4-16-2024

# 5675 WiredCats Scouting Hub

Sebastian Smiley
*Western Michigan University*

# WiredCats Scouting Hub App

## Table of Contents


Table of Contents ..... 1
Abstract ..... 3
Problem ..... 3
- Need ..... 3
- Objective ..... 4
- Terms, Acronyms, Glossary ..... 4

Problem Analysis and Research ..... 5
- API Data ..... 5
- Internet Connectivity ..... 5
- Future Configuration ..... 6

Requirements ..... 6
- Laptop Functionality ..... 6

Internet Connectivity ..... 6
CSV Data ..... 6
API Data ..... 7
Match Reports ..... 7
Alliance Reports ..... 7
Standards and Constraints ..... 7
Applicable Standards ..... 7
Constraints ..... 8
Cost ..... 8
Scope ..... 8
Time ..... 8
Ethical Analysis ..... 8
Identify the Moral Issue ..... 8
Identify Additional Facts Helpful to Making the Decision ..... 9
Identify the Alternatives Available ..... 9
Identify the Personal Impacts to the Decision-Maker ..... 10
Ethics Assessment According to Contemporary Theories of Business Ethics ..... 10
Shareholder Theory ..... 10
Stakeholder Theory ..... 10
Utilitarianism ..... 10
Rights ..... 11
Justice ..... 12
Virtue Theory ..... 12
Ethical Conclusions ..... 13
System Design ..... 13
Testing ..... 14
Results ..... 15
Realization of Requirements ..... 15
Realization of Standards and Constraints ..... 15

Testing Results ..... 16
Future Work ..... 16
Modular Design ..... 16
Direct Data Collection ..... 17
Long-Term Data Storage ..... 17
Conclusion ..... 18
Appendices ..... 18
Project Management Plan ..... 18
Progress Reports and Minutes ..... 20
Development Costs ..... 64

# Abstract

The WiredCats Scouting Hub was created to provide FIRST team 5675 with an application that allows them to make more informed strategic decisions regarding their competitive play. The app synthesizes data from multiple sources, parsing multiple data formats into a single source of truth. It also presents data to users through graphs and charts and provides interactive tables. The application meets the requirements set forth by the leadership of team 5675, effectively providing the capabilities they seek.

# Problem

## Need

Mattawan WiredCats FIRST Robotics team members dedicate considerable effort to observing matches of other teams and researching their scores on various websites. This manual data collection and analysis, involving visualization through graphs and tables, is time-consuming. This extensive process diverts attention from their primary focus: enhancing their team's performance and optimizing their robot's capabilities.

Additionally, leveraging this data to strategize on alliance formation is crucial when they rank among the top teams. The current manual method is cumbersome and prone to errors, potentially leading to suboptimal alliance selections and missed opportunities.

## Objective

The WiredCats Scouting Hub facilitates the team's data selection and interpretation steps. By integrating scheduling data from the FIRST API with scoring data from AppSheet into a single user-friendly platform, the application simplifies the visualization of vital statistics. It also automates the generation of match reports, presenting essential data through intuitive graphs and tables for analysis.

Commented [DW1]: data selection and interpretation steps.

The app also includes a dedicated alliance selection feature. It allows the team to evaluate and rank potential allies based on various performance metrics, ensuring informed decision-making. This feature also helps keep track of chosen alliances, ensuring a strategic approach to team selection.

By streamlining these processes, the WiredCats Scouting Hub not only saves valuable time but also enhances strategic planning, improving the team's performance and success in competitions.

## Terms, Acronyms, Glossary

FIRST: An acronym for “For Inspiration of Recognition of Science and Technology.” Global youth robotics organization.

FRC: First Robotics Competition.

FIRST API: API provided by FIRST with competition data.

Team: a group of students that design, build, program, and operate a robot.

Match: A game played between teams scored by a referee for points.

Alliance: A group made up of 3 teams that compete together during a match.

Match Report: Graphical output from data gathered on a match between teams.

Scout: Student sent to watch a match and take down data from that match

AppSheet: A tool for creating WYSIWYG mobile applications built on top of Google Sheets.

# Problem Analysis and Research

## API Data

The Scouting Hub app we have created is meant to store and present data to the students by taking in two inputs: a CSV and an API result. The WiredCats manually collect the CSV data through AppSheet, whereas the API data is obtained from an external source and contains scheduling information. A significant portion of our initial planning stage was dedicated to deciding the optimal way to combine the data and present it in the app. Another large portion of our planning stage was deciding the best place to get our API data from in the first place.

Our exploration of potential API data sources identified two prominent contenders: the Blue Alliance API and the FIRST API. A thorough comparison revealed that the Blue Alliance API operates by querying the FIRST API, introducing an unnecessary delay in data retrieval and incorporating additional, non-essential information that did not align with our requirements. We chose to source our data directly from the FIRST API. This decision ensures immediate access to raw data, optimizing the timeliness and relevance of the information available to students through the Scouting Hub. By prioritizing direct access to the FIRST API, we significantly enhance the application's efficiency and eliminate the redundancy and delays associated with intermediary data sources.

## Internet Connectivity

Since there are connectivity constraints at competition venues, wireless internet access is restricted to specific zones to avoid interference with the competing robots. This limitation meant our application needed to remain functional regardless of internet availability. The application needed to accommodate out-of-date API data when the internet was unavailable and would also need the ability to parse CSVs with or without connection.

Our research led us to leverage the Electron framework for application development to address this task. Electron stood out because it could create a desktop application that could take advantage of web technologies but ultimately work online and offline. By choosing Electron, we created an application that does not rely on continuous internet access or additional microservices. This choice ensures that the application is always accessible to the students, eliminating potential disruptions to their analysis and planning during competitions.

**Commented [DW2]:** continuous internet access or additional microservices.

### Future Configuration

Another crucial aspect of our app's design was that it must have the capability to be reconfigured from year to year. Each January, FIRST redesigns the rules of their competition, changing the rules of the game so that teams have the opportunity to redesign their robots and rethink their strategies. Because of the frequently changing landscape of FIRST competition rules, our application needed a mechanism to adapt to such changes.

