

32

Accessible Wayfinding: McMapping Edition

by

Mollie Burke, Anne Catherwood, Kharajae Million, William Overmyer, and
Matthew Smith

Group 1

Submitted to Rebekah Michael and Abdou Fall,
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science in Information Technology

© Copyright 2021 Accessible Wayfinding: McMapping Edition

The authors grant permission to the School of Information Technology
to reproduce and distribute copies of this document in whole or in part.

Mollie Burke

Mollie Burke

04/27/2021

Anne Catherwood

Anne Catherwood (Apr 22, 2021 21:51 EDT)

Anne Catherwood

04/27/2021

Kharajae Million

Kharajae Million (Apr 22, 2021 21:53 EDT)

Kharajae Million

04/27/2021

William Overmyer

William Overmyer (Apr 22, 2021 21:54 EDT)

William Overmyer

04/27/2021

Matthew Smith

Matthew Smith (Apr 22, 2021 21:55 EDT)

Matthew Smith

04/27/2021

Abdou Fall

Abdou Fall, Faculty Advisor

04/27/2021

University of Cincinnati

College of

Education, Criminal Justice, and Human Services

April 2021

Table of Contents

List of Illustrations	ii
Tables	ii
Figures.....	ii
Abstract.....	1
Problem Statement.....	2
Introduction	2
Problem.....	2
Solution	3
Project Goals/Brief Methodology.....	4
Overview	5
Discussion.....	5
Project Concept / Solution.....	5
Design objectives	6
Methodology/Technical Approach	6
Design requirements	7
Procedures	8
User Profile	8
User Diagram	11
Testing.....	12
Overview	12
Testing Methodology and Approach	13
Scope of Testing.....	14
Objectives	15
Test Plan - QA Functional.....	15
Test Plan – User Acceptance.....	27
Test Results	28
Project Budget	29
Project Timeline	32
Problems Encountered and Analysis of Problems Solved	38
Firebase Authentication	38

Programming Language	38
Phaser Integration with Vue.js	39
Tiled	39
Programming language syntax	39
More Vue / Phaser compatibility issues	40
Future recommendations/Recommendations for Improvement	40
What would do you if had it all over to do again?	40
What would you do if you had more time?	40
What suggestions have you had from others?	41
What, if any, are your plans for this project?	41
Conclusion	42
Lessons Learned.....	42
References.....	45

List of Illustrations

Tables

Table 1.....	9
Table 2.....	10
Table 3.....	10
Table 4.....	27
Table 5.....	28
Table 6.....	29
Table 7.....	32
Table 8.....	37

Figures

Figure 1	12
----------------	----

[Click here to enter text.](#)

Abstract

Accessible Wayfinding: McMapping Edition is route-finding web application, compliant with guidelines laid out in the Americans with Disabilities Act Standards for Accessible Design, that provides users with accessible routes within the Arts and Sciences building, formally known as McMicken Hall. The University of Cincinnati's recent accessibility initiative uncovered significant deficits in the campus' overall accessibility. While this initiative has improved many elements of the university campus, it has yet to address needed improvements for those navigating inside UC's buildings.

The McMapping wayfinding web application is designed to provide students, faculty, staff, and visitors with mobility concerns the means to navigate through the Arts and Sciences building with peace of mind. By utilizing Phaser, the application will feature visual route guidance as an individual can first, select a starting location and a target destination within the establishment. The program provides the most accessible route to that spot. In addition to accessible route guidance, authenticated users can use saved routes and notification features to quickly identify previously used routes and to help other users navigate their way to success!

Problem Statement

Introduction

Contemplating the navigation of UC's campus can be daunting; finding the right building, the entrances, stairs, and classrooms is challenging for any one visiting campus. Even after basic layout knowledge is acquired, students with mobility concerns still find this experience a challenge. The Arts and Sciences building, formally known as McMicken Hall is outdated and was not designed with accessibility in mind. Thus, many students, staff, faculty, and visitors must plan for additional troubles that stem from that decades-old oversight. An individual might anticipate that required elevators could be out of order, or that one of the only two ramp access doors would be blocked by construction. Unforeseen obstacles might well make a hallway too narrow for a wheelchair's wheelbase, the sole usable stall in each bathroom could be flooded. Finally, in an emergency, those accessible exits might be incredibly challenging to access.

This is where Accessible Wayfinding: McMapping Edition comes the rescue. This web application routes any user between entrances, classrooms, bathrooms, elevators, and exits in a few simple steps. The application is readily available, guests and one-time users can load the site and immediately start routing, while users who want to save routes can create an account and do so. This flexibility allows any type of user to get what they want out of the application and nothing more, so they can focus on what matters most – learning!

Problem

The University of Cincinnati has a historic campus, but that history comes with shortcoming in design. A 2019 study conducted by EduCause, a third-party non-profit, found "...half of the 40,000

students surveyed rated their institution negatively in terms of accessibility.” and part of that comes from much of UC’s architecture having been designed before many architects worked with accessibility in mind (Gierdowski,2019). The Arts and Sciences building is the first building to be built on the University of Cincinnati’s Burnet Woods Campus location, Cincinnati architect Samuel Hannaford’s original design for the building was completed in September of 1895. It was later redesigned by another famed Cincinnati Harry Hake and rebuilt in its current location in September of 1950, forty years before the Americans with Disabilities act laid out required standard accessibility practices for construction. This mid-20th century design can make navigating between the halls’ thirteen departments, three floors, eight entrances (only two of which do not involve stairs) difficult for people whose needs are not met by the age of the architecture. Although it is outfitted with elevators, ramps, two areas of rescue assistance, and door access switches, the layout of the building is cumbersome, split up by stairs, often made inaccessible by construction, and the elevators are regularly out of order. This forces students and faculty members with disabilities to navigate through several challenging pathways to get to their destination, taking much more time than able-bodied students and faculty. This can be exceptionally daunting if you are new to UC or are a visitor.

Solution

Accessible Wayfinding: McMapping Edition is a wayfinding web application designed to help users navigate to their desired destination as quick and easy as possible. The web application is also fine-tuned to help users with physical disabilities navigate the building, avoiding stairs and emphasizing elevators and restrooms. This application is built using the Vue.js framework to implement a Phaser-based routing system, with a Firebase database to handle user accounts, allowing users to save routes.

Visitors, students, adjunct professors, janitors and more will have a personal travel guide to get to class quickly.

The saved routes feature is particularly useful for campus frequenters who need routing around the Arts and Sciences building. Users can quickly log in using their google account or creating an account using a preferred email and password. The saved routes feature is especially useful for campus frequenters who need routing around the Arts and Sciences building. Users can quickly log in using their google account or creating an account using a preferred email and password, those account credentials are then secured in the Firebase database. Users can also report bugs directly to the development team and have access to a map legend with descriptions of icons for visitors with vision impairments.

