Possibility

by

Carlos Lopez II, and Zachary Jepsen

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

Carlos Lopez II	4/13/2020
Carlos Lopez II – Project Manager	Date
Zachary Jepsen	4/13/2020
Zachary Jepsen – Software Developer	Date
Abdou Fall	4/13/2020
Abdou Fall – Faculty Advisor	Date

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

Prepared by
Carlos Lopez II, and Zachary Jepsen

Students of
University of Cincinnati
College of Education, Criminal Justice, and Human Services
School of Information Technology

April 2020

Table of Contents

Abstract 1
1.0 Introduction 2
1.1 Problem 2
1.2 Solution 3
1.3 Project Goals 3
1.4 Overview 3
2.0 Discussion 4
2.1 Project Concept 4
2.2 Design Objectives 4
2.3 Methodology/ Technical Approach 5
2.4 User Profile 5
2.5 Use Case Diagram 8
2.6 Project Schedule 9
2.7 Budget 14
3.0 Technical Elements 14
3.1 Network 14
3.2 Application 15
3.3 Database 15
3.4 Security 15
3.5 Application Architecture 16
3.6 User Interface Breakdown 16
4.0 Testing 18
4.1 Overview/Methodology 18
4.2 Scope of Testing 19
4.3 Objectives 19
4.4 Logging Test and Procedures 19
4.4.1 Stability Testing 20
4.4.2 Functionality Testing 21
4.4.3 User-Interface Testing 21
4.5 What was Learned During Testing? 22
5.0 Future Recommendations and Problems Encountered 23
5.1 Problems Encountered 23
5.2 Future Recommendations 23
6.0 Conclusion 25
6.1 Fall Semester 25
6.2 Spring Semester 25
7.0 References 27
8.0 Appendix 28
8.1 Appendix A: Poster 28

List of Illustrations

List of Figures

Figure 4: Project Timeline (Fall) 9
Table 1: Work Breakdown Structure (Fall) 10
Figure 6: Application Architecture 16
Figure 7: UI Login 16
Figure 8: UI Recipe 17
Figure 9: UI Restaurants 18
Table 4: Stability Testing 20
Table 5: Functionality Testing 21
Table 6: User-Interface Testing 21

List of Tables

Table 1: Work Breakdown Structure (Fall) 12
Table 2: WBS (Spring) 13
Table 3: Project Budget 14
Table 4: Stability Testing 20
Table 5: Functionality Testing 21
Table 6: User-Interface Testing 22

Abstract

According to a study done by TouchBistro, sixty-three percent of restaurant goers say that the type of food has the biggest impact on their restaurant choice. Many people that have dietary restrictions or specific eating lifestyles can have difficulty choosing a place to eat that meets their needs. Currently, there are applications that target one specific eating lifestyle, (e.g. vegetarian, vegan, gluten free, etc.) but none encompass the full spectrum of eating lifestyles a person can follow. The Possibility mobile application is a food finding application that allows users to enter their specific eating lifestyle and view menu options of nearby restaurants that accommodate their needs. The application features tools for reviewing restaurant choices and sharing recipes with other users. Whether a person is looking for vegan, vegetarian, keto, paleo, gluten-free, or allergy-free options, they will find the best place to eat using the Possibility App.

1.0 Introduction

In today's age eating a more plant-based diet is becoming a popular lifestyle decision. Being able to locate restaurants that abide by these specific eating habits can become a difficult task, especially when you are someone with specific dietary restrictions. There is a growing need in being able to determine satisfactory dining locations that meet peoples specific eating lifestyles. In order to alleviate these problems, an application will need to determine which restaurants will be most aligned with the individual's personal eating habits.

1.1 Problem

According to a study done by the Environmental Health Specialists Network (EHS - Net), most restaurants did not have dedicated areas and equipment for preparing and cooking allergen-free food. Whether a person is vegan, vegetarian, or has a food allergy, having dedicated areas and equipment for preparing food is important because customers rely on the restaurant staff to provide them with accurate information to maintain their eating habits. Simply choosing a restaurant can seem overwhelming when trying to ensure that an eatery meets your requirements.

Based on our research, there is not a solid mobile application available that can help vegans, vegetarians, or people with dietary allergies locate restaurants that abide by their eating habits. There are applications that exist like HappyCow, Green Kitchen, and Easy Vegetarian that target one specific eating lifestyle, but none that cover the entire spectrum of eating lifestyles. Eating out can be a group activity and being able to line up your individual needs with the group's interests could be beneficial for everyone. A tool to

help select a restaurant that can meet the needs of a vegan, vegetarian, or person with dietary allergies can help avoid the lengthy selection process of deciding where to eat.

1.2 Solution

Possibility is a mobile application that allows users to browse collections of restaurants and their menu items based on their eating habits (whether it be they do not eat meat, have a gluten free diet, or are vegan, etc..). What makes our application unique is that it is able to target all the different types of eating lifestyles. When it comes time to select a new place to eat at, the user can connect with others who eat the same way (being able to write their rate and view ratings from others), as well as share recipes that people can try from home. Our application finds restaurants for users of any diet and also allows users to find great recipes to try at home.

1.3 Project Goals

Our goal was to develop an Android mobile application that provides vegetarians, vegans, and others with food allergies the ability to find, view, rate, and review restaurants near them that meet their dietary needs. Possibility provides quick access to find places near them that they can eat at no matter what dietary restrictions they may have. Possibility promotes easy access to find places that not only meet specific diets or food restrictions but also provide users with information on how the food is prepared.

