OneDesk

Project Proposal

Session: 2022 – CS

Submitted by:

Khadija Javed	2022-CS-604
Zaid Latif	2022-CS-617
Moazam Ali	2022-CS-624
Muaz Khalid	2022-CS-628
Zohaib Saeed	2022-CS-633

Supervised by:

Ms. Zoha

Department of Computer Science, New Campus

University of Engineering and Technology

Lahore, Pakistan

Contents

List of Figures

List of Tables

Chapter 1

Proposal Synopsis

1.1 Abstract

Organizations today face significant challenges in managing complex workflows due to fragmented systems for project tracking, team communication, code management, and financial Management. Despite the availability of individual tools such as JIRA, Trello, Basecamp, and Slack, software teams still struggle with inefficiencies caused by switching between platforms, duplicated efforts, and the lack of centralized oversight. The rapid growth of the global project management software market further highlights the pressing demand for integrated solutions that streamline operations and enhance decision-making. A key differentiator of the proposed system is its emphasis on automation to minimize manual updates and reduce human error. Unlike traditional platforms where task progress and status updates heavily depend on user input, this solution leverages Git-linked commits, pull requests, and reviewer approvals to automatically update project status. Automated code summaries, workload balancing, and real-time error handling further streamline workflows, ensuring transparency, accountability, and accurate reporting without additional administrative overhead. Building on this foundation, the project proposes the development of a unified software management platform that integrates project and task management with milestone tracking, real-time team communication, secure file sharing, financial invoicing, and meeting scheduling. Unlike existing tools, the proposed system emphasizes contextual collaboration, embedding communication and invoicing directly within project workflows to minimize fragmentation and improve productivity. Managers will gain access to real-time analytics dashboards for tracking progress, performance, and financial metrics, supporting informed, data-driven decisions. The system will be developed using a modern technology stack, including Next.js,

React, TypeScript, Tailwind CSS, Django, Django REST Framework, and PostgreSQL, ensuring scalability, maintainability, and seamless integrations with tools like GitHub and Google Meet. A phased development methodology will guide implementation, beginning with project and task management modules and progressively integrating communication, invoicing, scheduling, and AI-driven summarization. By combining automation with integrated collaboration, this project aims to deliver a comprehensive platform that not only consolidates the strengths of existing tools but also eliminates their shortcomings. The unified approach will significantly enhance organizational efficiency, empower development teams to thrive in competitive environments, and enable them to focus more on productivity than administrative tasks.

1.2 Introduction

In today's rapidly evolving software development landscape, engineering teams face unprecedented challenges in managing increasingly complex workflows while striving to maintain productivity, collaboration, and delivery speed[1–3]. The global project management software market, valued at approximately $6.6 billion in 2022, is projected to reach $15.08 billion by 2030, reflecting a compound annual growth rate (CAGR) of 10.68% [4]. This growth highlights not only the increasing demand for project management solutions but also the urgent need for platforms that are deeply aligned with the realities of modern software engineering [5]. The digital transformation accelerated by remote and distributed work has fundamentally reshaped how developers collaborate. With over 77% of software teams adopting multiple project management or communication tools by 2023, reliance on disconnected platforms has grown significantly [3, 6]. Yet, despite this widespread adoption, many teams continue to struggle with fragmented systems that create silos, reduce efficiency, and slow down decision-making [7]. This fragmentation not only wastes valuable time but also increases the risk of misaligned tasks, delayed builds, and missed deadlines [1, 6, 8]. To address these limitations, integrated platforms have emerged as a new approach, embedding communication, analytics, and collaboration into cohesive ecosystems [9, 10]. Building on this shift, the proposed platform delivers a centralized solution that unifies project and task management, milestone tracking, real-time communication, secure file sharing, financial invoicing, meeting scheduling, and lead management. Unlike generic project management tools, it introduces automation to minimize reliance on manual updates. This ensures accurate project status, reduces human error, and saves valuable time. By combining contextual collaboration with intelligent automation, the system enables seamless workflows, greater transparency, and smarter decision-making, empowering software teams to thrive in fast-paced and competitive environments.

