COMP 208 / GROUP 21

REQUIREMENT WALKTHROUGH REPORT

submission time: 12 noon on February 16th, 2018.

Table of Contents

Project Description

Mission Statement

The aim of our product is to provide a mobile application that can aid in travelling by reminding users to get off the bus and providing passengers with landmark information.

Description of Users

We have identified several different types of customers that our project would be aimed toward and who would benefit from the use of the end product.

Firstly, individuals that are travelling and are new to a specific area would use and benefit from our application. The reasons for this are firstly, they would be able to find specific bus stops and travel to them, our application would ensure that the user would always get off at the correct stop. This means that an individual who is travelling would be able to explore the place they are in without having to worry about getting lost or not finding the right bus stop. Travelling can be very daunting for people, especially if they do not speak the local language and so by using an application that can take the worry out of the situation, the user will be more inclined to go and explore therefore enabling them to see more sights and view more of the area as they do not have to worry about the travelling aspect.

Another potential user of our system are individuals who may be from the area itself but would not know the precise location of the bus stop that they are travelling to. Therefore, our application would assist as they would be able to input the name of their required bus stop and they would be notified when they would need to get off of the bus. This particular individual would benefit from our project as they could use their existing knowledge together with our application to find their way around a new section of an area they are already familiar with.

Lastly, we have also identified a third type of individual that would find a use in our project. When you are on public transport such as a bus, particularly at busy periods when there are a lot of people, you may want to do something else when on the bus such as put headphones in and listen to music or watch a film or perhaps read a newspaper. Therefore you may not realise your stop is approaching, however our application would make sure that you got off at the correct stop.

Another issue that someone may have when the bus is busy is the fact that there may be a large number of people on the bus, both sitting down and standing and so therefore you may not be able to see out of the windows and so consequently you would have no idea where you were at a specific moment in time. Our application would prevent an individual from missing their stop at busy periods when they may not be able to see out of the window.

Mission Objectives

The project we will be embarking on is a Mobile Application designed and developed for the iOS operating platform. The Mobile Application will provide users with a service in which they are able to search for a specific bus stop by typing the stop name. Our application will then alert the user notifying them of when there required stop is approaching so they should be prepared to exit the bus.

The application will provide notifications to users in two different ways. Firstly, when the user is actually using our application they will get a notification in the form of a pop-up notifying them that they need to start to make their way off of the bus as their required stop is approaching.

Secondly, if the user has entered their destination bus stop in the application itself but does not currently have the application on screen, for example they are on Facebook instead, our application will send a notification to the notification bar of the user’s phone. By utilising multiple forms of notifications that can be sent to the user, we are therefore ensuring the user will not miss their required bus stop as they will always be sent a notification.

An additional feature that will be included within our mobile application will be a ‘Landmark’ function. The purpose of this element is to offer an educational aspect to our application as the user is able to see famous landmarks along the bus route that they are on so they can learn about the local area. This is particularly useful if you are travelling or are new to the area enabling you to learn more about it.

Our application is intended to make life easier for even the most seasoned of travelers as arriving in a new location can be daunting and so our project would take some of these worries away. Our product is intended to be universally available to consumers so that everyone has the potential to experience, use and gain from our application as it will be universally available. The fact that our project is in the form of a mobile application means that it is available to the masses, however, it also means that our application is to a degree future-proofed. This is because we would be able to update our application and create and add new features to it allowing for constant innovation. Users would then simply download the update for our application straight to their mobile phone so they could experience the new features that we would look to add.

Statement of Deliverables

Description of anticipated software

functional requirements (traveler view)

When we envisage people using our application, we see a process that would firstly involve the user downloading the application.

Secondly, the user would open the application. Next, they would need to allow our application to access the Global Positioning System (GPS) of the users mobile phone so that the application can track the users position to alert them for when they should begin to move off of the bus.

Then the user would need to allow the application to send notifications. This is so that our application can alert the user for when they need to get off of the bus through the use of notifications.

Following on from this the user would need to allow for our application to use mobile data so a connection can be established to the network to gain GPS location.

The next stage requires the user to accept the privacy policy that will be included within our application. This policy will state how the users data will be stored and what will be done with it etc. The user will then need to read the instructions that will be shown to them in the form of a pop up to inform them of how the application works and what they should know.

After this, the user will need to input their required bus stop in to the search bar which will have an auto complete feature as well as a history feature for convenience.
The user should then select the required bus stop. Our application should then display the

current location of the user and track them while they make their journey on the bus towards the bus stop they have selected to get off at.

When the user is within a certain distance of the bus stop that they previously selected, they will get a notification in the form of a pop up if they are on our application or a notification sent to their notification bar if they are off the app alerting them that they should make their way off of the bus as they are approaching their stop.

When the user has reached their stop and is off of the bus then they can press done and return to the destination input screen or they can press continue button and keep tracking without further notification.

