BuckIt!

A Project Report Presented to
The Faculty of the Computer Engineering Department

San Jose State University
In Partial Fulfillment
Of the Requirements for the Degree
Bachelor of Science in Software Engineering

By
Jie Peng Hu
Michael Hyun
Samnang Sok
Joshua Ventocilla
04/2018

APPROVED FOR THE COLLEGE OF ENGINEERING

ameng

Digitally signed by ameng
DN: cn=ameng, o, ou,
email=ahymeng@yahoo.com, c=US
Date: 2018.05.08 13:02:23 -08'00'

Professor Anna Meng, Project Advisor

Professor Rod Fatoohi, Instructor

Dr. Xiao Su, Computer Engineering Department Chair

ABSTRACT

Location based Event and Activity Suggester Application

By Jie Peng Hu, Michael Hyun, Samnang Sok, Joshua Ventocilla

There are a plethora of activities to do around us. From having a picnic in front of the Eiffel Tower to going skydiving over the Hawaiian Islands, there are countless activities and places to experience throughout the world. Sometimes the number of events to accomplish can be overwhelming, and one might not know what to do first, especially if they are traveling in a foreign country. While it's a good idea to plan which activities to choose for your itinerary, there will be times when you will end up wondering what to do next merely because one is not inspired enough. Instead of taking action, we are stuck looking through other people's Instagram or Facebook page wishing that we are doing the fun activities that they are doing.

Even with so many activities available to do, it is common to find one's self-sitting at home not doing anything on a beautiful Saturday morning because they cannot decide what they want to do. This may result from a lack of planning or lack of inspiration, but this ultimately leads to people missing out on experiencing what there is to do around them. Another problem that arises with traveling, in general, is that people do not stay in a single location for very long. This means that travelers only have a short amount of time to wander and explore places, and is most likely given just one opportunity to experience the area. Also, another common problem is that wrong planning leads to missing out on the most fun activity, going to the best events, or visiting the best places in the area. In turn, this leads to an unpleasant experience and biased commentaries.

BuckIt! is an application that allows users to create their own personal bucket list. BuckIt! also serves as a location-based activity suggestion platform which enables locals to post suggestions for activities, and example itineraries for other people to see. A voting system allows the suggestions to be upvoted and commented by others. This upvoting feature allows other users to evaluate which events are the most popular in the area. Suggestions also allow for comments by other users to give additional information so that people can be well informed and prepared. BuckIt's applications interface gives users access to activities that are suggested by locals which means that they will gain local knowledge of which activities are fun, or which item on the restaurant menu is the most popular. People who use BuckIt! will have the most authentic experience possible at any given time. BuckIt!'s trending page recommends users which activities are most popular based on their location and preferences so that they are given certain ideas of what to do. Users who comment on suggestions may help other users by providing additional information for them to consider upon reading those comments. For example, one comment could be "make sure to bring an umbrella as the weather can change suddenly." It is these features of BuckIt! that makes it a unique application; one that enhances the user's travel experience.

Acknowledgements

First and foremost we would like to thank Prof. Anna Meng for being our mentor and accepting our offer to be this project’s advisor. She has mentored us and guided us throughout the project from day one. We would also like to thank our friends and family for the support they have given us. Lastly, we would like to thank San Jose State University for this challenging, but extremely wonderful journey.

Table of Contents

List of Tables and Figures

Chapter 1 Introduction

1.1 Project Goals and Objectives

The goal of this project is to provide users with the inspiration to go out and experience the world based on recommendations from people who have already gone out and experienced it for themselves. Also, this application provides a platform for users to submit activities or events that are interesting and fun to do based on the user's location. Based on those submissions, users can upvote the submitted activities, and also view which activities are popular in a given area. Users are able to view all the local activities and events that are currently taking place so users will be inspired by the suggestions of other people to go out and explore. Users can browse for activities that take place around the world and be inspired to experience such events at some point in their lives. For such reason, BuckIt! also provides a bucket list feature for users to collect a list of activities they wish to do in the future.

1.2 Problem and Motivation

Everyone living in this world wishes to have an adventurous and fulfilling life. Being around others while traveling or doing fun activities creates lifelong memories. People strive to be happy, and we use experiences and memories to measure the fulfillment of our lives. There are plenty of ways that a person can plan out which activities to do in life. Ultimately, the most common way of listing adventures you wish to experience is by creating a bucket list. The concept of a bucket list originated circa 2006 as a way to keep a tab of the goals that you wish to pursue before you die. The list usually contains life fulfilling activities such as seeing the Northern Lights, skydiving, rock climbing in Arizona, etc. Many people create bucket lists, but the problem is that people lack the motivation to fulfill those goals.

We have concluded that people often imagine themselves doing the things they truly want to do. The problem isn't the imagination of doing these activities, but that people do not act upon those wishes because of the lack of motivation. Our ultimate goal is simple, we aspire to inspire and motivate people to go out and experience what this world has to offer.

1.3 Project Application and Impact

The application of the project impacts different individual areas academically, industrially, and societally. In terms of academics, each of the developers has spent a decent amount of years in building strong software engineering skills during the individual college journey. In other words, this project is essentially a great platform to demonstrate each individual's ability to perform the knowledge they have gained over the years. Some of the industrial and societal benefits of this application include: inspiring users to travel and to be active, helping the community by recommending local events, and creating social relationships amongst other users.

1.4 Project Results and Deliverables

The project is expected to produce a variety of deliverables upon completion. Such deliverables include:

1. Design review documentation
2. Implementation
3. Fully developed iPhone application

The design review is a document that provides a detailed explanation of the application design to cover all aspects of the project. The design review provides developers and interested parties sufficient product information to gain a better understanding of the overall system and the design and purpose of the application.

BuckIt! is developed to be a user-friendly mobile application so that users get the best experience using the app as much as possible. The final working application will be tested and demoed amongst peers once all functions are developed.Being able to demo the product to others will provide the team sufficient user feedback for application improvement. We hope to publish this in the Apple app store in the future when BuckIt! has been reviewed, tested, and redesigned.

1.5 Project Report Structure

The following sections of this report provides information about the BuckIt! application. Chapter 2 covers the project background and related work such as state-of-the-art summary and literature search. Chapter 3 describes the project requirements which consists of multiple diagrams, functional requirements, non-functional requirements , business domain, and business requirements in detail. Chapter 4 explains the system design by breaking the section down into different components and system constraints description. Chapter 5 will focus on the system's implementation, major implementation problems and challenges, and detailed solutions such as techniques, algorithms, methods and technologies used. In chapter 6, we discuss tools that were used to develop this project and chapter 7 covers experimental approach and testing. Finally, chapter 8 focuses on the conclusion of the project and what we intend to do in the future.

Chapter 2 Background and Related Work

2.1 Background and Used Technologies

One of the necessary backgrounds that is required for the development of this project is knowledge writing code in Swift; the programming language used for developing applications for multiple iOS devices. This project was purely implemented using the Swift programming language and the XCode IDE developer tool. The application runs on the current version of iPhone's operating system (iOS), which is version 11.0 as of March 2018. Similarly, sufficient knowledge of Firebase is required for this project since it is used as the application's main database. Facebook and Gmail authentication are integrated into BuckIt! for the convenience of users logging in to the application.

The architecture of the application relies on the traditional Model-View-Controller (MVC) which allows for better organization for the structure of the system. This way, we have a structured application that is easy to modify and test. Producing a BuckIt! website was planned, but due to time constraints a website will not be delivered. However, a website will be available in the future.

Table 1: Relevant Courses Taken

Course Name	Description
CMPE 137- Wireless Mobile Software Engineering	- iOS course that taught the fundamentals of iOS development - Learned MVC architecture - Learned Swift
CMPE 131 - Software Engineering I	- Ruby on Rails Project - MVC architecture
CMPE 133 - Software Engineering II	- UML diagrams (drawing/analysis) - Software architecture
ISE 164 - Computer Human Interaction	- Device analysis and design - Overall UI design and decision making to increase usability

2.2 Literature Search

Everyone in this world wishes to have an adventurous and fulfilling life. Being around others while traveling or partaking in fun activities creates lifelong memories. As people, we strive to be happy and we use experiences and memories to fulfill that goal. There are plenty of ways that a person can plan out things to do in life, however, the most common way of organizing all the things you wish to experience is by creating a bucket list. The idea of the bucket list originated circa 2006 as a way to keep a tab of the goals that you wish to pursue before you die. The list usually contains things such as seeing the Northern Lights, skydiving, rock climbing in Arizona, etc. Many people create bucket lists, yet the problem is that people lack the motivation to fulfill those goals.

BuckIt! is an application that allows users to create their own personal bucket list.
This app serves as a location-based activity suggestion platform that will allow users to search for activities to do in a given area. Suggestions will be posted via locals and personal bucket lists can be displayed for other users to see. Popularity of the activity is determined by the users by choosing to the activity. A list of suggestions that are posted can be upvoted y other users. This

allows other users to see which activities are the most popular. This will allow users to evaluate which activities are worth attending based on the upvote results.

The purpose of BuckIt! is to inspire users to go out and experience everything that the world has to offer based on the suggestion of locals. The following three articles below provide context to the appeal of bucket list as well as the benefits that are gained through it.

The first article titled, *Landscape and Well-Being: A Scoping Study on the Health-Promoting Impact of Outdoor Environments,* describes how being outdoor is very beneficial to our lives. The article mentions that being active outside helps to reduce stress, reduce social anxiety, and recover from depression. Furthermore, being active supports our physical health and improves our social relationship with others (Abraham, 2009). Being active also helps with our physical health, which has been proven to be directly correlated to mental health. One of the many goals of the application is to inspire users to go out and enjoy what the environment around them has to offer. In turn, this will hopefully result in a more active and healthy lifestyle.

Next, the article *Discovering Socially Important Locations of Social Media Users* discusses the importance of discovering social locations from social media users. A socially important location is defined as “places that are frequently visited by social media users in their social media life.” Knowing where social “hotspots” are located can be useful for advertising, recommendation systems, and urban planning. Currently, there is a naive algorithm made by Twitter that helps to locate socially important locations. These locations are found based on the location-based functionality of social media platforms such as Twitter (Dokuz & Celik, 2017). However, the authors of this article found that the results from the naive algorithm do not include data of the user preferences. The authors of this article added a method of recording individual user interest, which resulted in an algorithm that outperformed the previous algorithm. Similarly, BuckIt!’s voting system will allow the users to give either positive or negative feedback on a given activity post by upvoting and downvoting. By doing so, BuckIt! will have individual user preferences that will accurately depict which activities are best to do.

Finally, the article, *What’s on Your Bucket List?’: Tourism, Identity and Imperative Experiential Discourse,* goes into detail about the inception of the bucket list and what it means to us today. According to this article, the bucket list creates “ the notion that travel experiences offer self-fulfillment and are a measure the success or meaningfulness of one’s life” (Thurnell-Read, 2017). The idea of a bucket list makes people want to go try new things that they have never tried before because it gives them fulfillment and a measure of perceived success when they actually accomplish things on the list. Thurnell-Read writes about how the bucket list is a facet for self-fulfillment and self-actualization. The experiences that people have change their identity, which is why many people want to travel around the world. By experiencing the world, users can have a different opinion on life. In fact, this is essentially one of the main goals for building BuckIt! because users will have the opportunity to create and share their own bucket list.

