Project Management System Report

Tyler J. Latshaw

Southern New Hampshire University

March 6, 2022

Table of Contents

Project Management System Report

Based out of Lititz, PA, WebstaurantStore is the e-commerce division of Clark Associates - the number one kitchen and restaurant supply distributor in the world (Clark Associates, n.d.). With an infrastructure to support a growing business of more than 300,000 products, WebstaurantStore's in-house development team of around 300 employees handles nearly all of Clark Associates' programming. They support employees across the United Sates, despite lacking a project management system to track current and future development projects (Clark Associates, n.d.; W. Edwards, personal communication, January 3, 2022). The company recently approved the development of a custom project management system to alleviate the development department's strain and stakeholders' stress attributed to not having a system.

Project Design

The design and implementation of this project will be unique to WebstaurantStore meeting the specific needs of the company and its stakeholders. The project will feature an innovative approach that will provide users an easy way to view and manage current projects at the company. While the term "project management system" is usually associated with complex products like Microsoft Project, this new application will be easier to use and encompass only the functionality the company needs. This system was intentionally designed for the needs of the company, so it avoids the level of depth that is typically seen with a fully robust system. A high-level overview of the projects was deemed more suitable for the company's needs and will significantly impact the quality of life of the project management team and the stakeholders.

Problem Statement

WebstaurantStore's project management team of business analysts has historically been able to manage the day-to-day operations of leading development work through a series of

spreadsheets. However, this has become increasingly complicated as the number of business analysts continuously grows along with the number of projects. When the team had only a handful of employees, there was not only less information, but it was also more widely shared as most analysts were involved in several aspects of a given project. To date, the team has grown to more than 30 individuals, with teams structured by their respective business departments (W. Edwards, personal communication, January 3, 2022). While this team structure allows for more in-depth knowledge of the systems, coupled with lacking an actual project management system, it often means that the analysts are not fully aware of the status of each other's projects across more than 40 active and more than 100 upcoming projects. While the team does maintain a series of spreadsheets, they are often outdated and inaccurate due to the complexity of managing separate spreadsheets and the time it takes to edit each individually.

Apart from not having a central source of knowledge for projects, the impact of not having a project management system is seen beyond the project management team with other stakeholders and roles. While the development managers have access to the spreadsheets, the data is frequently outdated because of the time needed to update it by the analysts so the managers are unable to reliably use it for resource planning. Further, the business stakeholders, such as various directors and company executives, do not have a clear understanding of current and upcoming projects as they do not have access to the spreadsheets for road mapping for their departments. Likewise, the product owners on the project teams do not have details about their own projects that they are responsible for.

Stakeholder Needs

For this project, there are multiple stakeholders, each with their own needs and requirements. The way a business analyst will be using the system will look much different from

how a director in a business department will be using it. The stakeholders that have been identified for the project management system include the business analysts, the business analyst managers, the development team members, the development department managers and directors, and the business stakeholders.

Business Analysts

The business analyst team will be one of the primary users of the new project management system and will be directly editing the project information as opposed to other stakeholders that will just be reviewing the data. At WebstaurantStore, the business analysts lead development projects, similar to a project manager or coordinator. As such, the main requirement of the team is to have a website that will allow them to track critical information about their projects, such as who is working on the team, project milestones, and important links for the project, such as the project backlog or testing sites. Additionally, the business analysts need to have access to view and edit other projects in the event that they are covering for another analyst or are transitioning to that team. Finally, they will also need to use the system to submit their weekly status updates about each project.

Business Analyst Managers

The business analyst managers are the direct managers of the business analysts previously mentioned. Requirements for these managers will closely resemble the needs of the business analysts as they might be the lead on a project from time to time. Additionally, the managers oversee resource allocation for projects, so they will need a way to quickly view any of the projects their employees are assigned and all of the upcoming projects so they can plan based on the projections for the completion of current projects. The managers do not directly add status

updates, but they do review the entries of their employees so they will need access to view and edit updates (N. S. Musser, personal communication, January 13, 2022).

Development Team Members

WebstaurantStore development teams consist of developers, database administrators, quality assurance (QA) engineers, designers, and software architects. Despite the wide variety of job functions, each of these roles will have similar requirements for the system. Given that the business analysts will be maintaining the system and its data, the development team members will only need access to view the application but will not need to edit the information directly. The development team members will only check the system for periodic updates on their projects or information about upcoming projects. Of the stakeholders, they will likely be accessing the application the least and more if they are curious about project statuses as opposed to using it for reporting purposes (V. Sanchez, personal communication, January 20, 2022).

Development Managers and Directors

To date, the development department management team at WebstaurantStore and Clark Associates comprises of a chief information officer, vice presidents for software engineering and design, several directors, and more than a dozen managers who play an active part in their teams' development projects from a resource management standpoint. All managers, directors, and senior managers will be using the project management system in two manners. First, all of the stakeholders mentioned will be looking at the system to see the overall status of projects, similar to the other stakeholder groups. While each manager will have access to view all projects, the default view should only show projects under their hierarchical structure for projects where their reports are assigned (N. S. Musser, personal communication, January 13, 2022). Secondly, the

managers will need to be able to use the tool to plan out resources for upcoming projects, much like the business analyst managers.

Business Stakeholders

The final group of stakeholders comprises all users outside the development department, such as product owners, directors, and vice presidents. Other departments in the company include Customer Solutions, Logistics, Content, and Purchasing, among others. The main requirement for these business users will be to have a way to view current and upcoming projects for their department and view key information surrounding the project. This will eliminate the need to go to the business analysts every time they are looking for an update. Beyond this, the users will need a simple list that will show all of the points of contact for a project so that they can reach out to the appropriate contacts if and when a bug is discovered in an application or system, or they have an enhancement request (N. S. Musser, personal communication, January 13, 2022). Business users will only have access to view the system. They will need to request access from the business analyst team as the project may contain potentially sensitive information down the road.

Overview of Design

The project's overall design will be a standalone website that contains features to meet the stakeholders' needs. The website will feature four primary areas, including a dashboard highlighting current projects with the ability to filter for projects, a project page with the details on a project, a simple project list with points of contact, and pages to collect and view all of the business analysts' status updates. There will be secondary pages that will house settings, preferences, and global administration of the site, such as adding new users to access policies. The entire website will be run off a new database to store all of the system's data. The website

will be self-contained but will connect to some existing applications at WebstaurantStore through API endpoints. An example of the project page is shown in Appendix B: Project Details Page.

The project will be broken into four phases: requirements gathering, design and documentation, development, and user acceptance testing. Now that key stakeholders have approved the project at WebstaurantStore, it will be hypothetically assumed the project had a definitive start date of 1/3/2022. The project will last approximately ten weeks and will end on 3/11/2022 when deployed for use. The first phase, requirements gathering, will last one week and will be focused on researching the project, interviewing stakeholders, and discovering business rules and expectations for the project. The first milestone will be when the project is approved at the end of that week. Once approval is secured, phase two will begin by designing the data model and prototypes and creating the backlog for the developers. The second milestone will occur during this phase when the design is finalized, allowing the backlog to be created.

The third phase, development, will be seen in three iterations, lasting two weeks. The developers will code the entire system and pass it through quality assurance testing during this time. The third milestone will be at the end of the six weeks when development is complete. To ensure the system is accurate and works as anticipated, stakeholders will have one business week or five days for user acceptance testing. The final milestone, the project's release, will occur on 3/11/2022, which will officially end the project after ten weeks.

The primary challenge of this project is the time it needs to be completed. The entire project should be completed within a ten-week window based on the requirements. For a large development team, this may be an easy target to meet, but for a smaller project team with only a handful of developers, it will be trickier to manage. Given that the entire project and every single task is on the project's critical path, there is significant risk associated with the project plan. If

any task were to run over the allotted time, a later dependency task would need to compensate and be shortened accordingly. If not, the entire project will run late and be delayed. A Gantt chart of the project plan is shown in Figure 1.

