Tribhuvan University

Faculty of Humanities and Social Sciences

A PROJECT REPORT ON

TaskMe

Submitted to

Department of Computer Application

Banepa NIST College

In partial fulfillment of the requirements for the Bachelors in Computer Application

Submitted by

Sharad Humagain

Reg no: 6-2-1014-39-2019

Bikash Mainali

Reg no: 6-2-1014-26-2019

November, 2024

Under the Supervision of

Mr. Yubraj Neupane

Tribhuvan University

Faculty of Humanities and Social Sciences

Banepa NIST College

Supervisor's Recommendation

We hereby recommend that this project prepared under my supervision by SHARAD HUMAGAIN and BIKASH MAINALI entitled "**TaskMe"** in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

.....

SIGNATURE

Mr. Yubraj Neupane

SUPERVISOR

Department of Computer

Application Nayabasti, Banepa

Tribhuvan University

Faculty of Humanities and Social Sciences

Banepa NIST College

LETTER OF APPROVAL

This is to certify that this project prepared by SHARAD HUMAGAIN and BIKASH MAINALI entitled "**TaskMe"** in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

Mr. Yubraj Neupane Department of Computer Application Nayabasti, Banepa	Mr. Samish Shrestha Department of Computer Application Nayabasti, Banepa
Internal Examiner	**External Examiner**

ABSTRACT

"**TaskMe**" is a user-friendly web app designed to presents the development of a TaskMe application designed to streamline task management and enhance team collaboration. The application provides an intuitive and organized platform for users to create, assign, track, and prioritize tasks within projects. Inspired by popular project management tools like Trello and Jira, this manager incorporates key features such as progress tracking and individual boards like 'in progress' for effective visual management. The system enables users to monitor deadlines, assign roles, and integrate multiple workflows, making it suitable for individuals, teams, and organizations aiming to improve productivity and project transparency. The project focuses on creating a responsive, user-friendly interface and implementing efficient back-end functionality to handle real-time updates and multi-user interactions.

Keywords: task management, user-friendly interface, real-time updates, team coordination.

ACKNOWLEDGEMENT

The project work presented in this report has been carried out and presented at Banepa NIST College, Faculty of Humanities and Social Sciences Tribhuvan University of Technology as a part of Bachelors of Arts in Computer Application. Project is a test of not only technical skills but also team work and performance under various constraints. This journey cannot be successfully accomplished without help from experts.

Furthermore, we would like to thank our lecturers of the Department of Computer Application for their kindness in sharing their knowledge with us which in different ways has helped us in coming up with this project and being there for us when we needed them, our friends who have always been there to support us and our respondents who gave us feedbacks on improving our project work.

We will be ever grateful to our supervisor **Mr. Yubraj Neupane**, Lecturer without whose guidance, this project would not have become successful.

We are also grateful to our department coordinator **Mr. Samish Shrestha** Finally, our greatest appreciation and love goes to our families, friends and mentors and for sure this would not have happened without their unconditional love, care and support.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

API: Application Programming Interface

CASE: Computer Aided Software Engineering

CSS: Cascading Style Sheets DFD: Data Flow Diagram ER: Entity-Relationship

HTML: HyperText Markup Language

UI: User Interface

Chapter 1: Introduction

1.1 Introduction

Color In the realm of productivity, a task is any piece of work, duty, or responsibility that requires action to achieve a specific outcome. Tasks are the building blocks of projects and personal responsibilities alike, serving as the actionable steps that bring goals and objectives to fruition. Effective task management, the process of overseeing, prioritizing, tracking, and completing these tasks, plays a crucial role in achieving efficiency, productivity, and overall project success. Task management not only involves organizing and scheduling work but also includes communication and collaboration, especially when multiple team members are involved. It helps individuals and teams stay organized, avoid missed deadlines, and focus on high-priority items, ultimately enhancing their ability to meet both personal and organizational goals.

"TaskMe" is a task management application that has been developed to simplify the process of managing tasks and projects, especially for individuals and smaller teams that require a streamlined, easy-to-use solution. Unlike some task management tools designed for large-scale organizations, TaskMe focuses on core functionality and ease of use. It is built to address common issues like disorganization, lack of task prioritization, and communication gaps that can occur in traditional settings. TaskMe provides users with the ability to create and assign tasks, set deadlines, and define priorities, ensuring that all tasks are aligned with project objectives and are completed in a timely manner.

One of TaskMe’s standout features is its built-in communication tool, which allows users to add comments directly to tasks. This eliminates the need for separate messaging platforms and promotes more focused discussions that are directly relevant to each task. Additionally, TaskMe includes notifications and status updates to keep users informed of approaching deadlines and task progress, which helps maintain accountability and keeps everyone aligned. By offering a centralized, accessible platform focused on essential task management functions, TaskMe enhances productivity and ensures that projects remain on track from start to finish. Through intuitive task organization and efficient team collaboration, TaskMe empowers users to achieve better results with minimal complexity, transforming how they manage both individual tasks and collaborative projects.

1.2 Problem Statement

There are lots of drawbacks of i.e. previous systems could not accurately identify color of an object based on an input image. And mainly it requires lots of time i.e. delay in processing. Also, the system available are not so user-friendly and not easily accessible to a wide range of users, including those with limited technical expertise. Additionally, there are less standalone tools that help users who are artists to extract color palettes from the image. Hence, there is need of development of color identifier which could overcome cons of existing systems.

1.3 Objectives

The objectives of project are:

- To create tasks, assign them to team members, and set due dates.
- To prioritize tasks to focus on high-priority items.
- To develop a notification system to alert users of upcoming deadlines, assigned tasks, and important updates

1.4.1 Scope

The scope of TaskMe is to provide a straightforward, user-friendly task management application aimed at improving organization, collaboration, and accountability for individuals and small teams. The application will allow users to create, assign, prioritize, and track tasks within a single platform. Key features include setting deadlines, adding comments for real-time task-specific communication, receiving notifications for important updates, and monitoring task progress. TaskMe focuses on essential task management functions without unnecessary complexity, providing a simple solution for users to stay organized and productive. Advanced project management features beyond task tracking, such as resource management or third-party integrations, are outside the scope of this project.

Additionally, TaskMe aims to foster collaboration by allowing users to easily share tasks with team members, assign roles, and monitor individual progress. The application will support customizable task categories, enabling users to organize tasks based on priorities, project phases, or specific team functions.

1.4.2 Limitation

It has certain limitations due to its focus on simplicity and suitability for smaller teams and individual users. The application does not include advanced project management tools such as resource allocation, Gantt chart-based scheduling, or detailed project timelines, which are typically needed by large organizations managing complex projects. TaskMe is also limited in its integration capabilities, as it does not connect with other enterprise software or third-party applications, which may restrict users who require interoperability across multiple platforms. Additionally, while TaskMe supports basic task analytics, it lacks comprehensive reporting and data visualization features found in more robust project management tools. These limitations define TaskMe's scope as a straightforward and lightweight solution, ideal for users who need streamlined task tracking without the complexities of advanced project management functionalities.

1.5 Development Methodology

The Waterfall development methodology was selected for the TaskMe project due to its structured, sequential approach, which is well-suited for academic projects that require a clear, step-by-step process. This methodology allows for thorough documentation and a systematic progression through each stage of development, making it ideal for projects with fixed requirements and scope, such as TaskMe. In this project, the goal was to design and develop a task management application that serves as a tool to enhance personal productivity and collaboration in small teams, while demonstrating core software development principles.

The project began with the Requirement Analysis phase, where the scope of the TaskMe application was defined. This phase involved determining the specific functionalities that the application would offer, such as task creation, assignment, prioritization, and basic real-time communication through comments. The requirements were gathered based on the project's academic goals, with a focus on simplicity and usability to ensure the application would serve as an effective learning tool for task management. Once the requirements were clearly outlined, the project moved to the System Design phase, where the overall architecture of the application was planned. This included designing the user interface, database structure, and other necessary technical components to support the application's core functionalities.

With the design in place, the development proceeded to the Implementation phase, where

the actual coding of the application took place. During this phase, the features defined in the earlier stages were developed and integrated into the application. The implementation focused on delivering a functional and well-organized task management system that could be used to create tasks, assign them to team members, set deadlines, and monitor progress. Once the development was completed, the Testing phase followed. In this phase, the application underwent rigorous testing to identify any bugs or issues. Functional testing ensured that each feature worked as intended, while performance testing verified that the application performed smoothly under normal use conditions.