When researching, we realized that a common way to address similar problems in other applications is to make use of a dynamic schema. This solution allows data to have a structure while letting users reconfigure it as their needs change. In particular, we implemented this functionality by letting users import their data and then assign each data element to a specific category. These categories remain consistent from year to year, meaning that we can design other components of our app to work with them and, by doing so, prevent any underlying code from requiring modifications.

## Requirements

### Laptop Functionality

To ensure we were creating an app that was accessible and user-friendly, we identified cross-platform compatibility as a requirement. With the diversity of user preferences and the technology available to students, our application must be operational across all major operating systems. To achieve this, we used Electron to ensure the application would be compatible with Windows, macOS, and Linux. This decision guarantees that regardless of the operating system a student is using, everyone can access and utilize the application.

### Internet Connectivity

In the environment of robotics competitions, internet availability is scarce. Recognizing this, ensuring offline functionality became a non-negotiable requirement for our application. This feature was critical, and the Electron framework was the best way to achieve this functionality. It meant we could create a self-contained app that would run directly on the laptop regardless of internet connectivity, ensuring data analysis and alliance selection processes would always be available.

### CSV Data

As we began the development process, our client made it clear to us that they wanted our application to have the ability to interface with their previously built systems. With this in

mind, the team and the client mutually decided that one requirement for the app would be to import data in a CSV format. Considering that the structure of the WiredCats's data changes from year to year, this is not as straightforward as it initially might seem; however, we were able to implement a dynamic schema that can accommodate a variety of CSV formats and adapt to the changes in each year's FIRST robotics game.

### API Data

One aspect of our app that our client indicated was a requirement was the ability to pull scheduling data from an external API. They wanted the application to understand the schedule for a FIRST robotics competition without necessitating the manual entry of a competition's schedule. With this in mind we decided to implement the ability for the application to access the FIRST robotics API. The app would need to pull the necessary data and combine it with the data imported from the CSV to provide a comprehensive analysis.

### Match Reports

As the project began, our client indicated that one output that our app must be able to produce is that of a match report. This single-page report displays summary statistics and charts regarding the six teams competing in a given FIRST match. It is also necessary for this report to be readily printable so that members of the WiredCats can easily distribute copies of the generated reports even in an environment where access to a wireless connection is difficult or impossible.

### Alliance Reports

Another output that our client required for our application was that of an alliance selection report. Similarly to the match report, this single-page report presents various statistics for FIRST robotics teams. However, this report ranks every team in the competition in various categories, displaying the strongest teams at the top of the report and weaker teams in sequence below them. As with the match report, it is also required that this report be printable.

## Standards and Constraints

### Applicable Standards

Our application must be able to run on any reasonable host machine, including the ability to operate with full functionality on MacOS, Windows, or Ubuntu.

Modifications to the application's structure require changing its HTML. Modifications to the application's functionality require updating the project's JavaScript codebase. Formatting and layout modifications require changes to the CSS that defines the application's stylistic design. Note that interfacing with any data the application keeps in local storage will also require using SQLite syntax.

If additions are made to the application in the future, they should ensure that the project continues to follow SOLID principles. Furthermore, care should be taken not to unnecessarily bloat the application's size, considering that it is already relatively large due to being built on top of React and Node.

## Constraints

### Cost

One constraint for our project was that it must be free. The WiredCats are a non-profit organization that runs entirely on donations; therefore, any costs would make the project ineffective for them. This precluded us from using any hosting services, necessitating the exclusive use of local storage.

### Scope

Our project's primary scoping constraint was that it would only be used to import data and produce reports on that data—the application itself would not collect any data directly. This constraint was put in place to ensure that the application interfaced smoothly with team 5675's existing systems for data collection.

Furthermore, our project was also scoped to have functionality that works across years—although we were not expected to provide complete customizability for every possible scenario, having the ability to adapt the project to differing scoring systems was a must.

### Time

The time constraint for developing our application was that it must be completed between the beginning of WMU's 2023 Fall semester and the end of WMU's 2024 Spring semester.

## Ethical Analysis

### Identify the Moral Issue

This project has two main moral issues. First, we are creating this app only for a single FIRST team. This might compromise the competitive integrity of the FIRST competition if we unfairly advantage team 5675 over other teams. The second potential issue is that we are collecting data about other teams without their permission. This could violate their privacy,

considering we have not asked them whether collecting data on them or their robots is okay.

## Identify Additional Facts Helpful to Making the Decision

A few additional facts would help provide context as we consider whether this project is ethical.

What are the rules of the competition? If there are any rules we would violate by creating this app, that definitively points to a violation of competitive integrity. However, if there are no competition rules that would be violated, that's a point in favor of the creation of the app—although, of course, just because it is not explicitly listed as a violation of competition rules doesn't mean it is ethically allowable.

What options do other teams have available to them? Will they be unfairly disadvantaged if team 5675 has this tool available? If no similar options are available to other teams, team 5675's access to an app to aid their scouting abilities could give them an unfair advantage. However, if other teams have access to similar applications—merely without the same level of customization—that would point to the creation of the app being permissible.

## Identify the Alternatives Available

The choice facing our group is whether to design an app for the 5675 FIRST Robotics team to aid them in scouting their competition. Initially, it might seem that this is a binary choice—either we create the app or refrain from doing so. However, when analyzed more closely, it becomes apparent that there are more nuanced alternatives available that could allow for a compromise.

One possibility would be to design a scouting app as team 5675 specifies and make it publicly available to all FIRST teams. This possibility neatly sidesteps the issue of potentially providing a competitive advantage to team 5675; however, it also has disadvantages. If we made the app openly available, the issue of privacy would immediately become much more pressing. We might be required to track usernames, passwords, email addresses, or other personal information if we begin to allow other teams to use our app. Overall, this alternative introduces more problems than it solves.

Another option would be to create the app using only data that is already available to the public online rather than incorporating any data that is uniquely the result of team 5675's work. This would ensure we don't run into any privacy issues, but it would also sacrifice the effectiveness of our app.

## Identify the Personal Impacts to the Decision-Maker

The decision-makers in this context are the senior design group members working on the application. There are two main ways the decision-makers might be impacted by their choice of whether to proceed with the app's development.