Task Name	Duration	Start	Finish	Predecessors
Project Management System Implementation	**50 days**	**Mon 1/3/22**	**Fri 3/11/22**	
Requirements Gathering	**5 days**	**Mon 1/3/22**	**Fri 1/7/22**	
Organize Project Documentation	0.5 days	Mon 1/3/22	Mon 1/3/22	
Conduct Research	1.5 days	Mon 1/3/22	Tue 1/4/22	2
Interview Stakeholders	3 days	Wed 1/5/22	Fri 1/7/22	3
Secure Approvals	0 days	Fri 1/7/22	Fri 1/7/22	4
Design & Documentation	**10 days**	**Mon 1/10/22**	**Fri 1/21/22**	**1**
Design Data Model	2 days	Mon 1/10/22	Tue 1/11/22	
Create Prototypes	3 days	Wed 1/12/22	Fri 1/14/22	7
Finalize Design	0 days	Fri 1/14/22	Fri 1/14/22	8
Write Product Backlog	5 days	Mon 1/17/22	Fri 1/21/22	9
Development	**30 days**	**Mon 1/24/22**	**Fri 3/4/22**	**6**
Sprint 1	10 days	Mon 1/24/22	Fri 2/4/22	
Sprint 2	10 days	Mon 2/7/22	Fri 2/18/22	12
Sprint 3	10 days	Mon 2/21/22	Fri 3/4/22	13
Complete Development	0 days	Fri 3/4/22	Fri 3/4/22	14
User Acceptance Testing	**5 days**	**Mon 3/7/22**	**Fri 3/11/22**	**11**
End to End Testing	5 days	Mon 3/7/22	Fri 3/11/22	
Project Release	0 days	Fri 3/11/22	Fri 3/11/22	16

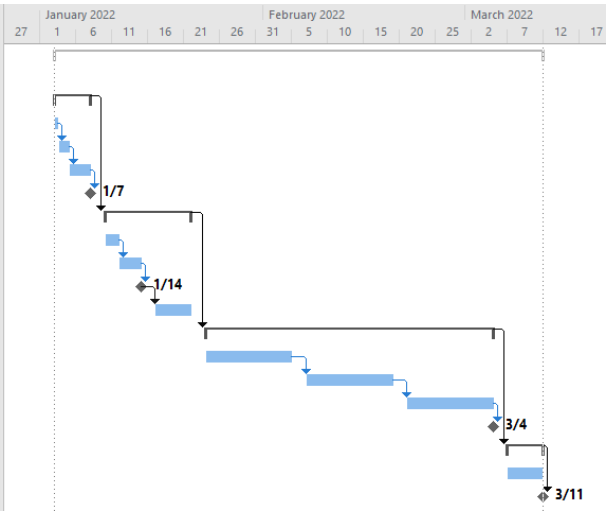


Figure 1 This Gantt chart highlights all of the primary phases and tasks of the project. The project will begin on 1/3/2022 with requirements gathering and will end on 3/11/22 with the deployment of the project.

Project Implementation

As previously mentioned in the project proposal, implementation will take several weeks. It will closely follow the Scrum framework, where development will be broken into two-week iterative cycles called sprints. During this time, the developers will work on smaller tasks toward the entire scope of the project rather than working on the bigger picture. This framework will allow the developers to work toward individual goals and keep the project on track. Due to the nature of the project, the implementation can be broken down into three separate sections comprising of developing the frontend interface, the data model, and the backend that integrates the system.

Frontend Implementation

The first of the three sections that should be implemented is the frontend comprising of all of the parts of the system that the user can see and interact with. Given that the database

integration and the backend are dependent on the frontend in terms of testing, the frontend will need to be completed first. To implement the frontend, the frontend developer working on the project will write the code and implement the site as a single page application (SPA) as it will be easier to maintain moving forward. The primary focus is getting the visual elements in place and creating a responsive design. The developer will not necessarily be coding anything for the backend or database, but should be developing the interface in preparation for those components.

Data Model Implementation

Depending on the specific pages being completed, the data model can be implemented concurrently with the frontend pages. The first step would be for the database administrator (DBA) to work with the developers to architect the data model and create an entity-relationship diagram. The diagram will show all of the database tables, their attributes, and how the tables are connected and share data through foreign key relationships. It is assumed this project will be using a relational database as all of WebstaurantStore's existing projects follow this (W. Edwards, personal communication, January 3, 2022).

After the data model is finalized, the DBA will be able to start creating the actual database, schemas, and tables so the team can use them. The DBA will need to work on setting up permissions, users, and objects so the system can remain secure. They will then need to set up several stored procedures. Similar to a method in an application, stored procedures will be used to retrieve, edit, insert, and delete information in the database either in one table or across multiple.

Backend Implementation

In general, the implementation of the backend will need to be completed after the frontend and database are completed. Depending on what pages and tables are complete, the

developers working on the backend may be able to work in tandem. For example, if the main project pages are completely coded and the appropriate tables and stored procedures are in place, the code for the backend can be started before the other pages and tables are complete. During the implementation of the backend, the primary goal will be to wire up the frontend interface to the database and other pages. The developers will finish up all of the final connections and integrations to other existing systems so that the entire tool is functional and can store data and give it back to the user.

Project Testing

Testing for this project will be completed in conjunction with development. It will be intentionally strenuous to ensure the integrity of the code and its overall validity, as this project could be used to make critical business decisions from the senior management level. At minimum, every single ticket completed by the team will be tested in full at least four times. Once a work item is completed, the developer who coded it will thoroughly test the functionality to ensure that it satisfies the acceptance criteria in the ticket. Once validated, they will send the item and their code into an internal peer review process where at least one other developer and potentially a senior developer will review the code and test it. If a ticket passes peer review, it will be assigned to a quality assurance engineer who will thoroughly test every detail of the ticket, ensuring the acceptance criteria has been met. They will also perform regression testing on it against everything already completed in the system. Lastly, the product owner will complete the final review and validate the ticket, confirming that the functionality is as originally anticipated.

At any point in the review and testing process, any assigned team members testing the ticket and code can reject the ticket and fail it if they find an issue. In the event of a failure, the

developer will make any necessary changes to the code, and then the entire review and testing process will start over again. This cycle is designed to be rigorous. Since the team is working on smaller tickets of the larger project instead of entire pages as a whole, they are more likely to spot bugs in the code early on and prevent more significant issues down the road when they are closer to the deadline. In addition, once the entire system is coded by the development team, a group of stakeholders that are representative of all of the end-users will be given access to the development application so they can test the entire site at one time as opposed to in pieces. These stakeholders will be looking at the bigger picture of how they would use the site from day-to-day. As long as all of the requirements were well-documented upfront, there should not be any significant changes from the stakeholders.

Report Summary

Given the deadline, resources, and cost needed to implement this project in its entirety, it will not be fully completed in the provided timeframe in actuality. Assumptions, however, can be made on the progress and implementation of the project. Additionally, significant progress has been made to develop a complete data model and the entire database, stored procedures, and several renderings of what the final user interface will look like once it is coded. All of these have been provided in the attached appendices.

When working with the product owner and stakeholders, it became apparent that one of the primary concerns was having a robust user interface for the website that would allow the users to get the information that they quickly need. The system does not necessarily need to be simple; it can be complex in the information and level of detail; it just needs to be easy to understand. Several pages underwent multiple design iterations based on example screenshots of existing systems until a final rendering was achieved. The designs were given to the product

owner and several stakeholders, and all were approved without any changes needing to be made. The preliminary designs will be adequate to provide a starting point for the frontend developers to build the entire frontend. A rendering of the project's dashboard homepage is shown in Appendix A: System Dashboard, and a view of a project's detail page is shown in Appendix B: Project Details Page. A view of the project list is shown in Appendix C: Project List, and the status update screens are shown in Appendix D: Weekly Status Update.

The entire data model was also created during the design process for this project. A brand-new database will be created to house all of the tables needed to support the system and four separate schemas. Different schemas will allow for easier maintenance of the system and help create a logical separation of the data for the developers. The first and largest schema is the 'Projects' schema, containing all project-specific information such as the list of projects, links, and integrations. There will also be a 'Members' schema that stores all of the user-specific information for the system, a 'Companies' schema for storage of department, division, and company information, and a 'StatusUpdates' schema for all of the business analysts' weekly status updates. An entity-relationship diagram showing the entire database structure is shown in Appendix E: Entity Relationship Diagram. For clarity, each of the schemas is color-coded in a unique color.

Along with the data model, the entire database has also been created using SQL Server Management Studio based on the entity-relationship diagram. The database is comprised of twenty-two separate tables made from a combination of regular data tables, linking tables, and lookup tables. The tables have been connected using foreign key relationships and have been appropriately seeded with example data that can be used for the development team for testing. Further, more than fifteen stored procedures have been created to allow users to retrieve, edit,

and insert data into the database. These procedures will be called directly from the website to update the database based on the action from the user. The stored procedures range from simple in nature to complex based on the joins and subqueries. The stored procedures that have been created are shown in Appendix F: Database Stored Procedures. While being implemented, the project will also integrate into WebstaurantStore's existing systems, such as its human resources platform to get user data, Microsoft Teams to send messages, and several other internal applications.