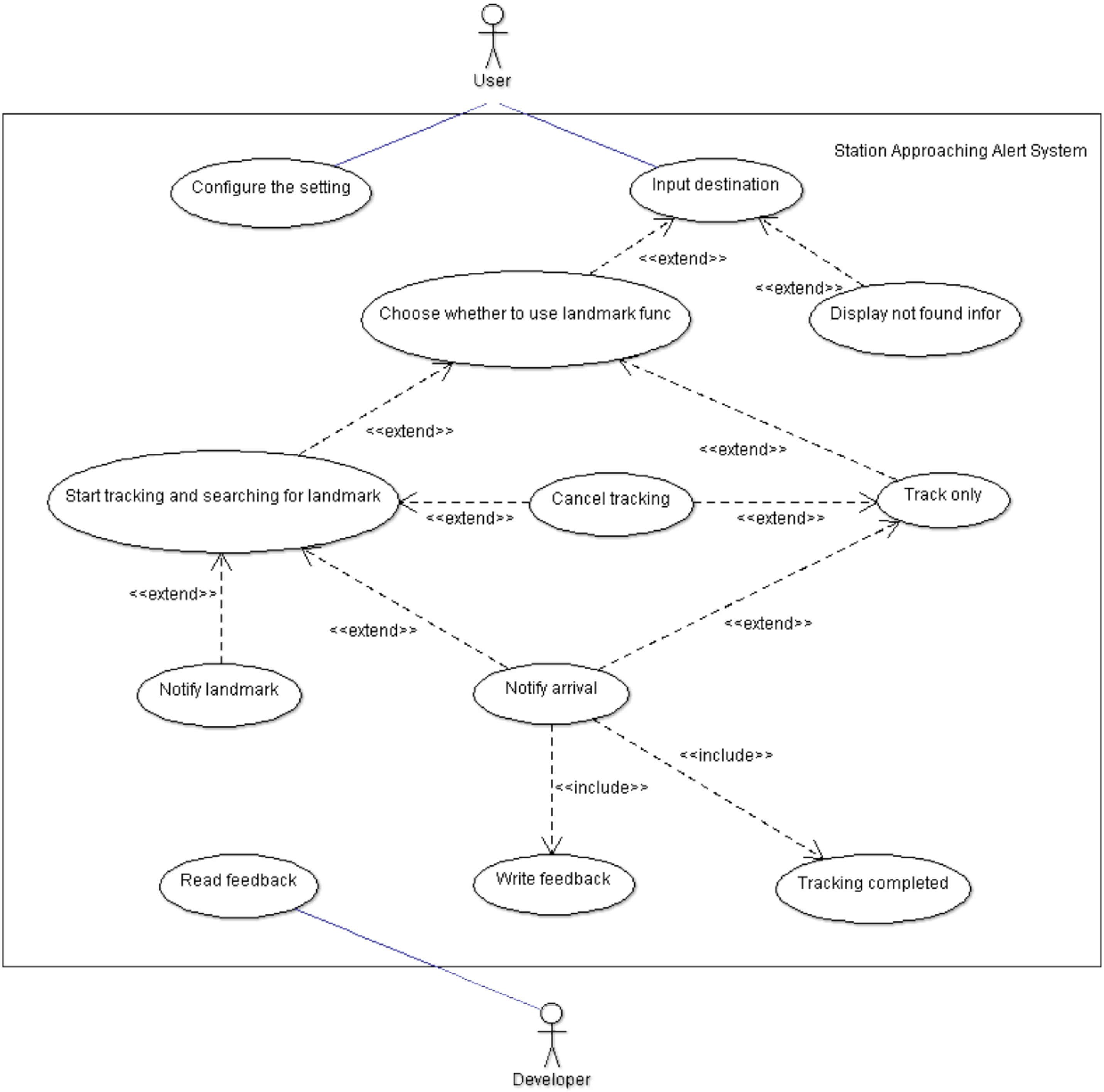
Before they do either of these things, our application will ask for some feedback so that we can collect data and find out where our application excels and where improvements could be made allowing us to make our application better to make the user experience easier and more enjoyable.

Another function for users is changing the configuration including distance parameter and landmark type.

functional requirements (developer view)

Developer will receive feedback from user and make improvement on the application.
After each journey the user takes using the app, the app would pop up a window inviting the user to provide feedback based on their journey experience. The feedback can either be simple useless – useful or detailed opinions about what they would like to see in the app for the future. By conducting statistical analysis on the result and reading the feedback, the developer can decide what to improve in the next version and whether or not the app is accepted by the user.

system boundary diagram



non-functional Requirements (Verifiable)

A. Accuracy:

By combining the multiple criteria including the distance proximity, the current speed of the bus and other factors, the app ensures that the user is notified at least three minutes (customable) before the bus arrives at destination.

B. Robustness:

The frequency that the app quits accidently (for the reason of itself instead of the operating system or other app) is strictly less than 1 times in every 100 executions.

