# ClientView Project Report

ALEX URBINA, Virginia Polytechnic Institute and State University, U.S.A.
SEAN SULLIVAN, Virginia Polytechnic Institute and State University, U.S.A.
SRI VANGARU, Virginia Polytechnic Institute and State University, U.S.A.
YUXUAN GUO, Virginia Polytechnic Institute and State University, U.S.A.

Clients often struggle to track the progress of their software projects due to the complexity of existing project management tools. Technical jargon and the lack of clear visual representations make it challenging for non-technical clients to grasp the current status of a project. Additionally, real-time communication of updates is often inefficient, leading to delays in feedback and decision-making. To address these issues, we propose a simplified, real-time project tracking system that pulls data directly from internal tools like Jira. Our solution provides a user-friendly dashboard that organizes project information into key features, displaying their current status, estimated completion dates, each task's importance, and other relevant updates. This approach offers enhanced transparency, ease of use, and real-time updates, allowing clients to stay informed without needing technical expertise.

**ACM Reference Format:**
Alex Urbina, Sean Sullivan, Sri Vangaru, and Yuxuan Guo. 2024. ClientView Project Report. In *Proceedings of CS3704: Intermediate Software Design (CS3704 '24).* ACM, Blacksburg, VA, USA, 6 pages. https://doi.org/10.1234/5678901.2345678

## 1 Introduction

In the software development industry, clients and stakeholders often face significant challenges when tracking project progress [4]. The complexity of existing project management tools, such as Jira, often leads to confusion, especially for clients who lack a technical background. While effective for development teams, these tools present information in a way that is not easily understood by clients. As a result, it is difficult for clients to understand the current status of the project, leading to frustration and a feeling of being disconnected from the development process.

In addition, project updates are often communicated through meetings, emails, or detailed reports filled with technical jargon, which makes it difficult for clients to provide timely feedback. The lack of real-time communication and visibility can lead to delayed decisions, inconsistent expectations, and potential project setbacks. Moreover, the lack of clear visual representations further makes it difficult for clients to monitor key project milestones and identify critical obstacles that may affect the project timeline.

### 1.1 Motivating Example

Consider a scenario where a non-technical client, Sarah, is sponsoring the development of a customer relationship management (CRM) system. Using traditional project management tools, Sarah struggles to understand the technical

Authors' Contact Information: Alex Urbina, Virginia Polytechnic Institute and State University, Blacksburg, U.S.A., ; Sean Sullivan, Virginia Polytechnic Institute and State University, Blacksburg, U.S.A., ; Sri Vangaru, Virginia Polytechnic Institute and State University, Blacksburg, U.S.A., ; Yuxuan Guo, Virginia Polytechnic Institute and State University, Blacksburg, U.S.A., .


Manuscript submitted to ACM

updates and gauge the project's progress. She frequently needs to schedule meetings with the development team to get clarifications, causing delays in decision-making.

With ClientView, Sarah accesses a simplified dashboard that immediately presents the overall project progress. She can view current sprint activities organized in business-friendly terms, visualize task dependencies through intuitive graphs, and track estimated completion dates for key features. Most importantly, she can understand the business impact of ongoing development work without requiring technical translation. This scenario demonstrates how ClientView bridges the communication gap between technical teams and non-technical stakeholders, enabling more efficient project monitoring and decision-making. In this manner, ClientView also demonstrates a clear benefit to the software engineers as well. As they would no longer need to take extra time to de-jargonize their current project state for the clients, this would help enhance their productivity.

## 2 Related Works

As stated before, the idea behind creating a progress tracker software engineering projects is not new. Jira is a very commonly used program that allows issue tracking and agile project management [2]. Although our program has a very similar goal, it looks to improve upon this idea and remedy some of the issues that have risen as a result of widespread usage of Jira. Some of those issues include steep learning curves, performance issues, and overwhelming customization.

Our program would aim to streamline the information so that clients with no background in software engineering can accurately understand the state of the projects that they support. Additionally, there are other project management programs such as Zoho Projects [3] and Asana [1], though they suffer from similar issues. ClientView's novelty lies in its focus on simplifying technical information specifically for non-technical stakeholders while maintaining integration with existing development tools.