1.4 Overview

The remainder of the final report outlines in detail how the project was completed. The report includes the following sections: project concept, design objectives, methodology and technical approach, user profile, use case diagram, project schedule, technical elements, testing, and conclusion.

2.0 Discussion

2.1 Project Concept

Possibility is a responsive mobile application capable of retrieving restaurant data from different Application Programming Interface (API) sources and also retrieving location services of the user and restaurants. Users will be able to use this data to determine the best local restaurants that can provide them meals that meet their dietary needs. The idea for this project came from Carlos who went vegan for six months. He had difficulty locating restaurants that would abide by his eating lifestyle and wanted an application that could make the process of finding a restaurant easier.

2.2 Design Objectives

The features possibility includes:

- Users have the ability to find and view restaurants that fit their specific needs
- Users can search for restaurants using keywords such as type of food, restaurant name, location, etc.
- Users can share recipes containing foods that meet their specific dietary restrictions
- Users will be able to rate different restaurants and recipes

A few of our initial goals that had to be abandoned for the scope of this project were:

- Give users the ability to find travel destinations (hotels) that meet their diet to allow even easier travel arrangements when looking to go on vacation
- Allow users to track meals eaten from restaurants
- Message other users
- Develop for iOS and desktop applications

2.3 Methodology/ Technical Approach

The primary objective of Possibility is to deliver a solid mobile application that allows users, no matter their dietary restrictions, to find a place to eat. To accomplish this goal, we needed something that is cross-platform and easily configurable across multiple platforms. We also needed something on the backend that is scalable as the userbase grows. The Xamarin framework paired with Microsoft Azure fits these criteria near perfectly. The codebase for all platforms (Android, iOS, and Windows) is shared through Xamarin using C# and the .NET framework. We are utilizing the Google Maps API for users to find their restaurants and get directions to where they would like to eat. On the Azure side of things, we used Azure CosmosDB to store our data and Azure Blob Storage to store images. This allowed our databases to be scalable as time goes on. For user authentication we used Auth0 as it keeps user credentials secure and allowed us easy access to social login implementation.

2.4 User Profile

Our user profiles for the Possibility mobile application provided the team with details of the audience for which the application is designed. And served as a reference throughout development.

In ***Figure 1: User Profile – User*** below, the primary user is defined, these are people who follow a specific diet or have dietary restrictions. The interface that the user primarily deals with is the restaurant finder screen that contains search bars and GPS locations specific to what kind of restaurants meet the criteria of food they are looking for.

User Profile Form

PROJECT:

Possibility – Restaurant finder application for people with dietary restrictions or specific diets

POTENTIAL USERS:

- Users with dietary restrictions
- Users who follow a specific diet
- Users trying to find places or recipes for friends with dietary restrictions or specific diets

SOFTWARE, INTERFACE, AND RELATED EXPERIENCE:

Most users will have had experience with navigating mobile applications. Possibility will not require users to have used any similar software in the past. This app will provide easy to understand functions to find food that fits a specific diet.

EXPERIENCE WITH SIMILAR APPLICATIONS:

- HappyCow
- Yelp
- OpenTable
- Zomato
- Foursquare
- LocalEats
- Zagat

TASK EXPERIENCE:

Users will have prior experience:

- Using a mobile application from an Android Device
- Creating and analyzing restaurant reviews
- Locating restaurants based on user input

FREQUENCY OF USE:

This application will be used by the users on a daily basis. Ideally users are able to find places to eat for multiple meals a day.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Visually fluid and easy to navigate UI
- Short response times
- Responsive to autofill and dropdown selection
- Ability to share reviews

Figure 1: User Profile – User

In ***Figure 2: User Profile – Admin*** below, the admin user is defined, these are the developers. This role will deal with managing the user accounts and setting up/modifying the restaurant/food data.

User Profile Form
PROJECT: Possibility – Restaurant finder application for people with dietary restrictions or specific diets
POTENTIAL USERS: - Developers - Administrators
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: Admin users will have had experience with navigating and developing mobile applications. Admin users will have had experience with Android mobile application development.
EXPERIENCE WITH SIMILAR APPLICATIONS: - HappyCow - Yelp - OpenTable - Zomato - Foursquare - LocalEats - Zagat
TASK EXPERIENCE: Admins will have prior experience: - Using a mobile application from an Android Device - Managing user accounts - Data storage
FREQUENCY OF USE: This application will be used by the admins on a daily or weekly basis. Admins will need to manage user accounts and make sure that menu data and restaurant data stay up to date.

KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:

- Visually fluid and easy to navigate UI
- Engaging layout
- Map/Location services

Figure 2: User Profile – Admin

2.5 Use Case Diagram

The following diagram, ***Figure 3: Use Case Diagram***, displays the use case for Possibility. The Diagram depicts all users of Possibility along with the corresponding tasks each user has access to when interacting with the application.