Project Goals/Brief Methodology

The McMapping Accessible Wayfinding application is a web application that aims to provide accessible routing for users with disabilities, helping them find their way through the Arts and Sciences building. The application allows users to enter a start point and end point, at different areas within and outside of the building, and then will draw the fastest accessible route between those points. Users will be able to use quick access routes that direct them to a specified room such as restrooms and elevators. Users will also be able to login and save their class schedules to make transitioning between classes easier and using different internet-connected devices to do so seamless. McMapping was developed using Vue.js, Bootstrap Vue, Firebase Database, Firebase Authentication, Tiled map creation software, and the Phaser game engine.

The McMapping team followed the methods of project management styles and agile sprint development cycles that are further outlined in the procedure section below; these procedures kept the project moving forward in the development lifecycle. Scrum methodologies were used to emphasize

teamwork, accountability and interactive sprints that were progress driven towards well-defined goals. The focus on communication, transparency of tasks, the inspection of those tasks, and, if needed, of the tasks in question. Three user profiles defined development: the developers and administrators, student and faculty members, and guest users. The user diagram displays how users and authentication services interact with the web application. The preliminary budget outlines costs for each labor and software resource, totaling to an operating cost of \$30,082 for two years. The project timeline goes through a series of goals and milestones both course-led for the project management side and milestones of development that had been decided on, such as adhering to web standard guidelines, from 8/20/2020 to 4/13/2021.

Overview

The remainder of this report covers how the McMapping Accessible Wayfinding application has progressed. The report includes the following sections: project concept and solution, design objectives, methodology and technical approach broken down by design requirements and procedures, user profiles, user diagram, project budget, project timeline, problems encountered, analysis of problems solved, lessons learned, and plans for the spring semester.

Discussion

Project Concept / Solution

The project idea originated when one team member experienced trouble while navigating the Arts and Sciences building. After noticing that when certain accessible routes were closed off or undergoing maintenance the challenges that delayed transit from location to destination were far more available and less taxing for able-bodied students, thus the initial plan for interior wayfinding software

was conceived. The McMapping Wayfinding Application aims to provide users with a reliable and straightforward way to generate accessible routes to navigate through the establishment.

Design objectives

McMapping needs to allow users to enter a start point and end point and generate a route between the two. This process must be self-explanatory and have clear directions as inexperienced users will not have the benefit of learning from outside official sources. Phaser, a game engine, was determined to be the most convenient way of displaying map data, while also allowing easy access to Artificial Intelligent (AI) pathfinding that would automatically generate the shortest routes. It also has a very non-restrictive user-interface system that allows designers to convey information in a wide variety of ways.

Users need a way to block off paths they determine are inaccessible, as well as mark inaccessibility for other users if they choose to do so. This not only allows them to re-route themselves after finding obstacles but pre-emptively help others route around issues as well. Phaser allowed obstacles to be added easily, and a Firebase server allows information to travel between users easily.

Originally there was a hope to integrate UC's campus wide wayfinding application with McMapping, or at least some utilization of their hardware. However as that was being handled by a third-party development company it was determined to be non-feasible. Utilization of google maps API to determine a rough estimate of user starting positions still being considered, but its viability versus the difficulty in implementation has yet to be determined.

Methodology/Technical Approach

Design requirements

Consistent progress was made after choosing design requirements early in the development cycle. After determined the functional requirements for the project, focusing on what the website needed to do, and what value McMapping should provide to the application users, the following guiding design requirements were established.

- 1) Web Content Accessibility Guidelines: The application must adhere to Web Content Accessibility Guidelines, as outlined in the W3C Accessibility Standards (Henry, 2019). Therefore, BootstrapVue was chosen instead of Vuetify because it allowed rapid and flexible implementation of Bootstrap 4 components into Vue.js. Additionally, it supports the goal of an easy-to-use application. While users might not be aware of what Bootstrap is, they are familiar with its distinct styling.
- 2) Ease of Use: Beyond what is covered by web accessibility guidelines, the application must also load quickly, run smoothly, and not utilize an unreasonable amount of users' system hardware while running. It is important to make sure that users do not stop using the application because of unfriendly backend design.
- 3) User Accounts: The application must be able to store user accounts and sessions as well as saved and frequently used routes. Our team chose Firebase Authentication because it is a powerful login component that requires minimal development time. Server-side authentication is more performant and secure than client-side authentication. The database holds necessary application data such as user accounts.
- 4) Data Storage: The application must know all the map coordinates as they relate to floor and room numbers. Vuex manages state by serving as a centralized store for application data. Phaser and Vue.JS use the Vuex store to pass data to one another allowing our Phaser map to integrate seamlessly into a Vue.JS component. The routes are saved in a Firebase database with

the user so that they can be retrieved on a per user basis and then pushed to phaser through the Vuex store.

Procedures

The procedures followed contributed to the success of the project; the McMapping team created a working agreement, the McMapping Accords, that outlined how each member of the team would be expected to conduct themselves while working on this project. The Accords set the expectation for each team member and clearly defined responsibilities and requirements for the group. All signatories agreed to have a minimum of two meetings per week, once for a weekly check in and once for weekly retrospective. This allowed everyone to stay very intricately connected and aware of what other team members were working on throughout the week. Additionally, this allowed individuals to ask for help from others when needed, which improved development times. Discord was used as a main method of communication; available as an app for mobile and PC, Discord enabled constant communication throughout the project lifecycle.

Mob programming sessions allowed us to work through blockers as well as provide shared understanding. Meanwhile GitHub provided the necessary source control to work asynchronously while maintaining the cohesiveness of a unit.

User Profile

There will be three types of users interacting with the McMapping web application, the owners, frequent-users, and guests. The owner users include administrators and web application developers who are responsible for creating and maintaining the application post-launch. The frequent-user group consists of students, staff, and faculty with the common interest of navigating the Arts and Sciences building. They will want data about their activities saved for later use, to avoid the need for repetitive

user input. The guests are users who visit the campus infrequently and need to quickly find routes to their destination within the facility but do not need their information saved.

Table 1: Web Application Developer/Administrator - User Profile

Project: Accessible Wayfinding Application
User Profile Form 1
Potential Users: Application developers/Administrator
Application: Vue.js, Bootstrap, HTML, CSS, GitHub, Phaser, Command Line, JavaScript, CircleCi, Firebase
Software and Interface Experience: The duties of this user suggest that the user is familiar with Vue.js and GitHub commands using a terminal. The user should also be familiar with coding used to create the web application. Handles banned users and checks user reports.
Prerequisite Experience: This user should be familiar with languages used to create the web application along with the VueJS JavaScript libraries and the Bootstrap framework.
Task Experience: Using the integrated languages and applications used in the development and maintenance of the web application.
Frequency of Use: Post-deployment of this application, the user will interact with the application on an “as needed” basis to maintain, update, and fix any bugs discovered after the initial launch of the application. The updates will keep the web application’s security requirements, functionalities, and policies up to date.
Key interface Design Requirements that the Profile Suggests: This user needs to be familiar with the programming and scripting languages used to develop and maintain the web application.