In conclusion, it is crucial for people to keep exploring the world and creating new experiences. From bucket lists to random activities, social media is one of the many platforms for people to find creative things to do. In that way, BuckIt! provides a means of keeping track of one's bucket list so that they can experience the world in an authentic way.

2.3 State-of-the-art Summary

Modern technology has changed the way we live in a drastic way. The beginnings of social media came with the desire to connect with people at all times, and yet it seems as if it has done the opposite by allowing us to "connect" from the comforts of our bed. Social media has changed our society and made us more focused on the pictures on our profile than the quality of our lifestyles. BuckIt! is a revolutionary application that encourages a social lifestyle by using technology as a guidance to go out and experience the world.

BuckIt!'s main function is unique in the sense that it provides users with authentic suggestions. For example, if a user wanted to go to Europe for vacation, other mobile apps such as Yelp would suggest popular tourist spots, like the Eiffel tower, Leaning Tower of Pisa, and so forth. BuckIt! provides a more authentic approach because suggestions are posted and upvoted by locals. Events suggested by locals and are rated by votes proves to be better because of the fact that locals know what is best where they live. For example, because In-N-Out is a burger restaurant unique to California, people know that In-N-Out is a staple for burger restaurants. However, Yelp reviews do not acknowledge the secret menu that In-N-Out offers, and yet locals could give you information about the secret menu, including which items on the menu are worth trying. It is for this reason that BuckIt! is unique and better than other similar applications. As a globetrotter, you will most likely only have one opportunity to visit certain places. This is the reason why it is important to make the most out of the trip. We believe that crowdsourcing the knowledge of locals will allow for a more authentic and overall better experience.

There are plenty of features that makes BuckIt! unique. BuckIt! features a user friendly map interface which allows users to view their current location and see a list of activities that are popular among locals within that location. The map interface is becoming increasingly popular in recent years because it allows for a geographical based view over a list view. Users can easily see where events are taking place on a map, rather than be given an address to an event.

More importantly, BuckIt! allows users to create their own bucket list. As stated in the article '*What's on your Bucket List?': Tourism, Identity, and Imperative Experiential Discourse*, people feel the need to create a bucket list because by accomplishing the items on the bucket list, it gives them a sense of identity and fulfillment. We should be doing the things we feel will improve the quality of our life so that we can be happy. By allowing users to have a bucket list, it will show activities that you want to do in the future as well as activities that you have accomplished. Being able to keep track of completed events is important because after a few years, it is fulfilling to see all the fun things that you have done and accomplished.

With new technologies and the right tools for development, we hope to encourage users to go out and live their life to its full potential. With a simple and user friendly application, it will be easier than ever to create your own bucket list. The map interface is designed so that it is easier for users to browse the cities nearby the user's location.

2.4 References

Abraham, A., Sommerhalder, K., & Abel, T. (2009). Landscape and well-being: a scoping study on the health-promoting impact of outdoor environments. *International Journal of Public Health,55*(1), 59-69. doi:10.1007/s00038-009-0069-z

The article explains how being outdoors would improve health benefits. The author did a study by a special representation of qualitative reviews. The results proved that being out on a landscape can give the potential to promote our mental, physical, and social well-being. For example, the landscape sceneries connect with our sensory such as sight, hearing, smell, and touch. The connection between the sceneries and our sensory are the reason why we would boost our attention restoration for our mental mindset. We would also develop a better physical health by walking around. Lastly, we can reduce our stress by having a social life with others at the landscape.

Dokuz, A. S., & Celik, M. (2017). Discovering socially important locations of social media users. *Expert Systems with Applications,86*, 113-124. doi:10.1016/j.eswa.2017.05.068

The authors A. Dokuz and M. Celik conducted a study about an efficient way of analyzing important locations based on the dataset extracted from the user's social media account. The authors gained inspiration to create a more efficient searching algorithm compared to the naive approach created by Twitter. In addition, various Twitter's APIs were utilized to gather useful dataset from their users. This data played a great role in defining and shaping the direction of the study because the dataset provided crucial information on the locations that they were focusing on. One of the conclusions is that each area provided a different set of audience. For instance, if a pin-point on the map is located near a stadium, then the majority of the users will be most likely posting feeds about a current sport event. In this case, the authors categorized this behavior as a group-level socially important locations while gathering data from a single user is called user-level socially important locations. Last but not least, there are some constraints for the overall efficiency of the algorithm such as the restriction from users about not voting on certain locations to provide useful information.

Thurnell-Read, T. (2017). 'What's on your Bucket List?': Tourism, identity and imperative experiential discourse. *Annals of Tourism Research, 67*, 58-66. doi:10.1016/j.annals.2017.08.003

This journal article analyzes the concept of bucket list and the idea of tourism, experience, and selfhood. The bucket list concept helps people shape their idea of what life should be. We use bucket lists to jot down a list of activities for us to do throughout our lifetime. This idea promotes the pursuit of meaningful experiences and it reinforces our values on our self-worth and if what we are doing is truly worth the time. Thomas Thurnell-Read analyzes the reasons why people crave for things to do to make life more meaningful. The article mentions the idea that we envisage our own identity and selfhood by creating experiences through traveling. On the contrary, however, while pursuing to complete the amount of things done should be fun there are many factors as to why establishing a bucket list is almost impossible to many people. These include but are not limited to: social, economical, and personal issues that stop many people to start one. Most importantly, an authentic feeling of being in a place is different than living it through the lives of others. People like the feel of doing something rather than seeing and hearing about it. Thus, people would rather experience things together rather than simply trying to live it through another person's shoes.

Chapter 3 Project Requirements

3.1 Domain and Business Requirements

The following sections consists of class diagram state-machine diagram, and activity diagrams:

