

FFPAC Application Final Report

# Team GVSU

Chad Mersino

Chris Schaefer

Laura Young

# Abstract

The web application developed for the Faculty Facilities Planning & Advising Committee is used to log complaints pertaining to specific classrooms on any Grand Valley State University campus. Members of the FFPAC perform walkthroughs in which they visit every classroom on campus and perform an examination of the quality of that room. Before we were tasked to build this application, the FFPAC members were using an unorganized and impossible to maintain system. Our project eliminates those negative aspects and was designed with the intent to be used on any platform, especially mobile devices. All of the complaints that are noted are stored in a database that can be referenced at any time. With that, this information can be conveyed to maintenance for repairs which provides an overall benefit to the classrooms on each campus. This project was developed using a WordPress site with a MySQL database to hold all of the data. Using WordPress allowed for the group to develop using PHP to modify the site templates in order to build the features for the application. The users have their own accounts in order to access the site which not only enhances the security of this information but gives an entry the ability to be traced back to the specific user. This application enhanced the developers’ programming abilities along with being able to interact with actual clients to get a taste of professional software development.

# Introduction

## The FFPAC

The FFPAC is a group of professors that represent every area of study on campus. Every semester they do a walk through of every classroom on both campuses with two purposes in mind:

1. Finding any problems with the attributes in the room, checking to see if everything is in good working order for the new semester and making note of anything that needs to be fixed.
2. Checking the status of the complaints already found in the room previously, making note of anything that has changed pertaining them.

The attributes checked can range from the electronics found in the room, such as the projector and computers, to the number of chairs that should be available. Everything needs to match to the predetermined set parameters of the room and be in good working order.

## The Old System

### How It Worked

The FFPAC has been using an Excel spreadsheet to keep track of any issues they came across during their walkthroughs every year. The committee members would walk room to room and write any issues by hand. After the committee finished their annual walkthrough they would then take their hand written notes and enter them into an Excel spreadsheet and send it to one another to get everybody’s notes in.

### Why It Needed an Upgrade

The Excel spreadsheet method was a good way to keep track of the data for a short period of time. In this case, when you have over a dozen people trying to maintain a single document, the users are bound to run into problems. After a few years, the document grew to over forty pages of entries that was unorganized and difficult to read. We have to keep in mind that there are hundreds of rooms on both campuses. With a document this large that has to be shared with over a dozen committee members, maintenance of this spreadsheet was out of the question.

## The New System

The team was given the job of creating an application that would streamline the walk through process and make the room details easier to access. The new application removed the hassle of having to update Excel spreadsheets by storing all of the information in a database and eliminate the aspect of parsing through numerous pages to find the desired information. The new application gives the users the ability to pinpoint an individual room by selecting the campus, building and room. This action would instantly load all of the complaints for the targeted room. In the list of complaints for the selected room, the user would be able to view information such as the complaint, who entered the complaint, the status of the complaint (fixed, not fixed, in progress or long term) and the date and time the complaint was entered. From there the users are able to sort and search for any text that they wish in relation to a complaint. In addition to the complaints, the users have the ability to add and view notes for each individual complaint. During the walkthrough process, users can quickly change the campus, building, and room that they are in and add or view complaints at the same time.

### Benefits of the New System

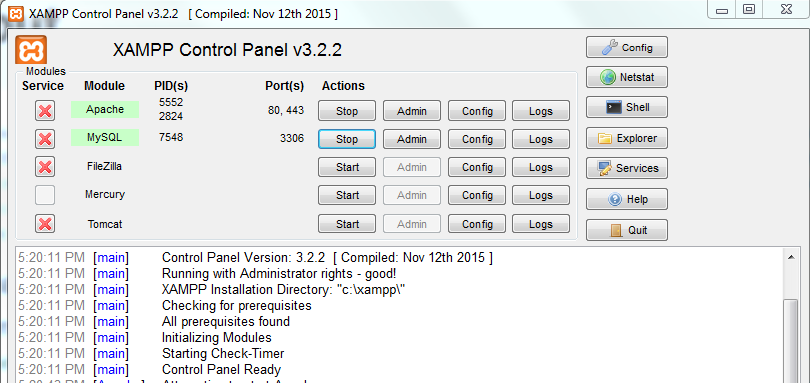
The new system gives the users the ability to quickly and easily find the information that they require. If the users come across an issue that is already in the database, they can perform a quick search and update the status of the complaint or add notes to enable maintenance personnel to get the most recent and up to date information. All of this information is stored in a centralized location that can be pulled up by any member of the FFPAC for reference. Since WordPress is mobile friendly, the ability to bring a tablet on any given walkthrough is now possible, which adds a tremendous advantage over a laptop and an Excel spreadsheet. Each complaint can now be tagged with a specific user, which is good for referencing later on.