Recommendations for Future Enhancements

For the purposes of this report, it is assumed the project management system has been fully developed and implemented at WebstaurantStore and was delivered on time to the stakeholders. Based on user interaction from the stakeholders, the new website has met all of the initial requirements gathered in the beginning. Given this, along with post-release user feedback, the project is successful, however, no implementation is truly perfect in the sense that it will never be without roadblocks along the way or updates for the future. Like most projects, this application has a few additional changes and enhancements that should be made to enhance the system's usefulness to users such as adding automation and the ability to road map future projects.

Successes, Challenges, and Lessons Learned

The overall implementation of the system is inherently a significant success for the company mainly because they do not currently have a project management system, but also because of the way it was designed. In terms of design, the physical user interfaces have proven to be user-friendly, given that they were all modeled off of existing applications at the company. The design is familiar to the user, so it is intuitive and empowering. Users can expect the

application to work in a specific way and get the results they are looking for when they navigate the website. Along with this, the data model has been designed not to recreate a completely standalone application but to complement all of the existing systems that the application will be integrating with. The focus for the entire system was to bring all of the data surrounding projects together to enhance the current workflow instead of being intrusive to users.

The biggest challenge that the project was faced with was the sheer scope of the project and its requirements. As previously stated, implementing this project is not generally seen at an enterprise such as WebstaurantStore; instead, they would likely have purchased a project management system from a third-party vendor. Given its other systems, it is a rather daunting task for the company to implement and maintain. Further, the project's timeline was challenging as most development projects of this size would typically last several months, not a few weeks.

The biggest lesson learned from the project is to focus on the minimum viable product, not the whole idea of the system with every feature possible. Given the narrow timeline for the project, the only way to see it through successfully is to determine the absolute necessities of the system from the stakeholders' points of view. These were determined by asking the stakeholders and users what a dealbreaker would be to stop them from using the system compared to what could be implemented shortly after in additional phases. Based on that research, the must-haves were simplified to just the project dashboard, project detail pages, and status update pages. Without these, the project would not have been successful. However, this can be a difficult task as some users might think that something would be a must-have when it is not.

Modification Strategies and Actionable Steps

Three overarching enhancements have been identified from gathering requirements at the start of the project, the development of the project, and user interaction after the release. These

enhancements will make a considerable difference to the system's overall function but were not requirements for the initial implementation. First, the primary request is to have a history log associated with each project. Currently, users can make changes to a project, such as changing the milestone dates or the status, but there is no way to view which user made the changes. This information is not currently tracked in the database, so it would not be backfilled once the user interface is implemented. The feature itself would be helpful to understand which users are editing projects and for record-keeping in the future when looking back at a project.

Secondly, users have requested a way to tie automated notifications to the system. The data model was created to integrate with Microsoft Teams and Mailroom, the company's email processor. Still, there is no way to configure automated notifications to be sent currently. Users would use these to be notified when a project is marked completed or is overdue, for example. Lastly, users need a way to achieve project road mapping. The road maps would be similar to a high-level overview timeline of a project or a specific team. For example, the Customer Solutions department would want to map out the next few quarters of projects to prepare for product owner resources on its side.

In terms of actionable items, it is up to the business analyst group to prioritize these modifications as they are the primary product owners for the project. They would likely focus on the history first, as every change being made until it gets implemented is not currently being tracked. Adding notifications to the system is a much easier task than building out road-mapping functionality, so it would likely be implemented second with the road mapping last. The primary business analyst will begin documenting the requirements, just like this project itself, to start implementing the changes. From there, the team will develop the new features in the same project format, just on a smaller scale.

Potential Issues

Due to the scope and use case for the system, there are a few potential issues that can be seen from the implementation. The biggest potential problem would be simply getting users to adopt the system and begin using it. Much like the current format of spreadsheets to maintain projects, the system is only as good and as accurate as the users that are maintaining it, namely, the business analyst team. Given that they will be required to maintain the data in this new format, the risk is lower but is not zero. Additionally, the entire system is reliant on users manually entering information. There is no automation and only some validation, so the chance for issues arising from data error is higher.

Another issue that the company may see is the lack of features associated with the system at this time. Third-party applications like Monday.com have perfected project management systems as it is their only function for conducting business. Companies like this have tools and features such as complex artificial intelligence, advanced metrics reporting, and intuitive dashboards that WebstaurantStore would not be able to develop and maintain within a timely manner (Monday.com, 2016). It would be unrealistic for the company to invest the time and resources into features like this when they likely would not be used to their fullest extent. The lack of functionality, however, could present itself as a more prominent issue down the road as the company expands its number of projects. Despite this, the project management system concept designed for the company has been widely accepted by the users and will be very useful for the users once it is fully implemented at WebstaurantStore.

References

Clark Associates. (n.d.). *About Us*. Www.clarkassociatesinc.biz; Clark Associates Inc. Retrieved January 9, 2022, from https://www.clarkassociatesinc.biz/about/

Edwards, W. (2022, January 3). *Development at WebstaurantStore* (T. Latshaw, Interviewer) [Personal communication].

Edwards, W. (2022, January 24). *Project Management Requirements* (T. Latshaw, Interviewer) [Personal communication].

Monday.com. (2016). *Monday*. Monday.com. https://monday.com/

Musser, N. S. (2022, January 13). *Project Management Brainstorming* (T. Latshaw, Interviewer) [Personal communication].

Sanchez, V. (2022, January 20). *Project Management System Needs* (T. Latshaw, Interviewer) [Personal communication].

Appendix A: System Dashboard

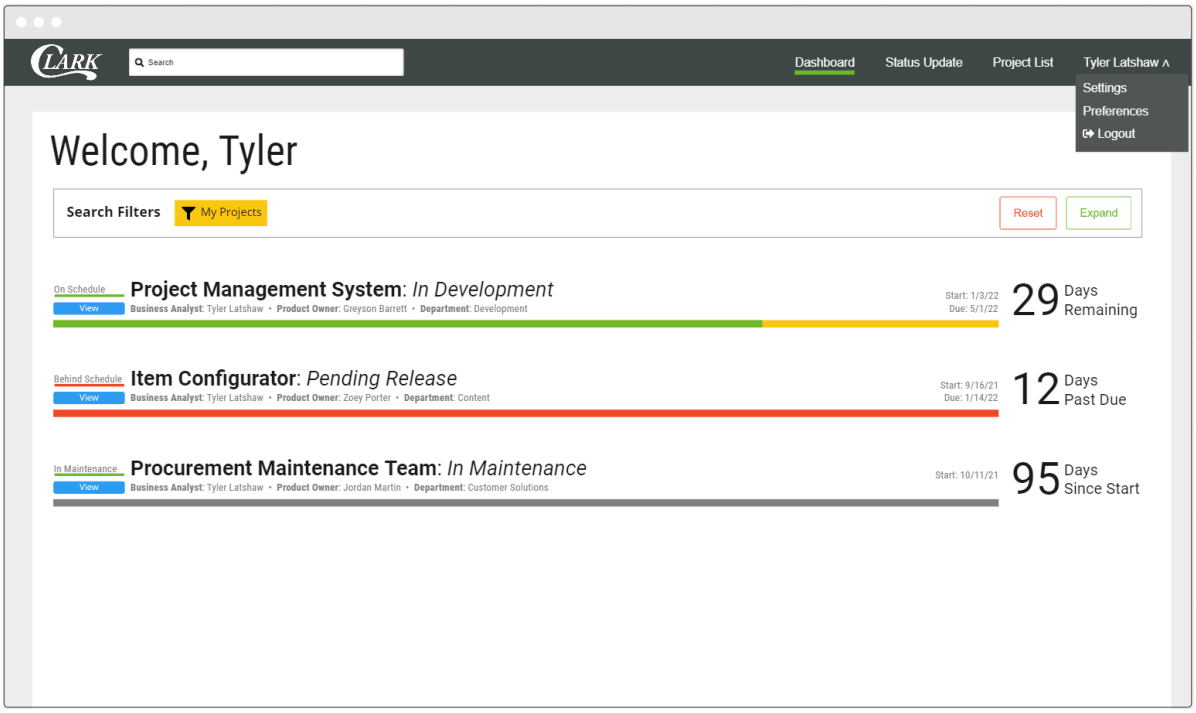


Figure 2 This prototype is a rendering of a user's homepage. Users will see all of their active projects by default, along with how long they have remaining. Users can access each of the project's detail pages from the view button.

