MINOR PROJECT MID DEFENSE PROPOSAL FOR THE DEGREE OF BACHLEOR OF COMPUTER ENGINEERING

TASK UNITY

[Course Code: PRJ 300]

Submitted by:

NISHANI KUMARI RAI	**[21070518]**
SAMAR BHATTARAI	**[21070531]**
SARAJ DHAKAL	**[21070536]**
SOVIYAT LAMSAL	**[21070541]**

UNITED TECHNICAL COLLEGE

Faculty of Science and Technology

Affiliated to Pokhara University

June, 2024

ABSTRACT

Task Unity, an innovative project collaboration platform, tackles the challenges of fragmented communication, task tracking and collaboration inefficiencies by leveraging MongoDB for flexibility and scalability. This centralized hub simplifies communication, task management and file sharing, enhancing team efficiency and fostering collaboration. With a user-friendly interface, dynamic data handling and scalability, Task Unity adapts to evolving project needs in industries like software development, marketing and construction. The Level 0 Data Flow Diagram provides a top-level view, featuring major processes, external entities and the Project Database. This proposal explores Task Unity's features, real-world case studies and its potential to reshape project collaboration, promising streamlined communication, efficient task management and successful project outcomes.

Keywords: Task Unity, Collaboration Platform, MongoDB, Task Management, Scalability, User-friendly interface, Project Outcomes

ABBREVIATION AND ACRONYM

CSS	: Cascading Style Sheets
DFD	: Data Flow Diagram
ERD	: Entity Relationship Diagram
HTML	: Hypertext Markup Language
HTTP	: Hypertext Transfer Protocol
JS	: JavaScript
RAM	: Random Access Memory
RDBMS	: Relational Database Management System
SQL	: Structured Query Language
NoSQL	: Not Only Structured Query Language
USD	: Use Case Diagram
VS Code	: Visual Studio Code
XML	: Extensible Markup Language

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Background

In recent years, the dynamic landscape of project management has undergone significant transformations, spurred by the evolution of collaborative work environments and the increasing complexity of tasks undertaken by teams across various industries. Traditional methods of communication and task management, once sufficient for simpler projects, are now proving inadequate in the face of intricate, multifaceted assignments. The emergence of dispersed teams, often working across different time zones, further accentuates the challenges associated with conventional collaboration tools. [1]

Recognizing the pressing need for a more sophisticated and streamlined approach to project collaboration, Task Unity comes to the forefront as a comprehensive solution. This platform is conceived in response to the growing intricacies of modern projects, aiming to bridge communication gaps, enhance task visibility and foster a cohesive working environment. As projects expand in scale and complexity, Task Unity strives to empower teams by providing an integrated space where communication, task management and collaboration seamlessly converge. By addressing the shortcomings of traditional methods, Task Unity endeavors to elevate project collaboration to new heights, ensuring that teams can navigate the intricacies of contemporary project management with efficiency and ease. [2]

1.2 Problem of Statement

Task Unity emerges as a solution to the pervasive challenges encountered by project teams in the realms of communication, task tracking and overall collaboration. One of the primary pains points it tackles is the issue of fragmented communication, which commonly arises in projects involving dispersed teams or those spanning different departments. Traditional communication methods, such as scattered emails or disjointed messaging apps, often lead to information silos, hindering a cohesive understanding of project progress and goals. Task Unity endeavors to streamline communication by centralizing discussions and updates within the platform, reducing the risk of misunderstandings and ensuring that all team members are on the same page.

Furthermore, the platform addresses the complexities associated with task tracking. In many instances, teams struggle to monitor and manage tasks effectively, leading to missed deadlines and project delays. Task Unity introduces robust task management features, allowing users to assign, track and update tasks in real-time. This not only enhances visibility into individual responsibilities but also contributes to a more organized and accountable workflow. By mitigating the challenges of task tracking, Task Unity aims to empower project teams to meet deadlines with precision and efficiency.

Collaboration inefficiencies often arise from the use of inadequate tools that fail to provide a holistic and integrated workspace for project-related activities. Task Unity transcends these limitations by offering a unified platform where team members can seamlessly collaborate on tasks, share files and engage in real-time discussions. This comprehensive approach to collaboration is designed to boost overall productivity by eliminating the need for juggling multiple tools and ensuring that all project-related interactions occur within a centralized and user-friendly environment. In essence, Task Unity seeks to address the root causes of inefficiencies, providing project teams with the tools they need to succeed in today's complex and fast-paced work environments.

1.3 Objective

The main research contribution of this project is the development and implementation of an online web system, Task Unity which serves as Project Collaboration Platform with the following objectives:

1. To unify communication, task management, file sharing and progress tracking on a single platform.
2. To prioritize streamlined workflows through integrated features and real-time tools.
3. To foster coordination, transparency and accountability among team members.

1.4 Application

Task Unity, when implemented, serves as a robust solution for enhancing project collaboration in various scenarios. One notable application is in the realm of project collaboration management. The platform is specifically tailored to streamline communication, task tracking and collaboration for diverse project teams. The following features underscore its relevance and effectiveness:

1. **Centralized Project Data Management**: Task Unity acts as a central hub for organizing and managing project data, including tasks, timelines and

documentation. It simplifies the process of searching, sorting and updating project-related information, ensuring seamless data accessibility.

2. **Event Coordination**: Task Unity facilitates the planning and execution of project-related events, such as team meetings, workshops and milestone celebrations. The platform includes event management tools, allowing teams to organize and track various project activities efficiently.
3. **Task and Progress Tracking**: Task Unity offers robust task management features, enabling teams to create, assign and track project tasks seamlessly. The progress tracking tools, including Gantt charts, provide a visual representation of project timelines, ensuring teams stay on course.
4. **Collaborative Documentation**: The platform enhances collaborative document editing, allowing team members to work together on project documentation in real-time. It includes commenting features and revision history, fostering transparency and efficiency in document collaboration.
5. **Communication Hub**: Task Unity serves as a centralized communication hub, facilitating real-time team communication and announcements. This feature ensures that all team members are informed, connected and aligned with project goals.

In essence, Task Unity transforms project collaboration by providing a unified platform for efficient communication, task management and collaborative document editing, ensuring the success and cohesion of project teams.

1.5 Features

Task Unity boasts a range of features designed to elevate project collaboration. From centralized communication and intuitive task management to seamless file collaboration and progress tracking, the platform ensures a comprehensive and user-centric project management experience. With user-friendly interfaces, robust integrations and versatile functionalities, Task Unity emerges as an indispensable tool for streamlined and efficient teamwork across diverse industries:

1. **Task Management:** Assign and track tasks with ease, allowing for better project organization.
2. **Communication Tools:** Real-time chat, discussion forums and notifications to facilitate seamless team communication.

3. **File Sharing:** Centralized storage for project-related documents, fostering easy access and collaboration.
4. **User Permissions:** Customizable access levels to ensure data security.

1.6 Feasibility Analysis

1.6.1 Economic Feasibility

Initial investment in Task Unity development is justified by potential gains in team efficiency, reduced project delays and improved project outcomes. The cost of implementation is expected to be outweighed by the long-term benefits.

1.6.2 Technical Feasibility

The platform leverages existing technologies and development tools, making it technically feasible. Any technical challenges encountered during development will be addressed promptly.

1.6.3 Operational Feasibility

Task Unity is designed to seamlessly integrate into existing workflows, minimizing disruption. User acceptance testing and training programs will be implemented to ensure smooth adoption.

1.7 System Requirements

1.7.1 Software Requirement

In accordance with the needs and preferences of the development team as well as the technologies selected to create and support the Task Unity, a project collaboration platform, different software are utilized:

- **HTML:** HTML is a standard markup language used to create the structure and presentation of web pages. It provides a set of tags and elements that define the content and formatting of web pages. HTML is the backbone of the World Wide Web and is essential for creating and displaying web content. [3]
- **CSS:** CSS is a styling language that is used to specify the visual presentation and arrangement of HTML elements on a web page. It enables web designers to manage a website's colors, fonts, spacing and other design elements. Together with HTML and CSS helps to produce web pages that are both aesthetically pleasing and consistent. [3]
- **JavaScript:** JavaScript is a programming language used for adding interactivity and dynamic behavior to websites. It enables client-side scripting and manipulation of web page elements and content. [4]

- **Python:** Python is a high-level, general-purpose programming language known for its readability and versatility. It emphasizes code readability and ease of use, making it widely adopted for web development, data science, artificial intelligence, automation and more. [5]
- **VS Code:** VS Code is a popular source code editor developed by Microsoft. It provides a lightweight and versatile coding, debugging and version control platform that supports various programming languages and extensive customization options. [6]
- **MongoDB:** MongoDB is an open-source NoSQL database, storing data in flexible, JSON-like documents. It supports dynamic schemas, making it suitable for diverse and evolving data structures. [7]
- **Django:** Django is a high-level web framework for Python that facilitates rapid development by providing a clean and pragmatic design, including an object-relational mapping (ORM) system, admin interface and built-in features for common web development tasks. [8]
- **Tailwind CSS:** Tailwind CSS is a utility-first CSS framework that provides a set of low-level utility classes to build designs directly in your markup. It allows for a highly customizable and responsive UI development, focusing on flexibility and simplicity in styling web applications. [9]

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

In the contemporary landscape of project management, the literature consistently underscores the critical role played by effective collaboration platforms. These platforms are not just incidental tools; they stand as linchpins for improving team productivity and, consequently, influencing the overall success of projects. As we delve into the existing body of work, a discernible trend comes to the forefront a shift in preference towards integrated solutions with a holistic set of features. Organizations are increasingly seeking collaboration platforms that transcend basic functionalities. The contemporary demand is for solutions that comprehensively address communication, task management, document sharing and progress tracking. This shift signifies a heightened awareness of the interconnected nature of project workflows, emphasizing the need for unified and integrated solutions to navigate the dynamic challenges inherent in collaborative work within the fast-paced and interconnected professional landscape of today. [10]