Perhaps the most prominent external motivation for the decision-makers to create the app is that they are programming it as part of their academic degree. Because they are being graded on their work, they have a vested interest in seeing the project through to completion.

Furthermore, the decision-makers also want Team 5675 to succeed. We have a personal relationship with the team and its members, meaning we want them to do well competitively. Although their success does not directly impact us, we are clearly biased toward creating the app because it helps them.

## Ethics Assessment According to Contemporary Theories of Business Ethics

### *Shareholder Theory*

The shareholders in this situation would be the members of team 5675. Viewing this problem from the perspective of Shareholder theory, we should focus solely on helping Team 5675 achieve its goals. According to their request, that means creating an application for them that can "collect and display data." Providing them with the best possible tools to compete in FIRST robotics competitions is our primary way to benefit the shareholders of team 5675. That means we should fulfill their request and create the application to their specifications, regardless of the impact on other teams or the external environment.

Viewed through this lens, creating the app is ethical. Under shareholder theory, it is not relevant that the app might put other teams at an unfair disadvantage, nor is it relevant that the data used in the app's operation might infringe upon the privacy of others. Because it would provide an advantage to team 5675 in their competitions—benefiting them by increasing their chances of victory—creating the app is ethical.

### *Stakeholder Theory*

From the perspective of stakeholder theory, we must consider not only the interests of Team 5675 but also the interests of other FIRST teams and the world at large. Stakeholder theory is where some potential ethical issues with app creation become apparent.

### Utilitarianism

When looking at a problem through the lens of utilitarianism, we must determine what choice will provide the greatest benefit for the greatest number of people while

simultaneously ensuring that costs and harm are minimized. Applying this theory to the problem of whether to create an application for the WiredCats provides a mixed perspective. Although there are clear advantages to creating the app for team 5675—as we explored earlier when discussing shareholder theory—there are also disadvantages for other teams.

Unfortunately, victory in FIRST robotics competitions is a zero-sum game: Team 5675 doing better necessarily means that other teams are doing worse. From this perspective, creating an effective app for the team would benefit a small number of people (a single team) while harming a large number of people (all other FIRST teams).

That said, the zero-sum nature of FIRST is well-known to all teams competing, and it's an expected part of the competition that each team will do their best to win. It's perfectly reasonable for the WiredCats to take advantage of all resources available to them in their pursuit of victory.

Regarding the app's privacy issues, this is unlikely to harm other FIRST community members. The information is only used internally by the team and is not shared openly. Furthermore, even in the event of a data leak, the information we collect is innocuous, noninvasive, and publicly available: were it to be widely publicized, there would still be no harm to anyone involved.

In summary, an analysis of the ethical issues through the lens of utilitarianism provides mixed results. Although utilitarianism clarifies that the privacy concerns inherent in the app's design are a non-issue, it also shows us that by helping team 5675, we are necessarily harming all other FIRST teams. Despite that, it's perfectly within the competitive atmosphere of FIRST to advantage a single team at the expense of others. Overall, utilitarianism does not provide a clear or unified answer on whether creating the WiredCats's application is ethical.

### Rights

When considering how the theory of rights applies to the decision to create an app for the WiredCats, two distinct non-basic rights must be considered: the right to fair competition and the right to be free from surveillance. Although, at first glance, it might seem that these rights are being violated, a more careful analysis shows that neither is genuinely under threat.

Regarding the right to fair competition, it may seem that creating the app provides an unfair advantage to the WiredCats. To be sure, the application does provide an advantage to the team, but characterizing that advantage as "unfair" would not be appropriate. There are applications available online that provide a similar service, albeit without the level of

customization requested by the WiredCats. Because other teams are already using similar technology, the WiredCats also using such technologies does not provide them with an unfair advantage; therefore, no rights are being violated.

Similarly, the right to be free from being surveilled is also not being violated by the proposed application. Although the word "surveillance" is certainly a loaded term, close inspection reveals that although the app does collect data, it comes nowhere close to any type of surveillance. As previously mentioned, the data collected is purely related to a FIRST competition, being noninvasive and publicly available. Based on a more thorough understanding of the app, no rights are violated by the collected data.

Based on the above points, no rights would be violated by creating an application for team 5675. Taking into account the precise purpose and functionality of the app reveals that any initial concerns regarding the rights it might be violating are mistaken.

### Justice

To analyze this decision under justice theory, we must determine whether creating the app is fair or whether it shows favoritism or discrimination. Similarly to the analysis under rights theory, it might initially seem that the app violates ethical principles by advantaging the WiredCats at the expense of other FIRST teams. However, it again becomes clear upon taking a more comprehensive look at the environment of a FIRST competition that creating an app for the WiredCats does not show favoritism to them—using one's resources to support a particular team is an expected part of FIRST.

#### *Virtue Theory*

Under virtue theory, we must determine whether creating an application for the WiredCats is more likely to support or undermine human life. In multiple ways, virtue theory suggests that continuing with the app creation is ethical. It is virtuous to help others; in this case, we support the WiredCats in effectively competing. Moreover, it is virtuous to inspire and educate members of the youth, and by assisting team 5675, we are benefiting the high school students who are members of the team.

When considering the above virtues and how creating an application for the WiredCats is analyzed under virtue theory, it is apparent that creating the app is virtuous. We selflessly benefit others in multiple ways and complete a virtuous task overall. Because the app supports human life—in particular, it benefits humanity's progression in the STEM fields—it is ethical to create the app.

### Ethical Conclusions

After considering all the facts and applying contemporary theories of business ethics, our decision on whether to proceed with the project is clear. Creating an application for the WiredCats is ethical, and we will be moving forward with our project. Although an analysis under utilitarianism provides a more nuanced perspective, the overall message from other ethical theories dominates: the advantages of creating an application for the WiredCats outweigh any minor disadvantages.

## System Design

The architecture of our application consists of a series of pages that the user can navigate to in any order they wish, seamlessly switching between configuring the app, importing data, and making use of its analytic functions. Each page can complete one or more functions related to the app's operation. A configuration file is used to persist user settings and API credentials. The database is used to persist scheduling, scoring, and layouts across invocation of the application. In particular, the pages of our app are the settings page, status page, API page, CSV page, match report page, and alliance selection page.