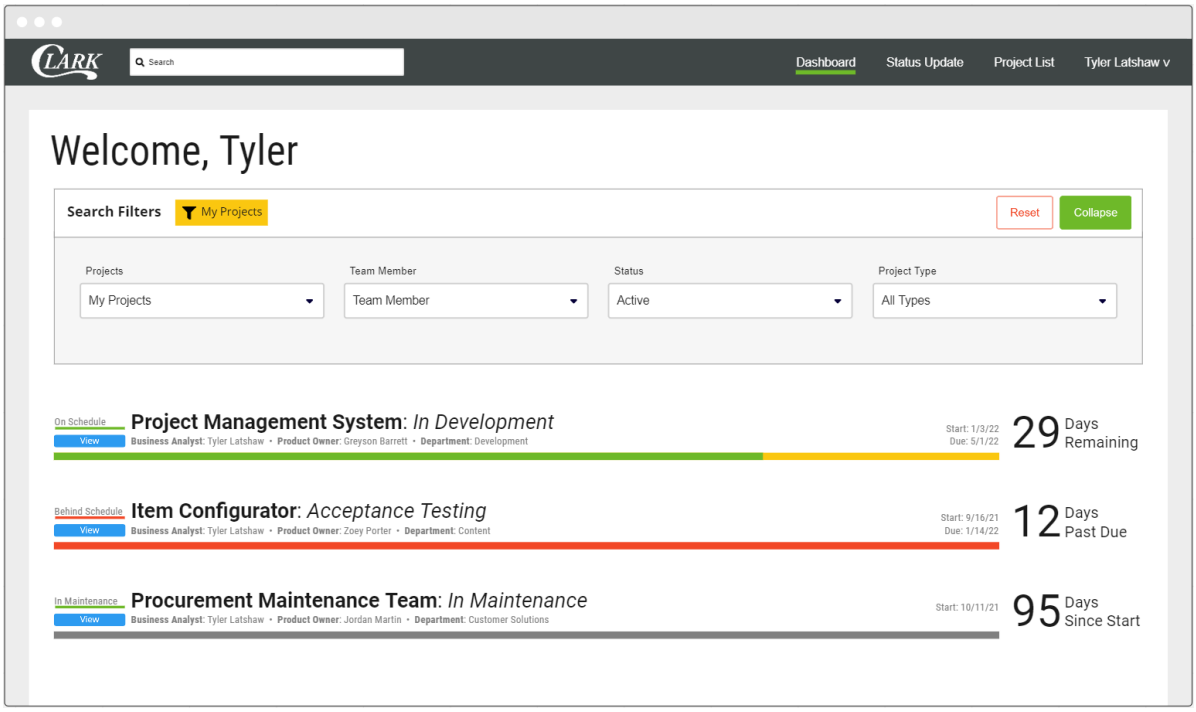


Figure 3 This rendering shows the user workflow from the homepage. Users can filter down the list of projects based on their projects, team members, project status, and type of project.

Appendix B: Project Details Page

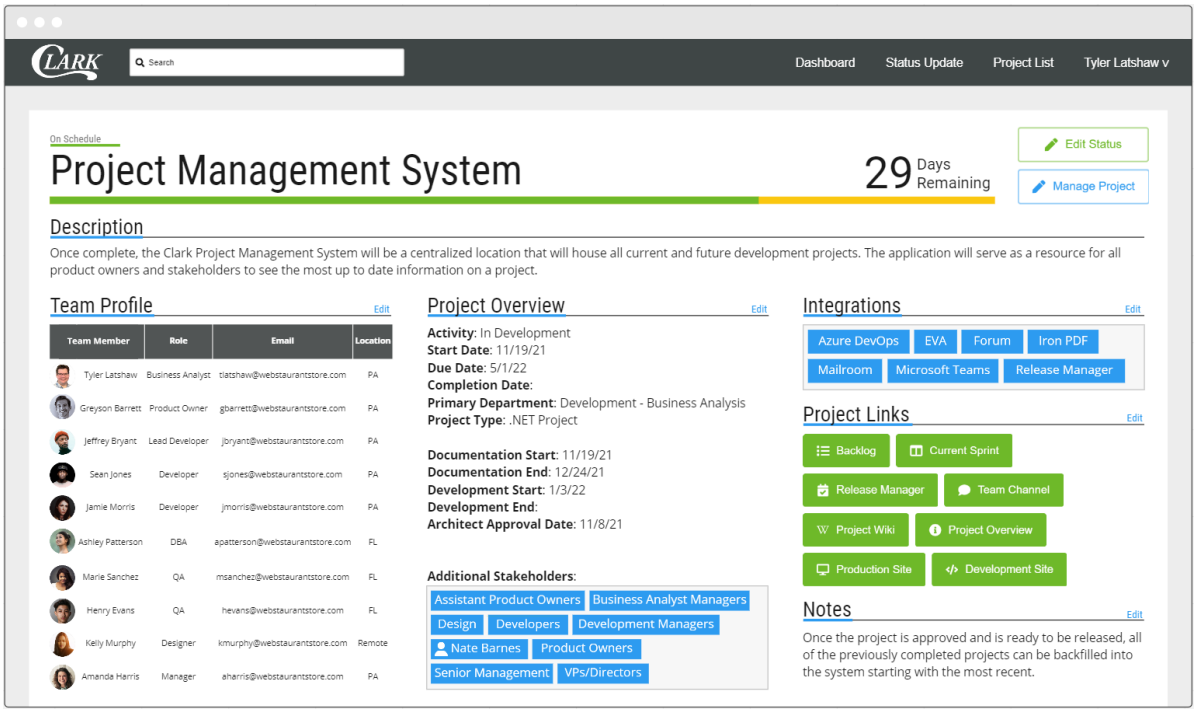


Figure 4 This rendering depicts the project detail page for a sample project. The page includes relevant project information, such as team lists, key milestones, and integrations.

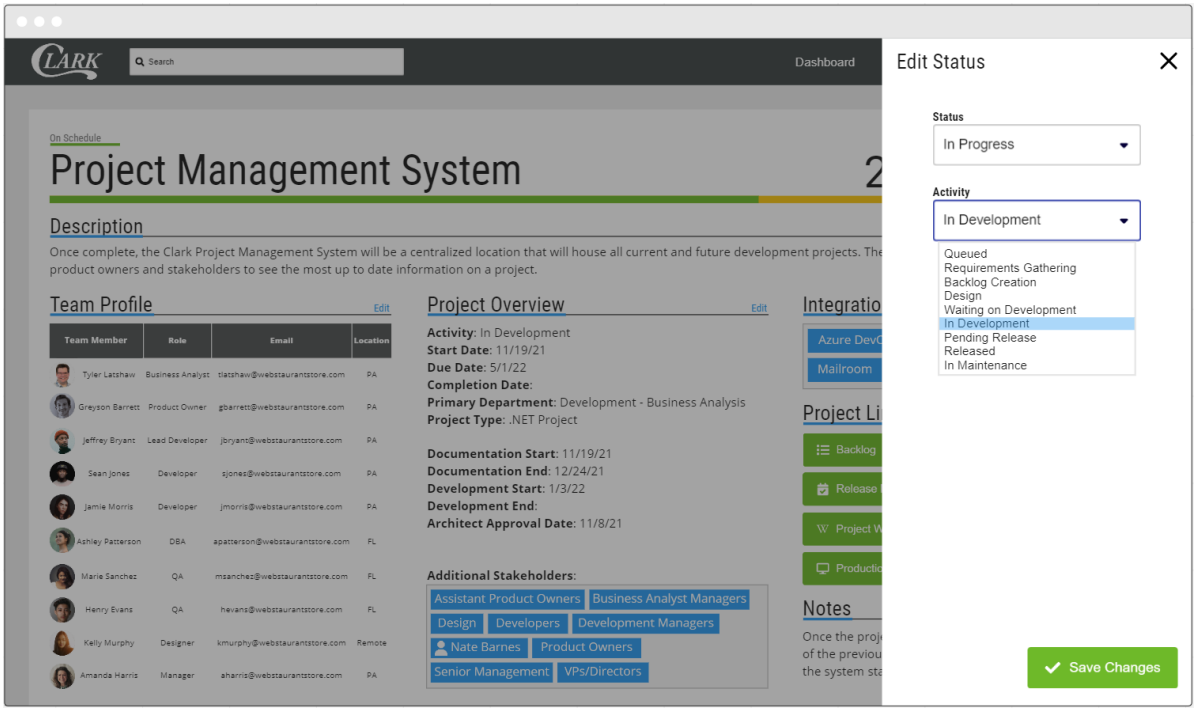
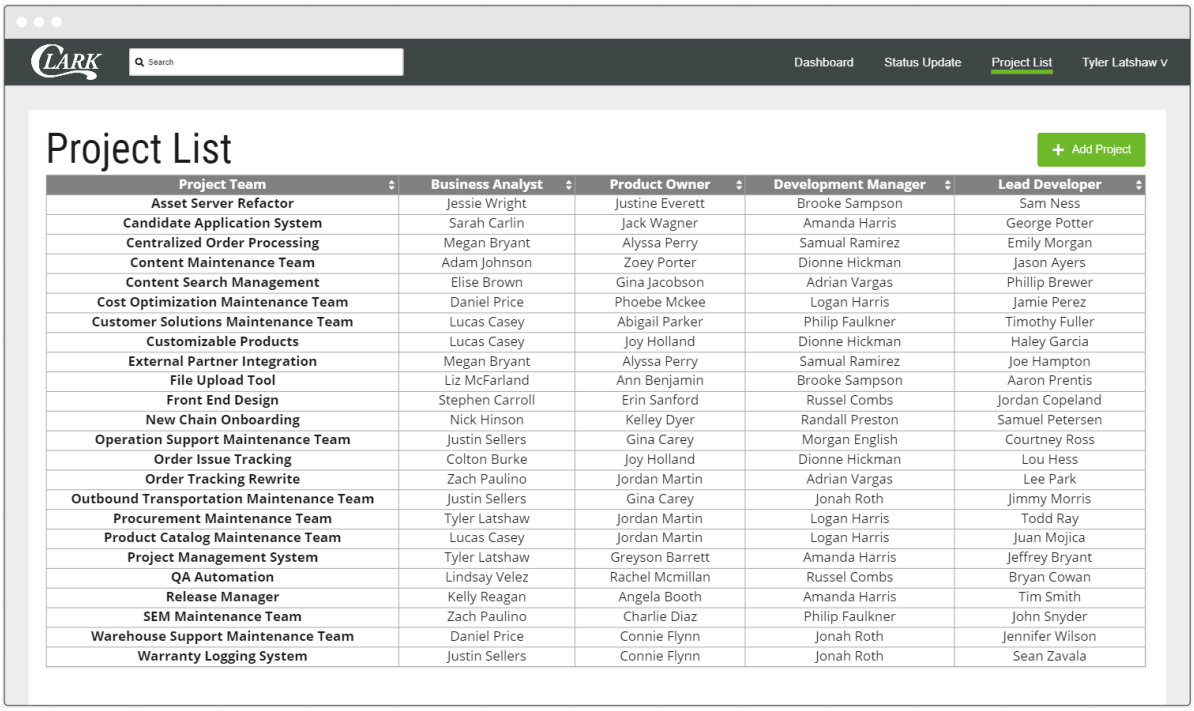