2.2 Case Study

2.2.1 ASANA

Asana, a widely recognized project management and collaboration platform, has gained prominence for its robust features such as task assignment, due date tracking and project timeline visualization. Notable success stories include Nasa, Uber and Airbnb, showcasing its versatility across different industries. [11]

2.2.2 TRELLO

Trello, known for its visually intuitive project management approach using boards, offers features like a card-based system, checklists and seamless integrations. Companies like Google and National Geographic have leveraged Trello for its simplicity and effectiveness in project collaboration. [12]

2.2.3 SLACK

While primarily a messaging platform, Slack plays a crucial role in real-time team collaboration. Features like channels and extensive integrations have led to success stories at companies such as Airbnb and Shopify, emphasizing its impact on streamlined communication. [13]

2.2.4 MONDAY.COM

Monday.com stands out for its flexibility and customization options in project management. Success stories from Discovery Channel and Carlsberg highlight the platform's adaptability and effectiveness in diverse project scenarios. [14]

2.2.5 JIRA (ATLASSIAN)

Jira, a popular choice among software development teams for issue tracking and agile project management, features sprints and seamless integration with development tools. Its success stories at companies like Spotify and LinkedIn underscore its importance in the software development landscape. [15]

2.2.6 WRIKE

Wrike, a project management tool with a focus on performance analytics, offers dynamic features like request forms, Gantt charts and automation capabilities. Successful implementations at companies like Airbnb and Siemens demonstrate its effectiveness in complex project environments. [16]

2.2.7 GITHUB

GitHub, a leading platform for version control and collaborative software development, has played a crucial role in fostering efficient teamwork among developers. Its features, including pull requests, code review and project management tools, contribute to streamlined collaboration in the software development lifecycle. Success stories from organizations such as Microsoft and Spotify highlight GitHub's impact on facilitating seamless collaboration, ensuring code quality and promoting open-source contributions. [17]

2.2.8 GIT LAB

GitLab, renowned for its comprehensive DevOps platform, plays a pivotal role in facilitating collaboration among software development teams. With features encompassing version control, continuous integration and project management, GitLab has been instrumental in optimizing workflows. Success stories from organizations like Siemens and Ticketmaster showcase GitLab's effectiveness in enhancing collaboration, ensuring code quality and streamlining the software development lifecycle. [18]

CHAPTER 3: METHODOLOGY

3.1 Software Development Life Cycle (SDLC)

The Waterfall model is the earliest SDLC approach that was used for software development. The Waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete. [19]

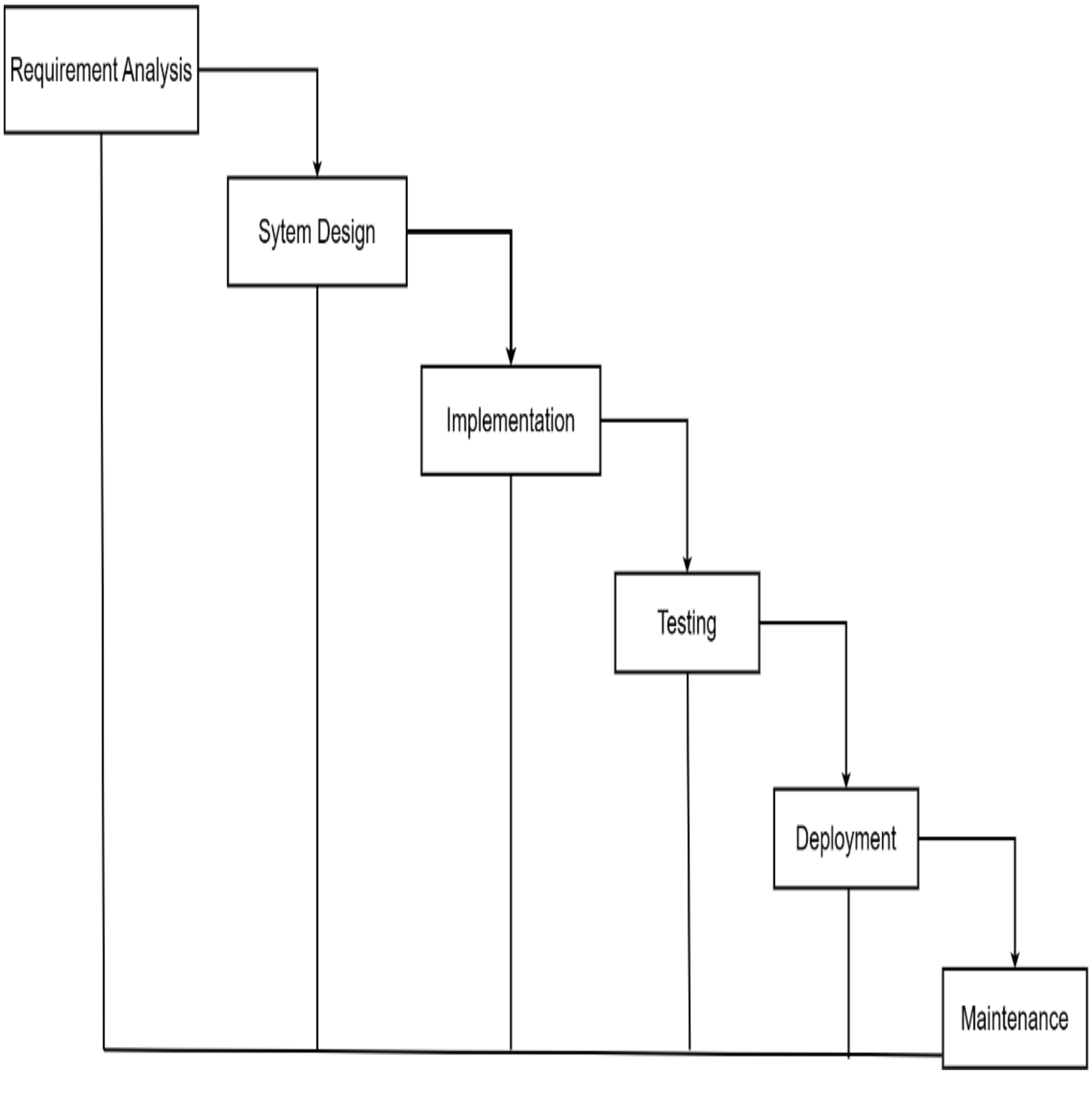


Figure 1 Software Development Life Cycle

The sequential phases in SDLC are:

- Requirement analysis: All the possible requirements for the successful development of system are gathered and analyzed among the group members in this phase.

- System Design: The requirement specifications from first phase are studied in this phase and the system design is prepared. This assist developers to analyze the hardware and software requirements. In our case software includes SQL server, Visual Studio, Adobe Photoshop and hardware includes a PC.
- Implementation: With inputs from the system design, the system is first developed in small programs called units.
- Integration and Testing: All the units developed in the implementation phase are integrated into a system for a test of faults and failures.
- Deployment: Once the functional and non-functional testing is done, the product is deployed in the customer environment.
- Maintenance: There are some issues which come up in the client environment or some better versions are released. Maintenance is done to deliver the changes and to enhance the product in the customer environment.

3.2 Flow of Project

A project flow diagram, synonymous with a project workflow diagram, serves as a visual representation of the sequential steps comprising the development of Task Unity, a comprehensive project collaboration platform. The initiation phase begins with the identification of goals, objectives and stakeholders, laying the foundation for the subsequent stages. Planning follows, where tasks, deadlines and resources are meticulously outlined, ensuring a strategic roadmap for the platform's development. [20]

The execution phase sees the realization of the planned tasks, with continuous monitoring and control to manage progress and address any deviations. In the case of Task Unity, the overarching project involves crafting a user-friendly web-based interface for seamless collaboration, communication and task management among project teams. The closing phase marks the culmination of the project, encompassing the completion of deliverables and a thorough project evaluation. [20]

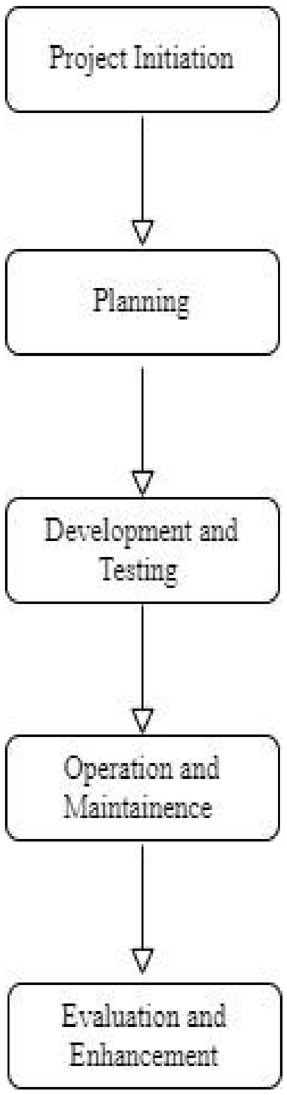


Figure 2 Flow of Project [Group Study, 2023]

- **Project Initiation:** Task Unity's initiation phase involves determining the need for the platform, defining its goals and specifications and assessing the feasibility of implementation. Stakeholders are identified, requirements documented and initial planning activities performed to establish a robust foundation for successful development and implementation. [20]
- **Planning:** The planning phase for Task Unity entails arranging and outlining activities, allocating resources and setting schedules for effective platform development. Additionally, potential risks are identified, mitigation plans devised and communication channels established to ensure a smooth and well-organized development process. [21]