## 3 Implementation and Testing Approach

### 3.1 High-Level Design

We chose a Model-View-Controller (MVC) architecture for ClientView because it facilitates a UI to interact with users and stores and retrieves information as needed, as used by most modern web applications. The Model layer will handle Jira API integration and data management, allowing for real-time synchronization of project data. The View layer will handle creating simplified, client-friendly dashboards and visualizations, making complex project information easy to understand for non-technical stakeholders. The Controller layer will manage transforming technical Jira data into simplified visualizations, handling the main requirement of making project progress more accessible to clients. This clear separation makes the system highly maintainable, as changes to the Jira integration won't affect the client interface, and new visualizations can be added without modifying the backend. The UIs of some of the main screens are as shown in Figures 1 and 2.

### 3.2 Implementation Process

This project aims to enhance the software engineering process as a whole by helping the clients that are sponsoring a project have better insight into the internals of it, regardless of which specific process is being used by the developers of that project. However, the software engineering process that is planned to be used in the development of this app itself will be Prototyping. This is because the best possible user experience that can be obtained through using our ClientView app will vary, but we can better understand how to reach that point by continuously receiving feedback

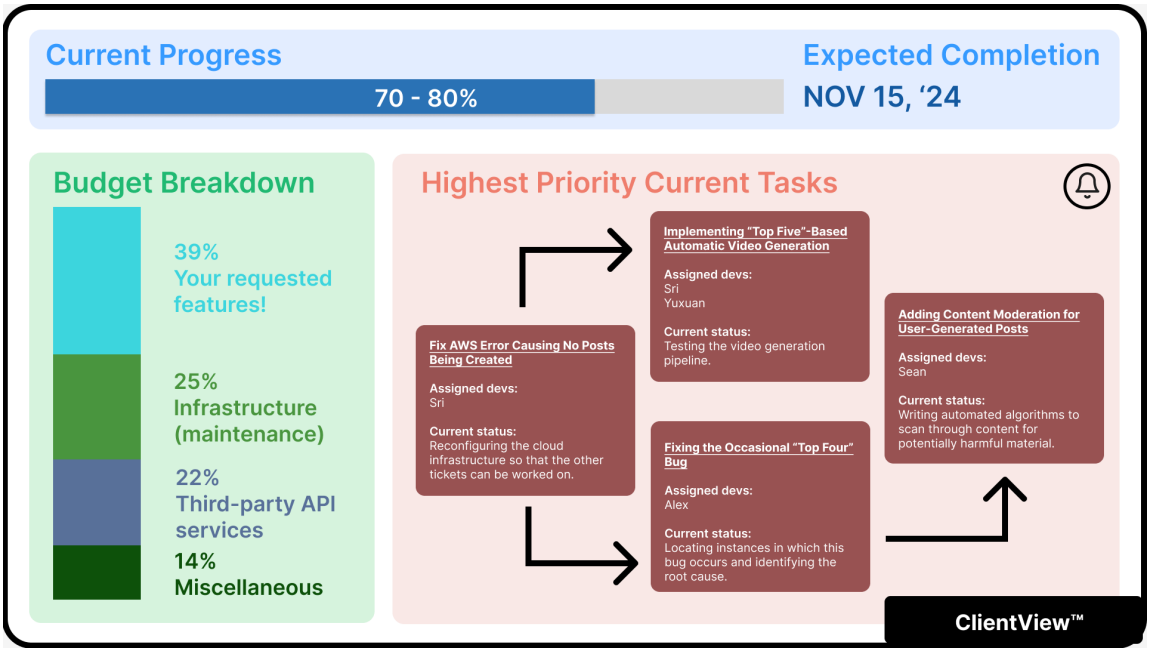


Fig. 1. The user interface of the home ClientView dashboard.

from our end users, who will be a combination of software developers and clients that are paying for products to be made by them. In other words, while developing our app, we will continuously build prototypes aligning with what we currently see as the best product for the developers and clients to use, maintaining communication with them on the current state of our app to iterate and improve it.