Table 1

Table 2: Student/Faculty – User Profile

User Profile Form 2
Potential Users: Student, Faculty
Application: Accessible Wayfinding Application

Software and Interface Experience: Users should have experience with GPS (Global Positioning System) applications such as Google Maps, or Waze and users should be familiar with text-based scheduling applications like Pocket Schedule Planner.
Prerequisite Experience: Google Maps, Apple Maps, Pocket Schedule Planner, Calendars, Safari, Google Chrome
Task Experience: Basic internet interactions: Using a keyboard to type classes into a schedule planner and search locations to navigate to.
Frequency of Use: When a user needs assistance navigating through the Arts and Sciences building. Frequency of use could be daily, weekly, monthly, or yearly.
Key interface Design Requirements that the Profile Suggests: The web application needs to be easily accessible to users, UX/UI needs to be user-friendly, route mapping needs to be accurate, and the application needs to be secure.

Table 2

Table 3: Guest/Visitor – User Profile

User Profile Form 3
Potential Users: Guest/Visitor
Application: Accessible Wayfinding Application
Software and Interface Experience: Users should have experience with a search functionality such as Google and GPS applications such as Google Maps, or Waze.
Prerequisite Experience: Google Maps, Apple Maps, Google Chrome, Waze
Task Experience: Basic internet interactions: Using a keyboard to search locations to navigate to.
Frequency of Use: When a user needs assistance navigating through the Arts and Sciences building. Frequency of use is on an as needed basis.
Key interface Design Requirements that the Profile Suggests: The web application needs to be easily accessible to guest users, UX/UI needs to be user-friendly, and route mapping needs to be accurate.

Table 3

User Diagram

The following user diagram (figure 1) lays out how the three major types of users (as outlined in “user profiles”) are expected to interact with the McMapping wayfinding application. This diagram includes the type of actions each user can perform, which actions are connected, and how these actions interact. The diagram also includes the role played by the authentication services as a third user-like entity, and how those services interact with other users operating the application.

Figure 1: Use Case diagram

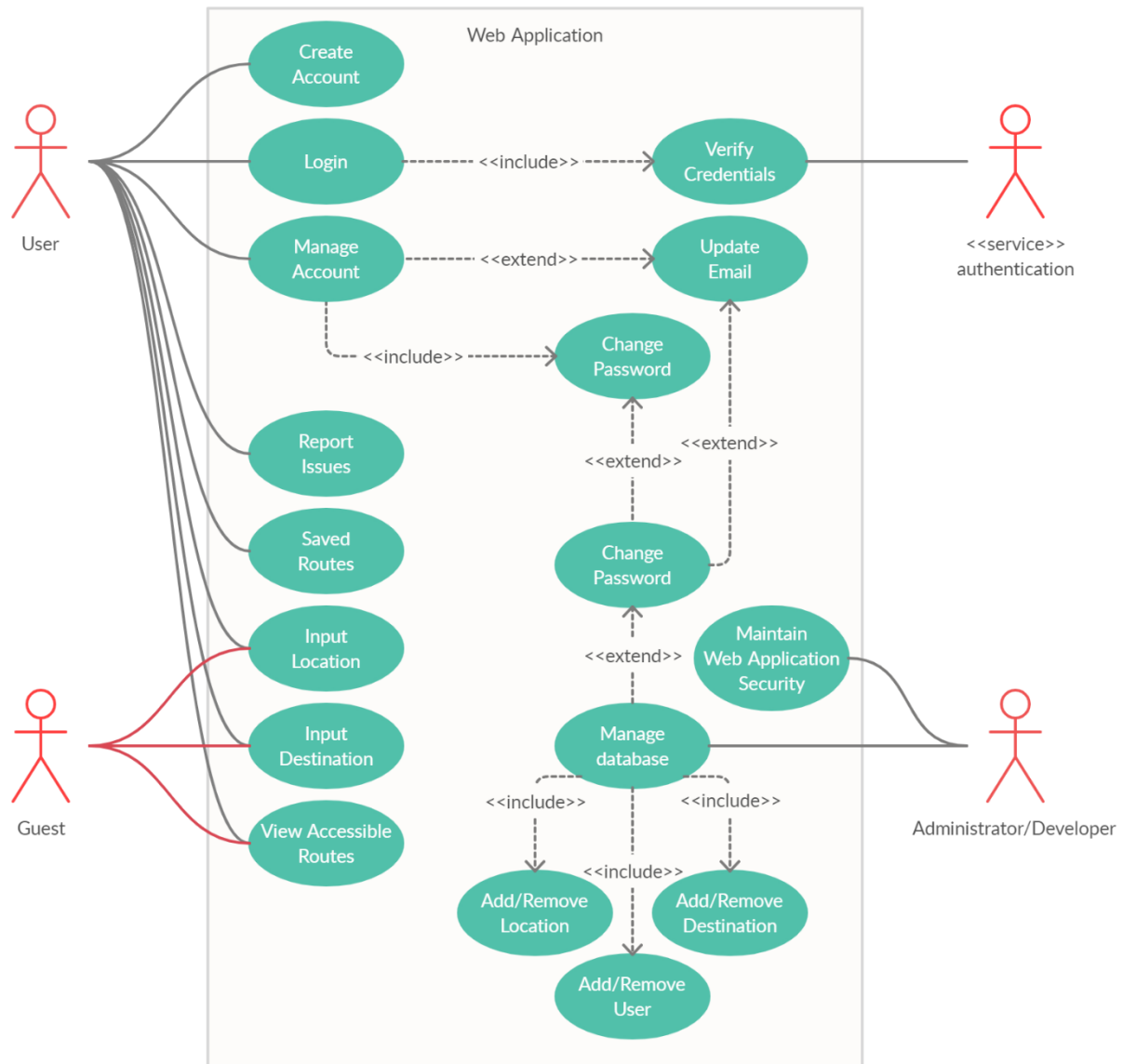


Figure 1

Testing

Overview

To determine McMappings' overall functionality, identify areas of improvement, and find bugs, thorough testing of the web application was required. During development, all members of the team

were expected to test their own code, as well as how well their work integrated with the full project.

After a sufficiently feature-ready product had been developed user acceptance testing was performed in which non-developers were asked to use the McMapping web site and to evaluate their experience.

Testing Methodology and Approach

The methodology of our testing process is two-fold: first we focus on Quality Assurance Functional Testing, then we focus on User Acceptance Testing.

The Quality Assurance functional test plan contains a smoke test case, integration test cases, and regression test cases. The smoke test is a test case where a user goes to the web address, logs into the site, and checks that the map loads on the page. The integration test cases check interactions between the Vue.js directions component, and the Phaser map and navigation bar components. The regression test cases go into greater detail on smaller components and interactions and are performed in their entirety when any changes in the code occur.

The User Acceptance Testing test plan contains a smoke test case and has test cases for each step of the end-to-end workflow. This covers the three core features: routing, usefulness of the map, and logging in. The user will be guided by test cases asking the tester to perform certain actions, like selecting a route from one classroom to another, and selecting a route from the entrance of McMicken to a room on another floor. After the tests are finished, testers are sent [a survey about their experience](#) using the application to catalog their responses. After analyzing feedback, the McMapping web application will be updated to better help those using it.

