

Krishna Raj Bagale

**DEVELOPMENT OF A GEOLOCATION APP IN IOS PLATFORM
AND ITS MARKET ANALYSIS**

**DEVELOPMENT OF A GEOLOCATION APP IN IOS PLATFORM
AND ITS MARKET ANALYSIS**

Krishna Raj Bagale
Bachelor's thesis
Spring 2016
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Information Technology, Engineering Department

Author: Krishna Raj Bagale

Title of the bachelor's thesis: Development of a Geolocation App in Ios Platform and its market analysis.

Supervisor: Kari Laitinen

Term and year of completion: 2016

Number of pages: 43

The main aim of this Bachelor's thesis was to describe the development of a geo location app for the iOS platform and analyze its market value.

The main objective includes severe research and analysis on iOS platform since it was my job to develop an application for iOS. Earlier, an android version of this application had been developed. IOS application has been developed using the newest Apple's language called Swift.

This thesis is the result of a research on a prototype of an application developed for all kind of ages, mostly the young generation, the development on the iOS platform and an analysis of the overall market strategy, which focuses on user tracking and determining how many users are inside in a certain bar or pub for the first time. Later on it can be updated for other services, too. The application can be developed further if the WEi! manager wants to build more features, and the study goes on continuing.

Keywords: iOS, Geolocation, Market Analysis, Git

PREFACE

Based on the findings of this research, the idea was surveyed in different places with over 280 people and it was very amiable to start in Oulu city.

I would like to thank Francisco Coquillat Colombás for his insightful scheme and my tutor teacher Kari Laitinen for being there as a supportive tool throughout my thesis project.

Finally, thanks to all the developer team members for being so patient when the process was slowing due to many reasons including the errors, which occurred in the Swift language for novice users like me.

April 2016, Oulu

Krishna Bagale

CONTENTS

ABSTRACT	3
PREFACE	4
TABLE OF CONTENTS	5
VOCABULARY	6
1 INTRODUCTION	7
1.1 Team Structure	8
2 USED TECHNOLOGIES	9
2.1 Apple iOS	9
2.2 Version Control	11
2.2.1 Git	13
2.2.2 gitHub	15
2.3 Swift Programming Language	16
2.4 SDK(Software Development Kit)	18
2.4.1 Xcode IDE(Integrated Development Enviornment)	18
2.4.2 iPhone Simulator	20
2.5 Segue Between View Controllers	21
3 THE GEOLOCATION APP	24
3.1 How It Works	24
3.2 Implementation	26
4 USER INTERFACE	28
5 NATIVE APPLICATION	29
6 MARKETING STRATEGY	30
6.1 Application Survey	31
6.2 Market Analysis/Customer Analysis	36
6.3 SWOT Analysis	38
5 CONCLUSION	39
REFERENCES	41
APPENDICES	43

TERMS AND ABBREVIATIONS

Android studio	Android's IDE
App	Application
API	Application Programming Interface
GIT	Version Control Software
GPS	Global Positioning System
GUI	Graphical User Interface
iOS	iPhone Operating System
OS	Operating System
PDF	Portable Document Format
UI	User Interface
Xcode	Apple's IDE
IDE	Integrated Development Environment
SDK	Software Development kit
VCS	Version Control System
WEI!	Where Everybody is!(Application's full form)
2G	Second Generation of Mobile Telephone Technology

1 INTRODUCTION

WEi! will be the first available application nowadays that allows you to spot the most popular places and events around the city, it displays useful information such as the male/female ratio or how crowded some place is, so you can decide whether to go or not to a bar, depending on your interests. Also, having a big stable user base, an analytics platform for business owners will be provided so that they have a better understanding of the demographics around their premises so that they can make business changes according to them. Available for all the main different mobile Operating Systems, in which a normal user (i.e. who is going to use the application for the purpose of what the application is meant to display) would be able to see in real time how many people there are in a bar/club/event/establishment/retail unit/beach or any traceable space.

The users would have a possibility to see a map where to select the desired locations, and see the information. They would also have on the display the percentage of people according to the capacity, and the percentage of each gender. Additionally, they would be able to see the events coming, as well as other interesting information for the user. A feature that WEi! is going to integrate as well will be the possibility to add to a wish list their favorite places so that they do not have to find them every time. A professional user would have some more options in addition to normal user capabilities to benefit their business/businesses. They would have a possibility to add events inside their establishment and to sell tickets through it, thus increasing their revenue.

Also they would have a possibility to promote their establishment inside the app by advertisements. Additionally, they would have analytics to see the amount of people and percentage of each gender at any given time interval. The attractive of WEi! is simply a business idea that has been proven to be successful due to competitor analysis. The only competitor is not developing any activity regarding the common business idea with WEi!. In fact, even though the company still exists, it already announced its closing due to privacy and legal issues which WEi! will not have due to different technology and form of operation.

1.1 Team Structure

My job was to develop an application on the iOS platform.

TABLE 1. Showing our WEi! team structure.

Name	Nationality	Role	Tasks
Carlos VonEckarts-berg	Paraguay	Co-Founder	Consultant
Alessandro Esposito	Italian	Co-Founder, assistant	Consultant Marketing assistant
Alberto Guerra	Spain	Chief developer	Main developer. In charge of back-end and correct development as well as best practices
AlvaroMahillo	Spain	Android Developer	UI development for Android version
Francisco-Coquillat	Spain	Co-Founder, Developer Assistant	General developing assistance. Graphic designer Coordinator
CamiliaKoljonen	Finland	Marketing Strategist	In charge of market research and development of marketing plan
Krishna Raj Bagale	Nepal	iOS developer	iOS developer working closely with Android developers

2 USED TECHNOLOGIES

2.1 Apple iOS

iPhone OS is a UNIX-based mobile operating system developed by Apple Inc. It was published on 29 June 2007 with the first iPhone 2G and it is designed to be used with a touch screen. Since its release, iOS has gone on to become a multi-platform operating system that powers many other hardware devices developed by Apple. It is the second most popular operating system after Android. Augmented reality applications have changed the way applications used to be back in days. Browsing the real world like browsing the web allows us to recognise things and get immediate information. Apple revealed iPhone SDK to develop applications for the iPhone operating system. Originally it is Cisco's core operating system. 270,000 iPhones were sold in first two days of its release. [1]