3.1.1 *Class Diagram*

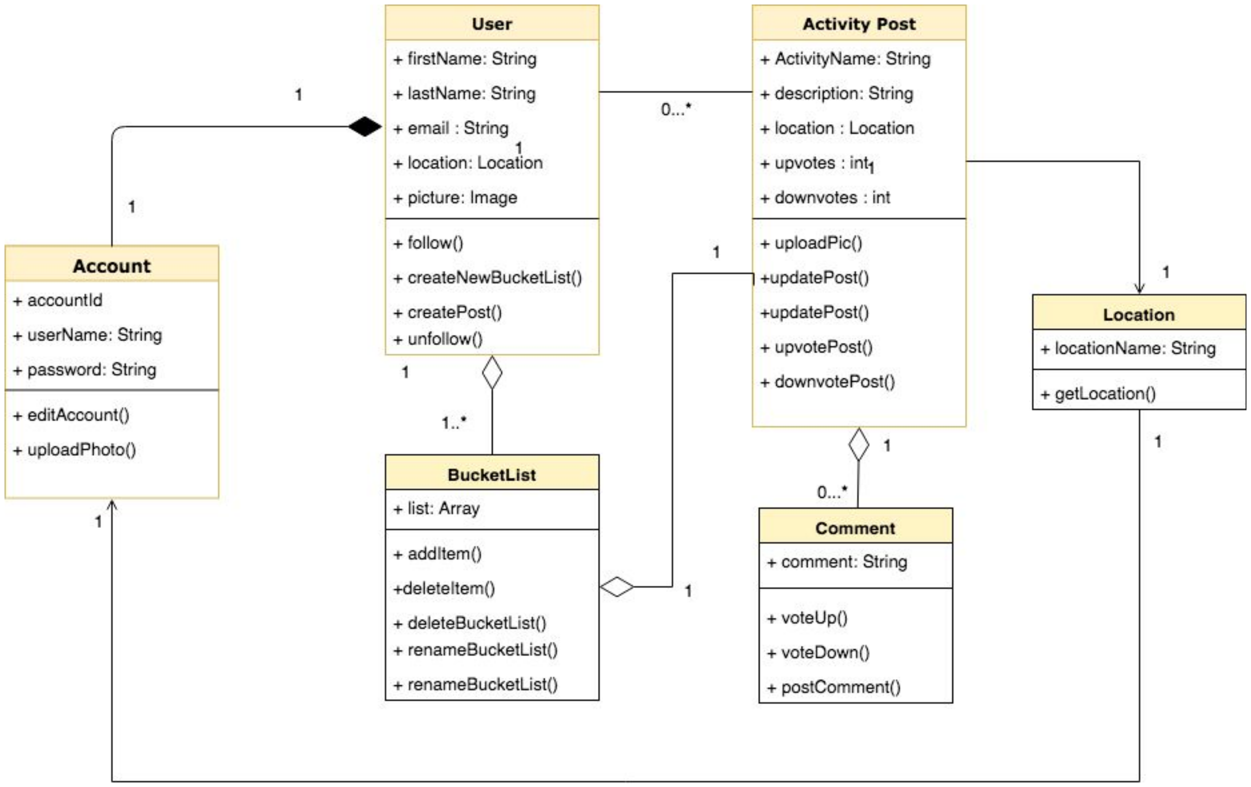


Figure 1. Class Diagram

3.1.2 *State Machine Diagram*

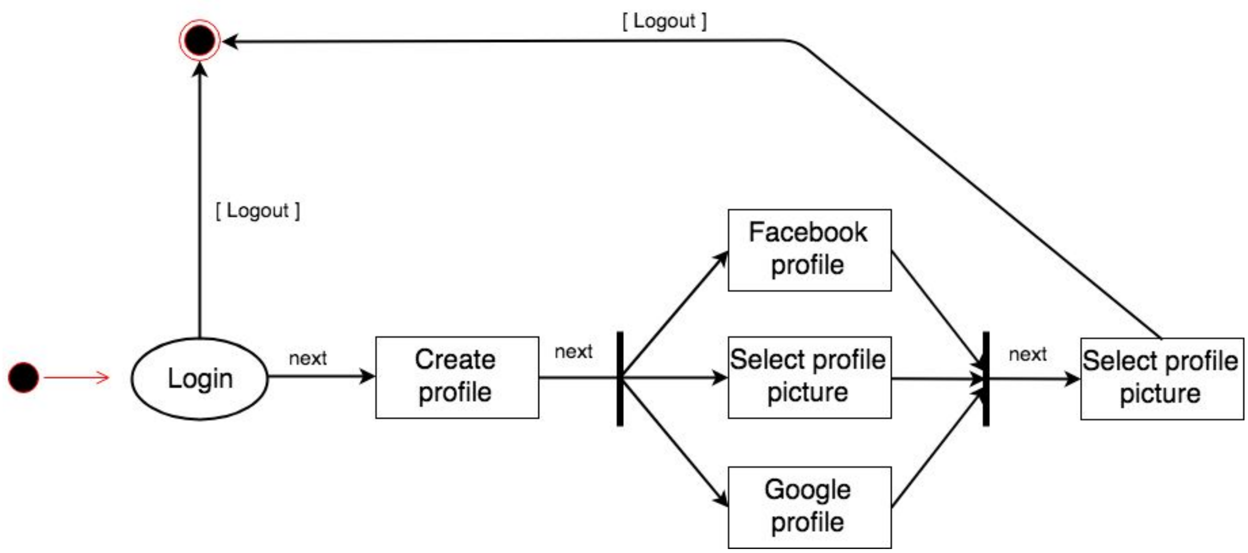


Figure 2. Create Profile

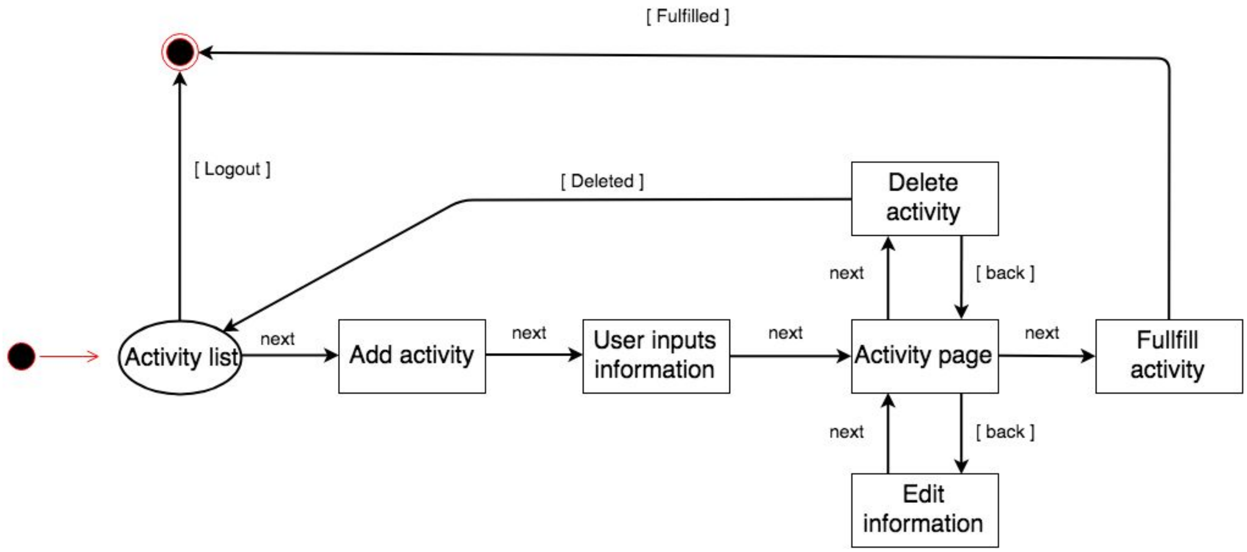


Figure 3. Create Activity

3.1.3 *Activity Diagram*

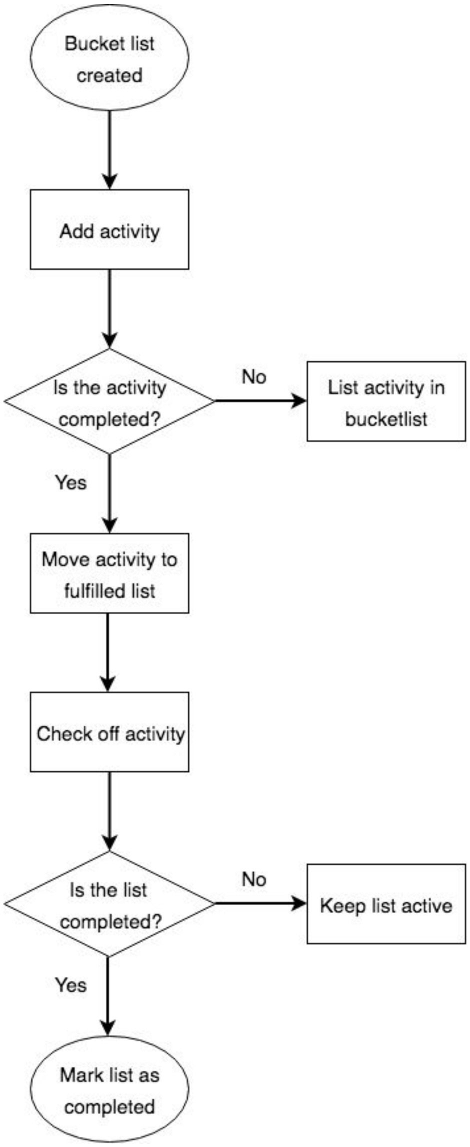


Figure 4. Bucket list activity diagram

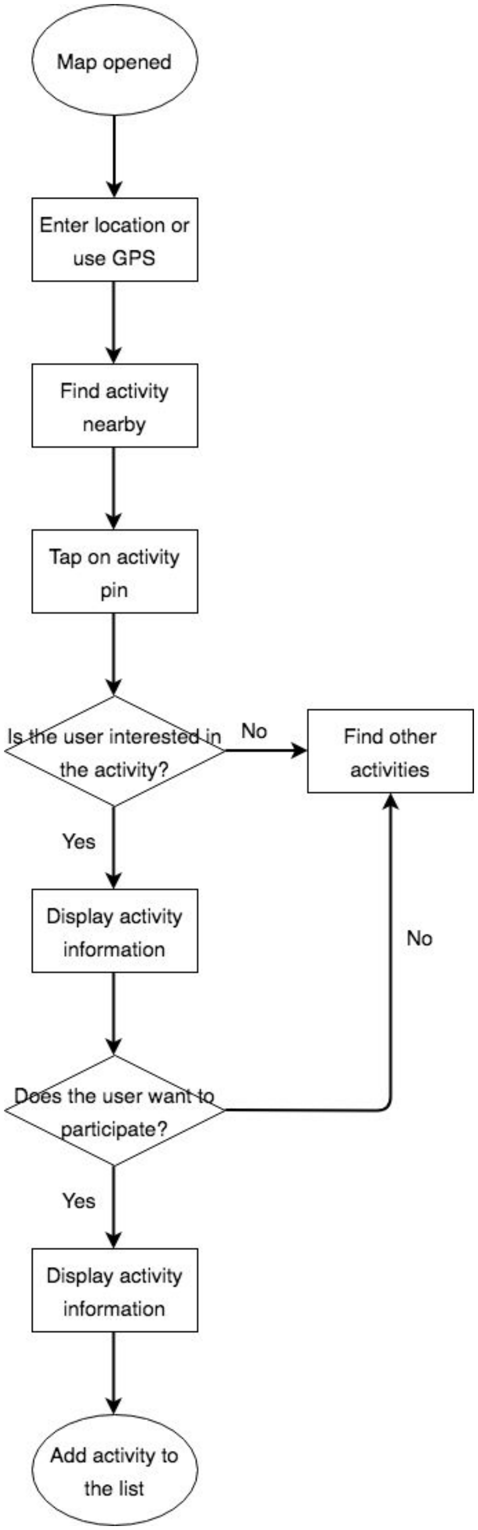

Map opened
Enter location or use GPS
Find activity nearby
Tap on activity pin
Is the user interested in the activity?
No
Find other activities
Yes
Display activity information
No
Does the user want to participate?
Yes
Display activity information
Add activity to the list

Figure 5. Find activity Activity diagram

3.2 System (or Component) Functional Requirements

3.2.1 User Account

- REQ-UA1: Users shall be able to sign up for an account
- REQ-UA2: Users shall be able to log in to their account
- REQ-UA3: Users shall be able to log out of their account
- REQ-UA4: Users shall be able to activate their account
- REQ-UA5: Users shall be able to login using their Facebook account
- REQ-UA6: Users shall be able to login using their Gmail account

3.2.2 Vote

- REQ-VT1: Users shall be able to upvote activity suggestions
- REQ-VT2: Users shall be able to view upvote count

3.2.3 Bucket List

- REQ-BL1: Users shall be able to create their own bucket list
- REQ-BL2: Users shall be able to edit their own bucket list
- REQ-BL3: Users shall be able to delete their bucket list
- REQ-BL4: Users shall be able to fulfill their bucket list
- REQ-BL5: Users shall be able to scroll through their bucket list
- REQ-BL6: Users shall be able to view trending page

3.2.4 Geo Mapping

- REQ-GM1: Users shall be able to pinpoint activity location
- REQ-GM2: Users shall be able to comment on location suggestion
- REQ-GM3: Users shall be able to view events at their current location

3.2.5 Profile

- REQ-PRF1: Users shall be able to edit their profile
- REQ-PRF2: Users shall be able to upload a profile picture
- REQ-PRF3: Users shall be able to view their profile

3.3 Non-functional Requirements

3.3.1 Usability

- The user shall tap on the screen to trigger an action
- The application shall know the user’s current location
- The map interface shall glide smoothly

3.3.2 ***Reliability***

- The system shall authenticate users accordingly
- The system should restart the application within 3 seconds in case of failure while keeping the current changes
- The user's information shall be encrypted and stored in the database and should be readily available when needed.
- Database tuples shall not be affected should the system crash

3.3.3 Performance

- The system shall be able to add or delete activity within one second of adding or deleting the activity
- The system should display an error message within three seconds of a failed state
- The list of activities shall be updated in no more than one second for every activity added in the users list 99% of the time
- The upvotes and downvotes shall be logged in the page and seen within 0.8 seconds of the time the user has voted
- The user's profile shall display the information in no more than 1 second

3.3.4 Supportability

- The application is updated based on the most recent iOS version
- The user shall be required to update their current version once the latest version is available and running in the App Store.
- The application should be compatible with previous iOS versions should the latest version be available
- The application should be available on other mobile platforms in the future.

3.3.5 Implementation

- The system shall be written using Apple's Swift programming language
- Firebase shall be used as the applications database
- Github shall be used for the projects repository

3.3.6 Legal

- The product shall be copyrighted accordingly
- The system should clearly state the "Terms and Conditions" documentation
- Developers should provide proper accreditation to any third-party applications used during the development process.

3.4 Context and Interface Requirements

3.4.1 Context Diagram

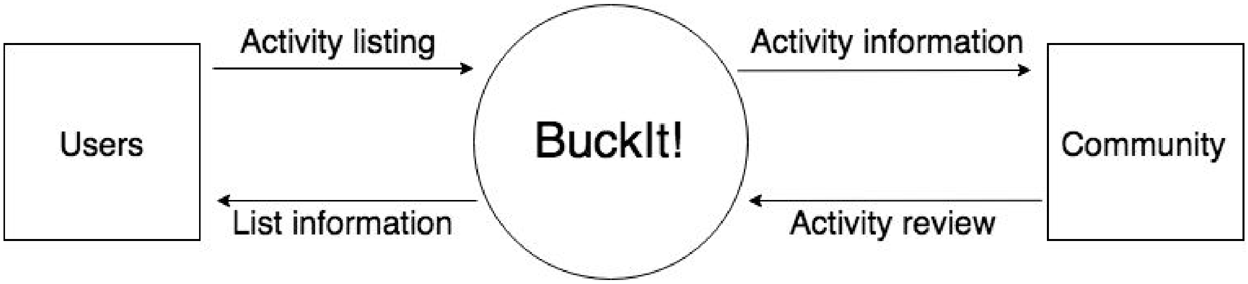


Figure 6: Context Diagram

3.4.2 User Interface

- The application interface shall provide the login for the user using Facebook and Gmail login
- The map interface should assist the user for easy navigation
- The map shall pinpoint location of the activity using geo mapping.
- Tapping on the location of the activity on the map shall direct the user to the listed activity
- The user shall swipe right and tap on complete to complete the activity
- The user shall swipe left and tap delete to delete the activity
- The user shall swipe right and tap on complete to complete the activity listed
- The Completed list of activity shall update once the user confirms the activity is complete

3.5 Technology and Resource Requirements

3.5.1 Hardware

Since the product is a mobile application there is no direct hardware being used. The physical GPS application being used is managed by the GPS application on the mobile phone. The use of this application requires an iOS compatible device such as an iPhone or iPad.

3.5.2 Software

Because the application is made for iOS devices, the application shall be written using Apple's own language which is Swift. Xcode is used to write and compile code for development. The database server being used is managed by Google since Firebase is being implemented as the applications database. Facebook and Google API is being used to support login features because users are able to login using their Facebook account or Gmail account.