C. Data Integrity:

The user location data is not usable by any other application outside the application and conforms to the local law about location privacy.

D. Speed:

The application could perform without noticeable delay when the user in within a network with speed faster than 1Mb/s

E. Cellular data requirement:

The application ensure it would not consume more cellular data than it required, which should be at most 1.5 times than using a well-established navigation app.

F. Usability:

The application is minimal and ensures that user without any previous experience could learn how to use the application within 5 minutes exploring

Constraints

Firstly, the mobile application is constrained by the operating system which iOS development requires. The minimum development operating system requirement for iOS application is Mac OS X 10.6 which has low compatibility with current major operating system such as Windows [1].

Secondly, programming language is also a constraint while developing iOS applications. Objective-C and Swift are most commonly used to write iOS programs since the two fundamental simple language are both object-oriented which can support functionalities packaged in Apple’s Cocoa frameworks [2]. Therefore, the choice of programming languages for iOS application development is limited.

Thirdly, basic map features of the application is restricted by the application programming interface (API). As a tool simplifying programming by abstracting internal implementation details, a map application programming interface has been chosen to provide fundamental map functions during the iOS development [3]. Hence, as a map-based application, it has great dependency of the map API.

Finally, with regard to hardware, the GPS navigation system and network connection are both constraints for the application. Since the application needs GPS to locate and track the users, it is necessary for mobile devices to configure a GPS module. Moreover, the application sends notification to end users through the Internet. Accordingly, it is crucial to ensure the network connection for the application.

Description of anticipated documentation

A. Software Requirements Specifications Document (SRS):
SRS document lays out functional and non-functional requirements in detail. A collection of use cases is listed to display interaction between external and internal of the system as well. Additionally, there is a description of project purpose, hardware and software constraints, and a general project plan.

B. Software Design Document (SDD):
SDD aims to describe a concrete design flow including designing interfaces, the application of data structures, algorithms and APIs.

C. Test Documentation:
The document includes the test plan, testing strategies and testing results.

D. User manual:
User manual is a technical communication document offering assistance to end users who are not familiar with the application. The document will provide instructions for each function and further help.

E. Final project report:
The project report will provide an overall description of the project. Furthermore, the strengths and drawbacks will be evaluated and future improvement will be discussed in the end.

Description of anticipated experiments

A. Accuracy

Invite a group of volunteers (at least 30 persons) to try out our app in Liverpool. Require them to record if the app is accurate (notify them to get off bus at least 3 minutes)

Simulate the situations using the debug function provided by the map software. For example, the app can be tested using the Xcode debug function, which places a virtual moving point on the map. With correct speed and location, this can simulate real-life situations easily.

B. Robustness

Test the application for 500 times. Record the average down time and calculate the average. If the average frequency is less or equal to 1%, then the requirement is met.

C. Data integrity

Inspect the code. Check if the location data is private

Apply the common security techniques to attack the app. If the data is not accessible or readable by the external application, this requirement is satisfied

Look up the local location privacy terms and compare with the disclaimer included in the app. Check if there is any violation.

D. Speed

Place the phone in an WIFI environment that the speed is strictly set to be 1Mb/s. Test the application in the environment for 10 times consecutively and check if there is any noticeable delay (delay that can be sensed by human eyes).

E. Cellular data Requirement

Run the app for 10 time consecutively. Record the cellular data usage and compare it with the GoogleMap or Apple Map. If the cost is strictly less than 1.5 times the usage of Google Map or Apple Map, then this requirement is met.

F. Usability

Find a group of volunteers who have no technical background and relatively little experience using the mobile application. If they managed to operate the app with no reported difficult after reading the in-app instruction and use it for two times, this condition is met.

Description of evaluation methods

The app contains all the functionalities written in the requirement documents, including notifying user when approaching the destination, notifying the user about the nearby landmarks and so on.

Gather a group of volunteers (at least 50 persons) and asks them to try out our app in Liverpool for a week and comments on the usefulness (whether our app does limit the possibility of getting off in the wrong station). Also, conduct interviews with those volunteers who are willing to be interviewed. Record their feedback and check how many positive feedbacks are there.

Compare with existing similar products such as cityMapper, with respect to the accuracy of notification, the usage of cellular data and the usability of the app.

Conduct of the Project and Plan

Background research

This application is designed for reminding user to get off the bus in proper time and to provide more information about the local landmarks. To make sure our application is needed, we have conducted a survey about bus experience and a comparison of the existing bus related applications.

Based on the data from our survey and questionnaire, some research has been done for analyzing and solving the problem.

