CREATING A WEB-BASED SOLUTION TO STREAMLINE BUSINESS PROCESSES AT UNCW's TECHNOLOGY ASSISTANCE CENTER

Nathan Davis

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Congdon School of Supply Chain, Business Analytics, and Information Systems

University of North Carolina Wilmington

2024

Approved by

Advisory Committee

Dr. Toni Pence, Member

Dr. Marni Ferner, Member

Dr. Kevin Matthews, Chair

TABLE OF CONTENTS

ABSTRACT

Creating a Web-Based Solution to Streamline Business Processes at UNCW's Technology Assistance Center. Davis, Nathan. 2024. Capstone Paper, University of North Carolina Wilmington.

The Technology Assistance Center at UNCW (University of North Carolina Wilmington) has been helping staff and students alike for many years. Some of the technology has been around for many years as well. This capstone project will go over current issues with TAC by listing problems and how they are affecting current TAC as well as why these issues need to be addressed because of things like burnout. The paper will then discuss ways to remedy the problems discussed. There are key attributes that need to be kept in mind throughout the project to keep the goal in mind. The paper then shows a list of features and planned features if time allows it. Further discussion of the main user interface, app integration, and automation technology will have a huge impact on project success. It came down to a decisive choice after looking at the options with SharePoint, Power Apps, and Power Automate being chosen for their integration together with Microsoft. A schedule of tasks to be performed and a proposed timeline is also given. The project's future will also be considered as this is for TAC and will be passed down once the first stage is complete.

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1: INTRODUCTION

The Technology Assistance Center (TAC) is a splendid work environment where I have had plenty of great experiences and learned many important lessons and skills. They hold a close place in my heart in the time I have worked for them. The people there all feel like they care about their jobs and are motivated individuals. Since my start at TAC, some of the systems are unfortunately outdated and have not been touched in many, many years. Individually this has not affected productivity enough to be detrimental to the overall functionality of TAC, but the buildup over time has left some asking if things could be implemented more efficiently.

Businesses like to promote efficiency as it is a leading factor in how well a business function may operate. This can go down to any scale and apply the same principle that improving efficiency will improve results in some form. It is also important to consider how efficiency will affect the individual workers. If efficiency is too reliant on workers, then it could cause quick burnout. “Job demands can … psychically and psychologically affect employees and result in job strain” (McCormack and Cotter, pg. 38). There are many solutions to limit worker burnout. One of the more recent ones is the automation of repetitive systems. Automation not only allows workers to do tasks that have more meaning but also allows for a refreshing job experience. Burnout is an issue for companies and leads to workers leaving or striking for a better job environment if they feel mistreated.

Because of limited rest periods, “There is no time to catch your breath. Organizations seem to be testing the theory that people can work flat-out forever. But, with no time to recover, people soon find that their exhaustion just builds” (Maslach and Leiter, pg. 38). This can have a significant impact over time on any worker. This is why it

is important to find ways to alleviate this stress. “Additional demands might be manageable if they were given more resources; extra support or equipment can turn increased demand into an opportunity” (Maslach and Leiter, pg. 38). As stated above, with extra support, it is possible to make improvements.

Another study suggests that greater control over how they handle situations at work decreases mental and physical health issues (McCormack and Cotter, pg. 38). There is an importance in letting workers have a say on how they do things when it is appropriate. When someone feels restrained and unable to do things in a way that feels appropriate, they can feel pressure put on themselves. “Lack of control can take several different forms including supervisors who micromanage or organizations that do not allow employees to participate in policy or other work-related decisions. Lack of control is particularly stressful in an environment in which downsizing and job restructuring are taking place" (Maslach and Leiter, pg. 44). During 2023, TAC was moved around frequently, and with many policy changes made without the control of the workers and staff, it was stressful. It changes how we operate completely. More stress is put on the staff with little rest time, and little say in how things are done.

Currently, some of the policies in place are being changed to accompany the staff and workers more to reduce stress. The idea of “professional autonomy” says when someone “can participate in and influence workplace decision-making" they can have a level of autonomy which we need as humans (McCormack and Cotter, pg. 43). There needs to be an important middle ground on what is appropriate and what is not. That is why looking for improvements to reduce the level of stress for the workers and staff at TAC is important, as it is for any stressful job.

With the rising cost of hiring new members with onboarding, training, etc. It is

essential to make the job experience as frictionless as possible. There is never going to be a perfect solution to this problem but incorporating many ways to alleviate current workers is important. Taking user feedback is highly recommended and used extensively in some companies. Where employee happiness is their goal. They understand that to achieve efficiency in employees, keeping them happy is a small price to pay. Bigger corporations have many ways to accommodate employees. Places like Google offer tons of benefits that most people in the field know about because of how much word of mouth gets around.

The work environment at TAC is comfortable for student workers and it allows everyone to get along well. This does not mean that people are not doing their jobs, it can be relaxed because everyone is motivated to do their job. Because of the work of the upper-level student workers and staff, it can be a more laid-back experience for newer workers. The level of work needed to be balanced by upper workers and staff can be excessive. Getting behind on tasks limits the amount of progress TAC can make on improvements so we are stuck at a standstill.

There are many ways to improve efficiency at TAC. One proposed way to do this is to make a centralized website that would benefit TAC all around. Focusing on assisting workers with questions that might be directed at the staff and automating monotonous processes. The first thought is how TAC can incorporate something like this. First, we must look at individual problems and how they are currently handled.

CHAPTER 2: DESCRIPTION OF THE PROBLEM

The following is a discussion of the current problems at TAC. In the dynamic landscape of communication and task management at TAC, several persistent challenges have emerged, demanding innovative solutions to enhance efficiency and foster a more cohesive work environment. The foremost issue revolves around the prevalence of repetitive questions within Microsoft Teams channels, leading to information overload and potential knowledge loss. As the TAC community heavily relies on Teams for communication, the need to streamline the access to previously answered questions becomes paramount. Additionally, the scattered nature of vital links and infrequently used websites pose a hindrance, with their complexity further compounding the problem.

The onboarding process for new workers is identified as a tedious undertaking, requiring manual efforts from upper-level staff and student workers. Furthermore, disseminating crucial announcements proves challenging, with Microsoft Teams serving as the primary method, albeit with its user-friendliness concerns. Repetitive tasks also emerge as a significant concern, potentially contributing to burnout among workers. In this context, the introduction of automation is considered, balancing the need for efficiency with the caution required in navigating the rapidly evolving landscape of artificial intelligence. Addressing these challenges is not only essential for optimizing current operations but also for preparing the TAC community for the transformative technological advancements on the horizon.

The Problems

Repetitive Questions. Issues with questions that are repetitive and have been answered and/or written down. On Microsoft Teams, we have channels that offer workers the option to ask someone: student workers or staff as a collective to see if anyone has

encountered the issue. These channels are recommended for use when the question may be frequently asked. This goes back to the issue of information overload. Since there are a lot of places that must be frequently visited and used it is easy to get overloaded.

This has been an issue for some time at TAC as we have been using MS Teams for communication between staff and workers for some time now. Staff have been dealing with this issue and want a way to harness all the previous questions or find a better way for users to access them. The help channel alone has thousands of messages that have been responded to with a high success rate. This is beneficial and could spark many knowledge-based articles that may not be made. The help channel also shows what worked and what did not work. This information is invaluable and would be a huge loss if the information disappeared. We must find a way to demonstrate this information. The students and staff are usually separated from each other, so we must heavily rely on MS Teams for any type of communication.

We have put in place some sort of higher-up worker or staff at the library TAC desk for assistance but no one knows everything. It has helped alleviate some of the stress of Teams. This keeps the staff who are already doing more than they should, in some cases, from needing to answer as many basic questions as possible.

Having student workers who have been loyal to TAC for many years is another way we have prevented repetitive questions, but it is not always so easy to keep people for many years. This is in part due to a better position that will benefit their future or just a better-paying job. Therefore, most of the knowledge is held by the staff and the common knowledge may be lost when older TAC workers leave.

Websites/Information Sharing. Issues with websites/information that is not used frequently enough but is still important. They are important in certain scenarios that could

save a lot of time. Sometimes certain information/websites are completely overlooked and forgotten about even in training and they are not told till later about it. We do not currently have a centralized way to store all links and why we need them. They stay scattered and forgotten about by a good portion of the workers. Places like ITSISE which checks when IPs have connected to the network are useful in certain scenarios but not used frequently.

This has always been a recurring issue and word of mouth is especially important since that is how a lot of the information would spread. Since the changes to where everyone is stationed, it is more difficult to hear or see someone using technology that was only mentioned in places like training. The website/information is used so infrequently, and workers are also full-time, students most of the time that it is just not important enough to remember. It really will become increasingly difficult to keep up with the changing environment at TAC if there is no convenient place to store what current tools we have at our disposal.

Not only are these websites forgotten about, but they also have a wide range of complexity leaving some stumped about how to use them properly. I have personally seen people, including myself, not only forget about the website but then turn around and ask how to even use the website. Having the staff / knowledgeable student worker explain to someone every time an infrequently used website is accessed is tiring. I do not work even close to full time at TAC and at the most was only working 15 hours a week and did not have questions directly to me like staff do. I can only assume that staff get bombarded with questions regarding these types of infrequently used websites constantly.

Onboarding. Issues with onboarding for new workers are currently a tedious process and require a lot of manual work from the upper-level student workers and staff.

Due to the constant change in how things are being done, it leaves much room for improvement. With little time to make those improvements with only a limited number of hours in a day and a limited number of workers who can do anything about these improvements. Workers with more responsibility and assigned multiple tasks must put their tasks on the back burner, which is unproductive and inefficient.

One of the positions at TAC is the main onboarding position that makes sure the new workers are well-equipped. They are not full-time staff but are close to it in some cases with similar responsibilities. From my personal experience, they are all great people and do an excellent job at work, so they are given a lot of tasks. It can be quite overwhelming and prone to burnout if tasks take too much of an extended period. This leads to a less productive and less energetic worker. It cannot be completely blamed on them. In some cases, the workload cannot be distributed more evenly. This is why it is necessary to help alleviate the stress they have at certain times of the year, mainly right after hiring new workers by making onboarding more streamlined.

Announcements. Issues with getting information across to a large audience like announcements (such as time sheets and due dates). Currently, Microsoft Teams is the method to get information across, and the occasional email. Not a lot of people check Teams constantly and sometimes not at all. I have personally had issues with getting Teams to work at all on certain computers. MS Teams is quite frustrating to some of the workers and does work well but can sometimes not be so user-friendly. Not everyone shares this sentiment, but I am sure there is a better way to handle these announcements for TAC specifically.

Many times, during the year, we will have to do tasks like time sheets or making time off requests. This process should be more streamlined and easier for a worker/staff

to do so they do not keep putting it off. I have seen some level of integration with the current system and When2Work which is a good step in the right direction. I know that I see the announcement, but only at a glance, and I wait until later unless I am thinking about it when nothing else is going on. I usually just help someone and do not have time right away in a shift to complete the task, so it slips my mind and others as well.

Some announcements hold more value than others but are currently mixed in with everything. It will be our top priority to get those high-value announcements quickly and effectively to the workers so that they will have a more constant reminder. We do not want to have it shoved in their face too often. Just like a pop-up on your computer, if you see them too often, they are completely ignored, that is just how the human brain works.

Repetitive Tasks. Issues with repetitive tasks. As mentioned earlier, repetitive tasks can lead to burnout and lower efficiency in unmotivated workers. This touches on many of the other issues and can lead to improving performance overall. Many tasks at TAC are currently repetitive. Things such as making requests to staff, staff needing to answer the same questions, and small things causing constant mental resets to get back to work lower efficiency and could be improved upon.

CHAPTER 3: PROPOSED SOLUTION

The following is a description of possible solutions to the problems with the current processes at TAC. For each problem, I discuss an approach that is suitable for what we want in the system. It will discuss the feature that was mentioned in the corresponding problems section. Some will be an individual feature and others will be a process change that is necessary to fix the issue.

Approach to Remedying the Problems

Repetitive Questions. The way I want to go about improving upon a lot of the slowdown at TAC is to make a website that works on at least improving these core issues in the system. Having a centralized location that can accommodate a lot of these problems in one way or another will increase efficiency. Having to click around less and having to look around on different platforms less constantly will help with familiarizing to a centralized resource.

Going back to asking questions on teams for help is beneficial for workers but does cause issues when the same answers are not being looked at afterward. The number of questions asked on these help channels is quite large and is not being utilized very well. I hear the staff telling us to use it as a resource, but it is rarely used. Adding a feature to ask questions through a search function where you can look at the Teams channel and KB article titles among other useful information instead of looking in various places sounds like a good improvement on paper to me.

Website/Information Sharing. Having to find links to websites that are not publicly available is another weak point when it requires asking staff. Making a centralized location where links can be stored and updated would improve efficiency. There should be a brief explanation of how to use each website efficiently since some

websites are not straightforward about this. Just giving the link to the website does save time but not explaining the importance of each one will also lower its chance of being used if no one knows when it is useful. It is important to make sure that everything applies at the newest workers' knowledge level to prevent things from needing to be explained.

