Technical Specifications

Broadway Builder

CECS 491A - Software Engineering
Vatanak Vong

December 13, 2018
The Fantastic Five
Lexzander Saplan 014177252 (Team Leader)
Bryce Moser 012446081
Abi Castro 017258252
Abdul Latif 016084768
Jose Ramirez 014411356

Contents

1 Hardware Requirements

Minimum Computer Specs

CPU	Intel Pentium G630 (2 cores)
RAM	2GB DDR3
Storage	64 GB
Network	Must be connected to the internet (WiFi / Ethernet)

Recommended Computer Specs

CPU	Intel i5 (4 cores 3.0+ GHz)
RAM	4GB DDR4
Storage	128 GB
Network	Must be connected to the internet (WiFi / Ethernet)

2 Software Requirements

Server Hosting / Deployment

Microsoft Azure	Amazon Web Services- Elastic Cloud Computing (EC2)	Serverless Server - AWS Lambda (Faas)
Pros • Offers a Service Level Agreement of 99.95% allowing high availability • Azure also advertises itself as a leader in IaaS security. Cons • Not good for new developers as it requires more extensive platform expertise than most other options	Pros • Storage can be added as needed • Pricing is very affordable since you pay only for the amount of storage you need. Cons • Limiting amount of data that can be stored due to storage. • Larger sized instances can become very expensive. • Instance specifications are not always available before purchasing.	Pros • Engineers no longer have to maintain the server since the responsibility will be of the Faas provider. • More time can be dedicated to developing and improving features • Lower operational and development costs Cons • Not best for long running process. • Third parties will control operations therefore you must play by those rules.

Recommendation: We recommend going with Amazon Web Services - Elastic Cloud Computing (EC2) because it has competitive pricing offering the minimal hardware we need. Offers include 1 vCPU and 0.5GB RAM for $4.23/month. AWS is also a very popular option for server hosting and we believe its positive reputation is something we can have faith in when making this choice.

Web Server

Internet Information Service (IIS 10)	NGINX v1.15.5	Apache v2.4.35
Pros • It integrates well with ASP.NET and MS SQL which are technologies we will recommend we use. • IIS has built in authentication options and other security options. • It allows for remote management so developers can manage from anywhere. Cons • It is limited to only being used on a Windows platform	Pros • NGINX is lightweight and fast at serving a large number of users by using an event-driven asynchronous architecture to handle requests. • It is scalable and works well with other web-servers which can extend its functionalities. Cons • NGINX does not have as many extensions and will require third party web servers. This can lead to complications with the application.	Pros • It can be installed easily and it is highly reliable. • It is maintained and updated on a regular bases. That being said, technical support is highly available. Cons • Has a higher learning curve making it not ideal for new developers, since it requires expertise of its modules and services

Recommendation: We recommend IIS due to its robust features. It includes many built-in security features necessary to develop and deploy a secure application. It also integrates nicely with the back-end application framework we are going to use.

Operating System

Windows 10	Mac OS v10.14 (Mojave)	Linux (Ubuntu 18.04)
Pros • Most widely used platform making new member assimilation easier • Can install many packages to enhance functionality Cons • Can be used on a multitude of different machines, causing compatibility issues. Requires more extensive debugging	Pros • Gives the user less configuration options, allowing for a more consistent environment to implement and debug for. • Most common environment used for front-end development Cons • Not able to upgrade hardware components by virtue of using an Apple product	Pros • Open-Source and free • Features bash shell which is one of the most commonly used tools for web development Cons • Might require more members to dual-boot Linux since it is least widely used

Recommendation: Windows 10 is recommended because most of the functionality the other 2 options provide are able to be done using Windows 10 using packages and software. It remains the most widely used Operating System and is deemed easier to use for a productivity and/or development computer.