- **Development and Testing:** Task Unity's development and testing phase revolves around creating a robust software application, ensuring functionality, performance and reliability. Programming, database implementation and user interface development transform the platform from concept to a fully functional system. Rigorous testing, including unit, integration, system and user acceptance testing, is conducted to identify and rectify any defects before deployment. [22]

- **Operation and Maintenance:** The operational phase for Task Unity necessitates constant maintenance to ensure continued functionality and relevance. Duties include system maintenance, user assistance, data backups, security updates and performance enhancement, with the aim of sustaining efficiency and addressing user concerns. [23]
- **Evaluation and Enhancement:** Task Unity undergoes continuous evaluation and improvement to enhance usability and functionality. Developer feedback, usage tracking and recurring assessments inform decisions for upgrades and modifications, ensuring that the platform evolves to meet changing requirements and consistently delivers optimal performance. [24]

3.3 System Design

3.2.1 E-R DIAGRAM

An ER Diagram is a graphical representation that depicts the entities, attributes and relationships in a database system. Entities represent real-world objects or concepts attributes represent the properties or characteristics of entities and relationships define the associations between entities. ER Diagrams use symbols such as rectangles for entities, ovals for attributes and lines with different notations for relationships. [25]

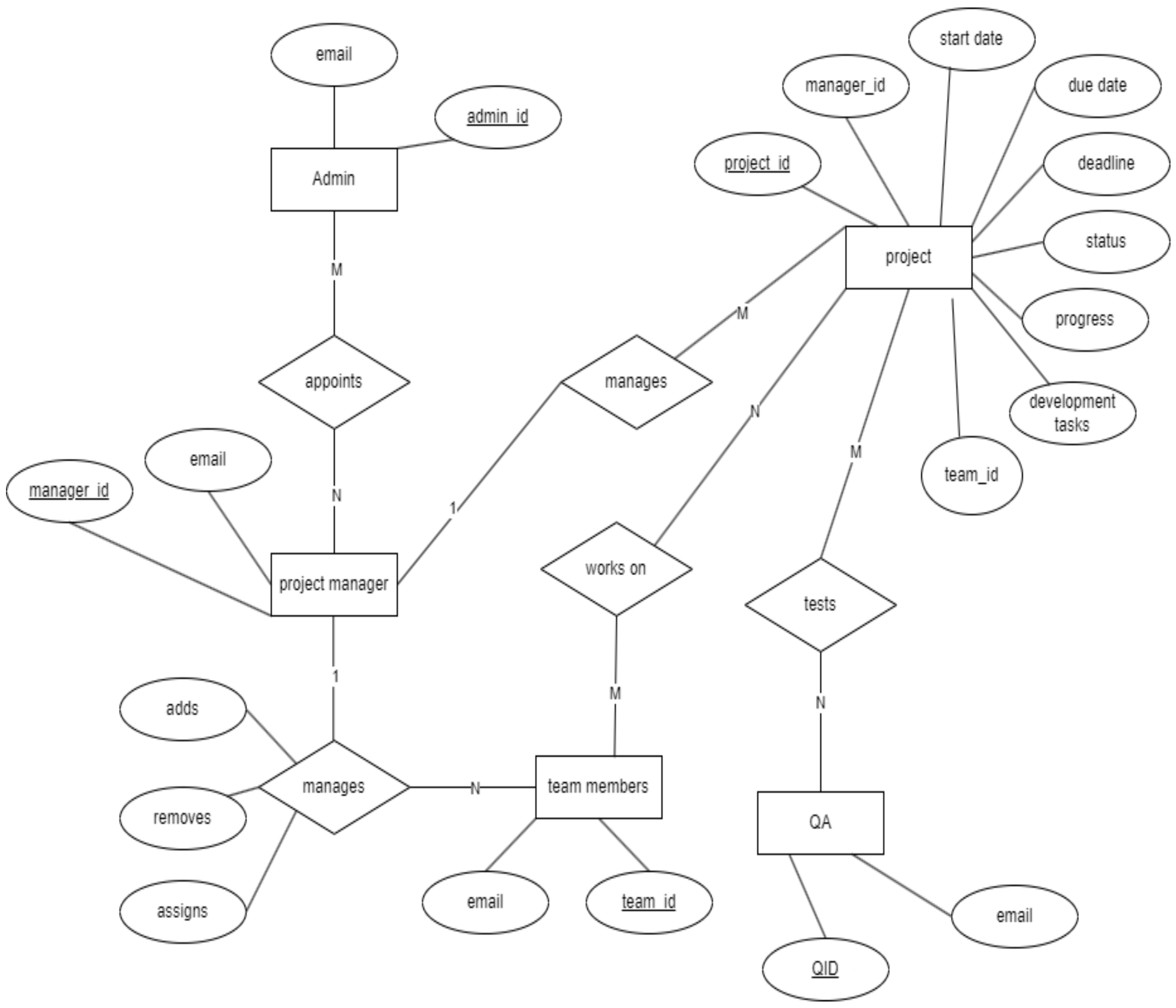


Figure 3 Entity- Relationship Diagram [Group Study, 2024]

3.2.2 Data Flow Diagram

A Data Flow Diagram (DFD) is a visual representation that depicts the flow of data within a system or process. It illustrates the movement of data from its sources, through various processes and to its destinations, highlighting the interactions between different components of the system. [26]

- Level 0 Data Flow Diagram:

  The Level 0 Data Flow Diagram (DFD) provides an overview of the entire system by showing the high-level processes, external entities and data flows. It represents the system as a single process and provides a big-picture understanding of how data flows into and out of the system. The Level 0 DFD serves as the foundation for developing more detailed DFDs. [26]

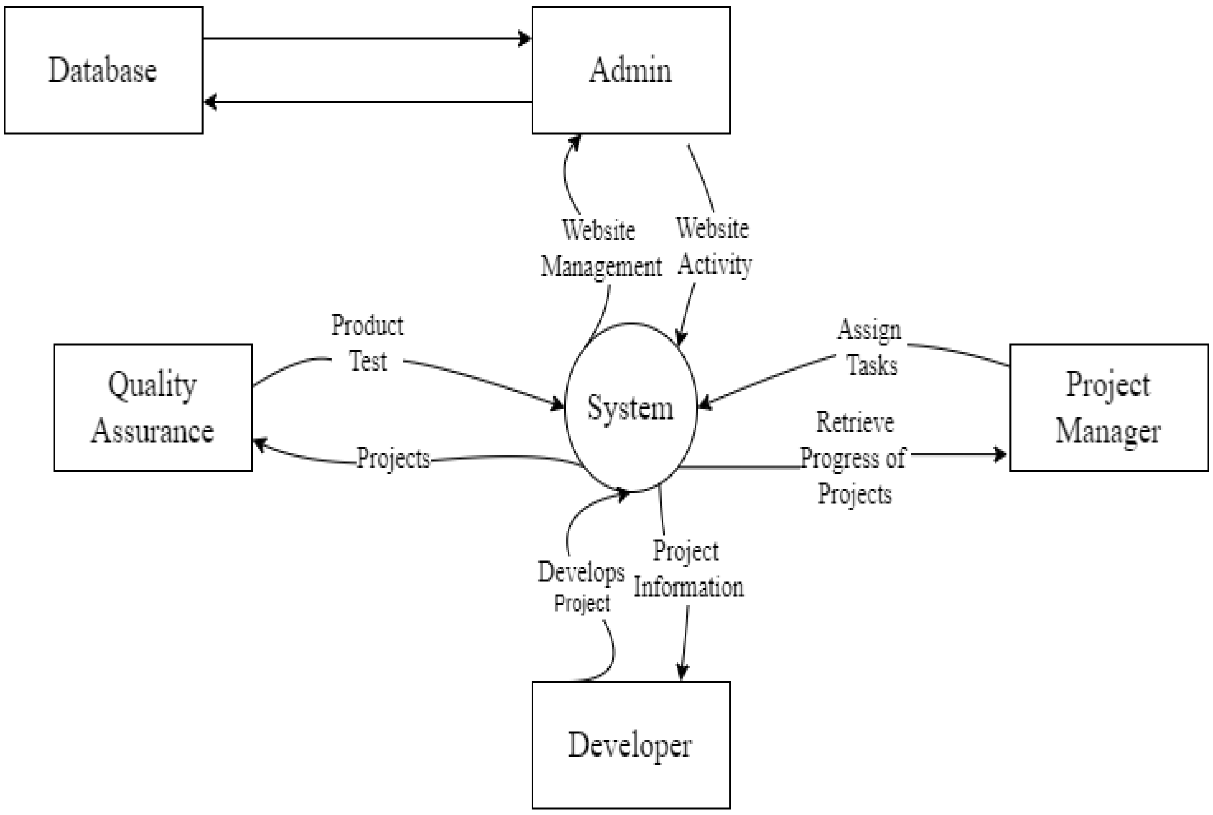


Figure 4 Level 0 DFD [Group Study, 2024]

- Level 1 DFD

  The Level 1 Data Flow Diagram is a more detailed representation of the system, expanding on the Zero Level DFD. It decomposes the system into major processes and sub processes, showing the interactions and data flows between them. The Level 1 DFD provides a more granular view of the system's functionality and serves as a basis for creating the Level 2 DFDs. [27]