The settings page allows users to edit a JSON file to configure the application's overall attributes. This allows the application's functionality to be updated from year to year and lets the user modify some aspects of the app's functionality without needing an understanding of the app's underlying codebase.

The status page shows the current status of the application's data; in particular, it determines whether it is accurate or potentially outdated. If the data hasn't been updated recently or it determines there is new API data available, it displays this information to the user so that they are aware they are operating with a potentially invalid data set. We chose to merely display this information from the user rather than requiring up-to-date data because perfect data is not always available in the chaos of a FIRST competition.

The API page allows the user to pull in scheduling information from the FIRST API, allowing the app to use a competition's canonical schedule without relying on the user to input that data manually. This page also stores relevant data in the database so that other pages can automatically fill in scheduling information as appropriate.

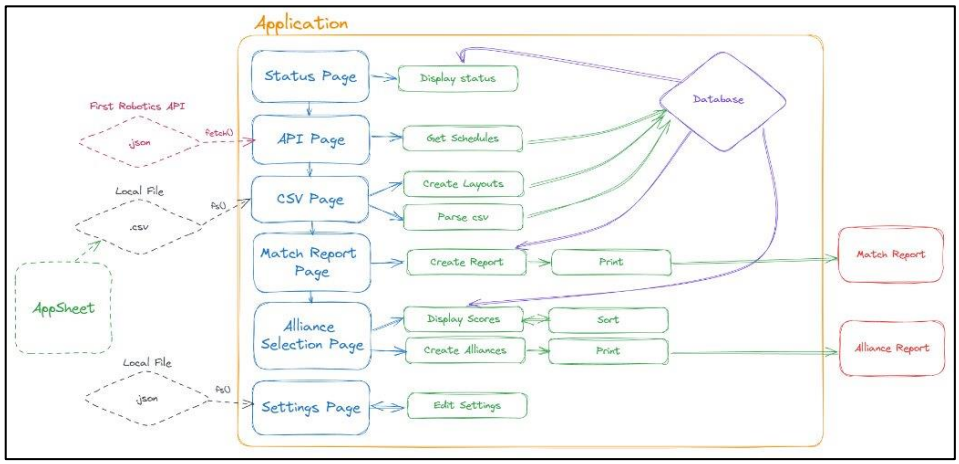
The CSV page presents an interface for the user to import data from a CSV into the application. It also allows the user to rename and reconfigure the various fields in the CSV to make them more appropriate for display in other portions of the application. Furthermore, the user can import the CSV data multiple times as they obtain more

complete data throughout a competition, with each new instance of the CSV overwriting previous data.

The match report page lets users obtain a human-friendly report on a given FIRST robotics match. It uses scheduling information stored in the database to automatically fill fields, letting the user provide a match number and have the application handle the rest. If they so desire, the user can also override the automatically filled data to account for a situation where the automatically provided data is incomplete or invalid. Moreover, this page provides an interface for printing the generated report, providing a physical copy of digital data that can be easily distributed in a competition environment where obtaining a wireless connection is impossible.

The alliance selection page shows an interface for the user to dynamically rank other FIRST teams at a competition and decide which teams are desirable. It also allows the user to combine teams into alliances, updating the list of teams to show which ones remain available. Finally, the user can print a physical copy of the alliance selection report, similar to the match report page.

The diagram below shows how all the pages in the application fit together and how they interface with the database and components external to the application.




## Testing

We employed multiple measures to ensure that our project was thoroughly tested. We took advantage of continuous interactive testing, a JavaScript unit testing framework, and repeated quality assurance checks during the code review process. By using a

combination of multiple forms of testing, we were able to iterate quickly while also ensuring our project's integrity was not sacrificed.

When developing, the first test that any new code must pass is the simple heuristic of whether it visibly breaks our application. Taking advantage of Electron's real-time nature and the frameworks that support it, we were able to see the impact of code changes in real-time.

Once a given addition to the application is deemed feature-complete, we then used Jest to verify that the addition does not cause any functionality regressions. Jest is a lightweight JavaScript unit testing framework that sat on top of our project and allowed us to quickly check that functions respond consistently and accurately to any given input.

Before merging code into our main branch, there is one last test that any code must pass—another member of the development team must review it. Although we didn't institute a hierarchical structure of code developers and code reviewers, ensuring that there is always a second set of eyes on any new additions to the code base guaranteed that a mistake or oversight on the part of any single individual wouldn't cause the project to become non-functional.

By instituting this multi-layered approach to testing, we could maintain and feel confident in the integrity of our project with little additional effort.

# Results

## Realization of Requirements

Our team was able to meet all requirements set forth by the client at the beginning of the project. This includes all our major requirements explained in this report, such as internet connectivity and gathering API data. There were no additional requirements requested through the project development.

The final version of this application allows CSV parsing and API data calls to provide detailed data for processing. It also provides detailed reports that students can directly interact with in the form of tables and graphs, and it allows for printing of the data as well. All functionality of this application replaces the manual data concatenation and processing that the WiredCats were doing by hand in previous years.

## Realization of Standards and Constraints

During the completion of our project, we effectively adhered to its standards and constraints. Our development process, and especially the testing strategies that we

employed, ensured that we were always on track to complete our project with all requirements fulfilled.

In particular, our completed application can function on any operating system, meeting the standards expected of us. We accomplished this through the use of Electron, which is web-based and easily portable to a system of the user's choice. Furthermore, the languages we used in our project—HTML, JavaScript, and CSS—directly support this constraint and allow us to easily address the requirements levied upon us.

We were also able to meet the constraints of our project. Specifically, we completed this project with no costs, on time, and while meeting all the scoping requirements specified by the client.

### Testing Results

During development, our testing methods were effective in verifying the application's continued functionality, integrity, and consistency. By the end of the project, the testing we used gave us confidence that WiredCats's app would continue to operate effectively.

By harnessing the unique real-time nature of our project's framework, we increased the speed of development by providing instantaneous feedback whenever modifications occurred. Moreover, by using a combination of automated and manual testing, we gave developers the ability to quickly iterate while also ensuring that any modifications to the codebase were verified by multiple contributors. Overall, our multi-pronged approach to testing provided numerous benefits and enhanced our ability to produce reliable code expediently.