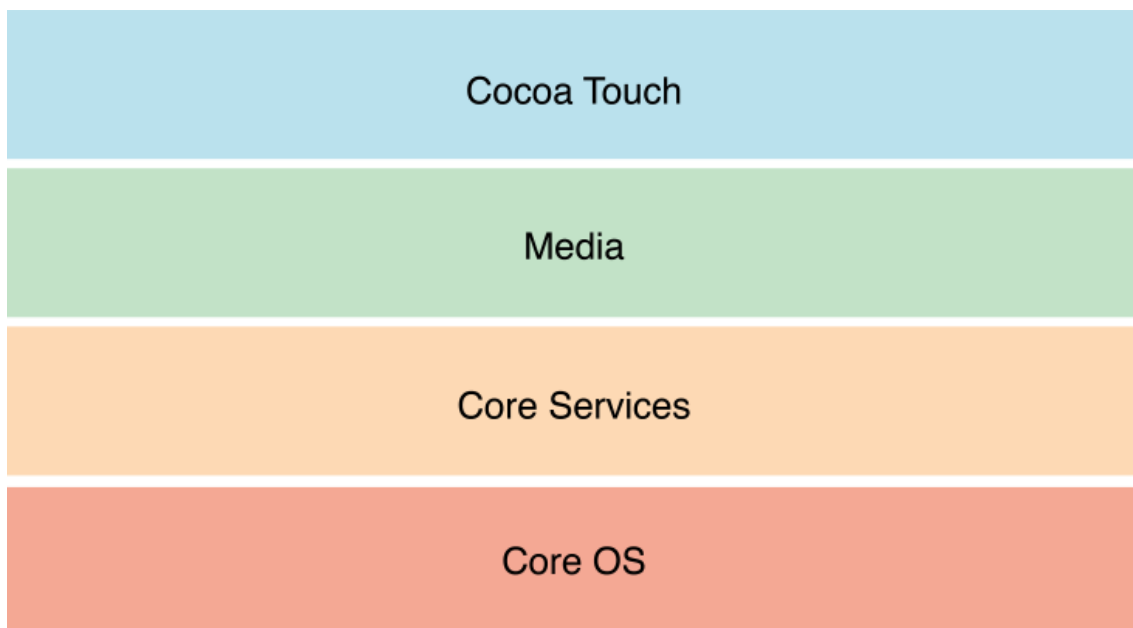


Figure 1. iOS Layers. [1]

iOS OS runs on iPhone, iPod touch, Apple TV and iPad devices. As shown in the figure 1, iOS is built on four different layers. These four layers interact with the hardware. Core Os is closer to the hardware. Core OS is built on the 4. X BSD Unix cornel so it is low level programming. It is associated with networks,

sockets and file systems. It is mostly built with a C language and there is absolutely no object orientation.

Core Services is a common service for both MAC OSX and iOS OS. It contains programming concepts and collections like threads, arrays, and dictionaries. It includes everything that we want to connect to core OS. It is object oriented. It contains the Core Foundation and Foundation frameworks. This layer also contains individual technologies to support features such as location, iCloud, social media, and networking.

Media layer deals with multiple services including talking over the phone, texting, taking pictures, reading PDF files, playing games. This basically means that it contains the graphics, audio, and video technologies. Without the media layer a phone could be a dumb phone. It is associated with a class and methods that can be used easily unlike low level on other layers. iOS provides a built-in support for apps running on either Retina displays or standard-resolution displays. For vector-based drawing, the system frameworks automatically use the extra pixels of a Retina display to improve the crispness of your content.

Cocoa Touch includes buttons, text fields, labels, alerts, navigation controls, multiple tab bars and scrolling which are called a user interface kit framework. It is completely object oriented. There is no need to worry about low-level programming. Cocoa touch allows the use of hardware and features that are not found in Mac OS X computers and are thus unique to the iOS range of devices. Cocoa Touch provides key frameworks like Foundation Kit framework, UIKit Framework, GameKit Framework, iAd Framework, MapKit Framework, Address book UI framework, EventKit UI framework, Message UI Framework, Notification Center Framework, PushKit Framework, and Twitter Framework for developing applications on devices running iOS.

2.2 Version Control

Version control, also known as revision control or source control is a system that keeps logs to a file or set of files and records its changes so that it is possible to recall specific version later. It manages the changes to documents. Each change is known as revision and are associated with a timestamp and the person making the changes. There used to be only one open source version control software called CVS(Concurrent Versions System or Concurrent Versioning System). CVS is a client-server free software revision control system in the software development field. [12]

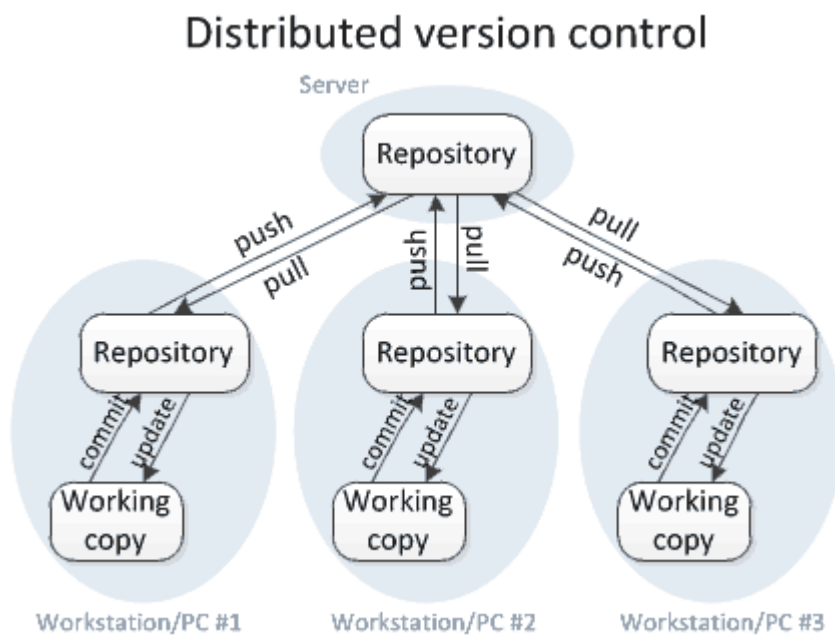


FIGURE 2. Distributed Version Control System. [13]

