

Agile Wizards: A Project Management Application for Agile Development

Omar Basheer
Computer Science Department
Ashesi University
Accra, Ghana
omar.basheer@ashesi.edu.gh

Abigail Owusu
Computer Science Department
Ashesi University
Accra, Ghana
abigail.owusu@ashesi.edu.gh

Cyril Kujar
Computer Science Department
Ashesi University
Accra, Ghana
cyril.kujar@ashesi.edu.gh

Abstract— Our project addresses the need for an integrated project management system that combines user-friendliness with technical efficiency. Existing solutions pose challenges in user experience and workflow efficiency. Drawing insights from research and existing software, we aim to enhance communication, collaboration, and task management. Interviews and surveys with software professionals emphasize the significance of task prioritization and intuitive interfaces. By integrating the strengths of various tools, our project seeks to empower users with an innovative and user-centric project management solution.

I. INTRODUCTION

The primary objective of our project is to develop a comprehensive project management system that seamlessly combines user-friendliness with technical efficiency. This system is meticulously designed to enhance project planning and cater to the specific requirements of software development teams and stakeholders.

Within the realm of project management, issue tracking holds a pivotal role, enabling teams to monitor, meticulously organize, and effectively prioritize tasks, bugs, enhancements, and more. However, the current landscape of existing solutions presents various challenges related to user experience and technical efficiency.

These challenges underscore the pressing need for an improved and more intuitive project management system that addresses these concerns and elevates the overall efficiency of project management processes. Our project goes beyond traditional issue tracking and aims to offer a holistic solution that empowers teams to navigate complex projects while maintaining a user-friendly and technically efficient environment. In this literature review, we delve into the specific challenges of existing solutions, emphasizing the unique value proposition of our project in meeting these demands.

II. RELATED WORK

To justify the development of our project, we conducted a thorough analysis of related articles from reputable journals and examined notable existing software technologies in the domain.

The first research paper, authored by Lanubile, Ebert, Prikladnicki, and Vizcaíno, discusses the crucial role collaboration plays in software engineering. It underscores the importance of integrating collaboration tools and issue-tracking systems, aligning well with our project's aim. The authors highlight the challenges faced by global collaboration, emphasizing the need for integration between collaboration tools and issue tracking systems. This emphasizes the need to

seamlessly integrate reporting features within collaboration tools, a key feature of our system. By doing so, the system can enhance communication and collaboration among team members while effectively managing tasks. The paper identifies standard tools essential for effective collaboration in software engineering, including version control systems, issue trackers, and communication tools [1]. This research advocates for the integration of trackers within collaboration tools for reporting issues or bugs. Additionally, it highlights the integration of communication tools within collaboration tools, without overwhelming users with notifications. This led to the contemplation of the integration of direct communication in our software.

Furthermore, the integration of version control systems within collaboration systems prompted consideration for potential version control integration with systems like Git. The paper's support for trackers within collaboration tools for issue report aligns with the project's objective of seamlessly incorporating issue reporting capabilities within our system. Similarly, the highlight on integrating communication tools within collaboration tools without overwhelming users corresponds to the system's focus on efficient communication features that enhance collaboration without being intrusive.

The second research paper, authored by Calefato, Giove, Lanubile, and Losavio, presents a case study exploring the optimization of development and communication tools to support agile practices in an Italian software company. This study resonates with our focus on seamless communication integration within our system. The study revealed inconsistencies in the use of communication tools such as email, Skype, and Slack, but highlighted that emails were the most used means of internal communication among developers [2]. The authors advocate for a refactoring of Jira boards into high-level and detailed boards to cater for the information needs of different stakeholders [2]. Incorporating a similar feature in our software system would allow it to cater to varying information needs in project management.

The paper also promoted task automation and the integration of Slack with project management software. The use of emails as the primary means of internal communication and the potential automation of emails for task notifications were of interest and could be applied to our system. Also, the need to distinguish between high-level and detailed boards calls for the contemplation of the inclusion of such a feature as well.

Mesaros et al. explore the application of Agile Scrum project management in the food processing industry, focusing on how project breakdown and representations on Jira boards. The paper describes the Agile process, then explores the

decomposition of complex projects into Epics made up of user stories and tasks [3]. This offers valuable insights into the application of Agile Scrum in the context of our system. Issues and tasks can be broken down for efficient tracking and management. The research offers a practical application of agile project management in a specific industry. This provides guidance that can aid in refining our project's implementation of Agile Scrum methodology.