Chapter 4 System Design

4.1 Architecture Design

BuckIt! was built based on the Model-View-Controller (MVC) architecture. This classic design pattern allowed us to have full control of the system without interruption to the overall system. This is also called the three-tier architecture that separates each tier into its own components.

The following figure provides a layout of the architecture of the system based on the MVC design pattern.

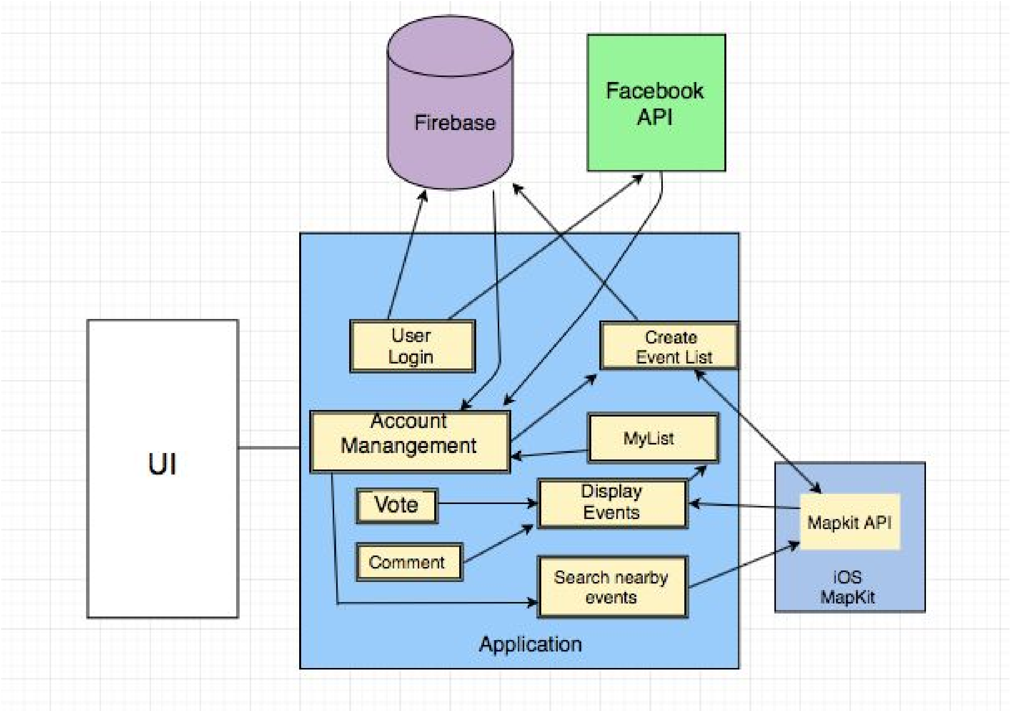


Figure 7: BuckIt! Project Architecture

User Interface

The main user experience window that displays the user's account information. It provides button controls for users for better user experience.

User Login

This represents the initial screen when the application is running. Users will be able to sign up by tapping on "create an account," which will send the user to another interface to fill out their information. Login information will be stored in Firebase after completing each step to create an account. Also, users will be able to log in by authenticating with their Facebook, or Gmail account. They must proceed with any one of the processes to get access to the application.

Account Management

This section of the architecture provides the main control room where users can create events, edit their profile information, and also have a list of their favorite events. For instance, users will have the ability to:

- **Create event list:** A running man with a plus (+) icon will be available for users to be able to create an event. Events will be stored in a list so that users will be able to see what events have been created. The form will have functionalities such as uploading a picture, name, description, event location, and date. Users must fill in the forms in order to proceed to create the event.

- **Edit profile:** An edit button will be available where users can edit their profile information such as user name, password, and other information.

Mapkit

The user will be able to activate the map tapping on the map icon located at the navigation tab on the bottom of the screen. The map view will appear which will show the current events taking place. Events will be shown via a pinpoint icon and uses a picture of the event uploaded by the user contained within the pinpoint icon at the location. When the icon is tapped, it will redirect the user to the event's detail interface where it shows the amount votes the event receives and all of the user comments. This section shall also provide the following features using the Mapkit:

- **Search Nearby Events:** all nearby events will be searched automatically when the user activates their location service. This is done through the maps autocomplete function.

- **Display Events:** All events will be displayed as a list format. This format shall show the details of the event such as pictures, name, description, and location. Users will be able to click on the location, and it will show the GPS navigation to the event. The Display Events also have the voting section and comment on the user interface.

    - **Vote:** This function is displayed using the bucket icon. When tapped, the icon will change into a star icon with a digit in the bottom right of the icon which indicates that the user has voted for the event.

    - **Comment:** This is displayed as a textbox where users are allowed to type comments about the events to inform others about their experience at the place.

4.1.1 Use Case Diagrams

In order to help us understand the internal system design we created a use case diagram to help us visualize on how the application works. The system has 6 features such as create profile, browse activities, post idea, create bucket list, follow, and unfollow users.

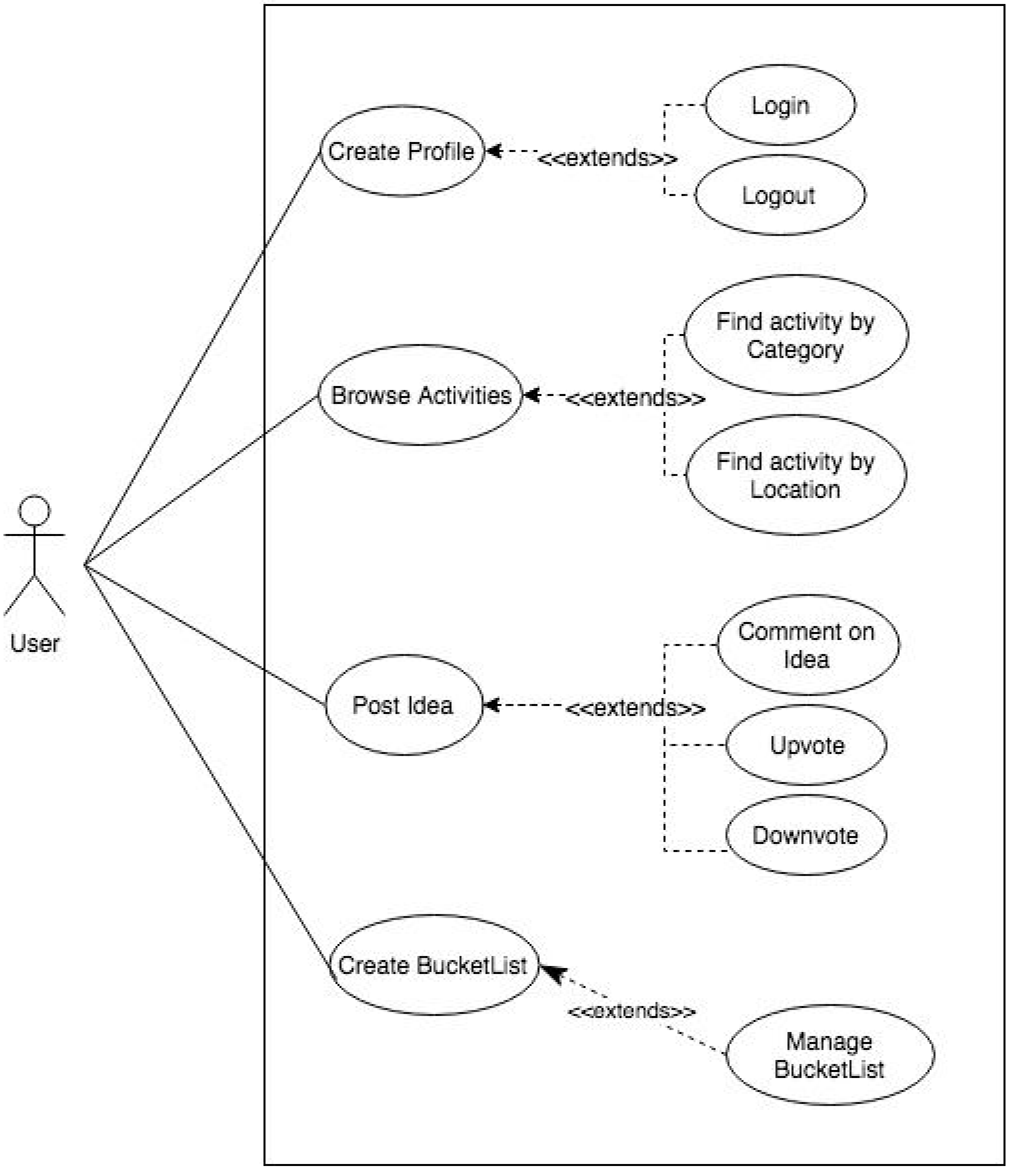


Figure 8: BuckIt! Use Case Diagram

4.2 Interface and Component Design

The user interface of our application consists of simple screens that provide easy navigation, which increases the usability and enhances user experience. Similarly, the fixed bottom navigation bar allows users to have access to any functionality at all times regardless of where the user is currently browsing.

As seen below, the main screen is composed of two sections (see figure 9). First, the top part of the screen shows the logo of the mobile app. The second screen is designed so that the user has the ability to signup for an account in the case they are a new user, or the user can opt to login if they are already registered (see figure 10). After the user is successfully logged into the application, the user is redirected to the profile screen where it shows their basic information such as full name, username, number of buckets, number of completed activities, number of followers and following friends within the platform. Also, under this section, there is a list of a series of buckets accompanied with a picture and a title that the user chooses to set on.

