Final Report

Abstract

As classrooms and faculty adopt new technologies into their teaching methodologies, the need for a simple, elegant control scheme increases. At Grand Valley State University, we've seen overhead projectors become LCD projectors, the introduction of a teaching station with several AV inputs, and the addition of cameras and microphones for recorded lectures. Users may have had difficulty navigating the use of each piece of equipment and have experienced frustration when attempting to combine the classroom technology into one cohesive teaching environment. The goal of the Smart Classroom solution is to reduce the friction often experienced by faculty, while at the same time taking advantage of the Internet of Things and connected devices. Our proposed solution uses a web application connected to a database that allows users to set technology preferences prior to their class session, resulting in a less stressful and more effective teaching environment. Future contributors providing updates to this project will allow the user to scan a QR code located in each classroom, and watch as their preferred settings are implemented in the room's technology.

Each user must be verified and explicitly given permission to set their own technology settings by the administrators of the project. Once the user has the proper permissions set in the database, they may personalize their settings for specific classroom technologies. Additionally, an inventory of what technology exists in each classroom is available in the application. Future updates to this project will allow the user to set preferences, scan a QR code located in each classroom, and watch as their preferred settings are implemented in the room's technology.

Introduction

Professors and students who struggle with setting up their classrooms and classes will have access to a simplified and easier solution. As a user accesses the application, they will be confronted with recommended settings for their particular classroom and layout style. The user will be able to use and modify these settings through a web application interface. Making modifications to these settings will update a database that stores every user's personalized settings. These settings are recalled when logged in. The system of CRUD (Create, Read, Update, Delete) is implemented through the use of REST calls running on a Node.js server. The database is hosted via the cloud through a service called MongoDB Atlas. The REST calls are made through MongoDB's query language and then parsed through JSON. When these calls are made, JSON is passed back and forth between the application and the database. The web application is generated and regenerated after it receives data from the database. This creates an environment that is driven primarily by what exists in the database. From the data that exists in the database, the user is able to manipulate that data, which in turn will create specific settings for that user. These settings would then be used to drive any technology in the classroom that those settings exist for. The system is organized by the diagram below (Figure 1).