# Body

This section will go over the basic building blocks needed to create this application for the FFPAC. This includes what technology we used, the frameworks used to implement our changes, the languages associated with the software and how the individual technologies tie together. We will also get into how this web application holds up in terms of reliability, security and other aspects outside of the technical requirements.

XAMPP

This software was used in the beginning of development when the application was first being developed. XAMPP is a free software we utilized to tie together our Apache server, our database and our PHP code for Wordpress into one location. XAMPP even uses those services in its name X + Apache + MariaDB + PHP + Perl. This program made it tremendously easier for development because all of the pieces we needed to run the application were already tied into one location. It offered us an easy to use interface to enable each service specifically depending on our needs at the time. Almost always we would use just Apache and MySQL since those were optimized to function on our WordPress web application.



## PHP

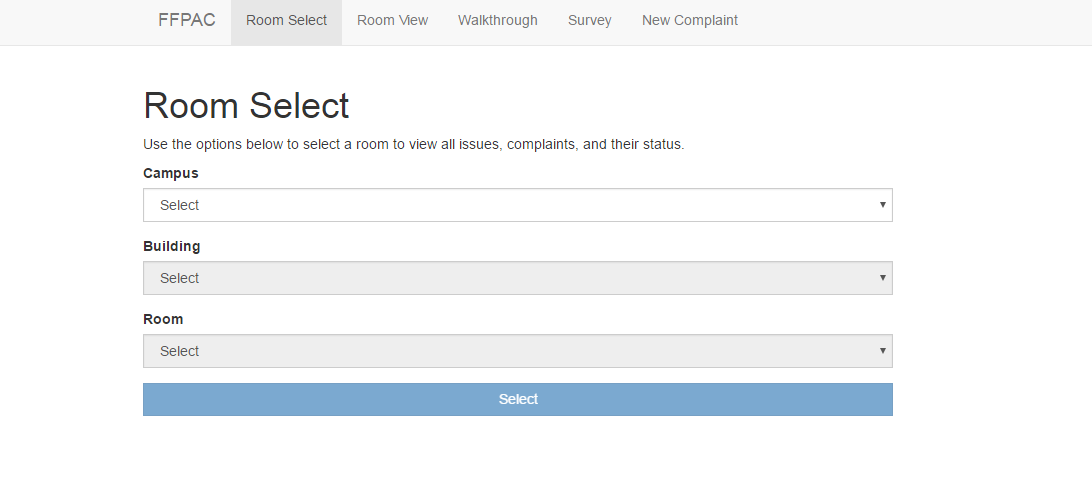
PHP is the programming language that we used to develop the FFPAC web application. This powerful scripting language allowed us to modify our WordPress templates in order to generate our desired features. PHP is open source and can be used in a variety of mediums to accomplish various tasks. The language is mainly used for server side scripting, command line scripting and for writing desktop applications. For our web application, we used PHP’s abilities to write server side code, which is found inside the site templates that we started out with. This is what gave each page it’s distinct characteristics and features. Each page we made was custom by using PHP inside the templates we created and these methods were tested during development, especially through rigorous input testing. Whenever we came across a logical error, we could see instantly how incorrect the solution was and we were able to make up for that by updating our logic and running the application once more. This language constantly gets updated and grows more in its functionality for web based applications. PHP can be used on any Operating System including but not limited to Linux, Mac OS, Windows and AIX.