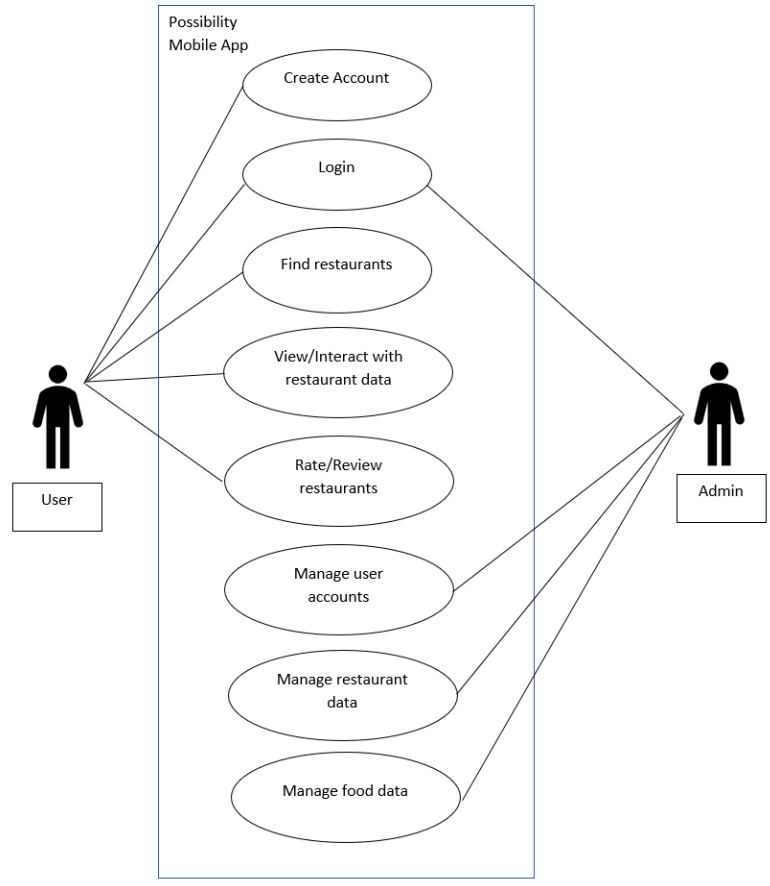

Figure 3: Use Case Diagram

2.6 Project Schedule

Figure 4: Project Timeline (Fall) and ***Figure 5: Project Timeline (Spring)*** below displays our project timeline, and process flow. The timeline is broken up into deliverables, research, development, and testing with various sub sections in each section. The project started on 8/26/2019, and testing and development was completed 3/23/2020.

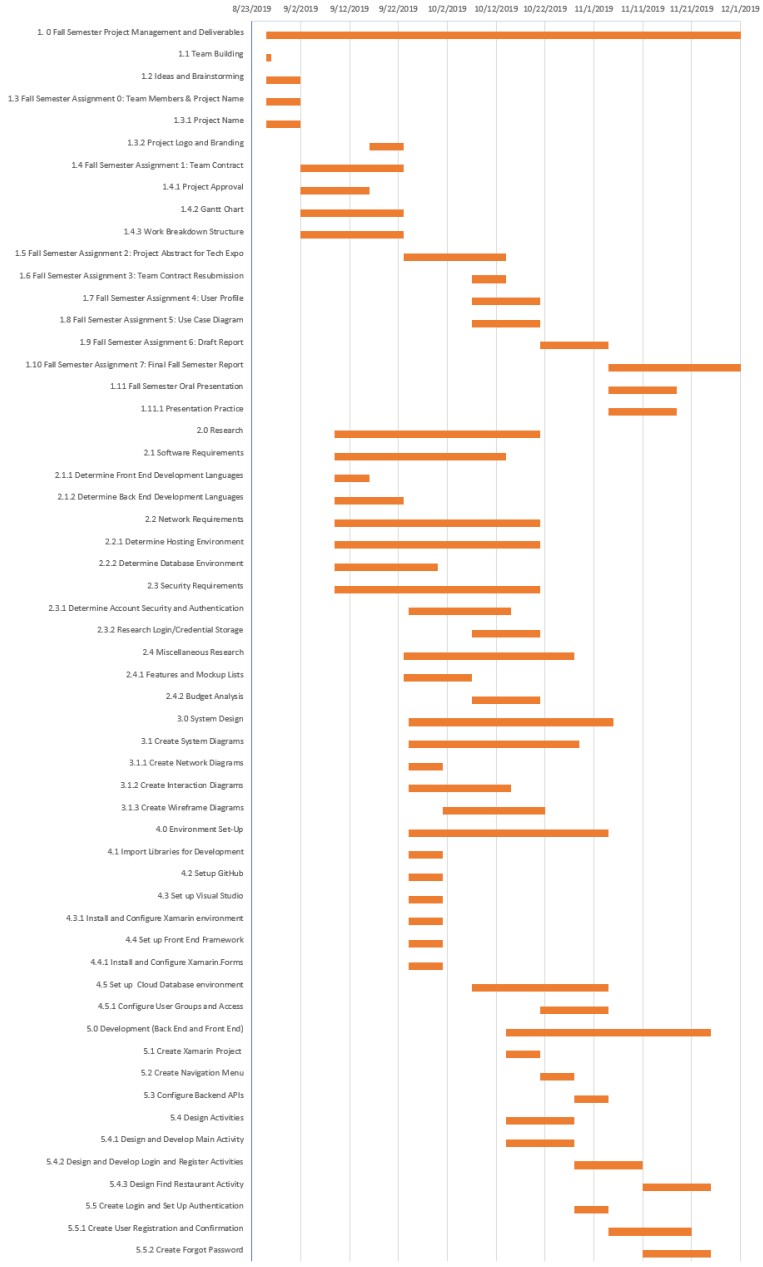

Figure 4: Project Timeline (Fall)