Browser

Google Chrome (v71.0 and v69.0.3497)	Mozilla Firefox (v63.0 and v62.0)	Internet Explorer (v11.0 and v10.0)
Pros • Features Predictive Phishing Protection as an added layer of preventive security • Overall performance and load times are a little fast than other browsers Cons • Resource intensive by using a heavy portion of memory to run	Pros • Uses less memory resource than other browsers by limiting its content processes count to 4 at any given time • Better management of tabs and faster session restore Cons • Complies with W3C standard but apart from that, site compatibility is not supported well	Pros • Full hardware acceleration for text, graphics, and video • Notification when add-ons slow browser performance Cons • Slow startup and shutdown • Slow Javascript engine for worse performance

Recommendation: We recommend using Google Chrome as our web browser of choice as it has great security and overall best performance. Although it uses more memory than browsers like Mozilla Firefox and Internet Explorer, it handles web applications more smoothly and securely.

Back-End Framework

Express.js (v4.16.3 npm)	Flask (v1.0.2)	ASP.NET Web API (v2.2)
Pros • A very simple and fast framework, with one of the bigger libraries. • One of the earliest frameworks node.js frameworks to be born and has retained its popularity • Supports many third-party plugins. Cons • There does not seem to be any universal way or arranging things. Can become difficult to manage as it becomes more developed.	Pros • Extremely flexible framework that can scale into larger projects very well by adopting a modular design • Flask adopts a very simple and clean implementation style, making it incredibly lightweight without sacrificing much power. Cons • The library is not as big as other competitors. • Only includes basic features; limited overall. • Does not have any form of database or ORM.	Pros • The .NET framework provides a massive amount of extensibility and third-party components • Very naturally integrates with SQL server • A mainstream product with many developers working on it. Meaning plenty of support and documentation. Cons • Scalability of projects becomes a bit more costly when projects become very large.

Recommendation: Although ASP.NET Web Api is a younger technology we recommend it due to it being able to work on an IIS server and how development on it will be quick and smooth. This way we can progress quickly and smoothly. Testing is sophisticated and an important feature to have in order to make sure we present a good experience to the user from the start.

CSS Preprocessors

SCSS (v1.14.1 npm)	LESS (v3.8.1 npm)	Stylus (v0.54.5 npm)
Pros • CSS syntax friendly • It offers variables for whatever you want • It uses nested syntax • Ability to make mixins (or functions) for code reusability	Pros • Creation of variables and variable interpolation to be used in places such as selector names, property names, etc • Lazy Evaluation allows variables to not have to be declared before being used • You can define variable names using another variable	Pros • Closest thing to a full programming language for CSS • You can define functions without listing parameters
Cons • Overuse of nesting can create create problems such as performance issues and difficult maintenance • Requires Ruby to be installed	Cons • Maintenance can be difficult • Debugging is harder	Cons • Heavy reliance on whitespaces • Not under active development so there are lots of known bugs

Recommendation: We recommend using SCSS as it provides the use of variables and functions that allow for code re-usability. It also allows nested syntax which helps in styling during development. All CSS preprocessors have a difficult time keep up with maintenance, but we should be able to manage this more efficiently using SCSS.

CSS Framework

Bootstrap 4.1	Foundation 6	Bulma (v0.7.1 npm)
Pros • Free professional templates and plugins • Most recent version is now more modular Cons • HTML is non-compliant • End products tends to look similar to others who use it, unless actively avoided	Pros • Customizable user experience for different devices • Javascript utility libraries are publicly accessible Cons • Higher learning curve than its competitors • Less popular in general, thus less helpful community forums	Pros • Lower learning curve than its competitors • Fully responsive when implementing to make things easier Cons • Still in development phase

Recommendation: We recommend using Bulma (v0.7.1 npm) as the CSS Framework of choice. Although it is still in development phases, we consider it to be more powerful than its competitors because its low learning curve and high customizability are worth the trade-off. We also think because it is a newer framework, it will help distinguish our website from others, all the while being light-weight.