In addition to academic research, we explored notable existing software systems that have made defining contributions to issue tracking and project management. These systems serve as valuable benchmarks and sources of inspiration for our project.

The first notable software is Jira by Atlassian. This software stands as a prominent issue and project management software widely adopted within industries. It supports various software design methodologies, including Scrum. Our project aims to draw on Jira's strengths to enhance our own system. Jira's effectiveness in issue and project management serves as a guide in optimizing our tracking and project planning features. Many Jira users often struggle with its complexity, making it less user-friendly, especially for startup teams or those without technical expertise. In contrast, Trello offers a more intuitive design but falls short when handling complex projects. Our project seeks to merge Trello's user-friendly interface with Jira's technical efficiency, creating a simplified environment where users can easily manage complex tasks.

GitHub and GitLab are industry-leading platforms that provide version control and collaborative software development support. They provide centralized project repositories, issue management services, and orchestrated workflows. Our project aims to provide integration with these platforms to leverage their services. Learning from features and user experience of these systems will guide us in developing a robust issue tracking system with enhanced version control capabilities, optimizing the collaborative development process.

III. REQUIREMENTS ELICITATION AND ANALYSIS

The survey conducted among a diverse range of stakeholders including software engineers, project managers, and students has yielded valuable insights into the challenges, preferences, and suggestions related to issue tracking systems for project planning. Among software engineers, one recurring challenge is the inefficient workflow support offered by existing systems, with instances where these systems fail to align with project-specific workflow requirements, potentially leading to planning and execution deficiencies. Scrum managers emphasized the need for seamless integration with Agile methodologies to ensure efficient sprint planning and execution. Chief technology officers emphasized the importance of scalability, security, and integration capabilities within the system to align with organizational technology standards. Project managers emphasized the need for comprehensive project progress visibility and effective collaboration features. The Quality Assurance Team stressed the necessity for clear bug tracking and testing integration. IT support highlighted the importance of system stability and ease of maintenance. UI/UX designers emphasized the need for an intuitive interface that supports efficient design collaboration. Interestingly, some respondents across these roles reported no significant issues, finding the

basic functionalities of the systems satisfactory. However, a common concern expressed by several participants across different roles is the perception of software bloat and unclear navigation in these systems, indicating a need for streamlined and intuitive interfaces.

In terms of the most valuable features or functionalities in issue tracking systems, the ability to prioritize tasks emerged as a consensus among respondents. This feature was seen as instrumental in maintaining a focus on critical tasks, helping to improve overall project efficiency. Additionally, respondents emphasized the importance of tracking the various states of tasks and having visibility into task assignments. Roadmaps were also highlighted as valuable, indicating the need for long-term project planning and strategy alignment.

Regarding suggestions for improving user experiences in issue tracking systems, a diverse range of perspectives emerged. While some suggested the provision of supplementary training resources, such as videos or online presentations, to enhance user proficiency, others felt that the current systems meet their needs adequately. Furthermore, many participants called for a cleaner and less cluttered user interface (UI), emphasizing the importance of an intuitive and user-friendly design to enhance overall usability.

Overall, the survey findings present a multifaceted understanding of the challenges and requirements related to issue tracking systems, friendly interfaces, git integration, reflecting the diverse needs and preferences of software engineers, project managers, and students. These insights can serve as a valuable foundation for further research and development efforts aimed at optimizing issue tracking systems for improved project planning and management across various user groups.

IV. REQUIREMENTS SPECIFICATION

A. User Requirements

- Upon logging in, users should see a dashboard displaying ongoing projects and their progress, providing a quick overview.
- Users should have the option to use either a high-level or detailed project board, depending on their role within the organization.
- Users want to easily track the progress of their tasks and projects through clear task statuses and priorities.
- Users desire the ability to create, prioritize, and manage tasks within a project, including setting deadlines and assigning team members.
- Users want a clear visual indication of task priorities to easily identify and focus on high-priority tasks.
- Users expect an intuitive and user-friendly interface that simplifies navigation and enhances overall usability.

- Users should be able to seamlessly integrate the system with Git for efficient version control and collaboration.
- Users expect to receive automated email notifications for code updates and important repository activities to stay informed.
- Users look for features that facilitate seamless collaboration among team members, improving overall project efficiency and communication.
- Users expect the system to perform reliably, ensuring smooth and uninterrupted access even during peak usage times.
- Users value the security and privacy of their data within the system and expect appropriate measures to be in place to safeguard it.
- Users anticipate the system to be compatible with various devices and browsers to ensure access from different platforms.

