UC Route Assistant

by

Jon Macbeth, Hayden Worrix, Adam Ralston

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

_____Hayden Worrix_____04/27/2020_____

Hayden Worrix Date

_____Adam Ralston_____04/27/2020_____

Adam Ralston Date

_____Jon Macbeth_____04/27/2020_____

Jon Macbeth Date

Technical Advisor: **__Abdou Fall_____04/27/2020_____**

Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
April 2020

Table of Contents

Section **Page**

Table of Contents (cont'd)

Section **Page**

List of Illustrations

Tables

Item **Page**

Figures

Item **Page**

Acronyms and Abbreviations

API: Application Programming Interface
GCP: Google Cloud Platform
IT: Information Technology
JSON: JavaScript Object Notation
MVC: Model View Controller
QA: Quality Assurance
SDK: Software Development Kit
UC: University of Cincinnati
UI: User Interface
UML: Unified Modeling Language

Abstract

UC Route Assistant is a mobile web application that allows the user to input their schedule and needs and determines the most optimal appropriate route for them to take on campus. This application was originally inspired by the vast quantity of new students coming to the University of Cincinnati each year. With over 7,000 new first year students in the 2019 fall semester, it is apparent that there is a need for assistance navigating the large campus. We built a mobile web application that enables the user to upload their course schedules, extracurricular activities, and accessibility preferences. UC Route Assistant processes this information and leverages Google Maps Application Programming Interface (API) to determine the fastest route that the user can take in order to reach their various destinations, while accounting for accessibility preferences and other major on-campus events. This results in less headaches having to reroute, less surprises running into blockades, and overall reduced travel time for the user.

1. Introduction

1.1 Problem

There are thousands of new students that are accepted into the University of Cincinnati each year. The 2019 school year has seen an enrollment of approximately 46,000 students, of which, around 8,000 are first years. This school year also serves to be the 7th year in a row that UC has broken their enrollment records (Suro, 2019, para. 2, 5). With so many of these students coming to UC for the first time, many are not familiar with the scope of campus and its layout. Even those who have visited campus before may find it challenging to navigate the hilly terrain of campus. Each semester offers new challenges to students navigating to new classes located in unfamiliar buildings. In addition to these new students, there are numerous other students who have disabilities that may require specialized routes while traversing the campus. The problem is that there is not an application that individualizes and routes for its users, all while taking into account class schedules, campus events, and other specialized preferences. A simple tool that could determine and present the most efficient routes for individuals would greatly benefit the University of Cincinnati community.

1.2 Solution

UC Route Assistant is a mobile web application that allows the user to input their schedule and needs and determines the most optimal and appropriate route for them to take. It can factor in things outside of classes such as clubs, events, eating, or other activities. It can also determine specialized routes for students who have mobile disabilities and are unable to use staircases. This application is beneficial for new students who are unfamiliar with the campus, students with certain specialized needs, and those who want to optimize their schedule and reduce travel time.

1.3 Project Goals & Methodology

The goal of this project was to solve the problem stated in section 1.2. We designed an easy to use mobile application that would enable users to input schedules and events and allow them to optimize their travel more effectively, as well as schedule and plan for multiple events ahead of time. The number one requirement and goal of this application was to assist users in conveniently and effectively planning out their daily routines at UC, with travel times included. Another important goal of this project was for the application to be useful to both new and seasoned individuals at UC alike, with features that benefit all groups of people. Our developmental methodology can be described as waterfall development, as we incorporated sequential development steps to complete this project.

1.4 Overview

The remainder of this final report outlines in detail how the project was completed. The report includes the following sections: project concept, design objectives, methodology, technical

discussion, testing, budget, timeline, problems encountered, future recommendations, and our conclusions.

2. Discussion

2.1 Project Concept & Solution

The use of a mobile application to navigate is already a familiar concept for most Americans with the use of Google Maps. Recent research shows that in 2019 93% of Americans between the ages of 23 and 38 owned a smartphone as well as 90% of Americans between ages 39 and 54, 68% of Americans between ages 55 and 73, and 40% of Americans between ages 74 and 91 (Vogels, 2019, para. 2). This data shows that the prevalence of smartphones makes it an ideal platform for navigation and their portability is convenient while traveling on foot.

However, utilizing the Google Maps API to build wayfinding mobile applications that are designed to focus on specific locales, such as a campus, is not necessarily a novel idea. For example, a "smart campus map" in the form of a hybrid Android app for the Federal University of Technology's Bosso Campus in Nigeria was created for the purposes of aiding students as they navigate campus (Odekunle, 2018). A similar "mobile campus touring system" was created for Fu-Jen University in Taiwan, which also employed the use of augmented reality to supply students with additional information about buildings, such as the opening hours of a library or general details about a building (Chou, 2012). The SmartUJI mobile applications, as described in the journal article "Enhancing integrated indoor/outdoor mobility in a smart campus," also utilizes an augmented reality user interface for its map utility. These applications are also part of a middleware

platform that can consume data and services from other sources to better inform the user with contextual details (Torres, 2015).

Though such applications may already exist, none of them are tailored to the University of Cincinnati the way ours is. Our application similarly possesses the ability to consume other web or data sources at UC to integrate their information with the application. For instance, event notifications could be sent to the app that would update the map to show where the event will take place, warn of potential areas of congestion, and assist the user in navigating through or even avoiding the traffic altogether.

One of the most important distinctions of all is that our app is developed to better serve users with mobility challenges. Other applications similar to ours were designed for the general public, but the intention of ours is to also help those who may be using wheelchairs or at least have difficulty walking up or down sharp inclines or stairs. The University of Cincinnati main campus contains a significant number of hills and stairs, so this has added benefits for those at UC. Filters could be created to adjust routes based on mobility restrictions as well as to serve other needs such as finding paths more suited for inclement weather or more suited for nighttime navigation.

This app was first proposed by Hayden Worrix and quickly caught the interest of the rest of the team. Each of the team members could relate to the challenges of navigating the main campus at UC, especially when special events are going on. It was also obvious how the hilly terrain and numerous stairs present on campus could present additional challenges to students and visitors with mobile disabilities. Each of the team members proposed filters that could be applied to the app that would find the most optimal route such as consideration of events being held on campus, construction being performed, inclement weather, or to account for terrain obstacles for users with mobility challenges. Additional ideas were proposed to utilize a camera on mobile

platforms to input class schedules, potentially use augmented reality to better assist users, and utilize layouts of the building to provide additional assistance navigating the buildings themselves, not just the campus.

2.2 Design Objectives

There were several design objectives that we created after the initial inception of our project. Our first goal was to create a functioning mobile application that allowed users to interact with the app and understand the time it would take for them to get from point A to point B, on UC's campus. Another goal that we had was for the application to enable users with mobile disabilities to better plan their routes in accordance with the types of routes they are capable of traversing. Additionally, our team wanted the application to assist users in scheduling their days with regards to their destinations and intended arrival times, such as in the case of a student with multiple classes throughout the day. We also had some initial goals that had to be abandoned due to time constraints and developmental challenges, such as implementing augmented reality within the application to further assist the user in their navigation. This feature was originally intended to not only guide the user to their destination, but also assist the user in navigating specific buildings and help locate the exact rooms they wanted to get to within the buildings. We also had to abandon a more robust rollout of the accessibility features for handicap users, as this would have required us to manually map the entirety of UCs campus and code in specialized routes and blocks for each staircase on the campus.

2.3 Methodology & Technical Approach

Our developmental methodology and design requirements allowed us to meet our goals by

ensuring that we had a clear scope of what we were trying to accomplish, and how it was intended to be built. Specifically, by noting the initial design requirements of our application we were able to create bare proof of concept prototypes that allowed us to explore how the various intended functionalities were meant to interact with each other. The developmental procedure that we followed, known as waterfall development, can be described as a sequential building of application framework and functionality. This sequential building allowed us to work from a very basic application with limited functionality, to a more robust application with various interconnected parts and features for the end user to utilize.