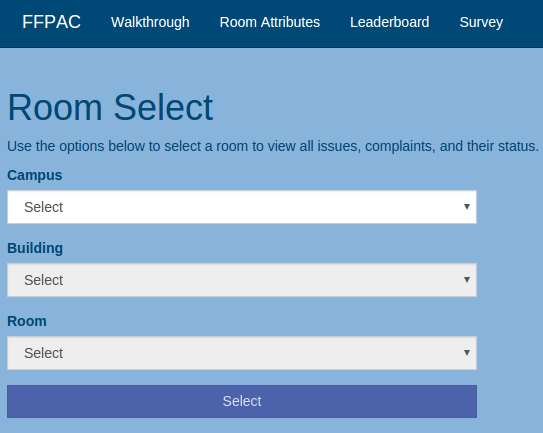
## CSS and SASS

CSS stands for cascading style sheets which is used to define the rules for the elements in HTML markup. These are basic in nature but the simplicity of giving character to a website brings about a huge change in user experience. The application needs to function properly for the users along with being pleasant to work with. This is not only concerned with the structure of the web page but also the theme. The group believed in making this a GVSU themed application to tie into the rest of the sites under the GVSU domain. For this application, we decided to use SASS which stands for syntactically awesome style sheets. This follows a similar pattern to CSS however, it provides the ability to nest elements and add mixins that build the rules for more complex elements. SASS uses a compiler to go through each file and output a single or multiple cascading style sheets. The way SASS files work is simple after some practice. By nesting elements that have similar traits under one another, the SASS compiler generates the rules for the nested elements using the same rule for each. Our team took advantage of the power of SASS and used it to build our style sheets that gave the FFPAC web application the look and feel that it has. For example, whenever the user generates another window or adds another entry, the SASS recognizes a new element has been generated and applies that CSS sheet to that element so it remains consistent with the rest of the application.

## Bootstrap

Bootstrap is a free front-end framework that gives designers the ability to design and build responsive websites faster and also allows for modifications to enhance the look and feel of a website. Our team took full advantage of Bootstrap by being able to spend more time focusing on the database and performing the required logic for the FFPAC web application.

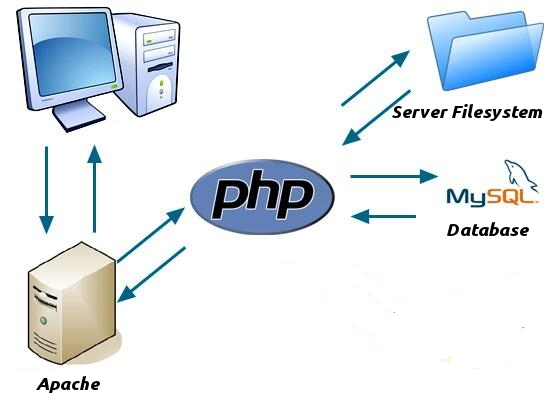
Room select page with no custom CSS



Room Select page after modifying Bootstrap’s default CSS using SASS

## Apache HTTP Web Server

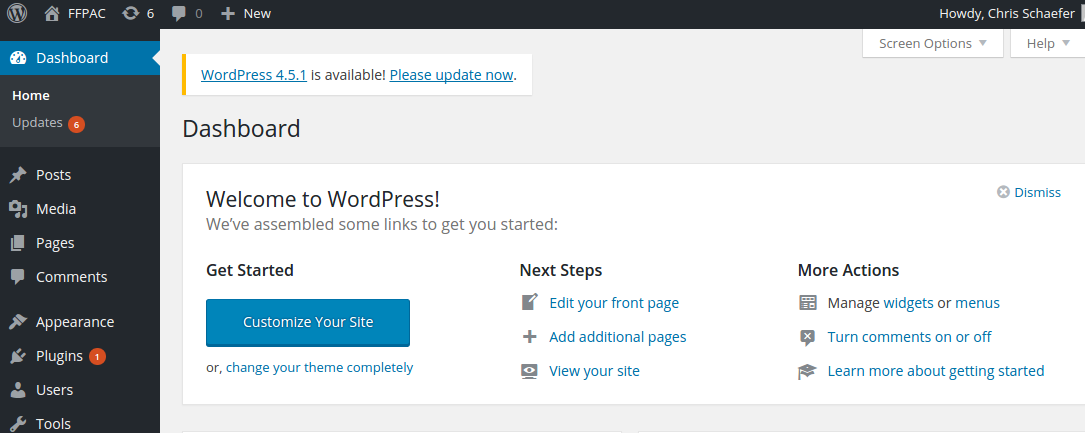
Apache is a lightweight web server that is ran locally on one individual machine. This program is the engine behind the FFPAC web application when it came to development. When a request comes in from a user’s browser, Apache is on the other side receiving the requests and serving the application to the user. Since 1996, it has been the most popular web server that powers over 100 million web sites. Apache can be configured to run PHP, Perl, Python and CGI applications. Since we wanted create the FFPAC web application in PHP, Apache was the most logical decision for the job. By generating quick and on the fly changes to our application, Apache was our server that showcased all of our code changes instantly. This was critical throughout the development process because making minor changes and then having to push the newly updated code to a server outside of our local host is time consuming and a pain to deal with. Apache would be used to test our development changes and then from there, it was sent off to our main server for everyone to see the new improvements. The group members could do a Git pull on our repo and instantly have a live local server up and running to make their desired feature changes while testing at the same time. Once that development session was over, the developer would push the new updated code to Git for other members to pull and test themselves. This was our main development style and without Apache to constantly spool up or web application, development would have been more tedious and time consuming.