Questionnaire result of bus experience:

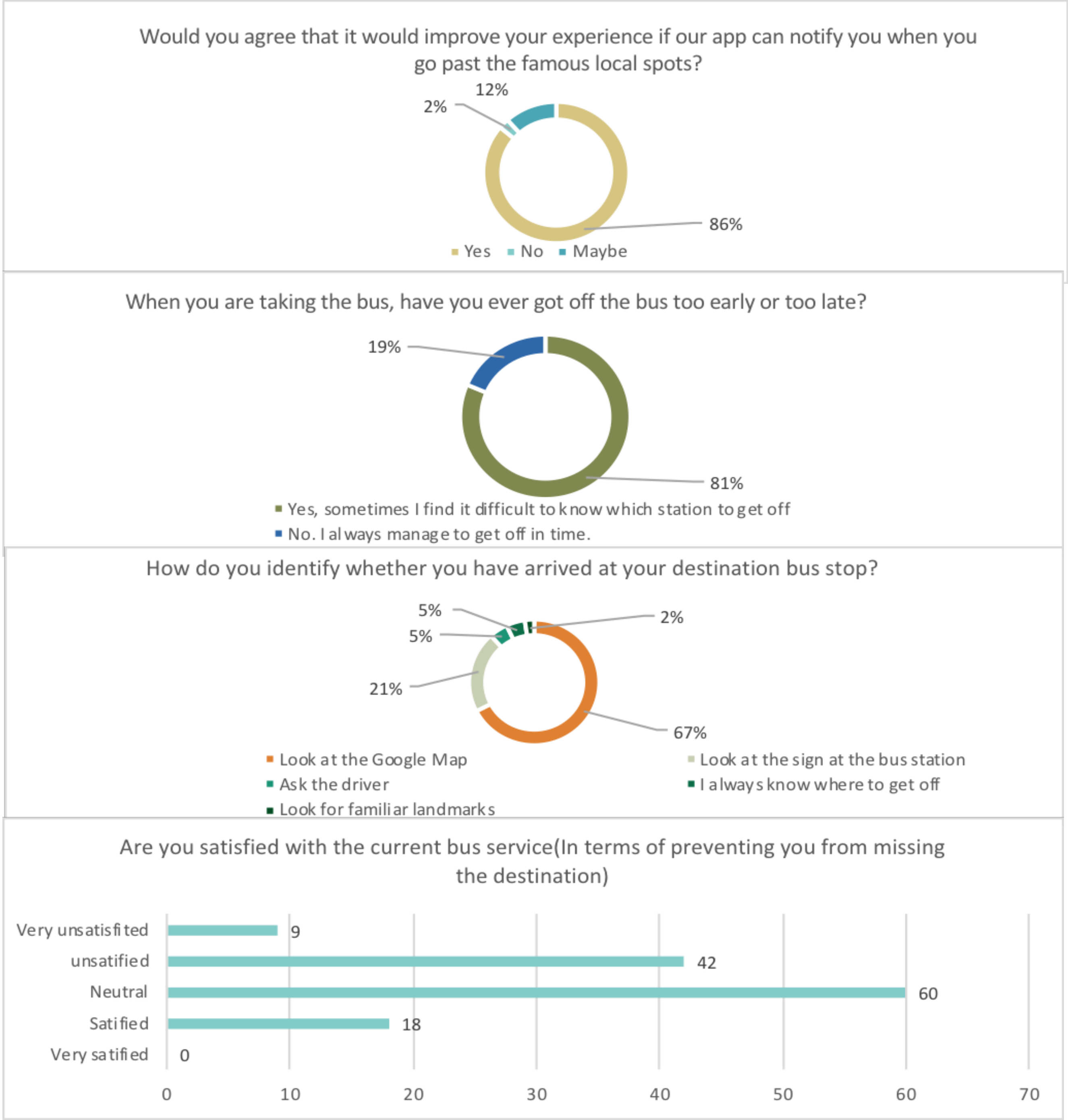


Figure1. The Result of the Questionnaire

According to the Figure 1, over 80% people have the experience of missing the bus stop and most people solve this problem by using the map application. More than 85% of people think that the current bus service in Liverpool cannot satisfy their demand.

Based on this phenomenon, one mobile application to alert people getting off the bus is useful for people's daily life. In addition to avoid missing bus stop, helping user to have better experience, sending landmark notification can also be added to the mobile application

Survey about analogous bus applications (based on IOS System):

App Name	Function1	Function2	Function3	Function4	Rates
First Bus	Nearby stops	Journey Option	Journey Details	Live Bus times	4.5
Bus Times London	Arrival Alert	Route Planner	Bus Stop Finder	Live Bus times	4.6
London Bus Checker	Nearby stops	Live bus times at every stop	Smart journey planning	Explore your route	4.5
London Live Bus Countdown	Live bus times	Journey Option	Journey Details	Traffic cameras, plan journey…	4.6
UK Bus checker	Live bus times	Journey Options	Explore your route	Live bus times as notification	4.4
Arriva UK Bus App	Live bus times	Journey details	Journey Options		3.4
Here WeGo	Journey Options	Offline maps	Journey Details	Turn-by-turn drive navigation	4.1
ETA-Arrive on time	View travel time to all your favourite places	Traffic condition considered plan	Live bus times	Share arrival with interactive message.	3.9
Moovit: Public Transport info	Choose best route	All nearby in real time	Estimate arrival time and alerts delays	Arrival alert	4.5
Liverpool Bus & Train Times	Live bus times	Journey Details	Real-time directions	Arrival alert	4+
EdinBus	Stop location	Track your way	Live bus times		3.5
Next Bus Times for London	Live bus times	Manage bus stop	Search bus stop		4.4
Transit Live transport Times	Stalk bus time	Customized Bus selection			4.1
Stagecoach Bus	Live bus times	Information of stops passing	Save journey route for quick access		1.9