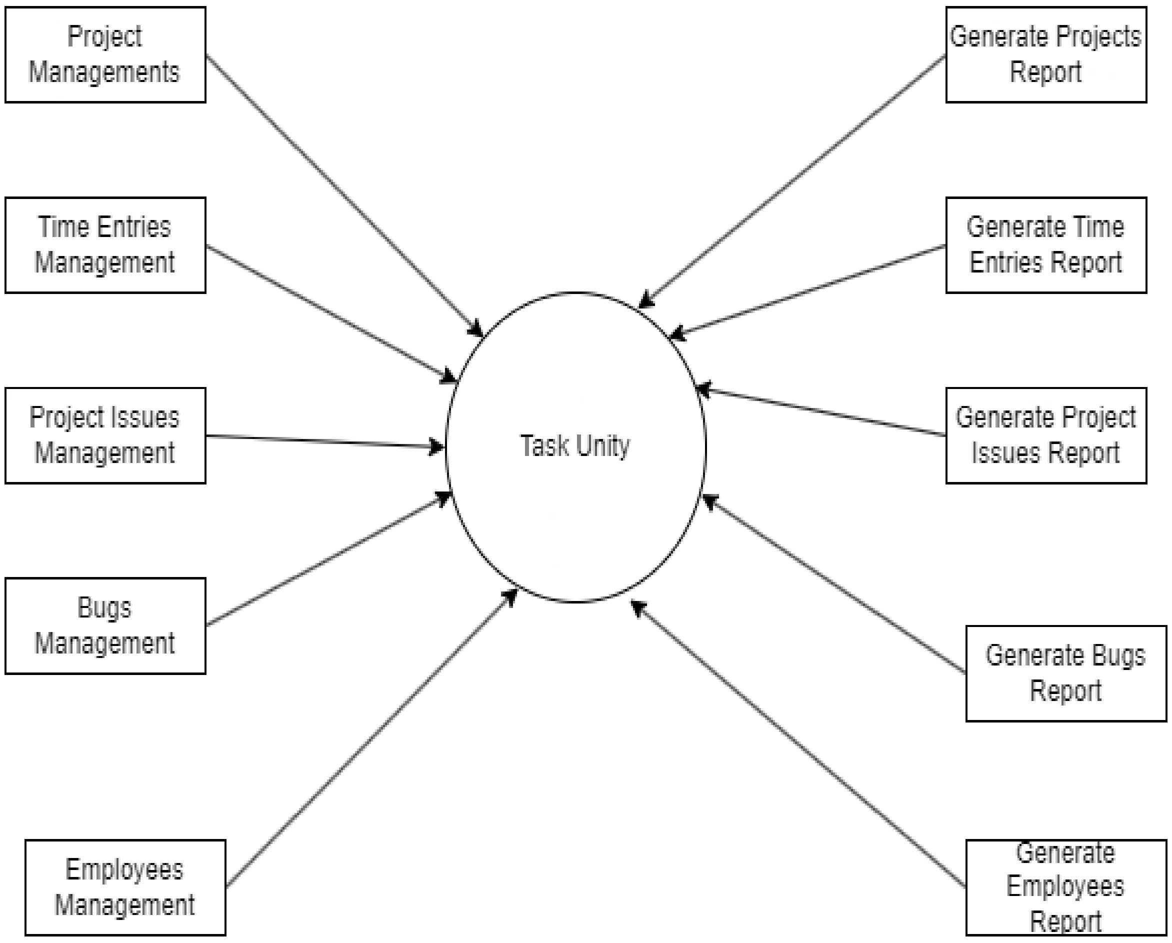


Figure 5 Level 1 DFD [Group Study, 2024]

- LEVEL 2 DFD

  The Level 2 Data Flow Diagram delves further into the Level 1 DFD by breaking down each major process into more detailed sub processes. It provides a comprehensive view of the system's processes, data flows, data stores and external entities at a more specific level. The Level 2 DFD helps in understanding the inner workings of the system and aids in system analysis and design. [28]

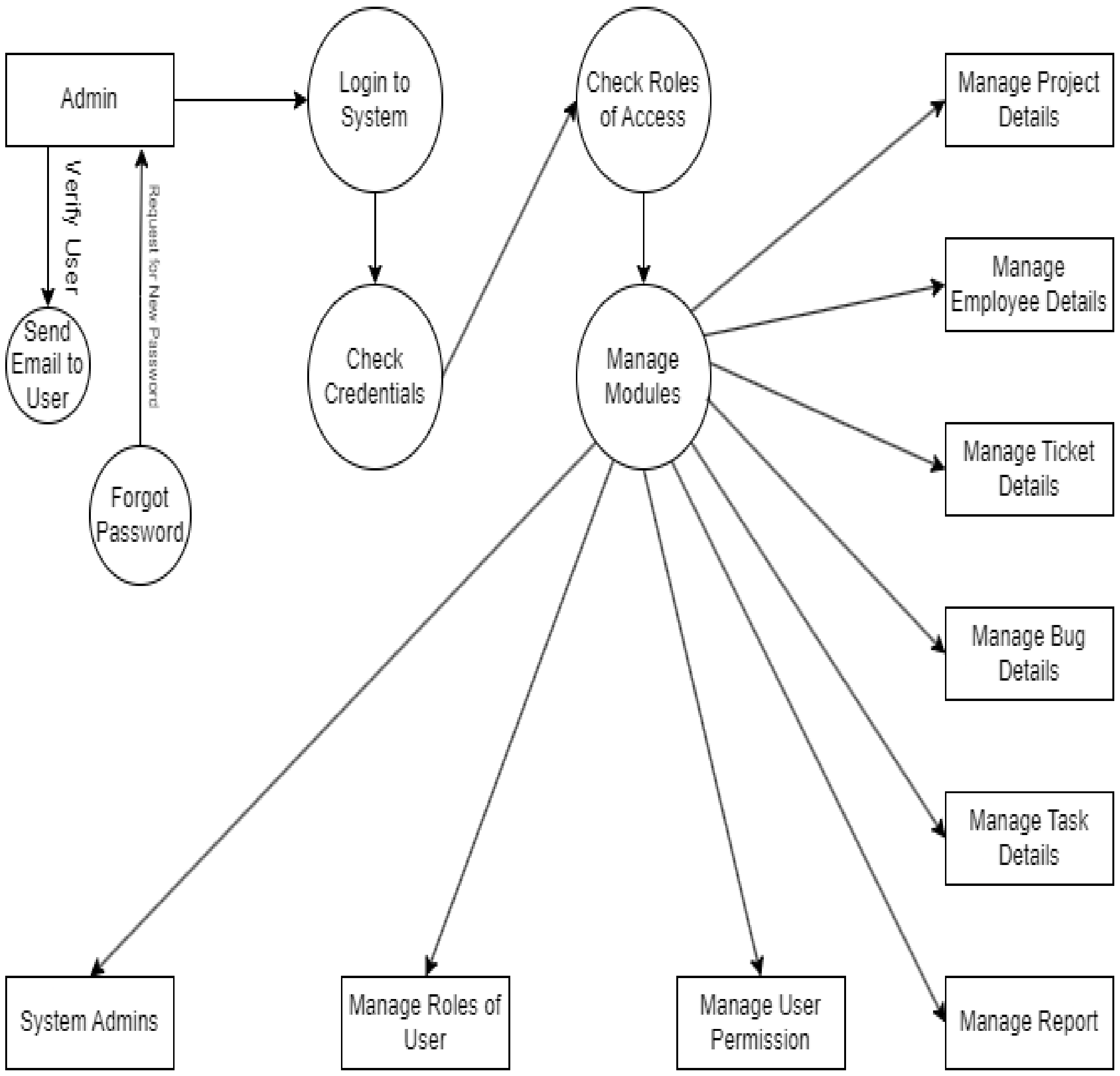


Figure 6 Level 2 DFD [Group Study, 2024]

3.2.3 USE CASE DIAGRAM

A Use Case Diagram is a visual representation of the interactions between actors (users or external systems) and a system. It illustrates the functionalities and behaviors of a system from the user's perspective. [29]