Apache’s interaction with our code and MySQL

## WordPress

WordPress is an open source tool that has a framework and templates that are written in PHP. In today’s era, it is mainly used for building blogs however it can be modified to create much more meaningful applications based on a theme. By design it can be installed and configured in under 5 minutes which was great for starting out this project. This takes a lot of the stress of building websites away because the programmer can get right down to creating content and modifying the design of a website. The ability to reference an open API for any of WordPress’ functions and features was another benefiting factor that was a clear choice for the group to work with. The learning curve for the group was not as extreme compared to other software that is similar to WordPress. Once we were able to generate our site, the group could then start dissecting the theme and figure out what direction to go in for this application. Another advantage of WordPress is their ability to keep constantly updating their software which aids in the lifespan of our application. With their extensive updates comes a lot of built in functions that need to be maintained constantly. The biggest feature the group enjoyed was the fact that the site was always mobile friendly and it took very little work to ensure our application would properly function on those devices.



WordPress Dashboard

## WordPress Enhancements

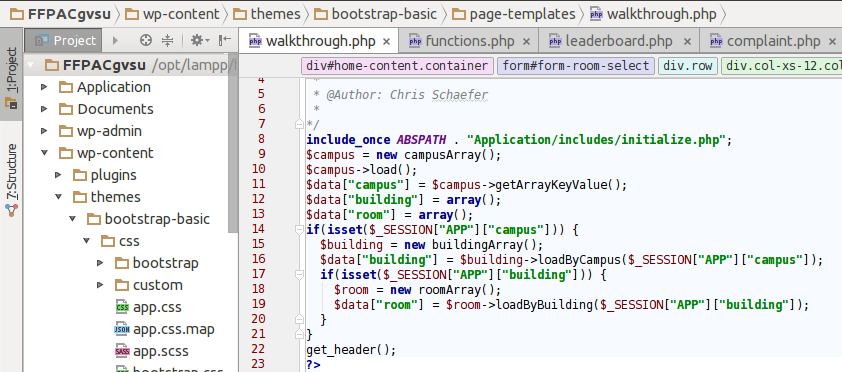
There are an infinite number of ways to enhance a WordPress application outside of the pre-defined templates out of the box. Plugins are the most common but they require a deep understanding of how the framework functions depending on how advanced of a plugin is being made. For our application, we chose to use a simple template enhancement that adds logic to the templates to allow our users to perform the necessary CRUD (Create, Read, Update, Delete) operations on the additional database tables. By adding this functionality to the web application, we were able to customize every necessary part of logic required by the FFPAC. This also made development on the front end much easier since it was extremely easy to pull values from the back end database. WordPress already defined the basic logic for database management and this application takes advantage of that as it’s core functionality. This also reassures the safety of this information since WordPress take this as a staple feature in their application templates and periodically updates all of these extra services with the templates.

### How It Works

Every WordPress page is assigned a template by the developer. These template files, at the most basic level, define how the page looks through HTML markup and the cascading style sheets. Since WordPress is written in PHP, this also gives control of logic and workflow to the programmer. By creating individual template files for each page that required a specific workflow, we were able to enhance WordPress to do exactly what was required by the clients specifications. Each page was its own specific instance of code we implemented into that given template which is why all of the pages are different in their functionality.

