Technological Feasibility Analysis

11/8/2019

DigiTool Inc.

Project Sponsor: Xi Zhou

Team Faculty Mentor: Fabio Santos

Team Members

Traybourne North
Noah Baxter
Spencer Clark
Langqing Dai
Miguel Quinones

Table of Contents

1.0 Introduction

1.1 Project Overview

The Digital Logic course is one of the most fundamental courses an engineering student can take, whether they are in the Electrical Engineering program, the Computer Science program, or even the Mechanical Engineering program. This course teaches students to be able to work with basic logic reasoning tools, such as Karnaugh maps, truth tables, and switching functions. In the past years, students have had issues adjusting to these new concepts and so a Java applet was created for students to use for studying purposes. The problem is that this applet is no longer functional.

1.2 Problem Statement

The applet that students were using is no longer functional due to modern browsers not supporting Java applets, which is the basis of the problem that our project sponsor has. Our sponsor is Dr. Xi Zhou who is an Assistant Professor of Practice here at NAU. He instructs the class that this project will support in the future. In addition to the aforementioned problem, we have a couple of other problems we will need to address:

- The original applet did not provide support for the majority of the topics of the course, instead only covering a few. Because of this, students could only practice some of the course topics and if they had issues with any topics not covered in the applet, they were unable to do anything about it.

- Another issue with the original applet was that a scoring system was used to grade students on their work done in the applet, but this system was largely arbitrary, meaning that scores were seemingly random and didn't always correlate with the answers a student got.

- The new application will need to support a host of new requested features, such as additional learning modules for students, a dashboard that will show all the learning modules, and more as requested by our client.

1.3 Team Information and Solution

DigiTool Inc., comprised of team leader Traybourne North and team members Noah Baxter, Spencer Clark, Langqing Dai, and Miguel Quinones, along with team mentor Fabio Santos, was tasked by Dr. Zhou to revamp the original Java applet. He wants to make the applet usable again and he also wants to add new features to it, so that students taking Digital Logic may once again use this tool to assist themselves with studying and practicing the topics of the course. We have a number of solutions in mind:

- Our first solution to this problem is to rewrite the original applet in TypeScript, as well as using the relevant frameworks and other tools, so that it can function as a web application, which will make it usable to students again.

- We also plan to add more modules to the project that will cover other topics within Digital Logic at our project sponsor's discretion.

- Finally, we will add the new features that were requested by our sponsor, such as a way for users to create accounts and save their progress, provide a dashboard that will show a user's progress in the modules, and allow the application to be accessed across multiple platforms.

1.4 Paper Overview

This technological feasibility analysis will begin by looking at the technological challenges that we expect to encounter with this project, such as any frameworks, databases, languages, etc. that we might use. Afterwards will be the Technology Analysis section, which will cover in depth the issues we expect to have, our possible approaches and solutions to those issues, the metrics we will use to evaluate our solutions, and our explanations for the solutions we move forward with. We will then explore how all solutions will combine and work together in the Technology Integration section, which will focus more on how the technology we decide to use will all coexist with one another. Finally, we will conclude this document by reviewing the technology we will be using, stating our confidence in our solutions, and finishing off by saying how our project will address our sponsor's problem.

2.0 Technological Challenges

Below are all the various technologies and challenges we need to evaluate to deliver solutions to our customers. For each technology, priority-related challenges are listed, the most important of which is the reduction in demand.

2.1 Programming and scripting languages

The language we decide to use moving forward is undoubtedly one of the most important aspects of our project. Because our client wants the final product to be a web application, we would want to use a language that is suited for web-based programming. As such, there are a few factors we need to take a look at before we decide what language to use:

- Ease of Use
    - We would want to use a language that most members in the group, if not all members are familiar with. The language should be easy to learn if necessary and easy to work in as it will be the backbone of our project. The language lastly has to fit our project needs in order to ensure that few to no problems occur.
- Maintainability
    - The language we move forward with needs to be easy to maintain. We don't want to use a language that might be difficult to debug, nor should it hard to read.
- Scalability
    - The language we choose should be able to handle an ever-increasing load of resources. Because we are rewriting an existing project and adding onto it, scalability is an important factor for us.