Onboarding. Making quizzes that can be submitted and are easy to create would be a good system. We do have a canvas page for tests which works well so we could just add a link to that if possible or something similar. It could be implemented with tasks through MS Teams. Another feature we want to add is a way to train workers for chats because currently, someone must manually do that process. I do not know how plausible something like AI would be but that is if time is abundant to investigate that feature.

We currently use Canvas at TAC to regulate tests for all the student workers. There are tests for being qualified to do certain tasks at TAC and to be promoted as a student worker to do things like remote work. It works very well, and I have not had any issues with it. The more prominent issue is with training new workers for online chatting with clients online since they need to be able to respond in a reasonable amount of time. They should be able to answer an assortment of questions on their own and only have issues with edge cases. There is a lot of information to remember but we want to keep the basic questions to a minimum or at least show them how to get that information themselves. The website/app should improve this.

Announcements. It would be a good idea to try and make an announcement board that would help with putting the information a user needs to know at the front of the page instead of having to go to Teams which some people are reluctant to use at times. Having it on the website when you first enter would make it streamlined. They must go on the

website to reach all the other features, so the odds of someone seeing the announcements go up. Integrating it so that the manager can make the announcements on Teams so that it goes directly to the website or make it so that the manager can make announcements on the website.

I would like to only show the most vital information on this announcement board. We could have another board on a section tab that shows other less immediately important information. This would help us, as humans, filter out the announcements board altogether since we do not want to flood workers with information that may not directly affect them like a technical issue that is only affecting a tiny portion of the shift

Another idea for showing technical issues is to be able to post the issue, and once resolved, a staff member could remove it with their permission. It would show just the issues for workers and a small description of what the problem is. The project has a lot of small functionality that makes for a significant impact, it is a sum of all its parts.

Repetitive Tasks. One of our big goals is to automate as much as possible where it would benefit the workers and improve efficiency. For many of these processes, we want to automate the steps or automate them. The goal is to be available for everyone to get their tasks completed and allow workers to start other tasks sooner. It would benefit the upper-level workers and staff because they would have more time to focus on what they are doing. Standardizing a way of doing something also improves efficiency as a feature stays consistent. I want an overall cleanup of tasks that are long overdue for automation as I am sure the other staff and workers are too.

Some skeptics are worried about how automation will affect their jobs. Some jobs are more prone to automation but, it should not be seen as bad technology. At TAC, automation will mean more time for our staff and student workers to do things that are

more enjoyable because no one likes to do a monotonous job all day long because of eventual burnout. There can be a case made that some people enjoy that monotonous task, but the consensus would not agree.

Automation is the future and should be used with some precaution since it is still a rapidly changing environment. In a few years, we could see a major shift that would be completely unknown to us today. The rapid change in AI (Artificial Intelligence) has shown how unexpected technology can be and this sentiment will only grow as the next boom in the world's technology era starts with AI being the focal point.

User Stories

When making these user stories, I consulted with the manager and staff at TAC. We sat down and discussed each user story and made sure they all made sense and if I missed anything. From what we gathered, the current list of user stories seems sufficient for getting the project started. At the beginning of the project, I asked my manager what she wanted to see in the web-based application. She listed off the attributes she wanted to see in the site like searchability, maintainability, permission-based, secure, and scalable. We plan to keep these things in mind when making the project.

When first sitting down with my manager, we went over a basic template that she had made just showing what basic design she had in mind. From there we wrote user stories to help to capture what was on the template in greater detail. The staff and I have been coming up with ideas to add to the site, but I have been working on what new features would be worthwhile to make. Here is a list of the basic user stories we have come up with so far.

Mobile Friendly:

- As a user, I want to be able to access the app from my phone, so that I can

sign in without having to go onto the website

Information Acquisition:

- As a user, I want to find links easily to different important places that I might use daily so that I can be more efficient and ask fewer questions.
- As a user, I want a guide on how to use these important links, so that I do not get confused on how to use them.
- As a user, I want information/links to be in a centralized location where it will not be moved allowing for an easier time to remember where things go.

Search Bar:

- As a user, I want to easily find KB articles so that I can assist clients more easily.
- As a user, I want to have a search bar so that I can easily find useful information.
- As a user, I want to be able to ask questions and get recommended KB articles so that I can get help without asking someone else first.

Log in Feature and Account:

- As a user, I want to easily be able to log in, so that I do not have to waste time.
- As a user, I want the system to save information I put on the site so that I can see it when I come back to the site.
- A user, I want to be able to customize a profile so that others can see who I am.
- As a user, I do not want my personal information visible on the website

because it is unnecessary.

- As a user, I want an easy alternate way to check in and out at the beginning and end of my shift so that I can save time not having to load teams.
- As a user, I want to see who is currently working so that I know who to ask for help.

Quizzes (higher-up position only: staff, team lead):

- As a TAC staff member, I want to make quizzes, so that we can train workers.
- As a TAC staff member, I want to be able to assign quizzes to users, so that we can train workers.
- As a TAC staff member, I want to be able to grade quizzes, so that I can give constructive feedback.

Alerts and Announcements:

- As a user, I want to quickly be able to get into contact with another user so that productivity is increased.
- As a user, I want to see important information so that I can stay updated on what is happening at TAC.

Security:

- As a TAC staff member, I want to be able to regulate what student workers can access so that they can only do what they should (least privilege).
- As a TAC staff member, I want the site to be private and only accessible if permitted to view the site so even if someone gets a link, they cannot access it.

Staff Related:

- As a TAC staff member, I want to see who is online on the website so that I

can know who is working.

- As a TAC staff member, I want to be able to remove and/or make new announcements so that I can keep students up to date.
- As a TAC staff member, I want to have the help channels that have a lot of questions previously answered accessible in the site, so that we do not have to constantly answer the same questions.
- As a TAC staff member, I want to be able to access Admin sites in one place, so I do not need to go to multiple places.

Extra:

- As a user, I want to easily look up UNCW accounts on the website so that I do not have to use a script.
- As a user, I want to quickly send staff on duty DUO bypass codes and temporary password requests so that I do not have to use MS Teams every time.

Important Attributes to Keep in Mind

This section will discuss attributes that the staff and manager at TAC all agreed were needed and important to keep in mind while working on the project. Each one has its own place and is not an individual feature but more of a code to follow when thinking about features and if each of the attributes is being properly included. This section will discuss why they are needed and important to the success of the project.

Searchability. Searchability is the top priority of the website. This is because if we cannot find anything in a short amount of time it will not be used. After all, that is the current problem with how things are stored. Being able to access the correct information or service quickly will improve the user experience and improve efficiency. Getting

correct knowledge base articles, frequently asked questions, etc. is a core requirement. This way of thinking can apply to everything on the site/app. It will be more beneficial in the long term to make everything searchable when it would make sense to do so.

Maintainable. An easily maintainable website and apps are required to keep the project continuing after I leave TAC and college. It is important to have a plan for software after it is deployed because it requires maintenance and updates when needed by the system. This is a basic practice taught in the software development life cycle to have a plan for monitoring or upkeep. Things like the apps would require documentation that shows how to update certain features since it can get quite complex. Since we are using SharePoint as the main host for the apps, it is already known by the people I work with and has a good user interface.

There will have to be some level of trust that I put in the future staff that work on the project in later phases. This website/app is going to be the first iteration focusing more on core features and adding more afterward if time allows it. The level of the current staff is well suited enough to make changes. This is why what technology we use will be beneficial in the future for sustainability. In no way do I plan to forgo assisting with the future of the project since it is for a place that has given me a lot of good memories and deserves improvement from my personal experience. It is good practice to make sure the system stays intact and is checked on as part of the software development life cycle. I plan to stay connected afterward and help when it is applicable for me to do so.

Permission-based. A permission-based website is another priority that is easily fulfilled by using a Microsoft product because I already have the accounts, it just requires assigning the correct roles and making it more easily manageable by staff. Other

alternatives would require a system to save and remember who is who which adds a lot more complexity over SharePoint. It is important to use resources that are already available instead of trying to recreate the wheel. This is important to the system because we need to allow staff to have advanced permissions compared to workers who should just have viewing permissions.

Staff will have a lot of functionality on the web/app since they need to be able to post things like announcements, edit links, add new information, make changes when needed, change permissions of users, assign tasks, and more. Most of the other features will help staff and workers to function more efficiently with systems that will improve performance. Users will need to use the web/app to view and access information and to have more efficient means of doing their daily tasks with a better user interface. The functionality will improve over time with later upgrades when needed.

Automation. Automation is taking over the world in every industry. It has had a mixed perception by people who are worried if it will take their jobs. Some repetitive jobs are being taken but it can lead to jobs being opened for less monotone activities and have more meaning. Either way, it is the future and I plan to use automation for some of the features in the program like announcements and making things semi-automated or just more streamlined. It is clear why we want to adopt automation into the project.

Security. Security is of the utmost importance to this project. Making sure that users of the website/app do not have their personal information shown is being thought about when designing anything. The staff of the website should have access to certain areas that workers should be restricted from. Workers will only have access to review their quizzes that have been taken or surveys and tasks that have been completed. We want to implement the least-privilege method of cybersecurity which means users only

get permissions for what they need to use. Upper staff should have access to see who can view and edit certain sections of the site. Workers should not be able to edit any of the site information, just access and use the buttons given by the application and website.

Cross-Platform Accessibility. We wanted a solution that is not only available on laptops/desktops but is also accessible through mobile. Cross-platform integration has received widespread success because more people can get quick access to what they need. The main functionality of the site/app will be more efficient through a laptop or desktop computer but having more options is important to keep in mind. Making sure that the app formatting is correct across all versions is always a challenge. Without fine-tuning one version of the app could be completely unusable and that leaves a bad image for that user in the future.

The mobile version should focus on readability and having functionality where it makes sense. Since the app is trying to be as simplistic and straightforward as possible, there should not be clutter on a small screen. This is an important design issue that must be addressed early with multiple approaches ready in case one does not work. Many businesses will treat non desktop users as an afterthought and depending on the situation it may not be important enough. In our case, it has enough precedence to be considered more than an afterthought. For the SharePoint site, there should be a lot fewer issues because they already convert well from one platform to another. This is an added benefit of using software from a well-established company that is one of the best in its industry.

Planned Features

In the following section, I will talk about features that are planned but we are not sure if there is enough time to work on in the spring. They are features with great implications if they are added but must be held until core features are implemented. This

will be a list of features that will be continually added to in the future.

Duo/Password Reset Requests. An interesting feature that is on the back burner is a way to easily send DUO and password reset requests to staff through the website. It could be as easy as adding a box for a name and a heck box saying they were ID verified; only allowing it to be sent if the box is checked. The current method is using teams and the help channel to tell the staff a username and that they are ID-verified. This would be a great feature because the current method used is to ask through teams. A better way would to be able to just put in a username and check a box saying that the ID has been verified and it will be sent off to a staff member. The staff member will then be able to send back a code, this works for both temporary passwords and duo bypass codes.

Chat Bot Feature. One feature that I wanted to implement is automated training through something like a chatbot. I have been experimenting with different types. They can learn new inputs if a question is asked that has not been asked before by letting the user input an answer. With this method, it would be difficult since each question must be framed precisely and is not like ChatGPT since ChatGPT has been trained on trillions of words. While the chatbot I have made only knows around 100 words.

ChatGPT has an API that can be integrated into something like Python. I can make a scenario and teach it based on a lengthy message that will allow ChatGPT to roleplay as a client who does not know how to do anything. This will allow the person being trained to work through scenarios. It would need to go through a lot of testing to make sure what the client does is valid. It should not give false information which could be a major problem. AI poisoning can be a real issue in a business where data cannot be tampered with. I do not see it being affected by this but having incorrect information will be a major problem if not done correctly.

The issue with using something like ChatGPT is that it will charge you for every input and output. This is not sustainable if trying to stay on a budget, it can easily go over the budget if abused. It would be a feature that would need maintenance if it were added as a main core feature. This will only be used for training so it should not go out to the wrong people. This means it is a feasible idea. ChatGPT has shown a lot of improvement with version 4.0 and will continue to improve.

TeamDynamix Integration. Our ticket system is done through something called "TeamDynamix". We use this to create, edit, and resolve tickets for every client or issue we have. This would be a beneficial feature because, for example, we could pull "parent" tickets which are bigger issues that span over multiple clients so a worker would directly see it. This falls under announcements but integration with the website/app would be the wanted feature. Currently, MS Teams has integration with TeamDynamix so it may be possible through Teams with my current knowledge. I see this as another way to centralize the information more in one place. I cannot currently surmise if it is possible to make tickets through the website/app. It would require further research and testing if time allows me to work on this feature.

Quiz making – Canvas. The current system to give out quizzes and tests at TAC is through Canvas. It is very familiar to most of the workers since the same program is used for courses that all students take. Making quizzes and giving them out to workers is simple and has already been established. Switching over to something else that has less quality and that would require a lot of work might not be a good use of time. This would be a feature towards the end of the project if other core features have been correctly implemented.

CHAPTER 4: DISCUSSION OF AVAILABLE TECHNOLOGIES

This section will discuss the current available technologies and the pros and cons of each as well as which solutions were chosen after a discussion of the alternatives. Each section for Front End, Back End, and Automation will have a table to highlight each option. Many of the options were close to becoming the option but something or another caused the choice to be certain.