## PHPStorm / JetBrains Inc

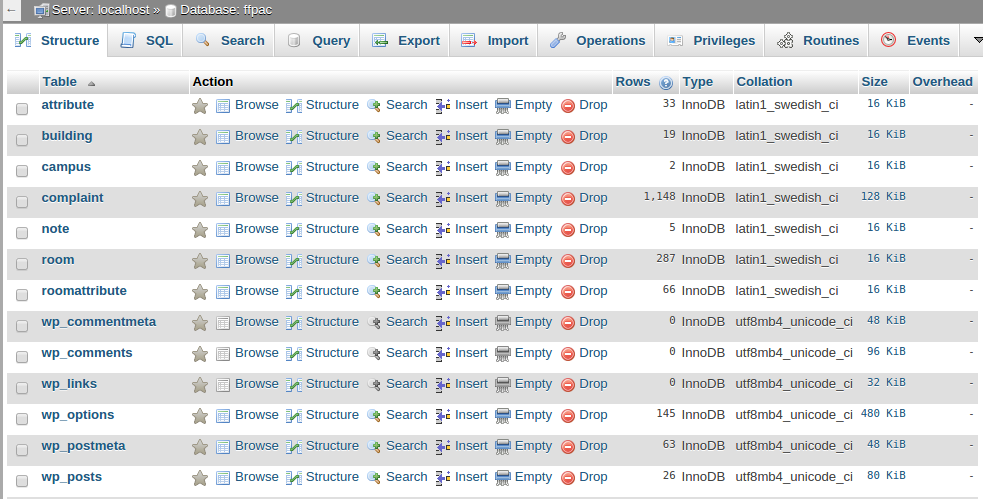
JetBrains Inc. has been a leader in IDE’s that provide support for code completion, GIT integration, built-in servers and debuggers. For the FFPAC web application, we used JetBrains Inc’s PHPStorm to make development and debugging faster and more effective. The most useful features that we took advantage of for the application were code completion and GIT integration. All of JetBrains IDE’s index all of the code from the beginning of a project to the end. This was extremely helpful working with WordPress due to the plethora of built-in functions that are hard to remember. Code completion helped save time looking up specific functions. The other key feature that we took advantage of was GIT integration. Learning GIT can be a struggle but with PHPStorm we were able to use the intuitive GUI provided to push and pull from GitHub. With the helpful features provided, our team was able to make development and collaboration faster and easier.



PHPStorm IDE

## MySQL / MariaDB

MySQL is the driving force of our application that brings everything together into one location. It is a database that stores and returns data that we save. MySQL was acquired by Oracle and came to be known as MariaDB. Both MySQL and MariaDB serve the same purpose of providing a reliable database that our application communicates with. These two databases have become extremely widespread and can be used with almost any programming language. This makes MySQL / MariaDB an unbelievably useful tool to any programmer. Our language of choice for the FFPAC web application, PHP, was originally designed to work with MySQL but has been extended to include the functionality to communicate with all sorts of other databases including Microsoft SQL, DB2, and SQL Lite. MariaDB is officially supported by Drupal, ERP5, Kajona, MediaWiki, Moodle, MONyog, ownCloud, Plone, WordPress, Yii Framework, and Zend Framework. As of October 2015, the XAMPP stack switched to MariaDB as well. This is great because there is extensive documentation no matter where we search and with previous courses that covered database systems, this part of the project was straight forward. Since we also receive credible information for our campuses, it gave our application meaning and cemented the purpose of this application for the clients since our database was populated with the most up to date classroom info.