Centralized version control

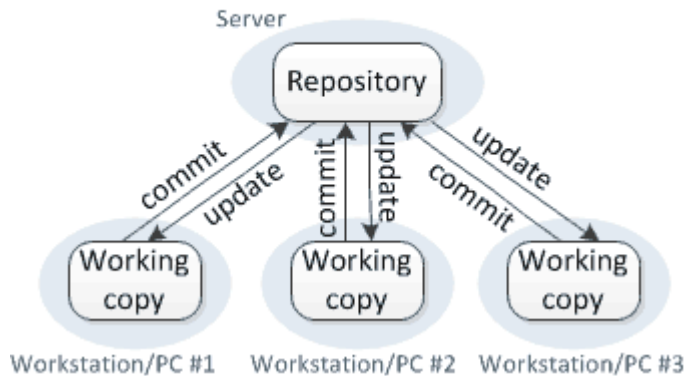


FIGURE 3. Centralized Version Control System. [13]

The requirement of the version control system is very generous nowadays. It organises, controls, tracks, maintains and restores over revisions. For the web designers and developers, it is very wise to use VCS. There are two types of version control structures. They are a distributed version control and centralized version control. The main difference between the distributed and centralized version control, as shown in the figures 2 and 3 is the number of repositories. In distributed version control, there are multiple repositories whereas in the centralized version control, there is only one repository.

In programming a lot of people work together in a single or multiple project. For example, while building an iOS application or any other, we have a bunch of files in a bunch of folder and many people can work on them together, hence the necessary of version control. Having a version control allows you to track what you worked on and what your friends are working on so that the clashing and override will not happen.

2.2.1 Git (Version Control Software)

Git was initially designed and developed in 2005 by Linux Kernel developers for the Linux Kernel development. GIT is a free and open source distributed version

control system. It allows developers to track changes. GIT runs on a command line in a local machine. Since it is distributed, the connectivity does not block work. Its commands are easy to learn and use progressively. GIT is not file-based; unlike other systems it stores data in a different way. It stores information in the form of snapshots. If there is an issue with the latest version of the code and a developer wants to go back to the older version, it has history tracking and it is possible to check out the older version, thus the use of GIT is more essential. [3]

There are two types of data structures, mutable and immutable. GIT makes a good use of the local files stored on a computer. I have used GitHub as VCS throughout my thesis project. Here are some of the commands with the description on how they function.

```
#To create new repository

git init

# Configures the author name and email address to be used with the
commits.

    git config --global user.name "krishna bagale"

    git config --global user.email oceankrish76@gmail.com

#Lists all the branches in the repo, and also tells us what branch we're currently

    git branch
# Lists all the modified files
    git status

# Shows specific differences, helps compose a commit message
    git diff

#To view the conflicts against the base file
    git diff -base <filename>
# Commits just the files I want to
    git commit file1 file2 -m "Descriptive commit message"

#Initializes a git folder
git init myproject

# Creates a working copy of a local repository

git clone username@host:/path/to/repository

#Notifies the files and puts them to get ready to commit.
git add .
# Commits file or files
git commit -m "first commit"
# Creates new remote called origin
```

```
git remote add origin https://github.com/oceankrish76/geolocation.git

# To send changes to the master branch of the remote repository
git push -u origin master

# To fetch and merge changes on the remote server to the working re-
pository
git pull origin master

#To merge a different branch into the active branch
git merge <branchname>
```

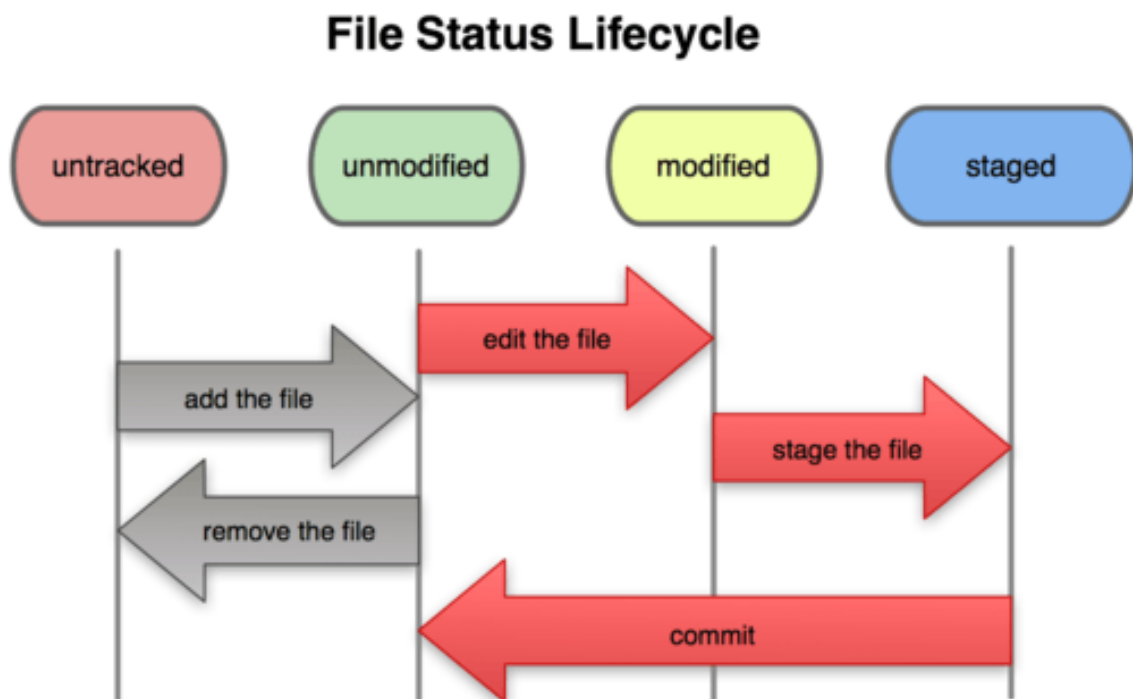


FIGURE 4. Git File Status Lifecycle [14]

Figure number 4 shows the git file status lifecycle. To determine the files as to where they are, git status command can be used.