Following successful testing, the application entered the Deployment phase. Since this was an academic project, the deployment phase was focused on making the application available for further testing and evaluation by peers and professors. After deployment, the project entered the Maintenance phase, where any issues or bugs identified post-deployment were addressed. This phase also allowed for minor improvements or modifications to enhance the application's performance and usability, based on feedback from testers.

1.6 Report Organization

The report can be organized into 5 chapters which are given below:

Chapter 1 includes introduction includes the brief introduction of the system, statement of problem, objectives, scope and limitation.

Chapter 2 includes background study and literature review includes the previous work related to the systems and similar works were studied and are summarized.

Chapter 3 includes system analysis and design includes different feasibility analysis and designed system architecture, system flow diagram, dataflow diagram.

Chapter 4 includes implementation and testing includes various implementation method and tools and also contains description of testing.

Chapter 5 includes conclusion and future recommendations includes outcomes of the system, conclusion to the system and description about what features can be added in the future.

Chapter 2: Background Study and Literature Review

2.1 Background Study

The rise of digital project management tools has transformed how individuals and teams organize, track, and complete tasks. These tools provide structures for task creation, priority management, and status updates, helping users to streamline work and meet deadlines effectively. Task management software such as TaskMe enhances productivity by allowing users to break down larger goals into manageable subtasks, set priorities, and monitor progress. TaskMe aims to support these needs by incorporating a structured task flow, allowing users to calculate task progress percentages, adjust time estimates based on priorities, and receive status updates that provide a clear view of task advancement. Visual feedback through color-coded progress indicators further helps users quickly gauge progress, making TaskMe particularly useful in fast-paced or team-based settings where tracking multiple tasks is essential. By combining these features with a user-friendly interface, TaskMe addresses common challenges in task management, supporting users in managing time and resources more efficiently.

2.2 Literature Review

The Task management applications play a crucial role in enhancing team productivity by providing tools for collaboration, task assignment, and progress tracking. Popular platforms like Jira, Trello, and Confluence have set the standard for managing tasks, but with varying levels of complexity, customization, and scalability. There is a growing demand for a more intuitive platform that incorporates the best features of these tools while being easy to use for small to medium-sized teams.

Trello is a widely known task management tool that offers a simple, visual approach to organizing tasks through boards and cards. According to Zhang [1], its primary advantage lies in its user-friendly interface, which is ideal for teams needing quick task organization without steep learning curves. However, Green [2] highlights that Trello's limited features for reporting and task dependencies make it unsuitable for large or complex projects.

Jira, in contrast, is known for its robustness and extensive feature set, which is especially useful for software development teams. Thompson and Williams [3] emphasize that Jira's sprint planning, issue tracking, and reporting capabilities are beneficial for large teams

with complex projects. However, its steep learning curve can be challenging for users who are not familiar with Agile methodologies, as noted by Brown [4]. Despite this, Jira's highly customizable nature makes it a go-to solution for large enterprises that need extensive control over their project management processes.

Confluence, often used in tandem with Jira, is a collaboration tool that focuses on documentation and knowledge sharing. It enables teams to create, edit, and collaborate on content in real-time. Adams [5] explains that Confluence’s integration with Jira ensures seamless synchronization between project management and documentation, fostering better communication within teams. However, while Confluence excels in knowledge management, it lacks the task-tracking depth that more task-oriented tools like Jira or Trello offer.

Despite the success of these platforms, there is an increasing need for task management tools that balance simplicity and functionality. Miller and Davis [5] argue that many teams, especially those not involved in software development, require a system that is easy to use but still powerful enough to handle complex workflows. TaskMe aims to address this gap by offering a streamlined, user-friendly interface with customizable task management features and the provision of a real-time collaboration for the users.

Chapter 3: System Analysis and Design

3.1 System Analysis

Requirement collection provides detailed analysis of user requirements, functional & non- functional requirement and system requirement. The front-end is done using HTML & CSS whereas in back-end ReactJS is used for Client side Node.js for Server side. MongoDB is used for Database programming. The system is designed with the series of processes starting with requirement analysis, design, implementation, testing and maintenance. Hence, Waterfall Methodology is used for the development of the system.

3.1.1 Requirement Analysis

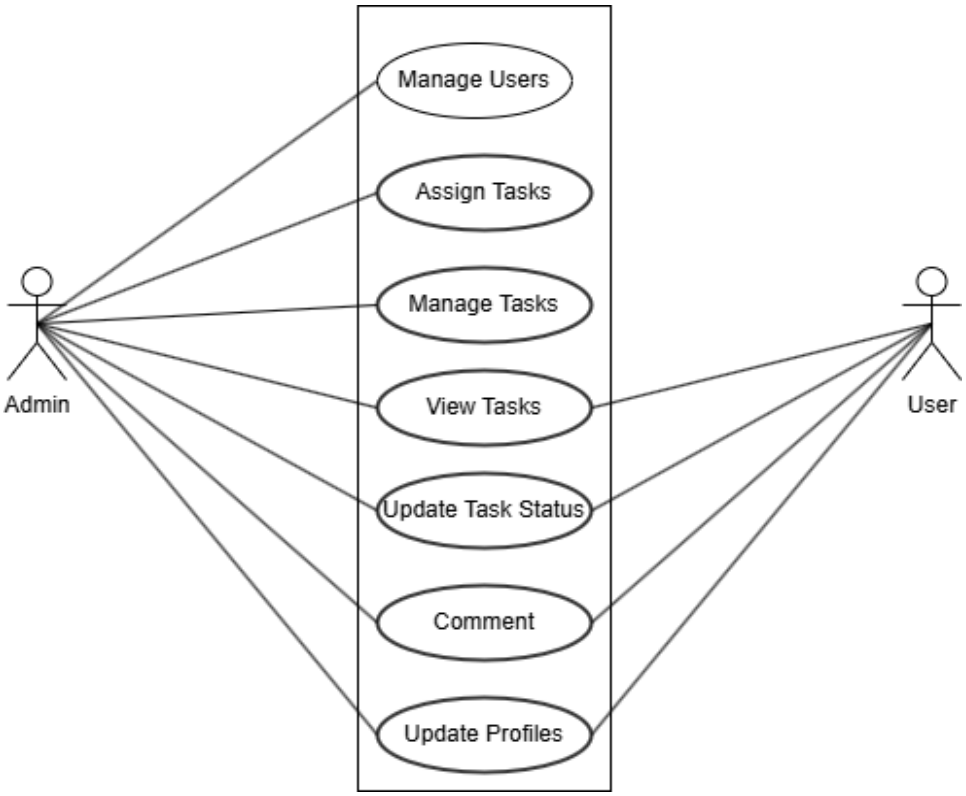


Figure 3.1 Use Case Diagram

i. Functional Requirement:

The use case diagram for the cloud-based task management application outlines the main functionalities available to two primary actors: Admin and User. The Admin has extensive control over the system, including the ability to manage users, assign tasks, and handle various task properties. This role is responsible for creating, editing, activating,

deactivating, or deleting user accounts, as well as assigning tasks to individuals or teams. Additionally, the Admin can update task details, priorities, and statuses to ensure tasks are accurately tracked and organized.

Both Admin and User can view tasks, allowing the Admin to monitor overall task progress while enabling Users to see tasks assigned to them. Users have the ability to update task statuses, marking them as "in progress" or "completed," which supports real-time tracking and communication within the team. Both roles can also add comments to tasks, facilitating collaboration and allowing team members to communicate within the task interface. Finally, both Admins and Users have the ability to update their profiles, including personal information and passwords.

This structured separation of functionalities enables a collaborative, organized approach to task management, ensuring that Admins maintain oversight and control, while Users can efficiently engage with tasks assigned to them within a streamlined system.

ii. Non-functional Requirement:

The system has form-based interface for data entry and stores reports in formatted in a table and for user friendliness. The system has reasonable short time response. The system has good performance as response time is short. The system doesn't crash in middle of process as it is reliable. System can be considered secure as only admin can view user's information.