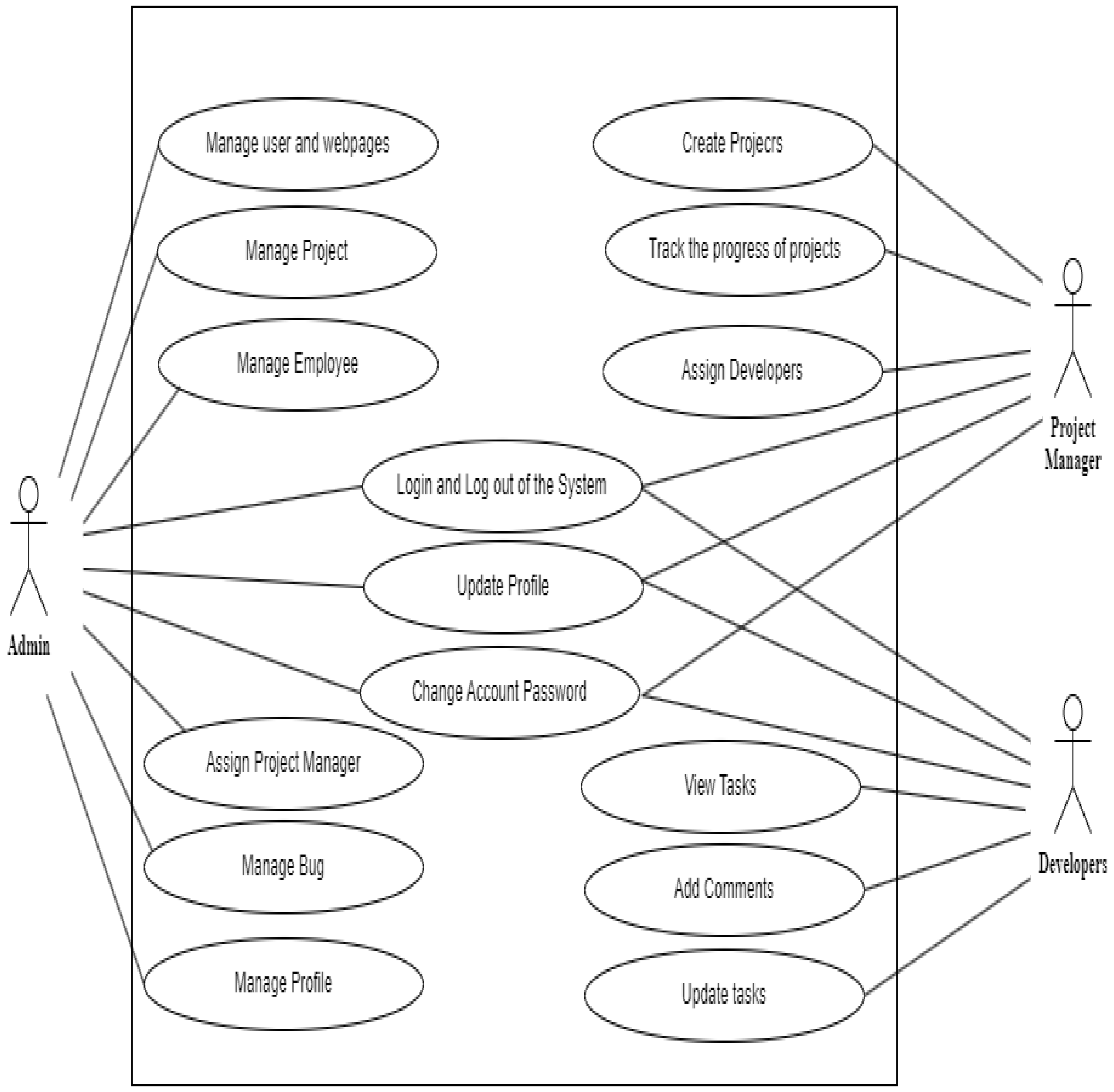


Figure 7 Use Case Diagram [Group Study, 2024]

3.2.4 System Flow Diagram

A system flow diagram, also known as a system flowchart is a visual representation of the flow of information or processes within a system. It typically uses symbols and arrows to illustrate the sequence of steps and decision points involved in a particular process or system. [30]

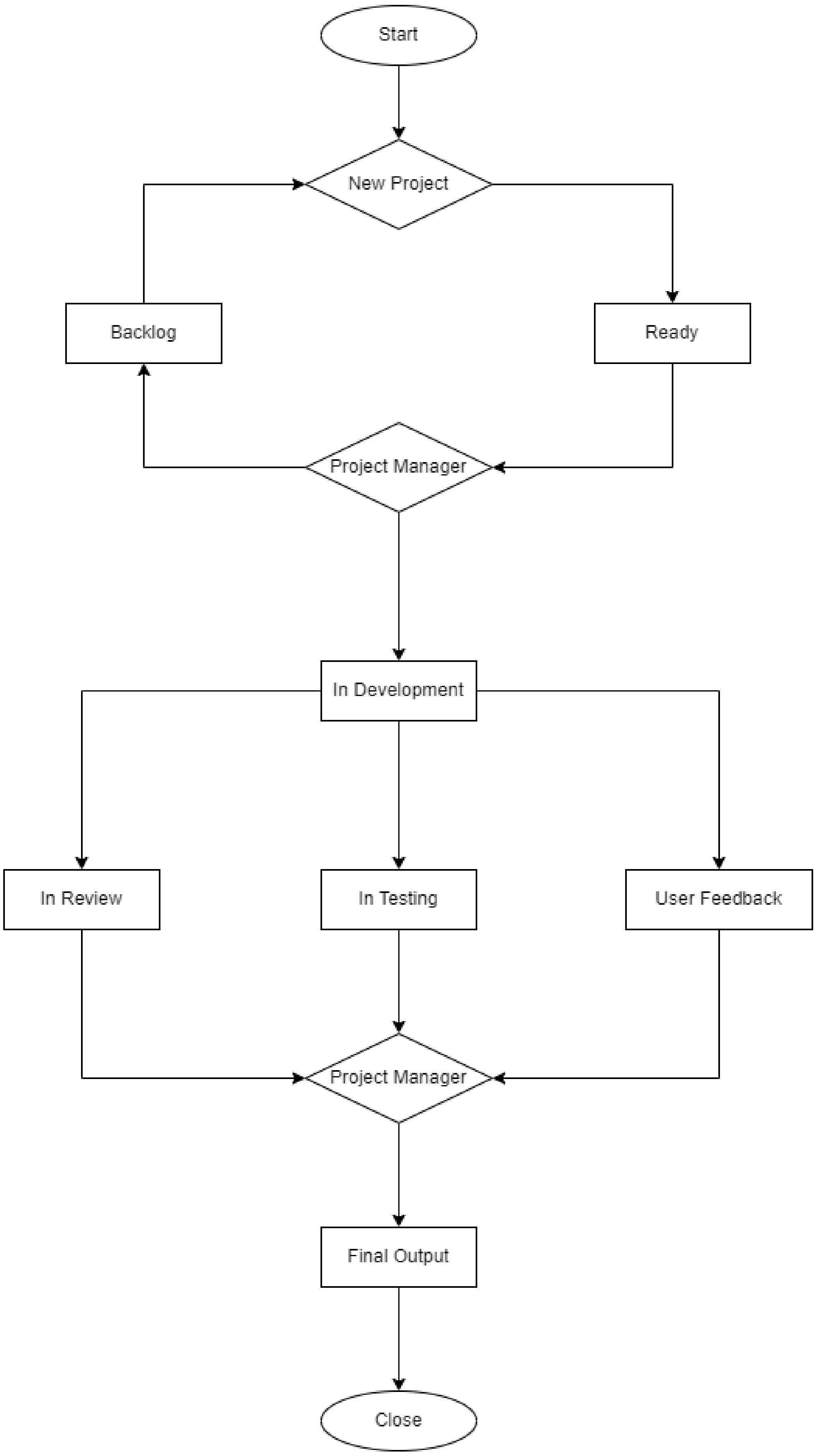


Figure 8 System Flow Diagram [Group Study, 2024]

3.4 Database Management

Task Unity relies on MongoDB, a dynamic NoSQL database, to underpin its data architecture. The choice of MongoDB is strategic, driven by its unparalleled flexibility and scalability, ensuring Task Unity's adaptability to the evolving landscape of project collaboration.

Key Aspects of Database Management:

1. **Flexibility and Adaptability:**

   MongoDB's document-oriented model allows Task Unity to store data flexibly, accommodating changes in user data, project details, tasks and communication history.

2. **Efficient Data Storage:**

   The structured database efficiently stores user profiles, project specifications, task details and communication history, ensuring responsive indexing and retrieval.

3. **Dynamic Project Elements:**

   Task Unity's database design handles the dynamic nature of project elements, effortlessly adapting to changes as teams collaborate and projects progress.

4. **Scalability for Growth:**

   MongoDB's scalability supports Task Unity's growth, seamlessly handling expanding user bases and increasing data volumes while maintaining optimal performance.

5. **Streamlined Communication History:**

   The database efficiently manages communication history, ensuring easy retrieval of past interactions and providing a comprehensive record of project discussions.

By leveraging MongoDB's strengths, Task Unity's database management approach becomes a linchpin in the platform's success. It not only ensures the efficient storage and retrieval of critical project data but also positions Task Unity as a flexible, scalable and adaptive solution in the realm of collaborative project management.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Output

Task Unity tackles communication gaps and streamlines collaboration with a central communication hub featuring real-time chat, forums, and notifications. Integrated task management empowers teams to create, assign, and track tasks with progress visualization, fostering collaboration and boosting project efficiency. Data visualization tools provide valuable insights to ensure project success.

4.2 Web Development

4.2.1 Frontend

The front-end environment for this project comprised of interactive, desirable and understandable web pages; Home page, Login Page, Sign-up page, Dashboard pages are illustrated below:

Sign-in Page

This is the page where user can login.

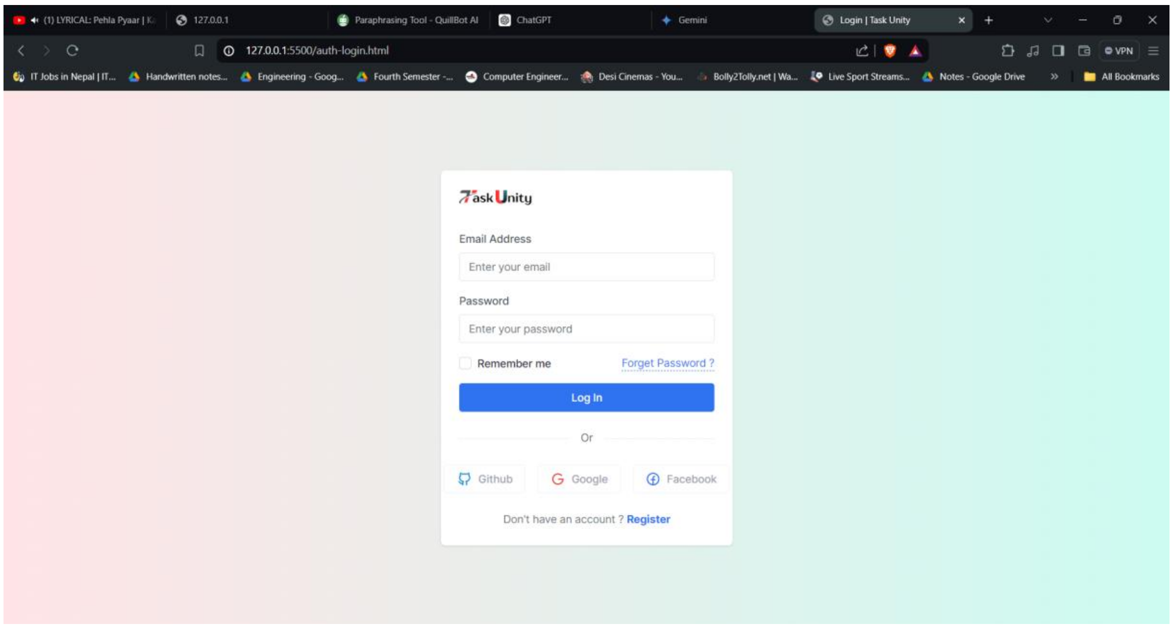


Figure 9 Sign in Page

Password Recovery Page

This is the page where user can reset their password.