2.2.2 GitHub

Unlike other VCS, GitHub is a web-based hosting service. It is very user-friendly and handy for projects to keep tracks. GitHub uses the GIT revision control system. It is written using Ruby on Rails. GitHub is a collaboration platform, which allows us to host remote GIT repositories. Github is built on top of GIT and allows us to have both local and remote copies of the project. So in that sense it is very much a cloud-based GIT service. It is a code sharing and publishing service for coders and developers so that they can collaborate on what they are working on. It stores the main file in a place called the "Repository" and gives contributors to the project an ability to download the file as a clone and make changes to the clone. It is the best place to share code with friends, co-workers, classmates, and complete strangers. Millions of people use GitHub to build amazing things together. [4]

2.3 Swift Programming Language

Swift is a brand-new object-oriented programming language developed by Apple Inc. specifically for Mac and iOS app development. It is designed to provide a seamless compatibility with Cocoa and Objective-C. The development of Swift was begun in 2010 by Chris Lattner, with the collaboration of many other programmers. On June 2, 2014, the WWDC app became the first publicly released app written in Swift. Swift is fast, modern, safe and interactive programming language. Swift is faster than Objective-C and Python. It is modern with features like closures, generics, and type inference. Since it uses type inference, it has powerful string processing capabilities. Unlike languages like Objective-C, Swift does not require the creation of separate interface and implementation files for custom classes and structures. In Swift, you define a class or structure in a single file, and the external interface to that class or structure is automatically made available for other code to use. Swift is completely native to Cocoa and Cocoa Touch. Swift is readable like Objective-C and designed to be familiar to Objective-C developers.

It is a case sensitive language. In Swift it is not necessary to write semicolons at the end, but if we are writing multiple statements in a single line, we must write it down. Swift does not require a main function to start with. Swift integrates well with Objective-C programs and reduces the common patterns of Objective-C. It is one of the fastest growing programming languages in history. Swift embraces modules, eliminating the need for headers and the code duplication. It uses Objective-C runtime, allowing Objective-C, Objective-C++ and Swift code to run within a single program so that the apps already made using Objective-C can be edited using Swift.

Swift has no header files. The Swift language is relatively small, because many common types, functions, and operators that appear virtually everywhere in Swift code are actually defined in the Swift standard library. [2] [3]

Closures:

In Swift Closures, there are self-contained blocks of functionality which can be passed around and used in the code. The closures in Swift are similar to blocks in C and Objective-C and to lambdas in other programming languages.

Generics:

In Swift a generic is a function with the type that can be determined using a placeholder value, and the placeholder value is actually determined when the function is called.

let keyword defines a constant which cannot be changed afterwards, whereas var defines an ordinary variable, which can be changed later. Constants are declared with the let keyword and variable with the var keyword. You can declare multiple constants or multiple variables on a single line, separated by commas:

```
var x = 0.0, y = 0.0, z = 0.0
```

2.4 SDK (Software Development Kit)




iOS SDK is a software development kit developed by Apple Inc. and released in February 2008 to develop native applications for iOS. It allows developers to make applications for the iPhone and iPod Touch, as well as test them in an "iPhone simulator". Whenever we choose to develop a native application, it is always a good idea to use iOS SDK so that it is possible to use all the device's capabilities. [8]

2.4.1 Xcode IDE (Integrated Developer Environment)

First released in 2003, Xcode is an integrated development environment developed or created by Apple Inc. It contains software development tools used for developing OS X and iOS software. Essentially, it is available in Mac Store for free. Xcode includes editors, compilers, and other necessary tools. It supports development for Apple's devices for example Mac, iPhone and iPods. It is designed to operate as a single window interface. It is really viable to use GIT while creating projects and uploading to repositories. It allows us to design and develop code and finally compile and submit to the App Store. When we type Xcode download in Google, it will take to the page where there's the latest version of Xcode available to download. It will open up in the Mac Store where there's a button install. Each version differs in interface and many other features. After installing and if we have already few projects built, Xcode opens like in the figure 5. [10]

x



-  **Get started with a playground**
Explore new ideas quickly and easily.
 -  **Create a new Xcode project**
Start building a new iPhone, iPad or Mac application.
 -  **Check out an existing project**
Start working on something from an SCM repository.
- Show this window when Xcode launches

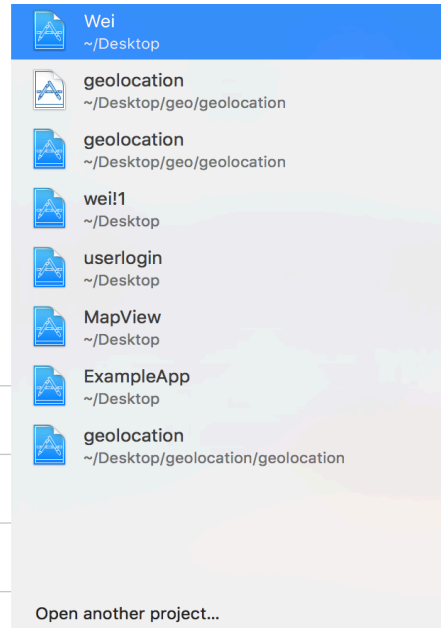


FIGURE 5. Welcome Screen of Xcode.

Xcode is a foremost application that has been used to make iPhone applications down the ages. Xcode also comes with documentation that can be accessed by pressing the Option key and double-clicking on the methods.

2.4.2 iPhone Simulator

iPhone Simulator, originally called as Aspen Simulator, is a simulator for Mac OS X, which can be used to test iOS applications by simulating the iOS technology. iPhone simulator is not an emulator. It is one of the fastest simulators found. But to load the applications in devices is possible only after paying an iOS developer program fee, which is \$99.00 USD per year. Testing is faster because the developer does not need first to upload the application to a mobile device. There are two different ways to access iOS Simulator through Xcode. The first way is to run the app in iOS Simulator, and the second way is to launch iOS Simulator without running an app.



FIGURE 6. Simulator running "Hello World" App.