### 3.3 Testing Approach

The testing strategy for ClientView uses multiple layers of validation and verification, combining both white box and black box testing approaches. Our plan utilizes unit tests to verify the accuracy of data and integration tests to ensure reliable connectivity with the Jira API.

Our black box testing strategy focuses on validating the system's functionality from an end-user perspective, particularly emphasizing the client dashboard interface. The test plan includes:

(1) **Dashboard Visualization Testing:** Verifying that project progress is accurately displayed through progress bars, charts, and task lists. Test cases validate that percentage calculations match the underlying data and that task status updates are reflected correctly in real-time.

(2) **Data Synchronization Validation:** Testing the accuracy and timeliness of data synchronization between Jira and ClientView. This includes verifying that updates in Jira appear within the specified 30-second window and that all task attributes are correctly mapped.

(3) **User Interface Navigation Testing:** Ensuring all UI elements are accessible and function as intended, with particular attention to the task dependency visualization and timeline features. Test cases cover both standard navigation paths and edge cases.

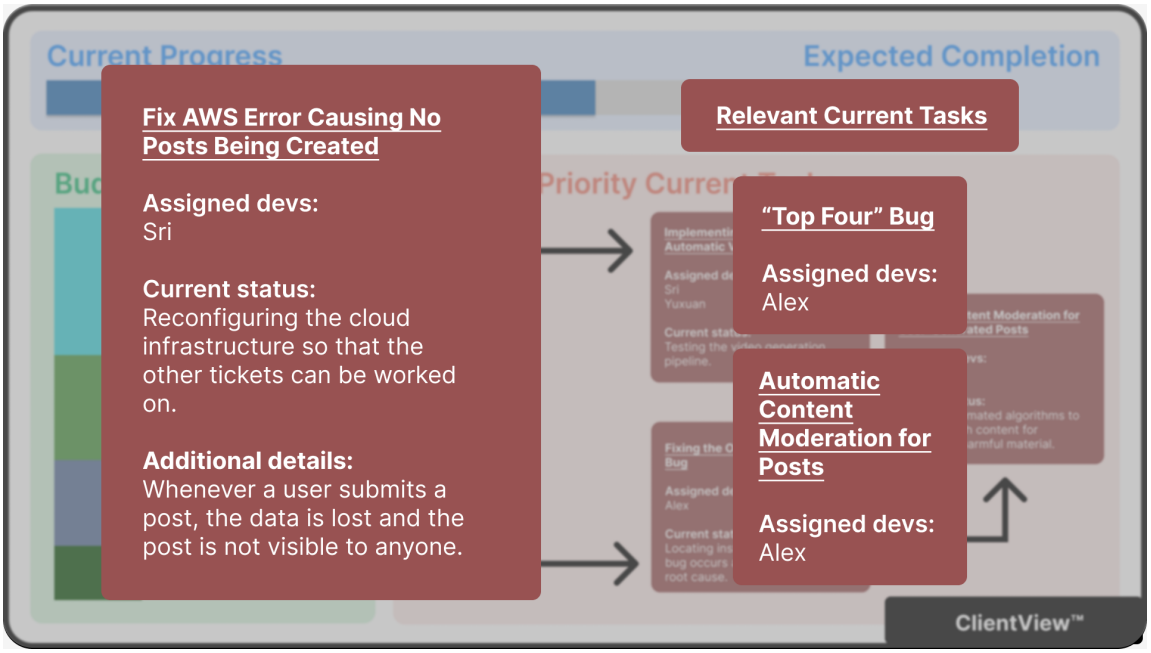


Fig. 2. The user interface after opening an example task on ClientView.

(4) **Performance Under Load:** Validating system responsiveness when multiple users access the dashboard simultaneously and when processing large datasets from Jira.
(5) **Error Handling and Recovery:** Testing system behavior when encountering API failures, network issues, or invalid data formats, ensuring appropriate error messages are displayed to users.