Available Options for Main User Interface

Google Workspace. This product is a great option since Google and Microsoft are very big competitors who both want to take as many customers from the other as possible. This would cause a high level of competition between the two companies. Competition always benefits customers the most, so we want competition since having a monopoly would allow the company to choose whatever prices they want. Google Workspace has a lot of similar functionality that other big companies offer but with integration with Google products. Unfortunately, we need a product that integrates well with Microsoft products (Google Workspace, 2023).

Confluence. Confluence is run by Atlassian and is used for many team projects to improve workflow. Confluence's primary focus is on documentation, knowledge sharing, and real-time collaboration. Both platforms can be used if their strengths are leveraged properly. The project for TAC requires a different set of tools though. Trying to improve the current systems with automation and lowering the overall number of pages needing to be traveled to get valuable information is the highest priority.

Confluence integrates well with its set of tools that Atlassian makes, but like Google Workspace, we need a tool that works well with Microsoft, not Atlassian. They all have a huge audience backing them and will continue to grow over time in their

respective needs and uses. Confluence does collaborate better than most, but it would require everyone to make an account, which can be a hassle (Atlassian, 2023).

Dropbox Business. Dropbox has a lot of good features like a user-friendly interface and intuitive management, making it easy for teams to adopt and use. It does not offer tools like back-end and automation which are elevating the TAC project to where it needs to be. With testing, having something that does not cost the school extra money is very appealing and what we need for this project. The project would have a lot higher chance of being approved even if it is not perfect since it is free to work in and should stay supported for a long time (Secure team collaboration - Dropbox Business, 2023).

SharePoint. SharePoint is familiar and user-friendly enough to be maintained by someone else afterward and is modular which makes things simplistic. Trying to make a full-stack website with all the features could easily cause problems elsewhere like compatibility with the school and connecting it to MS Teams. The security at UNCW would cause many roadblocks. These are avoidable by using something that is already verified on the system like SharePoint, and it is free to use with the college.

SharePoint is responsive and allows for mobile use. This is already made available out-of-the-box and does not require any tweaking. Other alternatives do offer the same feature, but it is a very handy feature and is with Microsoft. It can help solidify not having to use teams to sign in. Having a button that can be quickly pushed to sign in over manually going to a channel, then having to open the message UI and type “Check In” can be quite tedious to some over a simple button push.

SharePoint has a lot of great templates to look through that offer great insight into what the product offers. This can easily give inspiration to ideas and allow, at a glance, to see if certain features are possible. There is also the possibility of making graphs that help

track work being done by workers and staff. Microsoft has many powerful tools for this. You can specifically get a job working on SharePoint sites and because of this, there are plenty of useful resources and guides (Narayn, 2023).

TAC wants to use the SharePoint site to host information as well, which is one of the more basic features. Allowing for certain access by staff only and everything else for workers. There are a few permission levels including Full Control, Design, Edit, Contribute, and Read (Narayn, 2023). The manager would get access to Full Control while giving the staff edit or contribute roles and workers read-only. This is to follow the principle of the least privilege as a safety precaution. We have a general trust for each other at TAC but must also be cautious. Another security group is Site visitors, Site members, and Site owners (Narayn, 2023). We must make sure we give appropriate access to each member. This will be a basic place to get information, but we want to store most of the quick-access information on the app, when possible, to keep everything in a centralized location. The app will go onto the SharePoint site but currently, it is recommended to open the site in a full tab.

Since SharePoint is a Microsoft product, we are limited to the licensing that we have from the college. There may be some limitations involved with this. From what I can tell, that is not affecting any of our work now. The only thing I have noticed is that some of the connectors do not work when trying to add small panels to pull information from other places outside of SharePoint. This is fixed with another solution we are going to incorporate later. A summary is found in Table 1.

Furthermore, there is a lot more complexity to the app development process with OutSystems. The pricing for the full service is very expensive. The free version has limited features. The complexity is unneeded, and a much simpler development process

would be preferred for this project. The TAC project will not require something so demanding (High-performance low-code for APP Development, 2022).

Table 1.

Available Options for Main User Interface

Alternatives	Pros	Cons	Notes
SharePoint	Familiar and user-friendly to staff and student workers. No sign-up is needed since it is connected to a Microsoft account. Already connected to all the data we have in OneDrive. Connected to MS Teams. Reliable well-established company.	Limited to the licensing bought from the school on extra features. Limited permissions on what connectors can be added through SharePoint because of College security policies.	
Google Workspace	Highly successful businesses backing it (Google). Lots of features are also in SharePoint.	Does not integrate well with Microsoft products, which is what we want.	It is better not to interlace two different competitors' products when working on a system with multiple products. Choose one or the other.
Confluence	Good for improving workflow. Good documentation, knowledge sharing, and real-time collaboration. Great integration with Atlassian products. Lots of backing and will continue to grow over time.	Does not have the tools needed for the project we have in mind. Not as user-friendly and would require everyone to make an account which can be a hassle.	
Dropbox Business	User-friendly interface and intuitive management, making it easy for teams to adopt and use.	Limited tools	

Google AppSheet. AppSheet is a strong choice for organizations heavily invested

in the Google ecosystem. It excels in cross-platform compatibility, easy data source integration, and AI capabilities. It is suitable for straightforward to moderately complex application development. It suffers from the same issue as Google Workspace where the TAC project is heavily reliant on Microsoft integration (Google AppSheet, 2023).

Power Apps. Power apps are the main way to make most of the functions work. There is also Power Automate but I will discuss that later. Power apps allow me to make a website-like app that I can do things like push a button and in combination with Power Automate send a message in a team's channel saying anything, but for example, “Check In”. This would be used once when opening the app originally or optionally skipped if signed in already. So, it cannot be spammed. Making it lead you to another page when you first enter the app means, it cannot be spammed preventing someone from thinking they did not click it.

Having a similar exit feature to leave the program that redirects would prevent spam in a team's channel which would look bad otherwise since it is supposed to be a professional environment. It is also important to consider how someone would access this sign-in system. It would currently seem slower to sign in with this method. Since some people come in right at the time, they are supposed to be there they may just use the Teams app instead. They should not be forced to have to sign in every time they need to access Teams. It needs to be a flexible system as an alternative to typing “Check In” through Teams.

Many companies use a sign-in or a company card to sign in. The way we sign in is a lot more manual and saves money on making a system but can be tedious and many people will forget to sign in and out completely. It would improve the overall number of check-ins and check-outs for some people who forget. For the people who do not forget

and do not like using Teams, it will help improve performance as well.

This is just one of the many features planned to be in the core version of the app. Power apps are quite versatile and allow for a lot of customizability in the right hands. A solid foundation is important before working on a project to know the capabilities early to prevent later setbacks. Even though I have prior experience in many programming languages, I do feel that Power Apps is easy to learn but would be hard to master with all the small tricks that are possible. This is the perfect tool because it is highly supported by Microsoft and will become even better over time. It is quite easy to add things like tables to store information and then update it with current information when needed. Being able to hold data and edit as needed will be useful. I can see storing data to be useful for things like user feedback, KB article suggestions, quizzes, etc. A summary is found in Available Options for Automation

Zapier. Zapier is versatile for connecting a wide range of non-Microsoft apps, making it a strong choice for users who want to automate tasks and workflows across various platforms. It is particularly user-friendly and accessible to users with minimal technical expertise. Organizations can use both Power Automate and Zapier in tandem to cover a large variety of automation needs. Zapier is great for non-Microsoft app integrations. If a case does come up for a non-Microsoft related automation this may come in handy. It is exceptionally user-friendly, which is a feature we want (Zapier, 2023).

Make. Make offers a very large amount of app integrations that go from popular to niche. It excels at multi-step automation scenarios with conditional logic, data transformation, and error handling. It has a user-friendly visual builder using a drag-and-drop interface allowing for no coding experience. The pricing for Make is good but lacks

the integration of other products. Make's large variety of templates shows it a very tempting option (Make, 2023).

Table *2*.

Available Options for Automation

Zapier. Zapier is versatile for connecting a wide range of non-Microsoft apps, making it a strong choice for users who want to automate tasks and workflows across various platforms. It is particularly user-friendly and accessible to users with minimal technical expertise. Organizations can use both Power Automate and Zapier in tandem to cover a large variety of automation needs. Zapier is great for non-Microsoft app integrations. If a case does come up for a non-Microsoft related automation this may come in handy. It is exceptionally user-friendly, which is a feature we want (Zapier, 2023).

Make. Make offers a very large amount of app integrations that go from popular to niche. It excels at multi-step automation scenarios with conditional logic, data transformation, and error handling. It has a user-friendly visual builder using a drag-and-drop interface allowing for no coding experience. The pricing for Make is good but lacks the integration of other products. Make's large variety of templates shows it a very tempting option (Make, 2023).

Table 2.

Available Options for App Integration

Alternatives	Pros	Cons	Notes
Power Apps	Very versatile and user-friendly. Plenty of guides on how to use Power Apps because it is a Microsoft product. Integrates very well with	Some features are limited by the license that is bought from the college. It has its completely own way of doing things that require upfront effort to figure out.	

	SharePoint and other Microsoft products that may be used. Many tutorials online on how to use the product.		
OutSystems	More extensive development capabilities. Geared towards bigger organizations for big projects	Required to download an app maker to edit the software. The user must create an account. No web version of OutSystems Very expensive. More complex, which is not what we want.	Requires too many variables to be effective at TAC. Not simplistic enough, which is one of the basic requirements.
Google AppSheet	Part of the Google ecosystem. Excels in cross-platform compatibility.	It is not a Microsoft product Basic and not as many features as other products that are similar.	

Nintex. Nintex has a high level of workflow automation capabilities. It provides a user-friendly interface for creating and managing complex workflows that automate business processes. It seamlessly integrates with things like SharePoint and has low-code or no-code options for workflows, forms, and applications. This allows users of varying levels to use Nintex. It offers things like data extraction, document generation, and more. These are all nice features but come at a high price tag that would never get approved since this is a capstone project, not a business (Nintex UK Ltd., 2023).

Power Automate. Power Automate is another powerful piece of software that can be used on its own or with things like Power Apps. It can use webhooks to communicate between two software. Automate allows users to create their own “flows” meaning it can do a simple action that can have a wide variety of implications. For example, a user could make a flow that allows them to send certain emails directly to a Teams channel. This

serves a valuable purpose in a company that gets email announcements and then retypes the entire message on Teams again. Power Apps can easily implement this type of feature. Allowing for plenty of automation, which is what we want to do for this application.

Power Automate allows for Python scripts to be run which allows for a lot of extra room for an improved experience for the user and especially for the developer if they have used Python before. Python is a powerful language that UNCW has switched to being the main language taught to undergraduates in the Computer Science program. This allows me to take advantage of the ability to write scripts.

With Power Automate, there is the ability to completely start from scratch on building automation or to use a template for popular automation. The complexity of the automation system is very large and can require some training and guidance when first making an automation. The versatility allows for a large assortment of options under the Microsoft family of products. It can allow for a lot more notifications going directly to staff and to keep the lives of workers in one place. It may simplify things, but it should be noted that the manual way be taught as well just in case of system failure of the app. Power Automate can pull information and display it while keeping all that in the background just to see the neatly displayed parts. When looking at making a flow manually in Power Automate, it looks quite confusing and complicated but with some practice it makes sense. The coding part of Power Automate is also very straightforward and has many guides on its uses (Narayn, 2023). A summary is found in Solutions Recap

While looking at options for the main user interface, many of them were appealing. The one we are going with is SharePoint because of the familiarity and ease of access and use. Many of the staff do not want to learn a new main user interface and

would prefer to stick to something like SharePoint. It also seamlessly connects with things like OneDrive to store documents.

There are a few features that require Microsoft's capability. SharePoint was the best option over other alternatives. Power Apps is one of the main reasons that SharePoint was chosen. Users can use power apps separately from SharePoint or through SharePoint. Combined with Power Automate we can automate tasks easily without having to do much extra work allowing for time on more demanding features. With UNCW having a Microsoft 365 License we get access to a lot of things that would otherwise need to be paid. This is a huge step over all the other options with this fact alone.

With the school's security policies, Power Apps is one of the only products we can use with SharePoint that is not going to require asking for permissions. If we asked for permissions for things that are restricted we would get denied since it is a project by an individual graduate student and not a team of paid workers. It has a high amount of customizability and its front-end as well and works like an app. It is meant to work on phones and on computers with some tweaking. It shows a lot of promise and connects directly to our school's Microsoft accounts, which helps with a log-in system.

Table *3*.

Solutions Recap

While looking at options for the main user interface, many of them were appealing. The one we are going with is SharePoint because of the familiarity and ease of access and use. Many of the staff do not want to learn a new main user interface and would prefer to stick to something like SharePoint. It also seamlessly connects with things like OneDrive to store documents.

There are a few features that require Microsoft's capability. SharePoint was the best option over other alternatives. Power Apps is one of the main reasons that SharePoint was chosen. Users can use power apps separately from SharePoint or through SharePoint. Combined with Power Automate we can automate tasks easily without having to do much extra work allowing for time on more demanding features. With UNCW having a Microsoft 365 License we get access to a lot of things that would otherwise need to be paid. This is a huge step over all the other options with this fact alone.