2.5 Segue Between View Controllers

To share data between two view controllers, a segue can be used. In this application, I have used a show segue. The list of segue are shown in the figure 7. The list of segue appears while dragging and dropping the button from one view controller to another. This segue contains a feature that can help users to go back to the previous navigation. [12]

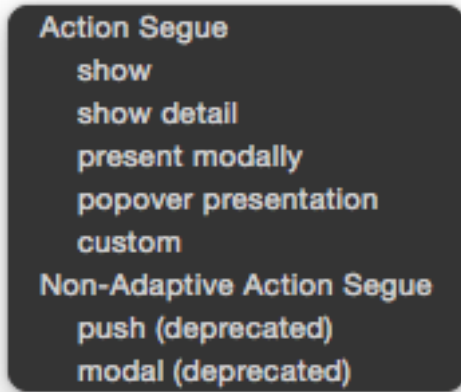


FIGURE 7. List of segue

The starting point of a segue is a button, a table row, or a gesture recognizer that initiates the segue. The end point of a segue is the view controller that you want to display. [12]

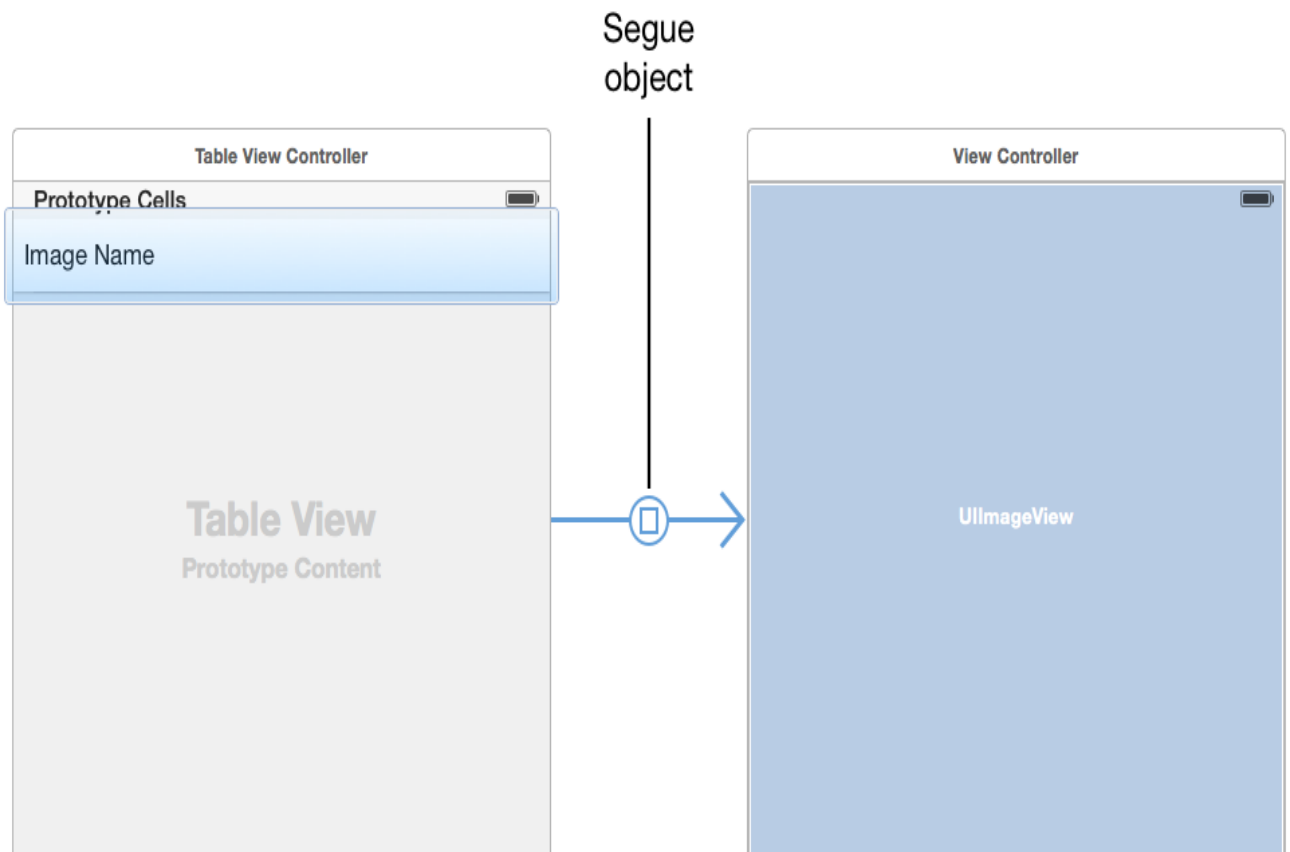


FIGURE 8. A segue between two view controllers. [12]

The view controllers that can be found on the bottom right corner of Xcode can be dragged and dropped to the main screen board and connect each other using segue as shown in the figure 8 and figure 9.

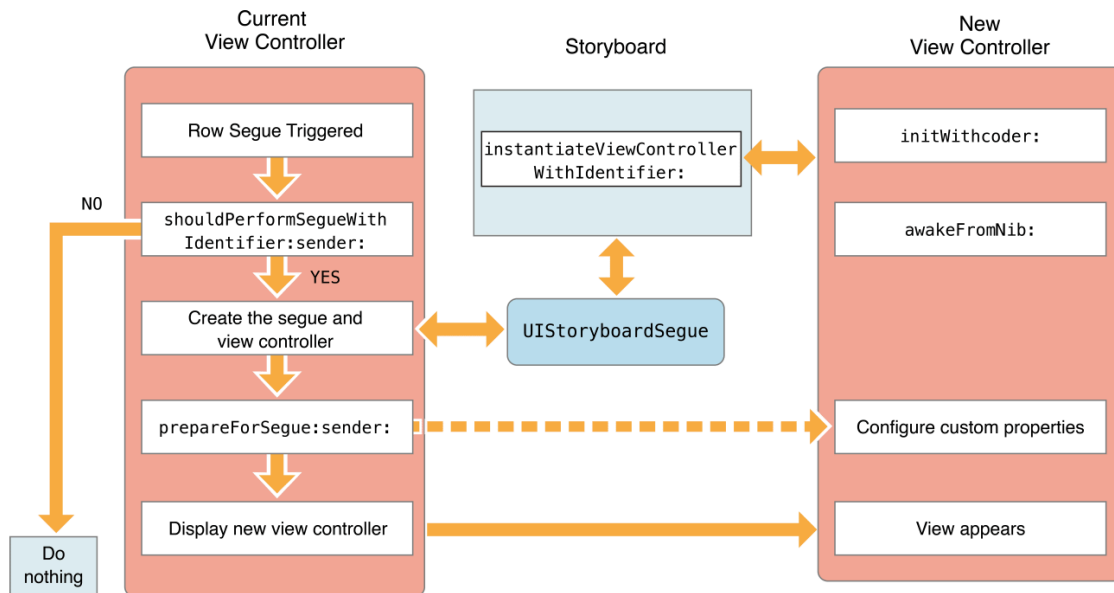







FIGURE 9. Displaying a view controller using a segue. [12]

TABLE 2. OS X Segues [15]

Name	Interface Builder symbol	Description
Show		Present the content in a new window.
Modal		Present the content as a modal dialog.
Popover		Present the content as a popover anchored to an existing view. Specify the behavior (<code>NSPopoverBehavior</code>), preferred edge, and specify the anchor view.
Sheet		Present the content as a sheet attached to the originating window.
Custom		Create your own behaviors by using a custom segue.