Table 1: Comparison between map application.

According to the table 1, it can be drawn that most applications have common functions. However, only three of them have the function of sending arrival notification. At the same time, those applications who has the function of sending arrival notification always receive higher score. Also, none of those applications provide the function of sending landmark notification.

In conclusion, just few applications provide the function of alerting people to get off the bus. Thus, creating a new mobile application with alarming function is of great needs for people who live in UK.

Technical Research:

To make the implementation stage more easily, we have done research on the tool, application services and application frameworks that we will need to use.

A. For development tools and language:
To develop our application in iOS platform, we need to use IDE – Xcode to develop [4]. The language applied to develop is Swift [5].

B. For the application services and framework:
In this application, we need to track and display user locations and send notifications to them. Also, designing a user interface is necessary. In this case, we need to become familiar with Core Location framework, User Notification framework, MapKit and UIKit.

a) Core Location framework support developer to obtain the geographic location and orientation of a device. [6]

b) MapKit will provide support to display embedded maps into application views and could draw points of user destination on the map. The text completion for user typing in the name of the point of interest is also provided. [7]

c) User Notifications supports the delivery and handling of local and remote notifications. [8]

d) The accurate of location is important for our application, so we need to have the best accurate, by using "kCLLocationAccuracyBest". [9]

C. Data Required:
From users' view, the application requires their GPS location data, network information

and destination information.

From developer’s view, user’s locations in the form of two-dimension map information is needed. Data about the location and the name of bus stations is acquired as well. In addition to that, developers also need to specify a range of value which indicates the distance between destination and alert position.

User’s information will be collected from their mobile phone by calling GPS interface and storing user’s input.

The basic map and bus information will be obtained from MapKit API. A range of value will be set by programmer by default. User could also change it in setting function.

Design methods and documentation

Design Method:

Top-down design method
Use-case diagram
Activity diagram
Pseudo-code for the key methods
Interface design
Evaluation design

Design documentation:

A. Introduction
   a) Goals and Objectives
   b) Intended Audience
B. Project Overview and Scope
   a) Core features
   b) Major constraints
   c) Technologies used
C. System Design
   a) Use Case Diagram
   b) Activity Diagram
   c) Sequence Diagram
D. Function Design
   a) User’s Track

b) Detect destination

c) Landmark Notification

d) Search bus stops

E. User Interface Design

a) Main view Design

b) Setting view Design

c) Pop-up Design

F. Restrictions, limitations, and constraints

G. Testing Issues

a. White Box Testing

b. Black Box Testing

c. Feature Testing

d. Testing cases

H. Future improvements

Implementation and testing

A. Hardware and software will be used:

Hardware: iPhone, laptop

Software: Xcode

B. What testing will be carried out:

a) Accuracy of sending notification in time;

b) Accuracy of user's location

c) Robustness

d) Data integrity

e) Data usability

Project plan

Plan description

A. 7th February – 14th February

a) Project Description (Joshua Brown)

b) Statement of Deliverables (Yiting Wang, Qiyao Zuo, Li Ju)

c) Conduct of Project and Plan (Yaxi Lei, Weiyi Zhang)

d) Bibliography (Yaxi Lei, Weiyi Zhang)

B. 15th February – 16th February

Slack(For accidents).

C. 17th February – 20th February

Review meeting preparation (all members).

D. 21st February – 26th February

a) Main View and Setting View design(Joshua Brown)
b) Pop-up notification design(Joshua Brown)
c) Customer track function design(Li Ju, Qiyao Zuo)
d) Landmark Notification function design(Qiyao Zuo, Li Ju)
e) Detect arrival(Yiting Wang, Yaxi Lei, Weiyi Zhang)
f) Searching destination(auto-complete) function design(Yaxi Lei, Weiyi Zhang, Yiting Wang)
g) Arrival notification function design(Weiyi Zhang, Yiting Wang, Yaxi Lei)

E. 27th February – 2nd March

Writing Documentation (all members).

F. 2nd March – 5th March

Slack (for accidents).