Scope of Testing

During testing the experimental users were surveyed about their interactions with the McMapping site, and their responses were judged to gauge the reliability, useability, and effectiveness of three core features.

- Routing: the user selects a starting point and a destination point. We plan to test this extensively as this is the core functionality of our application. Included in the scope of this testing:
 - User selects a route from the entrance of the building to a classroom
 - User selects a route from one floor to another
 - User selects a route to a restroom
- Usefulness of the map: the user can clearly identify classrooms, elevators, and restrooms. We plan to test this because it is critical that our users can quickly and clearly understand the map so that they can navigate throughout the building with ease. This testing will be completed as a part of our questionnaire.
- Login: the user can register or login to the application and view their saved routes. We are testing this because users without an account cannot view their saved routes. Included in the scope of this testing:
 - Register successfully
 - Login successfully
 - Forgot password link
 - While logged in only
 - Ability to save routes
 - Ability to view saved routes

Objectives

The tests run on the McMapping web application allowed us to create a quality product by helping analyze aspects of the webapp, with tests designed to fulfill key goals.

- Confirm all major functions of the application and core functionality is working as intended, in compliance with our end goals of the application.
- Identify and resolve bugs found while testing as soon as possible to implement another testing iteration before final submission.
- Test saved routes to make sure specified routes are being saved to the application's database.
- Ensure the application correctly finds routes inside of McMicken Hall and displays that information to the user in an easy-to-understand way.
- The Webapp is hosted on a publicly reachable server before presenting at the IT Expo.

Test Plan - QA Functional

Test Case #	Test Name	Test Description	Step #	Step Description	Expected Result	Data Needed
Access-QA-01	Guest Smoke Test	Going to the URL for the website homepage	Access-QA-01.1	Go to https://vue-project-302920.ue.r.appspot.com/	The webpage should be titled Accessible Wayfinding: McMapping Edition. There should be a map of floor 1 of the Arts's and Sciences building, along with a floor selection	N/A

					field, and go button field.	
			Access-QA-01.2	Check that the webpage matches the expected result for 01.1	The results will match	N/A
Access-QA-02	User Smoke Test	Logging in to the website	Access-QA-02.01	Go to https://vue-project-302920.ue.r.appspot.com/	The webpage should be titled Accessible Wayfinding: McMMapping Edition. There should be a map of floor 1 of the Arts's and Sciences building, along with a floor selection field, and go button field.	N/A
			Access-QA-02.02	Check that the webpage matches the expected result for 02.1	The results will match	N/A
			Access-QA-02.03	Click the Sign In button	You will be taken to the Sign In page	N/A
			Access-QA-02.04	Use the username and password in the Data Needed	You will be taken back to the main page. The directions field will	The username is testCapstone@gmail.com . The

				column to fill out the fields. Then click Sign In	now have a new tab "Saved Routes"	password is Test123
Access-QA-03	Sign out	Signing out of an account	Access-QA-03.01	Click the Sign Out button.	The page will refresh and the Saved Routes tab will no longer be on the Directions field	N/A
Access-QA-04	Account Creation	Creating an account	Access-QA-04.01	Click the Register button	You will be taken to the register page	N/A
			Access-QA-04.02	Enter a username and password, then click register	You will be taken to the main page. The directions field will now have a new tab "Saved Routes"	Any username and password will work if it is not one that has been used previously.
Access-QA-05	Direction one floor	Routing between rooms on one floor	Access-QA-05.01	Choose a room number for the first floor in the start dropdown.	The room number will populate the start dropdown	N/A
			Access-QA-05.02	Choose a room number for the first floor in the end dropdown. Then click Go.	There will be a blue line on the map showing the route between the two rooms. There should be a yellow	N/A

					arrow next to floor 1's button on the select a floor field.	
			Access-QA-05.03	Click the floor 2 button to be taken to floor 2's map. Repeat steps 05.01 and 05.02 for floor 2	There will be a blue line on the map showing the route between the two rooms. There should be a yellow arrow next to floor 2's button on the select a floor field.	N/A
			Access-QA-05.04	Click the floor 3 button to be taken to floor 3's map. Repeat steps 05.01 and 05.02 for floor 3	There will be a blue line on the map showing the route between the two rooms. There should be a yellow arrow next to floor 3's button on the select a floor field.	N/A
			Access-QA-05.05	Click the floor 0 button to be taken to floor 0's map. Repeat	There will be a blue line on the map showing the route between	N/A

				steps 05.01 and 05.02 for floor 0	the two rooms. There should be a yellow arrow next to floor 0's button on the select a floor field.	
Access-QA-06	Direction floor to floor	Routing between rooms on two different floors	Access-QA-06.01	Choose a room number from the first floor in the start dropdown.	The room that was selected should populate the dropdown field.	N/A
			Access-QA-06.02	Choose a room number from the second floor in the end dropdown. Then click Go.	There will be a blue line on the map showing the route between the room on the first floor, to the nearest elevator. Once the line reaches the elevator, the map should change to be the second floor. The line should route from the same elevator to the room on the second floor. There	N/A

					should be a yellow arrow, pointing right, next to the first-floor button on the select a floor field. There should be a second yellow arrow, pointing left, next to the second-floor button.	
			Access-QA-06.03	Choose a room number from the second floor in the start dropdown.	The room that was selected should populate the dropdown field.	N/A
			Access-QA-06.04	Choose a room number from the third floor in the end dropdown. Then click Go.	There will be a blue line on the map showing the route between the room on the second floor, to the nearest elevator. Once the line reaches the elevator, the map should	N/A

					change to be the third floor. The line should route from the same elevator to the room on the third floor. There should be a yellow arrow, pointing right, next to the second-floor button on the select a floor field. There should be a second yellow arrow, pointing left, next to the third-floor button.	
			Access-QA-06.05	Choose a room number from the third floor in the start dropdown.	The room that was selected should populate the dropdown field.	N/A
			Access-QA-06.06	Choose a room number from the ground floor in the end dropdown.	There will be a blue line on the map showing the route between the room on	N/A