The table 2 shows the names of the segues and their interface builder symbol and describes how to use them to the iOS platform.

3 THE GEOLOCATION APP

3.1 How It Works

The app would work in the following way for a normal user:

1. The normal user downloads the app, which would be ideally free of charge.
2. After opening the app for the first time, the user will be asked to select his gender and age
3. A map will be displayed to the user with all the different locations available.
4. When a location is tapped, the information of the amount of people, and the percentage of each gender will be displayed.
5. A button inside the pinned place would take the normal user to the establishment/event/place space inside the app where the normal user would be able to see interesting information about the establishment/event/place, and would have different options/opportunities.

The app would work in the following way for a professional user: (Note: a professional user may have his own app. It depends on technical viability.)

1. The professional user downloads the app, which would ideally be free of charge (to download).
2. The professional user will decide what is interesting for them:
 - a. Using the app as a normal user (this kind of users will not be counted in the statistics of the amount of people, neither the sex statistics, due to irrelevancy)
 - b. Analytics of their establishment, and possibly about other establishment (according to law)
 - c. Possibility to post events
 - d. Possibility to sell tickets through the app

Depending on the options selected, the professional user will be billed a certain amount of money for using WEi! services.

1. After opening the app for the first time, it would ask to fill the company information.
2. After filling the company information
3. Inside the company's profile, several options might be available, depending on the membership level chosen.

The membership level will be easily upgradeable. The users can have a GPS system to navigate to hidden items in various locations, track users and show their current location using Swift, and share their location in real time.

3.2 Implementation

Google Maps Api has been created from the Google developer's website. And that Api is used in AppDelegate.swift.

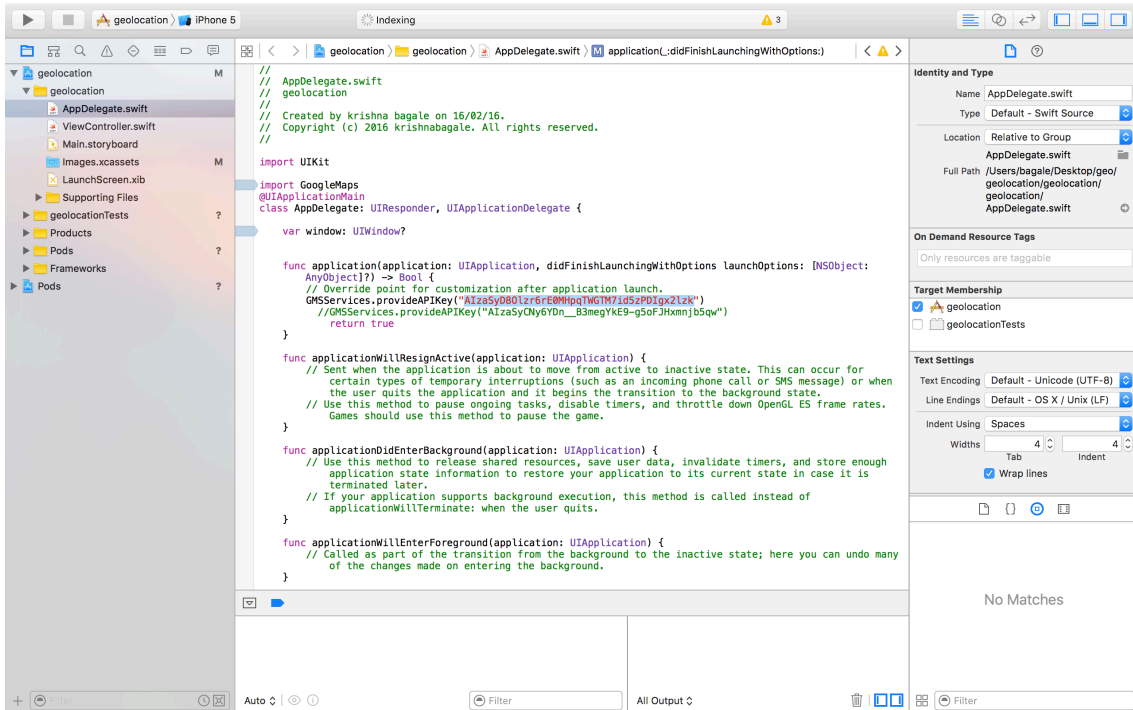


FIGURE 10. Using Google Maps Api in AppDelegate.swift.

And to make it work, Google Maps needs to be imported as import GoogleMaps as shown in the figure 10. It needs to be imported in ViewController.swift as import GoogleMaps to use the latitude and longitude of the location, which is shown in the figure 11.