2.2 Database Technologies

Our project is a learning system, so the account information of both students and teachers is necessary, as well as the internal information of the system, which requires database support. The problem would primarily be the database communication between the database and our web application. We want the database to validate user login information and we also want the database to load a user's game state progress so that a user resumes from where they left off if they were to log off a computer and log back on later in the day to resume their progress. A couple challenges consist of these factors:

- Security
    - The database must ensure that important information of the user is not leaked.
- Performance
    - Although we do not want this web server to bear a large load, it must at least ensure that there won't be a lot of problems within one semester to affect the actual use. Therefore, any database selected must be able to scale and process performance far beyond expectations.

2.3 Back-end communication

Since we are using a database, we will need a way to communicate with it. Most frameworks do not have a built in system for using a database and because of this, we will need some kind of library or package to do it. Thankfully TypeScript has at least one to pick from for many popular databases. We will be focusing on:

- Reliability
    - We must be able to ensure our requests to the database always reach their destination and do not get lost. What we mean by this is that information passed such as usernames and passwords must be able to get sent to the database and back successfully. The same rules apply for our web application. While some options have their own error checking, fewer errors overall is better.

- Features
    - We do not need a lot, but having diverse features will prevent us from having to manipulate the ones provided to get what we need and will make the overall process much cleaner. Having more features also enable us to produce at a higher rate since we wouldn't have to use other softwares / tools to produce what we need for our project.

2.4 Version control

Working together synchronously on code as a team can be a difficult challenge to overcome. We know that using Git will be our number one priority, but finding the right version control system is our goal. There are a couple factors that our team has to keep in mind when finding the right version control system such as:

- Productivity
- Governance

Productivity includes how fast a developer can pull, update, and sync changes out of a remote repository or how fast a developer can commit or push changes back to that remote server. It also includes code review which is important for other developers and programmers. This ensures that all merges of feature branches are well developed, tested, and desired.

Governance represents read and write permissions throughout a repositories. This ensures that permissions can be bound securely within our repository so that no one outside of our group can modify the contents of our project.

2.5 Front-end libraries / frameworks

Since our solution will include both front-end and back-end communication, we need a robust web framework as the foundation for web services and other technologies. The framework we choose for this project will determine which challenges can be effectively addressed. Below are some challenges and needs of the web framework:

- Security
    - Sensitive data will be archived and protected by our systems and framework, since our project involves a lot of user information.

- Scalability
    - Our customers want this to be a developmental, continuously optimized project. Many of the data and user feedback inside the system need to be updated in a timely manner.
- Interface with other technologies (DB, API, library, all other technologies mentioned)
    - Application friendly, built-in templates and deployable.
    - The ideal framework will have native database management in the front end.
- Language
    - We all agree that this project should be reflected in the form of a web page, so finding the right language will be a challenge.

2.6 Tools and remote services

When it comes to tools and remote services, ease of use and reliability are the challenges that come to mind when applying them to our project. These tools consist of IDEs that will help with our project's development as next semester comes around.

We need tools that can easily be used to not only save time, but boost performance as well. On the other hand, we need tools that can be reliable to increase our project progression once we start working on its implementation. Remote services can also be another challenge since some services can be costly or difficult to set up for both the client and the developers that work for the client.

2.7 Modeling languages and schema definitions

Modeling languages and schema definitions aren't a challenge the project itself, but are a challenge that we will face during our design process. The modeling languages and schema definitions we choose will impact the blueprint of our project and whether or not that blueprint can be implemented or not.