Our MySQL database

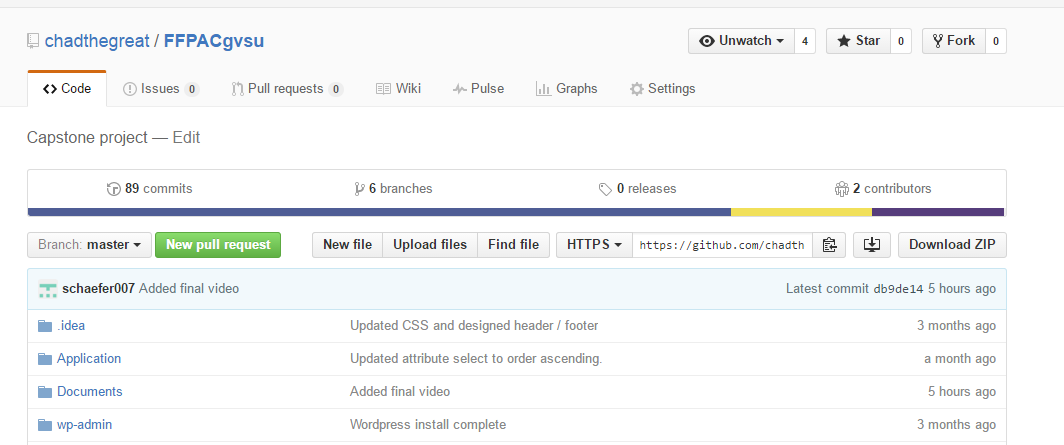
## Nearlyfreespeech.net

NearlyFreeSpeech (NFS) is a great hosting service for sites with generic needs. Although it is independent of any third-party portal, the infrastructure does depend on a lot of open source software although sometimes this can come with restrictions. Either a stand-alone server or a small virtual machine is instantiated based on the size of the site. SFTP and SSH are automatically configured for the user to access their site. Their clustered hosting network mitigates downtime that would normally be experienced with a singular stand-alone server. However, load balancing, DNS failover and other configuration measures are restricted. NFS Host manages most of the server configuration but is perfect for dynamic sites. Simple LAMP stacks are served without much hassle. All MySQL services are run independently from the site server instance. Advantages of this include less resource consumption, replication automation and automatic installation of database management GUIs. Disadvantages include less control over your environment which can easily be worked around in most cases. Another convenient toolset found by hosting with NFS is account and domain management, which helps with reducing the restrictions on our environment.

After creating a NFS member account, you are directed to the dashboard. Here you can add personal information and payment information to fund your site, which was not necessary in our case. The personal information is also integrated when purchasing domain names and the information is automatically populated for the whois information. NFS allows users to check domain availability and purchase them in a few easy steps. This is helpful when creating a site. After registering a domain name, the system is set up with their own DNS servers which use your purchased domain names as an alias to the IP address of your server. This process is a good introduction to how creating a site from scratch is completed. NFS has plenty of documentation on how their system automatically configures these features for you.

## GitHub

GitHub is arguably the most common website that allows teams or individual users to store and maintain their projects. GitHub provides a clean front end to view any changes that are made to a project. GitHub extends the functionality of the popular version control software Git to give users a visual representation of their projects. This is how our group was able to transfer code cleanly and offered various steps of development along the way. It gives us the ability to fall back on previous versions of our app if we needed to restart implementing a feature or if some error were to happen. Having that safety net in software development is critical in some instances and it is better to be safe than sorry. This technological aspect was also a core requirement for the Capstone course because of the numerous features it possesses outside of storing hard code. We used this for logging the group’s hours spent on the project., displaying project information along with statistics specific to each programmer for the entire project.

FFPAC GitHub Repo, example of folders with formal document write up’s / our WordPress code

## Tying It All Together

For this project, each of the listed pieces of technology had to come together to form a functioning, aesthetic, and pleasant application for the end users. We first had to start by deciding what we wanted to use and were relatively familiar with to get this project on its feet. We determined that WordPress was a reliable and secure framework to do that. From there we had to set up each of our machines so that we could all collaborate. We began by getting XAMPP installed to serve the WordPress application followed by Git and PHPStorm to allow us to productively begin work on the application. After we had all of the software installed we installed WordPress and cloned the repository on each of our machines. Slowly but surely we extended WordPress’s default functionality to include the logic that was required and enhanced the default Bootstrap stylesheets to create a pleasant theme that flowed with Grand Valley’s. After we had the application put together and functioning we then used Git’s functionality to clone onto our Nearlyfreespeech.net server and keep the code up to date. MySQL’s export and import functions helped us to keep our data synchronized across all of our application instances. After all was said and done, all of the pieces of the puzzle came together to produce a useful and elegant web application that we believe will prove to be a solid proof of concept to the FFPAC.

## Areas of Technical Growth

The areas of technical growth are many and the best way to approach this would be from the top.