## Future Work

Although our application meets the requirements and constraints of our client while remaining within scope, there are still some aspects of our design that we would improve further, given more time and resources. Below, we briefly discuss the design patterns we've employed to ensure that future modifications are achievable as well as some of the features that we would modify or add to our application.

### Modular Design

Fortunately, our team was introduced to SOLID design principles early in development, and we took those principles to heart. We've done our best to ensure our app is easily extensible without requiring significant modifications to previously coded components.

Perhaps most notably, we've implemented the single-responsibility principle throughout every part of our codebase, even going so far as to have it inform how our application is presented to the user. Each page of the app presents the user with a unique way to interface with its underlying data—we've kept the functionality on each page focused and unique, with the database being used to provide statefulness and transfer data between pages. This means adding additional functionality merely requires adding a new page, allowing previous pages to remain as they are.

By utilizing SOLID principles in our application, especially the single-responsibility principle, we have provided opportunities for future developers to efficiently expand upon the work we have already completed.

## Direct Data Collection

One additional function we would have liked to implement for the app would have been the ability to gather data directly rather than relying on external data sources to be imported. This functionality was briefly discussed at the beginning of the project, but the team and the client mutually decided to leave it out of scope for this iteration of the application.

That said, we've left scaffolding in place that would allow this functionality to be added to the app in the future. As discussed previously, adding a new page that allows for this functionality would not be an overwhelmingly difficult task. Furthermore, we've also made sure that our database objects are generic and universally serializable, with the ability to coerce any source of data to fit our application's requirements.

Given enough additional time, we believe this would be a valuable addition to our application. Should future developers wish to add this functionality, it would ease the process of getting data into our application by forgoing the time-consuming process of importing external data.

## Long-Term Data Storage

Another feature that would be a valuable addition to our application would be the ability for it to store data over the long term. Currently, our application only uses a host machine's local storage for all database components. This approach was chosen because one of our scoping requirements was that the application not incur any recurring costs, which makes server costs or hosting fees prohibitive.

We recognize, however, that if this capability were funded and implemented, it would be a noticeable improvement to the application's overall functionality. It would be ideal to have data available regardless of a user's device. Given enough resources, the ability to store data on a server or in the cloud would be a greatly beneficial addition.

# Conclusion

Before the creation of the WiredCats Scouting Hub, the WiredCats relied on a patchwork of processes to import, collate, and display their data. As a client, the WiredCats asked our team to create an application that interfaces with their existing systems while automating much of the work that they previously performed manually. We successfully created the WiredCats Scouting Hub to address the WiredCats' problems with their previous systems.

Primary requirements for the project were nominally related to the nature of FIRST robotics and the unique environment of a FIRST robotics competition. For instance, because of the lack of wireless connection available at FIRST robotics competitions, it was a requirement that our application function without the user of a wireless signal. Furthermore, it was also a requirement that our app's reports be easily printable, for similar reasons. Additionally, because our application is designed to interface with the WiredCats's previous systems while using external data sources to aid in automation, combining data from imported CSVs with data from an external API was also necessary.

Users of our application navigate through various pages to access our app's full functionality. They can configure it, letting the application adapt to changes to the FIRST robotics ruleset from year to year. They also can import data from multiple sources, including a CSV from Google Sheets and the external FIRST API. Furthermore, all of this functionality must be able to occur regardless of the host machine's operating system.

The completed application meets all the requirements and constraints laid out by our client while also following all appropriate standards and staying within scope.

# Appendices

## Project Management Plan

Our team used multiple approaches and systems to ensure easy communication between all team members. This allowed us to organize ourselves efficiently and effectively and assign each member specific and actionable responsibilities. By using a combination of techniques, we were able to systematically make progress on our application.
During the beginning of development, the team quickly established our use of Discord as a primary method of communication, allowing a convenient way to coordinate our disparate schedules. We used Discord as a medium to set up in-person meetings and as a platform to host virtual meetings in situations where face-to-face meetings were impossible.
Speaking of meetings, the team fell into the habit of frequent—albeit not regularly scheduled—meetings. By having frequent, brief meetings, we were able to ensure that

members of the team were always aware of the project's current status. This let us keep ourselves up to date on the most relevant tasks, meaning that all individuals were working on a relevant task without duplicating any effort.

Another way that we organized our team and secured the effectual completion of our application was to ensure each team member had a clearly defined role and set of responsibilities. Some roles were officially designated, and some came about organically as we better understood each team member's unique strengths. A list of some of these roles is provided below:

- Team Leader: This person is responsible for coordinating the actions of the other team members and casting the tiebreaking vote in situations where the group is split on how best to proceed.
- Client Liaison: This person is responsible for interfacing with the client, scheduling meetings with them, and translating industry-specific jargon into terms familiar to other team members.
- Secretary: This person is responsible for taking meeting minutes, tracking tasks that are behind schedule, and submitting progress updates.
- Coding Lead: This person is responsible for thoroughly understanding all technical aspects of the project. We found this role necessary to aid in communication between team members and prevent information siloing.
- Stylistic Lead: This person is responsible for ensuring the application's desirable look and feel and maintaining appropriate use of WiredCats branding.

If there were anything to improve, it would be to make more effective use of task-tracking software such as a Kanban board or GitHub issues. Although we used such features, this only occurred late in development. We could not take full advantage of these organizational tools, and in hindsight, there were moments when we could have significantly benefited from them.

Despite these minor foibles, our project management plan effectively kept our team organized and on track. We were able to effectively delegate responsibilities to the team members best able to handle them and complete our project in a way that was satisfactory to us as engineers and to our client.