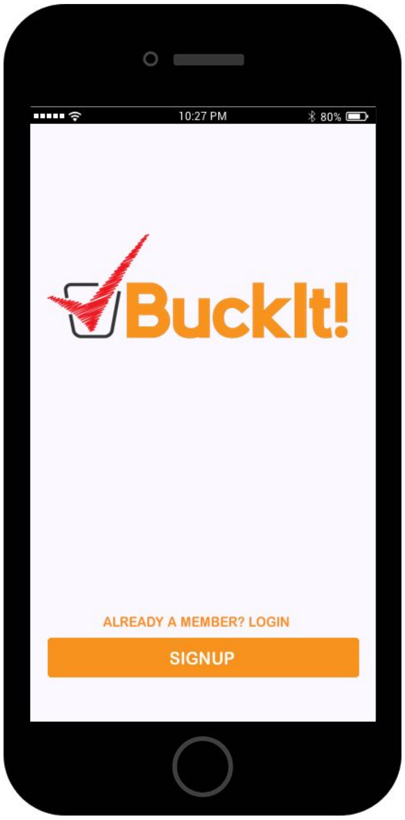


Figure 9: Main UI

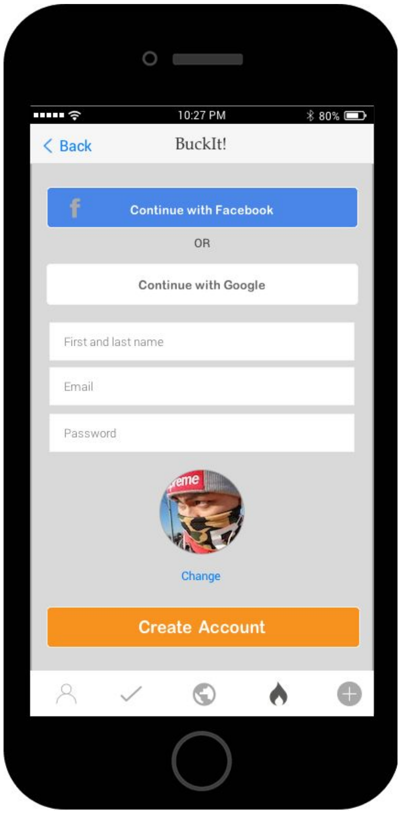


Figure 10: Login UI

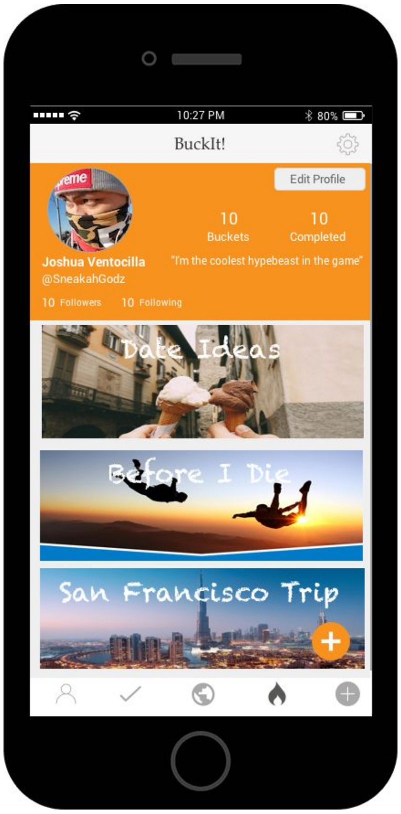


Figure 11: Profile UI

In addition, once the user has completed an activity, it will be stored on a separate screen for future references (see figure 12). We also offer a user-friendly map layout for users to have a better visual on the suggested activities around the user's current location (see figure 13). This greatly improves the user experience because it allows the user to have a reference on the actual location with respect to other places that he or she might be interested in exploring.

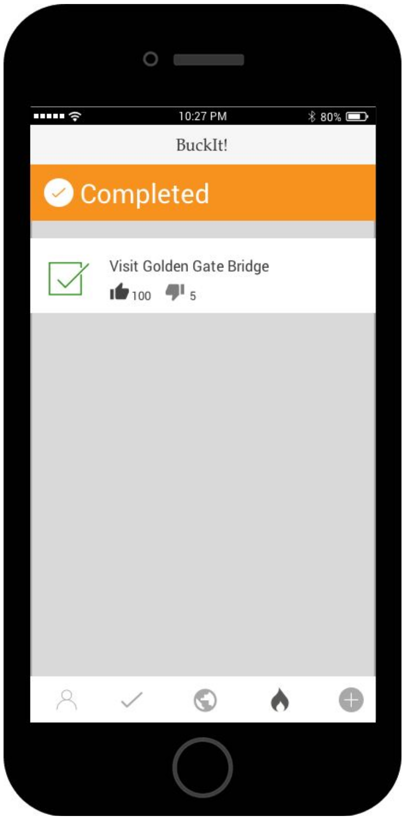


Figure 12: Completed activity UI

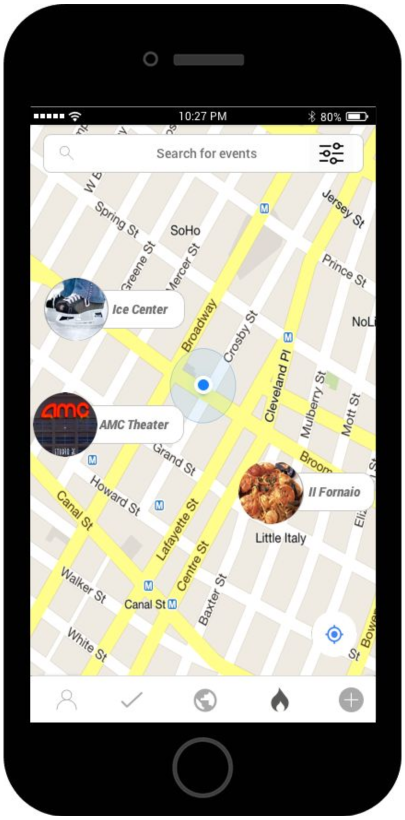


Figure 13: Map UI

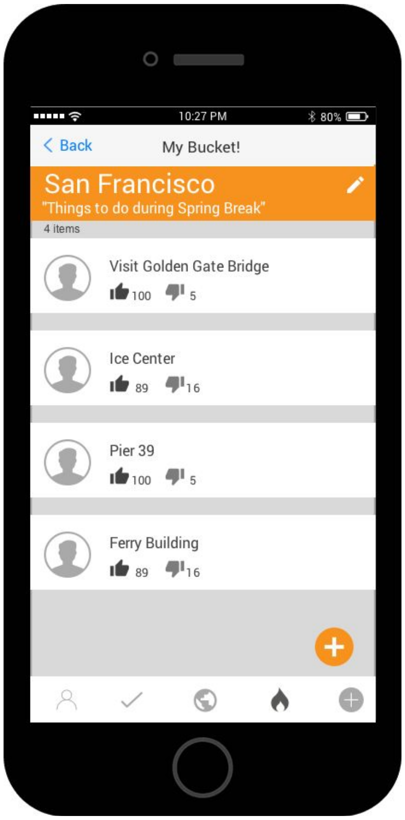


Figure 14: Bucket-List UI

The application contains four main components for the well functioning of the overall performance, with the help of two interfaces. Each component is essential as each of them satisfy some portion of the functional requirement, or minimum marketable features. More details:

- **Profile:** This essential component allows the user to create an account with his or her basic information that will be securely stored and encrypted through the third-party platform Firebase. For this reason, this current component requires the Firebase interface in order to perform the activities.

- **Bucket List:** This second component provides the main concept of our application. As mentioned above, the app allows users to create their own bucket lists, in which they can add certain activities that they would like to keep track. For this reason, this component is closely related to the activities component. Similarly, bucket list component's data is stored in the database that Firebase hosts.

- **Activities:** This component is another reason why the app exists. As an event suggestion application, this component is crucial because it holds basic information such as activity and location details of each of them. This information is also saved in the cloud via Firebase.

- **Mapping:** This last essential component allows to have a location based platform in the app. Since this component is implemented through the Apple's native mapping feature, the Mapkit interface is required for this project. Similarly, this component is closely related to the activities component because it provides the coordinates that correspond to each activity. In other words, the same event that was created in the previous component will reflect in the map.

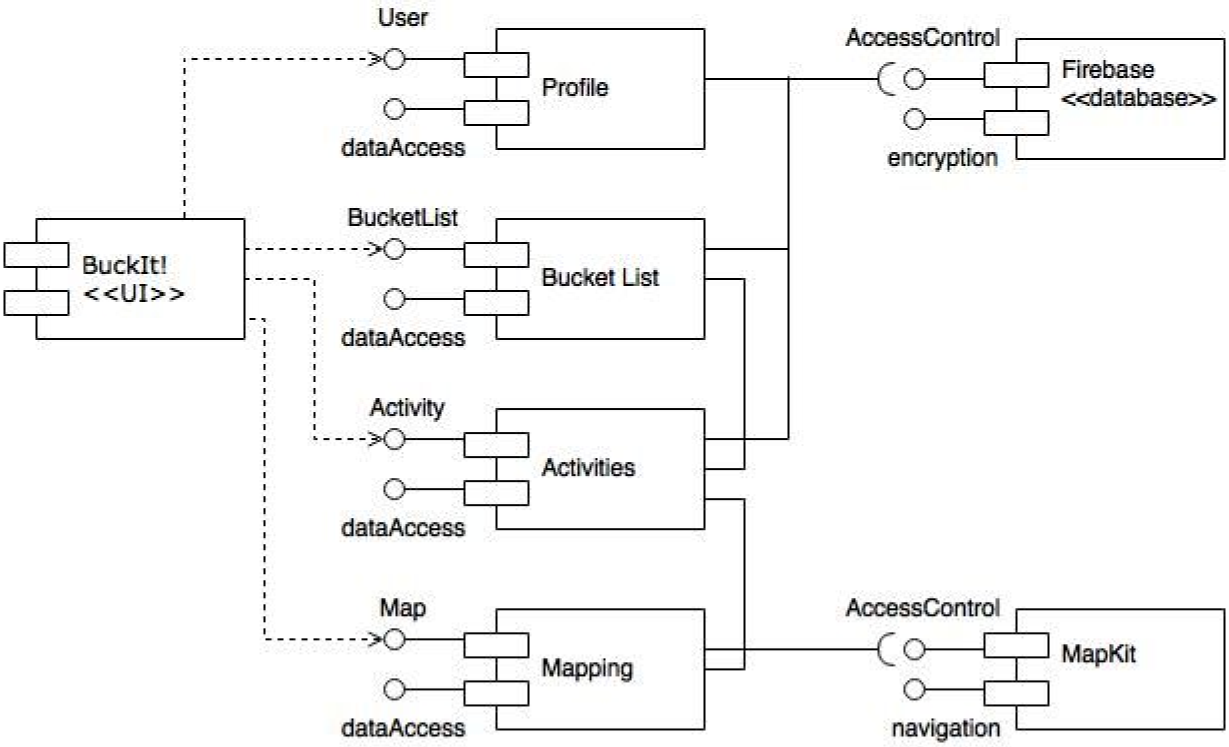


Figure 15: BuckIt! Component diagram

4.3 Structure and Logic Design

The basic structure of the application consists of the first interaction from the user through their iPhone device, which is obvious because the app would not be usable without it. Once the users have successfully installed the BuckIt! on their device, they can start using it right away. In the picture below, you can see that the application will prompt the user to the main UI (see figure 9) and the data interaction in the backend is via internet connection that attempts to communicate with the firebase database stored in the cloud. If the authentication process is successful, then users are able to explore different functionalities on the app as described above.

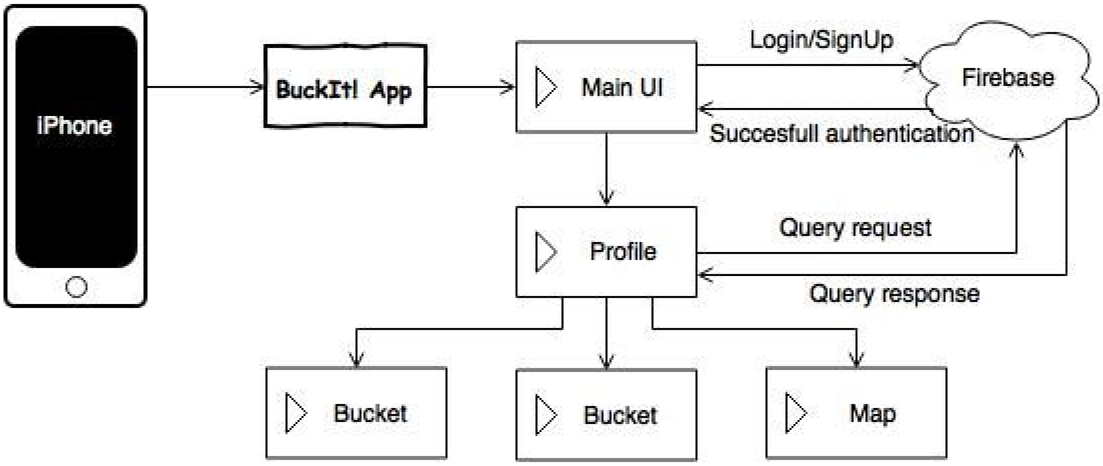


Figure 16: BuckIt! Structure and logic of the data flow

4.4 Design Constraints, Problems, Trade-offs, and Solutions

4.4.1 Design Constraints and Challenges

The team will be anticipating specific constraints throughout the development of this application. Some of those design constraints include: economic, resources, society and environment, software, safety, and reliability. The following constraints shall be discussed below.