With the school's security policies, Power Apps is one of the only products we can use with SharePoint that is not going to require asking for permissions. If we asked for permissions for things that are restricted we would get denied since it is a project by an individual graduate student and not a team of paid workers. It has a high amount of customizability and its front-end as well and works like an app. It is meant to work on phones and on computers with some tweaking. It shows a lot of promise and connects directly to our school's Microsoft accounts, which helps with a log-in system.

Table 3.

Available Options for Automation

Alternatives	Pros	Cons	Notes
Power Automate	Powerful software that is backed by Microsoft and will continue to get support. Connects to things like Power Apps seamlessly saving plenty of time on the backend work. Compared to other alternatives, it has similar features but is free with the license the college has.	The system used for manually making automation flows is complicated to work in. Limited to mainly Microsoft only products, would need to reach out to another product if wanting to get information from a non-Microsoft product.	The cons can be fixed if thorough research is done on how to use the manual automation maker piece of Automate.
Zapier	Versatile for connecting a wide range	Only need to connect to	

	of non-Microsoft apps, making it a strong choice for users who want to automate tasks and workflows. User-friendly with minimal technical expertise required.	Microsoft products, could be useful outside of Microsoft products for the future.	
Make	Many app integrations vary from popular to niche. Excels in multi-step automation with conditional logic, data transformation, and error handling. User-friendly visual builder using a drag and drop interface allowing for a no code experience. Has a lot of templates.	Not a great option for integrating with Microsoft Costs money to use.	This service has a lot of promise, and I could see myself using it in other projects in the future.
Nintex	High level of workflow automation capabilities. User-friendly interface for complex workflows. Integrates seamlessly with certain Microsoft products. Low-code or no-code options for workflows, forms, and applications.	Expensive price tag for large businesses only.	This service is much larger scale than this project will ever be. I would recommend this to a business looking for a well-funded project.

As for the automation side, we chose Power Automate for its integration with the other two solutions and simplicity of use with a lot to learn for improvements. Power Automate has all the features we need and does not feel bloated with features. It connects directly with Power Apps without having to import anything and allows the ability to automate tasks for almost all Microsoft products.

Limitations of SharePoint

SharePoint is limited to what form factor the website can take. It is standardized and will always have that SharePoint look to it. Through Power Apps, it is possible to customize that website to a certain liking. You cannot full-screen an app in SharePoint, you would have to go to the app directly for it to be full-screen. What features you see on

things like SharePoint, Power Apps, and Power Automate is what you get, and learning more complex features can be tricky.

The college blocks the use of connectors through SharePoint. A workaround is using some of those processes through Power Apps instead of SharePoint. The limit of what I have used so far has been promising but they are on Power Apps, and I cannot directly connect them to SharePoint which is unfortunate. With security rules for the college, I highly doubt I will get any leeway from them, so finding good workarounds will be important.

Limitations of Power Apps

Some limitations of Power Apps are that trying to make the app look good in both the web version and the phone version could take a good amount of work to get correct. Both versions are important for us to have but we want to primarily focus on the web version since that is where the main amount of use will be. Power Apps can be limited on certain tools like programming languages and have to be done in another program that has to be learned which takes more time away from the project. Being able to take information is very reliant on what the college will allow us to do in terms of permissions so workarounds will be important.

Limitations of Power Automate

Power Automate has a good variety of automation that are premade and connectors to use for different functionality but is still limited on where it can pull information from. It will work fine for this project, but I can see that it is limited if needing to be used on a project outside of Microsoft. The limitations of Power Automate are not very apparent currently and it has what we need for the project currently.

CHAPTER 5: PROPOSED PLAN AND TIMELINE

The Gantt chart (Figure 1) will show what my plan is for Spring 2024 on completing each part. Below the chart is what I will be doing in each section in more detail. It will help me to stay on track to make sure I have ample time to complete the project and have extra time to polish and add extra planned features to the project.

Figure 1. Project Workflow Schedule Gantt Chart

1. SharePoint Setup
    - Start on the main SharePoint page and work on getting the correct information uploaded and visible to only staff while other information is to student workers
    - Have links and whatnot available on SharePoint directly as well as the app later
2. SharePoint Completion

- Testing with staff to see if they can use SharePoint to its fullest or if it is missing anything needed
- Make sure there is no bloat features

3. Power Apps Page Setup (Front Pages)
    - Create a basic layout of all the core features so that they can just have data input into them for functionality:
    - Front page (announcements, tasks assigned, reminders, etc), Tasks page/onboarding, Profile page, Useful links page, Search page for KB articles, Requests page to communicate with staff on different things, Staff only page (task automation and permission-based things), TeamDynamix Page (WIP, still figuring if this is completely possible), Chatbot page (WIP), Add more after these are done, Work on sign-in feature for app through Power Apps and Power Automate
4. App Pages Formatting (cross-platform)
    - Adjust the appearance of the Power Apps so it functions well on different devices
5. Announcements/Tasks/Reminders Page (main page)
    - Create a spot for messages to be seen by everyone who accesses the app. Ability to scroll to see old messages as well
    - Create tasks panel if anything is assigned there by management.
    - Create a panel that shows current issues and is removable by management
6. Profile Page
    - Create a page for users to add a bio and get in contact with other members. It will show who the user works under, their position, a profile picture, and

no personal information

- Ability to add description
- Should auto assign manager to Courtney (TAC manager)
- Should show online status if checked in, should turn off when TAC is not open.
- Have other people be able to view the profile (contact info, description, pfp)

7. Useful Links Page
    - Create a page that holds a majority of the links to websites that we may need to visit
    - Add a tutorial for the sites that need them / not very intuitive

Figure 2. Project Workflow Schedule Gantt Chart

8. Search Bar Page
    - Create a search bar for websites we use daily (Knowledgebase, ticket search, etc.) that will show options before hitting enter in the search
9. Staff / Higher Tier Worker Only Page
    - Create a page for staff to receive messages from users for requests and the ability to give back codes through the page
    - Any other features the staff would like
10. Student / Staff Requests Page
    - Create a page for users to request things like Password reset requests (with checkmark to confirm ID check), DUO bypass code requests (with a checkmark to confirm ID check)
11. TeamDynamics Page

- If possible, connect TeamDynamix tickets to the website to show ticket information

12. Chat Bot Page
    - Create a chatbot that can answer basic TAC troubleshooting questions after being trained
    - Used to train new workers to prevent staff/team leaders from needing to train manually for basic things since it takes hours. They would still be around to answer questions. Using a chatbot can be risky since it may not be 100% right. There would need to be rigorous testing.
13. Project Testing
    - Take time after all the core features are done to go over everything and make sure it works as intended.
    - Bring in users to test the system and see if they find any issues. Do a small release to only a select group of users then expand
14. Project Polish
    - Change things and add features if requested and are possible in a reasonable amount of time.
    - Make sure the project meets all the requirements set by the proposal and staff.

CHAPTER 6: RESULTS AND IMPLEMENTATION

As work on the project is coming to an end, there were many challenges along the way. Many skills are required to work on a capstone that can be called completed. While working on the Web-based solution at TAC, I have followed through with most of the tasks I sought to complete, finding creative solutions for some. This section will discuss how the implementation went on the various aspects of my application and if I see them as satisfactory or needing future work. As this was a broad project with many working parts, there were bound to be sections that had to be less polished, so I had to cut myself short on some wanted features to keep this project in scope but my workers at TAC and I are happy with the results. As per the Gannt Chart, I was able to stay at least plus or minus a week on task depending on the perceived difficulty being different from the actual difficulty.

First, I want to discuss the relationship between the core issues listed in prior sections and how I handled implementing them or the problems faced limiting their addition to the application and the fundamental attributes. I focused on adding parts to the project that felt like it was not trying to be MS Teams 2.0 and more of a complementary service since I cannot compete with Microsoft, and it was also just meant to streamline the daily activities at TAC. Without the help from individuals at work (Courtney Lee, Ixchel Peralta-Martinez, Eric Guetta, and Raymond Khamthong) for assisting with testing and seeing things I did not notice the project would not have gone as smoothly, so thank you.

Resolutions to the Problems

Repetitive Questions. With the TAC knowledgebase chatbot that was added to the app, the members of TAC who have tried it say it will be useful in reducing questions

asked from human to human when it is not necessary. It is trained to answer questions and can be expanded upon in the form of adding specific prompts for tricky questions or using documents/websites with a large amount of essential information. The IT department is also working on their own Copilot chatbot so hopefully, it can be added to the TAC app in the future but there are no guarantees.

The useful links page/document view page is another great place to prevent users from asking repetitive questions and hopefully train themselves on where to find information without help through the simple user interface.

Web/Information Sharing. The way information will be shared on the app will be through the Useful Links page, which will have categorized links. This will help with organizing where information is. It is also able to have more links added to it in the future under each category allowing users to scroll to see all the links if more are added. Documents are another way to share information, and staff can add that to the app when more important documents are needed. The website works by putting the rows into buckets (0-5) and displaying them in multiple different columns like “General”, “Educational”, “Work-Related”, etc.

Some methods tried to pull information from Teams to show that Power Apps worked. The issue was it would only show around 30 results and there was no way to go further back. So, the idea of taking information from the TAC teams' channel was scraped but through things like the chatbot and document/link sharing there will be a reduction in the number of questions asked.

Onboarding. Onboarding for this project was not directly worked on, but some of the systems in the project can alleviate the onboarding process. Through things like tasks which can be assigned individually or to multiple people. For future work, it can present

tasks to future TAClings and they will have a place to look. We decided to keep Canvas as the home for onboarding. This is because it is professionally made and is used by UNCW. Since we only use it, there will not be much of an issue with redirecting a link to onboarding.

Announcements. The announcements task is at the forefront of the app and will be one of the first pages you will see when entering the site. It is integrated using a SharePoint list to hold the information that is secret and not disclosed to members of TAC. It displays information through a built-in gallery object that allows data to be viewed in a list style allowing for easy viewing. It runs on a timer that will reset every few seconds and will display new announcements. We want to keep these announcements used only for important matters.

One of its features of it is the ability to set a timer that will automatically remove the announcement on a set day to prevent clutter. There is also a manual option for admin users to have access to with a confirmation button to prevent misclicks. Using a rich text editor that is built into Power Apps I was able to make the announcements look exactly like the announcements in Teams allowing for a lot of customization in the text. One of the issues that I noted when making the project relating to the announcements is making sure that I find an effective way to display the announcements. Since the screen size is limited, it was important to find a good aspect ratio for the announcements piece. Because of the extra space I had after making the announcements I decided to also include a tasks board because that was one of the extra features that my manager was interested in.

Both the announcement and task board have ways to create, edit, and remove any task if you have permission to go to the creation page for either. There is also a remove

from task board button that allows users assigned the tasks to take their name off the list. This was tricky to implement in Power Apps, but it shows a list of names and removes those from the list who have checked off the task. They both run on timers that will refresh the pages. The task has a function that will make it turn red when the due date for the task is approaching and another when it is past the due date. There is an email icon button that only certain admins can push to send an email of the announcement to get a wider audience for people who do not check the app as an option. It will be disabled after the email is sent.

Repetitive Tasks. In the next section, there will be more discussion about the Duo/Password reset requests, but they are something that will help automate and use fewer clicks or time wasted moving around. A lot of the processes that go behind making tasks and announcements are quite tedious but making it less demanding and quicker to get done will be a step in the right direction. Especially with tasks. Tasks are not used currently at TAC and that is something that my supervisor wants us to be using more in the future. The task panel is streamlined and allows for users to see it quickly without going out of their way. On the staff side, creating anything is on one page and they can remove or edit on that same page. Staff also have the option to remove things like announcements or email the announcement out to users who are using the app saving time by sending an email and posting the message on the announcements separately.

Features of the Solution

In this section, I will try to avoid repetition by repeating what will be said in other sections. One of the sections mentioned elsewhere was the Announcements/Task page. This section will cover direct features and their pages in the capstone TAC app instead of how they correspond to prior sections of my paper to prevent confusion. Each page has

information that has an editor for admins to change. They are quite repetitive to talk about so I will not discuss them much. They will have IDs that make each unique as well as marking which users have created or edited a row of data.

Home Page. The home page is the landing page for the app. It hosts things like a box that shows which users are checked in and which are not indicated by a green or gray circle. There is also the option to be away which will show a yellow circle. If someone forgets to sign out, a Power Automate script will run at midnight EST and turn all users offline. Other features of the home page is the buttons that take you to other parts of the app. These include: “Profile”, “Guild”, “Chatbot”, “Requests”, “Announcements/Tasks”, “Useful Links/Documents”, and a “Sign In”, “Away”, and “Sign Out” buttons. There is also access to an admin page that is only available if the user has at least “Basic” level admin privileges. The last feature of the home page is the search bar which allows users to search ticket numbers with our internal ticketing system or search keywords that will be used when redirecting you to the Knowledgebase website.

On the admin page, there is a list that shows whose admin and what permission level they are. The only admins that can see this are the highest-level admins, “Full Access”. They can add, remove, or change a user's admin level. One user must always be a “Full Access” admin. Also, the ability to change a user's guild is here, and the ability to make a chat in MS Teams with any group of users.

Duo/Password Reset Requests. The password request page had a lot of trial and error to get it working properly with the limited options in power apps. The requests page has a username input text box with a checkbox to say you’ve verified the ID of the client. Once that is done the user will see an area near the bottom of the page that has the request they created, and it will show who worked on the request and the code once the request is

completed. This is the same for password reset requests and DUO bypass codes. Once the user deletes the request it will be archived so that admins can look at all requests that have been made.