3.1.2 Feasibility Analysis

i. Technical Feasibility

The project is technically feasible, as it is built with the MERN stack (MongoDB, Express.js, React, and Node.js), which is known for its scalability and ease of maintenance. Since both the backend and frontend are JavaScript-based, any developer familiar with JavaScript can easily manage and update the system. MongoDB's document-oriented database structure is designed to handle large volumes of data efficiently, making it suitable for managing growing amounts of task-related data as the system scales. Additionally, the use of modern security practices, such as JSON Web Tokens (JWT) for secure authentication, ensures that data remains protected, which is particularly important for cloud-based systems. This combination of scalability, maintainability, and security makes the project technically sound and sustainable.

ii. Operational Feasibility

The system is highly feasible, as it provides a responsive and intuitive user interface, thanks to Tailwind CSS and Headless UI. These tools ensure that users can easily interact with the system, regardless of their technical expertise. Since the application is cloud-based and accessible through any internet-enabled device, it is compatible with various platforms and suitable for teams working remotely or across different locations. Additionally, there is no need for specialized training, as the interface is designed to be user-friendly. The adaptability to remote work and the system's ease of use contribute to high acceptance among team members and efficient operational functionality.

iii. Economic Feasibility

Development of the system is highly feasible, with minimal initial costs. Both Node.js and MongoDB are open-source technologies, reducing licensing expenses. Although cloud hosting may involve additional costs, these can be minimized by choosing a cost-effective cloud provider or utilizing free-tier options during the initial stages. In the long term, the system could result in cost savings by automating task management, reducing manual labor associated with project coordination, and minimizing delays. Additionally, as users are already familiar with standard UI elements, no additional training costs are necessary. The system is thus cost-effective both in development and in its potential to enhance productivity.

iv. Schedule Feasibility

The time required to complete the project is calculated and classified using the following Gantt Chart:

Time / Phase	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Study And Analysis	2W											
Data Fetching & Scrapping				1W								
Implementation & Coding			3W									
Testing				3W								
Documentation		9W										
Review											2W	
Presentation						1W						1W

Figure 3.2 Gantt Chart

3.1.3 Object Modeling using class and Object Diagrams

Object modeling is a crucial step in software development, where the system's entities, their attributes, and relationships are represented using class and object diagrams. For the Task Management System, we will create simplified class and object diagrams.

Class Diagram: A class diagram represents the static structure of the system, depicting the classes, their attributes, and the relationships between them. Here are some key classes for the Task Management System. This diagram specifically illustrates the relationships and attributes of the main classes in the TaskMe tool, giving an overview of how they interact within the system.

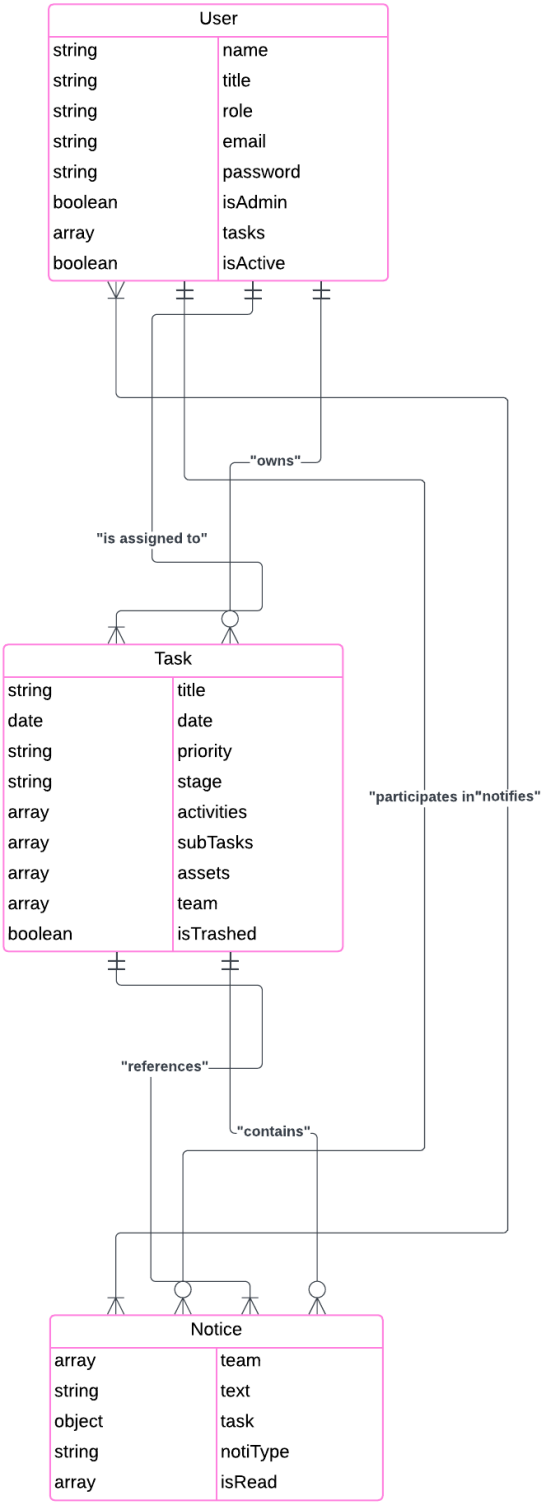


Figure 3.3 Class Diagram

3.1.4. Process Modeling using Activity Diagrams

Activity diagrams are a type of behavior diagram in UML (Unified Modeling Language) used to model the flow of activities and actions in a system. They are particularly useful for visualizing the steps, decision points, and parallel activities within a process. For the TaskMe System, we can create an activity diagram to illustrate the typical workflow involved in property listing and transaction handling.

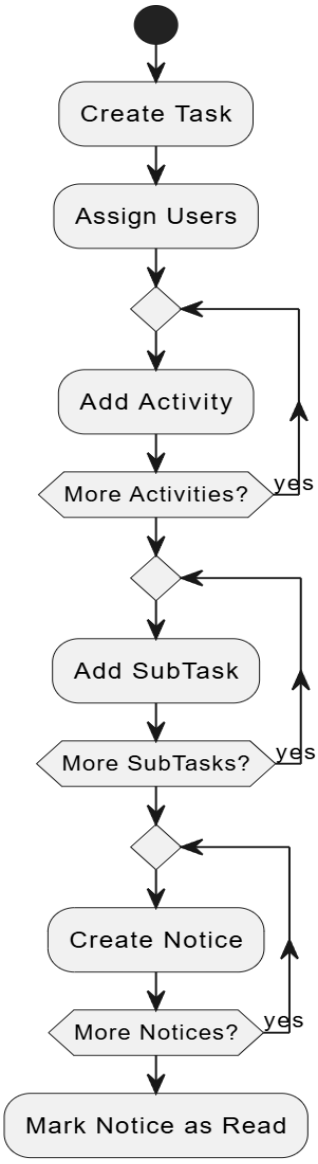


Figure 3.4 Activity Diagram

3.2. System Design

3.2.1 Refinement of Class, Object, State, Sequence, and Activity Diagrams

The refinement of UML diagrams involves adding more details and refining the representation to provide a comprehensive view of the system. Let's briefly discuss the refinement process for each type of UML diagram:

i. Class Diagram Refinement:

Add additional classes and their attributes to represent a more complete set of entities and their relationships in the TaskMe System.

Specify access levels (e.g., public, private, protected) for attributes and methods to enforce encapsulation and data hiding.

Include associations, aggregations, and compositions to illustrate the relationships between classes more precisely.

Add multiplicity notations to associations to indicate the cardinality between classes (e.g., one-to-one, one-to-many).

Introduce inheritance (generalization and specialization) to represent hierarchical relationships between classes.

ii. Object Diagram Refinement:

Include more instances (objects) of classes in the diagram to depict a snapshot of the system's state at a specific moment.

Add object interactions, such as method invocations, to show how objects collaborate and communicate during a particular scenario.

iii. State Diagram Refinement:

Specify detailed states and transitions for each state to depict the complete lifecycle of an object or a system.

Include actions, events, and conditions that trigger state transitions to provide a more comprehensive view of the system's behavior.

Incorporate nested states and concurrent states to model complex behaviors and concurrent activities.