These tests, along with user acceptance testing with actual clients, could provide valuable insights into real-world usage patterns and requirements. Through this comprehensive testing approach, we ensure that ClientView meets both its technical specifications and user experience goals.

## 4 Deployment Plan

To deploy our ClientView project, we would follow a very structured approach by incorporating key concepts that we discussed in class such as requirement elicitation, SOLID design principles, high-level design, low-level design, and software processes. With requirement elicitation, we would gather feedback from both technical teams and non-technical clients to unsure a user-friendly dashboard adequately addresses the clients pain points. Then by using SOLID design principles, such as modular design and separation of concerns, we would create a system where each component operates independently and collaboratively. For high-level design, the architectures could include modules for pulling data from internal tools like Jira. This could then be presented to the user through the dashboard. The low-level design will focus on implementing specific components, such as API's for data extraction, and back-end services for data synchronization. During deployment, we would use a roll out approach, which ensures that the system would integrate well with existing tools.

Maintenance will involve monitoring system performance, and addressing user-reported issues. Lastly, by following software processes, like Agile development, we will adapt a method for iterative updates. This allows for quick refinements to be made, ensuring a reliable experience for the clients. Overall, this approach elicits a scalable, maintainable, and transparent solution for client project tracking.

## 5 Conclusion

In conclusion, our ClientView project has successfully developed a user-friendly solution to simplify project tracking for non-technical clients. By addressing the challenges of technical jargon and lack of clear visual representations, our system provides enhanced transparency, real-time updates, and ease of use. However, like any system, it has certain limitations that must be acknowledged.

The first limitation lies in the synchronization with real-time data from Jira. While the system is designed to refresh data every 30 seconds, intermittent delays or inaccuracies in synchronization can impact the reliability of displayed information. This occasional latency may affect client trust in the system's real-time capabilities. Secondly, the current reliance on Jira limits the platform's usability for clients who use alternative project management tools. This dependency reduces its accessibility for a broader audience. Additionally, while we have simplified the user interface, some non-technical clients may still face difficulties understanding certain visual elements or navigating the dashboard effectively. These challenges highlight the need for onboarding sessions or user training to enhance usability. Lastly, scalability remains an issue. For projects involving a large number of tasks or highly detailed workflows, the system may experience delays in data processing and rendering, which can hinder the user experience.

In the future, we have identified several promising avenues for future work to address these limitations. To broaden the platform's applicability, we plan to expand integration capabilities to support additional project management tools, making ClientView versatile and appealing to a wider range of clients. Advanced data visualization features will also be introduced to provide clients with deeper insights into project workflows and potential risks. To improve accessibility, we aim to develop a mobile application that mirrors the dashboard functionality, enabling clients to track their projects conveniently from any location. To foster continuous improvement, we plan to implement a feedback system that allows users to share their suggestions and concerns directly within the application. This feature will ensure that the platform evolves based on user needs and preferences. Moreover, we intend to optimize the system's scalability by refining the data caching mechanism and enhancing task loading logic. This will enable the platform to handle large datasets and concurrent users more effectively. Enhanced security measures, such as end-to-end encryption and multi-factor authentication, will also be implemented to safeguard sensitive project data.

By addressing these limitations and implementing these enhancements, ClientView has the potential to evolve into a comprehensive and robust project tracking solution. It will cater to the changing needs of users while maintaining its core focus on simplifying project management for non-technical clients.

## Acknowledgments

We would like to extend our thanks to Dr. Chris Brown for providing us with background knowledge on various software engineering processes.

## References

[1] Asana. 2024. Coordinate projects-online tools, Templates & App • Asana. https://asana.com/uses/project-management
[2] Atlassian. 2024. Discover the features that make Jira so easy to use. https://www.atlassian.com/software/jira
[3] Zoho Coorporation. 2024. Zoho projects: Online project management platform. https://www.zoho.com/projects/
[4] Stephen A Leybourne, Vijay Kanabar, and Roger David Hand Warburton. 2010. Understanding and overcoming communications complexity in projects. https://www.pmi.org/learning/library/overcoming-communications-complexity-ambiguity-projects-6631