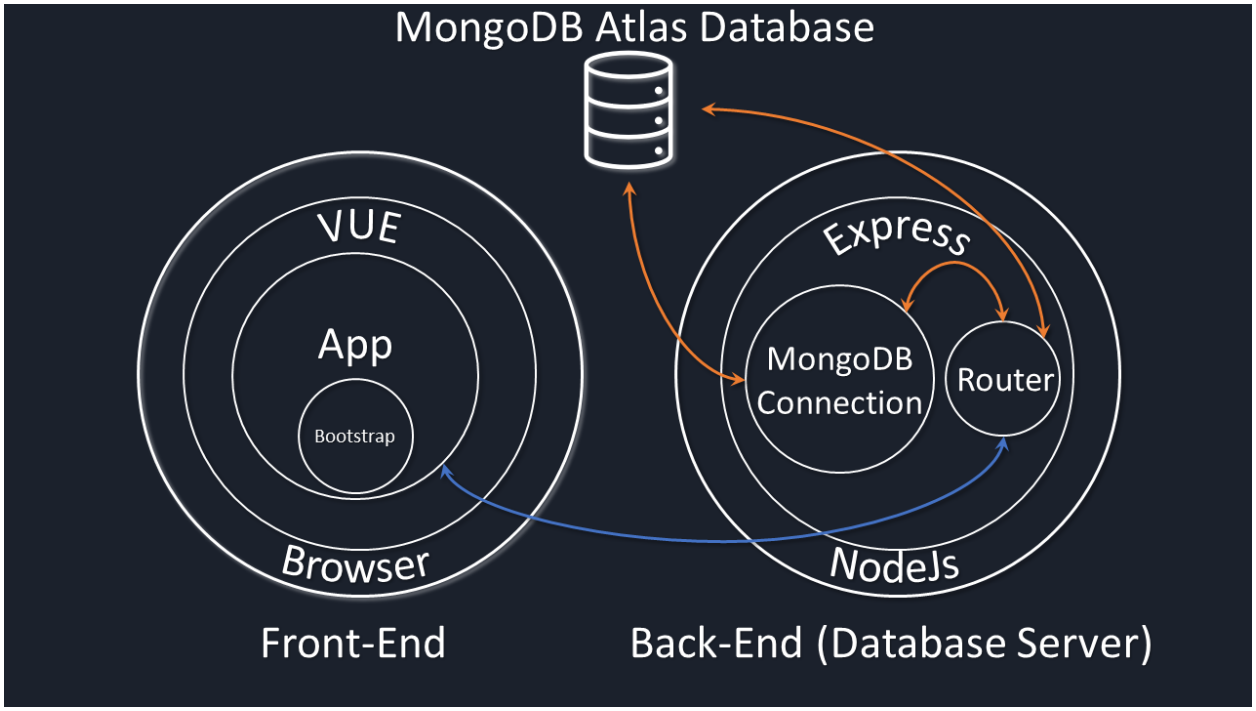

Figure 1: Organization of the System

Smart Classroom

Traditional classrooms are quickly becoming a thing of the past, especially when it comes to their tech. With the COVID pandemic at the start of 2020, classrooms have been moved out of favor with some students, while others prefer the in class style of learning. In most universities, classrooms weren't built to handle this new wave of students. It is time to take the initiative and become the future that students are needing to accommodate their various learning methods. We are in need of a way to make our classrooms intelligent and react to these needs. What Smart Classroom intends to bring to classrooms is professor control to make those changes to their liking and to present these controls in an easy to understand format.

Using a database driven web application, we were able to accomplish this. Settings are manipulated, saved, and displayed through an easy to use interface. These settings in turn could be used to manipulate the technology that is used in a smart classroom. When a professor enters the classroom, they would initialize those settings by scanning in, the technology would then react to their specified settings. This technology includes, but is not limited to, projectors, microphones, cameras, and smart whiteboards. This strategy is displayed in Figure 2.