G. 6th March – 27th March

a) Main View and Setting View implementation(Joshua Brown)
b) Pop-up notification implementation(Joshua Brown).
c) Detect arrival implementation (Yiting Wang, Weiyi Zhang)
d) Searching(auto-complete) implementation (Yaxi Lei, Yiting Wang)
e) Landmark notification (Li Ju, Qiyaozuo)

H. 28th March – 3rd April

Integrate and Test (Weiyi Zhang, Yaxi Lei)

I. 4th April – 11th April

Documentation writing (all members).

J. 11th April – 20th April

Slack (for accidents).

K. 20th April – 3th May

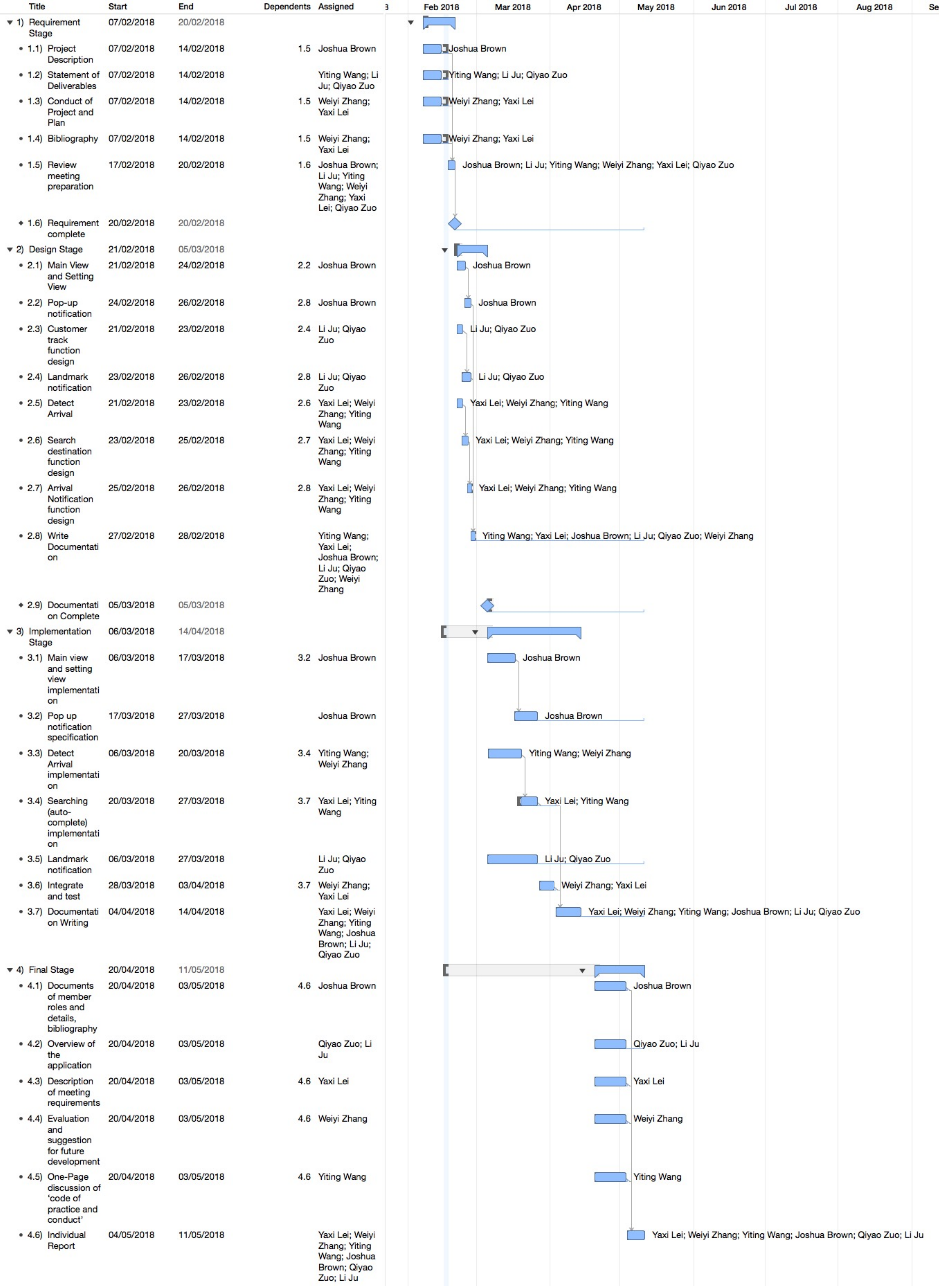
a) Documents of Member roles and Details; Bibliography (Joshua Brown)
b) Overview of the application (Qiyao Zuo, Li Ju)
c) Description of met requirements (Yaxi Lei)
d) Evaluation and suggestion for future development (Weiyi Zhang)
e) One-page discussion of ‘codes of practice and conduct’ (Yiting Wang)