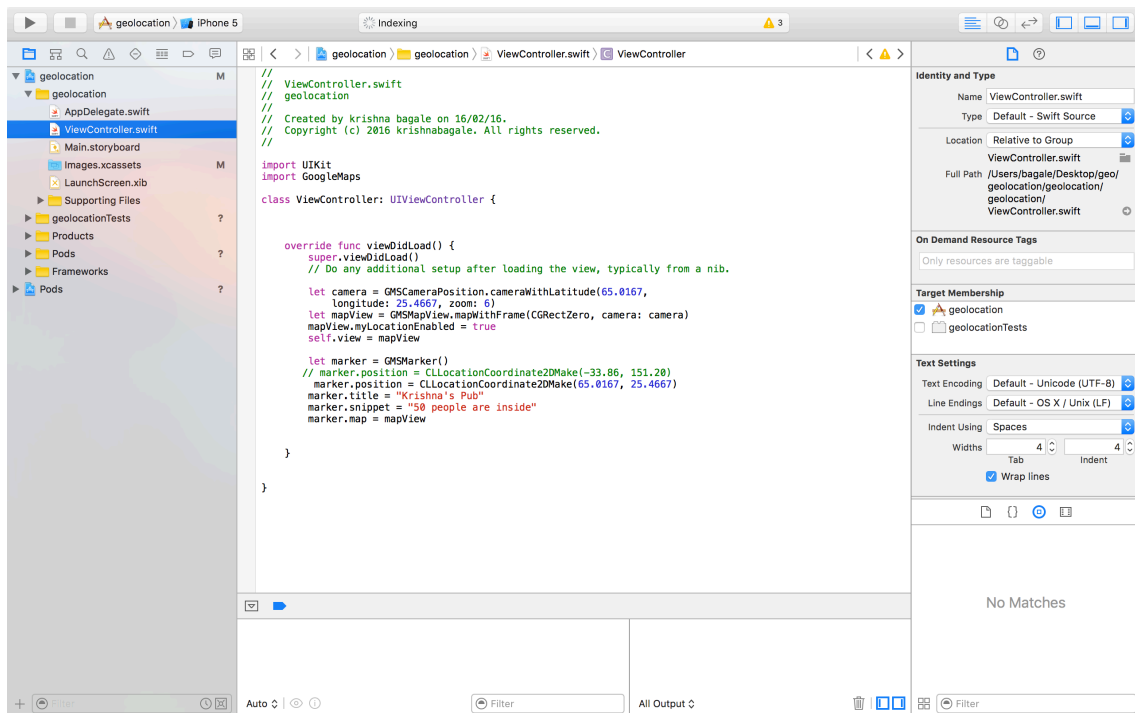


FIGURE 11. Using camera, latitude and longitude of the location.

4 USER INTERFACE

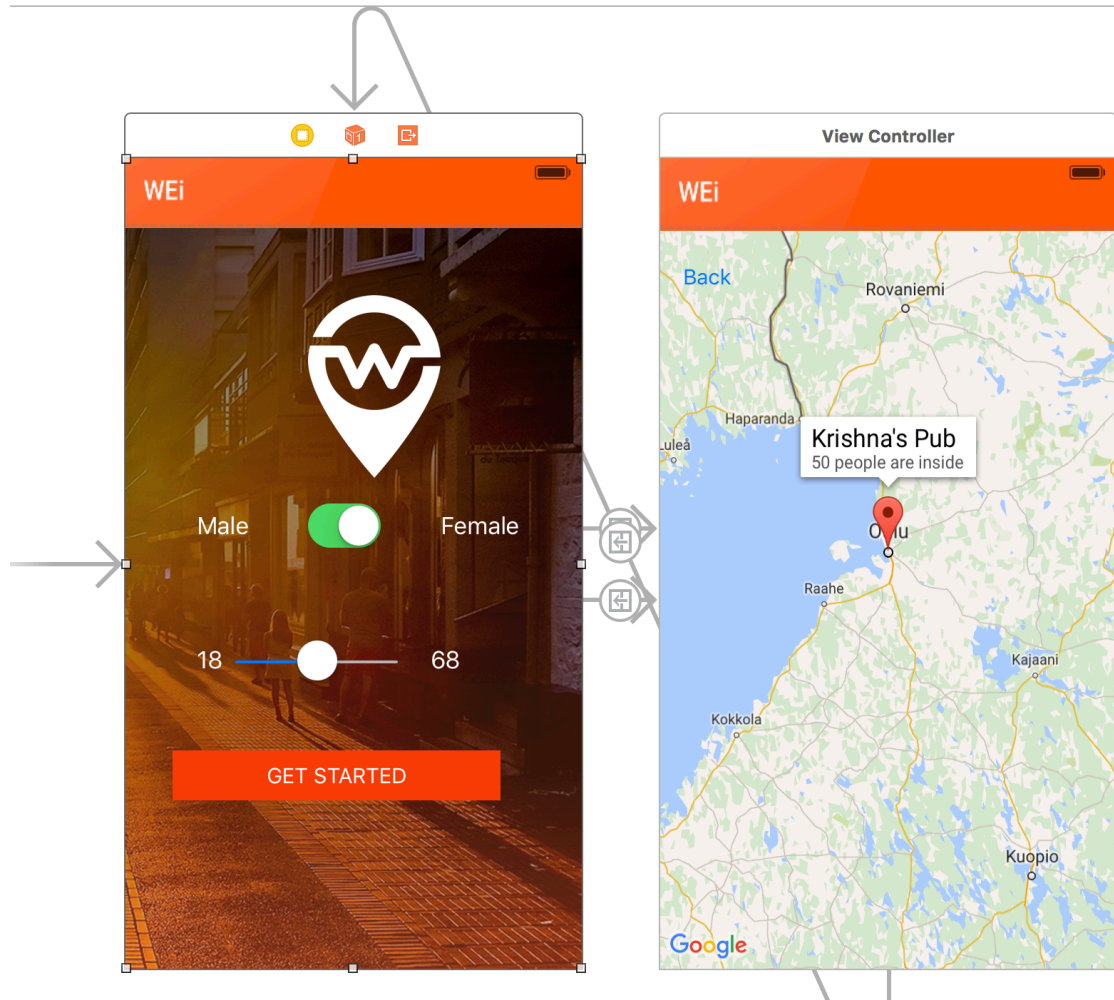


FIGURE 12. Updated Navigation screen 1 and 2.

Once the app opens it asks whether the user is male or female, and what the age is. After the get started button is clicked, it navigates or redirects the user to a new page with map. The app shows the current location of the users and the nearest bars, pubs or restaurants. It also tells the users how many people are inside those places. We will get this information through the server and Google Map Api system.

5 NATIVE APPLICATION

A native application (native app) is an application program that has been developed for use on a particular platform or device. According to Dictionary.com native when referring to computers means: “designed for use with specific type of computer” [11]. It is a smartphone application which is coded in a specific programming language, such as Objective C, Swift for iOS and Java for Android operating systems. Native apps have access to a phone’s various devices, like calendar and mail integration, file managers, etc. It requires low-level code and has fast performance. Usually, in comparison to a web application, its development cost is high. Depending on the platforms, it has different screen sizes so it is very responsive in that sense. And for example in iPhone, there is no back button thus a developer should provide a back button while building an application. Some features like that have to be taken into consideration when it comes to a native application. [11]

Native applications give the full access to device hardware. After the development and while processing to upload on app store, it costs some amount of money. Native applications offer advanced UI interactions and are highly demandable from usability point of view. For each platform iOS or Android or Windows need a separate application. Some advantages of a native mobile application are push notification, full mobile device usage, sell in app store, easy user interface, in-app purchases and ads and promotion through app store, it can also be used offline. Although it takes much development time, it provides a good user experience and efficiency. But from the search engine point of view, it is usually invisible to several search engines. [11]

The application, which I have developed is a native application coded with the Swift programming language. Unlike web apps, native apps are installed physically on a device and are therefore always available for the user, even when the device is on a Airplane mode. [11]

6 MARKETING STRATEGY

As the Roman philosopher Marcus Tullius Cicero aptly said: "Before beginning, plan carefully" [16]. Careful planning is what WEI! Business owner and all the members are doing. This application is a product of a company which has not been established yet, but it will be established in near future.