If our team can't represent our modeling languages and schema definitions in an accurate way through visualization, the way we implement our project will be much more

difficult. For example, if we were to use a paper and pencil to represent what we wanted in our web application compared to using a website that can create professional diagrams, each member within our group would have a harder time understanding the flow of a modeling language or schema definition. Because of this we would have to consider this factor when selecting or modeling languages and schema definitions.

- Reusability
    - The ability to reuse a blueprint once it's been completed through importing. This is for future redesign purposes in case we need to redesign our blueprint or if our client wants to add new features to our blueprint as we progress in our project .

3.0 Technology Analysis

After figuring out what technological challenges our project could potentially face, it was now time to analyze each element involved in our project.

3.1 Programming and scripting languages

Several options were discussed when choosing a preferred language for this project. Since the original project was written in Java, we decided to find languages that were very similar to Java, that way it could be easier for us to convert it when it was time to implement the project. Software qualities such as ease of use, maintainability, and scalability were also looked at when choosing our programming and scripting languages. Depending on the number of students in each class that our client teaches, many users could be accessing our application at the same time, which is why scalability is very important when it comes to our project.

HTML and CSS were some of the scripting languages we decided to use since our project needed to run on a website. HTML ensures that the browser our project is displayed on could properly display text or load images and CSS allows the appearance of our website to look detailed as we continue to implement the project.

Python was one of the options to decide from as a programming language. It's known to contain extensive support libraries which could help with web development and it also has incredible integration features that make a web application usable on basically all operating systems. On the other hand, Python is dynamically typed so design restrictions could possibly occur which could require more testing time. It also has undeveloped database layers which is another negative because our project needs a fully

developed and secure database [1]. We'll need a secure database to hold student / faculty usernames and passwords along with each person's individual progress towards module completion. This allows a user to resume where they left off if they were to log off and log back on.

The next language we looked at was JavaScript. It's simple, versatile, client - side based, and most importantly, it's compatible with a wide variety of programming languages. One thing that we are aware of is the debugging process within JavaScript. A single code error can prevent the whole web application from running, so it's important to test code frequently in small pieces, so developers can find errors quicker. Another thing to understand is security in JavaScript. Since JavaScript is a client side language there could be issues with user validation simply because it takes place on a web browser, rather than a server [2].

Finally, we looked at TypeScript as another potential programming language to use. TypeScript is a superset of JavaScript, and it has the same features as JavaScript in addition to a few features of its own. The most significant benefit it has over JavaScript is the fact that it is statically typed as opposed to dynamically typed. TypeScript itself is compiled into plain JavaScript on compilation, so it will also work in all browsers. Its community and documentation allow it to be understood by anyone with a programming background and it also provides a plethora of tools that help provide code analysis and ease of use toward a developer. One of the only downsides to TypeScript is that it does not have a package manager of its own; it instead utilizes npm. Aside from that, TypeScript is essentially an easier-to-use JavaScript because it offers easier maintainability and readability. Since it's a superset of JavaScript itself, TypeScript comes with the benefits that would already be provided by JavaScript alone [3].

Figure 1

	Python	TypeScript	JavaScript
Ease of Use	✓	✓	✓
Maintainability	✓	✓	✓
Scalability	✓	✓	✓
Choice	X	✓	X

Programming language comparison

Based on the results from Figure 1 and the needs of our project, we decided to choose TypeScript as our programming language. A key benefit towards using

TypeScript is its ability to prevent certain runtime errors while running web applications because of its type-safe error handling [4]. All three languages can be usable towards a user, maintainable, and scalable depending on how we go about our design process [5]. However, we are aware that issues could still arise when switching languages, but it should be easy to resolve. Another alternative solution would be to use JavaScript if for some reason, we run into issues with TypeScript, since its coding structure is similar to TypeScript. This makes it easier to convert TypeScript into JavaScript if any issues arise from using TypeScript. In the future, a demo of our project can be created that utilizes TypeScript to show that it will work in all browsers as a web application, which is the main goal of our project.

3.2 Database Technologies

When it comes down to database technologies, there are two types of databases to choose from. Once that has been decided, we can then narrow it down to which database could best meet our project's goal.