1. From the start it was decided to use WordPress and it was something that most of the team had no idea how to use. Learning the functionality and use was a great tool that each of us will continue to use in our future endeavours.
2. GitHub was something that each of the team members have used at one time or another in their college careers. The skill of using GitHub was further defined in this class as we learned how to use it correctly. It is something that will help us a great deal in future.
3. The milestones provided a great deal of growth, starting with the paperwork needed and ending with the presentation. With each milestone, the team improved significantly in the quality provided and it showed in the work.
4. In many past classes, the members of the team had been told many times what to possibly expect from future clients. It was good to see that most of the things we were told were true and that it is something to expect in the future.

# SECEPP Issues

After studying the SECEPP in depth, the group justified all of the attributes in principle 3 to be applicable to this development process. This section will elaborate more on what the group thinks the best example is to explain how most of the attributes are covered during this project.

3.01. Strive for high quality, acceptable cost, and a reasonable schedule, ensuring significant tradeoffs are clear to and accepted by the employer and the client, and are available for consideration by the user and the public.

* The group believes in the work that they produce and realizes that this application can have a significant impact on the FFPAC. When optimizing the mobile aspect of the application, we wanted to be sure that the code was catering to these devices seeing as this is the ultimate convenience to make notes while on walkthroughs. This also was a feature that included no extra cost, didn’t involve scope creep and this was agreed on by everyone involved in the project.

3.02. Ensure proper and achievable goals and objectives for any project on which they work or propose.

* The goals that we set out for were all achievable within the given time frame. There were some moments where features that were brought up by both sides were evaluated based off a number of things. Each idea was tested for the necessity for that feature, the length of time it would take to implement and how optimized is it to be an acceptable feature for the application.

3.03. Identify, define, and address ethical, economic, cultural, legal and environmental issues related to work projects.

* This was a basic consideration at the beginning of the project that was quickly laid to rest in terms of concerns for the group. Our application in no way has any ties to any of these aspects listed in the rule. Since the nature of our application is simple, it leaves us with no possible way to cross any unethical or immoral lines there are in today’s society. There is not a chance that anything inside the application can be taken offensively. The only thing that can be accounted for is the work we put into the project was genuine and that we did it for the good of adding convenience to those on the walkthrough.

3.04. Ensure they are qualified for any project on which they work or propose to work, by an appropriate combination of education, training, and experience,.

* For this project, two of the group members were in need of performing research on the different aspects in the application. The biggest thing that those members needed to work on was learning PHP and how it is used in WordPress. Thankfully, WordPress offers extensive documentation and through the education of GVSU, the development process was relatable to normal class projects except on a larger scale. One member in particular was extremely experienced in this type of work and was more than willing to offer up help to the rest of the group when problems arose.

3.05. Ensure an appropriate method is used for any project on which they work or propose to work.

* Developing on this project was a total eye opener in terms of coding in a professional manner. When choosing how we were to go about this, we considered how often we would contact the clients, what features we wanted to be implemented and how long each feature would take to get onto the website. We chose to contact our clients whenever we felt there was a significant change in the functionality of the application. Even though this was not as often as other groups, we were able to still develop an application with the client’s desires in mind.

3.06. Work to follow professional standards, when available, that are most appropriate for the task at hand, departing from these only when ethically or technically justified.

* Two of the group members are currently in the field of software development and are getting an actual view of how professional business are ran. They were able to draw from those experiences at work and apply them to the project to showcase their knowledge and take this course much more seriously. When it came to communicating with the employees, through email and in meetings, we were sure to address any of their concerns and alleviate with the best of our abilities. The group was professional in communication, which made obtaining notes for revisions much easier.

3.07. Strive to fully understand the specifications for software on which they work.

* In the first meeting we held with the FFPAC, the group was extremely attentive to the discrepancies the members had with their old system. From that, we were able to point the clients in the right direction by using systems similar to this application that can be modified to their needs. Every time we contacted the clients, the group constant questioned if this application is what the client would see fit to their needs. We were extremely happy to receive any feedback and quickly make any changes possible to create the perfect application for them to use.

3.09. Ensure realistic quantitative estimates of cost, scheduling, personnel, quality, and outcomes on any project on which they work or propose to work and provide an uncertainty assessment of these estimates.