Figure 5 Building off the project details page, this rendering shows the fly-out drawer used to edit project information such as the status and current activity.

Appendix C: Project List



Project Team	Business Analyst	Product Owner	Development Manager	Lead Developer
Asset Server Refactor	Jessie Wright	Justine Everett	Brooke Sampson	Sam Ness
Candidate Application System	Sarah Carlin	Jack Wagner	Amanda Harris	George Potter
Centralized Order Processing	Megan Bryant	Alyssa Perry	Samual Ramirez	Emily Morgan
Content Maintenance Team	Adam Johnson	Zoey Porter	Dionne Hickman	Jason Ayers
Content Search Management	Elise Brown	Gina Jacobson	Adrian Vargas	Phillip Brewer
Cost Optimization Maintenance Team	Daniel Price	Phoebe Mckee	Logan Harris	Jamie Perez
Customer Solutions Maintenance Team	Lucas Casey	Abigail Parker	Philip Faulkner	Timothy Fuller
Customizable Products	Lucas Casey	Joy Holland	Dionne Hickman	Haley Garcia
External Partner Integration	Megan Bryant	Alyssa Perry	Samual Ramirez	Joe Hampton
File Upload Tool	Liz McFarland	Ann Benjamin	Brooke Sampson	Aaron Prentis
Front End Design	Stephen Carroll	Erin Sanford	Russel Combs	Jordan Copeland
New Chain Onboarding	Nick Hinson	Kelley Dyer	Randall Preston	Samuel Petersen
Operation Support Maintenance Team	Justin Sellers	Gina Carey	Morgan English	Courtney Ross
Order Issue Tracking	Colton Burke	Joy Holland	Dionne Hickman	Lou Hess
Order Tracking Rewrite	Zach Paulino	Jordan Martin	Adrian Vargas	Lee Park
Outbound Transportation Maintenance Team	Justin Sellers	Gina Carey	Jonah Roth	Jimmy Morris
Procurement Maintenance Team	Tyler Latshaw	Jordan Martin	Logan Harris	Todd Ray
Product Catalog Maintenance Team	Lucas Casey	Jordan Martin	Logan Harris	Juan Mojica
Project Management System	Tyler Latshaw	Greyson Barrett	Amanda Harris	Jeffrey Bryant
QA Automation	Lindsay Velez	Rachel Mcmillan	Russel Combs	Bryan Cowan
Release Manager	Kelly Reagan	Angela Booth	Amanda Harris	Tim Smith
SEM Maintenance Team	Zach Paulino	Charlie Diaz	Philip Faulkner	John Snyder
Warehouse Support Maintenance Team	Daniel Price	Connie Flynn	Jonah Roth	Jennifer Wilson
Warranty Logging System	Justin Sellers	Connie Flynn	Jonah Roth	Sean Zavala

Figure 6 This prototype shows the main project listing page. This page will be used by stakeholders to quickly know who the business analyst, product owner, development manager, and lead developer are for a project.

Appendix D: Weekly Status Update

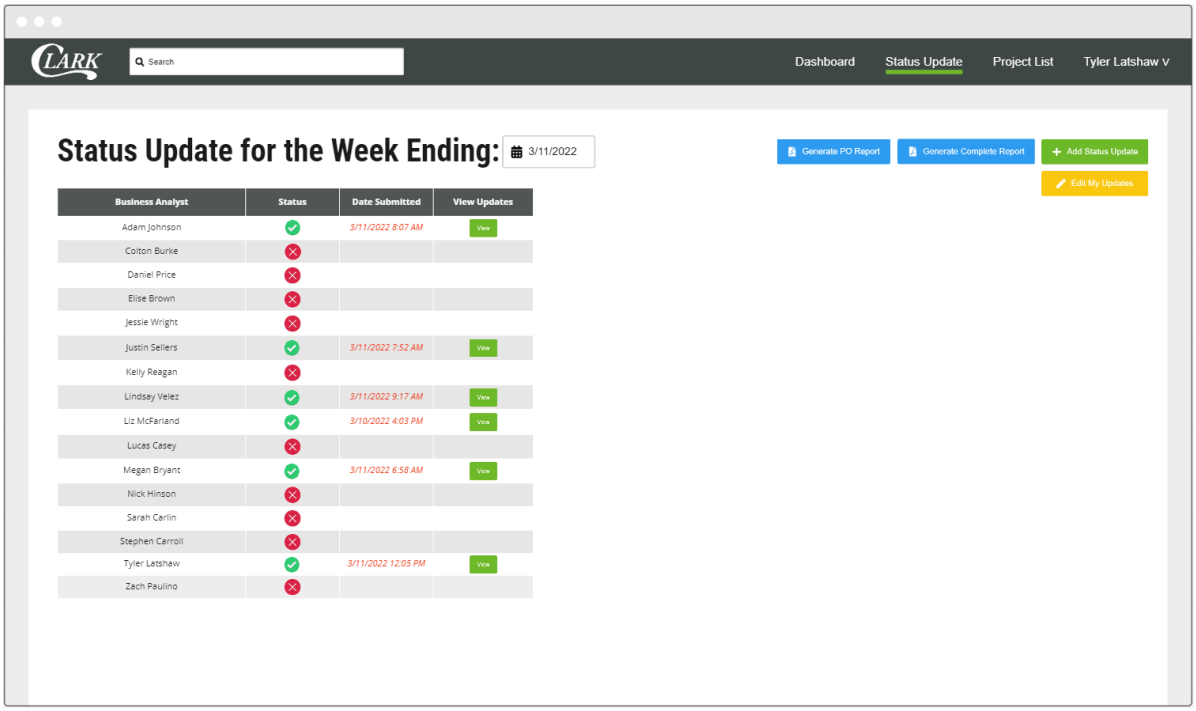


Figure 7 This prototype shows what the main status update page will look like. This page provides managers a quick reference of which business analysts have submitted their weekly status updates and which have not yet.