2.4 User Profiles

User profiles are descriptions of groups of people who are expected to use, interact with, or manage the UC Route Assistant application.

2.4.1 All Profiles:

Students and Staff – Most frequent users of application, primary intended end user

Visitors – Additional users of application, those who are only visiting UC temporarily

Administrators – Backend managers who have access to change and update the application

2.4.1.1 Application Title:

UC Route Assistant

2.4.1.2 Experience with Similar Applications:

- Google Maps
- MapQuest

- Waze

2.4.2 User Profile for Students and Staff:

2.4.2.1 Background:

The first user profile is students, particularly new students at the University of Cincinnati's main campus. Included in this user profile are faculty members that are similarly high frequency users of the campus. All users would be owners of mobile devices, such as smartphones, who would be interested in navigating to specific areas of the campus to attend classes, work, or events.

2.4.2.2 Software, Interface, and Related Experience:

It will be expected that users who own and regularly use smartphones and small tablets will be the ones utilizing this application. They should be familiar with mobile applications, particularly ones used for navigation, such as Google Maps.

2.4.2.3 Task Experience:

The user will be presented with a simple UI that will enable them to input the name of the building to which they wish to travel. After entering the building's name, the user may select filters based on their preferences. These may include finding routes that are more suitable for individuals in wheelchairs, routes that will offer better protection from adverse weather, and routes that bring the user near specific sites along the way such as near the bookstore or places to eat. The application will also warn users about events and can reroute them to avoid congestion. These features can be used by both visitors to main campus as well as students. However, students can also create an account and upload their schedule so they may be presented with an efficient route

every day that will take them from building to building per their daily schedule for that semester. Additional details about the buildings and routes to take within the buildings will be provided by linked information to destination sites on the map.

2.4.2.4 Frequency of Use:

We expect that this mobile application will be frequently used by students at the start of each new semester and by freshmen new to campus. Staff members will have similar frequency of use to students. Many different students on campus will have various temporary uses. Some will want to attend events and just want to find out how to get to the stadium. A staff member might also use the application for a few days that they are at the campus such as various employees contracted through the University of Cincinnati. Some students and staff may only use the visitor access if their regular parking spot is inaccessible.

2.4.2.5 Key Project Design Requirements That the Profile Suggests:

- Present a clean, simple UI that appeals to visitors of older generations as well as younger generations of students.
- Keep accessibility in mind for the application by including an accessibility filter.
- Provide useful filters that can improve the experience of navigating with the app.
- Give students the option to create accounts to maintain their schedule on the system, so the application utilizes this schedule by default.

2.4.3 User Profile for Visitors:

2.4.3.1 Background:

The second user profile is for guests and visitors who are not familiar with campus or who need more temporary access. These guests receive routing assistance and can still adjust constraints to navigate to their destination, such as avoiding inclement weather or finding the best route suitable for a wheelchair, or for those who have specific desires, such as a route that is close to places to eat. All users would be owners of mobile devices, such as smartphones, who would be interested in navigating to specific areas of the campus to attend classes or events.

2.4.3.2 Software, Interface, and Related Experience:

It will be expected that users who own and regularly use smartphones and small tablets will be the ones utilizing this application. They should be familiar with mobile applications, particularly ones used for navigation such as Google Maps.

2.4.3.3 Task Experience:

The user will be presented with a simple UI that will enable them to input the name of the building to which they wish to travel. After entering the building's name, the user may select filters based on their preferences. These may include finding routes that are more suitable for individuals in wheelchairs, routes that will offer better protection from the weather, and routes that bring the user near specific sites along the way such as near the bookstore or places to eat. The application will also warn users about events and can reroute them to avoid congestion. Additional details

about the buildings and routes to take within the buildings will be provided by linked information to destination sites on the map.

2.4.3.4 Frequency of Use:

Many different visitors to campus will have varying uses that may be temporary depending on the frequency of their visits to campus. Some visitors will come to attend events and just want to find out how to get to the stadium. A visitor might also use the application for a few days while they are on campus temporarily, such as various employees contracted through the University of Cincinnati. Others may only use the visitor access if their regular parking spot is inaccessible.

2.4.3.5 Key Project Design Requirements That the Profile Suggests:

- Present a clean, simple UI that appeals to visitors of older generations as well as younger generations.
- Keep accessibility in mind for the application by including an accessibility filter.
- Provide useful filters that can improve the experience of navigating with the app.

2.4.4 User Profile for Administrators:

2.4.4.1 Background:

The back-end user consists of the administrators who will update data and features of the system for better usability throughout the life cycle of the application. An example of data administrators may upload are blockages around the university, such as sporting events that obstruct different paths around campus. Additional data examples include points of interest around

campus, such as new buildings and businesses, or even construction on campus that will impact routes for an extended period of time.

2.4.4.2 Software, Interface, and Related Experience:

The administrator should have familiarity with web services and the Google Maps API. They should be familiar with mobile applications, particularly ones used for navigation such as Google Maps.

2.4.4.3 Task Experience:

He or she will be presented with a UI with unique access to send notifications to end users and make data updates. The administrator will also be responsible for deploying feature changes to the application.

2.4.4.4 Frequency of Use:

We do not expect that the administrator will frequently interact with the application except when it is time to promote major feature updates. The administrator may send notifications from time to time alerting end users about events or situations on campus that may warrant their immediate attention, or to supply data updates.

2.4.4.5 Key Project Design Requirements That the Profile Suggests:

- Present a clean, simple UI with intuitive controls to send notifications and updates.
- Create secure accessibility unique to the administrator's role.

2.5 Use Case Diagram

Figure 1: Use Case Diagram below depicts a use case diagram which shows the intended interactions of the various users and groups within the application.