B. System Requirements

- The system should provide a clear display of all ongoing projects and their progress for quick review upon logging in.
- The system should support both high-level and detailed project boards to cater to different levels of project overview and information.
- Users should have the ability to create new tasks for a specific project, set task priorities, define task status, allocate assignees and supervisors, and set deadlines for each task.
- For each project, users should be able to view the status of each task within the project, including their priority, progress, and other relevant details.
- Users should be able to easily view and identify the priority level of each task to effectively manage their workload and prioritize their actions.
- The system should seamlessly integrate with Git for version control and automate email notifications for code updates and repository activities.
- When a developer's code passes all stages of review by the project manager and is pushed to the main branch, the system should automatically mark the associated task as completed on the AgileWiz board.
- An automated email notification should be sent to all supervisors and assignees upon completion of a task or when there are updates related to a task.

C. Functional and Non-functional Requirements

1.1 Functional Requirements

- FR001: The system shall display a clear progress overview of all ongoing projects for quick review.
- FR002: The system shall support both high-level and detailed project boards, allowing users to choose and switch between them.
- FR003: Users shall be able to view task details, including priority, and manage tasks within a project (create new tasks, set priorities, statuses, deadlines, assignees, and supervisors).
- FR004: Users shall be able to easily view the priority level of each task to effectively manage their workload.
- FR005: The system shall integrate seamlessly with Git for version control.
- FR006: The system shall send automated email notifications to supervisors and assignees for code updates and repository activities.
- FR007: Upon successful code review and integration, the system shall automatically update the task status to 'completed' on the AgileWiz board.

1.2 Non-functional Requirements

- NFR001: The system shall be intuitive to use. Upon onboarding, 95% of users, within the first 30 minutes, should be able to navigate through the interface and perform basic functions without assistance.
- NFR002: The system shall respond to user actions within 2 seconds for at least 95% of interactions, regardless of the user load.
- NFR003: The system shall have a maximum downtime of 30 minutes per month for scheduled maintenance and updates. Unscheduled downtime should not exceed 10 minutes per occurrence.
- NFR004: The system shall comply with industry-standard security protocols, ensuring data encryption in transit and at rest. Unauthorized access attempts shall not exceed three per minute.
- NFR005: The system shall support a user base increase of 200% within a year while maintaining consistent performance and response times.
- NFR006: The system shall be compatible with the latest versions of Chrome, Firefox, Safari, and Edge browsers, ensuring consistent functionality and appearance.

- NFR007: The system's codebase shall have a maintainability index of at least 80, as measured by standard code analysis tools, indicating ease of future maintenance and updates.