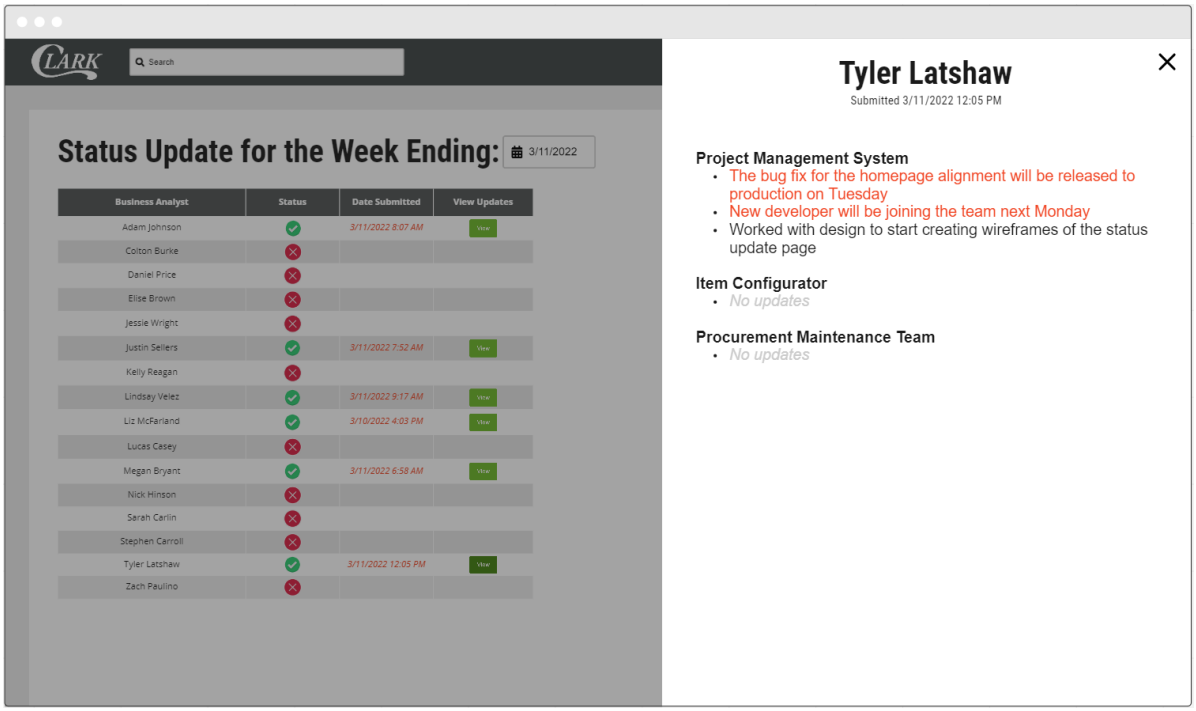


Figure 8 Building off of the main status update page, this rendering shows a slide-out drawer for the status updates entered for a specific business analyst. Updates in red text will be shared in the product owner version of the status updates.

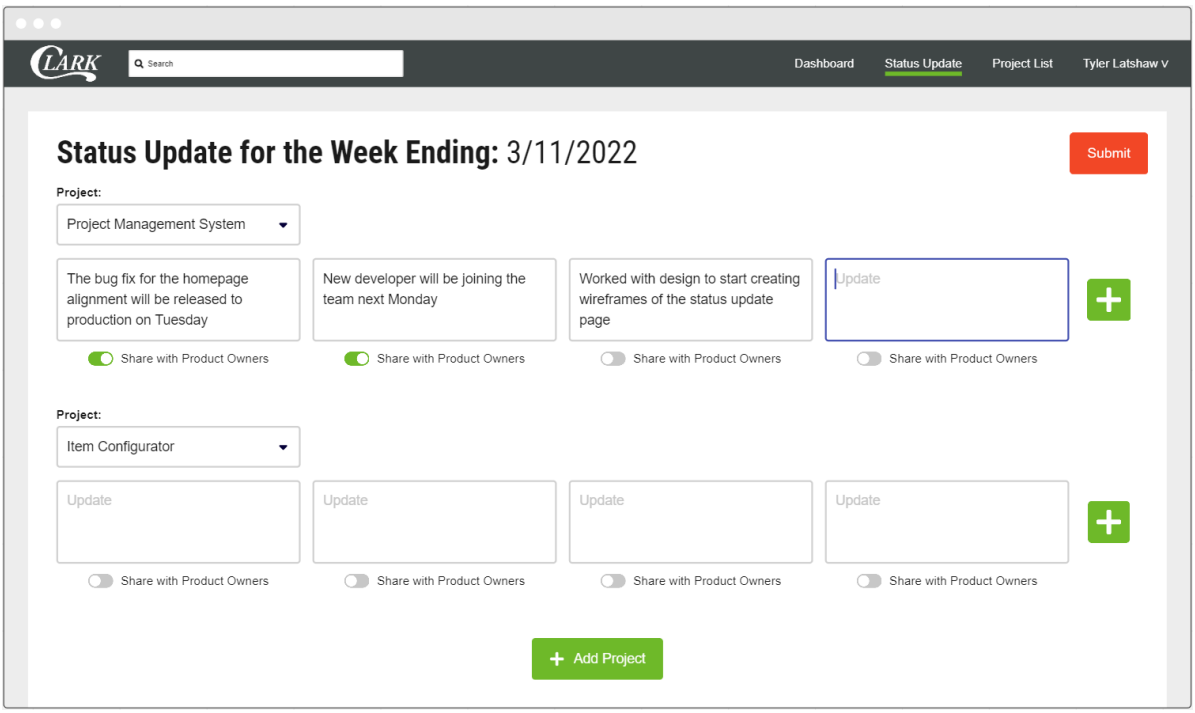


Figure 9 This prototype is a rendering of the status update edit screen. Business analysts will use this page to initially submit the updates and then edit existing updates if they need to.

Appendix E: Entity Relationship Diagram

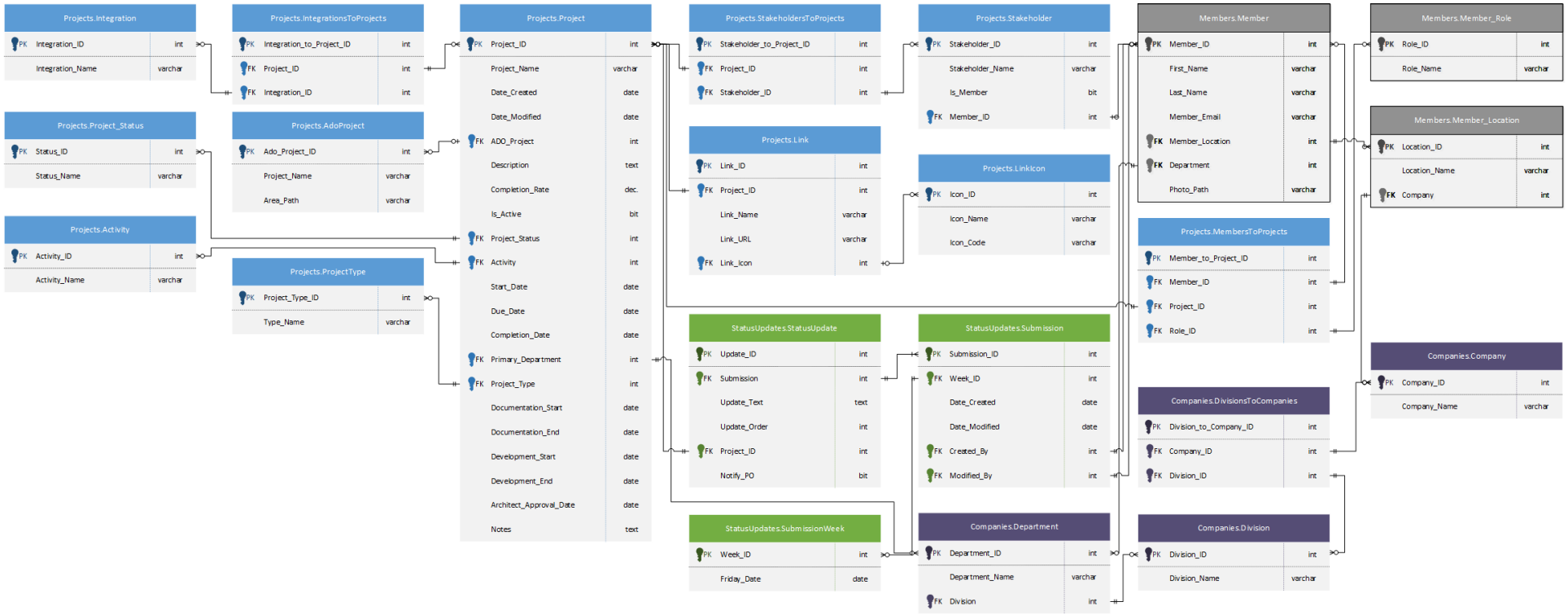


Figure 10 This entity-relationship diagram illustrates the data model that was designed for the project management system. The entire database is comprised of four schemas. The ‘Projects’ schema is shown in blue, the ‘StatusUpdates’ schema is shown in green, the ‘Members’ schema is shown in gray, and the ‘Companies’ schema is shown in purple.