1.3 Problem Statement

Software teams face inefficiencies from fragmented tools that separate project management, and communication, while relying heavily on manual updates. This highlights the need for a unified, automation-driven platform that seamlessly integrates development and collaboration.

1.4 Objectives

The primary objectives of the proposed development management platform are

- To develop an integrated project and task management system that enables teams to assign tasks, manage dependencies, and track milestones in real time [1], [4].
- To provide a unified communication channel that reduces context-switching and enhances collaboration among team members .
- To simplify project-related financial management by enabling seamless expense invoicing and tracking.
- To automate task updates by connecting Git commits and pull requests directly with project status and error handling [8, 9].
- To enable meeting scheduling within projects with automatic visibility in the unified calendar.
- To improve communication efficiency via automated deadline reminders,and chat platform integration [9].
- To balance workloads by auto-assigning tasks, monitoring team capacity, and predicting potential delays [9].

1.5 Scope

The scope of this project is defined to support the above objectives:

- The platform will include task assignment, progress tracking, unified project calendars, and milestone reminders to streamline project execution and ensure timely delivery[1], [4].

- The platform will enable **real-time collaboration** through a unified communication channel that integrates chat, file sharing, and code snippet exchange, minimizing reliance on third-party tools.
- The platform will support **project-related financial management** by providing seamless invoicing and expense tracking features.
- The platform will **automate task updates** by integrating Git commits and pull requests to ensure project status is updated without manual intervention [8, 9]].
- The platform will include **meeting scheduling features** with automatic integration into a shared project calendar.
- The platform will improve communication efficiency through **automated deadline reminders and chat platform integration** [9].

  The platform will enable **intelligent workload management** by auto-assigning tasks, monitoring team capacity, and predicting potential delays [9]

1.6 Related Work

Unified software development management platforms have become essential tools for modern organizations, aiming to simplify complex workflows by consolidating project management, communication, code tracking, invoicing, and collaboration into single ecosystems. However, most existing solutions remain fragmented, forcing teams to juggle multiple disconnected tools. Despite their features, most platforms depend on manual status updates, creating gaps between actual development work and reported progress. This disconnect between Git commits, and task tracking slows teams down and reduces accuracy.

JIRA [1] and Trello[2] are widely used project management tools known for their task tracking and workflow visualization capabilities. JIRA excels in complex project tracking, reporting, and issue management, primarily catering to software development teams. Trello offers a user-friendly Kanban board interface for managing tasks and workflows. However, both platforms lack fully integrated team communication and financial management features. Our platform will incorporate JIRA and Trello's project and task management strengths, such as customizable workflows and Kanban boards, while adding native team communication and invoicing capabilities to provide a comprehensive solution Slack [3] is a leading platform for real-time team communication and collaboration, providing instant messaging, file sharing, and integration. Its strength lies in enhancing inter-team communication and coordination. We plan to build on Slack's communication features by embedding contextual messaging and file sharing directly within project and task modules, eliminating the need for external communication tools.

Basecamp offers simple project coordination and team communication but is not tailored for software development. Our system enhances this concept by aligning coordination features with developer workflows, integrating automation, and supporting coding-specific tasks [4].

Bitrix24[11] attempts to provide an all-in-one solution, covering multiple functions from communication to management. However, it suffers from a cluttered UI and is not developer-centric. Our platform takes the opposite approach: emphasizing usability, clean UI, and seamless integration[9].

By combining the strengths of these tools while addressing their limitations, our system delivers a developer-focused, automation-driven platform that unifies project execution, collaboration, and analytics. Unlike existing tools, it embeds automation and productivity features directly into the development lifecycle, ensuring seamless workflows, reduced manual effort, and improved accountability [5].