SYSTEM DESIGN

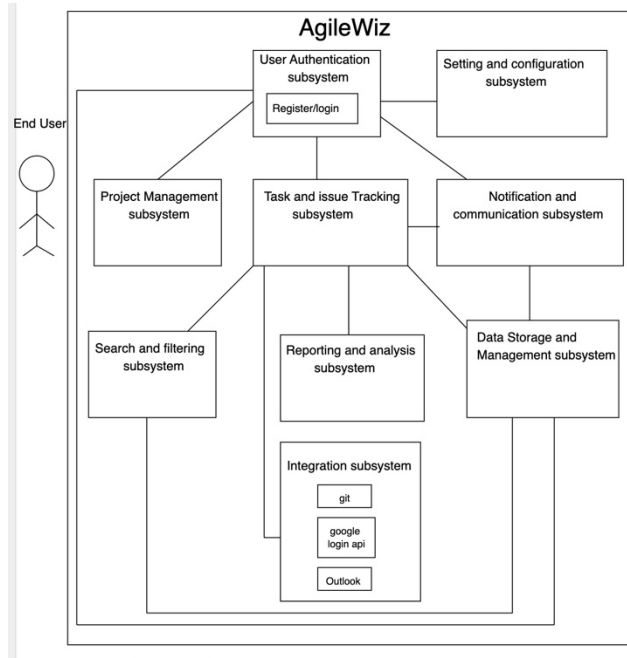


Figure 2: High-Level System Diagram

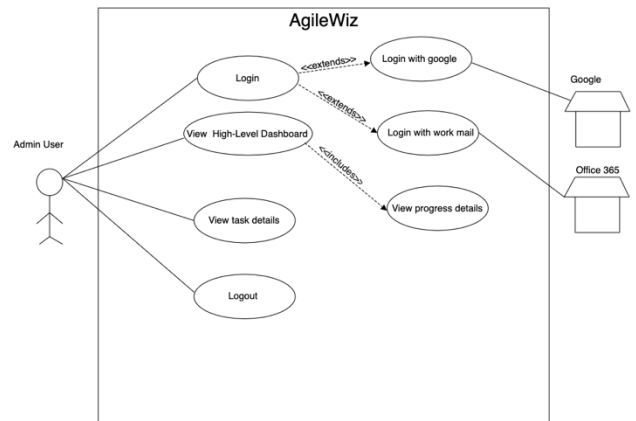


Figure 1: Organization Manager Use Case

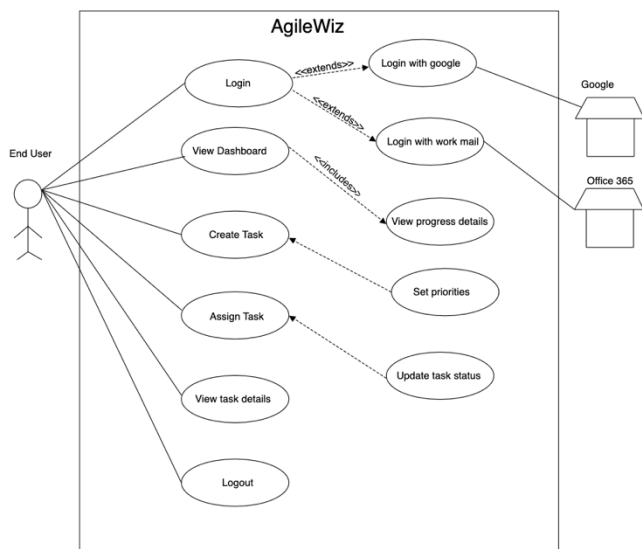
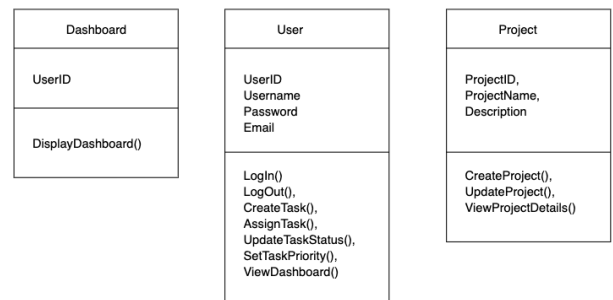