On the admin side of requests, there is a sound that plays for them on any page if they are on the app and a button that takes them to the request page. request and DUO bypass codes are in the same table they are redirected to only one page. They then can set themselves as responsible by clicking a button on the request, so multiple people aren't working on the same request. After this is done and the request is completed, the user who made the request will hear a sound that tells them their request is done. Each request is set to the user who sent it so only they will hear a notification and not everyone else on the app.

The way to implement this was that I had to use a gallery to show the whole data table. I had to rely heavily on the filter function so that only certain users could see the request based on who made it. We do not have many requests at once, but I made it this way because it is scalable and can handle a lot of traffic. There is a template in Power Apps I used to edit the data in the tables which will update the galleries.

Chat Bot Feature. The chatbot takes advantage of the license that UNCW has for Copilot Studio which gives access to making chatbots. The knowledgebase website was used as the base and more specific prompts were added later. The chatbot is used to answer an assortment of questions. This is beneficial because a lot of niche questions pop up daily. Also, not every TACling gets the same training and sometimes the trainer forgets to mention something or only briefly mentions it. Having a chatbot that can pull information directly from the chatbot helps to stop TAClings from asking the same question to others. Using a chatbot will help some with automation. Using the chatbot

does come with testing it with a lot of different questions to see how it answers them since just giving the chatbot the link to the KB website leaves a lot to interpretation.

TAC wants to prevent errors/misinterpretations from occurring and give the uninformed TACling and client the wrong answer with varying levels of detriment. The system that Copilot Studio uses is a node that points to another node after a certain function happens which has a good amount of customization. With this feature, we can add prompts like “What are TACs hours?” and be given accurate answers. Some of the issues with this may occur with the fact that I had to manually add the correct

TeamDynamix Integration. With seemingly no way to directly connect TeamDynamix to Power Apps, the best way we can integrate is by adding a search bar for ticket search. Methods like iframes were tried since html is somewhat available in Power Apps but the ability to use them is disabled. With this in mind, my chair and I looked into the Iframes on the TeamDynamix page to see if we could find any link we could use for client search or ticket description search but there seemed to be no patterns in the URLs that would allow for this because they used sessions which cannot be generated externally through Power Apps.

Guilds. TAC guilds are a system that was made a year or so ago to help promote more teamwork between workers at TAC and to get them incentivized to improve ticket quality and to provide better in-person support. There are three guilds, and each one is run by a team lead. The rewards were things like pizza parties. We have a list of ways that workers can gain and lose points. I made a streamlined way for the staff/team leads to add and remove points from users with a UI that allows them to quickly look at individuals or a guild to see how they are doing against other guilds. The information seen by regular users is just what guild has what points but if you are an admin user you

can go into each guild and see who has points and why, how many points, and a button that redirects to a ticket. Admin users can change who is in what guild but only at the highest admin level to prevent unwanted changes from a lower-level admin user to follow the least privilege.

Profile Page. The profile page gives the app some character by making it more customizable and making a better impression on the users. It does not show any personal information and can do things like adding a description of yourself and currently adding a LinkedIn profile link to the page. In the future, it would be nice to add the ability to make a collage. There is also a chat box on the left side of the profile page. It currently lets you talk to anyone who sends a message in it and is not filtered by any group. One of the ideas behind it could be a guild chat system to talk to your guild only but it is currently unfiltered. The chat allows users to expand a message by clicking on it since the message box is quite small and only fits maybe seven or eight words. The expanded box allows for a few hundred words. To achieve the chat feature, it took a gallery that uses a unique identifier to sort the list backward so the last row in the database is shown first. With the message and name of the user showing.

The profile page shows users their UNCW profile picture, first and last name, and email address, and they have the option to add their join date. There is also the ability to directly chat with a user which sends them a direct message in Teams. In the future, it would be nice to add it to the app but for now, Teams works well. Everything is also sorted in alphabetical order so users can find names that way when looking for a certain profile. There is no comments section on a profile page, and I do not see that being added in the future.

Useful Links and Document View Page. This allows staff to add useful links under

certain categories that are premade. “General”, “Educational”, “Work-Related”, “Documents”, “Teir2+”, and “T2+ Documents” are the current six categories with the last two being reserved for tier two workers and above. This is to help declutter for tier-one workers who do not need to see that information, or it may be private and can only be viewed by higher-up workers. For documents, the app allows users to upload a PDF document through the PDF viewer object in Power Apps. It will make a button that when pushed will show a PDF that covers most of the screen that can be closed with the x button that is placed at the top right of the screen.

Inside the creator for this part of the app, there is a category, name, description, URL for links, and upload for PDFs as well. There is a field that cannot be changed that puts who created this link or document view to help prevent anything inappropriate from being uploaded. In the view of the workers, they will see a column of buttons with words that say what each link does. In the case of the documents, it will show the document name given. On the right of the button, there is an info button that will tell users a description of what the link does and how they can use the website or document given. Because of the limitations using the info button, it was not possible to add rich text to the description to add links and for bolding. This design choice was influenced by the help of my capstone chair, Dr. Matthews.

Fundamental Attributes

Searchability. Searchability was one important factor even among this list of attributes since we wanted it to be simple to use and navigate. On the main home page, there is a search bar that can be used to look up ticket numbers from TeamDynamix or search the Knowledgebase articles by keywords. This has served to be quite effective as a baseline of what we want; which is to have everything in one central place. The search

works by checking if the text input is all numeric (ticket #) or if there are any letters (KB search).

Another feature relating to searchability in the project is things like being able to look up created requests, announcements, tasks, guild points, etc. This is important when there is too much information to manually sort through it. Being able to search by a user's name or to have a user interface is much more manageable.

Maintainability. TAC and I are currently deciding who will be handling the app in the future. My TAC supervisor at work is going to the documentation on each part of the app that will go into depth on how to add/remove parts. I want to specially make it clear how to use parts that give me trouble and give others trouble. The idea while I was making the project was to make as many parts as possible work well enough that they do not need to be changed or can be changed easily. As well as making options to add and remove things in the app so that users do not have to go into the editor as much. With clear enough documentation and guidance from me now and in the future if issues occur. The goal is to shoot for maintainability by a user who is not as tech-savvy. I do not expect someone to know right away how to make certain parts move the correct way since Power Apps uses its programming language. It will be a learning process for the user and hopefully, Microsoft keeps adding some of the features Power Apps needs.

Permission-based. The TAC app's permissions are easy to solve since they are already connected to their UNCW accounts. Each user is automatically entered in with no permissions and the highest-level admin will need to give them proper permissions if they should be high level. Using this system I disabled or enabled objects based on if they have proper permissions or not. This is a simple solution that allows users to not go places they shouldn't. When they log into the app, they are always sent to the home page

as well since it's an app and will not take them to pages they should not go to. If they can make it to a page they should not, the pages show nothing since they do not have permissions.

Another thing that is noted is that there always must be at least one admin at the highest permissions because a lot of other features would be lost. I did add a button for a developer to get access in the event of this happening somehow even through the prevention method. This is because the only way to change the data is quite complicated, requiring the owner of the SharePoint site, enabling editing in the settings, making all the rows unhidden, and then you can edit the data, then disabling editing. This is the case for all the tables since this is the only free and available option in the UNCW environment, which has proven to give a lot of limitations.

Automation. The DUO bypass/password reset page helps a lot with automating the process for both requestors and staff. It keeps it more streamlined and reduces clutter. There can also be automation seen in the announcements board which will auto-remove announcements after a set date which is shown during creation.

Security. Security is a big point in any project where the CIA triad is important. I will go over how I used the triad to keep the app secure. This will give a feel for the app's security. I will not mention authentication or non-repudiation since they are subcategories of the CIA triad and will be discussed in those three.

Confidentiality

In the app, I use least privilege and deny by default practices to prevent any unauthorized access of information that users should not see. I do this by only assigning users the base level, they are not a part of the admin group unless the highest-level admin adds them (manager). This is to prevent people from seeing things they should not see.

Examples include which TAClings have what guild points and why. The TAClings should only see guilds and how many points each one has. Another example is only showing tasks to the users they are assigned to. If they are not assigned the task there is no way that the user can see it. Through the SharePoint site, we can edit the data without letting users view the data, preventing important information from leaking.

Integrity

One way integrity can be viewed in the app is when an admin views a page where things like announcements, tasks, or guild points are made. It will always show who the last person was to edit the document. This is to prevent someone from altering what it says to get someone else in trouble. Since the accounts are connected to Microsoft, it helps a ton with securing and making sure only the correct users are allowed on the app, and they are held liable for things they do since it is all documented.

Using SharePoint lists has made storing data very secure and keeps the integrity of the app by not allowing modifications from unauthorized users, keeping data accurate and reliable. When entering the app, the user will have to be signed into Microsoft already preventing another user from getting into the app. Also, the app itself is internal so it would be difficult for an outside threat actor to get access since it will be housed on MS Teams or shared through a link. The way to access the app requires being added to the power apps and the SharePoint and Teams channel having many layers of protection from outsiders getting in.

When changes are made to things like admin privileges, the “Full Access” admins will be notified of the change and who made the change to prevent it from going unnoticed. It will keep people accountable for their actions. This can be spread to other areas to help notify users about tasks and other things they may want to know about.

Availability

Since Power Apps is a Microsoft product, as long as Microsoft is running and having no issues the servers will have no issues. This is a big plus side to using Power Apps. I do not have any issues with connecting and keeping a stable connection throughout testing. Being able to redeploy the app is quick and will notify the user of an update within minutes.

Cross-Platform Accessibility. Cross-platform accessibility is a bit tricky with Power Apps. There was something called Power Pages that seemed better for cross-platform on web pages but that was not accessible to me as a student. Through Power Apps, there are many ways to try and work on cross-platform compatibility, but it still feels limited and currently should only be used in landscape, in portrait the app works but looks like it was not made for portrait mode. In Power Apps, some containers allow for the app to make objects responsive. With the attempts I have made in making it work, it has shown some success in things like the chatbot where it works well. On pages with a lot of parts that take up the whole screen in landscape, there becomes the issue that buttons and objects become too thin in portrait mode, this would require making whole new extra pages that run when the app detects a user is on a different aspect ratio. I have tried to implement this in the past, but it needs more tinkering.

CHAPTER 7: EXPECTED VS ACTUAL PLAN TIMELINE

In this section, there is a chart (Figure 2) showing the original planned Gantt chart timeline and what the actual timeline was. There is some disparity between the graphs due to difficulties or more time needing to be allocated for completion. Purple indicates where both time slots overlap. Blue shows the original timeline where they do not overlap. Finally, red shows where actual time was spent on each part. Some parts like SharePoint were worked on throughout the project because I would need to keep making new databases for storing information. Also needing to add columns to already existing SharePoint Lists (databases). Another reason for the large amount of time allocated to SharePoint is that I went through a search to find the easiest way to store data. I did not start on SharePoint until a month or so later.

Originally using Microsoft Excel documents was tried, allowing for data to be hidden and stored and read and written to. You had to share the document for the data to be accessible in the app. The only issue was that all these were not possible at the same time. You also had to share the document for the data to be accessible in the app. If I made the document hidden, it would not allow for users to write or read into it. If not hidden users to look at the data. If I made the data unable to be changed, it would not allow for writing at all. I also tried Dataverse which was unsuccessful, and I also received constant messages about it being outdated and no longer supported and to switch to something else. The other product they wanted me to use was not accessible to students with my permission level so that was a no because I could only make the DBs in my environment and not UNCW’s environment. The last option was SharePoint Lists which come with their complications. The lists must be set up in a specific way or the DBs will not work properly.

To set them up. I had to make a custom group that required specific permissions. Another issue occurred with making the List hidden since they had to be shared to be viewed and changed by users of the app. The solution was that each database has a filter that can only be removed by the owner. The filter is so obscure that there is no viable way to abide by the filter and view it. Even if they can view anything they will not be able to edit. This is another part I had to go in and change. This was done by making every column value hidden. The only issue with this is if I want to change anything I must go back in and manually make all the columns "manual" again instead of "hidden." There are also some columns like "attachments" that you can disable which are enabled by default.

The method I have for storing works well and is secure and tested. Pulling this data from Power Apps is importing the SharePoint List and making a gallery that lets the developer make a list of the variables. The second disparity is in the "Profile Page", this is due to making a few changes to avoid too much white space on the page and removing some uncertain features like a chat box for anyone to talk to any other user. It was scrapped because it did not fit the intent of the page and felt out of place. Adding things like a quick message button to MS Teams, being able to add your LinkedIn, and adding TAC join date.

The staff and higher tier page were worked on in two separate intervals because there were features that needed to be added first or looked at before adding permissions. I wanted to give everything its permission once I felt like the app was more complete. Having to remove where I have myself as the only person who can see certain buttons so I would not be spammed notifications by the testers.

The "Chat Bot Page" was worked on very early since I had already come up with

the idea in the Fall semester. This was just additional touches on the model. The IT department is working on their chatbot with the same or similar questions. Hopefully, they can add something similar, I do not think they are aware of this chatbot and found out about the Copilot Studio a few months after this project started. Someone at TAC is working with them so there is a chance this project's chatbot could be used for inspiration or modified.