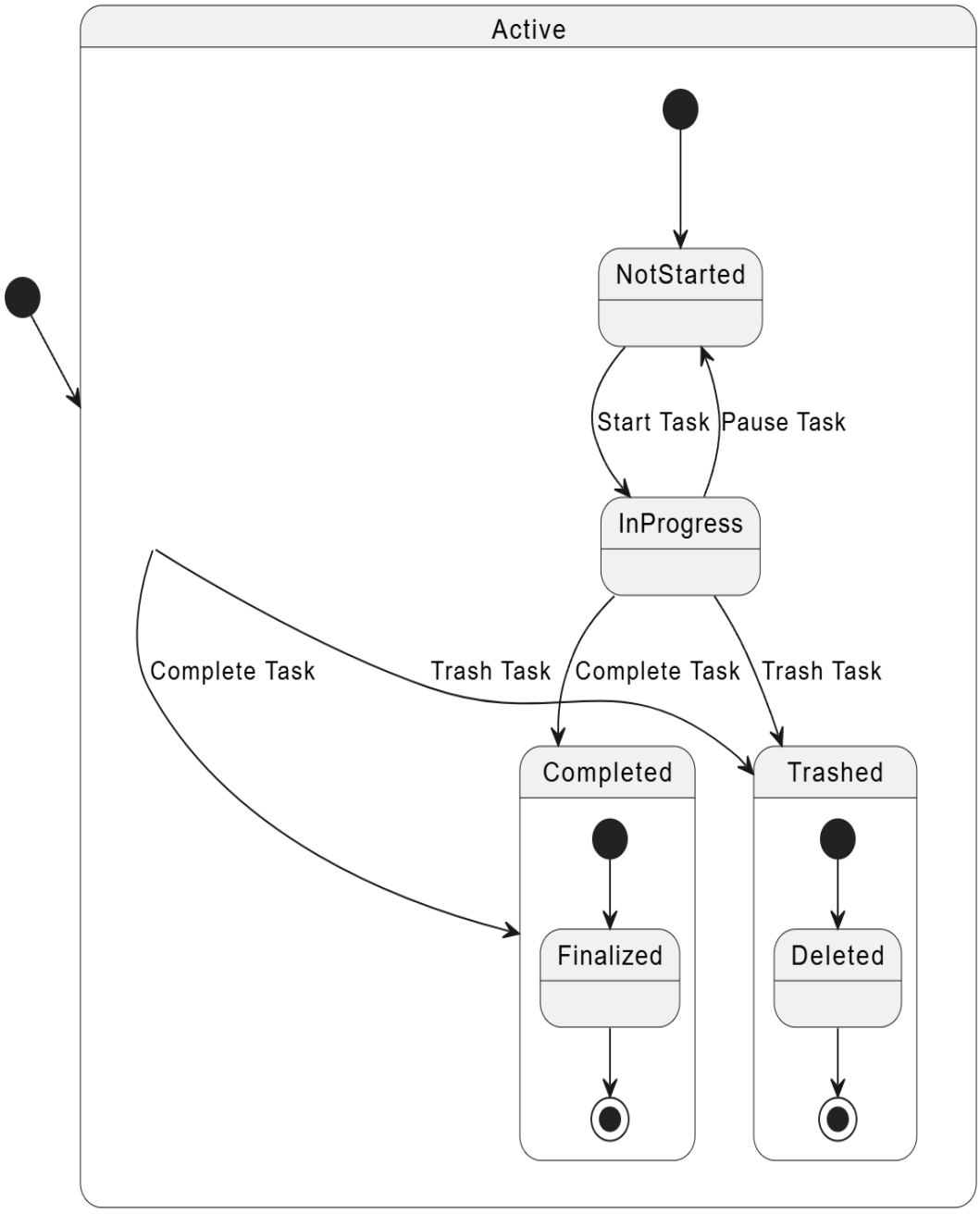


Figure 3.5 State Diagram

iv. Sequence Diagram Refinement:

Add more lifelines (representing objects) and messages to illustrate the sequence of interactions between objects in various scenarios. Include return messages and loop constructs to show iterative processes or recursive calls. Represent conditions, branches, and alternative paths to account for different decision points in the sequence of interactions.

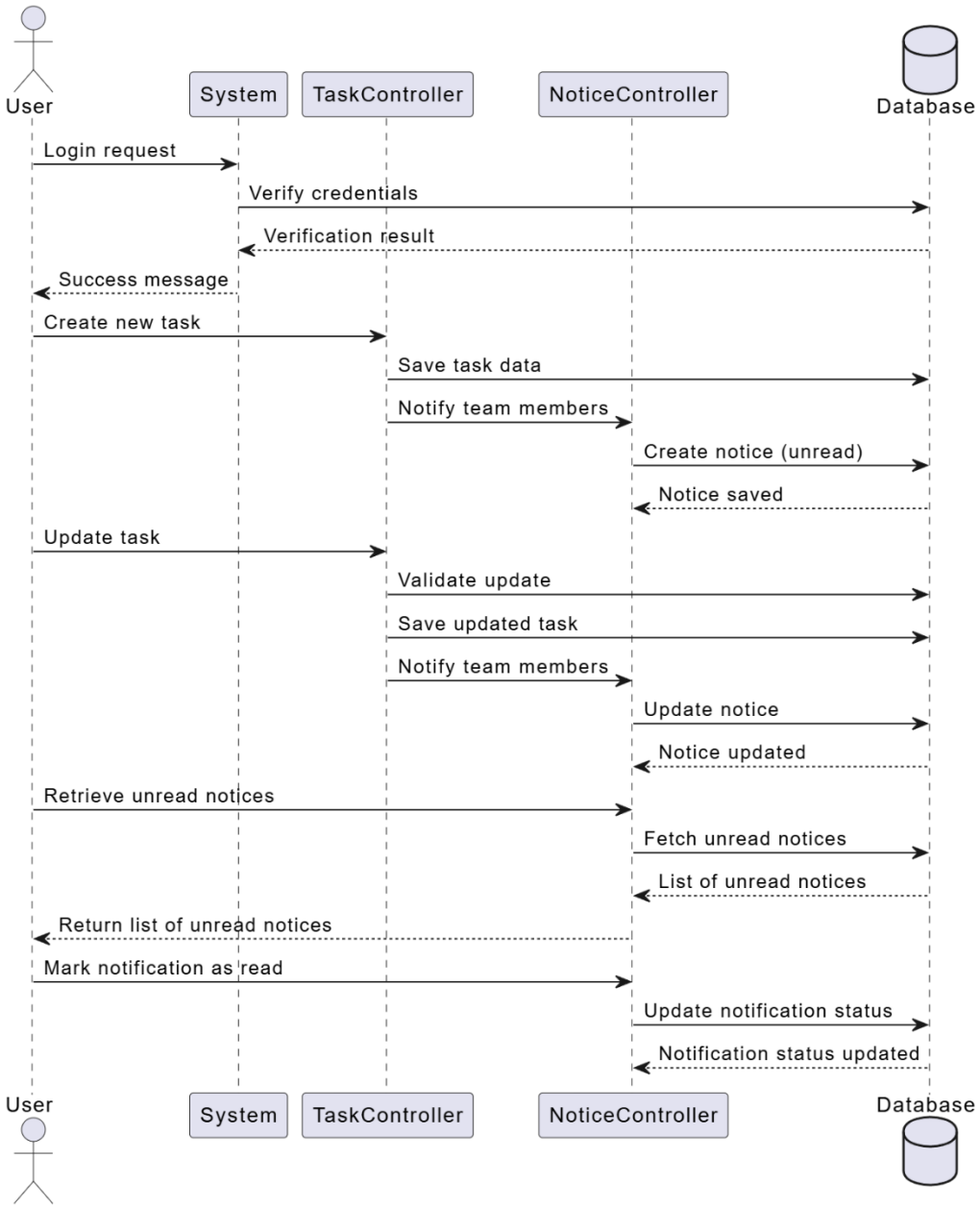


Figure 3.6 Sequence Diagram

v. Activity Diagram Refinement:

Specify additional activities, sub activities, and decisions to model more complex processes and workflows within the TaskMe. Add more detail to actions and control nodes to provide a more comprehensive understanding of the flow of activities. Incorporate swimlanes to show the involvement of different actors or roles in the activity flow.

3.2.2 Component Diagrams

A component diagram in UML (Unified Modeling Language) is used to represent the physical and logical components of a system and their relationships. It provides a high-level view of the system's architecture and helps in understanding the modular structure and dependencies among components.The component diagram is shown below:

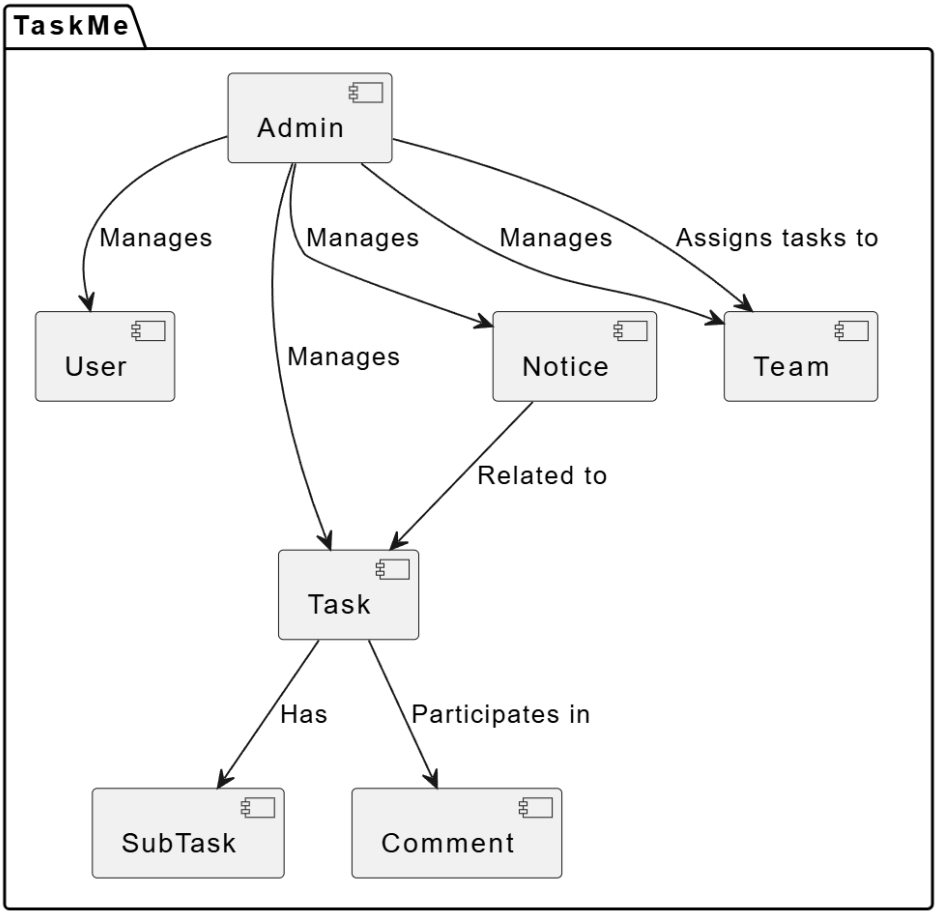


Figure 3.7 Component Diagram

3.2.3 Deployment Diagrams

A deployment diagram in UML (Unified Modeling Language) is used to represent the physical deployment of software components and their relationships on hardware devices or nodes. It provides a high-level view of how the system's components are distributed across different hardware nodes in a real-world environment.

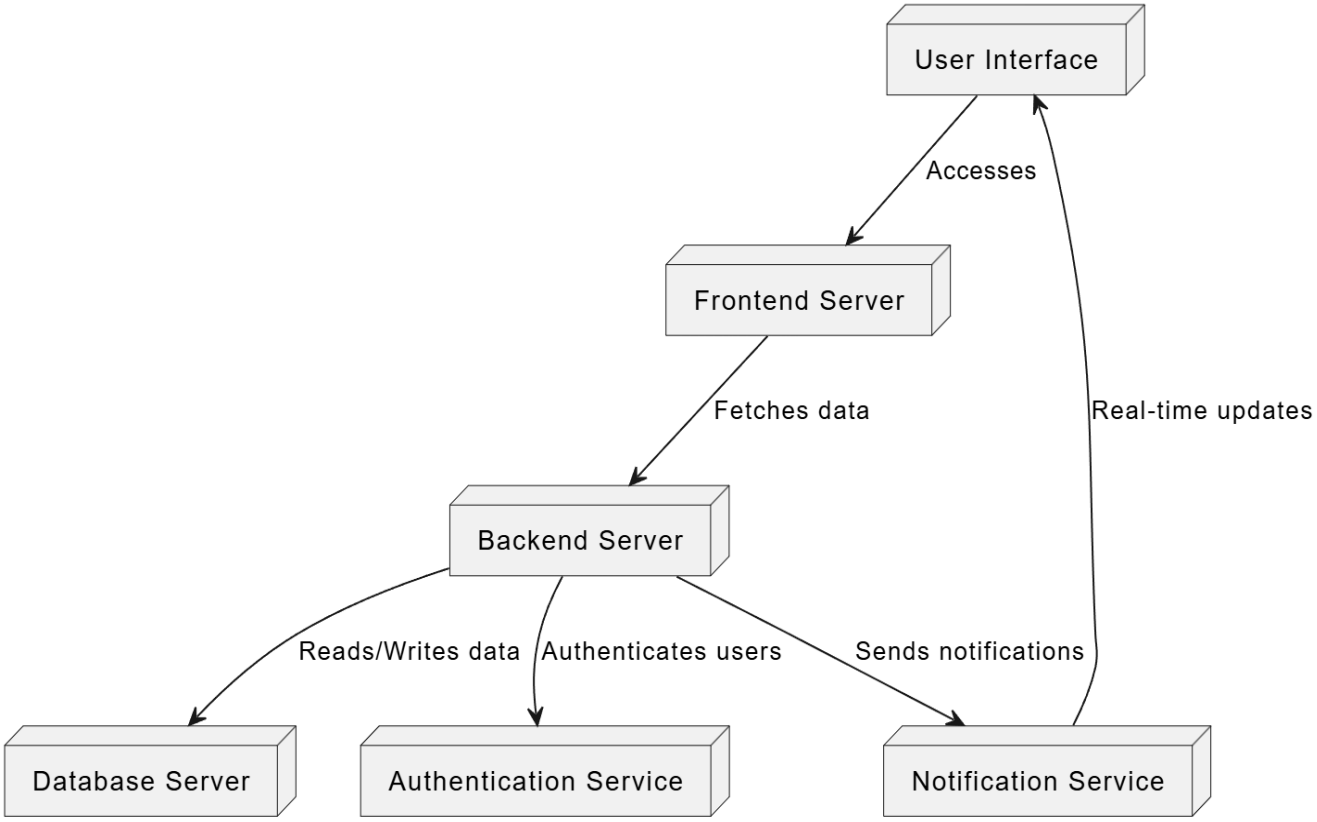


Figure 3.7 Deployment Diagrams

3.3 Algorithm Details

Task Progress Tracking Algorithm

The The Task Progress Tracking Algorithm is a key feature in the task management web app that calculates and estimates the completion progress of tasks based on their subtasks, priority, and start date. This algorithm helps users track their tasks efficiently and provides an estimated time to completion.

The algorithm works by calculating the progress percentage based on completed subtasks, estimating the remaining time until task completion, and providing a status update based on the current progress. It takes into account the task's subtasks, priority level, and time elapsed since the task's start date. The time complexity of the algorithm is O(N), where N represents the number of subtasks within a task. This is because the algorithm needs to iterate through all the subtasks to calculate the completion percentage and estimate the remaining time. In most cases, the algorithm is highly efficient, as it performs calculations based on the existing data without needing to reprocess large amounts of information multiple times. One of the most valuable aspects of the algorithm is its ability to track tasks in real-time. As users complete subtasks, the algorithm dynamically updates the task's progress, offering immediate insights into the overall task’s completion. By estimating the time left to complete a task, the algorithm helps users better manage their time. With clear insights into how much work remains and when they can expect the task to be completed, users can plan their schedules more effectively.

Steps:

Step 1: Calculate Progress Percentage

Determine Total Subtasks: The algorithm first checks how many subtasks are assigned to the task. Subtasks are individual actionable steps that must be completed for the main task to be considered finished.

Track Completed Subtasks: It then checks how many of these subtasks have been completed. Subtasks can have statuses such as "not started," "in progress," or "completed." The completed subtasks are counted towards progress.

Progress Percentage: The progress percentage is calculated as the ratio of completed subtasks to the total subtasks. The formula is:

Progress Percentage= (Completed Subtasks / Total Subtasks)×100

- If all subtasks are completed, the progress percentage is 100%.
- If no subtasks are completed, the progress percentage is 0%.
- If halfway through, the progress would be around 50%.

This percentage gives a direct measure of how much of the task has been completed.

Step 2: Estimate Remaining Time

Once the progress is known, the algorithm estimates how much time is left for the task to be completed. This is based on the number of subtasks left to complete, as well as other factors like the task's priority.

Base Time Estimate: When a task is created, it often has a base estimate of how long it will take to complete, which could be based on an average time for each subtask or a pre-defined estimate by the user.

Priority Adjustments:

- High priority tasks typically get completed faster. If the task has a high priority, the algorithm will reduce the estimated time.
- Medium priority tasks take normal time to complete.
- Low priority tasks may take longer to complete, so the algorithm adjusts the time by increasing the estimated duration.

Estimated Time Left Based on Progress:

The algorithm takes into account how much time has already passed and the current rate of progress.If a task has already made progress, the algorithm uses the current rate of progress to estimate how much more time is needed to finish the remaining subtasks.

Step 3: Update Task Status

The algorithm also updates the status of the task based on the progress and estimated completion time. This helps the user understand how close the task is to being finished.

Completed: If the task has reached 100% progress, it is marked as completed.

On Track: If the task is progressing well (e.g., it's about halfway done and the due date is still far off), the status is marked as "On Track."