A database can either be a relational or non-relational database. Relational databases can be used to keep data structured and unchanging. Non-relational databases can be used to store large volumes of data and they can be used for rapid project development. They're also good for cloud computing because non-relational databases can access files and use applications from any device that has Internet access. When analyzing both types of databases we have to consider what our project would need to store inside of a database. Usernames, passwords, user game states, and possibly questions created by our client are the only things our database should need to hold. We also have to think of the qualities needed for our database such as performance and security, which is vital to optimize our database [6].

Figure 2

	Relational Databases	Non - Relational Databases
Performance	✓	✓
Security	✓	X
Choice	✓	X

Database comparison

Based on the results from Figure 2, a relational database seems like a big problem saver that could impact our project in a very positive way. Security is a key factor in our project that needs to be executed to its highest level so that sensitive information can be kept in a private and safe location. Although non - relational databases can perform at high rates since data can all be stored in one place, it does contain a weak authentication security feature which affects the overall integrity of a database. This is heavily needed in order to secure all sensitive information within our database. The next thing to do is to choose a relational database management system which consists of MySQL or Microsoft SQL Server.

MySQL is a flexible and ideal relational database management system for data security. It can perform at a high rate and its able to scale at a high rate. Data acquired from this project won't be anything too advanced so MySQL seems to be a relational database management system to lean towards. But we have to look at some of the negatives of using MySQL, which consists of stability and functionality. Developers have been known to have stability issues when dealing with large amount of data and transactions. Also, some functionality traits of MySQL may need additional add-ons to be installed into MySQL [7].

Microsoft SQL Server offers a lot of functionality when it comes to being a relational database management system. The big plus side to using Microsoft SQL Server is the fact that it can work with other databases such as Oracle or DB2. A big downside happens to be that a developer has to pay in order to get support for the free version of Microsoft SQL Server. Things to consider would be security, functionality, scalability, and support [8].

Figure 3

	MySQL	Microsoft SQL Server
Security	✓	✓
Functionality	✓	✓
Scalability	✓	✓
Support	✓	X
Choice	✓	X

Relational Database comparison

Based on the results from Figure 3, MySQL seems to be the most logical decision to go along with our project implementation. MySQL offers full support though documentation and tips when using MySQL, while Microsoft SQL Server only offers full support if you pay a fee. On the other hand, it does lack security, but lacks it only when it comes to making transactions. Since our project involves no type of transactions between the user and another entity, there's no need to worry about security being an issue. MySQL can scale and support many users accessing our web application at the same time so there shouldn't be a problem with managing data collected within our database. In terms of functionality, extra time may be needed to install add-ons as we work on the project but besides that, everything about our project should integrate well with each other.

3.3 Back-end communication

Since we are using a database, we will need a way to communicate with it. We are not doing anything complex or time dependent, since the requirements for this are very low. We only need our back-end communication to be reliable and easy to use without being too sparse on features.

Through npm we have a few packages to choose from: mySQL-native, mySQL, and mySQL2. All of them are very similar and are more or less based off of each other. They all offer database connection, request creations, and pipelined queries for queueing and handling requests. MySQL-native has an API that could use a bit of work on readability [9]. MySQL adds on by allowing connection pools, more specific requests, joins, timeouts, and error handling [10]. It also has a much nicer to look to the API. Furthermore, mySQL2 adds onto mySQL to be faster with better performance, logs, a promise wrapper, compression, and secure socket layers [11]. MySQL2 looks very promising other than that it can be finicky with linux systems.

Figure 4

	mySQL-native	mySQL	mySQL2
Reliability	✓	✓	X
Features	X	✓	✓
Choice	X	✓	X

Back-end communication comparison

Based on the results from Figure 4, mySQL fits what our project database needs. Reliability is more important than some of the extra features mySQL2 has over mySQL such as prepared statements and extended support for encoding and collation. MySQL has an incredible API and a good amount of features to help with the development of our project.