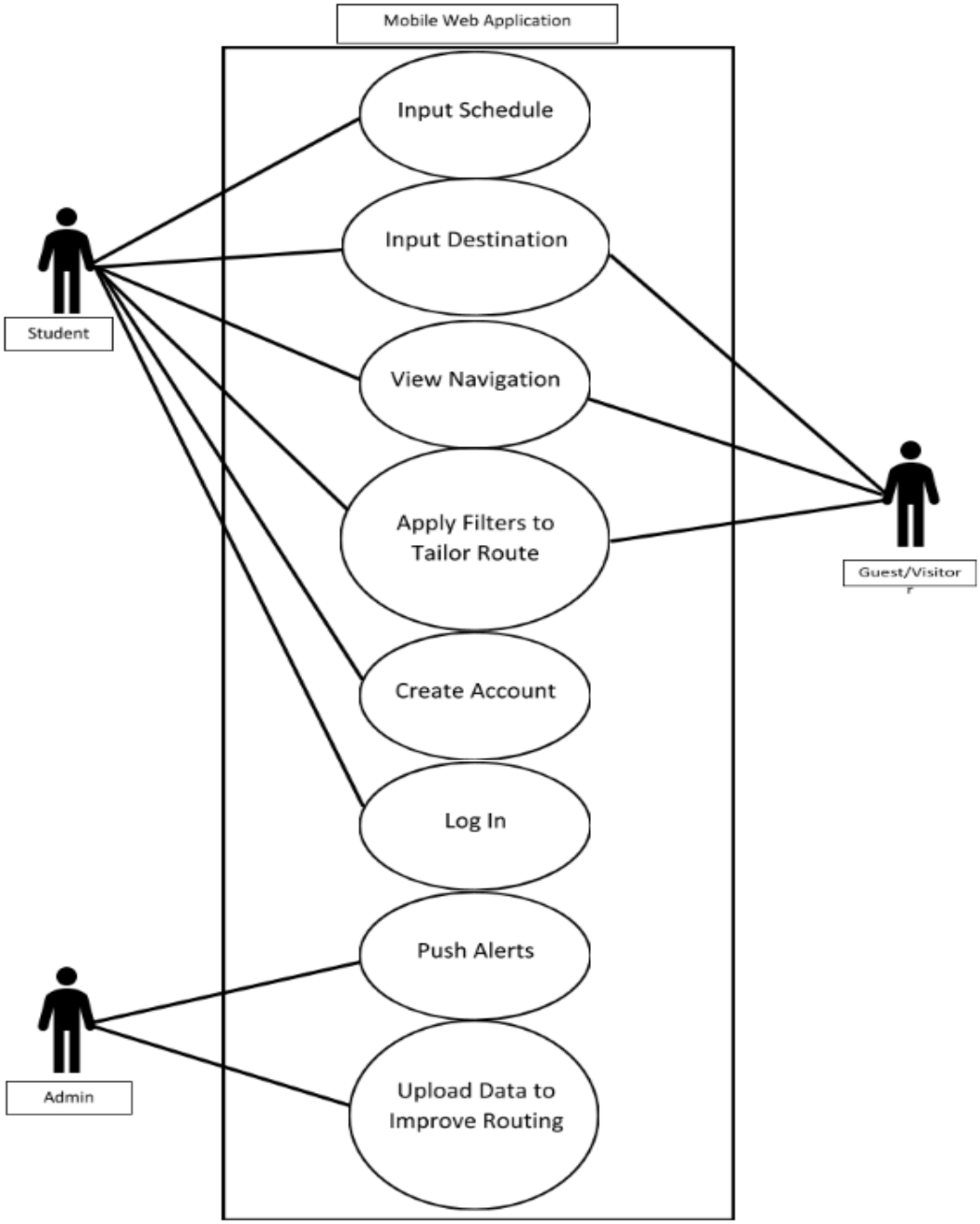


Figure 1: Use Case Diagram

2.6 Technical Discussion

2.6.1 Hardware/Infrastructure

We used Google Cloud Platform (GCP) for our infrastructure needs. Primarily, we used Google's Firebase Authentication and Firebase Cloud Storage. Hardware is limited to the user's Android phones and emulated Android devices on our development environments.

2.6.2 Application

Our mobile application is built in Android Studio on Windows desktops. Originally, development began in Xamarin.Forms for cross platform compatibility but none of our group members had Macintosh computers or Apple phones to develop and test the iOS portion of the application. Android Studio offered better compatibility with external libraries we used, such as Google Maps and Firebase. In general, our application is a Java based application designed with a model view controller architecture, otherwise known as an MVC design. This is made possible with Android classes for fragments and activities for the view and controller aspects. The core functionality for mapping in our application comes from Google Maps Software Development Kit (SDK) for Android. It allows us to showcase the map of UC and add custom locations for the different buildings and places at UC. User sign in and authentication is handled using Google's Firebase API. This allows us to provide a secure sign in method and even third-party sign in options. We are also using Google's Firebase Realtime Database API to store scheduling information for added courses. We have two primary activities, one handles the login aspect and, if a successful sign in occurs, the user proceeds to the main application activity. Screenshots of the applications log in, settings, maps, and schedule screens can be found below, in **Figures 2-5**.

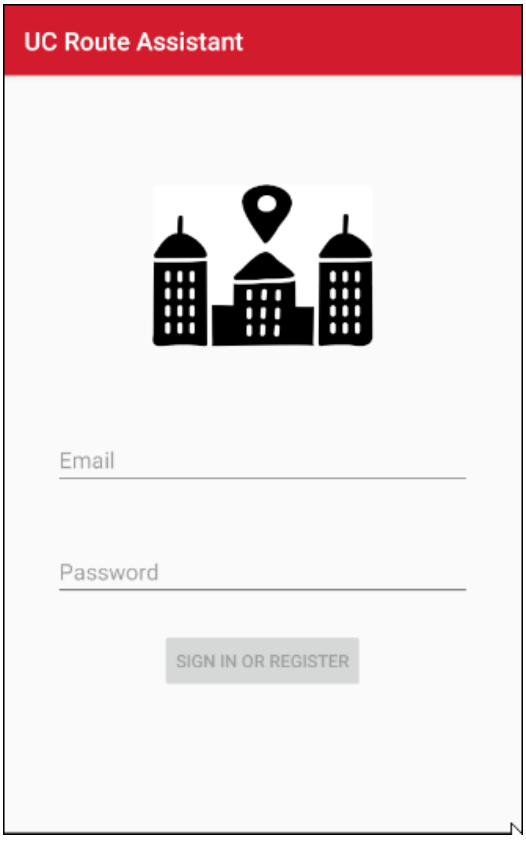


Figure 2: Log In Screen

Figure 3: Settings Screen

Figure 4: Maps Screen

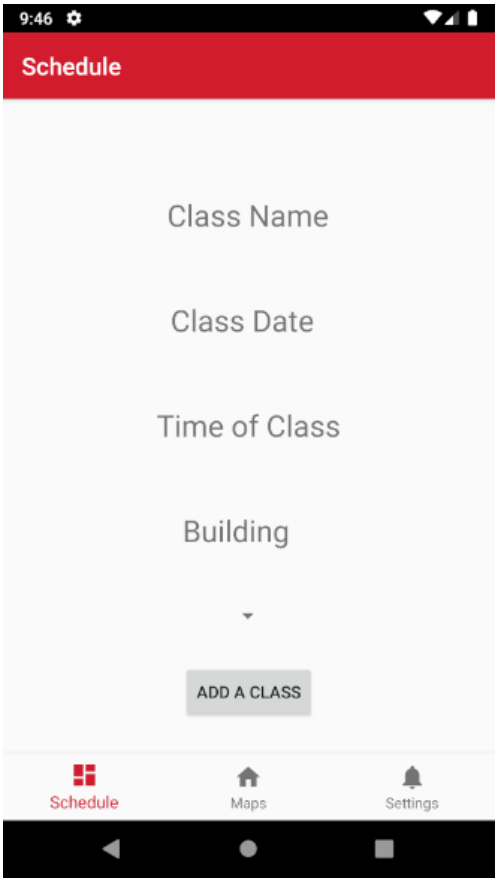


Figure 5: Schedule Screen

2.6.3 Database

We have two databases both hosted on Google Cloud Platform, a virtual platform that is known for reliability and speed. The first is our authentication database that stores user sign in information and handles third party sign in linkage. The second database is a NoSQL database hosted on GCP, which handles storage of JSON files containing user added classes.

2.6.4 Security

Most of the security of our application is handled using Google's services. Google's Firebase Authentication handles the transmission and storage of user information safely and securely. Specifically encrypting the API calls made from the local client to the database. Our local application code uses proper encapsulation to take advantage of the object-oriented nature of Java.

2.6.5 Technical Architecture Diagram

Figure 6: Technical Architecture Diagram below is a graphical representation of the technical architecture of our application.