The guild pages were not originally in the project scope but were a feature my supervisor wanted that would not require much additional effort. It is near the bottom of the combined Gantt chart. It is just a system to allow staff to give users points and have somewhere to keep track of those points.

During the project, I did periodical testing and added different people to the project and asked them to test new parts or just play around with it and give suggestions. Doing this allowed me to see bugs I never noticed quite often. Also leads to optimizations/polishing and future feature ideas. All around the project was mainly able to stay on course with a few extensions on certain parts that required constant improvement or updates.

I mentioned in previous sections that I wanted to make an onboarding section of the project. As far as I got for onboarding was the implementation of a tasks board for users to be assigned tasks that they can then complete so that the manager can at least see they acknowledged the task. Most of the onboarding is going to stay on Canvas – where it currently resides - because an onboarding system that would be needed could be a project on its own.

Towards the end of the project, I had a request to make a photo gallery page that is interactable. Since there was extra time I implemented it in an hour or so. It adds an

extra flavor to the overall feel of the project and especially makes the profile page feel more personal for each user. The supervisor also wanted me to keep the profiles of TAClings who are leaving / graduating as a sort of memento. This is already functioning that way, so it ends up being less work.

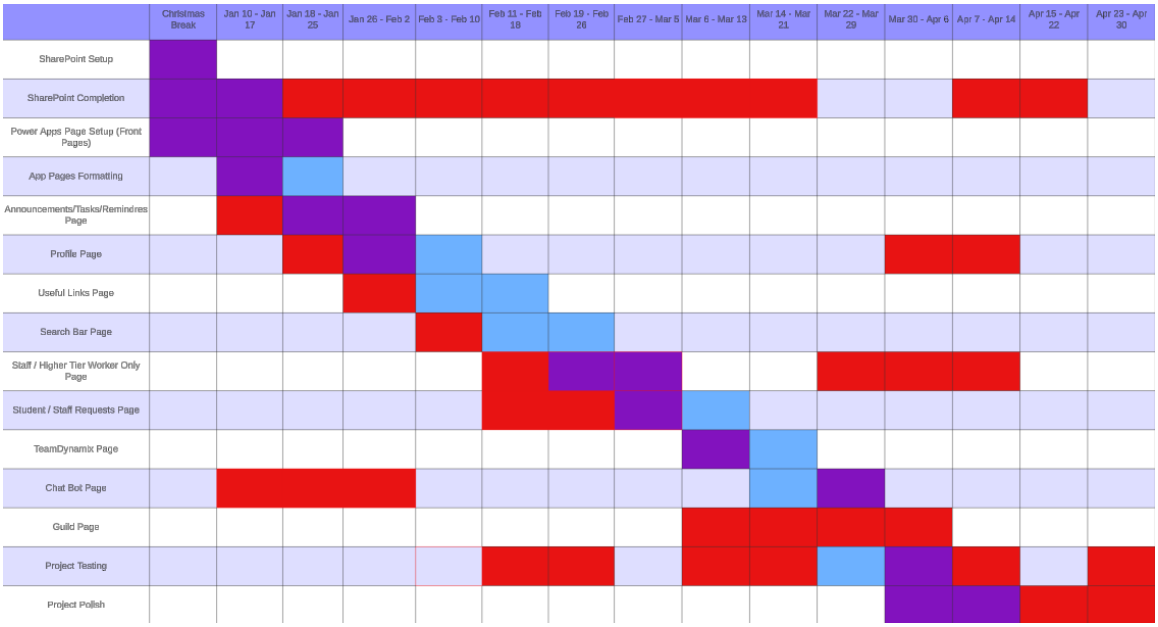


Figure 2. Project Workflow Actual vs Expected Scheduled Gantt Chart

Roadblocks and Problems that Occurred during Implementation

At the start of the project and throughout the project many parts slowed down the project's progress. In this section, these roadblocks will be discussed from the project beginning in Fall 2023 till now at the end of the Spring 2024 semester. These will include not just project app issues but any issues I encounter to keep everything documented.

At the beginning of the project, finding out what requirements were needed was a challenge, especially trying to find a time to meet with my TAC supervisor and other staff members for their input. I had the chance to meet with them two or three times, but we were not certain about what we wanted so I had to give some ideas on top of the basic concepts that they wanted. Once a month or two of requirements gathering in between

classes and working were completed, the base foundation was made.

The next roadblock was finding the proper place to house the application. With the limitations of keeping the project free (not including the school's MS license), as well as being limited on what can be used on UNCW's network. With certain aspects blocked by the administration, it made it especially challenging to find many options that would work within the project's scope. That is why Power Apps was settled on because it already is a part of the network and since we use MS products it is quite seamless with Teams. The paper was quite streamlined to make just having to look through alternatives first before making a final decision. Power Apps was also the platform that my supervisor was using to make the example app.

In testing out features and making prototypes I realized some parts of the project were not going to be as simple as they seemed originally. Finding a place to store information that works with Power Apps was quite challenging with avoiding paying fees and using what features are not blocked by admins. Many different storage services were tried. In a previous section, I mentioned all the services I tried, so I will not go into much detail here. It was an even worse problem since with permissions it was hard to test certain features while viewing the app and see information stored in Power Apps since I was the owner, and this required a second account that I could use for testing. It took some time to get someone else to help me test this but with a lot of slowdowns since I could not access their account whenever I wanted to test out something. This is one of the main reasons for it taking so long to find the correct storage service. This became less of an issue as I got further into the project and the storage was no longer a mystery.

Using a built-in Power Apps template helped a lot with the initial process of figuring out how everything worked along with the learning section of Power Apps

which gave good insight into features I did not know about prior. This saved a lot of time but led to me not knowing the optimal direction to take for each part of the project. Throughout the project I would come across some new way of doing something and would sometimes go back and change what I previously had, opting for the better approach which added extra time but, in the end, would be much better for the health of the application.

Features like galleries are the main way to show information in an organized list in Power Apps from an imported data source. Other ways are a bit trickier and require you to match a certain variable to view a specific row of information. There are many ways to get things working and luckily because of this many of the community members have found ways to improve on what Power Apps offers and they show this in YouTube videos or forum posts. I have seen plenty of features that users still want and have wanted for many years to no avail. Power Apps does receive updates semi-frequently and it seems like they want to make it functional, but it is not quite there yet. I hope in the future some of the future desired features are feasible. With the limitations set by the administrators at UNCW, I am sure some of these features are just not accessible or would cost a premium with Power Apps.

Some of the in-app issues included when features were not directly implemented into Power Apps, and I had to do a makeshift workaround and even with the support of the community, there was not much on it. The application I made using Power Apps does not seem to be a common idea since it is used more for business I've noticed. That said, I do not think it is a bad idea to try using Power Apps for this service. There are limitations but also benefits in that you have an interface that lets you create things quickly – even if they are limited. Some other features that I did not get to will be mentioned in the future

work section of the paper.

There are other small roadblocks that I came across from day-to-day work but were resolved by either coming back later with more knowledge in the future or workarounds. This is a common trend with using Power Apps; having to find workarounds. This became less of an issue as I was used to Power Apps since I knew what Power Apps could do and what it could not do much faster.

See Appendix A for images of the various pages developed during this process.

Things that Were Completed and Not Completed Checklist

In this section, I will discuss what was completed and not was not completed and the thought process behind why it was not completed. Some features were out of scope or just had better alternatives after later consideration. I looked at the user stories to reference what I planned on working on and what was accomplished.

Completed:

- Storing and making links easily accessible as well as documents with information explaining what they do in one place. Also allowing for quick searching through an assortment of links with a search bar.
- Finding Knowledgebase articles was made easier through the search bar on the home page allowing for keyword searches to articles which is an effective way with the system in place. The search bar also allows for ticket searching in one place.
- Using a chatbot to also ask questions from the Knowledgebase articles to improve on the searching by asking keywords which the chatbot will look up and give responses back with evidence to back up its answers.
- Proper login system that uses Teams to store the records of the logins and outs to

streamline the login process since you must go through multiple subcategories to get to the sign-in spot making this much quicker and standardized. The ability to see who logs into the system is also shown in a box with a green circle showing online users who are filtered to the top of the users who are offline as well as yellow for away users. This will help with knowing who is online without having to check the schedule on our phones through the app which can be time-consuming. Using this system in combination with the profile page you can quickly message another staff member or TACling with the “message” button. Users struggle on occasion with signing in and this attempts to improve the sign-in rates.

- Information is stored in SharePoint Lists (databases) which lets things users put on the app stay and are viewable by other users like in Profile pages for example without sharing too much.
- Through the announcements page being on the home page and its own dedicated page, users can quickly get access to important information with a higher priority right next to the sign-in button. With the tasks board it is also possible to know what the supervisor wants and keep it written down for them somewhere like the tasks panel will help them remember to do the task. Admin users can remove or send only one email for an announcement, so it gets more views.
- Staff and Owners can regulate what student workers can see and each admin and user is only able to see what they need to see to use the app to prevent anything private from showing. Nothing private is on the site but if there was it would prevent anything from showing. Staff do not have access to change the other user's permissions or add new users. Only the owner or “Full Access” users. So,

supervisors or managers are the only users who can access the app and must be a part of the TAC Teams channel as well as be a member of certain email groups which is completed when a TACling is hired.

- DUO bypass and password reset requests are streamlined in the app because we did not have any dedicated system for making these requests, so we use Teams. This will hopefully give some incentive for higher-ups to make an official one since there is a lot of benefits in making these processes streamlined benefitting not just older TAClings but newer ones too.

Not Completed:

- 100% Mobile-friendly
    - The reason behind not fully implementing this feature is that we will be mainly using this on our laptops/desktops in the office. I foresee little usage of the app on the phone. This feature was postponed for the future as the next editor should try implementing a phone version of the page by detecting aspect ratio change. Currently, the app functions well on a phone in landscape and somewhat in portrait with some stretched objects but with full functionality.
- Onboarding
    - The reason for not working on onboarding this project is due to the scale plus we already have people working on this on Canvas and it is well maintained and is developed. Bringing over some of the resources onto the app was the goal and in the future, more links and documents can be used for things like training. Onboarding at TAC is important to the future success of each worker. It is not possible to learn everything in the five-

day training at the beginning of a new TACling's first semester. Many TAClings still have plenty of questions that go unanswered and using the Canvas to help refresh a user and even attempting to move up to a higher position in the process is the goal. A lot of unique problems come into TAC and only experience can really help. So, using things like the Chatbot, Useful links and documents page, and other resources that were added hope to improve the current onboarding but not redefine it.

- Scraping information from the TAC Teams channel to avoid commonly asked questions.
    - An attempt was made to scrape information off of the Teams, but it would only allow about 20 entries and did not show more than that. Handling that much data would require a few weeks of processing through cleaning and finding a way through Power Apps to use that information probably through a document that shows the question and how it was answered and if it was successful. We do not want questions and unsuccessful answers in our training set for a machine-learning model.
- Looking up student accounts without using our current script since it is outdated.
    - The script is something that only works on the UNCW computers and allows us at TAC to see information on clients to help diagnose their issues. The script runs a command in the command prompt and Power Apps cannot access a command prompt. I wanted to keep this project about Power Apps, so I did not look elsewhere for other alternatives to this issue since it was not a huge issue if this was not implemented into the app.

Testing Plan and Results of Testing with Other Users

In this section I will discuss how each user listed helped with testing the TAC App I have been working on. Some helped more than others but having more users helps broaden what is tested and helps give more diverse opinions. Not everything users brought up was changed but I tried to incorporate or fix issues that were deemed worthwhile. Things like what the person was asked to test, details of the process (the result of testing, feedback, how processes might have changed), and a summary of the changes that resulted in testing overall. Users who went from (date to 5/6) would occasionally pop in the app and try features out but never said anything about issues.

Users who helped with testing:

- Ixchel Peralta-Martinez (2/15-5/6)
    - 2/15 - Told the user to tell me what they thought of the color design for the current app. Received somewhat negative feedback on the colors similar to the UNCW colors. Was told to make the colors slightly darker since the colors were too bright. This helped with improving the overall feel of the app by having set colors to use which slowly iterated over time.
    - 2/20 – Told the user to test the chatbot to see if it would allow her to open and view the feature. She discovered that she was unable to access the chatbot because she did not have proper permissions, which was previously unknown. This helped me figure out some of the problems with sharing to other users and now it has been made a lot more streamlined and understandable in the TAC App Manual.
    - 2/20 – The user was prompted to test out the suggestions feature on the chatbot page to add things they want to see or if there are any issues. The

issue was that they were unable to submit and made an error saying the request could not be completed. This was because the SharePoint lists were not shared correctly because of a lack of testing prior. Also, they mentioned adding text saying the submission was successful.

- 2/21 – The user was prompted to try using the guilds page as an admin. They were able to add points. They saw that the table was not interactive and just showed a static number of points for each gallery panel. I added a way to look in different guilds and the ability to select guilds > users > individual points that were allocated.
- 3/7 – The user was prompted to test out the duo/password page and to be the requestor and then the person who takes requests testing things like the sound notification and if the information was displayed properly. They did not find any issues with it directly. On other pages, the sound notification would ding multiple times until the user switches pages. This works well as an intended feature.
- 3/8 – The user was to see if the number of times certain things reloaded was too much or not. The user said that it was fine. So I did not tune the timers much for reloading certain aspects of pages.
- 3/12 – The user tested out the home screen and was prompted to try different buttons for the current app. They recommended I add a search bar to the home page too since the search bar was originally on a different page. This was a smart decision and looks great on the front page in my opinion. It adds to the idea of it being all in one without looking overbearing. I was even recommended a page layout since they have done

many front-end parts of projects and are good with designs.