## Progress Reports and Minutes

# Meeting Minutes for 2023-09-17 <!-- ISO-8601 format -->

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-09-17 <!-- ISO-8601 format -->

Meeting Time: 13:40-15:10 <!-- 24-hour time -->

Meeting Location: Floyd Hall

Attendance: <!-- Include team members and sponsors. -->

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong

- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley
- Chris Cross

Minutes taken by Allie Kalhorn and edited by Sebastian Smiley

## Topics of Discussion

- Clarifying terms
  - Team: a group of students that design, build, program, and operate a robot.
  - Alliance: three teams that compete together during a match.
- App Goals
  - Collect data in real time from online resources
  - Perform analysis on the data
  - Make predictions based on analysis
- What data are we collecting?
  - First Data API
  - Examples of current third-party APIs: TheBlueAlliance, Statbotics
  - Looking to visualize this data
  - Available data is generic--we want to provide specific analysis
- Hopes/Needs: we want...
  - A way to filter teams to determine the best ones
  - The ability to input own data
    - How to match up API data with user-provided data?
  - Configurability from year to year

- Data to be transported from online sources to a local database
- A prioritized list of alliance partners
  - Filters to determine this based on user-chosen heuristics
- The ability to utilize our data without an internet connection
- The ability to print the visualized data in a compact report
- Something to keep up morale
  - Could keep track of what student has input the most data
  - Could provide motivational messages
- How would we like this visualized?
  - Bar charts, pie charts, tables, graphs
  - Printable format
  - Consider accessibility options for B&W printed reports

## Summary

### Progress Made <!-- What have we accomplished since last meeting? -->

- Set up communication methods for the team (Email, Discord)
- Contact the sponsor

### Next Steps <!-- What do we want to accomplish before the next meeting? -->

- Collate and finalize system requirements
- Determine tech stack
- Set up GitHub and other collaborative tools

# Meeting Minutes for 2023-09-27

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-09-27

Meeting Time: 17:15-18:20

Meeting Location: Floyd Hall, G Lounge

Attendance:

- Allie Kalhorn
- Alex Fitzsimmons
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu

- Sebastian Smiley

## Topics of Discussion

- Githup
  - Everyone in the github
  - Git cheat sheet can be found there
- Attendance
  - Come to class!
- Work on the software requirement specifications
  - Have enough detail about requirements
  - Need more detail about UI
- Using Electron
- What input we'll be collecting
  - CSV input from the students
  - First API
- Things we will need
  - Data import
  - User input validation
  - Component to build the schema
  - Selection screen of information they need to see
- Presenting the data
  - Historical data/partial data/full data
  - lottery phase
  - alliance data
- Lottery phase

- like fantasy football
- remove teams as they are selected
- used to make best choice for alliance

- Deliverables
- minutes
- program diagrams

## Summary

### Progress Made

- Determined we are using electron
- Got a general structure of how we want the app and database to be set up

### Next Steps

- Find good way to schedule meetings
- Work on the SRS and requirements

# Meeting Minutes for 2023-10-02

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-10-02

Meeting Time: 16:00-17:25

Meeting Location: Floyd Hall

Attendance:

- Allie Kalhorn
- Alex Fitzsimmons
- Cullen Armstrong
- Daniel Wilkins
- Sebastian Smiley

- Joshua Vattiprolu

## Topics of Discussion

- What we need:
  - Tracking who has done what over the last couple weeks
  - Getting a GUI diagram
- Who's doing what next
  - Allie : Minutes + Spin up blank electron app w/ RXDB
  - Dan : Outline of SRS
  - Sebastian : Work with client on requirements
  - Alex, Cullen, Josh: Figma/UI
  - Everyone : Fill out 2 parts of the SRS
- Design walkthrough
  - Walking through what UI of each page will look like
  - Using rough-draft-application-flowchart.png from GitHub for walk through

## Summary

### Progress Made

- Determined general UI design by page
- Determined key features and some of their sub-features

### Next Steps

- Provide electron app as example in class
- UI Design rough draft
- Get requirements confirmed with client

# Meeting Minutes for 2023-10-09 <!-- ISO-8601 format -->

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-10-09 <!-- ISO-8601 format -->

Meeting Time: 16:00-17:10 <!-- 24-hour time -->

Meeting Location: Floyd Hall <!-- MEETING_LOCATION_HERE -->

Attendance: <!-- Include team members and sponsors. -->

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Sebastian Smiley

## Topics of Discussion

- Discussed prior week's progress
  - Putting completed work into [responsibility_tracking.md](/artifacts/responsibility_tracking.md).
  - Basic RxDB with Electron: working on moving from "hello world" to more advanced use cases.
- Discussing historical workflow diagram
  - AppSheet servers pass data directly through to Google Sheets, with GS being the authoritative source.
  - Terminology: Scouting Data vs. API Data
    - Scouting Data: comes from students collecting data through AppSheet. Comprised of both Match Scouting data and Pit Scouting Data.
    - API Data: comes from FIRST API, or in the context of the WiredCats systems from prior years,
  - When Scouting Data and API Data conflict, we should use API Data since it is more reliable.
  - When Scouting Data has more granularity than API Data, we should use Scouting Data.
- Discuss UI rough draft
  - Resolution to focus on function over form for the time-being.
    - Make sure that every relevant component is on the page.
  - At some point, chart components for printable reports will need to become modular.
- Going forward
  - Important to consider how tasks will be divided.
  - How will we be creating a schema-builder and report-builder.
  - How will we be mapping JSON data and parsing CSV data so both can be used in a database.

- Need to design a system that standardizes the data from both.
- Will end up with two pages, each with the ability to view data and create a schema for it.
- Consider using a system that will automatically pull data from Google Sheets.
- But we want the fallback of being able to directly import CSV data.

## Summary

### Progress Made <!-- What have we accomplished since last meeting? -->

- Analyzed the WiredCat's old method of doing things to check how we can improve.
- Rough draft of app's UI in figma.
- Formalized system for tracking what tasks are pending/completed.
- Tested skeleton of Electron app.

### Next Steps <!-- What do we want to accomplish before the next meeting? -->

- Finish SRS document.
- Continue work on Figma rough drafts.
- Electron app with "click button to add entry to DB" functionality.

# Meeting Minutes for 2023-10-16 <!-- ISO-8601 format -->

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-10-16 <!-- ISO-8601 format -->

Meeting Time: 16:05-16:35 <!-- 24-hour time -->

Meeting Location: Virtual (Discord) <!-- MEETING_LOCATION_HERE -->

Attendance: <!-- Include team members and sponsors. -->

- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

## Topics of Discussion

- How do we get everyone up-to-speed on Electron?
  - Make sure to read basic documentation.
  - Keep in mind that Electron is a thin wrapper around a website.
- How can we nail down our requirements document.
  - We have a clear picture of our requirements, but lets get them on paper.
  - Need to ensure a distinction between client requirements and our design choices.
- Make sure all of our artifacts are uploaded.
  - GitHub--Important to make sure everyone is on the same page.
  - Riipen--Important to make sure Dr. Shen and our sponsor know what's going on.