L. 4th May- 11th May

Individual Report

COMP208 Group Project: Gantt Chart

Group 21



Title	Start	End	Dependents	Assigned
▼ 1) Requirement Stage	07/02/2018	20/02/2018		
• 1.1) Project Description	07/02/2018	14/02/2018	1.5	Joshua Brown
• 1.2) Statement of Deliverables	07/02/2018	14/02/2018		Yiting Wang; Li Ju; Qiyao Zuo
• 1.3) Conduct of Project and Plan	07/02/2018	14/02/2018	1.5	Weiyi Zhang; Yaxi Lei
• 1.4) Bibliography	07/02/2018	14/02/2018	1.5	Weiyi Zhang; Yaxi Lei
• 1.5) Review meeting preparation	17/02/2018	20/02/2018	1.6	Joshua Brown; Li Ju; Yiting Wang; Weiyi Zhang; Yaxi Lei; Qiyao Zuo
◆ 1.6) Requirement complete	20/02/2018	20/02/2018		
▼ 2) Design Stage	21/02/2018	05/03/2018		
• 2.1) Main View and Setting View	21/02/2018	24/02/2018	2.2	Joshua Brown
• 2.2) Pop-up notification	24/02/2018	26/02/2018	2.8	Joshua Brown
• 2.3) Customer track function design	21/02/2018	23/02/2018	2.4	Li Ju; Qiyao Zuo
• 2.4) Landmark notification	23/02/2018	26/02/2018	2.8	Li Ju; Qiyao Zuo
• 2.5) Detect Arrival	21/02/2018	23/02/2018	2.6	Yaxi Lei; Weiyi Zhang; Yiting Wang
• 2.6) Search destination function design	23/02/2018	25/02/2018	2.7	Yaxi Lei; Weiyi Zhang; Yiting Wang
• 2.7) Arrival Notification function design	25/02/2018	26/02/2018	2.8	Yaxi Lei; Weiyi Zhang; Yiting Wang
• 2.8) Write Documentation	27/02/2018	28/02/2018		Yiting Wang; Yaxi Lei; Joshua Brown; Li Ju; Qiyao Zuo; Weiyi Zhang
◆ 2.9) Documentation Complete	05/03/2018	05/03/2018		
▼ 3) Implementation Stage	06/03/2018	14/04/2018		
• 3.1) Main view and setting view implementation	06/03/2018	17/03/2018	3.2	Joshua Brown
• 3.2) Pop up notification specification	17/03/2018	27/03/2018		Joshua Brown
• 3.3) Detect Arrival implementation	06/03/2018	20/03/2018	3.4	Yiting Wang; Weiyi Zhang
• 3.4) Searching (auto-complete) implementation	20/03/2018	27/03/2018	3.7	Yaxi Lei; Yiting Wang
• 3.5) Landmark notification	06/03/2018	27/03/2018		Li Ju; Qiyao Zuo
• 3.6) Integrate and test	28/03/2018	03/04/2018	3.7	Weiyi Zhang; Yaxi Lei
• 3.7) Documentation Writing	04/04/2018	14/04/2018		Yaxi Lei; Weiyi Zhang; Yiting Wang; Joshua Brown; Li Ju; Qiyao Zuo
▼ 4) Final Stage	20/04/2018	11/05/2018		
• 4.1) Documents of member roles and details, bibliography	20/04/2018	03/05/2018	4.6	Joshua Brown
• 4.2) Overview of the application	20/04/2018	03/05/2018		Qiyao Zuo; Li Ju
• 4.3) Description of meeting requirements	20/04/2018	03/05/2018	4.6	Yaxi Lei
• 4.4) Evaluation and suggestion for future development	20/04/2018	03/05/2018	4.6	Weiyi Zhang
• 4.5) One-Page discussion of 'code of practice and conduct'	20/04/2018	03/05/2018	4.6	Yiting Wang
• 4.6) Individual Report	04/05/2018	11/05/2018		Yaxi Lei; Weiyi Zhang; Yiting Wang; Joshua Brown; Qiyao Zuo; Li Ju

Risk assessment

Risk type

Risk Type	Possible risks
Tools	IDE/SDK cannot provide demanded functionality. The code generated by CASE tools is inefficient.
Team work challenges	Consensus cannot be made amongst the group. Bad communication lead to lower efficiency. It is impossible to recruit staff with the skills required. Key staff are ill and unavailable at critical times. Required training for staff is not available.
Estimation	The ability of staff to learn and implement new technology is underestimated. The scalability of software is underestimated. The time required to develop the software is underestimated. The rate of defect repair is underestimated. The size of the software is underestimated.
Technology challenges	The major challenges may require more time to be solved. Several challenges may beyond stuff's capability (lack of the knowledge about swift, CoreLocation framework and MapKit API), thus not all of the challenges can be overcame. Software components which should be reused contain defects which limit their functionality.
Requirements	Change to requirements which require major design rework are proposed.