Table 1.1: Related System Analysis

Related Platforms	Strength	Limitations	Proposed Solution
JIRA [1]	Complex project tracking, reporting, issue management, customizable workflows	Lacks integrated team communication and financial management	Extend JIRA-like tracking with native communication, invoicing, and Git-linked automation
Trello [2]	User-friendly Kanban boards for task/workflow visualization	No integrated communication or financial features	Combine Kanban ease with built-in messaging, invoicing, and developer-focused automation
Slack [3]	Real-time messaging, file sharing, team coordination	Standalone tool without task/commit linkage	Embed contextual messaging and auto-generated code summaries within project tasks
Basecamp [12]	Simple project coordination and team communication	Not tailored for software development workflows	Enhance coordination features with developer workflows, Git integration, and automation
Bitrix24 [11]	All-in-one business tool with multiple functions	Cluttered UI, not developer-centric	Focus on usability, clean UI, and seamless integration with development environments

1.7 Proposed Methodology/System

The development will follow a structured, phased approach to ensure the platform delivers maximum value to software teams in a coherent and efficient manner. The methodology emphasizes iterative development, starting with the most critical components for developer workflows and progressively integrating advanced automation and AI-driven features. This approach enables continuous evaluation, feedback, and refinement at each stage, ensuring the platform evolves in alignment with team needs and agile practices [7, 13]. The first phase of development will focus on establishing the project management module, enabling organizations to create and manage projects, assign team roles, define deadlines, and set milestones. This foundational layer will provide a clear structure for work organization and ensure that all subsequent modules are built upon a well-defined project framework [6]. Once the project management module is in place, the next phase will implement task management, allowing teams to break projects into individual tasks, assign responsibilities, track progress, and manage dependencies. This phase ensures that day-to-day operations are organized and monitored effectively, providing a bridge between high-level project oversight and detailed task execution [14]. Following task management, the platform will introduce team communication capabilities, enabling real-time messaging and file sharing within the context of projects and tasks. Integrating communication at this stage ensures that collaboration is seamless, reduces reliance on external messaging tools, and fosters faster decision-making. The next phase will focus on meeting scheduling, enabling teams to create and manage meetings with visibility in a unified project calendar. This integration ensures that milestones, deadlines, and meetings are all aligned in one environment, reducing fragmentation and improving coordination [13, 15]. Subsequently, the platform will integrate financial management features, enabling organizations to connect financial metrics such as invoicing or expense tracking directly to project workflows [16]. Finally, the platform will embed automation across all modules, transforming the system from a manual tracking tool into an intelligent ecosystem. Automation will include:

- Auto-updating task statuses through Git commits and pull requests.
- Reassigning failed tasks with error logs.
- Generating automated deadline reminders and code summaries.
- Secure task closure enforced with commits, passing tests, and reviewer approval.

By following this phased, iterative methodology, the platform will evolve from a core project management solution into a comprehensive, automation-driven ecosystem. Each phase builds upon the previous one, ensuring that the system remains coherent, scalable, and aligned with user needs [17]. This approach prioritizes functionality, usability, and automation, ultimately delivering a unified solution that eliminates fragmentation, reduces manual overhead, and enhances organizational efficiency.

1.8 Tools and Technologies

Table 1.2: Technology Stack Overview

Category	**Technology**	**Description**
Frontend	Next.js[18]	React-based framework for server-side rendering, static site generation, and optimal performance
	React[19]	Component-based UI library for building interactive user interfaces
	TypeScript[20]	Type-safe JavaScript for enhanced code quality and developer experience
	Tailwind CSS[21]	Utility-first CSS framework for responsive and modern UI design
	Chart.js[22]	Data visualization library for analytics dashboards and interactive charts
Backend	Django[23]	High-level Python web framework with built-in tools for rapid application development
	Django REST Framework [24]	Toolkit for building Web APIs with serialization, authentication, and pagination
	Python[25]	Core programming language for backend development
	REST API Design[26]	JSON-based API architecture with proper status codes and error handling
Database and Storage	PostgreSQL [27]	Advanced open-source relational database for primary data storage
	AWS S3 [28]	Cloud storage service for media files, generated videos, and static assets
External APIs	Google Meet API[29]	Enables meeting creation, participant management, and artifact retrieval
	OAuth 2.0 [30]	Secure authorization protocol for third-party API access