Economic
Competition with other applications proves to be a constraint for this project. Because the economy is doing well and there are plenty of applications that are readily available in the app store, being able to get other users to use BuckIt! will be a challenge.

Resources
There are certain constraints that fall under this category such as time and money. This team consists of full-time students with little to no money. Thus, we are developing this project using free resources and tools. Time is also a constraint since everyone on the team is a full-time students. Certain team members also have to manage their work schedules as well. This means that we are not fully dedicating the majority of the time on this project since there other classes and jobs that require attention.

Society and Environment
Social and environmental constraints are not applicable to this project. This is more direct towards hardware components and not software components.

Software
The team has a diverse knowledge of different programming languages. This could pose as a constraint because some could be more proficient in using Swift than others. Additionally, since Swift is a developing language, future updates to Swift could lead the project to have unexpected bugs. Swift 4 has just been released, and some of the team's developers are unfamiliar with the new updates to the language.

Safety
Safety constraints are not applicable to this project by default since this project is using tools that comply with safety standards.

Reliability
This product must be reliable; however, we do anticipate certain constraints within the development of this project. Some of this constraints include: systems must be up all the time, the application should always be accessible, the application should not shut down at any given moment, operational errors, and inadequate testing. The system must always be accessible by users and should not shut down at any time. Operational errors and inadequate testing should also be considered a constraint.

4.4.2 Design Solutions and Trade-offs

Just as discussed above, we are anticipating a series of constraints that directly lead towards challenges that we need to overcome as a team. The overall design for the development of the BuckIt! app required tons of effort and time that have helped us to come up with the best system design as possible. Since constraints come in various areas, it is essential to discuss the solution and tradeoff on each of the areas discussed above.

Economic
We are aware of the difficulty to boost the popularity of the BuckIt app, but we also know that the right commitment and dedication to finalize the app is our main priority. We are not only working on this project for the sake of completing another one but also inspired to be a potential competitor that can change the world with our app. As a result, we will gain tons of experiences upon completion of this project as one of the tradeoffs.

Resources
Surely, time is a critical aspect of the entire project. To overcome this challenge, we need to have a serious commitment and time management to accomplish our goals. Although it may seem to be a hardship, the team is aware of the positive outcomes. Consequently, some sacrifices will lead to a successful project as a tradeoff.

Software
As engineers, the team members are always ready to work through the process of learning any software as needed. A solution for this area is directly related to the fact that the team is obligated to keep up with any recent updates, so the potential changes are taken care sooner rather than later.

Reliability
This area is always important and challenging at the same time. As a solution, the team needs to be aware of any potential bugs at all times; in this way, the team will have the ability to fix them at early stages of the development process. Consequently, the tradeoff is to have high-quality working code that increases the reliability of the app, even after deployment.

The team is aware of all the constraints and challenges during the development of this mobile application. Although some areas are more challenging than others, each of team is surely important to take care of at some point. Just as one proverb written by Ray A. Davis says, “A challenge only becomes an obstacle when you bow to it.” Indeed, the team is ready to face challenges as it comes.

Chapter 5 System Implementation

5.1 Implementation Overview

BuckIt! is a location based event and activity suggestion application. This application is able to create and organize bucket lists for users while also being used as a platform to find activities available for participation using the user’s location. BuckIt! is implemented using Apple’s own Swift language as well as using Google’s Firebase for the application’s database. Several API’s are implemented to help with a more convenient login such as Facebook and Google login API. Github is also used for the projects repository.

5.2 Implementation of Developed Solutions

BuckIt’s user interface is implemented through Swift’s feature of creating the design by using storyboards. There is no current algorithms implemented, however, future algorithms regarding search will be implemented.

5.3 Implementation Problems, Challenges, and Lesson Learned

Implementation was a challenge when it came to developing BuckIt!. Furthermore, there were several problems that we came across throughout the development process. One of the few major problems we faced was developing the user interface. Creating the user interface and making sure the constraints were properly set was difficult and had to be adjusted for most of the devices that will be compatible with the application. Using Github was another challenge because of the potential bugs that had to be fixed every time there was a push and pull to the repository. This was even more difficult because half of the team were not familiar with using Github and had to be taught by other members of the team. Also, certain pushes to the repository almost likely created bugs within the application which had to be fixed before moving on to the next part of the project. Lastly, because we are all students, our time and schedule still continue to prove as a challenge when it comes to developing the application.

Because of the amount of potential bugs that pushing to and pulling from the repository could cause we agreed to set up a main branch that consists of sub branches relating to the features being implemented. This helps with being able to work on different features at different times. Only when that branch is completed is the branch then pushed to the master branch. Also, by practicing this method, the team is able to pull from the repository and acquire the latest push so that every team member has access to the updated code.

Chapter 6 Tools and Standards

6.1 Tools Used

A part of the success of this project is dependent upon the use of certain communication tools and software tools. The following tools were used over the course of this project.

6.1.1 Modern Engineering Tools:

6.1.1.2 JustinMind

In order to prototype our application, we used a free trial version of a prototyping tool called JustInMind. This tool helped us develop a functional prototype which allowed us to see what pages we needed and how they should interact with each other. The UI was also implemented, and it showed us what elements we needed on each specific page.

6.1.1.2 Trello

Trello is a project management web application that allows our team to track progress and see what is still on the product backlog. Since we are exercising agile methodology, this tool is especially helpful because it allows us to easily change our goals in the middle of the project. It also allows for easy management, since all of our information is in one place.

6.1.1.3 Slack

Slack is used as the main communication tool to keep everything on track. Slack allows the team to create multiple chat channels to help us organize our topic discussions. For example, there is a general channel that is used to discuss general questions about the project. There is also a meeting channel to discuss questions regarding meetings and when it would occur. Slack also gives us the ability to set reminders for meetings and deadlines to guide us through our project development.

6.1.1.4 Google Hangout

Google Hangout is our secondary option for group communication. The team uses this application whenever team members are not available to meet in person as a group due to individual schedules. Using Google Hangout allows each member to participate chatting so everyone is current and up to date on the project.

6.1.2 State of the Art Tools

Because this application is currently for iOS devices only, it requires specific software tools to complete while using the necessary development tools to assist with completing certain features. The application that is chosen to develop this project is using the Swift language from Xcode. Xcode is used to build everything for this project. We also used Facebook and Google API for the single sign-on feature so that users can easily authenticate themselves while using the application. Lastly, we used Firebase to store data for the application.

6.1.2.1 Git / Github

In order to maintain our project, we have chose to use Git and Github for version control. This allows us to all have the current version of the project and allows us to work remotely without having to meet up in person. We had to learn how to use Github, and it has been a very useful tool for our team.

6.1.2.2. Canvas

Canvas is a state of the art tool that we are using to submit our project documentation. This web application allows us to send documentation to our advisor. Submissions of documentation are seen by all members of the group. Also we refer to Canvas as a guide to check for project and assignment deadlines.

6.2 Standard

The standards followed in this project are used to make sure that the project meets certain guidelines. In terms of the software, we made sure to keep our version of XCode up to date and made sure to keep our project IDE up to date. In terms of our components, we did not use any deprecated code or components, in fear that those components would not be supported in the near future. Many of of icons are shaped like buckets, since our application theme is centered around the idea of bucket lists.

For our design, we decided to choose a certain theme that is universal and standard to the entire project. Every page in the application follows this theme and implements similar design patterns. Our interface follows certain standards, such as the swipe feature to add activities to a bucket list from any page. Clicking on any activity or bucket will take the user to the profile of the activity or bucket.

In terms of testing, the application must be bug free. We have many regression tests to make sure that older features are working. The standard for testing is very high, because we need to make sure that the application does not crash at any time. Our documentation standards are very high as well, as we have documented the different features of the application very carefully. In terms of naming conventions, we are following the swift conventions for naming variables. We are commenting every method and in certain areas that are hard to understand.

Overall, the standards that we have followed to this point will help us deliver a uniform and high quality product.

Chapter 7 Testing and Experiment

7.1 Testing and Experiment Scope

Functional testing, usability testing, and acceptance testing and unit testing are used to evaluate the application. Specific functionality and features of BuckIt! to be tested are listed below. In each of these systems, we have used the Definition of Done (DoD) approach.

The principal features to be tested are categorized in the following areas:

A. User profiles
B. Bucket lists
C. Activities
D. Geo Mapping

Each feature should meet the definition of done according to the following table:

Table 2: Feature description

Feature	Definition of Done (DoD)
User Profile	User's should be allowed to edit their profile. Bucket lists should be viewable in the profile page of the user.
Bucket List	User's should add or delete to their bucket list. The bucket list should be clickable that should direct the user to see the activities listed under the bucket list.
Activities	User's should be able to view, add, delete, and edit their listed activities. The activity should list the title, description, and location of the activity using geo mapping.
Geo Mapping	User activities should be displayed on the map upon creation of the activity. User's should also see current activities happening based on their location.

7.2 Testing and Experiment Approach

The main approach for the test phase includes certain black box testing such as functional testing, usability testing, and acceptance testing. Testing is to be done on the feature once the feature meets the definition of done. In this phase, functional testing will take place where each features are tested individually. Once the features needed are complete and ready then it will undergo usability testing where the interface of the product will be evaluated. Finally, the last phase will be acceptance testing where the product will be evaluated by the customer to see if the product meets the business requirements.

7.2.1 Functional Testing

This phase of the testing will focus on the evaluation of each minimum marketable feature that are implemented up to this point. Functional testing follows a set of test cases which determines whether the function passes or fails based on their individual criteria and is judged based on the expected outcome.

Table 3: Test case description for Gmail login function

Test Case #	1	Test Case Name:	Login with Gmail
Designed by:	Josh	Tested by:	Samnang, William
Short description:	User account requirement that allows user to gain access to the application using Gmail.	Completion date:	2/15/18

Step	Action	Expected System Response	Status	Comment
1	User clicks on the Gmail Sign In button	The Gmail login controller is loaded and prompts to user to input Gmail's email and password	Pass	Input accepted and the user is authenticated
2	Basic information such as email and full name is stored in the database	Basic information is successfully saved in the cloud Firebase	Pass	
3	The app requires user to upload a profile picture	User is able to select an image from the gallery	Fail	User clicked on save without uploading a picture
4	Repeat step 3	Repeat step 3	Pass	User is able to upload a picture before clicking on the save button

Table 4: Test case description for Facebook login function

Test Case #	2	Test Case Name:	Login with Facebook
Designed by:	Samnang	Tested by:	William, Michael
Short description:	User account requirement that allows user to gain access to the application using Facebook.	Completion date:	2/15/18

Step	Action	Expected System Response	Status	Comment
1	User clicks on the Facebook Sign In button	The Facebook login controller is loaded and prompts the user to input Facebook's email and password	Pass	Input accepted and the user is authenticated
2	Basic information such as email and full name is stored in the database	Basic information is successfully saved in the cloud Firebase	Pass	

3	The app requires user to upload a profile picture	User is able to select an image from the gallery	Pass	User is able to upload a picture before clicking on the save button

Table 5: Test case description for Bucket Page

Test Case #	3	Test Case Name:	Bucket Page
Designed by:	Michael	Design date:	William, Josh
Short description:	The user shall be able to create its own bucket list	Execution date:	3/15/18

Step	Action	Expected System Response	Status	Comment
1	User clicks on “Add” located at the top right of the profile page	New activity form is shown	Pass	
2	User fills up essential information requested on the form and submit changes	Data is saved after user clicks on the Submit button	Pass	
3	Latest bucket is added to the profile controller	Bucket is shown on the profile page after fetching data from the database	Pass	
4	Repeat step 3	Repeat step 3	Pass	

7.2.2 Usability Testing

This phase of testing will be based on the analysis of the design of the application. Each design screens will be analyzed and annotated. Annotations explain the reasons for the design and describes its heuristic evaluation.