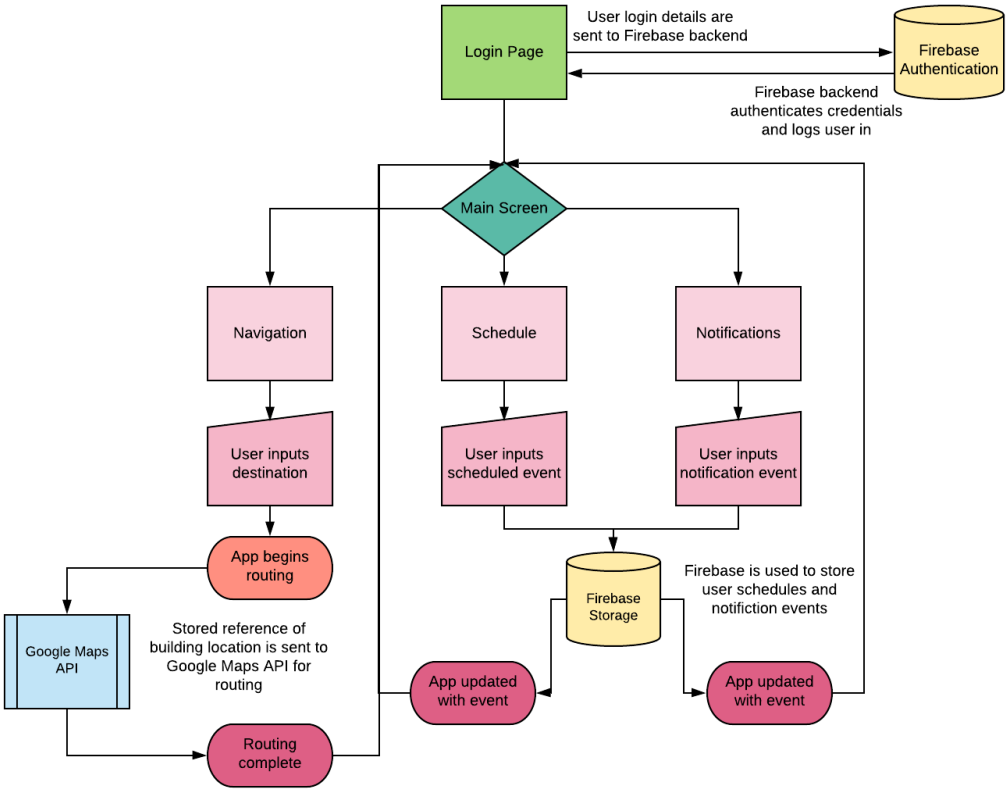


Figure 6: Technical Architecture Diagram

2.7 Testing

2.7.1 Overview and Methodology

This testing plan contains details about the tests that were conducted on UC Route Assistant in order to ensure that the application functioned as intended in as many test cases as possible. These tests were conducted from the various viewpoints of our expected users as per our user profiles:

- Students and Staff at UC
- Visitors
- Administrators

Our methodology for testing included a four-stage series of tests in order to ensure that a maximum number of issues were found and remediated before the final release of our application. These stages are as follows:

1. Developers ran functionality/QA tests as they built each individual feature of the application, ensuring that the feature is operational before finalizing the initial publication of the feature. They also made sure to do an ad-hoc test around different parts of the application that new functionality might impact.
2. Other team members ran functionality/QA tests as each feature was published, in order to ensure that the quality of these features is up to the expectations of ourselves, and our users.
3. The team methodically tested and recorded the results of each documented test case that was prepared regarding our initial objectives and our applications goals. As we tested, we did rounds of regression testing to verify older features were still functional as we progressed.
4. Finally, the team facilitated less standardized third-party testing as well, in order to determine if there were any outlying scenarios that our test cases were unable to identify. We had friends and other people not involved with the development of the application test to see if there were scenarios we as developers had missed.

2.7.2 Scope of Testing

The scope of our testing covered all features of our application that were developed and

intended to be published in our final product. Basic functionality such as application launching to more intermediate functionality such as menu navigation, and finally advanced functionality such as directional pathing were all tested. It is important to note that these features were tested and observed from the various viewpoints of our users, as there are different nuances between each of the user groups that could create different results for certain test cases depending on the user group. **Table 1: Test Plan** below shows the specific and complete scope of our testing scenarios to date.

TEST CASE	TIMES TESTED	PASS	FAIL	PASS/FAIL CONDITIONS
Application launches as intended				**Pass:** No crashes or freezes, launches initial app screen
Application is stable				**Pass:** No crashes or freezes in 1 hr sessions
Create account works as intended				**Pass:** User is able to make account that application then recognizes
Log in works as intended				**Pass:** User is able to log in
Log out works as intended				**Pass:** User is able to return to log in screen via log out button
Menu navigation works as intended				**Pass:** User is able to use any navigation button and it redirects through the different screens
User interface displays as intended				**Pass:** UI elements are sized and aligned appropriately, as agreed upon by the team
Inputting classes/destinations works as intended				**Pass:** User is able to input destinations

Clearing classes works as intended				**Pass:** User is able to clear classes
Route navigation works as intended				**Pass:** User is able to follow route navigation to reach destinations
Customizable notifications work as intended				**Pass:** User is able to set how long before each class start to be notified, and notification comes at this time

Table 1: Test Plan

2.7.3 Objectives

Our testing efforts were created and executed in order to achieve the following objectives:

1. All individual application features must be operational to the point where there are no major failure pathways possible for the end user.
2. All major application features must be operational for each specific user group, where appropriate.
    a. Students and Staff at UC
    b. Visitors
    c. Administrators
3. All new issues identified while conducting tests or remediating issues are to be documented for further analysis of the issue.
4. All documented issues mentioned above are to be reviewed and either fixed or labelled as insignificant before IT Expo.
5. All QA test cases must be agreed upon as passing by all group members before IT Expo.

6. All major issues or major failure pathways in the application workflow are to be remediated before the final presentation of UC Route Assistant at the IT Expo.

2.7.4 Logging Test and Procedures

In order to properly conduct these tests, we had to first establish pass/fail conditions for the various components and features being tested. Following this, we organized these tests and conditions into tables so that we were able to more efficiently organize our thoughts and results. As previously stated, we approached these tests in a four-staged series in order to minimize the possibility of a major bug getting by undetected by IT Expo.

If a bug was found to exist, one of the developers took ownership of remediating that bug and communicated this to the rest of the team, asking for assistance/clarification when necessary. Additionally, each team member was asked to perform functionality and quality assurance testing on the application during the months preceding the IT Expo. This was in effort to complete all of our documented test cases. We did not have a set schedule for these operations as our individual schedules varied in terms of availability. Each team member recorded the tests that were performed and the results of those tests after completion.

Finally, following the completion and remediation of our documented test cases, each team member was encouraged to perform less standardized at-will testing, where we explicitly attempted to “break” the application with unexpected inputs and workflows. This portion of our testing also included presenting the application to third-party friends and family to see if they found any issues that we missed.

2.7.5 Test Results

Table 2: Test Results below shows the results of our Test Plan.

TEST CASE	TIMES TESTED	PASS	FAIL	PASS/FAIL CONDITIONS
Application launches as intended	**10**	✓		**Pass:** No crashes or freezes, launches initial app screen
Application is stable	**10**	✓		**Pass:** No crashes or freezes in 1 hr sessions
Create account works as intended	**10**		✗	**Pass:** User is able to make account that application then recognizes
Log in works as intended	**10**	✓		**Pass:** User is able to log in
Log out works as intended	**10**	✓		**Pass:** User is able to return to log in screen via log out button
Menu navigation works as intended	**10**	✓		**Pass:** User is able to use any navigation button and it redirects through the different screens
User interface displays as intended	**10**	✓		**Pass:** UI elements are sized and aligned appropriately, as agreed upon by the team
Inputting classes/destinations works as intended	**10**	✓		**Pass:** User is able to input destinations
Clearing classes works as intended	**10**	✓		**Pass:** User is able to clear classes
Route navigation works as intended	**10**	✓		**Pass:** User is able to follow route navigation to reach destinations

Customizable notifications work as intended	**10**		✕	**Pass:** User is able to set how long before each class start to be notified, and notification comes at this time

Table 2: Test Results

2.7.6 What we learned during testing

During testing we learned that our application had various issues related to interconnected functionality of the different parts of the application. We also ran into a couple of issues testing customized routing with our initial framework platform of Xamarin.Forms and, following this testing, decided to pivot to using Android Studio for development. This was not only a lesson in developing this application specifically, but also a lesson in general when it comes to application development. We learned that the earlier you can test key features of your application, the earlier you are able to identify unexpected roadblocks. Through our testing we were able to conclusively decide on a change in framework, which allowed us to proceed with the development instead of remaining stuck on one specific key feature.