1.9 Team Members Individual Tasks

Table 1.3: Team Members and Responsibilities

Team Member	**Responsibilities / Tasks**
Khadija	• Develop backend RESTful APIs for project and task management modules • Implement business logic for Lead Management, including API endpoint for lead generation • Implement invoicing and analytics business logic • Design and optimize database schemas for projects, tasks, invoices, and leads • Integrate Google Meet API for meeting scheduling • Implement JWT authentication, encryption, and access control • Optimize backend performance with caching and query improvements • Design system architecture and modular backend services
Muhammad Zaid	• Build responsive user interfaces for project, task, chat, meeting, and lead management modules • Develop reusable components and maintain design consistency • Integrate frontend with backend APIs for all modules • Implement state management for project, task, communication, and lead workflows • Optimize UI performance and implement lazy loading • Write unit tests for frontend modules • Ensure accessibility and responsive design across devices

Team Member	**Responsibilities / Tasks (continued)**
Moazam Ali	• Automate task status updates by linking Git commits and pull requests • Reassign failed tasks automatically with error logs for faster resolution • Task Assignment Against Individual Expertise • Send automated deadline reminders and notifications to assigned members • Generate automated daily code summaries and project progress reports
Muaz Khalid	• Collaborate on building user-friendly interface components for all modules • Implement cross-browser and mobile responsive layouts • Connect frontend components to backend services • Assist in developing and maintaining component libraries • Participate in code reviews and testing • Optimize client-side performance and handle edge cases in tasks, chats, and dashboards

Team Member	**Responsibilities / Tasks (continued)**
Zohaib Saeed	• Oversee full-stack integration ensuring cohesive flow between project, task, chat, meeting, and lead management modules • Support frontend and backend teams in implementing phased development plan • Coordinate API design, feature planning, and integration of all modules • Lead sprint planning, standups, and code reviews • Ensure scalability, security, and performance across the platform • Supervise analytics and reporting module development and integration

1.10 Data Gathering Approach

For the proposed unified business management platform, we will gather requirements by talking to target users, and professionals from various industries and software houses to understand pain points, workflows, and expectations. Questionnaires and surveys will collect broader insights on feature preferences and usability needs across different sectors. Competitor analysis and document review will further inform the functional and technical requirements by identifying gaps in existing solutions. These methods will ensure a user-centered design aligned with real-world business needs.