				Then click Go.	the third floor, to the nearest elevator. Once the line reaches the elevator, the map should change to be the ground floor. The line should route from the same elevator to the room on the ground floor. There should be a yellow arrow, pointing right, next to the first-floor button on the select a floor field. There should be a second yellow arrow, pointing left, next to the ground floor button.	
			Access-QA-06.07	Choose a room number from the ground floor	The room that was selected should populate the	N/A

				in the start dropdown.	dropdown field.	
			Access-QA-06.08	Choose a room number from the first floor in the end dropdown. Then click Go.	There will be a blue line on the map showing the route between the room on the ground floor, to the nearest elevator. Once the line reaches the elevator, the map should change to be the first floor. The line should route from the same elevator to the room on the ground floor. There should be a yellow arrow, pointing right, next to the ground floor button on the select a floor field. There should be a second yellow arrow, pointing left, next to	N/A

					the first floor button.	
Access-QA-07	Selecting a floor	Selecting the map buttons in the Select a Floor field	Access-QA-07.01	Refresh the page to make sure you are on the first floor map	You should see the first-floor map once the page reloads.	N/A
			Access-QA-07.02	Click the second floor button	The map should reload and be a map of the second floor	N/A
			Access-QA-07.03	Click the third floor button	The map should reload and be a map of the third floor	N/A
			Access-QA-07.04	Click the ground floor button	The map should reload and be a map of the ground floor	N/A
			Access-QA-07.05	Click the first floor button	The map should reload and be a map of the first floor again, just like it was when the test began	N/A
Access-QA-08	Saving routes	Choosing a route, saving it, and running it	Access-QA-08.01	Log in with an account.	The saved routes tab will appear on the directions field.	N/A

			Access-QA-08.02	Choose a room for the start dropdown.	You should be able to choose a room from any floor.	N/A
			Access-QA-08.03	Choose a room for the end dropdown.	You should be able to choose a room from any floor.	N/A
			Access-QA-08.04	Click the Save Route button	The clicking animation should occur for the button.	N/A
			Access-QA-08.05	Click the Saved Routes tab and find the route you just saved in the list.	The route should be listed in the dropdown.	N/A
			Access-QA-08.06	Choose one of the listed routes and click the Go button.	The routing line should go between the two rooms	N/A
Access-QA-09	About Us	Going to the About Us page.	Access-QA-09.01	Click the hamburger menu in the top right corner	A dropdown list with three items "About Us," "Report Issues," and "Legend" should appear.	N/A
			Access-QA-09.02	Click the About Us link	You should be rerouted to the About Us page that contains our team's abstract and references.	N/A

Access-QA-10	Report Issues	Going to the Report Issues page	Access-QA-10.01	Click the hamburger menu in the top right corner.	A dropdown list with three items "About Us," "Report Issues," and "Legend" should appear.	N/A
			Access-QA-10.02	Click the Report Issues link	You should be taken to our team's GitHub issues page, where anyone can post any issues they have found on the site.	N/A
Access-QA-11	Legend page	Going to the Legend page	Access-QA-11.01	Click the hamburger menu in the top right corner.	A dropdown list with three items "About Us," "Report Issues," and "Legend" should appear.	N/A
			Access-QA-11.02	Click the Legend link	You should be taken to the Legend page, where the Map Legend is located.	N/A
			Access-QA-11.03	Click the Need More Help link at the bottom of the legend.	You should be taken back to our team's GitHub issues page, where anyone can post any issues they	N/A

					have found on the site.	
--	--	--	--	--	-------------------------	--

Table 4

Test Plan – User Acceptance

Test Case #	Test Name	Test Description	Step #	Step Description	Data Needed
Access-UA-01	Guest Smoke Test	Going to the URL for the website homepage	Access-UA-01.1	Go to https://vue-project-302920.ue.r.appspot.com/	Just the url in the step description
Access-UA-02	Account Creation	Creating an account	Access-UA-02.1	Find the Register button and click it	N/A
			Access-UA-02.2	Create an account using your gmail account credentials, or by using another email address	N/A
Access-UA-03	Direction one floor	Routing between rooms on one floor	Access-UA-03.01	Choose two rooms from the same floor to route between and use the web page to show you the direction you need to go	N/A
Access-UA-04	Direction floor to floor	Routing between rooms on two different floors	Access-UA-04.01	Choose a room from one floor and a room from a seperate floor to route between and use the web page to show you the direction you need to go	N/A
Access-UA-05	Selecting a floor	Selecting the map buttons in the Select a Floor field	Access-UA-05.01	Select the map buttons corresponding to each floor to view maps of each floor	N/A
Access-UA-06	Saving routes	Choosing a route, saving it, and running it	Access-UA-06.01	Choose two rooms anywhere in the building and have the web page route you between them	N/A
			Access-UA-06.02	Save the route you have selected	N/A
			Access-UA-06.03	Find the saved route in the saved routes list and run it again	N/A

Access-UA-07	About Us	Going to the About Us page.	Access-UA-07.01	Find the link for the About Us page and click it to go to that page	N/A
Access-UA-08	Report Issues	Going to the Report Issues page	Access-UA-08.01	Find the link for the Report Issues page and click it to go to that page	N/A
Access-UA-09	Legend page	Going to the Legend page	Access-UA-09.01	Identify the icons for restrooms and elevators on one of the floors, without looking at the legend page.	N/A
			Access-UA-09.02	Find the link for the Legend page and click it to go to that page	N/A
			Access-UA-09.03	If you were having trouble identifying the restrooms and elevators, attempt to identify them after looking at the Map Legend	N/A
Access-UA-10	Sign out	Signing out of an account	Access-UA-10.01	Sign out of your Accessible Wayfinding –McMapping account	N/A

Table 5

Test Results

The results of the Quality Assurance Functional Testing were as follows:

Test Number	Pass/Fail	Error	Date of Fix
Access-QA-01	Passed consistently	N/A	N/A
Access-QA-02	Passed consistently	N/A	N/A
Access-QA-03	Passed consistently	N/A	N/A
Access-QA-04	Passed consistently	N/A	N/A
Access-QA-05	Fail	1) The room coordinates were not matching up with their locations on the map 2)The routing line seemed to be going through walls	1)March 26, 2021 2)March 18, 2021
Access-QA-06	Fail	1) (Same error as above) The room coordinates were not matching up with their locations on the map 2) (Same error as above) The routing line seemed to be going through walls	1)March 26, 2021 2)March 18, 2021 3)March 14, 2021

		3)When the user was routed between rooms, the maps would be overlapping each other	
Access-QA-07	Passed consistently	N/A	N/A
Access-QA-08	Fail	Routes being saved twice would duplicate in the saved routes list	March 14, 2021
Access-QA-09	Passed consistently	N/A	N/A
Access-QA-10	Passed consistently	N/A	N/A
Access-QA-11	Passed consistently	N/A	N/A

Table 6

The results of the User Acceptance Testing were the features added to the web application:

- The restroom and elevator icons being outlined in black to be easier to spot on the map. ‘
- The map legend page for the users to go to if they have any questions about how to navigate the map and the routing features.
- The arrows that indicate which floor you are routing from and which floor you are routing to.

Project Budget

This budget (table 7) assumes that all project managers, programmers, and cyber security personnel work equal amounts and work 8 hours per week. Meetings take an hour and a half at minimum, with two meetings a week plus an extra hour to prepare requisite documents. This means four hours of work outside of the project per week for all team members. Four hours of programming research or testing is reasonable time for each team member to work toward action items determined during meetings as goals are met.

The budget is for a demonstration model of the project to offer to organization, such as the University of Cincinnati. Thus, the budget projects no access to UC-sponsored servers nor employment of long-term security or maintenance staff. Figure 4 does dedicate the last columns to the possibility of the project being implemented.