In addition, we learned that testing takes much more time than we initially expected. We originally planned on doing testing in an almost entirely ad-hoc fashion, without any scheduling or planning, but we quickly learned that this approach was more difficult than a reasonably structured testing plan. This change in our testing approach enabled us to identify a number of key issues with our application and allowed us to make the necessary changes moving forward with development.

2.8 Budget

All of the technical elements utilized in the development of UC Route Assistant were free

for the tier of product that we used. If we had not been able to use various Google products such as Google Maps API, Firebase Authentication, and Firebase Storage, we would have likely had a substantially increased project budget. As we were students completing a course requirement for senior design, we weren't being paid for our work on this project, but we calculated our initial budget as though development, research, and planning were paid labor. We estimate that the application took around 50 hours to complete per group member through research and development. At an average wage of $20/hr for 3 group members, we calculated a total initial projected labor cost of $3000. **Table 3: Budget** below shows our initial estimated costs as well as our final actual costs for the development of UC Route Assistant.

ITEM	INITIAL ESTIMATED COST	ACTUAL FINAL COST
Hardware	$0.00	$0.00
Software	$0.00	$0.00
License Upgrades	$0.00	$0.00
Presentation	$25.00	$0.00
Labor	$3000.00	$0.00
Total Costs	**$3025.00**	**$0.00**

Table 3: Budget

2.9 Objectives/Deliverables

Table 4: Project Milestones & Deliverables below depicts the project deliverables and deadlines. The left side of the table shows the milestone and the right side of the table shows the date this milestone was accomplished.

MILESTONE	DATE
Fall Semester 2019	**8/26/19-12/14/19**
Initiation	9/16/19

Team Contract	9/23/19
Project Abstract	10/14/19
Research	10/14/19
User Profile	10/21/19
Elevator Speech	10/21/19
Fall Draft Report	11/10/19
Design	11/18/19
Security	11/18/19
Implementation	12/14/19
Spring Semester 2020	**1/13/20-4/30/20**
Equipment Request Form	2/3/2020
Testing Report	2/10/2020
Revised Abstract	2/17/2020
Draft Poster	3/2/2020
Final Poster	3/16/2020
Final Presentation	3/31/2020
Final Report	4/13/2020
Tech Expo	4/14/2020

Table 4: Project Milestones & Deliverables

2.10 Project Schedule

Table 5: Work Breakdown Structure and **Figure 7: Gantt Chart** below depict the project's timeline and development schedule.

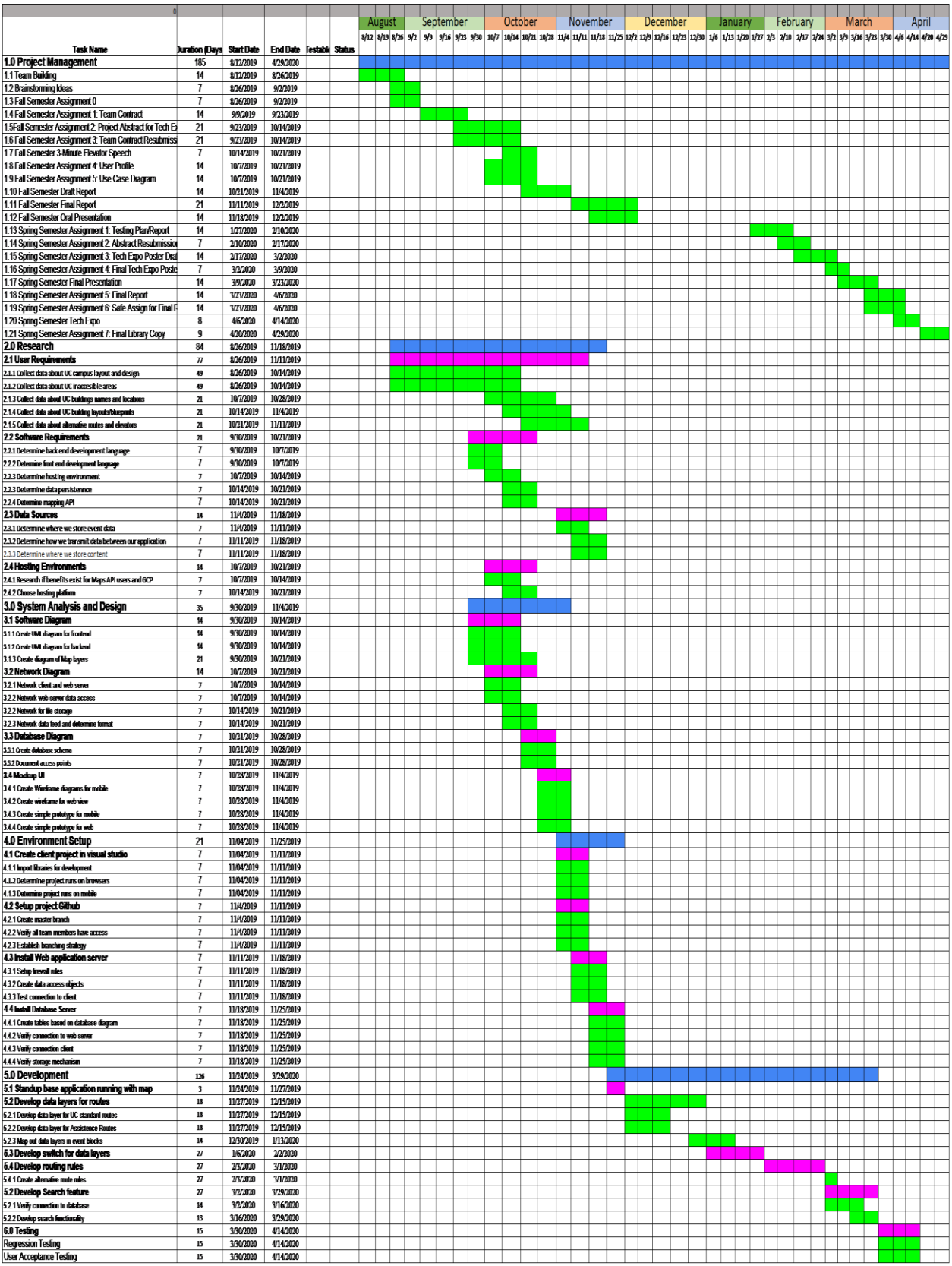
TASK NAME	DURATION (DAYS)	START DATE	END DATE
1.0 Project Management	185	8/12/2019	4/29/2020
1.1 Team Building	14	8/12/2019	8/26/2019
1.2 Brainstorming Ideas	7	8/26/2019	9/2/2019
1.3 Fall Semester Assignment 0	7	8/26/2019	9/2/2019
1.4 Fall Semester Assignment 1: Team Contract	14	9/9/2019	9/23/2019
1.5 Fall Semester Assignment 2: Project Abstract for Tech Expo	21	9/23/2019	10/14/2019
1.6 Fall Semester Assignment 3: Team Contract Resubmission	21	9/23/2019	10/14/2019
1.7 Fall Semester 3-Minute Elevator Speech	7	10/14/2019	10/21/2019
1.8 Fall Semester Assignment 4: User Profile	14	10/7/2019	10/21/2019
1.9 Fall Semester Assignment 5: Use Case Diagram	14	10/7/2019	10/21/2019
1.10 Fall Semester Draft Report	14	10/21/2019	11/4/2019
1.11 Fall Semester Final Report	21	11/11/2019	12/2/2019
1.12 Fall Semester Oral Presentation	14	11/18/2019	12/2/2019
1.13 Spring Semester Assignment 1: Testing Plan/Report	14	1/27/2020	2/10/2020
1.14 Spring Semester Assignment 2: Abstract Resubmission	7	2/10/2020	2/17/2020
1.15 Spring Semester Assignment 3: Tech Expo Poster Draft	14	2/17/2020	3/2/2020
1.16 Spring Semester Assignment 4: Final Tech Expo Poster	7	3/2/2020	3/9/2020
1.17 Spring Semester Final Presentation	14	3/9/2020	3/23/2020
1.18 Spring Semester Assignment 5: Final Report	14	3/23/2020	4/6/2020
1.19 Spring Semester Assignment 6: Safe Assign for Final Report	14	3/23/2020	4/6/2020
1.20 Spring Semester Tech Expo	8	4/6/2020	4/14/2020