Front-End Languages

JavaScript (ES6)	TypeScript (v3.1.1)	PHP 7
Pros • Has an technical advantage when dealing with JSON • Considered to be used with Single Page Applications • Allows both client and server side runtime environments • Client-side is very fast as it can be run immediately within the client-side browser Cons • Interpretation can be different depending on the browser • Difficult to debug when developing a large project	Pros • Allows the use of a supported IDE • For a large project, it might result in a better robust software • Better for collaboration and allows for static typing • Provides Object Oriented Programming features Cons • Needs to be compiled first then provides the output which is an extra step • Cannot be easily manipulated by a Content Management System	Pros • Considered to be used with Content Management Systems • Easier maintenance as it guides you toward code that is loosely coupled and has little repetition • It has extensive database support such as with SQL Cons • Can work with JSON but is more situational than other languages • Slower execution

Recommendation: We have recommended using Typescript as there is better room for collaboration in large scale projects. In case we reach roadblocks the community of typescript allows us to reach out and get support. Although the reduction of bugs in the code is dependent on how developers use the language we are confident that our team will constantly test features and make sure it is working properly before moving on. We have the ability in using an IDE which will improve debugging and better warnings involving potential errors in code. Since Typescript Code completion and IntelliSense provides a better development experience than just base Javascript for a large scale project.

Back-End Language

Java (SE 11)	Python (cpython) (v3.7.0)	C# (7.3)
Pros • Gigantic community, as well as a significant amount of libraries and APIs • Excellent documentation to reference • Has multithreading, allowing switching between threads more efficiently	Pros • Dynamic enough to be utilized for simplistic programs to the most scaled, agile, and complex data operations • Cost efficient in resources as it takes considerably less time to build something than other languages • Easy scaling with complex applications	Pros • Integrates well with Windows such as having an IDE to work with • Object oriented programming language • Can read and write files if given the Silverlight version allows it
Cons • Can be a bit overkill for smaller projects • Will cost money in 2019 to get any further updates • Code is bloated and complex in comparison to languages like Python	Cons • Has limitations with database access • Not a good choice for memory intensive tasks	Cons • Requires a plugin like Silverlight to run in a browser • Depends on an IDE to code with it

Recommendation: We have decided to use C# for our back-end language as it is a universal language and integrates well with Windows. We will be using Visual Studio for C# development so everything works out. We will be incorporating the Silverlight plugin to make sure C# is compatible with the browsers we will be working with. Although we need a plugin, C# works well with Microsoft which we will be using for our project.

Front-End Framework

Vue.js (2.0)	Angular 6	ReactJS (v16.5)
Pros • Is incredibly lightweight, taking up 20kb of space after zipping • Clean, simple integration; component logic and structure combined in one file Cons • Full English documentation is lacking, since original is written in Chinese • In danger of over flexibility with rapid expansion of the language	Pros • Has a large community for help and support • Enhanced RxJS makes for better callback-based code • Fast compilation; under 3 sec. Cons • It has a steep learning curve • Has been known to have migration issues when updating older versions	Pros • Has a large community for help and support • Easy to learn and simple syntax • Backed by Facebook Cons • Requires more time researching which supporting libraries to use since ReactJS relies on additional 3rd party libraries to provide basic functionality • Lacking separation of concerns, HTML and JS are often found in the same files

Recommendation: We have decided to use Vue as the front end framework of this application. It is very powerful despite its insubstantial size. Vue allows us to have flexibility when it comes to our simpler and more complex functions within the application. Vue even borrows some of the functionalities of Angular and React, such as: custom directories, JSX, and shadow DOM.

Database

MySQL 8.0	MS SQL 14.0	Oracle 18.1
Pros • Unlimited size for database/table • Fully-managed SBaaS through either Google or Azure Cons • Need to download third-party provider tools if working with .NET • Data restoration is time consuming due to execution of multiple SQL statements	Pros • Maximum database/ table size: 542,272 TB • Relations allow you to store each data only once - no duplicates Cons • More costly than MySQL, need to purchase licenses to use and run multiple databases • Only supports Linux and Windows platforms	Pros • Maximum database/ table size: 2PB • Has a resource description framework (RDF) store Cons • Requires purchasing of full license, which is costly • A notable amount of resources are required to even run Oracle; hardware upgrades sometimes necessary