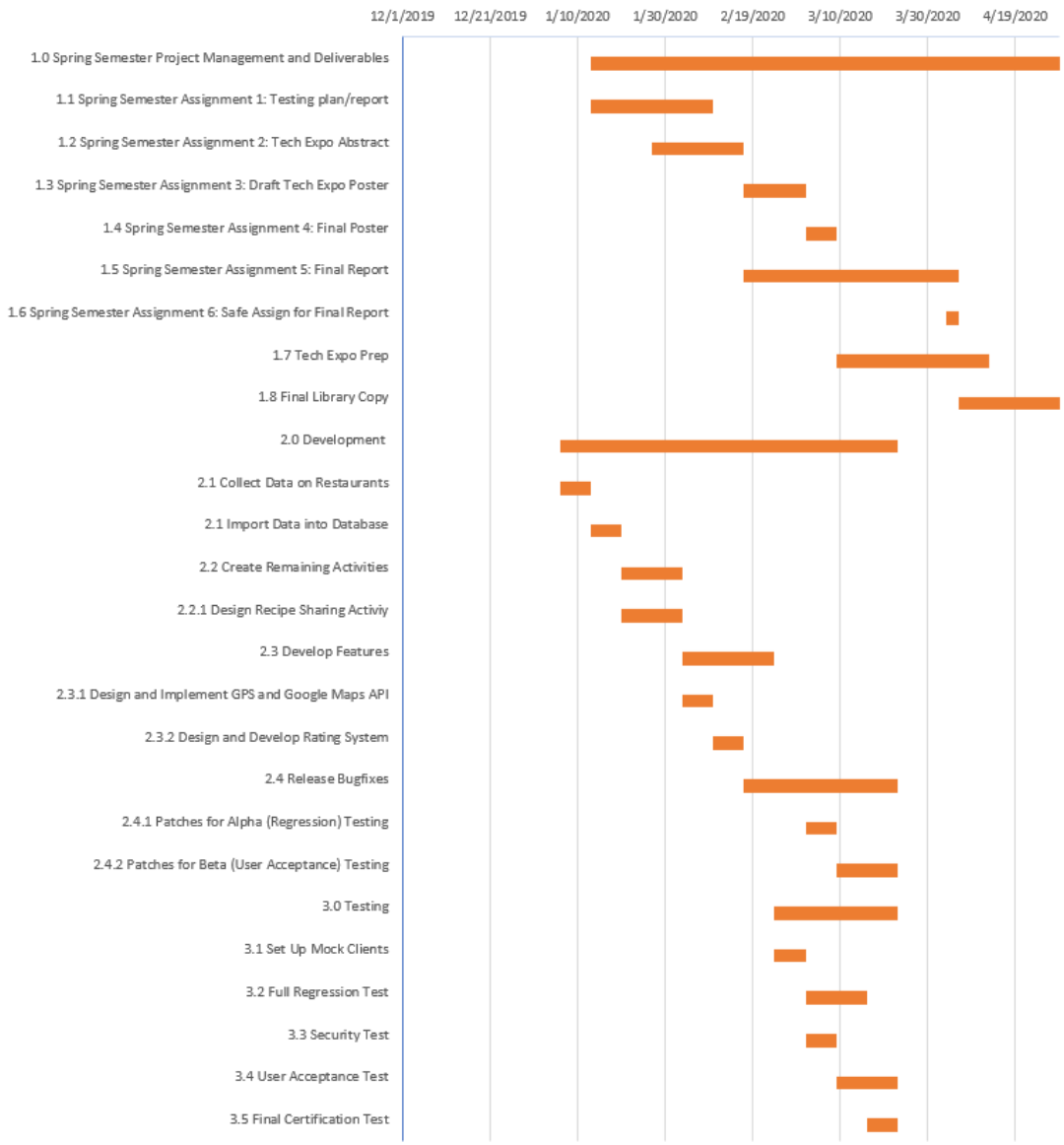

Figure 5: Project Timeline (Spring)

Table 1: Work Breakdown Structure (Fall) and ***Table 2: WBS (Spring)*** below displays our work breakdown structure (WBS). The WBS is broken up into deliverables, research, environment set-up, development, and testing with various sub sections in each section. The project was started on 8/26/2019, and testing and development was completed on 3/23/2020.

Task Name	Duration (Days)	Start Date	End Date
1. 0 Fall Semester Project Management and Deliverables	98	8/26/2019	12/2/2019
1.1 Team Building	1	8/26/2019	8/27/2019
1.2 Ideas and Brainstorming	7	8/26/2019	9/2/2019
1.3 Fall Semester Assignment 0: Team Members & Project Name	7	8/26/2019	9/2/2019
1.3.1 Project Name	7	8/26/2019	9/2/2019
1.3.2 Project Logo and Branding	7	9/16/2019	9/23/2019
1.4 Fall Semester Assignment 1: Team Contract	21	9/2/2019	9/23/2019
1.4.1 Project Approval	14	9/2/2019	9/23/2019
1.4.2 Gantt Chart	21	9/2/2019	9/23/2019
1.4.3 Work Breakdown Structure	21	9/2/2019	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	7	10/7/2019	10/14/2019
1.7 Fall Semester Assignment 4: User Profile	14	10/7/2019	10/21/2019
1.8 Fall Semester Assignment 5: Use Case Diagram	14	10/7/2019	10/21/2019
1.9 Fall Semester Assignment 6: Draft Report	14	10/21/2019	11/4/2019
1.10 Fall Semester Assignment 7: Final Fall Semester Report	28	11/4/2019	12/2/2019
1.11 Fall Semester Oral Presentation	14	11/4/2019	11/18/2019
1.11.1 Presentation Practice	14	11/4/2019	11/18/2019
2.0 Research	42	9/9/2019	10/21/2019
2.1 Software Requirements	35	9/9/2019	10/1/2019
2.1.1 Determine Front End Development Languages	7	9/9/2019	10/1/2019
2.1.2 Determine Back End Development Languages	14	9/9/2019	10/1/2019
2.2 Network Requirements	42	9/9/2019	10/21/2019
2.2.1 Determine Hosting Environment	42	9/9/2019	10/21/2019
2.2.2 Determine Database Environment	21	9/9/2019	10/1/2019
2.3 Security Requirements	42	9/9/2019	10/21/2019