Table 7: McMapping Budget

McMapping Budget (preliminary)			
Resource	Cost Per Unit	Expense Repetition	Total Cost
Labor			
Project manager (entry level)	\$50/hour	8 hours per week 30 weeks	\$12,000
Programmer (entry level) x3	\$20/hour \$60/hr for x3	8 hours per week 30 weeks	\$14,400
InfoSec (entry level)	\$35/hour	8 hours per week 30 weeks	\$8,400
Software			
Bootstrap formating guide	\$32	One time purchase	\$32
Domian name setup	\$10	One time purchase	\$10
Web hosting through bluehost	~\$10 monthly	24 months	\$240
Total cost for project, operating for 2 years		\$30,082	
Long Term Implementation			
UC web-hosting server	???	???	???
InfoSec (entry level)	\$35/hr	<4 hours per week Full year	< \$7140 Per year
Programmer (entry level) Mainatenance	\$20/hr	<4 hours per week Full year	\$4080 Per year
Programmer (entry level) Adding new Buildings	\$20/hr	4 hours research and planning 6 hours programming 2 hours bugtesting	\$240 Pre building added to service
McMapping Budget (final)			
Resource	Cost Per Unit	Expense Repetition	Total Cost
Labor			
Project manager (entry level)	\$50/hour	8 hours per week 30 weeks	\$12,000
Programmer (entry level) x3	\$20/hour \$60/hr for x3	8 hours per week 30 weeks	\$14,400
InfoSec (entry level)	\$35/hour	8 hours per week 30 weeks	\$8,400
Software			

Bootstrap forming guide	\$32	One time purchase	\$32
Domian name setup	\$10	One time purchase	\$10
Web hosting through bluehost	~\$10 monthly	24 months	\$240
Total cost for project, operating for 2 years		\$30,082	
Long Term Implementation			
InfoSec (entry level)	\$35/hr	<4 hours per week Full year	< \$7140 Per year
Programmer (entry level) Mainatenance	\$20/hr	<4 hours per week Full year	\$4080 Per year
Programmer (entry level) Adding new Buildings	\$20/hr	4 hours research and planning 6 hours programming 2 hours bugtesting	\$240 Pre building added to service
Additional Features (Estimated)			
Adding Mobile: Programmer (entry level)	\$20/hr	Coding: ~20 hours Possibly needed New Maps: same as adding a new building.	~\$400 Coding \$240 Maps
Adding text entry for room finding: Programmer (entry level) InfoSec (entry level)	\$20/hr \$35/hr	Coding ~8 hours Safety testing (from injection attacks etc.) ~2 hours	~\$160 Coding ~\$70 testing
Allowing users to add blockages: Programmer (entry level)	\$20/hr	Planning: ~2 hours Coding: >16 hours Testing: >5 hours	>\$460
Blockages/outages added to online account. Programmer (entry level) InfoSec (entry level)	\$20/hr \$35/hr	Coding >8 hours Safety testing (from injection attacks etc.) ~2 hours	>\$160 Coding ~\$70 testing

Table 7

Project Timeline

The following timeline for the project as laid out below (figure 5) is the current plan for the fall 2020 through spring 2021 production cycle of the McMapping wayfinding application. It outlines in which order and over what length of time tasks and deliverables are to be completed by the team but is in no way meant to be a rigid structure. Changes to the timeline will be made as needed, but the as it stands it will help determine if a task has taken too much time and needs the full attention of the team or outside aid from advisors to meet more firm deadlines. Any tasks beyond this scope will be considered and budgeted for later when more of the relevant information is available.

Table 8: Project Timeline

Task #	Task Name	Duration	Start Date	End Date
1.0	Project Management & Deliverables	252 days	08/24/2020	04/13/2021
1.1	Discussing contract draft	14 days	8/10/2020	8/24/2020
1.2	Project Planning	14 days	8/10/2020	8/24/2020
1.3	Fall 2020 Assignment 0	14 days	8/10/2020	8/24/2020
1.3.1	Team Members and Track	14 days	8/10/2020	8/24/2020
1.3.2	Problem Statement	14 days	8/10/2020	8/24/2020
1.3.3	Solution Statement	14 days	8/10/2020	8/24/2020
1.4	Fall 2020 Assignment 1 – Finalized Contract	21 days	8/10/2020	10/12/2020
1.4.1	Project Approval	21 days	8/10/2020	8/31/2020
1.4.2	Intent	21 days	8/10/2020	8/31/2020
1.4.3	Problem Statement	21 days	8/10/2020	8/31/2020

1.4.4	Solution Statement	21 days	8/10/2020	8/31/2020
1.4.5	Contact Information	21 days	8/10/2020	8/31/2020
1.4.6	Project Source	21 days	8/10/2020	8/31/2020
1.4.7	Project Objectives and Goals	21 days	8/10/2020	8/31/2020
1.4.8	Team Member Responsibilities	21 days	8/10/2020	8/31/2020
1.4.9	Project Scope	21 days	8/10/2020	8/31/2020
1.4.10	Deliverables	21 days	8/10/2020	8/31/2020
1.4.11	Project Timeline	21 days	8/10/2020	8/31/2020
1.4.12	Technologies Used	21 days	8/10/2020	8/31/2020
1.4.13	Team Rules	21 days	8/10/2020	8/31/2020
1.4.14	References	21 days	8/10/2020	8/31/2020
1.5	Assignment 2 - Project Abstract	43 days	8/31/2020	10/12/2020
1.6	Assignment 3 - Team Contract Resubmission	43 days	8/31/2020	10/12/2020
1.7	3 Minute Elevator Pitch	50 days	8/31/2020	10/19/2020
1.8	Assignment 4 - User Profile	50 days	8/31/2020	10/19/2020
1.9	Fall 2020 Assignment 5 – Use Case Diagram	50 days	8/31/2020	10/19/2020
1.10	Fall 2020 Assignment 6 – Draft Report	21 days	10/19/2020	11/9/2020
1.11	Fall 2020 Final Fall Semester Report	21 days	11/9/2020	11/30/2020
1.12	Fall 2020 Semester Presentation	21 days	11/9/2020	11/30/2020
1.12.1	Finalizing Presentation Roles/Documents	21 days	11/9/2020	11/30/2020

1.12.2	Presentation Practice	21 days	11/9/2020	11/30/2020
1.13	Spring 2021 Assignment 1 - Testing Plan / Report	84 days	10/19/2020	01/11/2021
1.14	Spring 2021 Assignment 3 – Draft IT Expo Poster	21 days	03/15/2021	04/05/2021
1.15	Spring 2021 Assignment 4 – Final Poster	7 days	04/05/2021	04/12/2021
1.16	Spring 2021 Oral Presentation	21 days	03/22/2021	04/12/2021
1.16.1	Presentation Practice	7 days	04/05/2021	04/12/2021
1.17	Spring 2021 IT Expo	1 day	4/13/2021	04/13/2021
1.17.1	Spring 2021 IT Expo Exhibit Building & Preparation	10 days	04/05/2021	04/13/2021
1.18	Spring 2021 Assignment 5 - Final Report	45 days	03/01/2021	04/13/2021
1.19	Spring 2021 Assignment 6 – Safe Assign Final Report	45 days	03/01/2021	04/13/2021
1.20	Spring 2021 Assignment 7 – Library Copy	45 days	03/01/2021	04/13/2021
2.0	Research	28 days	08/24/2020	09/21/2020
2.1	Building plan and layout	14 days	8/24/2020	09/07/2020
2.2	Discussing working with hubs already used by UC	7 days	08/24/2020	08/31/2020