Appendix F: Database Stored Procedures

In most relational database systems, database developers will create stored procedures to extract and manipulate data from the database tables. The stored procedures function similarly to a method in a computer program where they can be called, have information passed into it, and have data returned from it. The stored procedures will be called from the program written in the .NET Framework for this project. For this project, more than fifteen stored procedures were created to retrieve and update data. For reference, the query to create the procedure is included below, along with the execution scripts. The script to create the database is attached.

The following stored procedure will be executed from the homepage dashboard to retrieve all of the users that are a part of a project so that users can filter for a project by a user.

```
EXEC Members.sto_get_users

/*
CREATE PROCEDURE Members.sto_get_users
AS
	BEGIN
		SELECT M.Member_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Team Member'
			,COUNT(M.Member_ID) AS 'Project Count'
		FROM Members.Member M
		JOIN Projects.MembersToProjects MTP
		ON M.Member_ID = MTP.Member_ID
		JOIN Projects.Project P
		ON MTP.Project_ID = P.Project_ID
		AND P.Is_Active = '1'
		GROUP BY M.Member_ID, M.First_Name, M.Last_Name
		ORDER BY M.Member_ID
	END
*/
```

Used by the system homepage, the following code will create a process that will retrieve all of the primary project information based on the user, project status, or project type that is being filtered for. If a filter isn't used, a NULL value will be passed in, and all projects will be returned. Multiple execution statements are included for reference.

```
EXEC Projects.sto_get_abbreviated_project_info @Member_ID = NULL, @Project_Status = NULL, @Project_Type =
NULL
--EXEC Projects.sto_get_abbreviated_project_info @Member_ID = '75', @Project_Status = '1', @Project_Type =
NULL
--EXEC Projects.sto_get_abbreviated_project_info @Member_ID = NULL, @Project_Status = '1', @Project_Type =
'5'
```

```
/*
CREATE PROCEDURE Projects.sto_get_abbreviated_project_info (@Member_ID int, @Project_Status bit,
@Project_Type int)
AS
	BEGIN
		SELECT DISTINCT MTP.Project_ID
		INTO #Projects_Returned
		FROM Projects.MembersToProjects MTP
		WHERE ((@Member_ID IS NULL) OR (MTP.Member_ID = @Member_ID))

		SELECT P.Project_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member_Name'
			,MTP.Member_ID
		INTO #BA
		FROM #Projects_Returned P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Role_ID = '1'
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID

		SELECT P.Project_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member_Name'
			,MTP.Member_ID
		INTO #PO
		FROM #Projects_Returned P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Role_ID = '7'
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID

		SELECT PR.Project_ID
			,P.Project_Name
			,S.Status_Name
			,A.Activity_Name
			,#BA.Member_ID AS 'BA ID'
			,#BA.Member_Name AS 'BA Name'
			,#PO.Member_ID AS 'PO ID'
			,#PO.Member_Name AS 'PO Name'
			,D.Department_Name
			,P.Completion_Rate
			,P.[Start_Date]
			,P.Due_Date
		FROM #Projects_Returned PR
		JOIN Projects.Project P
		ON PR.Project_ID = P.Project_ID
		JOIN Projects.Project_Status S
		ON P.Project_Status = S.Status_ID
		JOIN Projects.Activity A
		ON P.Activity = A.Activity_ID
		JOIN Companies.Department D
		ON P.Primary_Department = D.Department_ID
		JOIN #BA
		ON P.Project_ID = #BA.Project_ID
		JOIN #PO
		ON P.Project_ID = #PO.Project_ID
		WHERE ((@Project_Status IS NULL) OR (P.Is_Active = @Project_Status))
		AND ((@Project_Type IS NULL) OR (P.Project_Type = @Project_Type))
		ORDER BY P.Project_Name

		DROP TABLE #BA
		DROP TABLE #PO
		DROP TABLE #Projects_Returned
	END
*/
```

The following stored procedure will retrieve a project's integrations. It will be called from the project's detail page.

```
EXEC Projects.sto_get_integrations_by_project_id @Project_ID = '20'

/*
CREATE PROCEDURE Projects.sto_get_integrations_by_project_id (@Project_ID int)
AS
	BEGIN
		SELECT I.Integration_ID
			,I.Integration_Name
		FROM Projects.IntegrationsToProjects ITP
		JOIN Projects.Integration I
		ON ITP.Integration_ID = I.Integration_ID
		WHERE ITP.Project_ID = @Project_ID
		ORDER BY I.Integration_Name
	END
*/
```

The following stored procedure will be called from the project detail page to retrieve all of the project's primary information, including milestones and notes.

```
EXEC Projects.sto_get_project_info_by_id @Project_ID = '20'

/*
CREATE PROCEDURE Projects.sto_get_project_info_by_id (@Project_ID int)
AS
	BEGIN
		SELECT P.Project_ID
			,P.Project_Name
			,P.Project_Description
			,S.Status_Name
			,A.Activity_Name
			,P.Start_Date
			,P.Due_Date
			,P.Completion_Date
			,P.Primary_Department
			,D.Department_Name
			,PT.Project_Type_Name
			,P.Documentation_Start
			,P.Documentation_End
			,P.Development_Start
			,P.Development_End
			,P.Arcitect_Approval_Date
			,P.Notes
		FROM Projects.Project P
		JOIN Projects.Project_Status S
		ON P.Project_Status = S.Status_ID
		JOIN Projects.Activity A
		ON P.Activity = A.Activity_ID
		JOIN Companies.Department D
		ON P.Primary_Department = D.Department_ID
		JOIN Projects.ProjectType PT
		ON P.Project_Type = PT.Project_Type_ID
		WHERE P.Project_ID = @Project_ID
		ORDER BY P.Project_Name
	END
*/
```

Called from the project detail page, the following will retrieve all of the project links, their name, and their icon for a specific project being passed into it.

```
EXEC Projects.sto_get_project_links_by_project_id @Project_ID = '20'

/*
CREATE PROCEDURE Projects.sto_get_project_links_by_project_id (@Project_ID int)
AS
	BEGIN
		SELECT L.Link_ID
			,L.Link_Name
			,L.Link_URL
			,LI.Icon_Code
		FROM Projects.Link L
		JOIN Projects.LinkIcon LI
		ON L.Link_Icon = LI.Icon_ID
		WHERE L.Project_ID = @Project_ID
		ORDER BY L.Link_Name
	END
*/
```

The project list page will list all active projects and their primary members, such as the business analyst, development manager, product owner, and lead developer. The following stored procedure will return this information when executed.

```
EXEC Projects.sto_get_project_list

/*
CREATE PROCEDURE Projects.sto_get_project_list
AS
	BEGIN
		SELECT P.Project_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member'
		INTO #BA
		FROM Projects.Project P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Role_ID = '1'
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID
		WHERE P.Is_Active = 1

		SELECT P.Project_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member'
		INTO #PO
		FROM Projects.Project P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Role_ID = '7'
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID
		WHERE P.Is_Active = 1

		SELECT P.Project_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member'
		INTO #DevManager
		FROM Projects.Project P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Role_ID = '5'
		JOIN Members.Member M
```

```
		ON MTP.Member_ID = M.Member_ID
		WHERE P.Is_Active = 1

		SELECT P.Project_ID
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member'
		INTO #LeadDev
		FROM Projects.Project P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Role_ID = '6'
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID
		WHERE P.Is_Active = 1

		SELECT P.Project_ID
			,P.Project_Name
			,#BA.Member AS 'Business Analyst'
			,#PO.Member AS 'Product Owner'
			,#DevManager.Member AS 'Development Manager'
			,#LeadDev.Member AS 'Lead Developer'
		FROM Projects.Project P
		JOIN #BA
		ON P.Project_ID = #BA.Project_ID
		JOIN #PO
		ON P.Project_ID = #PO.Project_ID
		JOIN #DevManager
		ON P.Project_ID = #DevManager.Project_ID
		JOIN #LeadDev
		ON P.Project_ID = #LeadDev.Project_ID
		ORDER BY P.Project_Name

		DROP TABLE #BA
		DROP TABLE #PO
		DROP TABLE #DevManager
		DROP TABLE #LeadDev
	END
*/
```