2.3.1 Determine Account Security and Authentication	21	9/24/2019	10/14/2019
2.3.2 Research Login/Credential Storage	14	10/7/2019	10/21/2019
2.4 Miscellaneous Research	35	9/23/2019	10/21/2019
2.4.1 Features and Mockup Lists	14	9/23/2019	10/1/2019
2.4.2 Budget Analysis	14	10/7/2019	10/21/2019
3.0 System Design	42	9/24/2019	11/4/2019
3.1 Create System Diagrams	35	9/24/2019	10/21/2019
3.1.1 Create Network Diagrams	7	9/24/2019	10/1/2019
3.1.2 Create Interaction Diagrams	21	9/24/2019	10/7/2019
3.1.3 Create Wireframe Diagrams	21	10/1/2019	10/21/2019
4.0 Environment Set-Up	41	9/24/2019	11/4/2019
4.1 Import Libraries for Development	7	9/24/2019	10/1/2019
4.2 Setup GitHub	7	9/24/2019	10/1/2019
4.3 Set up Visual Studio	7	9/24/2019	10/1/2019
4.3.1 Install and Configure Xamarin environment	7	9/24/2019	10/1/2019
4.4 Set up Front End Framework	7	9/24/2019	10/1/2019
4.4.1 Install and Configure Xamarin.Forms	7	9/24/2019	10/1/2019
4.5 Set up Cloud Database environment	28	10/7/2019	11/4/2019
4.5.1 Configure User Groups and Access	14	10/21/2019	11/4/2019
5.0 Development (Back End and Front End)	42	10/14/2019	2/4/2020
5.1 Create Xamarin Project	7	10/14/2019	10/21/2019
5.2 Create Navigation Menu	7	10/21/2019	10/28/2019
5.3 Configure Backend APIs	7	10/28/2019	11/11/2019
5.4 Design Activities	14	10/14/2019	11/21/2019
5.4.1 Design and Develop Main Activity	14	10/14/2019	10/28/2019
5.4.2 Design and Develop Login and Register Activities	14	10/28/2019	11/11/2019
5.4.3 Design Find Restaurant Activity	14	11/11/2019	11/21/2019
5.5 Create Login and Set Up Authentication	7	10/28/2019	11/4/2019
5.5.1 Create User Registration and Confirmation	17	11/4/2019	11/21/2019
5.5.2 Create Forgot Password	14	11/11/2019	11/25/2019

Table 1: Work Breakdown Structure (Fall)

1.0 Spring Semester Project Management and Deliverables	107	1/13/2020	4/29/2020
1.1 Spring Semester Assignment 1: Testing plan/report	28	1/13/2020	2/10/2020
1.2 Spring Semester Assignment 2: Tech Expo Abstract	21	1/27/2020	2/17/2020
1.3 Spring Semester Assignment 3: Draft Tech Expo Poster	14	2/17/2020	3/2/2020
1.4 Spring Semester Assignment 4: Final Poster	7	3/2/2020	3/9/2020
1.5 Spring Semester Assignment 5: Final Report	49	2/17/2020	4/6/2020
1.6 Spring Semester Assignment 6: Safe Assign for Final Report	3	4/3/2020	4/6/2020
1.7 Tech Expo Prep	35	3/9/2020	4/13/2020
1.8 Final Library Copy	23	4/6/2020	4/29/2020
2.0 Development	77	1/6/2020	3/23/2020
2.1 Collect Data on Restaurants	7	1/6/2020	1/13/2020
2.1 Import Data into Database	7	1/13/2020	1/20/2020
2.2 Create Remaining Activities	14	1/20/2020	2/3/2020
2.2.1 Design Recipe Sharing Activiy	14	1/20/2020	2/3/2020
2.3 Develop Features	21	2/3/2020	2/24/2020
2.3.1 Design and Implement GPS and Google Maps API	7	2/3/2020	2/10/2020
2.3.2 Design and Develop Rating System	7	2/10/2020	2/17/2020
2.4 Release Bugfixes	35	2/17/2020	3/23/2020
2.4.1 Patches for Alpha (Regression) Testing	7	3/2/2020	3/9/2020
2.4.2 Patches for Beta (User Acceptance) Testing	14	3/9/2020	3/23/2020
3.0 Testing	28	2/24/2020	3/23/2020
3.1 Set Up Mock Clients	7	2/24/2020	3/2/2020
3.2 Full Regression Test	14	3/2/2020	3/16/2020
3.3 Security Test	7	3/2/2020	3/9/2020
3.4 User Acceptance Test	14	3/9/2020	3/23/2020
3.5 Final Certification Test	7	3/16/2020	3/23/2020

Table 2: WBS (Spring)

2.7 Budget

The cost for us from this project was very minimal. The only actual expense we had was for two test devices so we could each test the application in real-time on a physical device rather than an emulator. Factoring in labor cost at an hourly rate of $20 an hour which is comparable real-world wages for this type of work you can see our calculated real-world total below.

The following table, ***Table 3: Project Budget***, displays the overall budget for the project along with estimates for real world costs when accounting for labor.

