

SE361 Final Report
Spring 2024

Table of Contents

Project Description	3
Sprint Overviews	3-4
Individual Reflections	4-6
Conclusion	6-7

Team Members and Roles

- Rayhan Kapadia (Full Stack Developer)
- Catherine Spratt (Full Stack Developer)
- Lucas Sturm (Front End Developer)
- Jacob Shank (Back End Developer)
- Jonathan Wang (Scrum Master)

Project Description

The project that was developed during the course of the spring semester was the IT Service Help Desk application. This project simulates a real information technology management system in which members of an organization working on a project raise tickets which voice their concerns. There are four major roles: Project Members, IT Help Support Team Members, Report Managers, and Administrators. In this application, Project Members would raise a ticket, have the Support Team solve the ticket, and eventually close the ticket. Report Managers would keep track of all the tickets and ensure they are being allocated properly and the right team members work on the right problems. There is also the ability for both Report Managers and Support Team Members to add people to help them on a ticket or to change the priority of the ticket set by the Project Member. The Administrator would have access to the overall application and would be able to see any side of the user interface.

Sprint Overviews

Sprint Zero was the beginning sprint of the entire project. This sprint involved mostly getting the team together and making project management decisions. The team also decided on The team chose to use Trello as the project management tool as it was incredibly simple and easy to use. The team also chose the IT Service Help Desk application as it seemed like the most exciting project of them all. A team name was settled on as Gugul Development. Group members also introduced themselves and became familiar with their teammates. In terms of development for Sprint Zero, there were many challenges. Visual Studio was a challenge for everyone along with maintaining a proper GitHub. Eventually, a simple login was created. The login GUI was functioning and connected to a database. This proved to be an issue as the database was locally and not through Microsoft SQL Server. Overall, the sprint was a good starting point and got things off the ground.

Sprint One was the most difficult sprint by far. The challenges were not related to programming, rather they were related to the software being used. Visual Studio did not work after installing it on a Windows machine for one member of the team. Others started to use the virtual machine and there were also issues with working between virtual machines and local machines. The biggest challenge was the GitHub repository. Sprint Zero artifacts were not located in the correct repository, so the team worked quickly to get a functioning repository up and running. However, there were issues with taking code from Sprint Zero into the new repository resulting in the creation of many different repositories. Eventually, this issue was resolved, but by this point, very little progress was made. The biggest progress made was the design of the application which was decided on along with a color scheme.

Sprint Two saw much more progress. After the disaster of Sprint One, the team decided to meet Wednesdays and Fridays outside of class every week. This allowed much more progress to be made. During this sprint, a working database was created in Microsoft Azure along with a registration form connecting to it. Users could connect to the database, create an account, and sign in. A ticket creation form was also made, however it did not have a database connection.

The success of this sprint made the team a lot more confident about this project and more goals were made for the next sprint.

Sprint Three saw much progress. The biggest addition was the transformation of the database. The database now included classes for users and tickets. The ticket class allowed for the ability to raise tickets and add them to the database and that functionality was finally completed. Registration was also completed as well. The GUI also saw some design changes with the team opting to display the program in full screen. The ticket creator forms also saw updates to the design and their appearances. A session class was created and passed through all of the forms to effectively access and manage the information in the databases. The team also started to develop the specific roles outlined in the project deception. Project Member functionality was completed. IT Help Desk Team Members were now able to see the tickets assigned to them. Report Managers could also now see all the tickets. This sprint saw great progress and set the tone for future sprints. There were also issues with the design with a lack of consistency and clutter. The team chose to address this issue in the next sprint.

Sprint Four started with major changes. The team went back to the drawing board and decided on a redesign of the GUI to allow for better readability and consistency. The form went from blue and purple to red, white, and blue. The full-screen mode was scrapped. The biggest challenge was getting the SMTP client to work. This would allow Report Managers and IT Help Desk Team Members to send emails to the Project member when a ticket has been solved. The SMTP client would also be necessary for the registration to have forgotten username and password functionality. This would become something that would have to be solved in the fifth and final sprint.

Sprint Five saw the completion of the application. All issues regarding the SMTP were resolved and the flow of the application was also altered to make it much smoother and easier to navigate. Most changes that were made were largely visual to make the application easier to read. By the end of Sprint Five, the application was completed and to the satisfaction of Gugul Development.

Individual Reflection

Rayhan Kapadia: As a full-stack developer, I worked on both sides of the application. A lot of my time was spent writing functionality into the project. I wrote many functions for the database class to allow for functionality. This allowed us to be able to raise tickets, respond to tickets, close tickets, and show tickets for IT Help Desk Members, Report Managers, and Project Members. I also wrote the session class and decided to not use attributes as C# offers because I prefer traditional object-oriented programming. I also allowed for the ability to click on tickets to open up the forms related to them and view their information by clicking on them. I also spent my time redesigning the forms to match the correct appearance Jacob decided on. Overall, I enjoyed this project because it connected much of what I learned in previous classes. The session class was a big “ah-ha” moment for me to understand why OOP is so fundamental to programming. The one thing I wish I could have done was try to make it scale as much as

possible. Applications will always break, but I feel that our application cannot scale to a real ticket system used in the workplace.

Catherine Spratt: My contributions to my software engineering team included developing the registration system, setting up the email, and working with Jacob to store registration in the database system accurately. I believe that I played an integral role in my team's progress because I worked on the frontend and backend to ensure functionality on the user's side and the admin's side of the application. Regarding the Sprints, I spent time working with Lucas in Sprints 0 and 1 to develop the GUI for the registration system to show all the information that allows the user to register. In Sprints 2 and 3, I worked with Jacob to save the information entered in the registration form to the Database class accurately. In addition, I went through every form in the application and ensured that there were proper naming conventions used to make the application consistent. Through the last two Sprints, I spent time working on the Forgot Password and the registration system forms to connect the email account to the database and ensure that if a user forgets his or her password, he or she can be contacted by email to receive the information from the database accurately. I also added functionality in the registration system to allow the IT administrator to register a user as one of three types of users with a special code. I greatly enjoyed working on this project with my teammates to create a functional product and the experience provided me with leadership skills that I can use in my internship this summer.