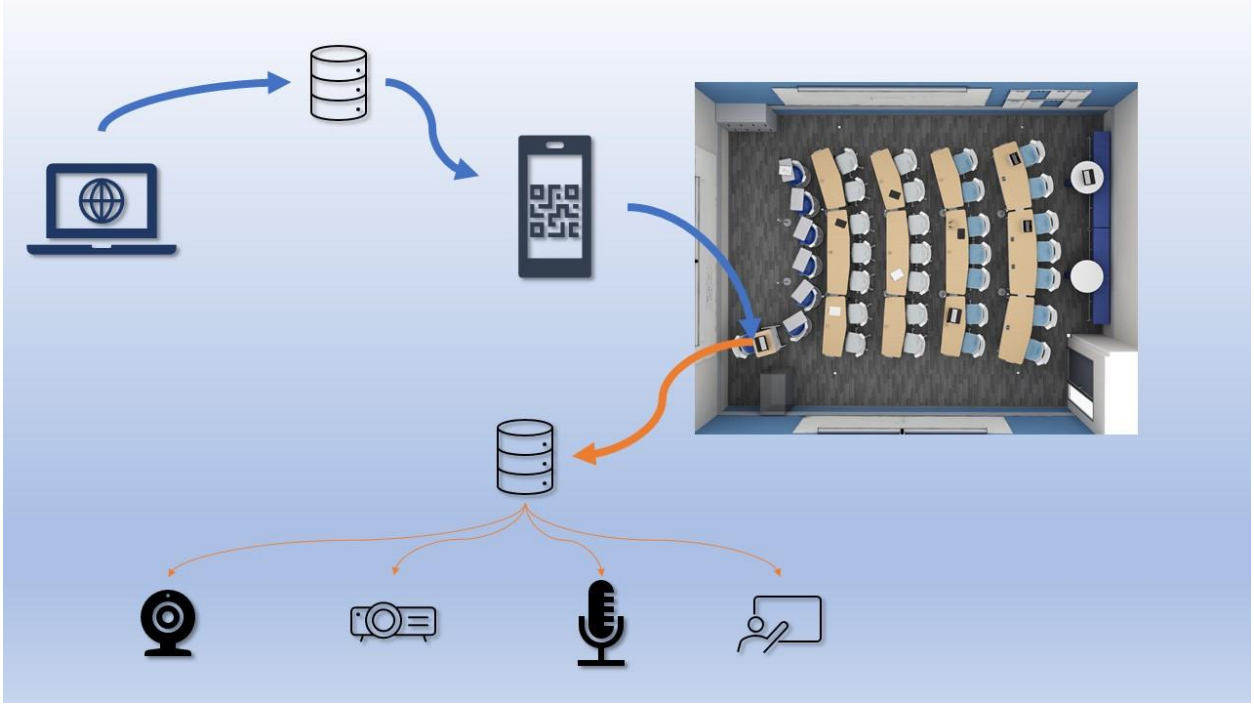

Figure 2: Smart Classroom Layout

For this iteration of Smart Classroom, our intention is to develop the base of what would become a bigger implementation of this strategy. The base we aimed to commit

ourselves to was to have a database driven web application, one that would take a professors settings and be able to display those as well as be able to have a professor edit and delete those settings as shown in the image of Figure 3.

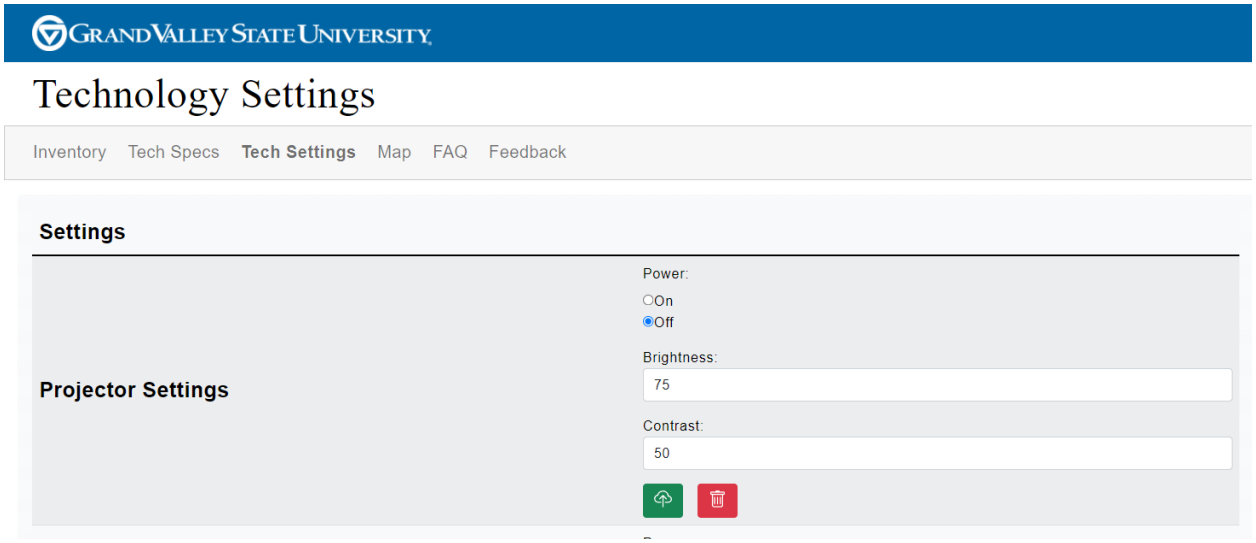

Figure 3: Technology Settings

Vue.js

The Smart Classroom front end was created using the Vue framework. Vue is a commonly used tool to develop user interfaces for web applications. Vue has been able to handle the requirements of this project, while also allowing for flexibility in areas of displaying data queried from a database. The core library is focused on the view layer only, and is easy to pick up and integrate with other libraries or existing projects. We used Vue to easily create a user interface that is both familiar and reactive. Vue allows for adaptability with other plugins and tools, which made it easy to work with as we needed to add functionality to the web application.

The Database

MongoDB Atlas is a free and cloud based database that has a small learning curve, but became very easy to use. Because of this, it was our preferred database. MongoDB Atlas allows for large amounts of data to be stored. This data included the technology that each smart classroom housed, data on buildings at the university, classrooms at the university, and each user's specific settings. The diagram for this database is as shown below (Figure 4).