Figure 3: End User Use Case Diagram

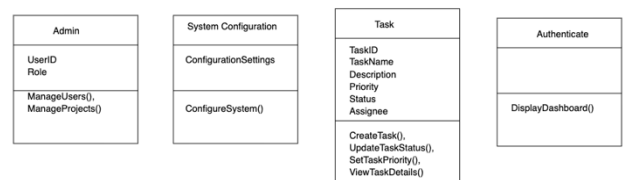


Figure 4: Class Diagram Objects

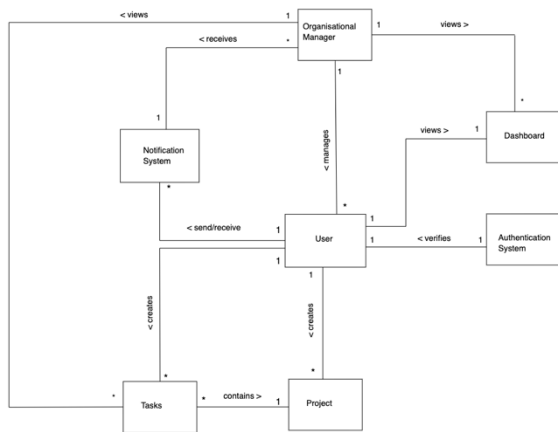


Figure 5: Class Diagram

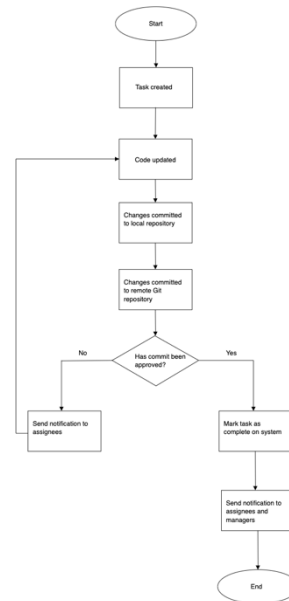


Figure 6: Flow Diagram for Git Integration and Task Status Update

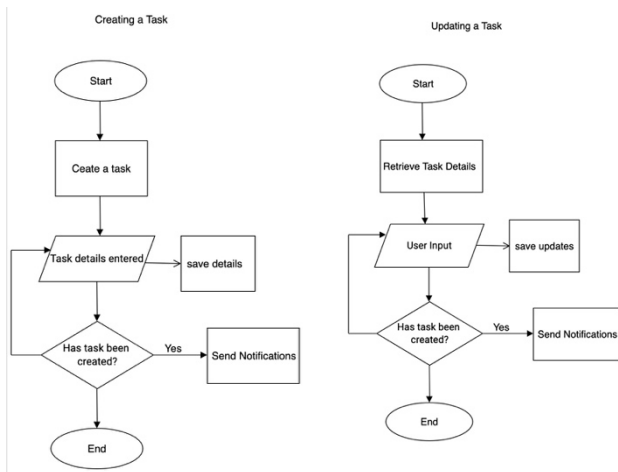


Figure 7: Flow Diagrams for Creating and Editing Tasks

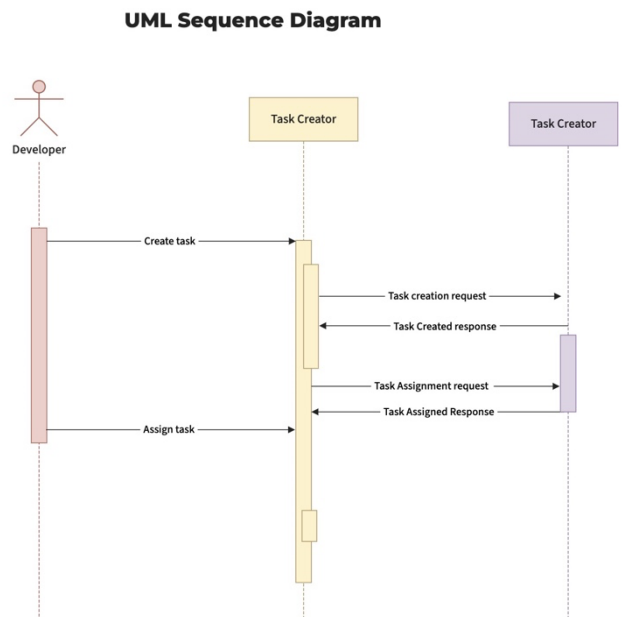


Figure 8: Sequence Diagram for Task Creation

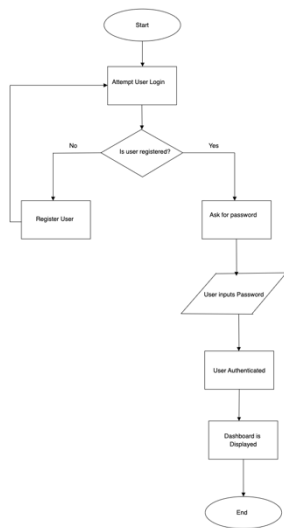


Figure 9: Flow Diagram for User Registration and Login

CONCLUSION

In summary, our project emerges as a response to the challenges posed by existing project management systems and

the need for a more user-friendly, feature-rich, and integrated solution. We have drawn inspiration from esteemed researchers, existing software technologies, and the valuable insights of software engineers, project managers, and students. Our goal is to bridge the gap between complex systems and user-friendly designs, optimizing version control and collaborative development while emphasizing effective communication within software development teams. By doing so, we aim to empower users with a streamlined and innovative tool that enhances their project planning and management experience.

Link to Figma Implementation:

<https://www.figma.com/file/xovFkas7QSuTyeIxCsYDyV/Untitled?type=design&t=DstzcaHdY9L7og9o-6>

REFERENCES

- [1] F. Lanubile, C. Ebert, R. Prikładnicki and A. Vizcaíno, "Collaboration Tools for Global Software Engineering," *IEEE Xplore*, no. 2, 2010.
- [2] F. Calefato, A. Giove, F. Lanubile and M. Losavio, "A Case Study on Tool Support for Collaboration in Agile Development," *IEEE Xplore*, 2023.
- [3] M. D. Mesaros, A. Coroian, O.-A. Mastan, A.-L. Longodor, K. Zoltan and M. Liliana, "AGILE Project Management Using Jira in Processing of Food Industry," *IEEE Xplore*, 2023.