Lucas Sturm: My role on the team was a front end developer. During the semester, I worked on the GUI design of the software. I designed a logo for the team and found a color palette that I thought would look professional to use in the design. I created some basic mockup ideas for the software UI and got the team's approval of the designs. Afterwards, I was responsible for implementing the UI designs on the project. I had designed most form pages myself and got a little help here and there. In Sprint 4, I worked with Jacob on improving the UI design. We had received some feedback from our peers stating that the UI design seemed a little inconsistent in color format and text was hard to read. Each member of the team created a mockup of a redesign to the software and then we voted on whose design we liked the most. Jacob's design was the best and I used the design he provided to rehaul the UI to that style. I ended up getting help from others to make a more consistent design across all of the forms. I also worked on formatting the form windows. We originally had the forms full screen however, we decided to just keep them centered in the middle of the screen in window mode. I did some light programming here and there however most of the time I spent designing the GUI. I wish I had helped with programming more but I think what held me back was not being comfortable with the programming language and not knowing how to help. I really enjoyed taking the class and working with my team on the project. I believe to take the knowledge that I learned from Scrum and apply it to my day to day work in the future. I also hope to use my experience designing the software and use it in a possible internship over the summer.

Jacob Shank: I feel that I was imperative to the group at all stages of development. At the beginning of the development process, I was essentially a database specialist. This means that our Azure-hosted database was managed and set up by me. The initial Database class and its core

functions were implemented by me, and then much other functionality was added on by myself and others. Further, I was the one to design, update, and maintain the database throughout the process. At multiple points the database needed changing, and I did so quickly so that the rest of my group could function. Even through the last sprint, I was doing work in the back end and database, to ensure that tickets were being assigned correctly.

Outside of the database, however, I still feel I was imperative. As we continued to develop, I felt that I essentially became a full-stack developer, and I left my fingerprint on every file in the stack. I helped in the early development of the Session class and User class before passing it off to Rayhan. Even more to the front end, I developed what we now use as the final design of the application, and the assets to make it happen. In the final two sprints, I spent many hours ensuring that the application felt good and easy to use, as I implemented a large number of the quality of life features in the application, including the full application restructure in Sprint 4. I also feel that I played an important role in testing the application, finding bugs, and solving issues. I found many issues with the application, and I was part of the design conversation in fixing them.

In general, I feel that I was an important part of the design process, as I was willing to disagree with a teammate, yet I aimed to understand when I was wrong and adapt to my teammates' good ideas.

Jonathan Wang: As the Scrum master, I oversaw a lot of what my team was doing. I gave them directions on what we could do next. Also, I led the daily meetings and helped my team determine what was going well and what needed to be changed or added. Basically, I understood the directions for the project and helped guide my team in meeting the requirements. For the coding part of the application, I tested out a lot of the code to see if it was working or not. I spent some time on testing out tickets by checking to see if they will open up the forms related to them. I also spent time on implementing the buttons of the tickets in the database and session classes.

I felt that overseeing what my team was doing was not a hard task. I could easily keep track of what everyone was doing and supposed to do because I recorded all the tasks in Trello.

Conclusion

Overall, the course and the project were very engaging and informative. There was a lot to learn about working within a team to develop functioning software from start to finish. However, there were still a fair amount of challenges that made working on the project difficult.

In the end, the goals set out at the start of the project were met and the application was finished in its entirety. The design was something that proved challenging initially, but the final product was clean and easy to read. The biggest challenge, however, was Scrum. Scrum is very flawed in its design. Jeff Sutherland, the inventor of Scrum, makes many assumptions to back his claim that Scrum is the best method of project development and management. Arguably the biggest issue with Scrum was the bi-weekly presentation. Presentations often got in the way of the flow of projects. Everything would have to be dropped for a presentation to be made and put together, and creating a demo was challenging when code was still being written. Another issue

was team velocity. Jeff Sutherlands makes assumptions that velocity will remain constant and projects will be much more predictable because of this. This cannot be further from the truth. Different scenarios are going on in people's lives and it can seriously affect team velocity. In some sprints, team velocity is incredibly high while in other sprints very little is completed. There are positives to Scrum, the biggest being daily standups. Standups are a great way to get everyone engaged and working together. Overall, Scrum has good intentions and ideas, however, implementing its strategies exactly as described would only be most efficient in an ideal scenario. What worked best for us in the end was having in-person meetings, which made a difference as everyone understood their assignments and their goals. Working together was also an exciting experience as a whole.

For the most part, the project went fairly smoothly. The biggest challenge was working with the virtual machine and working with GitHub, but this problem could have been mitigated if all group members possessed a Windows machine along with a working version of Visual Studios. When all members of the team had time and a lighter workload from other classes, significant progress was made. In retrospect, a lot of changes should have been made. The biggest being having all group members use a Windows machine, but there were also other issues. Design choices should have been different. Quality should have been the priority in each sprint rather than quantity. Pushing more functionality and focusing on quantity can have worse effects on quality. This resulted in forms having strange appearances and inconsistencies when they should have been consistent the whole time. There should have been more standardization of design choices along with programming choices. Staying consistent with naming conventions should have also been prioritized to keep code readability high. Without these standards from the beginning, we spent considerable time back-tracking in our rush to not leave something "half done." Many of these changes were fixed at the end, but keeping quality high from the start would have been a better idea.