Risk factors

Risk Type	Potential indicators
Tools	Complaints about IDE/SDK tools or weak functionality.
Team work challenges	Poor staff morale, poor relationship among team member, job availability.
Estimation	Many requirements change requests, customer complaints, failure to meet agreed schedule, failure to clear reported defects, failure to accomplish certain function.
Technology Challenge	Complaints about the complicated implementation. Late delivery of support software, many reported technology problems.

Risk analysis

Risk	Probability	Effects
It is impossible to recruit staff with the skills	High	Catastrophic
Several challenges may beyond stuff's capability(lack of the knowledge about swift, CoreLocation framework and MapKit API), thus not all of the challenges can be overcame.	Low	Catastrophic
Bad communication lead to lower efficiency.	Moderate	Serious
The ability of staff to learn and implement new technology is underestimated.	High	Serious
Software components which should be reused contain defects which limit their functionality	Moderate	Serious
Changes to requirements which require major design rework are proposed.	Moderate	Serious
IDE/SDK cannot provide demanded functionality.	Low	Tolerable
Consensus cannot be made amongst the group.	High	Tolerable
The major challenges may require more time to be solved.	High	Tolerable
Required training for staff is not available	Moderate	Tolerable
The rate of defect repair is underestimated	Moderate	Tolerable
The size of the software is underestimated	High	Tolerable
The scalability of software is underestimated.	Low	Insignificant
The code generated by CASE tools is inefficient	Moderate	Insignificant

Risk Handling

Stage		Priority (0-4)
Requirement Stage		
	Project Description	0
	Statement of Deliverables	0
	Conduct of Project and Plan	0
	Bibliography	0
	Review meeting preparation	0
	Requirement complete	0
Design Stage		
	Main View and Setting View	1
	pop-up notification	1
	Customer track function design	1
	Landmark notification	2
	Detect Arrival	2
	Search destination function design	2
	Arrival Notification function design	1
	Write Decumentation	1
	Documentation Complete	0
Implementation Stage		
	Main view and Setting View implementation	2

	Pop up notification specification	2
	Detect Arrival implementation	3
	Searching(auto-compelete) implementation	3
	landmark notification	4
	Integrate and test	0
	Documentation Writing	0
Final Stage		
	Documents of member roles and details, bibliography	0
	Overview of application	0
	Description of meeting requirements	0
	Evaluation and suggestion for future development	0
	One-Page discussion of 'code of practice and conduct'	0
	Individual Report	0

What if people in the team do not co-operate?

It really depends on the situation. In the situation where the team member do not want to do tasks which have been assigned to, other team members will encourage him or exchange tasks with him.

What if we discover that some requirements require too much background work?

First of all, we can learn the required knowledge and do the project simultaneously. We believe the project should be a single flow but an iterative process. What is more, we can assign different tasks to each team member, which means we can handle the work load more evenly.

What if some tasks seems to require much more than anticipated?

First of all, in our plan, we have left out some time for slack (about 20% more than planned time). Therefore, as long as things do not exceed too much, this risk can be handled. Moreover, if things do exceed more than what we planned (normal time + slack), we can rearrange our time. For example, in the design stage, if we spend too much time designing a specific function, we can then shorten our time for the implementation.

New skills

a) How to develop an application on iOS system.
b) How to use MapKit, core location framework to develop an application based on map.
c) Communication and collaboration skills.
d) Skills for a healthy group climate
e) Trust and self-disclosure
f) Problem analyzing and solving ability
g) Management and organization
h) Learning a new programming language
i) UI design skills and usage of UI Kit
j) How to work with API.
k) Xcode and interface Builder
l) Fluency with Swift
m) Conducting a complete develop circle of a software.’

Bibliography

[1] M. H. Goadrich and M. P. Rogers, "Smart smartphone development: iOS versus Android," in *Proceedings of the 42nd ACM technical symposium on Computer science education*, 2011, pp. 607-612: ACM.

[2] Apple Inc. (2010). *Why Objective-C*[Online]. Available: https://developer.apple.com /library/content/documentation/Cocoa/Conceptual/OOP_ObjC/Articles/ooWhy.html

[3] A. Jorgensen and J. A. Whittaker, "An api testing method," in *Proceedings of the International Conference on Software Testing Analysis & Review (STAREAST 2000)*, 2000.

[4] Apple Inc. (2018)[Online]. Available: https://developer.apple.com/develop/

[5] Apple Inc. (2018). *Swift 4*[Online]. Available: https://developer.apple.com/swift/

[6] Apple Inc. (2017). *Core Location*[Online]. Available: https://developer.apple.com /documentation/corelocation

[7] Apple Inc. (2017). *MapKit*[Online]. Available: https://developer.apple.com /documentation/mapkit

[8] Apple Inc. (2017). *UserNotifications*[Online]. Available: https://developer.apple.com /documentation/usernotifications

[9] Apple Inc. (2017). *CLLocationAccuracy*[Online]. Available: https://developer.apple.com /documentation/corelocation/cllocationaccuracy