- 3/25 – The user was asked to try out the profile page. They tried different features like adding a description which was the only aspect of the profile at the time. They said that it worked well and added character to the page and in the future to add some more features like socials and smaller ideas to fill large blank spaces.
- 4/23 – The user was instructed to see if they could view images on the home page that I was adding. It was during the testing of a photo gallery page that my supervisor asked for, which took about an hour to make in total. This was asked for because it would add character to the page. The user who tested it said that it did not look like it fit on the front page and said I should make a dedicated page for pictures instead. This helped to form the current home page look which has everything neatly organized.
- 4/23 – The user was prompted to try the useful links and documents page. They noted that I did not mention anything about only PDFs working for documents, so I added that in the description of the creation page. This helped me with making sure I explained everything in detail during the documentation part of the project.
- 4/23 – The user was prompted to check out the redesigned profile page and try to add their start date and pictures to their gallery. They were able to do so but would notice that the join date would reset when editing and then canceling. I made it so that this was not the case and noticed the LinkedIn part having the same issues so that was fixed. This is important because users will be reluctant to use features if they know they could lose

their information.

- Ben
    - 3/10 – The user was prompted to guide me through how they would use the project since they did not have a UNCW account. They noted that the profile page's chat did not look right and should be put on a different page since it did not feel like it belonged. I ended up replacing it with a personal photo gallery when I had more skills and knew how to make the personal photo gallery near the end of the semester. This makes the profile page feel more unique. The chat did not have a place to go so it was stuck to the profile page originally as a guild chat for each guild. It is now a general chat.
- Courtney Lee (2/20-5/6)
    - 2/20 – The user was prompted to test the profile page and look at other user's profiles. The user did not find any issues.
    - 3/12 – The user was prompted to test the announcements and tasks page by making both an announcement and a task and deleting and checking off on both as well. They were able to do both. They noted that there needs to be a confirmation screen for the announcements and tasks so that users do not accidentally click on it. This prompted me to look at other confirmation options for other pages I was working on.
- Eric Guetta (10/11-5/6)
    - 10/11 – The user was prompted to test out a task feature with MS Teams. He was able to receive the task and approve it and send it back. I was trying to figure out ways to improve on tasks that could be incorporated

into Teams. This was before starting work on the project.

- 3/25 - The user was prompted to test out the tasks page and create a task. The user noted that they need a way to show who is doing which task and editing as well. This prompted me to make sure I mark down who created something and show who last edited it as well. This is an attempt to keep people accountable if they do something wrong.
- 3/27 - The user was prompted to try asking the chatbot questions. They noted that some questions would be answered but when missing important follow-up information, the answers were only half completed. I investigated making prompts that were more specific and would answer follow-ups for commonly asked questions.
- 4/8 - The user went and tested the profile page and tested the about me section, join date, and LinkedIn. They all worked as intended. The user said I should add a search bar for the users instead of just having to scroll. I went back and made a lot of the other scroll-only drop-downs searchable which improved the overall searchability of the app.
- 4/14 – Prompted the user to try the email button to see if they were receiving an email with the exact message in the email. They were successful in doing this.
- 4/27 – The user was prompted to try the requests page and try being the requestor and the user who receives the requests. This worked well but the user said that I should store the requests as proof of the request system being used and working. This prompted me to make another table to store all these requests permanently, or until they are removed.

- Raymond Khamthong (4/16-5/6)
    - 4/16 - The user was prompted to test out being able to access the app since I had not added any new users for a while. They were successful in entering the app and viewing the page on their phone. They said that the page looks off on the phone but looks fine in landscape mode while being a bit small. This made me look more into a solution to this. I was able to detect when an aspect ratio was more portrait than landscape, but it would bug out constantly, so I put the idea on hold since the app is mainly used on a Desktop or laptop which is landscape.
- Alec Mueller (3/10-5/6)
    - 3/10 – The user was prompted to test out the SharePoint Lists and make sure all the data looked correct and synced in Power Apps to SharePoint. They also tested different alternatives to SharePoint Lists for the future but came across issues with not being supported by Microsoft or not connecting with the Power Apps for free, which is the goal. This helped reinforce my point that SharePoint lists were a good idea and current ones seem like the best option available.

SECTION 8: FUTURE WORK

Future of the Project

This project has implications far into the future for improvements outside of my help. The system is going to need to be passed on to the manager and allow them to decide what changes should be added later. This comes with the need to make the project accessible and easily editable behind the scenes. Since staff change and managers change eventually, there needs to be a way to hand over the project easily. Not only that, knowing what does what will require documentation on everything for example, if I was explaining this to my grandma. Other people outside of TAC might see this website like the dean or the people above the manager who may not be as tech-savvy. In any case, someone who might take charge of the project might not know how to use Power Apps and Power Automate. These are possible scenarios since they are not as widely known as Python or Java.

The current manager at TAC wants the project to be built from what I make after I graduate in Spring 2023. This first level will be core features and later, more specific features as they arise, and more is figured out about the capabilities of these Microsoft products. This project could be a good template for a later more professionally made website that has all the same features but in a full stack where there are more possibilities and a bigger team. The current version will have many promising features in the build being made in Spring 2024.

Working on the project through the second semester of the capstone course has shown many ups and downs from roadblocks and breakthroughs that have led to new discoveries. When creating and implementing an idea, sometimes things do not always turn out the way you originally imagined but they can create new opportunities and show

new paths to success.

Some Future Project Ideas.

- Look for a way to do descriptive ticket searches or client look ups directly in the app
- Add a portrait version of every page that switched on mobile portrait mode based on the aspect ratio.
- Connect tasks board to tasks in MS Teams
- Add a current issues board
- Make pages work better with portrait mode

REFERENCES

Atlassian. (2023). Confluence: Your remote-friendly team workspace. Confluence. Retrieved from https://www.atlassian.com/software/confluence

Automation that moves you forward. Zapier. (2023). https://zapier.com/

Google. (2023). Google AppSheet: Build apps with no code. AppSheet. https://about.appsheet.com/home/

Google. (2023). Google Workspace. Google Workspace: Secure Online Productivity Collaboration tools. https://workspace.google.com/

High-performance low-code for APP Development. (2022). https://www.outsystems.com/Portal/Trial_Portal

Make | automation software | connect apps design workflows. (2023). https://www.make.com/en

McCormack, N., & Cotter, C. (2013). Managing burnout in the workplace: A guide for information professionals. Elsevier Science & Technology. 27-56. https://ebookcentral.proquest.com/lib/uncw/detail.action?docID=1575005

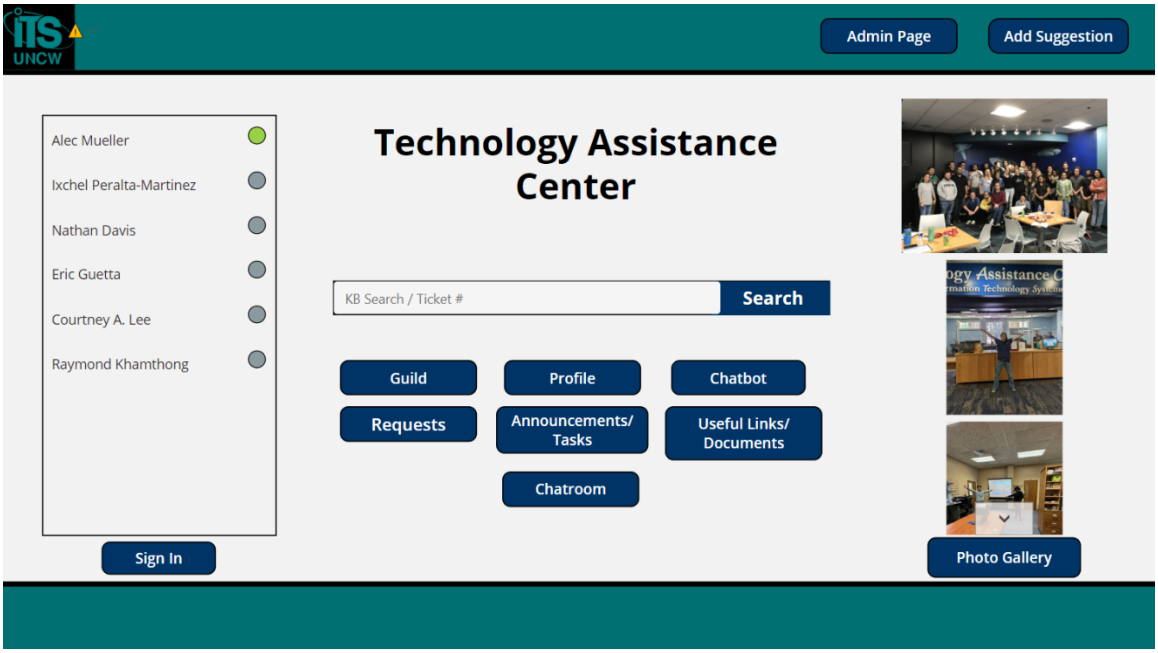
Narayn, H. (2023). *Building the Modern Workplace with SharePoint Online* (2nd ed., pp. 1–242). Apress Berkeley. https://doi-org.liblink.uncw.edu/10.1007/978-1-4842-9726-1.

Nintex UK Ltd. (2023). Process Management and Workflow Automation Software. Nintex. https://www.nintex.com/

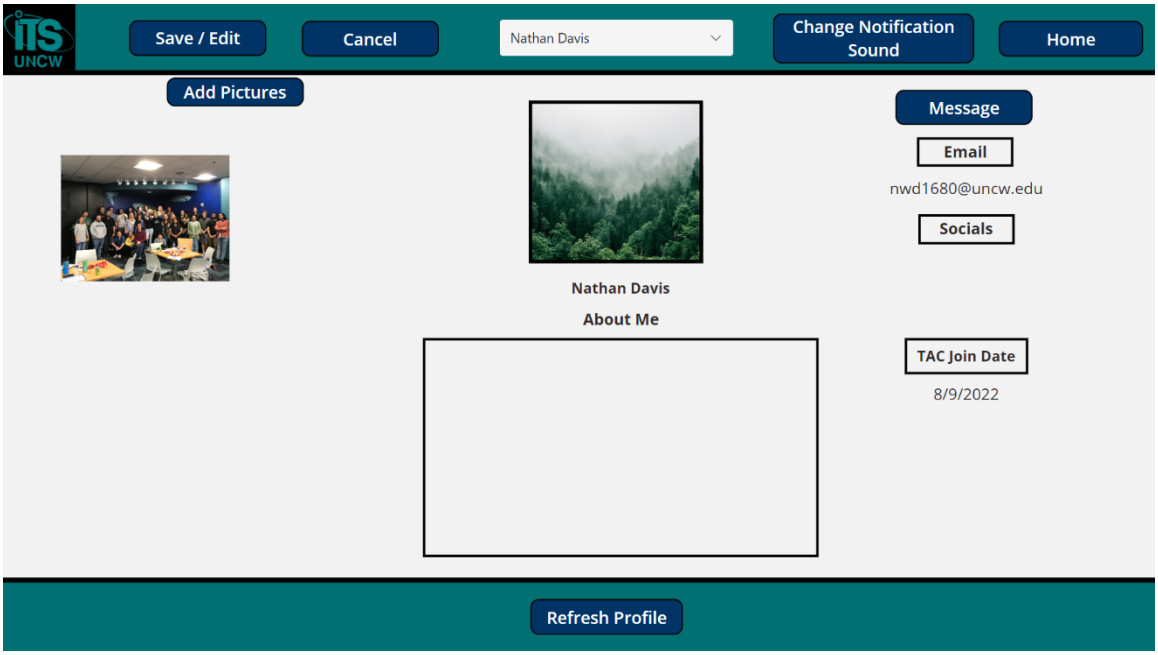
Secure team collaboration - Dropbox Business. (2023). https://www.dropbox.com/business.