3.4 Version Control

When choosing version control, we analyzed factors that consisted of productivity and governance. We then looked at BitBucket and GitHub and compared the two to see which version control would best fit our project needs.

One of the good things about BitBucket is that developers can simultaneously use code with integrated comments. You can also make requests, manage, and share your Git repositories. Another pro when it comes to BitBucket is the ability to easily gain access to private or public repositories. A bad thing about BitBucket is that it does cost money as development teams pass a team count of 5. This could affect future development towards our project if it were to be modified in the future. A con when it comes to BitBucket is its stability [12]. This can reduce productivity rates if a development team consistently encounters problems with work sessions [13].

What's nice about GitHub is its documentation that allows productivity to be very efficient. It provides developers with the content and support to help them share and manage code snippets at a fast rate. It also allows easy collaboration between developers by providing an online Git. This allows new users to not have to set anything up in order to get GitHub to function properly. A drawback could be the lack of security. If you were to do a project with very sensitive information, it has the potential to be misused unless your repository is private [14].

Figure 5

	BitBucket	GitHub
Productivity	X	✓
Governance	✓	✓
Choice	X	✓

Version control comparison

Based on the results from Figure 5 (based on a scale of 1 - 10), we chose GitHub because it has one of the largest communities when it comes to version control and it is

very easy to use. Also, GitHub makes it easier to collaborate with each team member and it allows us to look at previous versions of our work to check our progress as we develop our web application. What makes GitHub unlike other open-source repositories such as BitBucket, is the fact that GitHub has an extremely high storage limit (100GB cap) offered to free users. It also has a desktop client which makes it easier for developers to upload and edit pieces of code.This can be useful for us since our repository will be stored on GitHub.

3.5 Front-end libraries / frameworks

Choosing front-end libraries / frameworks to develop code can save our team a lot of time and it can also make programming easier. We judged the front-end libraries and frameworks that we're going to use based off of complexity and maintainability.

The frameworks / libraries that we chose from was AngularJS, ReactJS, and Vue.js. AngularJS is built with two-way data binding which reduces time in the development of code. It also has a large community to keep the framework up to date and it has numerous dependencies which define how pieces of code interact with each other. One thing that could be a problem would be the complexity of learning AngularJS in the short amount of time we have to develop this project. This could reduce the time we have to actually implement our project and it could create unnecessary stress towards the team. Performance is also a problem depending on how dynamic our web application plans to be [15].

ReactJS is an open-source JavaScript library and unlike AngularJS, it's easy to use and learn. It does have two-way data binding just like AngularJS, but instead comes with a large amount of reusable components that can be nested with other components to allow complex applications to be created just by using small building blocks. Code is easy to debug and its performance is extremely high since it has virtual DOM which enables a smoother and faster performance. One thing to be aware of is the poor documentation that comes with using ReactJS. ReactJS constantly changes and updates development which leaves no time to create proper documentation. Another thing to be aware of would be that other development tools may be needed to get a project to fully be functional. This all depends on the complexity of the project and what the project entails. ReactJS only handles the UI layers of an application and nothing else so choosing this library all depends on the needs of a project [16].

Vue.js is another framework that has incredible documentation and features. It also allows data binding between HTML and JavaScript so that coding can be easier toward a developer. What makes Vue.js different from AngularJS and ReactJS is its flexibility and TypeScript support. Each update comes with new features and components that improve the state of the framework to allow simpler and easier coding. Vue.js does

have a small community which can make finding solutions difficult, since resources are limited. Another negative can be the fact that Vue.js is considered to be over-flexible. This means that as more developers get involved with a project, an increase in problems can occur because over-flexibility can lead to the over-complication of a project [17].