## Summary

### Progress Made <!-- What have we accomplished since last meeting? -->

- Continuing Figma work as we scaffold the barebones of UI layout.
- Progress on requirements document based on client specifications.

### Next Steps <!-- What do we want to accomplish before the next meeting? -->

- Everyone completes a "Hello World" program in Electron.
- Continue smoothing out requirements document.

# Meeting Minutes for 2023-10-23

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-10-23

Meeting Time: 16:00-16:45

Meeting Location: Floyd Hall

Attendance:

- Allie Kalhorn
- Alex Fitzsimmons
- Cullen Armstrong
- Daniel Wilkins
- Sebastian Smiley

## Topics of Discussion

- Git
  - Going over how to use git
  - Dev branch, main branch
  - Standardizing comments in git and in code
- Partnering up for work
  - Alex + Cullen
  - Dan + Josh
  - Allie + Sebastian
- Work:
  - Alex + Cullen - create pages and navagation
  - Dan + Josh - set up csv parser
  - Allie + Sebastian - setup first robotics api

## Summary

### Progress Made

- Created and pushed a "hello world" Electron test app.
  - Read the documentation on Electron basics.
  - Followed along with Electron's "getting started" tutorial.
- Split up coding work into three sections, one section for each pair.

### Next Steps

- Setup basic calls to the FIRST Robotics API.
- Setup a method of reading and parsing CSV files.
- Setup a skeleton for site navigation.

# Meeting Minutes for 2023-10-30

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-10-30

Meeting Time: 16:00-16:30

Meeting Location: Floyd Hall

Attendance:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Sebastian Smiley

- Joshua Vattiprolu

## Topics of Discussion

- Branches
  - Merge dev branch after Monday meetings for future
- Folder Structure
  - Getting folder structure straight
  - Putting all code in one folder, moving the test app into that folder
- Last week work
  - Navbar going good
  - API call working
  - CSV parser
- Navbar
  - Make one design work
- API Call
  - Match up API data
  - Find way to visualize
  - Create table
- CSV Parser
  - Find way to visualize
  - Create table

## Summary

### Progress Made

- Initial Navbar

- API calls

- Start on CSV Parsing

### Next Steps

- Alex and Cullen: Learn to use branches, one navbar design

- Allie and Sebastian: create a table to display data from the API

- Dan and Josh: create a table to display data from the CSV

# Meeting Minutes for 2023-11-06

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-11-06

Meeting Time: 16:05-16:45

Meeting Location: Floyd Hall

Attendance:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu

- Sebastian Smiley

## Topics of Discussion

- Last week:
  - CSV Needs RPC Handler - Has to be async
  - API pairing and table looking good
- Instructor Feedback
  - Need sequence diagram and traceability matrix
- Requirements
  - Refining the requirements
- Flowchart -> Matrix
  - break down to create traceability matrix?
  - use draw.io to create the traceability matrix
- React?
  - discussion on if we swap to using react instead of just javascript

## Summary

### Progress Made

- APIs combined
- API table example started
- CSV Parser

### Next Steps

- Create major/minor requirements
- Create a sequence diagram for printing requirement
- Everyone look into React

# Meeting Minutes for 2023-11-13

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-11-13

Meeting Time: 16:00-16:30

Meeting Location: Floyd Hall

Attendance:

- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

## Topics of Discussion

- Requirements
  - Set up the more specific requirements
  - Will help with the traceability matrix
- Attendance
  - For meetings AND class
- Flow Chart
  - Work on it for next week!

## Summary

### Progress Made

- Refined requirements

### Next Steps

- Flowcharts
- Refactor code

# Meeting Minutes for 2023-11-13

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-11-13

Meeting Time: 16:00-16:30

Meeting Location: Floyd Hall

Attendance:

- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

## Topics of Discussion

- Requirements
  - Set up the more specific requirements
  - Will help with the traceability matrix
- Attendance
  - For meetings AND class
- Flow Chart
  - Work on it for next week!

## Summary

### Progress Made

- Refined requirements

### Next Steps

- Flowcharts
- Refactor code

# Meeting Minutes for 2023-11-20

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-11-20

Meeting Time: 16:00-16:45

Meeting Location: Virtual over Discord

Attendance:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu

- Sebastian Smiley

## Topics of Discussion

- Code turned to react
  - Changed code over to react instead of just JS
- Split up work
  - Selected sub-requirements to create sequence diagrams

## Summary

### Progress Made

- Flowchart components

### Next Steps

- Each person take their part of the chart to create sequence diagram

# Meeting Minutes for 2023-11-27

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-11-27

Meeting Time: 16:00-16:25

Meeting Location: Virtual Over Discord

Attendance:

- Alex Fitzsimmons
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

## Topics of Discussion

- Note that UML diagrams are heavily based on class-based design.
- Consider how granular we want to get with our sequence diagrams.
  - Diagram things at the page level for now.
  - We can break things down to the object level later if necessary.
- When working on an Electron app, non-static fonts/images can have difficulties loading.
- Best practices
  - Keep in mind coding styles and uppercasing/lowercasing guidelines.
  - Keep code clean and readable by organizing folder structures appropriately.
  - Consider a file-formatter/linter for spring semester.
  - More discussion to follow in spring semester.

## Summary

### Progress Made

- Diagrams based on last meeting's notes.
- Everyone got an up-to-date version of the Electron app running locally.

### Next Steps

- Continue working on diagrams.
- Ensure all aspects of design are broken down into individual components.