APPENDIX A

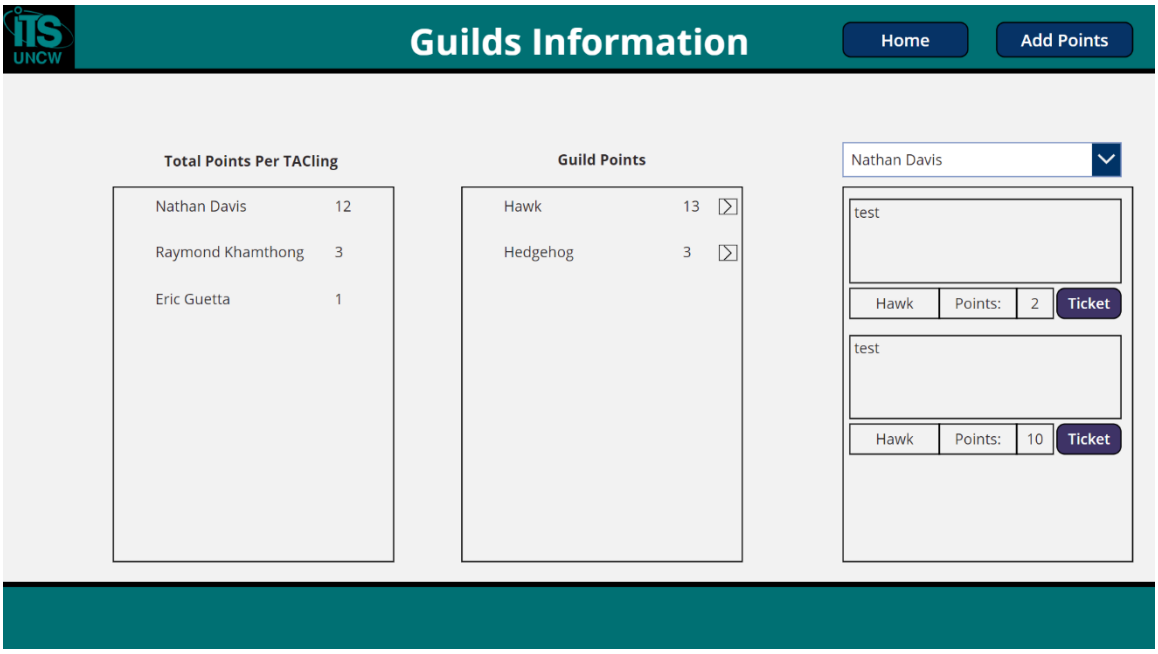
Application Photos



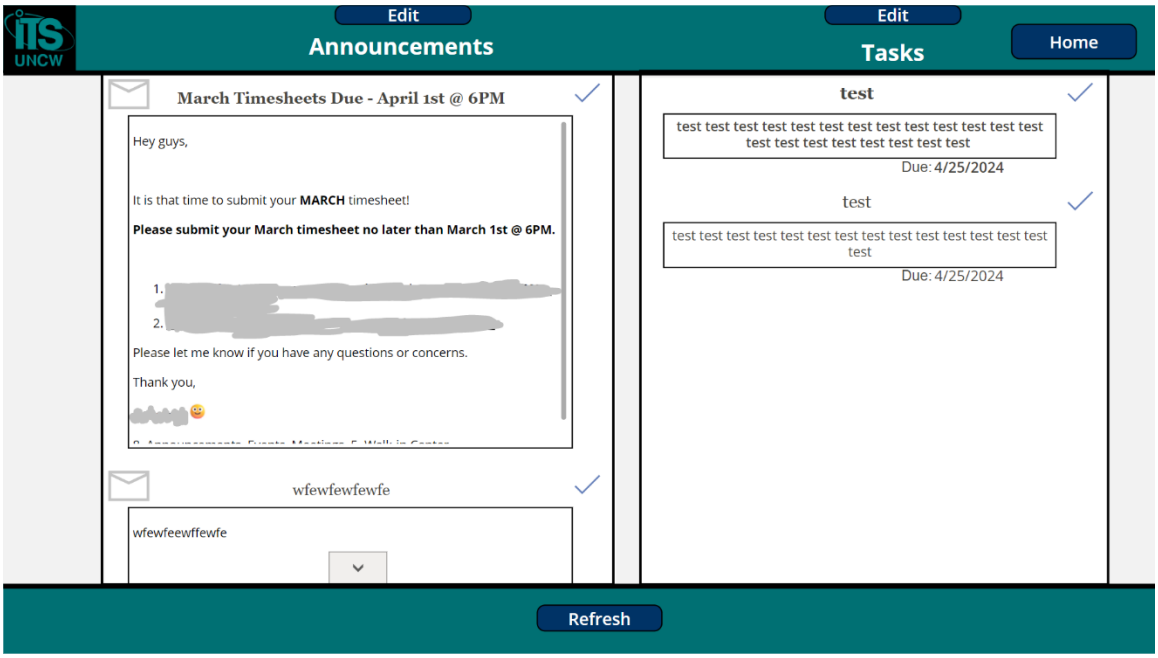
Main Page



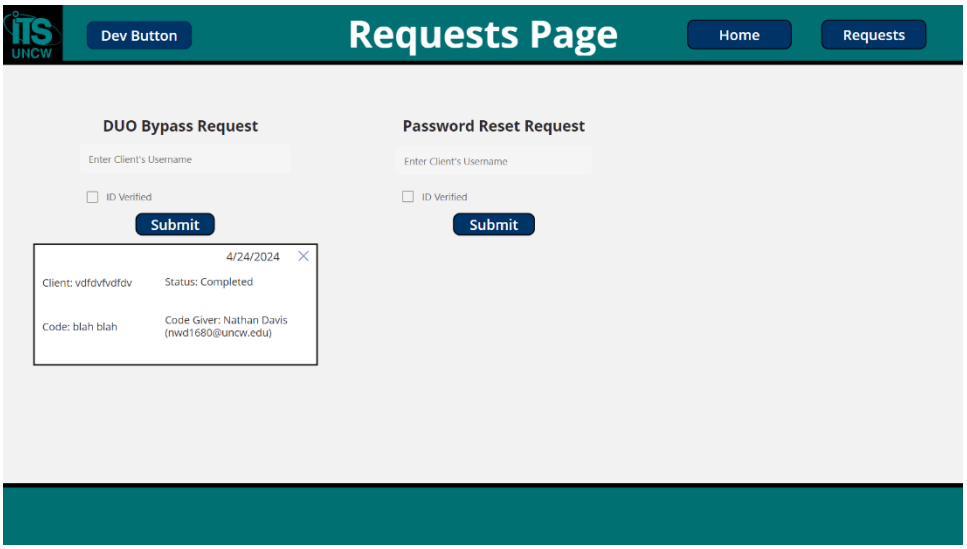
Profile Page



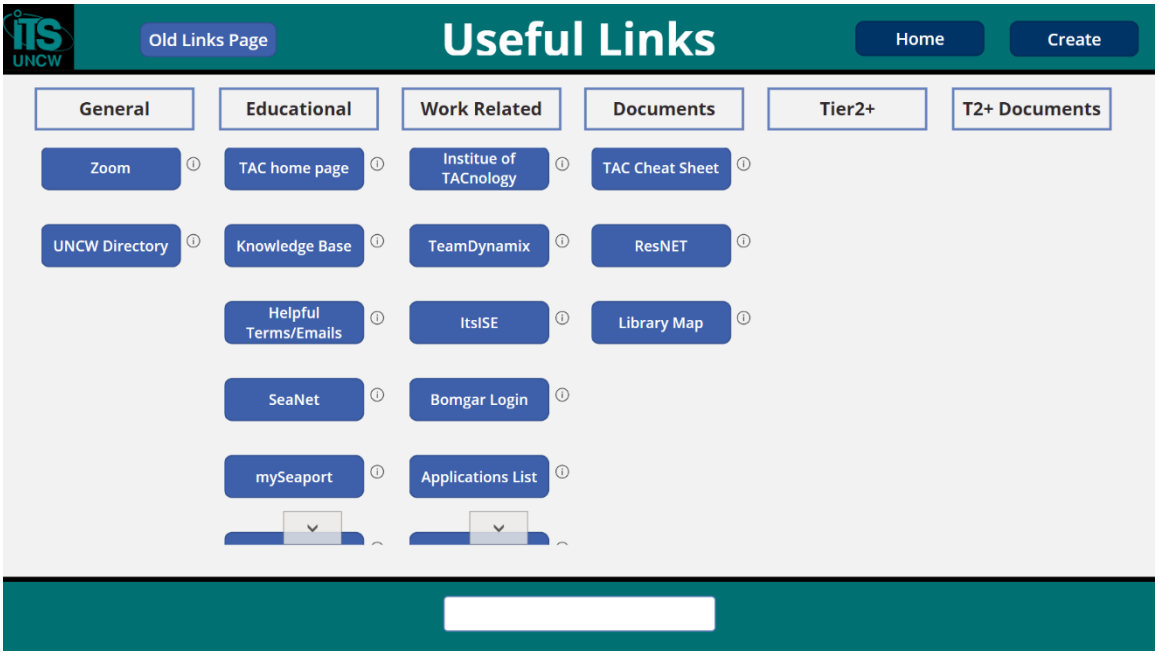
Guild Page



Announcements/Task Page

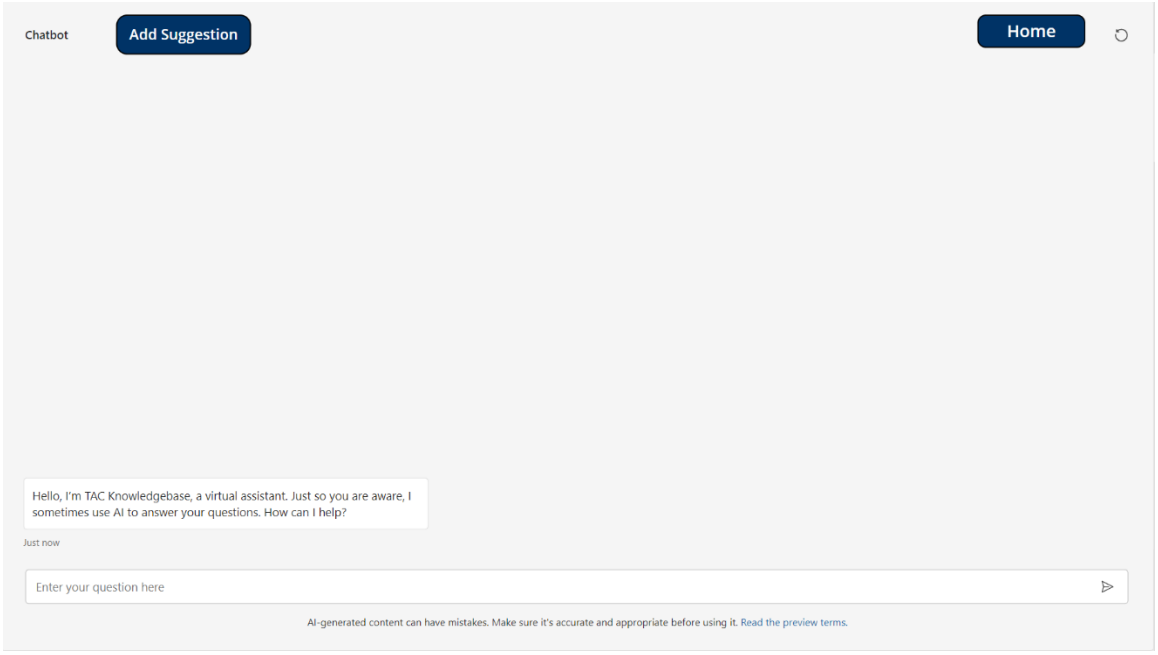


Requests Page

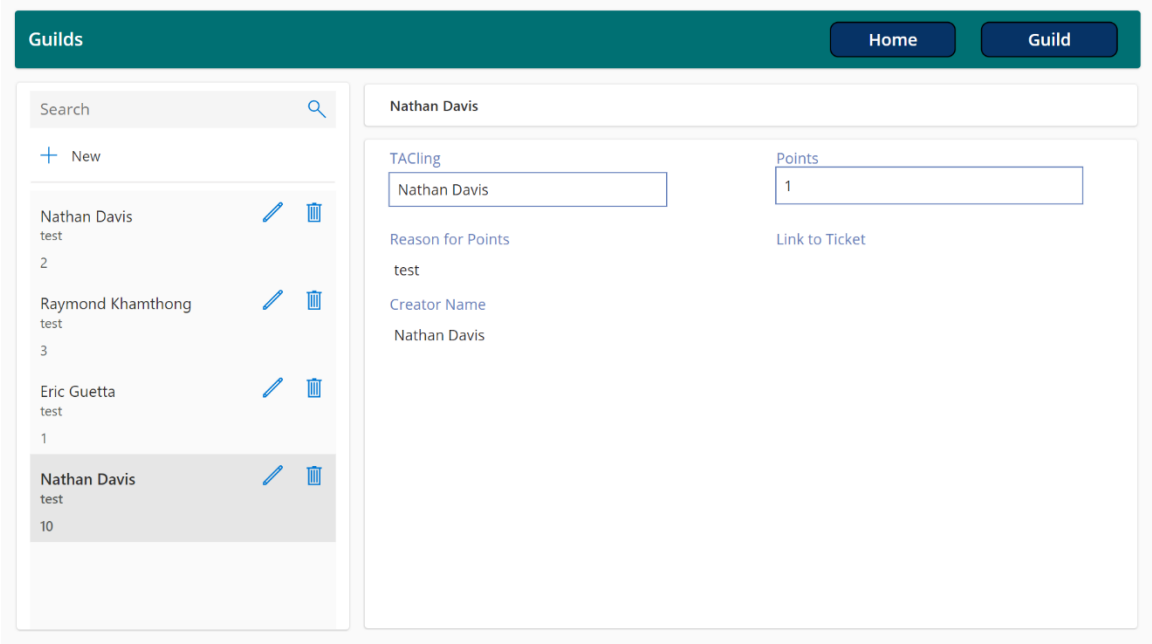


Useful Links / Documents Page

Photo Gallery Page



Chatbot Page



Guild Creation Page (Same Layout for Each Creation Page)