No.	Item	Unit Multiplier (Hours/Amount)	Unit Price	Line Item Total
Labor				
1	Labor	1600	$20	$32,000
Hardware				
2	Samsung Galaxy A20	0	$95	$95
3	Samsung Galaxy A20	0	$95	$95
Service				
4	Azure CosmosDB	0	$.20-$3 monthly	First Year $0
5	Azure Blob Storage	0	$.20-$3 monthly	First Year $0
Real World Total:				$32,190
Actual Total:				$190

Table 3: Project Budget

3.0 Technical Elements

3.1 Network

Our application connects to Auth0 user authentication API through an HTTP interface where the user is authenticated and from there will receive a user ID token to exchange for an Azure CosmosDB resource token in the cloud. CosmosDB stores the bulk of our information but gets stored images from Azure Blob Storage.

3.2 Application

Our application is built using Xamarin.Forms, which uses C# and the .Net framework. Xamarin is great for building cross-platform mobile applications and connecting with Azure our cloud service of choice. Using Xamarin we have created an Android application communicates with the Azure database to create, read, and update restaurant and recipe data. Xamarin.Forms allowed us to develop a sleek native application and have the ability, down the road, to port to iOS with much less work than re-writing the entire thing.

3.3 Database

We used Azure CosmosDB which is hosted in the Azure cloud. CosmosDB stores records as schema less JSON documents allowing various structures of data in one collection, which was great for us because different restaurants and recipes may different information available that another might not. Being schema less also means it is very quick when retrieving data. CosmosDB does not easily support storing images, so we used Azure Blob Storage for this. CosmosDB pulls from Blob Storage when needed to retrieve restaurant or recipe images.

3.4 Security

We used Auth0 for user authentication. Auth0 takes care of the security and privacy of our user accounts as well as allows social logins and two-factor authentication for added security. Auth0 has unique ID’s for each user allowing control over what data that user sees from CosmosDB. Using Auth0 saved us time and ensured us that our user’s data was going to be kept safe.

3.5 Application Architecture

Figure 6: Application Architecture below depicts the technology used in our application. Xamarin was the technology used to create the Android application. Xamarin connects to Auth0 for user authentication and security which then exchanges a user token with CosmosDB to access secured resources.

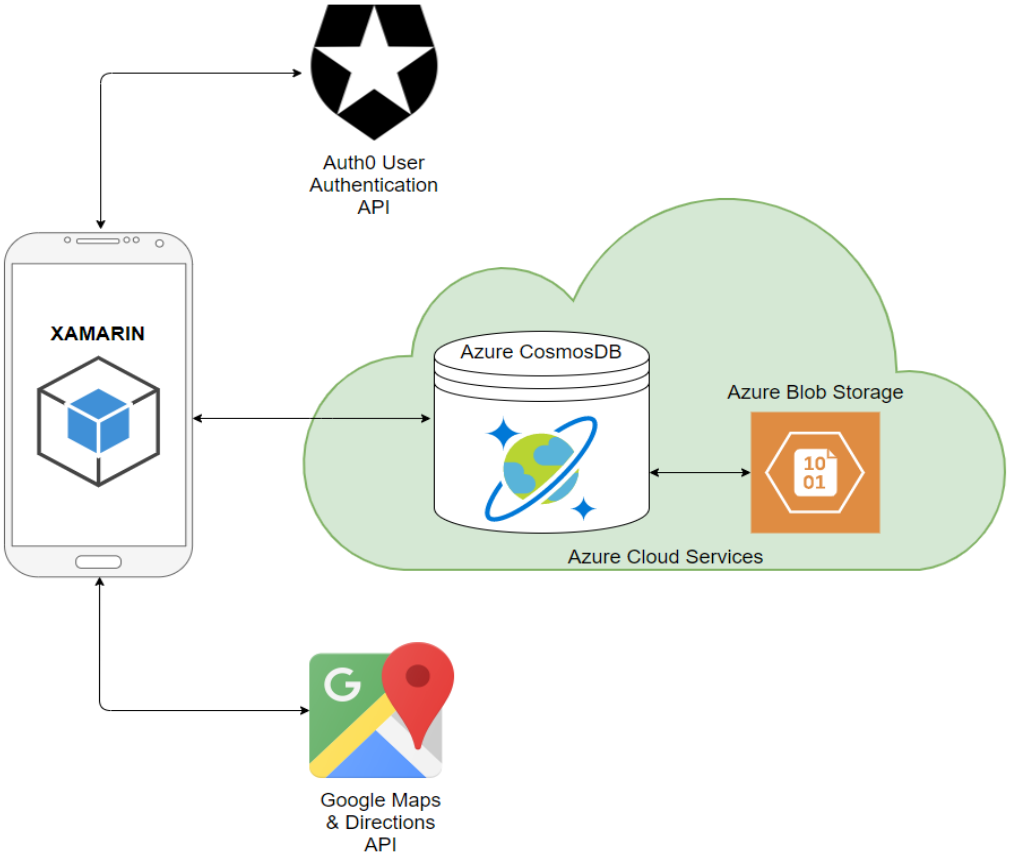

Figure 6: Application Architecture

3.6 User Interface Breakdown

Figure 7: UI Login, below depicts the login screen for the viewer to create an account or login.

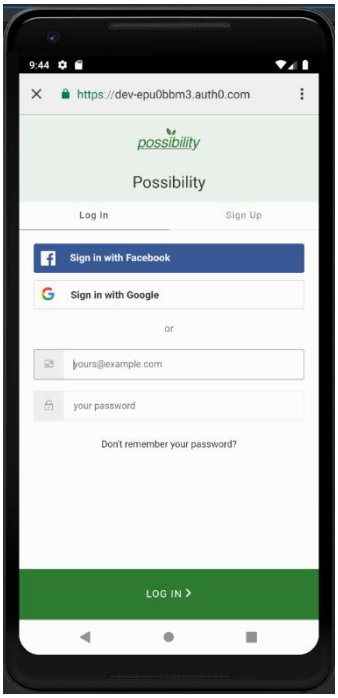

Figure 7: UI Login

Figure 8: UI Recipe, below depicts the recipe screen to view and create recipes.