1.21 Spring Semester Assignment 7: Final Library Copy	9	4/20/2020	4/29/2020
2.0 Research	84	8/26/2019	11/18/2019
2.1 User Requirements	77	8/26/2019	11/11/2019
2.1.1 Collect data about UC campus layout and design	49	8/26/2019	10/14/2019
2.1.2 Collect data about UC inaccessible areas	49	8/26/2019	10/14/2019
2.1.3 Collect data about UC buildings names and locations	21	10/7/2019	10/28/2019
2.1.4 Collect data about UC building layouts/blueprints	21	10/14/2019	11/4/2019
2.1.5 Collect data about alternative routes and elevators	21	10/21/2019	11/11/2019
2.2 Software Requirements	21	9/30/2019	10/21/2019
2.2.1 Determine back end development language	7	9/30/2019	10/7/2019
2.2.2 Determine front end development language	7	9/30/2019	10/7/2019
2.2.3 Determine hosting environment	7	10/7/2019	10/14/2019
2.2.3 Determine data persistence	7	10/14/2019	10/21/2019
2.2.4 Determine mapping API	7	10/14/2019	10/21/2019
2.3 Data Sources	14	11/4/2019	11/18/2019
2.3.1 Determine where we store event data	7	11/4/2019	11/11/2019
2.3.2 Determine how we transmit data between our application	7	11/11/2019	11/18/2019
2.3.3 Determine where we store content	7	11/11/2019	11/18/2019
2.4 Hosting Environments	14	10/7/2019	10/21/2019
2.4.1 Research if benefits exist for Maps API users and GCP	7	10/7/2019	10/14/2019
2.4.2 Choose hosting platform	7	10/14/2019	10/21/2019
3.0 System Analysis and Design	35	9/30/2019	11/4/2019
3.1 Software Diagram	14	9/30/2019	10/14/2019
3.1.1 Create UML diagram for frontend	14	9/30/2019	10/14/2019

3.1.2 Create UML diagram for backend	14	9/30/2019	10/14/2019
3.1.3 Create diagram of Map layers	21	9/30/2019	10/21/2019
3.2 Network Diagram	14	10/7/2019	10/21/2019
3.2.1 Network client and web server	7	10/7/2019	10/14/2019
3.2.2 Network web server data access	7	10/7/2019	10/14/2019
3.2.2 Network for file storage	7	10/14/2019	10/21/2019
3.2.3 Network data feed and determine format	7	10/14/2019	10/21/2019
3.3 Database Diagram	7	10/21/2019	10/28/2019
3.3.1 Create database schema	7	10/21/2019	10/28/2019
3.3.2 Document access points	7	10/21/2019	10/28/2019
3.4 Mockup UI	7	10/28/2019	11/4/2019
3.4.1 Create Wireframe diagrams for mobile	7	10/28/2019	11/4/2019
3.4.2 Create wireframe for web view	7	10/28/2019	11/4/2019
3.4.3 Create simple prototype for mobile	7	10/28/2019	11/4/2019
3.4.4 Create simple prototype for web	7	10/28/2019	11/4/2019
4.0 Environment Setup	21	11/04/2019	11/25/2019
4.1 Create client project in visual studio	7	11/04/2019	11/11/2019
4.1.1 Import libraries for development	7	11/04/2019	11/11/2019
4.1.2 Determine project runs on browsers	7	11/04/2019	11/11/2019
4.1.3 Determine project runs on mobile	7	11/04/2019	11/11/2019
4.2 Setup project Github	7	11/4/2019	11/11/2019
4.2.1 Create master branch	7	11/4/2019	11/11/2019
4.2.2 Verify all team members have access	7	11/4/2019	11/11/2019
4.2.3 Establish branching strategy	7	11/4/2019	11/11/2019
4.3 Install Web application server	7	11/11/2019	11/18/2019
4.3.1 Setup firewall rules	7	11/11/2019	11/18/2019
4.3.2 Create data access objects	7	11/11/2019	11/18/2019
4.3.3 Test connection to client	7	11/11/2019	11/18/2019

4.4 **Install Database Server**	7	11/18/2019	11/25/2019
4.4.1 Create tables based on database diagram	7	11/18/2019	11/25/2019
4.4.2 Verify connection to web server	7	11/18/2019	11/25/2019
4.4.3 Verify connection client	7	11/18/2019	11/25/2019
4.4.4 Verify storage mechanism	7	11/18/2019	11/25/2019
5.0 Development	126	11/24/2019	3/29/2020
5.1 Standup base application running with map	3	11/24/2019	11/27/2019
5.2 Develop data layers for routes	18	11/27/2019	12/15/2019
5.2.1 Develop data layer for UC standard routes	18	11/27/2019	12/15/2019
5.2.2 Develop data layer for Assistance Routes	18	11/27/2019	12/15/2019
5.2.3 Map out data layers in event blocks	14	12/30/2019	1/13/2020
5.3 Develop switch for data layers	27	1/6/2020	2/2/2020
5.4 Develop routing rules	27	2/3/2020	3/1/2020
5.4.1 Create alternative route rules	27	2/3/2020	3/1/2020
5.5 Develop Search feature	27	3/2/2020	3/29/2020
5.5.1 Verify connection to database	14	3/2/2020	3/16/2020
5.5.2 Develop search functionality	13	3/16/2020	3/29/2020
6.0 Testing	15	3/30/2020	4/14/2020
6.1 Regression Testing	15	3/30/2020	4/14/2020
6.2 User Acceptance Testing	15	3/30/2020	4/14/2020