The homepage will use the following stored procedure to populate the list of project types in the filter.

```
EXEC Projects.sto_get_project_types

/*
CREATE PROCEDURE Projects.sto_get_project_types
AS
	BEGIN
		SELECT PT.Project_Type_ID
			,PT.Project_Type_Name
		FROM Projects.ProjectType PT
		JOIN Projects.Project P
		ON PT.Project_Type_ID = P.Project_Type
		AND P.Is_Active = '1'
		GROUP BY PT.Project_Type_ID, PT.Project_Type_Name
		ORDER BY PT.Project_Type_Name
	END
*/
```

Called from the project details page, the following stored procedure will return all of the stakeholders associated with a project.

```
EXEC Projects.sto_get_stakeholders_by_project_id @Project_ID = '20'
```

```
/*
CREATE PROCEDURE Projects.sto_get_stakeholders_by_project_id (@Project_ID int)
AS
	BEGIN
		SELECT S.Stakeholder_Name
			,S.[Is_Member]
			,CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member Name'
		FROM Projects.StakeholdersToProjects STP
		JOIN Projects.Stakeholder S
		ON STP.Stakeholder_ID = S.Stakeholder_ID
		LEFT JOIN Members.Member M
		ON S.Member_ID = M.Member_ID
		WHERE STP.Project_ID = @Project_ID
		ORDER BY S.Stakeholder_Name
	END
*/
```

A table will be populated with the results of the following stored procedure on the project details page to show all of the users associated with the respective project.

```
EXEC Projects.sto_get_team_members_by_project_id @Project_ID = '20'

/*
CREATE PROCEDURE Projects.sto_get_team_members_by_project_id (@Project_ID int)
AS
	BEGIN
		SELECT CONCAT(M.First_Name, ' ', M.Last_Name) AS 'Member Name'
			,R.Role_Name
			,M.Member_Email
			,L.Location_Name
			,M.Photo_Path
		FROM Projects.MembersToProjects MTP
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID
		JOIN Members.Member_Role R
		ON MTP.Role_ID = R.Role_ID
		JOIN Members.Member_Location L
		ON M.Member_Location = L.Location_ID
		WHERE MTP.Project_ID = @Project_ID
		ORDER BY R.Role_Name, 'Member Name'
	END
*/
```

The project details page will call the following stored procedure to update the status and the activity for a project.

```
EXEC Projects.sto_update_activity_status_by_id @Project_ID = '20', @New_Activity = '6', @New_Status = '3'

/*
CREATE PROCEDURE Projects.sto_update_activity_status_by_id (@Project_ID int, @New_Activity int,
@New_Status int)
AS
	BEGIN
		UPDATE Projects.Project
		SET Activity = @New_Activity
			,Project_Status = @New_Status
		WHERE Project_ID = @Project_ID
	END
*/
```

Called from the main status update page, the following stored procedure will return all of the business analysts and whether or not they have submitted their status updates yet for the given week.

```
EXEC StatusUpdates.sto_get_status_updates_by_week @Week_Ending = '2022-03-11'

/*
CREATE PROCEDURE StatusUpdates.sto_get_status_updates_by_week (@week_ending DATE)
AS
	BEGIN
		SELECT M.First_Name
			,M.Last_Name
			,Sub.Date_Created
			,Sub.Submission_ID
		FROM Members.Member M
		LEFT JOIN
			(SELECT S.Submission_ID
				,S.Week_ID
				,S.Date_Created
				,S.Created_By
			FROM StatusUpdates.Submission S
			JOIN StatusUpdates.SubmissionWeek W
			ON S.Week_ID = W.Week_ID
			WHERE W.Friday_Date = @week_ending) Sub
		ON M.Member_ID = Sub.Created_By
		WHERE M.Department = '6'
	END
*/
```

The following stored procedure will be used when a user wants to view the status updates for a specific user and generate the PDF report for a given week.

```
EXEC StatusUpdates.sto_get_status_updates_by_id @Submission_ID = '6'

/*
CREATE PROCEDURE StatusUpdates.sto_get_status_updates_by_id (@Submission_ID int)
AS
	BEGIN
		DECLARE @User_ID int =
			(SELECT TOP (1) S.Created_By
			FROM StatusUpdates.Submission S
			JOIN StatusUpdates.StatusUpdate U
			ON S.Submission_ID = U.Submission
			WHERE S.Submission_ID = @Submission_ID)

		SELECT P.Project_ID
			,P.Project_Name
			,M.First_Name
			,M.Last_Name
		INTO #UserProjects
		FROM Projects.Project P
		JOIN Projects.MembersToProjects MTP
		ON P.Project_ID = MTP.Project_ID
		AND MTP.Member_ID = @User_ID
		AND MTP.Role_ID = '1'
		JOIN Members.Member M
		ON MTP.Member_ID = M.Member_ID

		SELECT UP.Project_ID
			,UP.Project_Name
			,UP.First_Name
```

```
                        ,UP.Last_Name
                        ,Sub.Date_Created
                        ,Sub.Update_ID
                        ,Sub.Update_Text
                        ,Sub.Update_Order
                        ,Sub.Notify_PO
                FROM #UserProjects UP
                LEFT JOIN
                        (SELECT U.Project_ID
                                ,S.Date_Created
                                ,U.Update_ID
                                ,U.Update_Text
                                ,U.Update_Order
                                ,U.Notify_PO
                        FROM StatusUpdates.StatusUpdate U
                        JOIN StatusUpdates.Submission S
                        ON U.Submission = S.Submission_ID
                        WHERE S.Submission_ID = @Submission_ID) Sub
                ON UP.Project_ID = Sub.Project_ID
                ORDER BY UP.Project_ID, Sub.Update_Order

                DROP TABLE #UserProjects
        END
*/
```

Called from the main weekly status update page, the following stored procedure will return all of the weeks that can be used for a status update, based on the Friday date.

```
EXEC StatusUpdates.sto_get_weeks

/*
CREATE PROCEDURE StatusUpdates.sto_get_weeks
AS
        BEGIN
                SELECT Week_ID
                        ,Friday_Date
                FROM StatusUpdates.SubmissionWeek
                ORDER BY Friday_Date
        END
*/
```

The following two stored procedures will be executed in tandem to insert a new submission and the respective status updates. The foreign key restraints will need to be executed as a transaction, so the entire process is completed instead of just part of it if something were to fail.

```
BEGIN
        DECLARE @Return_Submission_ID int
        EXEC @Return_Submission_ID = StatusUpdates.sto_insert_new_submission @Week_ID = '63', @User_ID =
'75'

        EXEC StatusUpdates.sto_insert_new_status_update @Submission_ID = @Return_Submission_ID,
@Update_Text = 'First status update', @Update_Order = '1', @Project_ID = '20', @Notify_PO = '1'
        EXEC StatusUpdates.sto_insert_new_status_update @Submission_ID = @Return_Submission_ID,
@Update_Text = 'Second status update', @Update_Order = '2', @Project_ID = '20', @Notify_PO = '0'
        EXEC StatusUpdates.sto_insert_new_status_update @Submission_ID = @Return_Submission_ID,
@Update_Text = 'Third status update', @Update_Order = '3', @Project_ID = '20', @Notify_PO = '0'
END
```

```
/*
CREATE PROCEDURE StatusUpdates.sto_insert_new_submission (@Week_ID int, @User_ID int)
AS
	BEGIN
		INSERT INTO StatusUpdates.Submission (Week_ID, Date_Created, Date_Modified, Created_By,
Modified_By)
		VALUES (@Week_ID, CURRENT_TIMESTAMP, CURRENT_TIMESTAMP, @User_ID, @User_ID)

		RETURN SCOPE_IDENTITY()
	END
*/

/*
CREATE PROCEDURE StatusUpdates.sto_insert_new_status_update (@Submission_ID int, @Update_Text
varchar(255), @Update_Order int, @Project_ID int, @Notify_PO int)
AS
	BEGIN
		INSERT INTO StatusUpdates.StatusUpdate (Submission, Update_Text, Update_Order, Project_ID,
Notify_PO)
		VALUES (@Submission_ID, @Update_Text, @Update_Order, @Project_ID, @Notify_PO)
	END
*/
```

Called from the status update edit page, the following will be called to update each status update for a given project.

```
EXEC StatusUpdates.sto_update_submission_by_id @Update_ID = '6', @Project_ID = '8', @Update_Text = 'New
status update goes here', @Notify_PO = '1'

/*
CREATE PROCEDURE StatusUpdates.sto_update_submission_by_id (@Update_ID int, @Project_ID int, @Update_Text
varchar(255), @Notify_PO int)
AS
	BEGIN
		UPDATE StatusUpdates.StatusUpdate
		SET Project_ID = @Project_ID
			,Update_Text = @Update_Text
			,Notify_PO = @Notify_PO
		WHERE Update_ID = @Update_ID
	END
*/
```