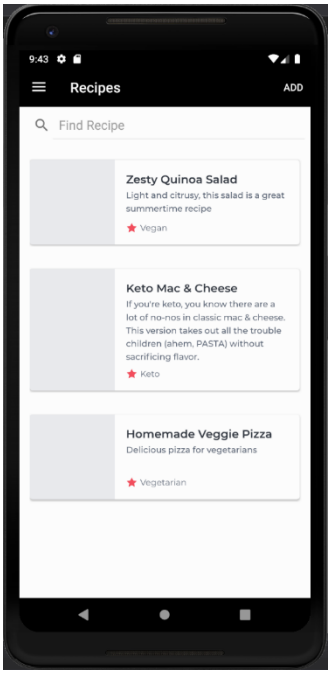

Figure 8: UI Recipe

Figure 9: UI Restaurants, below depicts the restaurant finder screen to view and locate restaurants.

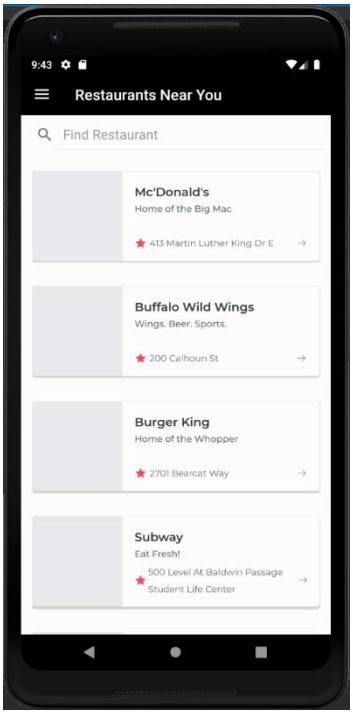

Figure 9: UI Restaurants

4.0 Testing

4.1 Overview/Methodology

Our testing methodology uses two approaches:

1. QA testing with test cases by the developers
2. User Acceptance Testing (UAT)

The first approach occurred throughout development of the application. When a feature was finished test case(s) were created to test functionality and stability of the feature. Both team members created the test cases as a team and then separately tested the cases. Tests and test cases are listed below in the section 4.4.

The second approach is UAT. Once the application passed all test cases the application was given to users for testing. They re-test the test cases below as well as just use the application

regularly for more testing. This allowed us to see what the users are having problems with and if there were test cases that did not properly account for everything.

4.2 Scope of Testing

Our test cases cover all major features of the application. Our testing strategy included testing the main functionality of the application, the application stability and compatibility, and the user interface. This covers end-to-end usage of the application, from creating an account to being able to search for a restaurant based on personal dietary restrictions. Our testing strategy also covers how the users feel and respond to using the applications interface.

4.3 Objectives

The goal of testing Possibility was to identify issues and bugs within the application that could affect users and admins in any given way. If a UI element is not working properly or if some database connection fails, these things were found through testing. During the testing phase, specific test cases that require multiple different sections of the application were tested.

4.4 Logging Test and Procedures

Three main categories were accounted for when developing test cases:

Stability - contains testing for performance issues, lagging, slowdowns, crashes

Functionality – contains testing to ensure features are doing what they are supposed to do (i.e., inputting log in information and clicking login will log the user in.)

User-Interface – contains testing for proper displaying of UI elements (buttons, text, error messages, etc.)

After testing a pass/fail condition was assessed depending on the type of test. For stability tests if no crashes occurred and minimal slowdowns, less than 2 per hour and under 5 seconds, this is considered a pass. For functionality, completing the action described in the test case without any

errors is a pass, otherwise it is a fail. For user-interface tests, if all UI elements appear as they should and are displayed properly it is a pass.

4.4.1 Stability Testing

Table 4: Stability Testing below, was designed to determine overall application stability and compatibility with one high-end and low-end android device. It addresses slowdowns, crashes, and any other lagging of the application. It is the only testing scenario where a variety of devices are used. The main questions answered through testing are:

1. Is there any lagging after running the application for multiple hours?
2. If there is lagging what was the user doing when it occurred?
3. Throughout use of application are there any crashes?
4. If so, what action was being executed?

Stability Test	Hours Tested	Crashes	Slowdown	Pass / Fail
Emulation Test	1	0	0	Pass
Emulation Test	2	0	0	Pass
Emulation Test	3	0	0	Pass
Samsung Galaxy S10+	1	0	0	Pass
Samsung Galaxy S10+	2	0	0	Pass
Samsung Galaxy S10+	3	0	0	Pass
Samsung Galaxy A20	1	0	0	Pass
Samsung Galaxy A20	2	0	0	Pass
Samsung Galaxy A20	3	0	0	Pass

Table 4: Stability Testing

4.4.2 Functionality Testing

Table 5: Functionality Testing below, was designed to determine if features within the application operate and give the desired results. For example, if a user searches for a restaurant, the correct restaurant is searched for and returned to the user. The main questions answered through testing are:

1. Can a user create an account?
2. Can a user login?
3. Can a user find restaurants near them that fit certain search criteria?
4. Can a user submit a recipe?
5. Can a user edit a recipe?