Table 5: Work Breakdown Structure



Task Name	Duration (Days	Start Date	End Date	Testable	Status
1.0 Project Management	185	8/12/2019	4/29/2020		
1.1 Team Building	14	8/12/2019	8/26/2019		
1.2 Brainstorming Ideas	7	8/26/2019	9/2/2019		
1.3 Fall Semester Assignment 0	7	8/26/2019	9/2/2019		
1.4 Fall Semester Assignment 1: Team Contract	14	9/9/2019	9/23/2019		
1.5Fall Semester Assignment 2: Project Abstract for Tech Ex	21	9/23/2019	10/14/2019		
1.6 Fall Semester Assignment 3: Team Contract Resubmissi	21	9/23/2019	10/14/2019		
1.7 Fall Semester 3-Minute Elevator Speech	7	10/14/2019	10/21/2019		
1.8 Fall Semester Assignment 4: User Profile	14	10/7/2019	10/21/2019		
1.9 Fall Semester Assignment 5: Use Case Diagram	14	10/7/2019	10/21/2019		
1.10 Fall Semester Draft Report	14	10/21/2019	11/4/2019		
1.11 Fall Semester Final Report	21	11/11/2019	12/2/2019		
1.12 Fall Semester Oral Presentation	14	11/18/2019	12/2/2019		
1.13 Spring Semester Assignment 1: Testing PlanReport	14	1/27/2020	2/10/2020		
1.14 Spring Semester Assignment 2: Abstract Resubmissio	7	2/10/2020	2/17/2020		
1.15 Spring Semester Assignment 3: Tech Expo Poster Drat	14	2/17/2020	3/2/2020		
1.16 Spring Semester Assignment 4: Final Tech Expo Poste	7	3/2/2020	3/9/2020		
1.17 Spring Semester Final Presentation	14	3/9/2020	3/23/2020		
1.18 Spring Semester Assignment 5: Final Report	14	3/23/2020	4/6/2020		
1.19 Spring Semester Assignment 6: Safe Assign for Final F	14	3/23/2020	4/6/2020		
1.20 Spring Semester Tech Expo	8	4/6/2020	4/14/2020		
1.21 Spring Semester Assignment 7: Final Library Copy	9	4/20/2020	4/29/2020		
2.0 Research	84	8/26/2019	11/18/2019		
2.1 User Requirements	77	8/26/2019	11/11/2019		
2.1.1 Collect data about UC campus layout and design	49	8/26/2019	10/14/2019		
2.1.2 Collect data about UC inaccesible areas	49	8/26/2019	10/14/2019		
2.1.3 Collect data about UC buildings names and locations	21	10/7/2019	10/28/2019		
2.1.4 Collect data about UC building layouts/blueprints	21	10/14/2019	11/4/2019		
2.1.5 Collect data about alternative routes and elevators	21	10/21/2019	11/11/2019		
2.2 Software Requirements	21	9/30/2019	10/21/2019		
2.2.1 Determine back end development language	7	9/30/2019	10/7/2019		
2.2.2 Determine front end development language	7	9/30/2019	10/7/2019		
2.2.3 Determine hosting environment	7	10/7/2019	10/14/2019		
2.2.3 Determine data persistennce	7	10/14/2019	10/21/2019		
2.2.4 Determine mapping API	7	10/14/2019	10/21/2019		
2.3 Data Sources	14	11/4/2019	11/18/2019		
2.3.1 Determine where we store event data	7	11/4/2019	11/11/2019		
2.3.2 Determine how we transmit data between our application	7	11/11/2019	11/18/2019		
2.3.3 Determine where we store content	7	11/11/2019	11/18/2019		
2.4 Hosting Environments	14	10/7/2019	10/21/2019		
2.4.1 Research if benefits exist for Maps API users and GCP	7	10/7/2019	10/14/2019		
2.4.2 Choose hosting platform	7	10/14/2019	10/21/2019		
3.0 System Analysis and Design	35	9/30/2019	11/4/2019		
3.1 Software Diagram	14	9/30/2019	10/14/2019		
3.1.1 Create UML diagram for frontend	14	9/30/2019	10/14/2019		
3.1.2 Create UML diagram for backend	14	9/30/2019	10/14/2019		
3.1.3 Create diagram of Map layers	21	9/30/2019	10/21/2019		
3.2 Network Diagram	14	10/7/2019	10/21/2019		
3.2.1 Network client and web server	7	10/7/2019	10/14/2019		
3.2.2 Network web server data access	7	10/7/2019	10/14/2019		
3.2.2 Network for file storage	7	10/14/2019	10/21/2019		
3.2.3 Network data feed and determine format	7	10/14/2019	10/21/2019		
3.3 Database Diagram	7	10/21/2019	10/28/2019		
3.3.1 Create database schema	7	10/21/2019	10/28/2019		
3.3.2 Document access points	7	10/21/2019	10/28/2019		
3.4 Mockup UI	7	10/28/2019	11/4/2019		
3.4.1 Create Wireframe diagrams for mobile	7	10/28/2019	11/4/2019		
3.4.2 Create wireframe for web view	7	10/28/2019	11/4/2019		
3.4.3 Create simple prototype for mobile	7	10/28/2019	11/4/2019		
3.4.4 Create simple prototype for web	7	10/28/2019	11/4/2019		
4.0 Environment Setup	21	11/04/2019	11/25/2019		
4.1 Create client project in visual studio	7	11/04/2019	11/11/2019		
4.1.1 Impot libraries for development	7	11/04/2019	11/11/2019		
4.1.2 Determine project runs on browsers	7	11/04/2019	11/11/2019		
4.1.3 Determine project runs on mobile	7	11/04/2019	11/11/2019		
4.2 Setup project Github	7	11/4/2019	11/11/2019		
4.2.1 Create master branch	7	11/4/2019	11/11/2019		
4.2.2 Verify all team members have access	7	11/4/2019	11/11/2019		
4.2.3 Establish branching strategy	7	11/4/2019	11/11/2019		
4.3 Install Web application server	7	11/11/2019	11/18/2019		
4.3.1 Setup firewall rules	7	11/11/2019	11/18/2019		
4.3.2 Create data access objects	7	11/11/2019	11/18/2019		
4.3.3 Test connection to client	7	11/11/2019	11/18/2019		
4.4 Install Database Server	7	11/18/2019	11/25/2019		
4.4.1 Create tables based on database diagram	7	11/18/2019	11/25/2019		
4.4.2 Verify connection to web server	7	11/18/2019	11/25/2019		
4.4.3 Verify connection client	7	11/18/2019	11/25/2019		
4.4.4 Verify storage mechanism	7	11/18/2019	11/25/2019		
5.0 Development	126	11/24/2019	3/29/2020		
5.1 Standup base application running with map	3	11/24/2019	11/27/2019		
5.2 Develop data layers for routes	18	11/27/2019	12/15/2019		
5.2.1 Develop data layer for UC standard routes	18	11/27/2019	12/15/2019		
5.2.2 Develop data layer for Assistance Routes	18	11/27/2019	12/15/2019		
5.2.3 Map out data layers in event blocks	14	12/30/2019	1/13/2020		
5.3 Develop switch for data layers	27	1/6/2020	2/2/2020		
5.4 Develop routing rules	27	2/3/2020	3/1/2020		
5.4.1 Create alternative route rules	27	2/3/2020	3/1/2020		
5.2 Develop Search feature	27	3/2/2020	3/29/2020		
5.2.1 Verify connection to database	14	3/2/2020	3/16/2020		
5.2.2 Develop search functionality	13	3/16/2020	3/29/2020		
6.0 Testing	15	3/30/2020	4/14/2020		
Regression Testing	15	3/30/2020	4/14/2020		
User Acceptance Testing	15	3/30/2020	4/14/2020		

Figure 7: Gantt Chart

2.11 Problems Encountered and Analysis of Problems Solved

- **Underestimating the difficulties of designing the architecture of our application**

  We spent a lot of time discussing the features of our application and how it should look, but we did not give the same amount of time and consideration to how the various elements of the application, from the web servers to the database, would integrate to satisfy our requirements. We have since made great strides fleshing out the details of how our application will be set up, but it would be useful in future projects to design the overall architecture of the application as well as the UI and functional elements earlier in the planning phase.

- **Identifying features for our project that sets it apart as a unique product**

  As we researched and worked to complete our project, we noticed that there was more that needed to be done to set our product apart from similar ones. We came to realize that the feature to provide routes based on handicap accessibility exists. With this in mind, we brainstormed more features and ways to refine our current features to improve convenience and solve other issues that would encourage users to select our product over competitors.

- **Issues implementing custom routing using Xamarin.Forms**

  We ran into a number of issues implementing custom routing while attempting to utilize Xamarin.Forms as our developmental platform framework. These issues ultimately caused us to pivot into using Android Studio as our developmental platform, as navigation was the primary focus of our application. This led to our application not having cross-platform functionality, but instead being developed for Android devices alone.

- **University of Cincinnati campus shut down in light of COVID-19**

We ran into a number of issues related to communication and development as a result of COVID-19 becoming a global focus during the latter parts of the spring semester. Most importantly, we were unable to attempt to create any additional customized mapping related to UC's campus, as the campus was shut down and nobody felt comfortable walking around in public community areas for extended periods of time to manually map the campus. These issues ultimately led to a decrease in custom functionality for our application with regards to accessibility routing and building mapping at UC.

2.12 Future Recommendations

While planning, researching, and developing UC Route Assistant we ran into various hurdles that we had to overcome. If we had the opportunity to do it all over again from the start, we would have spent less time during the fall semester theorizing about the project and more time developing a proof of concept. This would have allowed us to recognize the developmental issues of working with Xamarin.Forms much sooner than we did.