* When it came to our milestones and describing what we needed to accomplish during those weeks of development, we were aware of everyone’s availability to meet up and their timelines of what they needed to get done. From there is when we plotted out our objectives when we had a firm grasp of what we needed to build for our clients

3.10. Ensure adequate testing, debugging, and review of software and related documents on which they work.

* The main motive for us to test this web application is by running through as many inputs as possible. We did this through a variety of accounts, characters and different ways to use the application on different devices. We were confident in the code we extended into the WordPress template and that was put to the test in our rigorous brute force inputs.

3.11. Ensure adequate documentation, including significant problems discovered and solutions adopted, for any project on which they work.

* This rule was showcased through every milestone we had during the development process. Every time we had a setback we were sure to elaborate on every one in detail so we can learn from that mistake and either work around it or try implementing a new feature to replace the one that was not possible.

3.12. Work to develop software and related documents that respect the privacy of those who will be affected by that software.

* Our WordPress site is protected by the user accounts that have access to this information. Since that is a basic feature of WordPress and the software we use is constantly updating their security policies, this worry was quickly alleviated from the start.

3.13. Be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized.

* Our data was acquired by Laura Pryzbytek, who is the Associate Registrar for our offices here at GVSU. She gave us the most up to date information on the campuses and was crucial to the functionality of our application.

3.14. Maintain the integrity of data, being sensitive to outdated or flawed occurrences.

* Due to the popularity of WordPress and the constant updates, we know that it is always being updated for security flaws or just enhancing security to keep up with technology. User passwords are securely stored in the database by means of hashing and are nearly impossible to crack. Even if the server our application is hosted on were compromised, our users passwords would not be.

3.15 Treat all forms of software maintenance with the same professionalism as new development.

* With our choice of framework, we are confident that our software will be able to last for quite some time after we are gone from GVSU. We are limited in our ability to enhance the software after we are gone but for the time being WordPress should keep the integrity of this project going.

# Conclusion

## Lessons Learned

Throughout our time working on the FFPAC web application, we have learned a lot. First, we have learned what it is like to collaborate and work effectively as a team. This includes communication which is the most important part of collaboration. We have all become extremely familiar with the popular version control software git. This enabled us to make changes to the application without the need for constantly sending each other our code. Second, we have learned about working for a company and how to document requirements and keep a schedule. This process has given us an idea of what the real world will be like and what we can expect after being hired by a future employer.

## Major Difficulties and Setbacks

The FFPAC web application was fun to work on but also came with its fair share of trouble. The biggest issue that we had was getting the application live on a server 24/7. We tried Amazon, GVSU, and Cloud 9. With Cloud 9 we were actually able to get the application up and running in minutes, however, the service was not free and would only stay running for a limited number of hours which did not allow the FFPAC the ability to look at it and find any changes they wanted at their leisure. The second issue that we had with hosting was trying to get the application running on GVSU servers. We were able to get the code on the servers but the database gave us permissions issues that we were not able to get resolved. We were able to borrow a server and get the application running.

Syncing the Grand Valley databases to our database was a large setback. Our hope was that we would be able to get the application running on Grand Valley’s server and have the database there as well. Unfortunately, through the process described above that was not possible. This limited our ability to import data directly in MySQL. However, if we had been able to get the database and application running on the Grand Valley servers we found out that for security purposes we would not be allowed to connect to the heart of the data that we needed. This meant that we would have to be given a CSV or Excel file with the data and enter it into the database manually. We were able to get the data entered manually and continue to build the application.

## Potential Enhancements

If time were to allow us to continue development on this application, there were more features brought up from the design critiques and from our clients that could serve as great additions to the web site. One feature that was brought up was the ability to upload files to notifications. We attempted on a couple of occasions to implement this into our new complaint note box but with no success. Going through the process of setting up more structures for holding this file along with keeping the integrity of the data allowed was challenging. If the deadline was not approaching as quickly as it had, this could have been another staple in the application. Another feature we wanted to get in was the survey page with our own custom survey that can store that information in our application. The group thought about this feature and decided that the old survey was fine because it offered the correct information to enter in however it would not be stored within our database. The group also had trouble implementing the email notification system due to the fact that we were skeptical about determining when notifications were necessary. We did not want to spam the user’s email with unnecessary notifications, however notifying the user when renovations occur would be potentially useful. Then there is the problem of having a maintenance worker actually update the site, which requires that individual to do more work than previously. So that idea could work only if everything can line up properly, otherwise it was best to leave that idea out for the time being.