Functionality Testing		
Test Case #	Test Case Scenario	Pass/Fail
1	User Account Creation	Pass
2	User Login	Pass
3	Find Restaurant	Pass
4	Add Recipe	Pass
5	Edit Recipe	Pass

Table 5: Functionality Testing

4.4.3 User-Interface Testing

Table 6: User-Interface Testing below, was designed to ensure elements on screen display and act as expected. For example, text sizing, elements going off the screen, etc. The main questions addressed through testing are:

1. Are UI elements displaying properly?

2. Is inputted text being cut-off?
3. Are pages disappearing when navigating?
4. During page navigation are all UI elements appearing and disappearing as appropriate?

UI Testing

Test Case #	Test Case Description	Pass/Fail
7	Navigate from Login to Find Restaurant	Pass
8	Navigate from User Information to Recipe Page	Pass
9	Navigate from Recipe Page to User Information Page	Pass
10	Navigate from App Settings to Login then to Find Restaurant	Pass
11	Proper image sizing	Pass
12	Text getting cut-off	Pass
13	Edited Recipes change accordingly	Pass
14	New Recipes appear in feed	Pass

Table 6: User-Interface Testing

4.5 What was Learned During Testing?

During the testing phase, we were able to identify specific bugs in our application and gain a sense of how the users felt when interacting with the UI. Many small navigation bugs were found once put in the hands of users and we likely would have never noticed them without the user input. This allowed us to go and fix the bugs quickly and have users re-test them to ensure issues were fixed.

5.0 Future Recommendations and Problems Encountered

5.1 Problems Encountered

Being a team of two developers there were many problems we encountered that we had not truly accounted for. The first thing being the scope. Coming into the project we had big plans to get a ton of features into the application. The problem came when we wanted to use new technologies that neither of us had ever used, Xamarin and Azure. We both underestimated the time it took us to learn these technologies and understand them. So that was the first problem we ran into. We obviously overcame that hurdle but took up more time than originally planned.

Xamarin uses C# which is a familiar language for both of us but the UI for Xamarin.Forms uses XAML. There are similarities XAML shares with other technologies we had already known but designing sleek, clean UI was not an easy task. We took many days just reading and testing different tools XAML has to find what we were looking for.

Azure was a much larger problem than the Xamarin issues we ran into. We didn't have any prior experience with cloud technologies so firstly learning about Azure in general and how it works was a long process. And then deciding on what Azure database(s) we wanted to use was a lot more additional research. Where we ran into the most problems was connecting the Xamarin code to the Azure database. This took multiple weeks with various Xamarin test applications being made and many videos and articles having been read.

5.2 Future Recommendations

If we had to complete our senior design project over again, we would use software and tools that we each had more familiarity with. As a group, we believe that this would have enabled us to generate more features for the application and potentially deliver a greater overall project. If we were given more time to work on the project, we would try to deliver on our stretch goals that we

had to abandon. These goals were to port the application to IOS and desktop and add the ability to search for hotels across the world that would comply with a person's diet. Other's had suggested that we implement a feature into the application that allows you to order your food from the stores directly to your home. We believe this would be a great implementation into our application because of the popularity of other applications like Uber Eats and DoorDash. Another suggestion that we have had for our application was to rework the UI for our restaurants page to overall look better. Currently, we do not have plans of continuing to work on our application. We are proud of the work that we put in for both semesters and we are both open to continue developing the application later.

6.0 Conclusion

6.1 Fall Semester

The process of creating an application from scratch has been challenging, but also a valuable learning experience. Our initial idea at the beginning of the semester was to create a mobile application that help people with different eating lifestyles find a suitable restaurant to eat at. We began researching if ideas like this one existed already and we could only find applications that targeted select diets. The primary differences between our application and other applications that exist in the market would be that (1) our application would be able to serve both people with dietary conditions and specific eating lifestyles, and (2) our application can connect users with one another to leave reviews and recommendations for recipes/restaurants.

After deciding on our project, we ran into the issue of deciding what technologies to use to bring our project to life. Upon completing research, we decided that our Front-end will be developed using Xamarin (.Net and C#) and the data will be managed and stored using Microsoft Azure Storage. Xamarin will provide a native experience on both platforms that will be easy to navigate and use. For storage of information we will be using Azure CosmosDB along with Azure Blob Storage. We have created the Azure Storage account we will be using and have these features with barebones databases ready to be implemented in our app.

6.2 Spring Semester

During the spring semester, we mainly focused on improving the main functionalities of our application and fixing bugs that we had found through testing. We chose to abandon the idea of messaging between users because there was not enough time for us to successfully implement this feature. This project has taught us the importance of testing in a software environment because it helped us identify where our application needed extra attention. With being a group of

two people, we feel we have enhanced our communication and teamwork skills throughout the course of this project. The work needed to be split between the both of us and if either of us needed help the other would step in and contribute where necessary.

7.0 References

17, M. (2018, May 18). Survey: More than one-third of Americans follow a special diet. Retrieved November 10, 2019, from https://www.newhope.com/market-data-and-analysis/survey-more-one-third-americans-follow-special-diet.

How Diner's Choose Restaurants [Restaurant Insights Report]. (n.d.). Retrieved from https://www.touchbistro.com/blog/how-diners-choose-restaurants.

How Restaurants Address Food Allergies. (2019, June 12). Retrieved November 10, 2019, from https://www.cdc.gov/nceh/ehs/ehsnet/plain_language/allergy-practices.htm.

Insight, F. (2019, February 27). One-Third of Americans Are Dieting, Including One in 10 Who Fast ..... Retrieved November 10, 2019, from https://foodinsight.org/one-third-of- americans-are-dieting-including-one-in-10-who-fast-while-consumers-also-hunger-for-organic-natural-and-sustainable/.

What is Xamarin? - Xamarin. (n.d.). Retrieved April 12, 2020, from https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin

8.0 Appendix

8.1 Appendix A: Poster