If we had more time to work on the application, we would implement more customized routing features based on the University of Cincinnati's campus specifically. This would entail having the team members manually map buildings and stairwell locations on campus to allow us to input custom blocks and pathing into our routing.

After pivoting to Android Studio, one of the primary suggestions we had from others was to allow the application to work on both Android and iOS devices, as iPhones are used prominently in our society as well. A key disadvantage of using Android Studio is the loss of this cross-platform functionality, and if we had been able to complete development using Xamarin.Forms this would not have been a concern.

Finally, we have no future plans for this project at this time. We initially envisioned that our application could be tied into the core suite of UC products, allowing students to automatically sync their schedules and classes using their UC single sign on. This comes with a number of security concerns regarding data and password privacy, and we ultimately decided against pursuing any future plans for this project.

3. Conclusion

3.1 Fall Semester 2019

The primary focus of our work in the fall semester was in detailing our schedule and plans to create the UC Route Assistant application. We ultimately decided on creating a mobile web application as we believed it would best suit the needs of users that we expected to be using our application. We also decided on a number of design objectives including but not limited to; ingestion of student schedules, ability to add in extra-curricular activities, wheelchair friendly mode to avoid stairs, and scheduling of large on-campus events in advance. At this point in project development, we have gained significant knowledge regarding project timeline management, project scope projections, and the importance of good communication. Throughout the spring semester we will be developing, testing, and improving upon our design for the UC Route Assistant application.

3.2 Spring Semester 2020

The spring semester was filled with opportunities, challenges, unforeseen events, and ultimately allowed us to learn lessons that nobody on our team expected. One of the largest lessons we learned during this semester was due to our pivot to utilizing Android Studio as our platform framework instead of Xamarin.Forms. This was a difficult decision, but one that had to be made due to issues implementing core features of our application using Xamarin.Forms. The lesson learned here is that it's extremely important to create and test core functionality as early on in the development lifecycle as possible, because there may be critical issues that result in

incompatibility between our intended application and the framework being utilized. Another important lesson that we learned this semester was the significance of being flexible with regards to events outside of your control. During this semester the entire world was altered as a result of COVID-19, a virus that came out of nowhere and caused society as we know it to change. This massive shift in what actions were appropriate for individuals impacted our project development and forced us to make changes with our applications functionality and what we would be able to accomplish, as a result of not being able to walk around campus and map out certain buildings and staircases.

Throughout the spring semester we all developed and enhanced different skills based on our project responsibilities. A few skills that our group honed over the course of the application's development were the overall commitment, communication, flexibility, and professionalism required to see through a project of this magnitude from start to finish. We completed a number of different objectives since the fall semester, including core pieces of our final paper, core functionality for our application, our poster that was intended to be presented at tech expo, our final presentation, and a complete redesign of this final paper.

As a whole, this project has had a significant impact on our team's understanding of the project development lifecycle, project management, and the flexibility required in undertaking a long-term developmental effort with unforeseen hindrances. It has been an exceptional learning opportunity and we are all more knowledgeable in various ways as a result of completing this project and course over the last two semesters.

Appendix A. Additional Information

Figure 8: Final Poster below is our groups final poster created for presentation at Tech Expo.

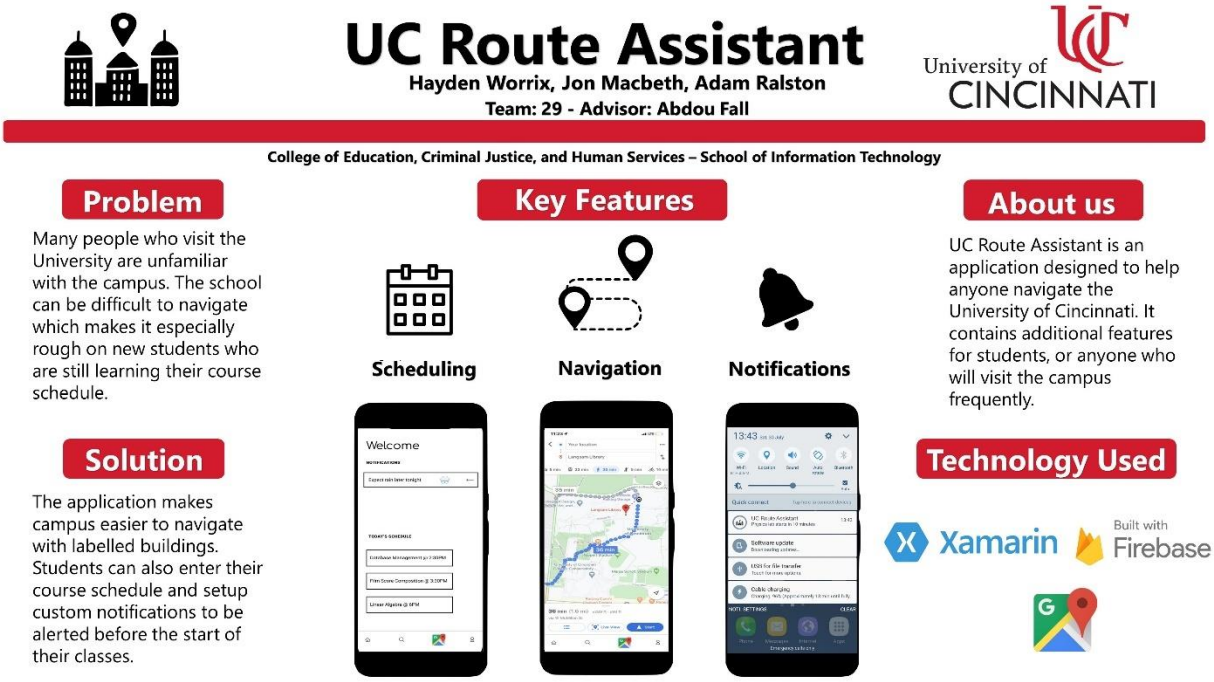


Figure 8: Final Poster

Tools Used:

Android Studio - https://developer.android.com/studio

Firebase - https://firebase.google.com/

Google Cloud Platform - https://cloud.google.com/

Google Maps - https://developers.google.com/maps/documentation

Java - https://docs.oracle.com/en/java/

Lucidchart - https://www.lucidchart.com/

Microsoft Office - https://www.office.com/

Appendix B. References

[1]Chou T.-L., ChanLin L.-J. (2012). Augmented Reality Smartphone Environment Orientation Application: A Case Study of the Fu-Jen University Mobile Campus Touring System. *Procedia - Social and Behavioral Sciences*, 46, pp. 410-416. doi:https://doi.org/10.1016/j.sbspro.2012.05.132

[2]Odekunle, M. O., Odo, E. K., Sule, I., & Adenle, A. A. (2018). Application of geographic information systems in creating smart campus map of federal university of technology, minna bosso campus. *I-Manager's Journal on Digital Signal Processing*, 6(3), pp.10-17. doi:http://dx.doi.org.proxy.libraries.uc.edu/10.26634/jdp.6.3.15680

[3]Suro, P. (2019, Aug 26). University of Cincinnati starts class with record-breaking enrollment for 7th straight year. *WCPO Cincinnati*. Retrieved from https://www.wcpo.com/news/education/incoming-university-of-cincinnati-freshmen-excited-for-first-day-of-classes

[4]Torres-Sospedra, J., Torres-Sospedra, J., Avariento, J., Rambla, D., Montoliu, R., Montoliu, R., . . . Huerta, J. (2015). Enhancing integrated indoor/outdoor mobility in a smart campus. *International Journal of Geographical Information Science*, 29(11), 1955-1968. doi:10.1080/13658816.2015.1049541

[5]Vogels, E. (2019, Sep. 09). Millennials stand out for their technology use, but older generations also embrace digital life. *Pew Research Center*. Retrieved from: https://www.pewresearch.org/fact-tank/2019/09/09/us-generations-technology-use/