity Analysis *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Generate Report *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

View Productivity Analysis *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Generate Report *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Logout *

	1	2	3	4	5	
Very Difficult	○	○	○	○	○	Very Easy

Usability Testing Feedback for Smart Task Management System (STMS) – UNIMAS

* Indicates required question

Section C: User Interface and Design

Please rate the following design elements from 1 (Very Poor) to 5 (Excellent).

Layout Clarity *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

Colour Scheme *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

Font Readability *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

Font Readability *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

Navigation Ease *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

Button Placement *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

Responsiveness (Page Speed) *

	1	2	3	4	5	
Very Poor	◯	◯	◯	◯	◯	Excellent

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

Button Placement *

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

Responsiveness (Page Speed) *

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

Overall Visual Aesthetics *

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

Back Next Clear form

Usability Testing Feedback for Smart Task Management System (STMS) – UNIMAS

* Indicates required question

Section D: Overall Evaluation

How would you evaluate the overall experience with STMS from 1 (Very Poor) to 5 (Excellent)?

Ease of Use *

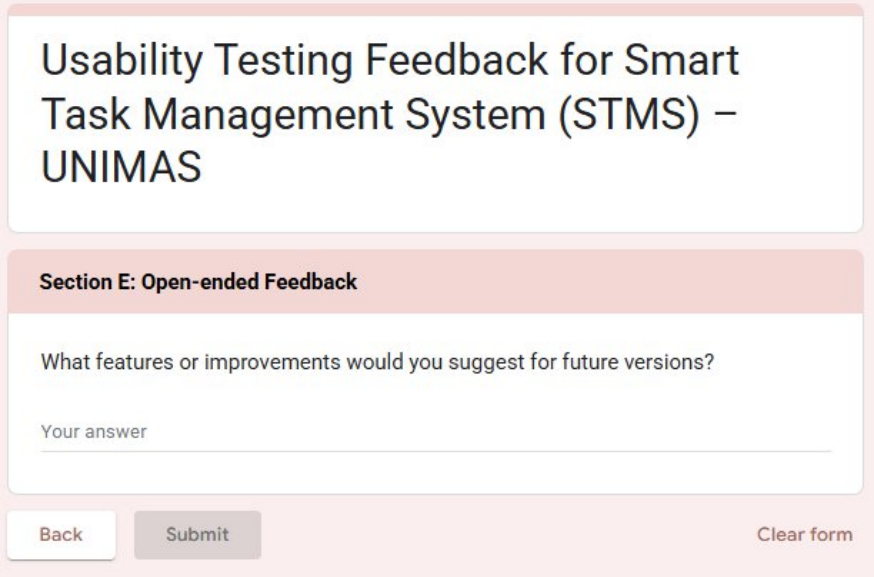
	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

System Efficiency *

	1	2	3	4	5	
Very Poor	○	○	○	○	○	Excellent

User Satisfaction *

1	2	3	4	5


Usability Testing Feedback for Smart Task Management System (STMS) – UNIMAS
Section E: Open-ended Feedback
What features or improvements would you suggest for future versions?
Your answer
Back
Submit
Clear form