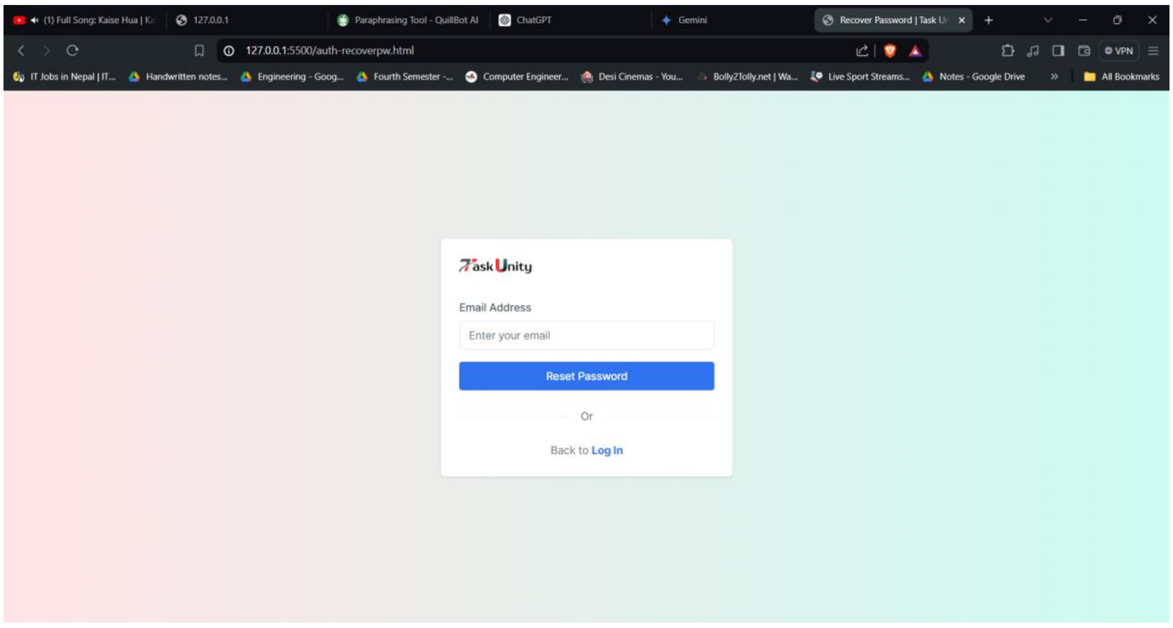


Figure 10 Password Recovery Page

Dashboard Page

This the page that user lands after logging in.

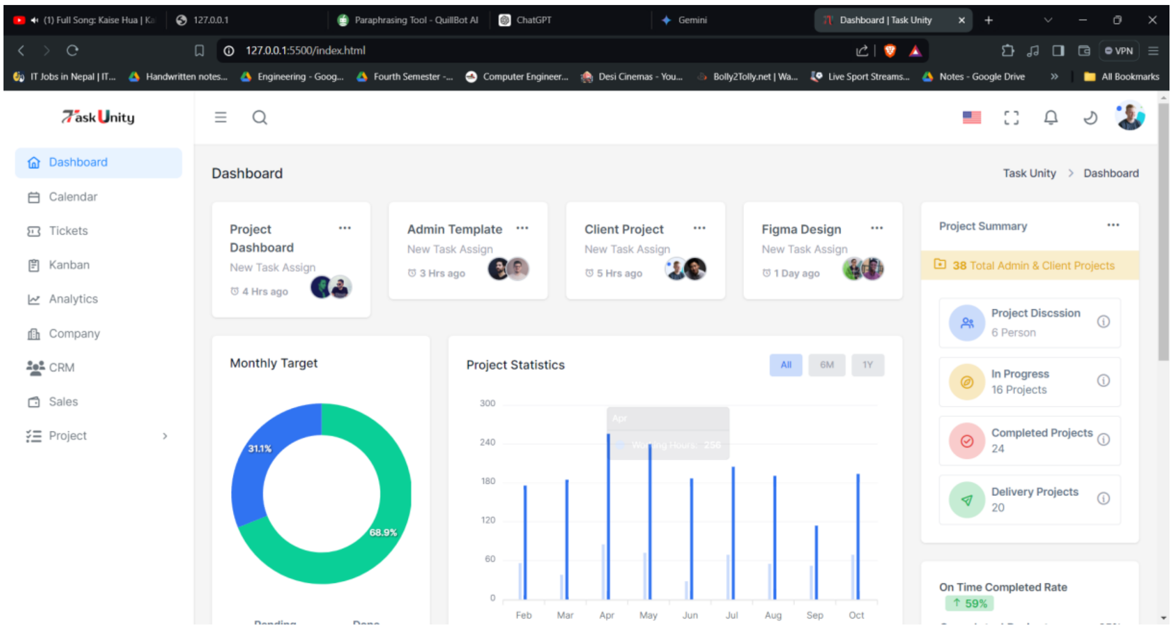


Figure 11 Dashboard Page

Kanban Page

This is the important page where we can see all the task that are ongoing.

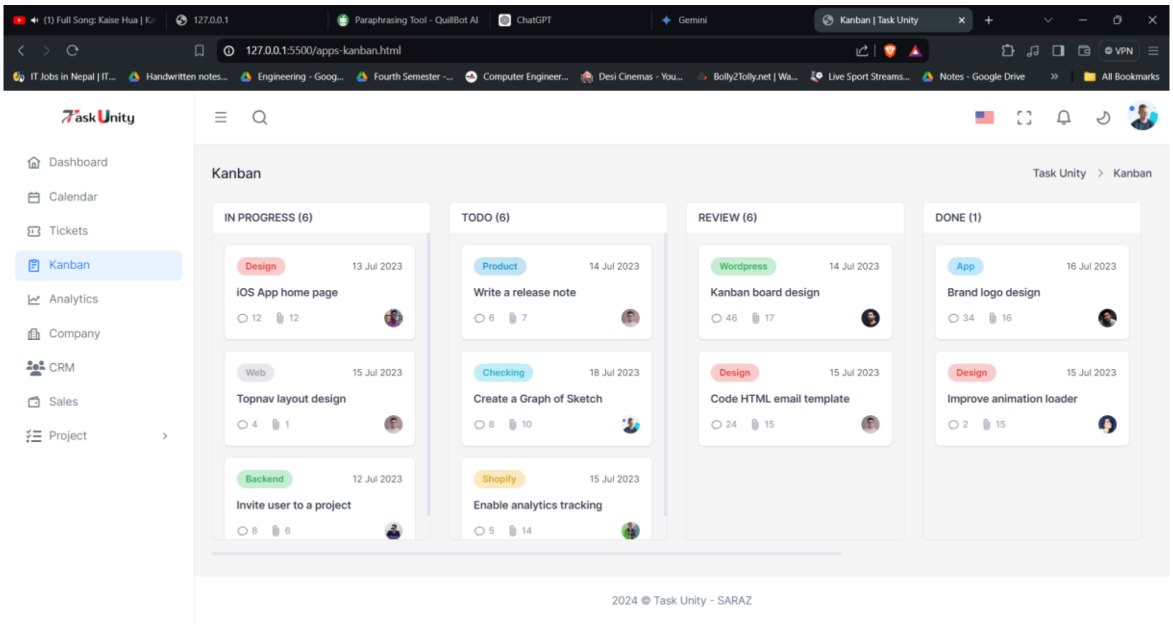
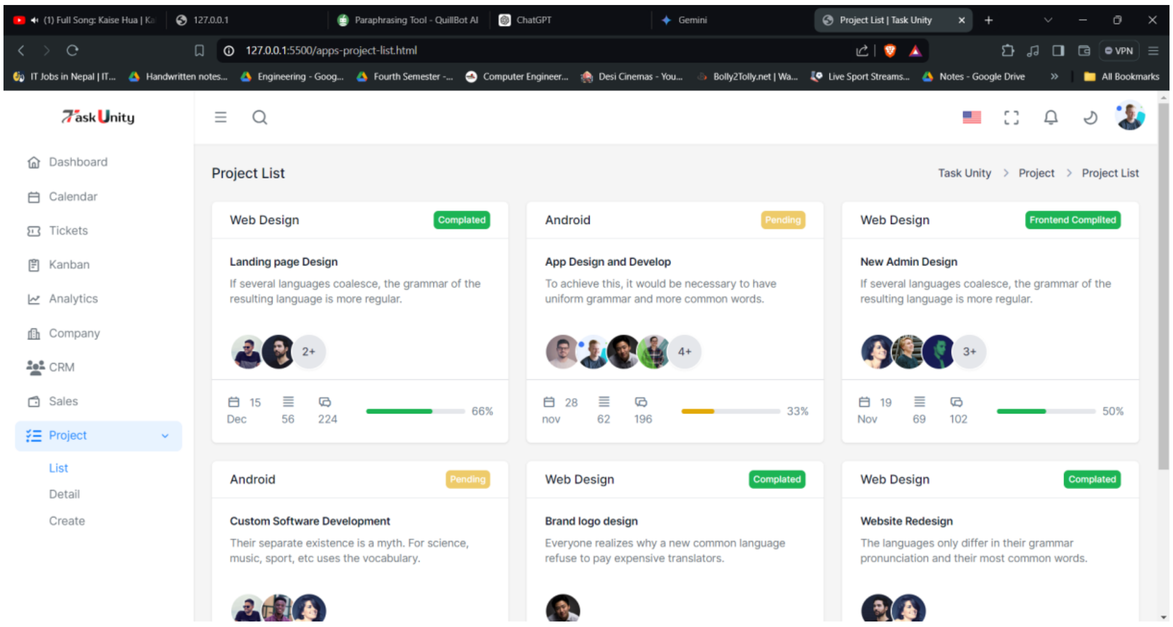