1.11 Timeline/Gantt chart

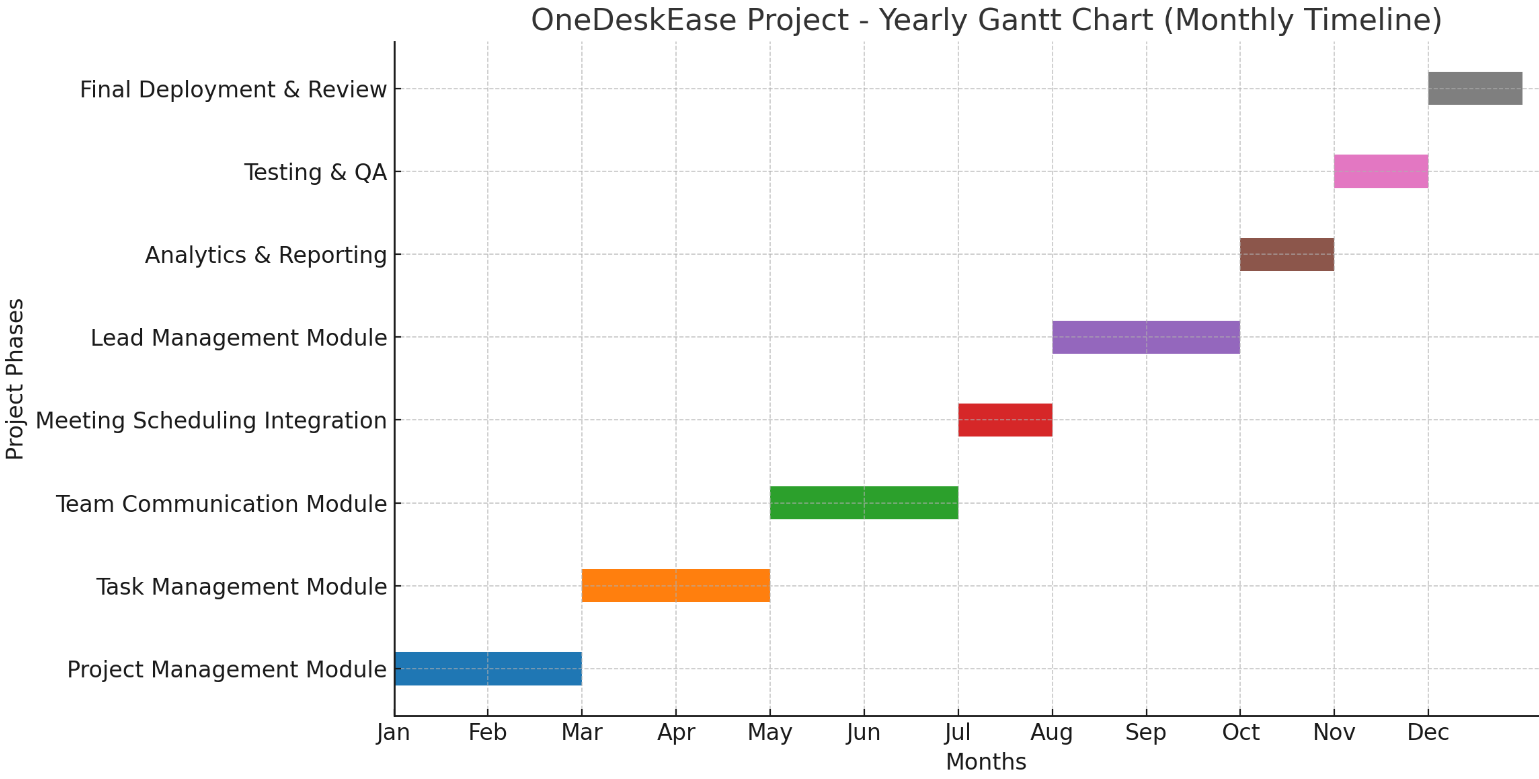

Figure 1.1: Gantt chart

References

[1] Atlassian. JIRA Software — project & issue tracking tool, 2024.

[2] Atlassian. Trello — organize anything, together, 2024.

[3] Slack Technologies. Slack: Where work happens, 2024.

[4] Business Research Insights. Business management software market size, share, growth, and industry analysis, 2024. URL https://www.businessresearchinsights.com/market-reports/business-management-software-market-118072. [Online; accessed 2-Oct-2025].

[5] D. A. Garza, A. S. Hanna, and M. J. Whited. Implementation of project management information systems (pmis) in construction. *J. Constr. Eng. Manag.*, 128(6):392–398, Nov./Dec. 2002. URL https://ascelibrary.org/doi/abs/10.1061/(ASCE)0887-3801(2002)16:1(23). [Online].

[6] V. Stray and N. B. Moe. Understanding Coordination in Global Software Engineering: A Mixed-Methods Study on the Use of Meetings and Slack. *arXiv preprint*, Jul 2020. URL https://arxiv.org/abs/2007.02328. [Online].