# Meeting Minutes for 2023-12-04

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2023-12-04

Meeting Time: 16:20-

Meeting Location: Floyd Hall

Attendance:

- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Sebastian Smiley

## Topics of Discussion

- Consider next steps for sequence diagrams.
- Showcase each person's sequence diagrams and get feedback.
- Understand how to represent database connections in sequence diagrams.
- We might be doing some microtasks over Winter Break.
- Consider how pulling data using the API occurs in practice.
  - How is blank data represented?
  - How will we store undefined/null values?
- Take note that kickoff for the 2024 FIRST Robotics FRC season is January 6th.

## Summary

### Progress Made

- Sequence diagrams created.
- Taught team members how to commit using Git.

### Next Steps

- Create a more thorough README file to allow understanding from others.
- Upload remaining sequence diagrams to GitHub.
- Meet early before next Friday's meeting to do a final check.

# Meeting Minutes for 2024-01-17

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2024-01-17

Meeting Time: 18:00-18:25

Meeting Location: Virtual

Attendance: <!-- Include team members and sponsors. -->

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu

- Sebastian Smiley
- Chris Cross

## Topics of Discussion

- Present the current status of the project
  - Pulls data for the 2023 REST API
  - Displays sample data in charts
- Expectations clarification
  - We are just reproducing the process from the manual spreadsheet process
  - We are designing the program to be component based for easy upgrades and changes
- Showcase of UI design
  - Showed mockup for match selection
    - Talked about the two ways to selection a match
  - Showed mockup for alliance selection
    - Talked about how the tools automates the manual process in real time
    - Client really like this feature

## Summary

### Progress Made <!-- What have we accomplished since last meeting? -->

- Gave an update on the projects to the client
- Received feedback on design path from client

### Next Steps <!-- What do we want to accomplish before the next meeting? -->

- Meet with instructor to determine semester changes to project requirements
- Produce a working prototype that works off 2023 data
- Deliver prototype to client to received process compatibility feedback

# Meeting Minutes for 2024-01-18

Group Number: 5

Sponsor Name: 5675 WiredCats

Project Team Members:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu
- Sebastian Smiley

Meeting Date: 2024-01-18

Meeting Time: 10:45-10:55

Meeting Location: Floyd Hall

Attendance:

- Alex Fitzsimmons
- Allie Kalhorn
- Cullen Armstrong
- Daniel Wilkins
- Joshua Vattiprolu

- Sebastian Smiley

## Topics of Discussion

- Ethics Analysis (Lampposts)
  - How can we breakdown this work? Resolution: heading by heading.
  - Ensure we are all working toward a coherent conclusion.
- Prototype
  - We need to create a functional prototype to show our client.
  - Focus on top priorities for something workable.

## Summary

### Progress Made <!-- What have we accomplished since last meeting? -->

- Team members reviewed lampposts article

### Next Steps <!-- What do we want to accomplish before the next meeting? -->

- Outline for lampposts article written
- Fill in outline based on agreed-to breakdown of tasks
- Write a series of product requirements for a minimum viable product

# CS 4910 – Software System Development and Design II: Implementation and Testing

## Project Progress Report

### Project Information:

**Team Members: Allie Kalhorn, Sebastian Smiley, Daniel Wilkins, Cullen Armstrong, Alex Fitzsimmons, Joshua Vattiprolu**

**Client: Mattawan WiredCats**

**Advisor: Jason Johnson**

**Report Date: 1/29/24**

### Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

Team met to discuss next steps, how we will break down the work, etc.

### Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

Met with the client on 1/17, we discussed what our current status is for the project and what the goals and expectations were for the next time we meet.

### Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

Our next goal is to create a working prototype for our program to demonstrate to the client. We are on schedule to get this completed.

### Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

We are currently working on parsing API and CSV data.

### Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

We currently have not run into any problems or risks.

# CS 4910 – Software System Development and Design II: Implementation and Testing

## Project Progress Report

### Project Information:

**Team Members: Allie Kalhorn, Sebastian Smiley, Daniel Wilkins, Cullen Armstrong, Alex Fitzsimmons, Joshua Vattiprolu**

**Client: Mattawan WiredCats**

**Advisor: Jason Johnson**

**Report Date: 2/9/24**

### Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

Team set up milestones on Github, worked on the graph components of the project and did some work on the practice project.

### Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

We did not meet with our client between this progress report and the last progress report.

### Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

Our next goal is to create a working prototype for our program to demonstrate to the client. We are on schedule to get this completed.

### Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

We are currently working on getting our graph components set up and working on the practice project.

### Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

We currently have not run into any problems or risks.

# CS 4910 – Software System Development and Design II: Implementation and Testing

## Project Progress Report

### Project Information:

**Team Members: Allie Kalhorn, Sebastian Smiley, Daniel Wilkins, Cullen Armstrong, Alex Fitzsimmons, Joshua Vattiprolu, Samin Al Mahi**

**Client: Mattawan WiredCats**

**Advisor: Jason Johnson**

**Report Date: 3/12/24**

### Team Activity Report:

*What has your team done since your last report. Indicate team meetings with a brief description of what was discussed, and a breakdown of any other activities your team engaged in since your last report.*

Set up page for Alliance selection, specifically tables with team data loaded in. We have also worked on the navigation of the application and organizing our GitHub repository.

### Client Interaction Report:

*Have you met with your client since your last report? What was discussed? What feedback did you client give you on your progress? Did you demonstrate a prototype?*

We did not meet with our client between this progress report and the last progress report. We attempted to meet with the client on 3/1 but he was not able to attend.

### Milestone Review:

*Briefly describe the phase of your project that you are currently working on. What is the planned date of completion for this part of your project? Are you ahead of schedule, on schedule, or behind schedule?*

Our next goal is to finish creating a working prototype for our program to demonstrate to the client. We are on schedule to get this completed.

### Issues (or stories):

*What issues are you currently working on? These are smaller tasks that are part of accomplishing your current milestone. They are also referred to as stories.*

We are currently working on getting our graph components set up and the Alliance Selection page. There is also a status page that is being worked on. We are mostly working on the UI elements of the project so when we load data in we'll be able to see it as intended.

### Problems and Risks:

*What problems have arisen, if any? How do you plan to address these problems and stay on schedule? Do you foresee any risks that may impact your project? If so, what are they and how do you plan to mitigate them?*

We haven't met with the client recently because the client wasn't able to meet so we haven't been able to show our progress and get feedback.

## Development Costs

There were no development costs associated with this project.