7.2.3 Acceptance Testing

In this phase of testing, the functionality of the feature will be tested individually. Once the feature is finished and definition of done is met will then the product be delivered.

7.2.4 Unit Testing and Integration Testing

Unit testing will be performed on the major functions of the code. Testing is done for every major function of the application for consistency and maintenance. After all functions have been tested only will the application undergo integration testing.

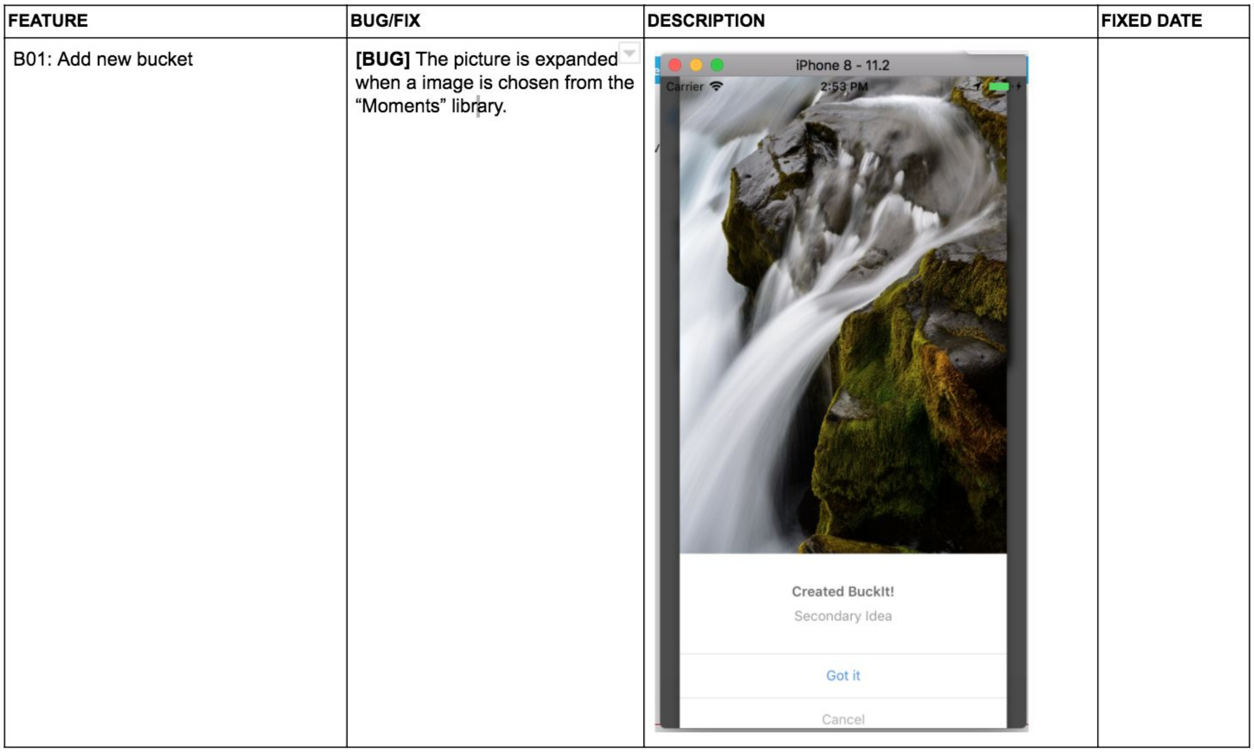
7.3 Testing and Experiment Results and Analysis

The team has thoroughly worked on fixing bugs as they appear during the implementation face. For this reason, every testing technique mentioned above serves a different purpose, and the careful application of the techniques allow us to build a more reliable and working mobile application.

Some of the experiment involved in this project includes the utilization of mapping API. In order to get the most of the mapping API, we experimented with an API called Mapbox at the beginning because this platform provides a good customization on the map layout. However, the lack of online tutorials regarding how to develop further obligated us to find a contingency plan. For this reason, we are currently making progress by using the native Apple MapKit instead of the first choice. The result of this experiment was pure knowledge of how Mapbox works because we could not actually use it on our mobile application.

The preliminary result of the testing include a successful execution rate where the majority of the test cases passes with the expected results from the system, even if the test case fails, we find a way to fix it as fast as possible. Similarly, we are planning to run regression and integration testing at the end of the feature implementation phase.

7.3.1 Bug Report Screenshots



FEATURE	BUG/FIX	DESCRIPTION	FIXED DATE
B01: Add new bucket	**[BUG]** The picture is expanded when a image is chosen from the "Moments" library.	iPhone 8 - 11.2 Carrier 2:53 PM Created BuckIt! Secondary Idea Got it Cancel	

Figure 17: Add new bucket bug

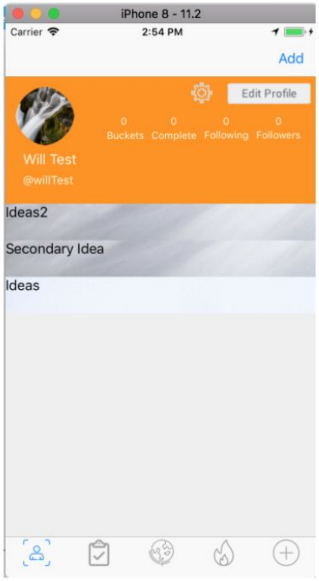
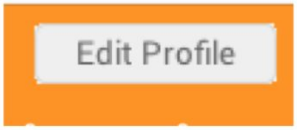
B02: New Bucket	**[BUG]** The latest addition of the bucket page is not fetched on the profile page. -> it appears if you make another request to the db like updating the profile page.	 iPhone 8 - 11.2 Carrier 2:54 PM Add Edit Profile 0 Buckets 0 Complete 0 Following 0 Followers Will Test @willTest Ideas2 Secondary Idea Ideas	
B03: Edit Profile	**[NOT BUG]**. This should be an actual button and not an image, so we avoid that it looks pixelated		April 1

Figure 18: Fetching bug

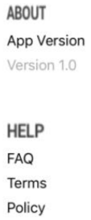
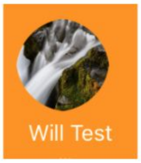
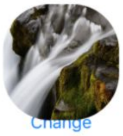
B04: Setting Page	**[NOT BUG]** Same as B03. Turn the 'About' and 'Help' tag as an actual label. Not an image.	ABOUT App Version Version 1.0 HELP FAQ Terms Policy	April 1
B05: Profile picture	**[NOT BUG]**. Image appears to be asymmetric		
B06: Edit Profile page	**[NOT BUG]** Change label is overlapping with the picture		

Figure 19: Settings, Profile, and Edit profile fixes

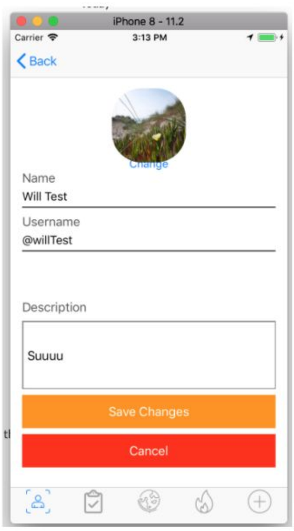
B07: Cancel button on Edit profile page	**[BUG]** The Cancel button does not trigger any action	 Name Will Test Username @willTest Description Suuuu 	April 1

Figure 20: Cancel button bug

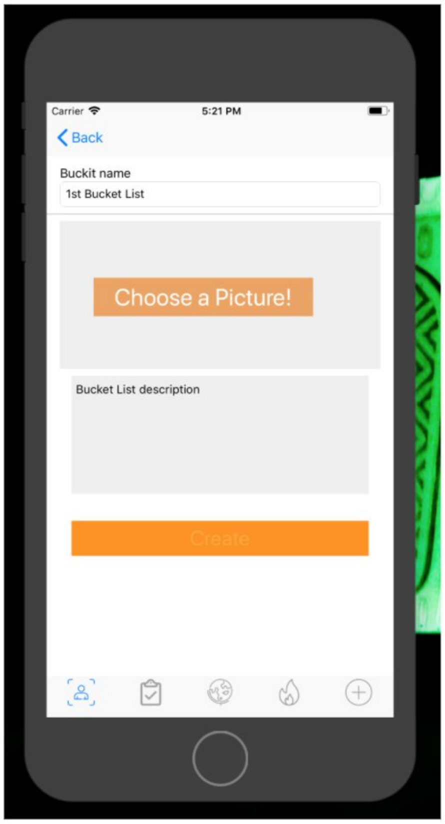
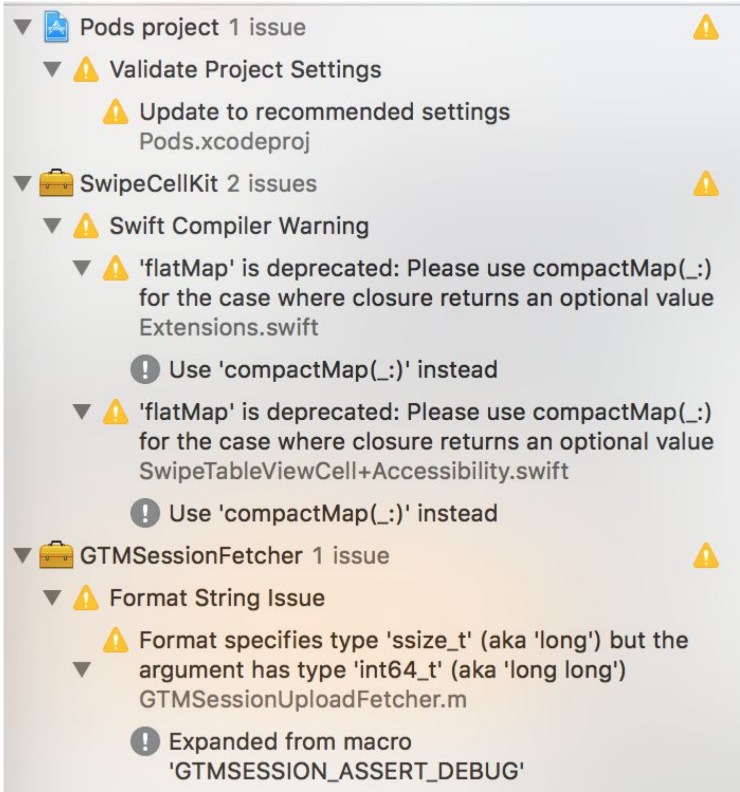
B08: Bucket List without picture	[BUG] Bucket list crashes when picture has not been uploaded	 Buckit name 1st Bucket List Bucket List description 	