Recommendation: We recommend using Microsoft SQL as it is mainly used with .NET and it makes integration simpler for windows projects. Microsoft SQL server uses indexes to support data and speed up performance. Built around a row-based table structure, this allows related data to be connected, avoiding redundancy and providing greater data integrity. There is also plenty of online support and documentation for MS SQL. It supports a variety of transaction processing, business intelligence, and analytics applications. MS SQL allows us to scale as the business grows. We will be strictly working on one database, therefore there won't be any costs spent on MS SQL.

Integrated Development Environment (IDE)

Visual Studio 2017 (v15.8)	Mono Develop (v7.5.1254)	SharpDevelop 5 (develop)
Pros • Excellent and broad range of plugins • Includes a product backlog used in agile development • Supports every kind of .NET development Cons • Slow operating speed if minimum system requirements not met: 1.8 GHz processor, 4GB RAM, bare minimum of 10GB free hard drive space, and SSD is recommended	Pros • Does not use as much resources as other IDEs • Great for beginners • Incredibly light weight, only taking up 200 MB Cons • Bad formatting. Doesn't offer much in terms of autocompletion and code formatting • Very limited in features compared to competing IDEs	Pros • Supports plugins • Great for those who want to start .NET development • Form Designer for C# Cons • No debugger • Does only Visual Basic and C# projects • 3rd party controls aren't handled well • Cannot do mobile or ASP web apps (no form designer)

Recommendation: We highly recommend the use of Visual Studio as it is highly superior to other IDEs in terms of .NET and C# development. It is a great tool for agile development as it comes with a backlog on projects. Although Visual Studio takes up resources, most developers should have a good enough computer to run it.

Wireframe Tools (Website Blueprint)

Balsamiq Cloud (v3.5.16)	Figma (October 2018)	Sketch (v52.1)
Pros • Intuitive interface, with an easy way for text customization • Large and extensible UI element library Cons • Anything not directly supporting Balsamiq library is not possible • No easy way to quickly play around with the product	Pros • Free • Multiple real-time team collaboration features: Liveshare, Dev Handoff, and Version Control • Can be used directly from the browser • Built-in commenting Cons • Workflow slows down with a lot of components	Pros • Designed specifically for web and mobile UI design and workflow • Built-in grid system • "Mirror" makes it easy to test designs on multiple devices • Full offline editing Cons • No Windows and Linux support • Buggy and lack of quality assurance

Recommendation: We decided to work with Figma as it is free for individuals looking to do 3 projects. It provides real time collaboration and can be used directly from within the browser. We will be using this tool to prototype our application so depending on the application components our workflow will get affected

Modeling/Diagrams

Draw.io	Gliffy	Lucidchart
Pros • Free • Automatic layout • Diagrams revert with pages • Great accessibility to the features Cons • No lockpoint integration • There should be a desktop; only a web application	Pros • Allows lockpoint integration • Very extensive library for various types of shapes • Great with Entity Relationship and Data Flow diagrams Cons • Does not provide PDF export • Does not provide team collaboration	Pros • Commercial-friendly license • Online repository for diagrams and collaboration • iOS app exists • Ability to import/export different file types Cons • Not free • Lack of customization options

Recommendation: We recommend using Draw.io for modeling and diagrams as it is free and provides great accessibility and features. It has the ability to export in multiple formats as well as the ability for multi-user collaboration. Draw.io provides a self-explanatory set of tools that is well sorted and provides easy functionality.