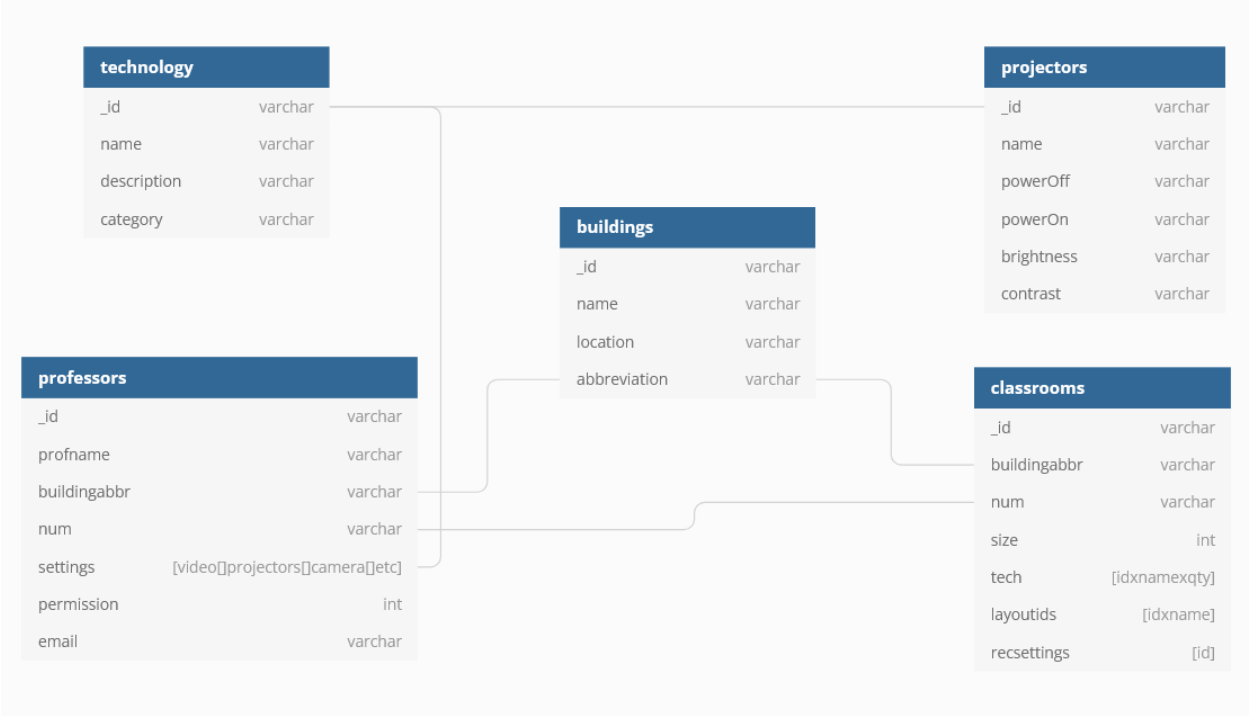

Figure 4: Database Diagram

To add functionality to our web application in using this database, we used a combination of express and MongoDB's library to query for the data in our database. Express was used to route our call properly to the correct query. MongoDB's library allows us to request certain data using the MongoDB query language. An example of one of these queries is shown in Figure 5.

```
app.patch('/patch/classrooms/:id', (req, res) => {
    var updateObject = req.body;
    var id = req.params.id;
    console.log(req.body);
    // MongoDB Specific Query
    roomcollection.updateOne({ _id: ObjectId(id) }, { $set: updateObject });
    res.send("Updated");
});
```

Figure 5: Database Query

The route in this particular example is "/patch/classrooms/:id", when given a specific id, the query will look for data that corresponds to that id. The query itself is presented in "`roomcollection.updateOne({ _id: ObjectId(id) }, { $set: updateObject });`". This will update the classroom collection in the database with what is passed in the body of the request. Other requests, such as `delete` and `put` are handled the same way, but will instead delete data and create data.