Figure 6

	AngularJS	ReactJS	Vue.js
Complex To Learn	✓	X	X
Maintainability	X	✓	✓
Choice	X	✓	X

Framework comparison

Based on the results from Figure 6, ReactJS seems like the library / framework that would best fit what our project needs. Since ReactJS can combine HTML and JavaScript for rendering, this can make things a lot easier for us as developers. The fact that ReactJS isn't difficult to learn can make things flow a lot smoother in our project development and implementation. In terms of dependency injections, ReactJS offers modules that can be installed to help with performance and efficiency in our web application. As we develop our web application, testing / debugging can be easily done through ReactJS. If we do experience problems while using ReactJS, it does have the ability to integrate with other frameworks / libraries such as AngularJS or Vue.js. This can be an alternative solution to any problems we may experience while working on this project.

3.6 Tools and remote services

Tools and remote services will be needed as a platform to help develop code as well as test it. This consists of using an NAU server to help test our website frequently, IDEs to help develop basic web code, and other libraries that increase production and the details of our project.

We plan on using a Linux / Dana server to test our web application. This is offered through NAU toward NAU students. This plays an important role since our project is designed for an NAU course offered on campus. The Linux / Dana server is fast, reliable, offers the best testing environment that matches where our website will be used most of the time. One problem could be that students taking the course would have

to learn how to connect to a Linux / Dana server in order to access it. An alternative solution could be to find another server, but this could lead to the project sponsor spending money for a website server. A better solution would be to create a short tutorial that shows students how to access our web application through the Linux / Dana server.

Brackets, Visual Studio Code, and Eclipse are other tools that our team can use to help develop our web application. Both can be used as a source code editor to develop and test our website in real time. Brackets, unlike Eclipse, constantly changes the live preview of a web application as a developer programs, which can reduce the time needed to test your website over and over again [18]. Visual Studio Code allows a developer to develop code in almost any language and offers a variety of plug-ins to assist with debugging, editing, and compiling [19]. Eclipse on the other hand, offers a plugin that allows you to edit JavaScript / TypeScript files and check for syntax errors as well as write in other languages besides just HTML, CSS, or JavaScript / TypeScript [20].

As a team, we decided to use Brackets and Visual Studio Code as tools to develop our project. We chose these tools because it offers features that are most beneficial to our productivity such as autocompletion, compilation within the application, and plug-in features. Visual Studio Code would be used for writing our code in TypeScript / JavaScript and Brackets would be used to constantly run our code with it's live preview feature. This saves our team a lot of time and it will help us get a lot of things done when it comes to developing our cutting edge web application. If code problems start to occur in either Brackets or Visual Studio Code, Eclipse can be used as an alternative solution to prevent code issues since it offers similar features compared to Brackets and Visual Studio Code. The challenges affect our productivity towards the projects, but don't affect the project implementation itself .

3.7 Modeling languages or schema definitions

The way we model languages or define schemas can be done in a variety of ways. The best way to represent these topics would be through drawing tools that can create a blueprint of how our project could be designed and implemented.

The best drawing tool to model the programming aspect of this project would have to be draw.io. It's a very common tool used to make flowcharts, process diagrams, org charts, UML, ER and network diagrams which can help significantly with the foundation and structure of how our project should be designed [21]. Since our project is a web application and involves software development, a UML diagram would best depict how everything will function synchronously.

When it comes to schema definitions, ERDPlus would be perfect for creating entity relationship diagrams, relational schemas, star schemas, and SQL DDL statements [22]. ERDPlus plays a vital role when creating the database blueprint for how data will be

stored and managed within our database. How we design modeling languages and schema definitions rely heavily on what the client wants in his project. Using both of these tools will help the flow of development run smoother since our team can base our project implementation off of the blueprints we create which consists of the database and web application.