Behind Schedule: If the task is far behind (e.g., only 10% completed and the due date is

coming soon), the status is marked as "Behind Schedule."

Critical: If the task is nearing its due date but has not made enough progress (e.g., only 20% done with less than 1 day left), the task is marked as Critical and the user is notified to take immediate action.

Step 4: Display Task Progress

To give users a visual understanding of the task's progress, the algorithm calculates the progress bar or indicator that will be displayed in the web app interface:

Visual Indicators: The visual indicator, such as a progress bar or colored circles, is updated in real time as the task's progress percentage increases.

Color Coding: The progress bar can also change colors based on progress:

- Red (0-25%): Indicates the task is barely started or behind schedule.
- Yellow (26-50%): The task is progressing, but still needs a lot of work.
- Blue (51-75%): The task is making good progress.
- Green (76-100%): The task is almost complete or already finished.

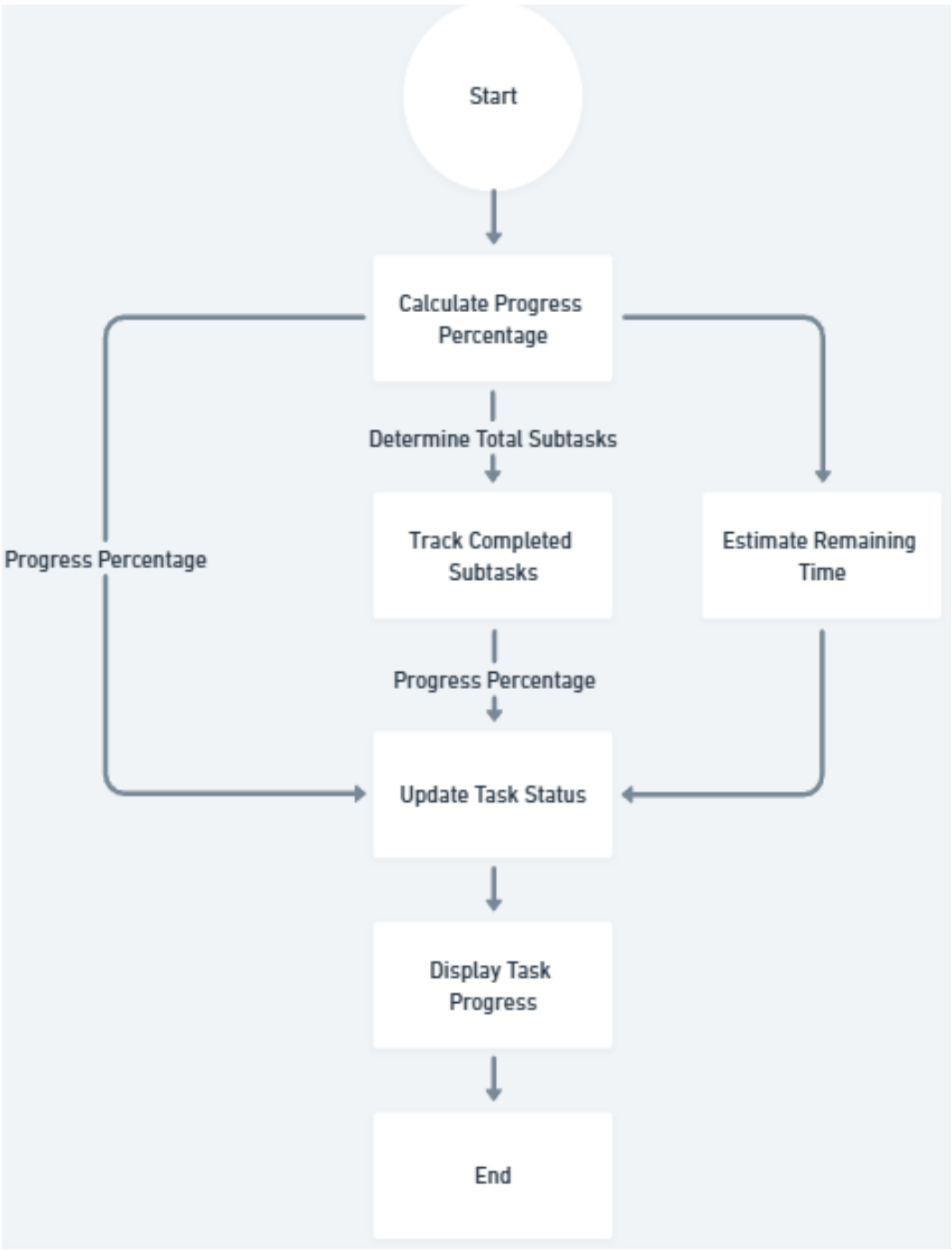


Figure 3.8 Algorithm Flowchart

Chapter 4: Implementation and Testing

4.1 Implementation

The implementation phase is where the actual development of the system begins, translating the design specifications into a working system. During this phase, developers write the code based on the system architecture and design documents created earlier in the project. The focus is on translating theoretical designs into functional components using appropriate programming languages, frameworks, and tools. Once the system is coded, it undergoes rigorous systems testing to identify bugs, verify the system's functionality, and ensure that it meets the requirements laid out during the design phase. This phase also includes unit testing, integration testing, and system testing to validate all components work together as expected.

4.1.1 Tools Used (CASE tools, Programming languages, Database platforms)

4.1.2 Diagramming tools such as draw.io were used for graphical representation of the data and systems. These were used to make flowcharts, component diagram, sequence diagram, state diagram, activity diagram, class diagram, Gantt Charts, etc. MSWord was used for the documentation. Visual Studio Code was used to write, edit and compile the codes. ReactJS, Node.js, CSS, HTML was used to build the web app. Express as a server-side framework was used to host the web app. MongoDB was used to design the database for data storage.

4.1.3 Implementation Details of Modules

Admin Dashboard Panel: Admin after logging in can edit or delete user's details or the users themselves through admin dashboard, and assign tasks to the users.

User Registration: Users have to register into the system through Admin before they can view their tasks. User need to verify email for completing registration process.

View Profile: Any user logged in into the system can see their profile and can make changes to update it.

Task Log: The Task Management Module is at the heart of the application. It allows admin to create, assign, update, and delete tasks. While give access to users on their own task which they are assigned to. It also integrates the concept of subtasks and priorities.

4.2 Testing

4.2.1 Test Cases for Unit Testing

The system comprises of registration form, login form, admin dashboard, color palette extraction dashboard and for each case unit testing is done.

Table 4.1 Login Functionality

Test Case ID	Test Case	Steps	Expected Result
TC-L-U-01	Valid Login	1. Enter valid username 2. Enter valid password 3. Click login button	User is successfully logged in and redirected to dashboard
TC-L-U-02	Invalid Username	1. Enter invalid username 2. Enter valid password 3. Click login button	Error message "Invalid credentials" is displayed
TC-L-U-03	Invalid Password	1. Enter valid username 2. Enter invalid password 3. Click login button	Error message "Invalid credentials" is displayed
TC-L-U-04	Empty Fields	1. Leave username field empty 2. Leave password field empty 3. Click login button	Error message "Please fill in all fields" is displayed

Table 4.2 Dashboard Functionality

Test Case ID	Test Case	Steps	Expected Result
TC-D-U-01	Load User Tasks	1. Log in as a user 2. Navigate to dashboard	All user's tasks are displayed correctly
TC-D-U-02	Task Count	1. Log in as a user 2. Check task count on dashboard	Task count matches the number of user's tasks

Table 4.3 Team Member Management

Test Case ID	Test Case	Steps	Expected Result
TC-TM-U-01	Add Team Member	1. Enter valid team member details 2. Click the add button	Team member is successfully added to the system
TC-TM-U-02	Edit Team Member	1. Select a team member 2. Edit details 3. Save changes	Team member details are updated successfully
TC-TM-U-03	Delete Team Member	1. Select a team member 2. Click the delete button 3. Confirm deletion	Team member is removed from the system

Table 4.4: Testing Password Change

Test Case ID	Test Case	Steps	Expected Result
TC-PC-U-01	Change Password	1. Enter current password 2. Enter new password 3. Confirm new password 4. Submit change	Password is updated successfully
TC-PC-U-02	Incorrect Password Entry	1. Enter incorrect current password 2. Enter new password 3. Confirm new password 4. Submit change	Error message "Current password is incorrect"
TC-PC-U-03	Confirm New Password	1. Enter current password 2. Enter new password 3. Enter a different password in the confirm field 4. Submit change	Error message "New passwords do not match" is displayed

Table 4.5: Testing Task Board

Test Case ID	Test Case	Steps	Expected Result
TC-TB-U-01	Add Task to Todo	1. Click 'Add Task' in Todo column 2. Enter task details 3. Save task	Task is added successfully to Todo column
TC-TB-U-02	Move Task to Progress	1. Select a task in Todo 2. Drag to Progress column	Task is moved successfully to Progress column
TC-TB-U-03	Move Task to Completed	1. Select a task in Progress 2. Drag to Completed column	Task is moved successfully to Completed column

Table 4.6: Trash and Restore

Test Case ID	Test Case	Steps	Expected Result
TC-TR-U-01	Move Task to Trash	1. Select a task 2. Click delete/trash button	Task is moved to trash and not visible in main task list
TC-TR-U-02	Restore Task from Trash	1. Go to trash 2. Select a task 3. Click restore button	Task is restored and visible in main task list
TC-TR-U-03	Permanent Deletion	1. Go to trash 2. Select a task 3. Click permanent delete button	Task is permanently deleted and not recoverable

4.1.1 Test Cases for System Testing

After unit testing, the whole integrated system is tested for security, usability, regression, recovery and migration.