[7] A. O. Conforto and D. C. Amaral. Agile project management and stage-gate model: A hybrid framework for technology-based companies. *J. Eng. Technol. Manag.*, 40:1–14, 2016. doi: 10.1016/j.jengtecman.2016.02.003. [Online].

[8] GitHub. Github: Where the world builds software, 2024.

[9] Taiga. Taiga - project management platform for agile developers & designers, 2024.

[10] Invoice Ninja. Invoice Ninja – Open Source Invoicing Application, 2024.

[11] Bitrix24. Bitrix24 — free crm, project management, collaboration suite, 2024.

[12] LLC Basecamp. Basecamp: Project management software, online collaboration, 2025. URL https://basecamp.com/. [Online; accessed 2-Oct-2025].

[13] B. Fitzgerald and K. Stol. Continuous software engineering: A roadmap and agenda. 2015.

[14] S. Maheshwari and R. Varshney. Collaboration technologies: Enhancing teamwork through real-time communication. *arXiv preprint*, 2021. URL https://arxiv.org/pdf/2104.04217. [Online].

[15] R. Nikravesh, J. Azizi, and T. Golshani. Investigating the impact of an integrated project management system on financial performance. Technical report, preprint, Jun 2024. URL https://www.researchgate.net/publication/381174511. [Online].

[16] H. Kerzner. *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. Wiley, 12th edition, 2017. URL https://www.wiley.com/en-us/Project+Management:+A+Systems+Approach+to+Planning,+Scheduling,+and+Controlling,+12th+Edition-p-9781119165354. [Online].

[17] A. Boehm and H. Jain. Enterprise Resource Planning (ERP): A Review of the Literature. 2020.

[18] Vercel. Next.js: The React Framework for the Web, 2025. URL https://nextjs.org/. [Online; accessed 2-Oct-2025].

[19] Meta Open Source. React – A JavaScript library for building user interfaces, 2025. URL https://react.dev/. [Online; accessed 2-Oct-2025].

[20] Microsoft. TypeScript: JavaScript With Syntax For Types, 2025. URL https://www.typescriptlang.org/. [Online; accessed 2-Oct-2025].

[21] Tailwind Labs. Tailwind CSS – Rapidly build modern websites without ever leaving your HTML, 2025. URL https://tailwindcss.com/. [Online; accessed 2-Oct-2025].

[22] Chart.js Community. Chart.js: Simple, clean and engaging charts for designers and developers, 2025. URL https://www.chartjs.org/. [Online; accessed 2-Oct-2025].

[23] Django Software Foundation. Django: The Web framework for perfectionists with deadlines, 2025. URL https://www.djangoproject.com/. [Online; accessed 2-Oct-2025].

[24] Django REST Framework Developers. Django REST framework: Web APIs for Django, 2025. URL https://www.django-rest-framework.org/. [Online; accessed 2-Oct-2025].

[25] Python Software Foundation. Python: The official home of the Python Programming Language, 2025. URL https://www.python.org/. [Online; accessed 2-Oct-2025].

[26] Roy Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. University of California, Irvine, 2000. URL https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm. The foundational Ph.D. dissertation defining REST.

[27] The PostgreSQL Global Development Group. PostgreSQL: The World's Most Advanced Open Source Relational Database, 2025. URL https://www.postgresql.org/. [Online; accessed 2-Oct-2025].

[28] Amazon Web Services, Inc. Amazon Simple Storage Service (S3), 2025. URL https://aws.amazon.com/s3/. [Online; accessed 2-Oct-2025].

[29] Google. Google Meet API (Google Workspace Developer Documentation), 2025. URL https://developers.google.com/meet. [Online; accessed 2-Oct-2025].

[30] D. Hardt and Ed. The OAuth 2.0 Authorization Framework. *RFC 6749*, 2012. URL https://datatracker.ietf.org/doc/html/rfc6749. OAuth 2.0 Standard.