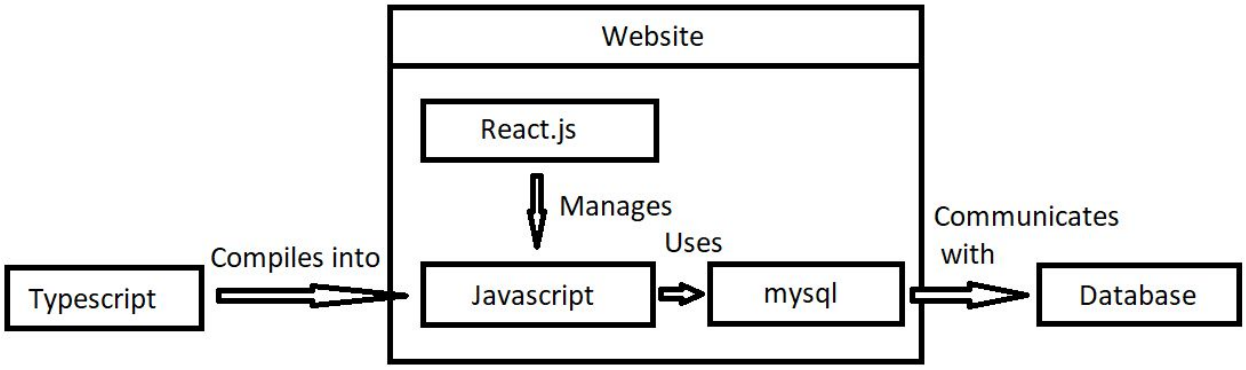
All in all, modeling languages and schema definitions aren't a challenge to the project itself, but are a challenge toward the project design. Using these websites to help design our project blueprints will help a lot with its productivity. This is because we can use the blueprints as a guide to match project implementations with its corresponding blueprint.

4.0 Technology Integration

With the current solution laid out, we do not have too many parts. We only have one language to work with, one framework, one library, and one database. Our framework and library is already written in Javascript. With no overlapping parts there should be very few situations where things don't "play nice" with each other. Since Javascript executes in the browser and we do not expect our app to be resource intensive, we do not have concerns for what system is running our code.

Our envisioned layout for the system is shown in Figure 7. As stated, there are no parts that are overly complex or redundant. The process is straight forward, and there are no foreseeable issues at the given moment.

Figure 7



Integration diagram

We will be writing all of our code in TypeScript. TypeScript compiles into Javascript so we will not need to worry about the browsers ability to execute the code.

We will however, need to add types for ReactJS and MySQL in order for the editor and compiler to not complain. As TypeScript is relatively popular, this has already been done for many libraries and frameworks, ReactJS and MySQL included. All we have to do is use npm with the command “npm i @types/react” for ReactJS, and similar for MySQL [23]. ReactJS will be the framework and will manage the code we make in TypeScript to provide us with a user interface and general flow of the program. Our TypeScript code will handle the program’s logic and use MySQL’s functions for communication with the database.

5.0 Conclusion

The problem we are trying to solve consists of taking an old Java applet and updating it into a usable web application. This cutting edge web application will help first year students enrolled in the intro to digital course understand hard to learn concepts such as Karnaugh maps, truth tables, and switching functions at a faster rate. In this document we analyzed several challenges we could face when updating this applet. We also analyzed several challenges that could potentially interfere with what our client wants. Some of the topics that were looked into consisted of what language to use, what database to use, front and back end technologies, etc.

After evaluating several different options in a variety of categories, our team has come to the following conclusions. The programming and scripting language we chose to use was TypeScript. The reason for this decision is our team feels that it will be easier to maintain in the future so that our program will remain usable. It also offers a higher level of scalability in the future if the client wants to update the web application. For our database, we have chosen to use MySQL simply because it would be easier to scale in the future as well. It also offers other benefits when compared against other databases such as security and reliability. As for our version control system, we chose to use GitHub because it can maximize team productivity and governance. Lastly, for our front end library / framework we chose to use ReactJS, since it’s much more usable and maintainable than the other options we compared it to.