2.3	Deciding which IDE to use for development	7 days	08/24/2020	08/31/2020
2.4	Research domain	14 days	08/31/2020	09/14/2020
2.5	Research Authentication	14 days	08/31/2020	09/14/2020
2.6	Research accessible web-development	14 days	08/31/2020	09/14/2020
2.7	Research downloadable web-app	14 days	08/31/2020	09/14/2020
2.8	Research Phaser Pathfinding	14 days	08/31/2020	09/14/2020
2.9	Research AutoCAD / Tiled map implementation for GPS location	21 days	08/31/2020	09/21/2020
3.0	Design	21 days	08/31/2020	09/21/2020
3.1	Creating application UI mock-up	7 days	08/31/2020	09/07/2020
3.1.1	User home page	2 days	08/31/2020	09/01/2020
3.1.2	Enter needed route page (what room you need to get to)	2 days	09/03/2020	09/05/2020
3.2	User story creation	7 days	09/07/2020	09/14/2020
3.3	User feedback – using wireframe	7 days	09/07/2020	09/14/2020
3.4	Database Diagram	7 days	09/14/2020	09/21/2020
3.5	Wireframe Diagram	7 days	08/31/2020	09/07/2020
4.0	Development	203 days	08/31/2020	03/01/2021
4.1	Set up Github repo	7 days	08/24/2020	08/31/2020
4.2	Set up VueJS	7 days	08/31/2020	09/07/2020
4.3	Set up CircleCI	7 days	08/31/2020	09/07/2020

4.4	Develop maps in tiled from blueprints	92 days	10/01/2020	01/01/2021
4.5	Integrate tiled maps into Phaser	120 days	11/01/2020	03/01/2021
4.6	Integrate Phaser into Vue	203 days	08/31/2020	03/01/2021
4.4	Develop user home page prototype (has saved routes, login optional)	77 days	09/21/2020	11/30/2020
4.5	Develop routing page (where users enter what location they are routing to) prototype	70 days	09/28/2020	11/30/2020
4.6	Develop changes to layout, colors, and how instructions are presented depending on what is chosen on the accessibility page prototype	77 days	09/21/2020	11/30/2020
4.7	Finalize version of home page	56 days	01/11/2020	03/01/2021
4.8	Finalize legend page	56 days	01/11/2020	03/01/2021
4.9	Finalize report issues and about us page	56 days	01/11/2020	03/01/2021
5.0	Database	35 days	08/31/2020	10/05/2020
5.1	Set up Database following database diagrams	14 days	09/21/2020	11/05/2020
6.0	Network and Security	77 days	09/21/2020	11/30/2020

6.1	Set up security measures for account creation	77 days	09/21/2020	11/30/2020
7.0	Testing and Quality Assurance	133 days	11/09/2020	04/12/2021
7.1	Create use cases	14 days	09/09/2020	10/19/2020
7.2	Create test plan document	14 days	11/09/2020	11/23/2020
7.3	Create test execution document	14 days	11/09/2020	11/23/2020
7.5	Create manual test cases for routing to each location within the Arts and Sciences building	14 days	11/16/2020	12/07/2020
7.6	Manual test case for user home page (has saved routes, login optional)	14 days	12/07/2020	12/21/2020
7.7	Manual test cases for routing page (where users enter what location they are routing to)	14 days	01/04/2021	01/18/2021
8.0	Implementation & Set up	21 days	08/31/2020	09/14/2020
8.1	Find domain hosting / QR Code	21 days	11/01/2020	02/01/2021
8.2	Set up Hosting	7 days	3/01/2021	3/07/2021

Table 8

Problems Encountered and Analysis of Problems Solved

The development of the McMapping wayfinding application has led to a variety of what could be optimistically called “unplanned learning experiences,” or problems that slowed or altered McMapping’s development. Early on, many of these issues arose from the lack of ideal resources, access to outside contractors and their work, or lack of experience that prevented developing the project as originally envisioned. Presently more technical problems have stalled development, primarily due to compatibility issues between Vue.js Phaser, and Firebase.

The initial plan for the project hinged on getting access to the tools being used to develop a campus wide wayfinding application, and scale those to fit some of our needs, focusing the design to one building and mobility within using maps created in the design software AutoCAD. As the design and implantation of that wayfinding app was not being handled by in-house developers, and instead third-party contractors, the team was unable to gain access to their tools or learn from their design of their software. Alterations to the original plan were made; the primary method of determining user positions was changed from GPS to user input, maps were to be created in a game-map editor, and Phaser would be used to display the required map and route information to users.

Firebase Authentication

The team wanted to incorporate a login function so that users could create an account for personalization, this required us to authenticate users. The first problem that was encountered with implementing the firebase authentication feature was that no-one on the team had experience on implementing the authenticator into a web application. To solve this, team members worked diligently and utilized GitHub branches to collaborate on the issue. With research accompanied by trial and error; Firebase Authentication was successfully implemented to verify users.

Programming Language

One of the earliest design requirements was building a web application that could be accessed via a browser or downloaded as a mobile application. While McMapping uses Phaser for its wayfinding component, using Android Studio with Java or Kotlin would have left user on non-Android platforms unable to access the application. Ultimately Vue.js was determined to be the optimal environment to build McMapping in, as it gave a lot of development flexibility to meet the requirements of the project. Not everyone on the team was familiar with Vue.js so a technical deep dive into the Vue.js library to better understand the language and what it could do for the project became required reading.

Phaser Integration with Vue.js

Integrating the game engine, Phaser, with the JavaScript library Vue.js proved difficult at the best of times. Particularly with the sprite movement and camera access. Phaser functions had to access the camera from the game object in a non-traditional fashion due to the Vue.js wrapper. This was solved by debugging the code, stepping through the problem error by error until the movement to worked.

Tiled

The team wanted to use the more detailed object layer as the way the users would see the walls and obstacles on the map. The object layer was created by drawing polygons on top of the image layer that had been created from a floor plan provided by the ARCHIBUS map directory, so it was much closer to the actual floor plan than what a tile recreation would be. However, the object layer, while it could be seen clearly in Tiled software, cannot be rendered visible to the user on the webpage. Tiled object layers are only to be used for programming purposes and to layer tiles onto. So, for the time being, a purely tile based map will serve as the primary map in McMapping, although it is not as clean looking as the object layer was. Future iterations of the tile map will change the tiles used and how the tile layer is structured to improve readability, reliability, and load times.