Version Control

Git (v2.19.1)	Team Foundation Server (v2018 Update 3)	Subversion (SVN) (v1.10)
Pros • It allows for local repositories meaning developers can work on their own machine and offline if need be. • Git is fast at sending files that have been changed. • You can easily tell what developer wrote what line which comes in handy when code needs to be fixed. Cons • Windows OS has very limited support • Does not work well with binary files, no support for exclusive-locks	Pros • The code management is superior to other products as the techniques are much less error prone • The usability and unit testing suite is very flexible as it can be automatically executed for regression testing. • Individual and team tasking can be applied to any methodology template a manager would want to use to manage a team Cons • It offers an offline model however it is unreliable. • Structure synchronization of code and functionality pushed from one to another can be buggy at times	Pros • Critical code can be centralized which allows better control of who has the code. • Has incremental revision numbers to better see the current version being viewed Cons • Costly use of space with branching and tagging processes • Being a file based system, instead of object based, makes merging and branching difficult • Unable to make commits offline

Recommendation: We recommend Git due to its speed, its branching feature, and able to tell who wrote what code. It is not necessary to be connected to the internet and can allow developers to work off their machine. When multiple developers are working on the same file conflicts can occur when they start to merge their work. This can be reduced with gits' feature called branching.

3 Security Requirements

Hashing Functions

Bcrypt (v3.0.1 npm)	Scrypt (v6.0.3 npm)	Argon2id (1.3)
Pros • Harder and slower for an attacker to brute force passwords • Performs hash-based message authentication code (HMAC) as specified by the "iterations" parameter. • Unlike Scrypt, which is 6 years old, Bcrypt has been part of the crypto community for almost 19 years now and still remains unbroken.	Pros • Optimizing features depending on the kind of hardware you are using. • One of the more faster hashing algorithms and thus increases efficiency	Pros • Utilizes both Argon2i and Argon2d, accessing memory in a password dependent order the first time and all subsequent passes access in a password independent order • Data-depending memory access eliminates side-channel timing attacks, good for cryptocurrencies
Cons • Bcrypt is slower than Scrypt and MD5 (milliseconds vs microseconds) • There is a 72 character password limit compared to Scrypt with 82 characters	Cons • An attacker who has access to the host that is used to hash passwords can have additional ways to figure the passwords.	Cons • Still a very young system, with a small amount of scrutiny • A previous version, Argon2i, has been susceptible to a

Recommendation: We recommend using Bcrypt because it performs multiple message authentications depending on the amount of iterations. Bcrypt does not use intensive resources in the usage algorithm like Scrypt. Although Bcrypt is slow, for password hashing and has remained an unbroken hash function since its creation.

Transfer Protocol

HTTPS	HTTP	HTTP/2
Pros • Encrypts data being transferred between two systems • Easy transfer from an HTTP system • Builds trust with visitors because they know they are on a secure website Cons • Need to purchase an SSL certificate (although there are some free ones)	Pros • The most basic means for communication between systems • Allows two systems to transfer data Cons • Data is not encrypted and can be intercepted by third parties	Pros • Faster loading times than HTTP and HTTPS • Improvement involving header compression and overall performance Cons • Not as well established or available

Recommendation: We recommend the obvious choice of going with HTTPS. Who doesn't want their data and credentials to be encrypted? It is pretty much a standard to use HTTPS to keep communication between systems secure. Purchasing a SSL/TLS certificate is worth it when the security of your user accounts are at stake.

Payment Information

Paypal	In-App Payment Info	WePay
Pros • Great customer service • Accepts a variety of different payment types • Electronic invoices • Market availability is international Cons • Users will need to make a Paypal account	Pros • Allows us to use a Payment Handler API. • Allows us to only make necessary features specific to our app • We provide the interface and means of securing their information. Cons • Features involving different currency need to be developed • We, ourselves, need to make sure that everything is secure as we are dealing with the customer's payment information	Pros • Equipped to handle complex payment requests • Offers a helpful support center for businesses using the gateway Cons • No phone number and an online Help Center which still makes it difficult to make contact

Recommendation: At this current moment, we plan on using our own payment system which we will be developing. This allows us to focus on features specific to what our web application needs. We can tailor the experience to how we want it while also taking into consideration that he scope includes all of North America and its territories. We do not need to incorporate international payments and any complex payment features. Our main goal is to provide the best security for customers as well as conversion rates from countries like Canada, Mexico, and other North American territories.