127.0.0.1:5500/apps-kanban.html
TaskUnity
Dashboard
Calendar
Tickets
Kanban
Analytics
Company
CRM
Sales
Project
Kanban
Task Unity > Kanban
IN PROGRESS (6)
TODO (6)
REVIEW (6)
DONE (1)
Design 13 Jul 2023 iOS App home page 12 12
Product 14 Jul 2023 Write a release note 6 7
Wordpress 14 Jul 2023 Kanban board design 46 17
App 16 Jul 2023 Brand logo design 34 16
Web 15 Jul 2023 Topnav layout design 4 1
Checking 18 Jul 2023 Create a Graph of Sketch 8 10
Design 15 Jul 2023 Code HTML email template 24 15
Design 15 Jul 2023 Improve animation loader 2 15
Backend 12 Jul 2023 Invite user to a project 8 6
Shopify 15 Jul 2023 Enable analytics tracking 5 14
2024 © Task Unity - SARAZ

Figure 12 Kanban Page

Project List Page

This is the page where we can see the list of projects and their status.



If several languages coalesce, the grammar of the resulting language is more regular.

To achieve this, it would be necessary to have uniform grammar and more common words.

If several languages coalesce, the grammar of the resulting language is more regular.

Their separate existence is a myth. For science, music, sport, etc uses the vocabulary.

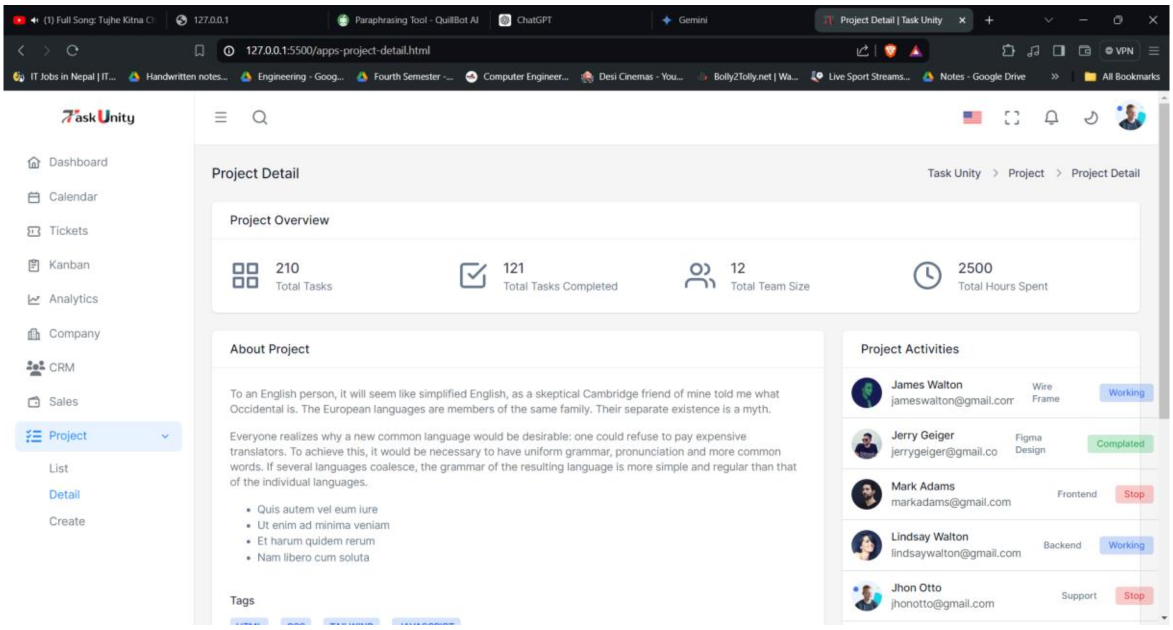
Everyone realizes why a new common language refuse to pay expensive translators.

The languages only differ in their grammar pronunciation and their most common words.

Figure 13 Project List Page

Project Detail Page

This is the page where every project details can be found.



210	Total Tasks
121	Total Tasks Completed
12	Total Team Size
2500	Total Hours Spent

About Project

To an English person, it will seem like simplified English, as a skeptical Cambridge friend of mine told me what Occidental is. The European languages are members of the same family. Their separate existence is a myth.

Everyone realizes why a new common language would be desirable: one could refuse to pay expensive translators. To achieve this, it would be necessary to have uniform grammar, pronunciation and more common words. If several languages coalesce, the grammar of the resulting language is more simple and regular than that of the individual languages.

- Quis autem vel eum iure
- Ut enim ad minima veniam
- Et harum quidem rerum
- Nam libero cum soluta

Project Activities

James Walton	jameswalton@gmail.com	Wire Frame	Working
Jerry Geiger	jerrygeiger@gmail.co	Figma Design	Completed
Mark Adams	markadams@gmail.com	Frontend	Stop
Lindsay Walton	lindsaywalton@gmail.com	Backend	Working
Jhon Otto	jhonotto@gmail.com	Support	Stop

Figure 14 Project Detail Page

Project Create Page

This is the page where project manager creates project for their employees.

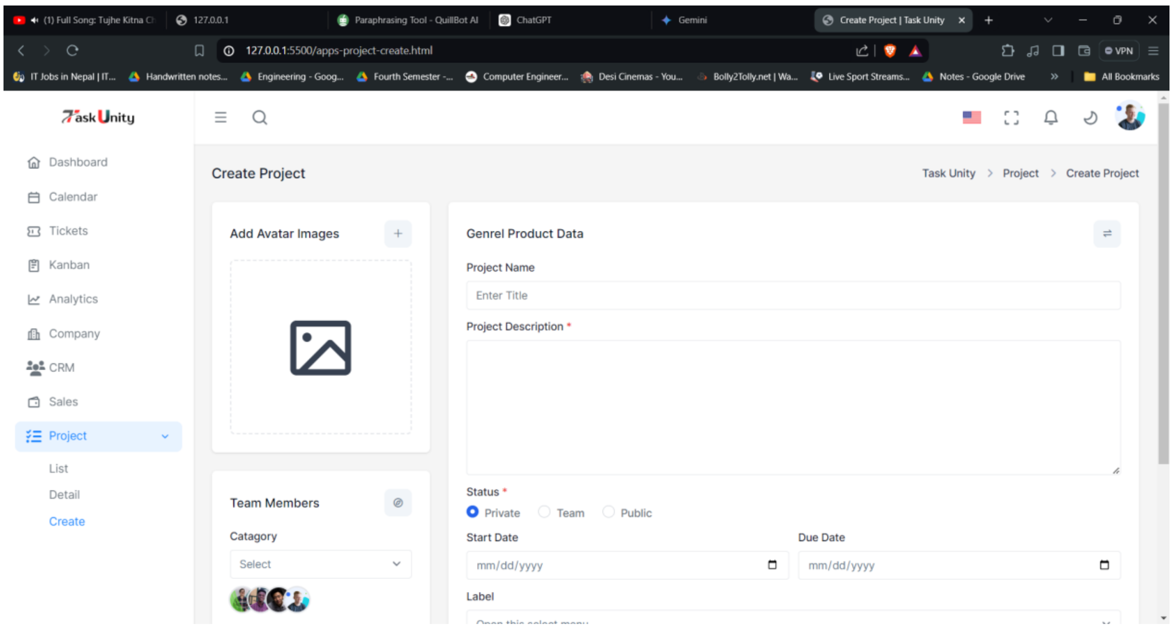


Figure 15 Project Create Page

4.3 Work Remaining

While Task Unity boasts significant progress, several crucial aspects require further development:

1. **Frontend Development:** Complete the web interface using the chosen framework (e.g., HTML, CSS, JavaScript). Develop functionalities like user interface elements, user interaction with tasks and communication features, and data visualization tools.
2. **Integration Testing:** Rigorously test the integration between the frontend and backend to ensure seamless data exchange and functionality.
3. **User Acceptance Testing:** Conduct user acceptance testing (UAT) with target users to gather feedback on usability, identify potential bugs, and refine the platform based on user experience.
4. **Deployment and Maintenance:** Develop a deployment strategy to launch Task Unity on a suitable hosting platform. Ongoing maintenance will be required to address bugs, implement security updates, and incorporate new features based on user feedback and future project requirements.
5. **Advanced Features:** Consider implementing additional functionalities based on project scope and user needs. These may include integration with third-party applications, mobile app development, or advanced reporting tools.