Programming language syntax

One problem encountered during the first iteration of design was creating methods and functions inside of Vue, as it is a new experience developing in this framework. The developer created methods included in the project would throw an error in the browser. The problem was solved by several in-depth team mob programming sessions to put together the correct structure and syntax and location for using methods inside of Vue.

More Vue / Phaser compatibility issues

Due to the nature of how Phaser was implemented Some functionality did not work as intended, such as phaser graphics. In this case it meant that when drawing a route marker line in phaser standard methods for clearing that route became unavailable. This led to a much less hardware-efficient method in which route marking graphics were created and destroyed in the same update function, which was always running, rather than when they were specifically needed on function call.

Future recommendations/Recommendations for Improvement

What would do you if had it all over to do again?

- The user interface would boast a more user friendly and encouraging atmosphere. It would flow smoothly with phaser and optimize our key features. Phaser is the centerpiece of our project and the UI, in its current form, did not highlight all the major features implemented into the project.
- The frontend framework would be more compatible with the Phaser component.
- The backend database would be PostgreSQL instead of Firebase.

What would you do if you had more time?

- The technical debt in the code base should be reduced to promote scalability. Automated tests would be added to validate features.
- Move the coordinates from the application to the backend database.

- User interface registration immediately logs the user into their account. Ideally, upon registering a user account, the user would be directed to a confirmation view. The confirmation page would request the user to confirm their email address in-order to log into their account to prevent individuals from creating fake accounts.
- Add functionality such as routing around obstructions and make obstructed routes available online to other users.
- Ensure that all parts of the application are accessible, as currently there is extremely limited functionality for users who have impaired vision.

What suggestions have you had from others?

- Some user integration testers found it difficult to follow the route visually sometimes, and testers who used earlier version suggested reverting to having an on-screen icon animate down the route in the direction the user should travel as had been implemented earlier.
- Mobile versions are universally desired.
- Android Studio and Kotlin would have been a better choice than Vue.js for Phaser integration.
- Due to Charles McMicken's controversial past it was recommended that references to McMicken be avoided all together as that is the direction the University has decided to move.
- Per the provided survey, it was suggested that a 'go' button be added to the directions box in-order to provide a smooth transition between selecting a route and obtaining the accessible route.

What, if any, are your plans for this project?

- Offer the project as open source for any interested parties in utilizing it around their own campus, offices, or other large building complex.
- Offer the project to UCs accessible wayfinding team to see if it could be utilized in the broader accessibility strokes UC is implementing.

Conclusion

This year has been a series of overcoming challenges and learning curves, acquiring new skills, and succeeding at project goals through countless hours of trial and error. The lessons learned section below illustrates some of the many, many lessons learned along the way, as well as illuminate how the development process has changed.

Lessons Learned

Throughout this project, every member of the McMapping team have learned a variety of new skills while refining old ones. Programming tools and tricks, compatibility, testing, database management, version control and teamwork have all seen some level of new growth between the McMapping team's members.

There were many skills learned and used to implement Firebase user authentication. User authentication refers to registering new members by creating accounts, logging in to created accounts, and linking the user's Google account. One such skill learned included how Firebase can be utilized in a web application to authenticate users. Firebase authentication is versatile and adding additional means of authenticating users through their already existing accounts like Google, Twitter, or Facebook only requires a few lines of code.

All team members working on the project have become more familiar with using Git and GitHub technologies as version control software to tie the project together while hosting multiple branches for testing purposes. GitHub source control was invaluable, allowing for a collaborative workplace that would not be available in the 2020 climate otherwise. The main learning curve for using GitHub was the use of the command line instead of a more modern user interface. This took some time for many to get

comfortable with but was a good decision, and much more precise and practical than using one team member's IDE or the GitHub website or Desktop Application.

Learning about the finer details of Phaser was also a substantial portion of this application development. The tools and steps required to be able to import maps into our application, route a series of lines between coordinates that correspond to rooms in the Arts and Sciences building, have an icon that represents the user follow that routed line, load icons for restrooms and elevator, and use button on click events to move between floors were all especially important to getting McMapping's core functionality.

Team members that were working on the phaser component have learned to use phaser scenes to break down the application into separate sections that can operate independently while also conveying information between each other. The primary use of this has been to create separate floor scenes for separate pathfinding that still allows one floor to communicate with another. For instance: a "start" floor can communicate which elevator a route was supposed to take to a "destination" floor, or a menu scene can open a map scene and activate functions therein. Scenes are widely versatile and are well suited to building applications that deal with multiple input and informational displays simultaneously.

Vue.js is a JavaScript framework for building user interfaces that links the data and DOM so that everything is reactive, and you make changes to the data within the Vue instance. This project uses Vue components so that they are small and reusable, as well as to have the ability to make changes that will only affect the component and not the app (Single File Components, 2020).

Those working on the Vue.js learned how to work with Vue.js in an idiomatic way. We enhanced our skills and understanding of how to use components and how to create router views. For example, we gained the understanding of how to implement Phaser within a Vue.js application. Additionally, we

were able to use Bootstrap and Vue to create a reactive front-end design that will scale from browser to mobile with ease.

The backend map creation was done entirely through Tiled, a free to use 2D level editing software used primarily for map creation in game development (Tiled Documentation, 2019). In this case also works well for map creation in application development. For the purposes of this application that uses phaser and vue.js components, all maps were converted to JSON files to be read by Phaser. The map's frontend UI was redesigned in Photoshop, based off the maps saved in UC's ARCHIBUS system, which holds a catalogue of all the university's maps. ARCHIBUS was invaluable for the map creation in both Tiled and Photoshop.

The Phaser team, while working on the maps that are used in the Phaser game scenes, worked to create map files in the using the Tiled game level generating software. That team learned how to use the software to generate a map for each floor in the Arts and Sciences building with three layers: an image layer, an object layer, and a tile layer. Of important note is the object layer as it is crucial to map creation because objects can be coded with collision for another object to interact with. This is important for routing using phaser, so that the route that is being traced will not go through walls. The Phaser Team also learned how to implement the json file of the tiled maps within the phaser scenes which allowed the tile values to be read for use in computations and routing.

The Phaser team learned a lot about optimizing phaser code, how phaser draws frames of animation, and how limiting the frames of animation drawn per second can prevent phaser from overusing hardware resources.

References

- Gierdowski, D. C. (2019, October 30). *2019 Study of Undergraduate Students and Information Technology*. Retrieved from EduCause: <https://library.educause.edu/resources/2019/10/2019-study-of-undergraduate-students-and-information-technology>
- Henry, S. L. (2019, March 13). *W3C Accessibility Standards Overview*. Retrieved from The World Wide Web Consortium: <https://www.w3.org/WAI/standards-guidelines/>
- Single File Components*. (2020, November 29). Retrieved from Vue.js: <https://v3.vuejs.org/guide/single-file-component.html#introduction>
- Tiled Documentation*. (2019). Retrieved from Tiled: <https://doc.mapeditor.org/en/stable/>