Table 4.7: Login Form

Test Case ID	Test Case	Steps	Expected Result
TC-L-S-01	Admin Login	1. Enter valid admin credentials 2. Click login button	Admin is logged in and redirected to admin dashboard
TC-L-S-02	User Login	1. Enter valid user credentials 2. Click login button	User is logged in and redirected to user dashboard
TC-L-S-03	Session Management	1. Log in successfully 2. Close browser 3. Reopen browser and TaskMe	User remains logged in (session persists)

Table 4.8: Dashboard

Test Case ID	Test Case	Steps	Expected Result
TC-D-S-01	Dashboard Navigation	1. Log in 2. Click on different dashboard sections	All sections are accessible and load correctly
TC-D-S-02	Dashboard Refresh	1. Log in 2. Add a new task 3. Return to dashboard	Dashboard updates to show the new task

Table 4.9: Team Addition

Test Case ID	Test Case	Steps	Expected Result
TC-TM-S-01	Team List Update	1. Add a new team member 2. Check team list	New team member appears in the team list
TC-TM-S-02	Team Member Permissions	1. Add a new team member with specific permissions 2. Log in as the new team member 3. Check assigned permissions	New team member has access according to assigned permissions

Table 4.10: Password Change

Test Case ID	Test Case	Steps	Expected Result
TC-PC-S-01	Login with New Password	1. Change password successfully 2. Log out 3. Log in with new password	User can log in with the new password
TC-PC-S-02	Password Change Notification	1. Change password successfully 2. Check user's email	User receives an email notification about the password change

Table 4.11: Trash & Restoration

Test Case ID	Test Case	Steps	Expected Result
TC-TR-S-01	Bulk Trash Operations	1. Select multiple tasks 2. Move to trash 3. Restore multiple tasks	All operations are performed correctly and reflected in UI
TC-TR-S-02	Trash Retention Policy	1. Move a task to trash 2. Wait for retention period 3. Check trash	Task is automatically deleted after retention period

4.3 Result Analysis

Result analysis is a critical step in the testing process that involves reviewing the outcomes of the executed test cases and identifying any issues or discrepancies found during testing. It helps in assessing the quality and readiness of the TaskMe for deployment.

Test Case Status: Review the status of each test case . Identify the number of test cases that passed successfully, those that failed, and any test cases that couldn't be executed due

to blockers or dependencies.

Defects and Issues: Record and categorize the defects and issues identified during testing. These could include functional bugs, user interface glitches, performance bottlenecks, security vulnerabilities, or other anomalies.

Severity and Priority: Assign severity and priority levels to the identified defects. Severity indicates the impact of the defect on the system's functionality, while priority reflects the order in which the defects need to be fixed.

Root Cause Analysis: Analyze the root causes of the defects and issues found. Understanding the underlying reasons helps in addressing them effectively and preventing similar issues in the future.

Regression Testing: If any defects were fixed during the testing phase, perform regression testing to ensure that the changes did not introduce new issues and that the system's existing functionality remains intact.

Coverage Analysis: Evaluate the coverage of testing, including the number of test cases executed, the percentage of code covered.

Chapter 5: Conclusion and Future Recommendations

5.1 Conclusion

The TaskMe application successfully achieves its goal of helping users manage tasks effectively by tracking progress, estimating completion time, and updating task statuses. Through features like percentage-based progress calculation, time estimation based on priority, and real-time visual indicators, the application provides a comprehensive view of task status, aiding users in staying on track and meeting deadlines. The integration of notifications and priority adjustments further enhances user productivity and ensures that important tasks receive the attention they require. Overall, the TaskMe application brings organization and clarity to task management, empowering users to handle their responsibilities more efficiently.

5.2 Future Recommendations

For further enhance the TaskMe application, several key features could be implemented to provide a more robust and versatile experience. First, adding automated reminders for upcoming task deadlines would improve time management by notifying users through emails or push notifications, helping them stay organized and reducing the likelihood of overdue tasks. Integrating detailed analytics could offer users insights into their productivity trends, such as average task completion times, productivity peaks, and task categories that consume the most time. By visualizing this data, users can identify areas for improvement and better allocate their time. Additionally, integrating TaskMe with other popular productivity tools, such as Google Calendar, Microsoft Outlook, Slack, or Microsoft Teams, would streamline workflows by allowing users to manage tasks across platforms. This integration would make the application a more cohesive part of users' daily routines, especially for those working in collaborative environments.

Moreover, incorporating AI-driven features could elevate TaskMe’s personalization and efficiency. For instance, AI could analyze user behavior to recommend priority adjustments, suggest optimal task completion times, or even identify patterns that help users optimize their work habits.

References

[1] J. Zhang, "User Experience and Design of Task Management Applications: A Case Study of Trello," *International Journal of Human-Computer Interaction*, vol. 37, no. 4, pp. 280-289, 2022.

[2] L. Green, "Limitations of Trello in Complex Project Management," *Software Development Review*, vol. 22, no. 6, pp. 93-101, 2023.

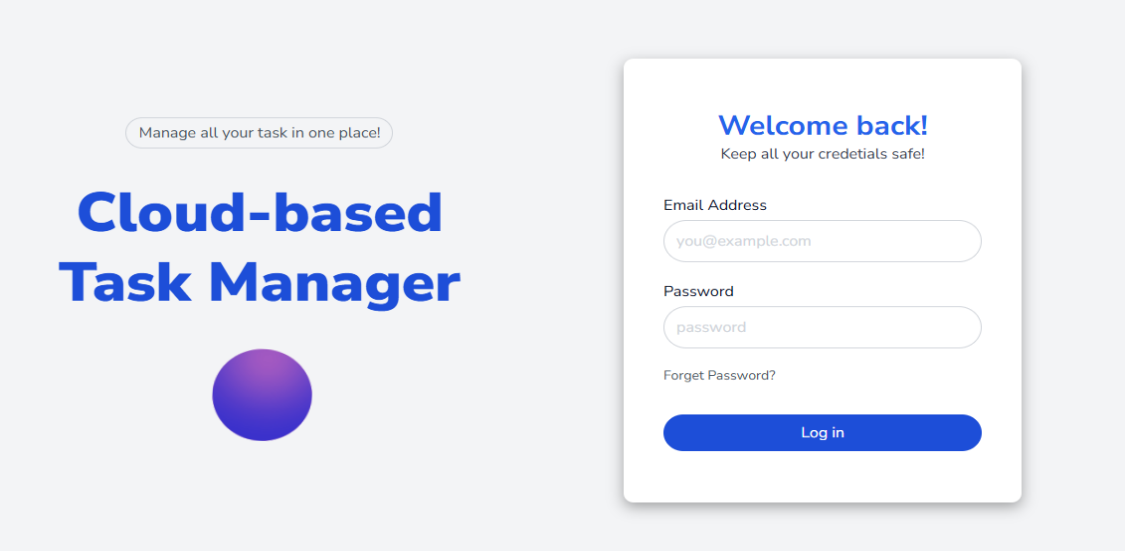
[3] M. Thompson and P. Williams, "Jira as a Tool for Agile Project Management," *Journal of Software Engineering*, vol. 19, no. 2, pp. 245-258, 2021.

[4] H. Brown, "Challenges of Using Jira for Non-Software Projects," *Project Management Journal*, vol. 18, no. 1, pp. 88-96, 2020.