Technology Settings

In order to lay the foundation for future implementation, the technology settings are easily accessible from the database using packages called ExpressJS, Axios, and MongoDB NodeJS Driver These personalized settings are only available to users that have been authenticated by Google OAuth 2.0 and have been given specific permission by administrators. In other words, not every user will be able to see or change their technology settings, even though they have access to the inventory database and the rest of the application. An example of this is shown in Figure 6 and Figure 7.

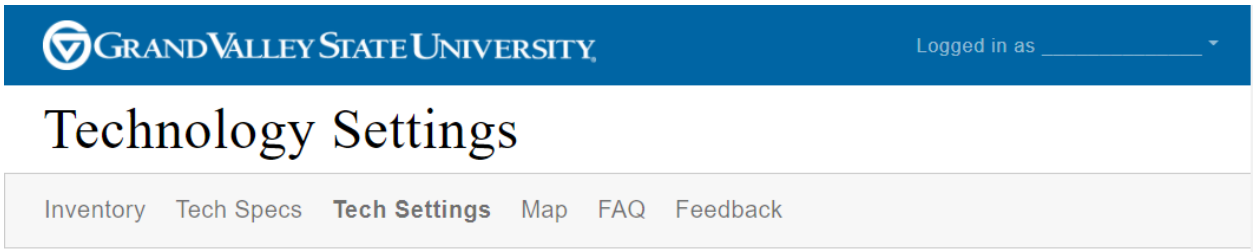

Figure 6: Logged in - Tech Settings

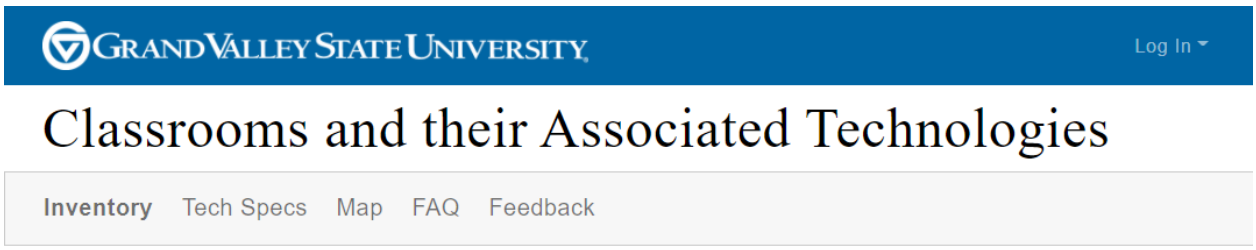

Figure 7: Logged Out - Tech Settings Removed

Personal Preferences

Personal preferences are an integral component of the Smart Start project. As such, the capability to change and apply settings easily and without frustration is our primary directive. We've implemented CRUD operations (Create, Read, Update, and Delete) with easy to understand icons and included pop-up messages called "toasts" to help the user understand the outcomes of their actions. Additionally, we've included logic to handle future cases where the user enters a room and would like current settings to be applied to the classroom's technology. This is all seen through the Technology Settings page and also shown in Figure 3.

Classroom Inventory Database

In addition to having settings that would allow a user to manipulate technology in a classroom. The web application also houses data detailing classrooms that have smart technology. This allows the administrators of the project to keep track of technology located in each classroom. Links to products, technical specifications, location, classroom size, and layout customization are also included. This is shown through the Inventory page of the web application and is shown in Figure 8.