4.4 Limitations

Task Unity offers a robust platform, but consider these limitations for ongoing development:

1. **Scalability & Performance:** Managing large user bases and complex projects might require infrastructure upgrades or cloud solutions to maintain real-time communication efficiency.
2. **Offline Functionality:** Users with unstable internet connections might benefit from limited offline functionalities like cached task details with synchronization upon reconnection.
3. **User Adoption & Engagement:** User-friendly interfaces, tutorials, and gamification elements can help reduce onboarding time and encourage continued use.
4. **Project Management & Security:** Prioritize features, ensure smooth integration with existing tools, and implement robust security measures to protect sensitive information.

4.5 Problem Faced

Developing Task Unity presented several challenges that required creative solutions:

1. **Data Integration:** Integrating data from different sources (e.g., user profiles, task details, communication history) can be complex. We'll address this by designing a well-structured database schema and implementing robust data management practices to ensure seamless data integration and retrieval.
2. **Real-Time Communication Challenges:** Maintaining real-time communication across geographically dispersed teams or large user bases can be demanding. Our team will explore solutions like optimizing code, utilizing efficient data synchronization techniques, and potentially integrating technologies like WebSockets to ensure smooth and reliable real-time collaboration features.
3. **User Interface (UI) Design and Usability:** Designing an intuitive and user-friendly interface is crucial for user adoption and efficient workflow integration. We'll conduct user testing throughout the development process to gather feedback and refine the UI, ensuring it's clear, easy to navigate, and caters to diverse user needs.
4. **Security and Scalability:** As user bases and project data volume grow, security threats and scalability issues become more prominent. We'll prioritize implementing robust security measures (authentication, authorization, encryption) to protect user data. Additionally, we'll explore potential solutions like cloud-based infrastructure or server upgrades to ensure Task Unity can scale effectively with increasing user demands.

4.6 Budget Analysis

The budget estimation for Task Unity considers several factors: project scope, desired features, implementation complexity, and any customization or integration needs.

Table 1 Time Estimation

Cost Element	**Estimated Cost Range (NPR)**
Infrastructure	NPR 5,000 – NPR 10,000
Training and Documentation	NPR 5,000 – NPR 10,000
Testing and Quality Assurance	NPR 5,000 – NPR 10,000
Maintenance and Support	NPR 10,000 – NPR 25,000
Total:	NPR 25,000 – NPR 55,000

4.7 Work Schedule

The time estimation for preparing our project can vary depending on various factors, such as the complexity of the project, the scope of work, the availability of resources and the specific requirements set by the recipient of the proposal. Depending on the complexity of the project, our work schedule takes a total time span of about three months ranging from the phase after the proposal submission. The following is the Gantt Chart representing our work schedule:

Table 2 Gantt Chart

S.N.	Activities	January		April				May				June		
	Weeks	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd
1.	Literature Review	■	■											
2.	Preparation of Proposal		■											
3.	Proposal Defense		■											
4.	Coding Initialization					■	■	■	■					
5.	Midterm Presentation								■	■				
6.	Coding Continuation										■	■	■	
7.	Report Preparation	■	■	■	■	■	■	■	■	■	■	■	■	■
8.	Final Defense of Project Report												■	■

January 3rd – January 4th Week: For our project, we research to gather different information about the project. And after that we are going to decide the topics of our project. Then deciding the topic, we research different tools which will be used in our project. We are also moving to different platforms to collect the related data for our project. We also started to write the proposal for proposal defense. And on the last of the January, we are ready for the first proposal defense.

April 3rd – May 2nd Week: We will begin our coding session, followed by our GUI design (menu bar, icon, etc.). During this session, we also discussed programming tools, web applications and database management. Over the next few weeks, we will keep testing our application and monitoring it for any issues related to it that may improve its performance.

May 2nd – 3rd Week: The mid-term defense will be initiated following code initiation. During this interval we will discuss the progress we have made since the project began, as well as the challenges we have encountered.

May 4th – June 3rd Week: Our project and website will be at the final stage of development. Any errors encountered will be tested and debugged. Our final defense day is the last day of testing and writing code. Our last week will be dedicated to presenting all the documentation we have collected from the beginning of the project. At the end of the project, we submit our report to the department that is responsible for it.

CHAPTER 5: CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

Task Unity tackles communication gaps and streamlines collaboration through a central communication hub, integrated task management, and data visualization tools. This user-friendly platform empowers teams to achieve greater efficiency, transparency, and project success. Looking ahead, Task Unity can integrate with third-party applications, explore AI and machine learning for automation and insights, and offer customization options for an enhanced user experience.

5.2 Scope for Future Enhancement

While Task Unity offers a solid foundation, there's always room for improvement:

1. **Advanced Features:** Integrate with third-party applications (e.g., cloud storage, project management tools) or develop functionalities like mobile apps and advanced reporting tools.
2. **Artificial Intelligence (AI):** Explore the potential of AI to automate repetitive tasks, predict project risks, or suggest personalized recommendations for improved project management.
3. **Machine Learning (ML):** Utilize machine learning for sentiment analysis within communication channels to identify potential project roadblocks or team morale issues.
4. **Customization and Flexibility:** Develop features that allow users to customize dashboards, workflows, or notification preferences for enhanced user experience.
5. **Global Expansion:** Implement internationalization and localization features to cater to a wider audience with diverse language and cultural needs.

By continuously evaluating user feedback and exploring these future enhancements, Task Unity can solidify its position as a leading platform for collaborative project management.

REFERENCES

[1] F. Lanubile, E. Christof, P. Rafael and V. Aurora, "Collaboration tools for global software engineering," *IEEE Software*, vol. 2, no. 27, p. 52, 2010.

[2] M. Li, C. Guangyu, Z. Zhe and F. Yi, "A social collaboration platform for enterprise social networking," *In Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, pp. 671-677, 2012.

[3] J. Duckett, HTML & CSS: Design and Build Websites, John Wiley & Sons., 2011.

[4] F. david, JavaScript: The Definitive Guide, O'Reilly Media, 2020.

[5] G. Van Rossum and F. L. Drake, An introduction to Python, Bristol: Network Theory Limited, 2003, p. 115.

[6] E. Gamma, "Visual Studio Code," Microsoft, 29 April 2015. [Online]. Available: https://code.visualstudio.com/docs. [Accessed 21 May 2023].

[7] C. Győrödi, R. Győrödi, G. Pecherle and A. Olah., "A comparative study: MongoDB vs. MySQL," *In 2015 13th International Conference on Engineering of Modern Electric Systems (EMES)*, pp. 1-6, 2015.

[8] J. Forcier, P. Bissex and W. J. Chun, Python Web Development with Django, Addison-Wesley Professional, 2008.

[9] A. Wathan, J. Reinink, D. Hemphill and S. Schoger, "Tailwind CSS," Tailwind Labs Inc., October 2017. [Online]. Available: https://www.tailwindcss.com/docs/. [Accessed 25 January 2024].

[10] K. Manikas and H. Klaus Marius, " Software ecosystems–A systematic literature review," *Journal of Systems and Software*, vol. 5, no. 86, pp. 1294-1306, 2013.

[11] D. Moskovitz and J. Rosenstein, "Asana Forum," Asana, 16 December 2008. [Online]. Available: https://forum.asana.com/. [Accessed 28 December 2023].

[12] Atlassian, J. Spolsky and Glitch, "About Us: Trello History, Logos & Customers | Trello," Trello Enterprise, 2011. [Online]. Available: https://trello.com/about.

[Accessed 2 January 2024].

[13] Slack Technologies, August 2013. [Online]. Available: https://slack.com/about. [Accessed 4 January 2024].

[14] "Our Story - monday.com," Public, February 2012. [Online]. Available: https://monday.com/p/about/. [Accessed 6 January 2024].

[15] "Jira | Atlassian Community," Atlassian, 2002. [Online]. Available: https://www.community.atlassian.com/t5/Jira-Software/ct-p/jira. [Accessed 16 January 2024].

[16] "Educational Work Management Guides | Wrike," Citrix Systems, 2006. [Online]. Available: https://www.wrike.com/guides/. [Accessed 23 January 2024].

[17] "GitHub Docs," Microsoft, 2008. [Online]. Available: https://www.docs.github.com/en. [Accessed 23 January 2024].

[18] "The DevSecOps Platform | Gitlab," GitLab Inc., 2014. [Online]. Available: https://www.about.gitlab.com/. [Accessed 24 January 2024].

[19] Y. Bassil, "A Simulation Model for the Waterfall Software Development Life Cycle," *International Journal of Engineering & Technology (IJET)*, vol. 02, 2012.

[20] J. K. Pinto, Project Management: Achieving Competitive Advantage, Pearson, 2016.

[21] Q. Fleming and J. Koppelman, Earned Value Project Management, Project Management Institute, 2000.

[22] R. K. Wysocki, Effective Project Management: Traditional, Agile, Extreme, Wiley, 2013.

[23] E. Larson and C. Gray, Project Management: The Managerial Process, McGraw-Hill Education, 2017.

[24] H. Kerzner, Project Management: A Systems Approach to Planning, Scheduling and Controlling, Wiley, 2017.

[25] C. Coronel, Database Systems: Design, Implementation, & Management, Cengage

Learning, 2016.

[26] A. Dennis, B. H. Wixom and R. M. Roth, Systems Analysis and Design, Wiley, 2014.

[27] A. Dennis, B. H. Wixom and D. Tegarden, Systems Analysis and Design with UML, Wiley.

[28] J. Valacich and J. George, Modern Systems Analysis and Design, Pearson, 2016.

[29] C. Larman, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development, Pearson, 2004.

[30] G. B. Shelly, T. J. Cashman and H. J. Rosenblatt, Systems Analysis and Design, Course Technology, 2007.