Our team then plans to integrate all our selected technologies into our final project by using TypeScript to allow our program to run in web browser without any of the current applets technical issues. We also plan to use ReactJS to help manage the TypeScript / JavaScript and MySQL for our backend database responsible for handling individual user accounts. DigiTool Inc. is incredibly confident that our choices will lead to us being able to fulfill our client’s requests. We are ready to provide Dr. Xi Zhou with an updated and usable product that will continuously aid in educating students for the years to come.

6.0 Reference Page

[1] Solutions, Mindfire. “Advantages and Disadvantages of Python Programming Language.” *Medium*, Medium, 24 Apr. 2017, medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121.

[2] Author, Unknown. “Why Developers like JavaScript.” *StackShare*, 2019, stackshare.io/javascript.

[3] Aux. “TypeScript: Pros and Cons.” *Medium*, Medium, 19 Oct. 2016, https://medium.com/@auxx/typescript-pros-and-cons-873529634099.

[4] Gio. “Type-Safe Error Handling In TypeScript.” *The DEV Community*, 5 May 1970, https://dev.to/_gdelgado/type-safe-error-handling-in-typescript-1p4n.

[5] Fleming, Ian. “ISO 9126 Software Quality Characteristics.” *ISO9126 - Software Quality Characteristics*, 1 Jan. 2009, http://www.sqa.net/iso9126.html.

[6] Wodehouse, Carey. “SQL vs. NoSQL: What's the Difference?” *Hiring Headquarters*, 1 Nov. 2019, https://www.upwork.com/hiring/data/sql-vs-nosql-databases-whats-the-difference/.

[7] Author, Unknown. “What Is MySQL? About MySQL RDBMS.” *Atlantic.Net*, https://www.atlantic.net/what-is-mysql/.

[8] “Home.” *TrustRadius*, 7 Nov. 2019, https://www.trustradius.com/products/sql-server/reviews.

[9] “Mysql-Native.” *Npm*, https://www.npmjs.com/package/mysql-native.

[10] “Mysql.” *Npm*, https://www.npmjs.com/package/mysql.

[11] “mysql2.” *Npm*, https://www.npmjs.com/package/mysql2.

[12] Fernández, Daniel, et al. “Bitbucket Reviews - Ratings, Pros & Cons, Analysis and More.” *GetApp*, 24 Sept. 2019, https://www.getapp.com/it-management-software/a/bitbucket/reviews/.

[13] Dev, Aditya, and Rashmi Vishwakarma. “BitBucket Advantages and Disadvantages.” *CareerHunt*, 9 Apr. 2018, http://careerhunt.net/introduction-to-bitbucket/.

[14] “GitHub Reviews and Pricing - 2019.” *Reviews and Pricing - 2019*, https://www.capterra.com/p/129067/GitHub/#reviews.

[15] Anon, (2019). [online] Available at: https://www.g2.com/products/angularjs/reviews [Accessed 7 Nov. 2019].

[16] “Pros and Cons of ReactJS - Javatpoint.” *Www.javatpoint.com*, https://www.javatpoint.com/pros-and-cons-of-react.

[17] “Comparison with Other Frameworks - Vue.js.” - *Vue.js*, https://vuejs.org/v2/guide/comparison.html.

[18] “A Modern, Open Source Code Editor That Understands Web Design.” *Brackets*, http://brackets.io/.

[19] “Visual Studio Code - Code Editing. Redefined.” *RSS*, Microsoft, 14 Apr. 2016, https://code.visualstudio.com/.

[20] Guindon, Christopher. “Eclipse Desktop & Web IDEs: The Eclipse Foundation.” *Eclipse Desktop & Web IDEs | The Eclipse Foundation*, https://www.eclipse.org/ide/.

[21] “Draw.io Diagrams - G Suite Marketplace.” *Google*, Google, https://gsuite.google.com/marketplace/app/drawio_diagrams/671128082532.

[22] Info@erdplus.com. “ERDPlus.” *ERDPlus*, https://erdplus.com/.

[23] “TS Types” *Npm*, https://www.npmjs.com/~types.