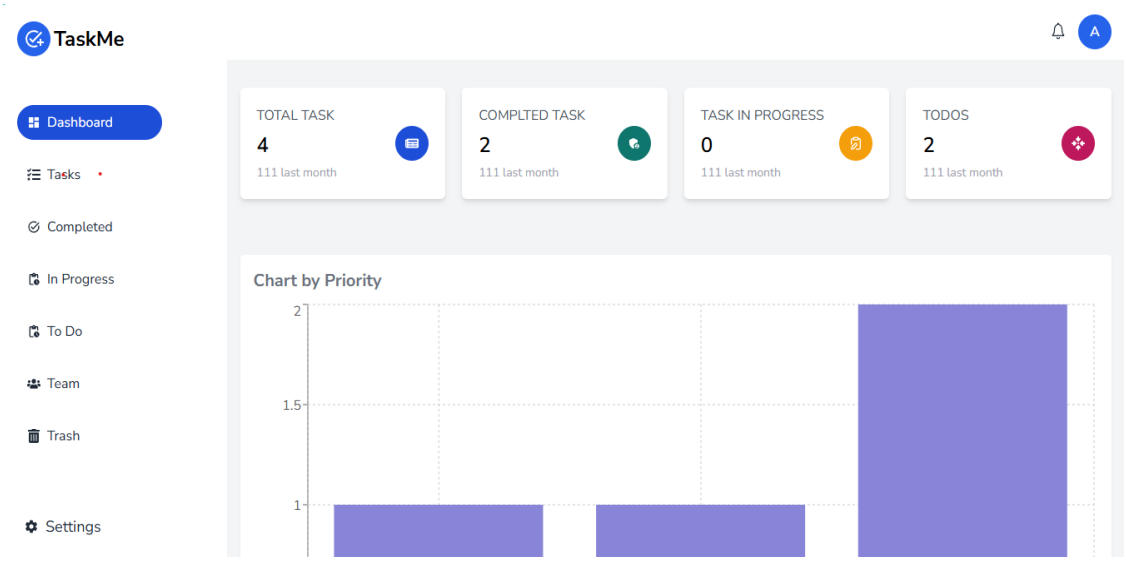
[5] K. Miller and A. Davis, "Task Management Tools for Modern Teams: Finding the Balance Between Simplicity and Functionality," *Journal of Digital Tools*, vol. 10, no. 7, pp. 102-113, 2022.

Appendices

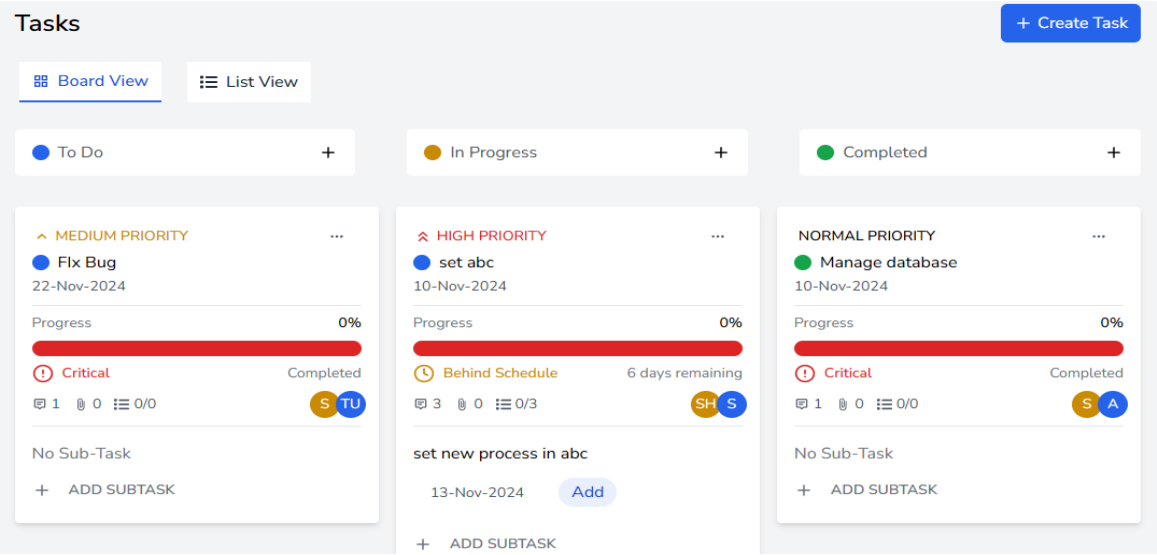
Login Page



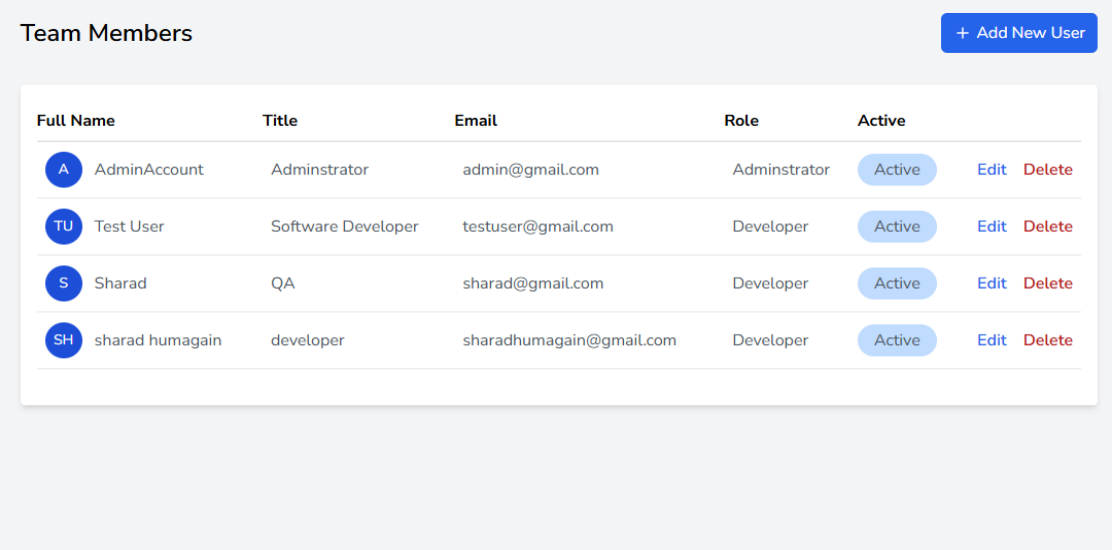
Admin Dashboard



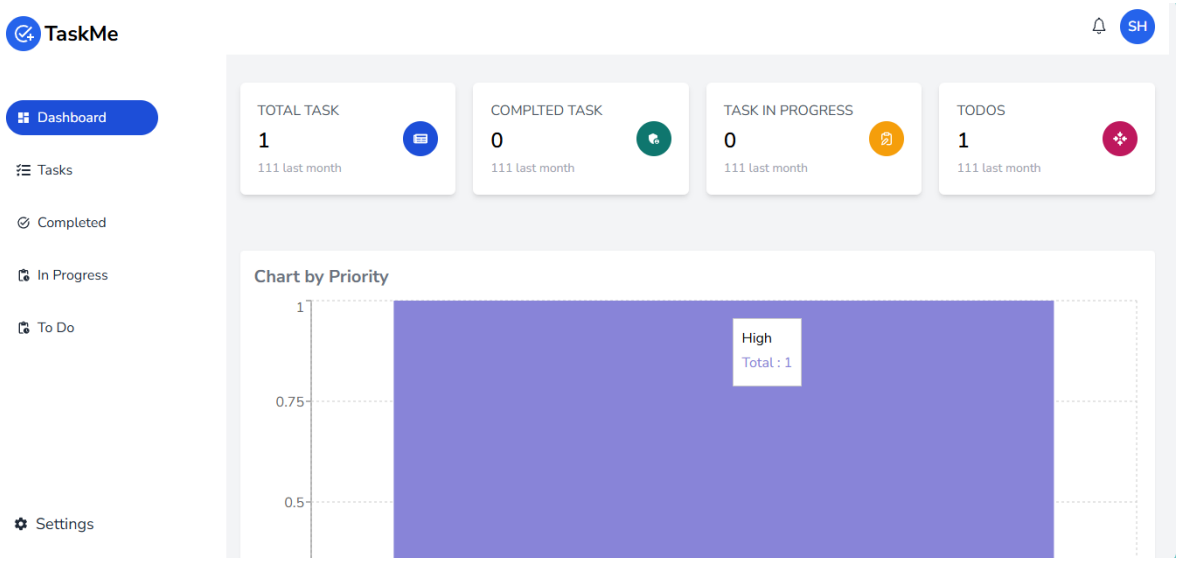
Task Dashboard



Admin Team Control



User Dashboard



List View of Task