Figure 21: Bucket list without picture bug

7.3.2 Warnings issued from Xcode



Pods project 1 issue

Validate Project Settings

Update to recommended settings
Pods.xcodeproj

SwipeCellKit 2 issues

Swift Compiler Warning

'flatMap' is deprecated: Please use compactMap(_:) for the case where closure returns an optional value
Extensions.swift

Use 'compactMap(_:)' instead

'flatMap' is deprecated: Please use compactMap(_:) for the case where closure returns an optional value
SwipeTableViewCell+Accessibility.swift

Use 'compactMap(_:)' instead

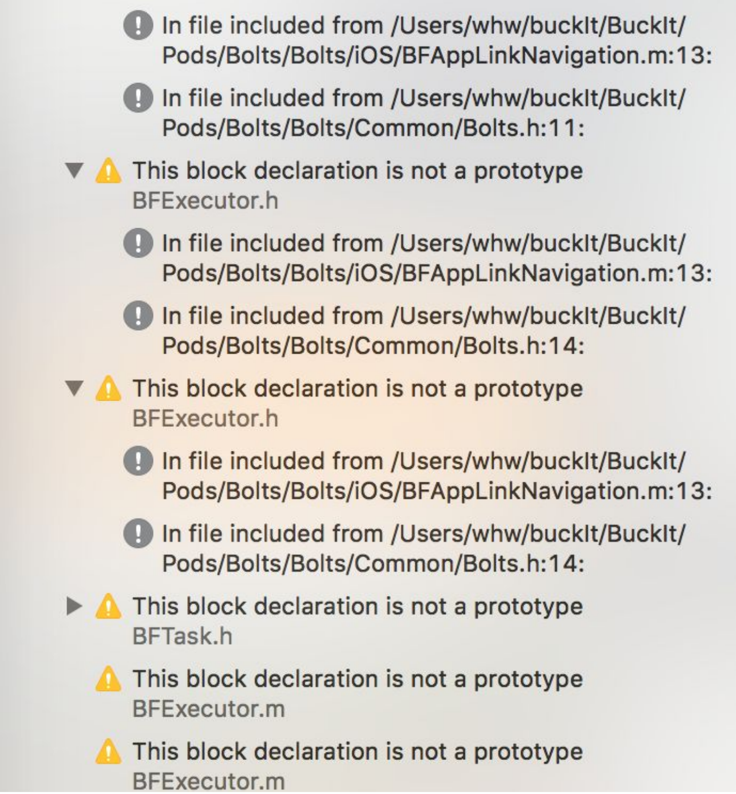
GTMSessionFetcher 1 issue

Format String Issue

Format specifies type 'ssize_t' (aka 'long') but the argument has type 'int64_t' (aka 'long long')
GTMSessionUploadFetcher.m

Expanded from macro 'GTMSESSION_ASSERT_DEBUG'

Figure 22: Pods warnings



In file included from /Users/whw/buckIt/BuckIt/Pods/Bolts/Bolts/iOS/BFAppLinkNavigation.m:13:

In file included from /Users/whw/buckIt/BuckIt/Pods/Bolts/Bolts/Common/Bolts.h:11:

This block declaration is not a prototype
BFExecutor.h

In file included from /Users/whw/buckIt/BuckIt/Pods/Bolts/Bolts/iOS/BFAppLinkNavigation.m:13:

In file included from /Users/whw/buckIt/BuckIt/Pods/Bolts/Bolts/Common/Bolts.h:14:

This block declaration is not a prototype
BFExecutor.h

In file included from /Users/whw/buckIt/BuckIt/Pods/Bolts/Bolts/iOS/BFAppLinkNavigation.m:13:

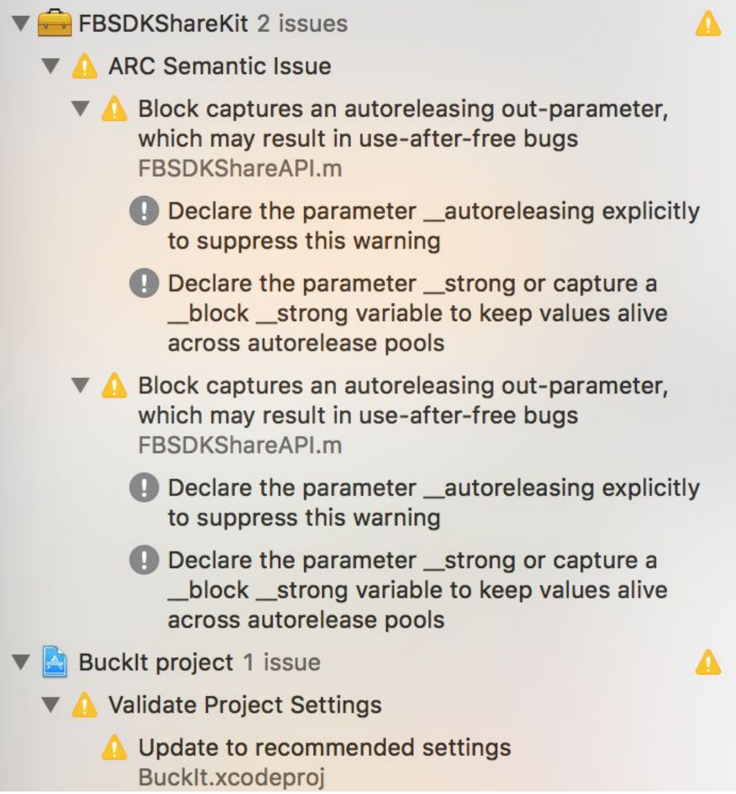
In file included from /Users/whw/buckIt/BuckIt/Pods/Bolts/Bolts/Common/Bolts.h:14:

This block declaration is not a prototype
BFTask.h

This block declaration is not a prototype
BFExecutor.m

This block declaration is not a prototype
BFExecutor.m

Figure 23: Pods warnings II



FBSDKShareKit 2 issues

ARC Semantic Issue

Block captures an autoreleasing out-parameter, which may result in use-after-free bugs
FBSDKShareAPI.m

Declare the parameter __autoreleasing explicitly to suppress this warning

Declare the parameter __strong or capture a __block __strong variable to keep values alive across autorelease pools

Block captures an autoreleasing out-parameter, which may result in use-after-free bugs
FBSDKShareAPI.m

Declare the parameter __autoreleasing explicitly to suppress this warning

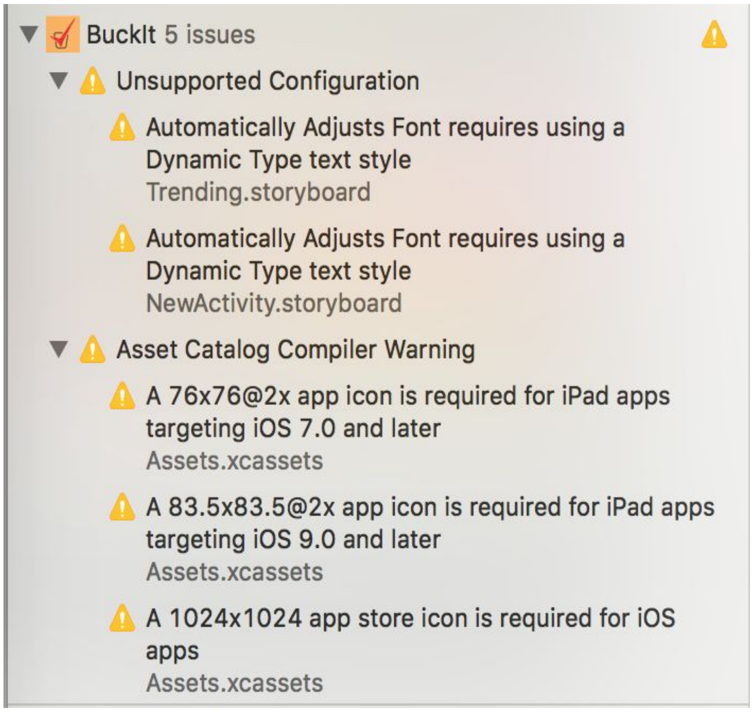
Declare the parameter __strong or capture a __block __strong variable to keep values alive across autorelease pools

BuckIt project 1 issue

Validate Project Settings

Update to recommended settings
BuckIt.xcodeproj

Figure 24: Pods warnings III



```
▼ ☑ BuckIt 5 issues                                              ⚠
  ▼ ⚠ Unsupported Configuration
      ⚠ Automatically Adjusts Font requires using a
        Dynamic Type text style
        Trending.storyboard
      ⚠ Automatically Adjusts Font requires using a
        Dynamic Type text style
        NewActivity.storyboard
  ▼ ⚠ Asset Catalog Compiler Warning
      ⚠ A 76x76@2x app icon is required for iPad apps
        targeting iOS 7.0 and later
        Assets.xcassets
      ⚠ A 83.5x83.5@2x app icon is required for iPad apps
        targeting iOS 9.0 and later
        Assets.xcassets
      ⚠ A 1024x1024 app store icon is required for iOS
        apps
        Assets.xcassets
```

Figure 24: Pods warnings IV

Chapter 8 Conclusion and Future Work

8.1 Conclusion

Even though the process of developing this project has been difficult and challenging, all of the minimum marketable features required for the project have been met based on the projects definition of done. We have successfully created an app that will allow users to create their own personal BuckIt! List and add activities into them. During this project, we followed the agile mindset as close as we could regardless of the hardship and challenges we faced during this course of the entire project. Similarly, each of us had the opportunity to learn new aspect of the iOS programming language, Swift, and the big takeaway from this experience is the ability to adapt to changes at any time. From unexpected software upgrades to limitations from certain technologies, we have always responded to these changes.

8.2 Future Work

Due to the time constraint and unexpected bugs during the development phase, we had to reconsider our game plan by leaving out some of the additional features that we had in mind since the beginning of this project. Thus, some of the future work include the integration of new features to our mobile app such as the ability to view other users' bucket list for future reference and add them. This way, we could include the functionality of adding/follow and removing/unfollow users from our "friend list."

References

Abraham, A., Sommerhalder, K., & Abel, T. (2009). Landscape and well-being: a scoping study on the health-promoting impact of outdoor environments. *International Journal of Public Health,55*(1), 59-69. doi:10.1007/s00038-009-0069-z

Dokuz, A. S., & Celik, M. (2017). Discovering socially important locations of social media users. *Expert Systems with Applications,86*, 113-124. doi:10.1016/j.eswa.2017.05.068

Firebase (2018, April). Add Firebase to your iOS Project. Retrieved from https://firebase.google.com/docs/ios/setup

Flaticon.com. (n.d.). Retrieved from https://www.flaticon.com/packs/essential-set-2

Price, M. (n.d.). iOS 10 & Swift 3: From Beginner to Paid professional. Retrieved from https://www.udemy.com/devslopes-ios10/learn/v4/overview

Thurnell-Read, T. (2017). ‘What’s on your Bucket List?’: Tourism, identity and imperative experiential discourse. *Annals of Tourism Research, 67*, 58-66. doi:10.1016/j.annals.2017.08.003

Yu, A. (n.d.). iOS 11 & Swift 4 - The Complete iOS App Development Bootcamp. Retrieved from https://www.udemy.com/ios-11-app-development-bootcamp/

Yp.py (2017, September). UI Datepicker for UI Text Field. Retrieved from https://www.youtube.com/watch?v=QBuYmP2p3AY