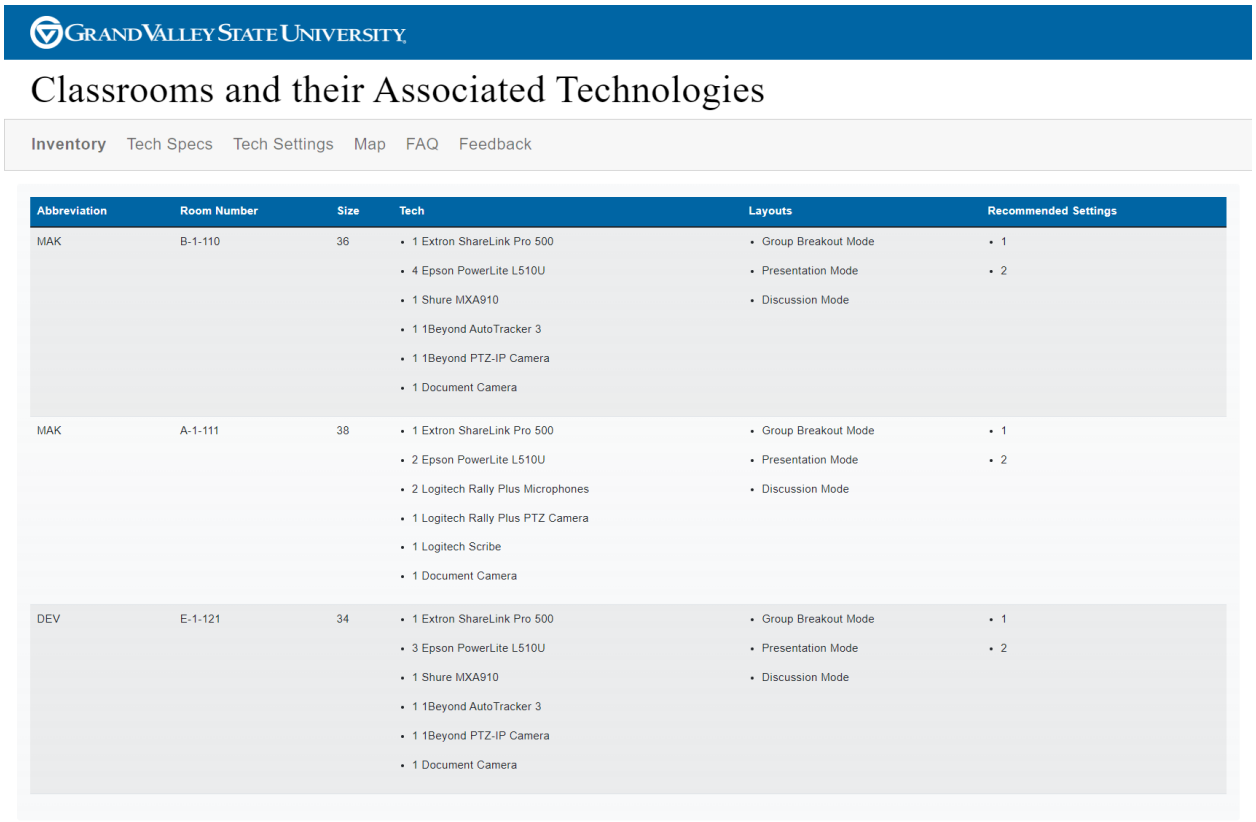

Figure 8: Classroom Inventory

Areas of Technical Growth

David Powers

MongoDB was the major component of my research and development in this project. Creating, developing and maintaining this database gave me the opportunity to learn the integration of this cloud based database system into a web application environment. I had only heard about MongoDB in previous years as a great tool for database management but had never had the chance to utilize its features.

I had only touched on web applications in previous semesters of college, we didn't have the chance at implementing an application that insisted on this much data to be driving it's interface. This was a first for me, and created a great learning experience.

Using tools such as bootstrap was also a first, as well as working under the Vue framework. These pushed development faster as it became easier once the process was learned. The tools that were used will be used again by me in the future.

Noah Mutersbaugh

My intention going into this capstone project was to fine-tune my skills with Vue.js and reactive web app development. I found several new packages that helped push the project forward and I learned new ways of presenting documentation and code to other developers to help bring everyone up to speed on the additions. I feel more confident in my abilities to design a web application from top to bottom and to present a product to others.

I had not worked with MongoDB before this project and David did a great job of helping me understand certain functionalities of it. I had the opportunity to perform my own calls to the database, organize data in a way that made sense, and discuss other possible solutions to explore what MongoDB is capable of. As an aspiring data analyst, the use of databases in my own personal projects is only going to grow, and I believe that this introduction to MongoDB will help me in this endeavor.

Mohammad Saleh

TECHNICAL GROWTH NEEDED HERE

Software Engineering Code of Ethics and Professional Practice

1.02 Moderate the interests of the software engineer, the employer, the client and the users with the public good.

During our development process, we worked with each other and the client to provide the best solution that would benefit the most people/users using our web application.

1.06 Be fair and avoid deception in all statements, particularly public ones, concerning software or related documents, methods and tools.

We had tried to be completely upfront about where we were heading with the project and how far we could take the project given the limitations we encountered.

2.01 Provide service in their areas of competence, being honest and forthright about any limitations of their experience and education.

As a team, each had inexperience in areas the other members were unaware about. This was quickly remedied at the beginning of the project, as we explained where our missing knowledge lay.

2.03 Use the property of a client or employer only in ways properly authorized, and with the client's or employer's knowledge and consent.

We had used assets that were given to us by our clients and had used them in accordance to their strict protocols for utilization.

2.06 Identify, document, collect evidence and report to the client or the employer promptly if, in their opinion, a project is likely to fail, to prove too expensive, to violate intellectual property law, or otherwise be problematic.

When we hit sprint 4, as a team we contacted our client to let them know that because of some of the limitations of the project we were unable to get the full extent of the project completed. We talked earlier on about the stages of the project and made mention of where we would end up as a final project.

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

We worked hard to reach the goals we set out for ourselves. The final project goal was not met, but sprint goals were made and completed throughout the project. Additionally, the project goals were adjusted appropriately as complications arose.

5.03 Ensure that software engineers know the employer's policies and procedures for protecting passwords, files and information that is confidential to the employer or confidential to others.

This is two-fold. We worked with the security and IT team at GVSU to understand what limitations we would have with this application, and it appeared there were many. We also worked to incorporate Google OAuth which helps protect those emails and passwords that would be used to change settings in our web application.

8.04 Improve their understanding of the software and related documents on which they work and of the environment in which they will be used.

Each individual of the team worked hard to understand their faults and where their limitations were. We each strived to learn more about the software we were creating and worked with each other to help the other understand tools that were being used.

Teamwork Reflection

Throughout the week we had meetings if something was close to crossing the finish line, such as sprints and documentation. We also had meetings if someone didn't quite understand something about the portion of the project we were working on. This helped clear up any misunderstandings and helped push the project forward. We also started out having weekly meetings with the client, this led to bi-weekly meetings, and then to emails. David focused primarily on developing and utilizing the database and its corresponding front ends (UI) he also helped with a small portion of Google OAuth and setting up the back end structure of the web application. Noah focused on client communication, documentation, and front end development as well as implementing the web application structure and Google OAuth. Mohammad worked on learning the process of Vue and web application development and utilizing this knowledge with implementing some front end components. As a team we were able to combine these focused efforts into the final product.

Conclusion

Although we didn't get the chance to actually implement changes in a physical classroom this semester, we did manage to reach several goals that were set by the client as we got deeper into the project. The primary goal was to have a web application that was controlled by a backend database. For this concept, we used MongoDB, but this could switch to a local database when the time comes to hand off the project. As we came to the end of the semester, our final goal was to prepare documentation and bring the software to a place that could easily be handed off to a future group as the teams at GVSU become ready to further develop the "Smart Classroom".

As a team, we were pushed forward as each member of the team brought their own assets. This led to the team's success in delivering the final product. The knowledge gained through our internships and knowledge we gained in database, web application development and general programming allowed us to reach our end goal. Utilizing the project management tools through Zenhub helped keep our ideas focused as we entered each sprint. Although we think we could have used this tool more often as we worked through the sprints, we recognize how useful this can be in the future for personal and professional projects. Since this is the first iteration of a multi-step project, we're certain that our contribution and conversations with several faculty members around GVSU IT teams will allow future development teams to receive the necessary technology and network security permissions to finalize the implementation of the application.

Future Work

Integration to physical classrooms will be the next phase of this project. This will involve explicit network permission from GVSU's IT department as well as in-depth communication with specific IT team members most familiar with the Extron unit, which is the central hub for technology within the classrooms. Our application will serve as a solid foundation for future development and can be easily altered to include communication between the application and technology within the classroom.