Here are five different strategies listed below that WEI! team is practicing

1. Target Market
2. How can the company will be benefited
3. Positioning product or service
4. Marketing methods, in this case Internet marketing.
5. Investment and control.

Whenever entering a market, there is a variety of option open. They can vary in cost, risk and the degree of control. The product is totally free of cost in the beginning. We'll test it in Oulu City. We have a technical innovative strategy, a severe security system and a viability free of cost.

One of the crucial factor in building a market entry strategy is time. Brand names do not appear overnight. It takes time, effort and money. A Large investment is also needed with an innovative efforts over a period of time. Physical distance and language barrier could be a problem in the beginning. Later on we'll have different languages in the application. The more the product line and/or the geographic area will be expanded, the greater will be the managerial complexity. Global strategies include country centered local businesses. Thus with different languages depending on countries we can approach the local market.

6.1 Application Survey

The application idea was surveyed with over 300 people via the Internet and in public meetings in different places. Some basic questions are presented here in figure 13.

What is your age?

- 18-20
- 21-25
- 26-30
- 30-35
- 36-40
- More than 40

Gender

- Male
- Women

How many times do you go out a month?

To party, to have a drink with friend...

- 0
- 1
- 2
- 3
- More than 3 times

Do you always go to the same places?

- Yes
- No
- Depends

Do you use any kind of tool that helps you to decide where to go out to eat/drink/have fun?

For example: Tillate, Facebook...

- Yes
- No

In case you answered yes, what tool do you use?

Facebook, Tillate, Instagram...

Your answer

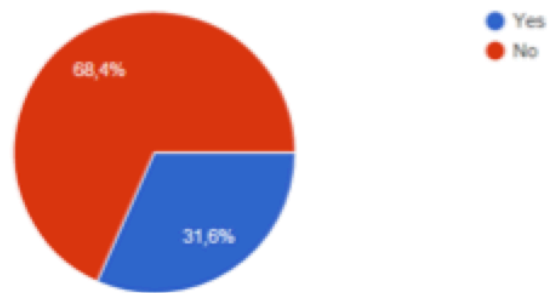
Do you find it useful to know how crowded a bar is before visiting it?

- Not very useful
- Useful
- Very useful

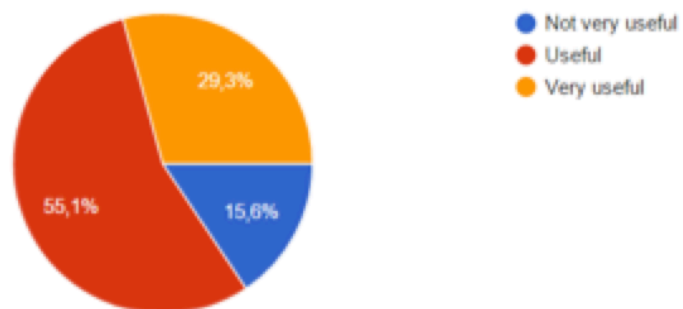
FIGURE 13. Showing our application survey questions.

Before starting to design and code, the idea was presented to the public, some questions were asked regarding it as an agile process through the help of the Internet and personal visits as well. The basic result is shown here in this link (<https://docs.google.com/spreadsheets/d/1Dsft431vzY49235X3cJVmFNYQx4DW22EZcyDIHRW5js/edit#gid=878395716>)

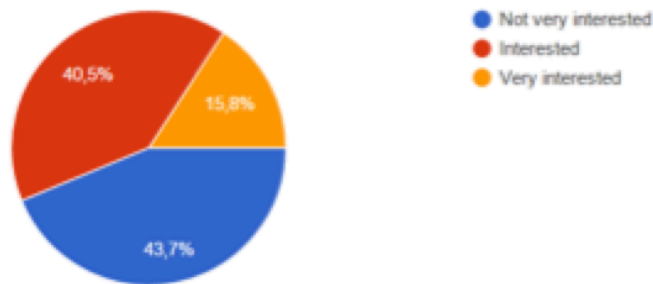
Do you use any kind of tool that helps you to decide where to go out to eat/drink/have fun?



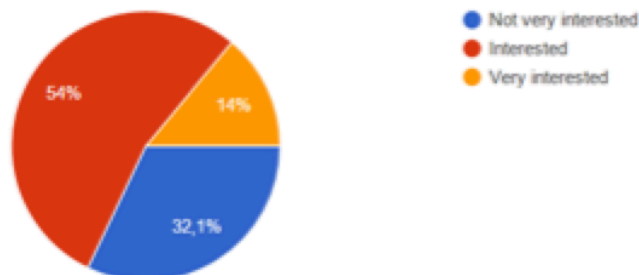
Do you find it useful to know how crowded a bar is before visiting it ?



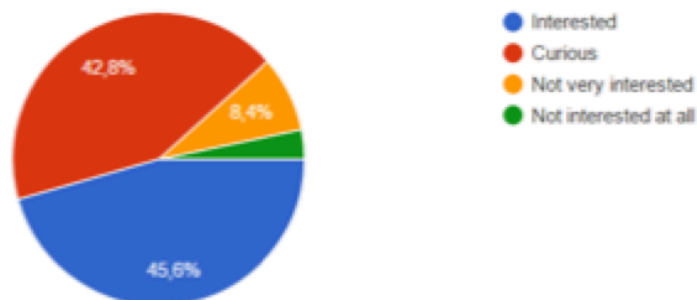
Are you interested in knowing how many man/women there are in a bar before visiting?



Are you interested in knowing if the people in the bar are younger/older than you?



Would you like to see special offers and events of you favorite club/bars, all in the same app?



The result of this survey reflects the willingness of using the type of app that we want to make, and shows high chances of being successful.

6.2 Market Analysis/Customer Analysis

App customers involve people from all age and gender groups. The number of smartphone users using mobile apps has been increasing so that as shown in figure 14, in 2015 there are 60% more users than in 2012.

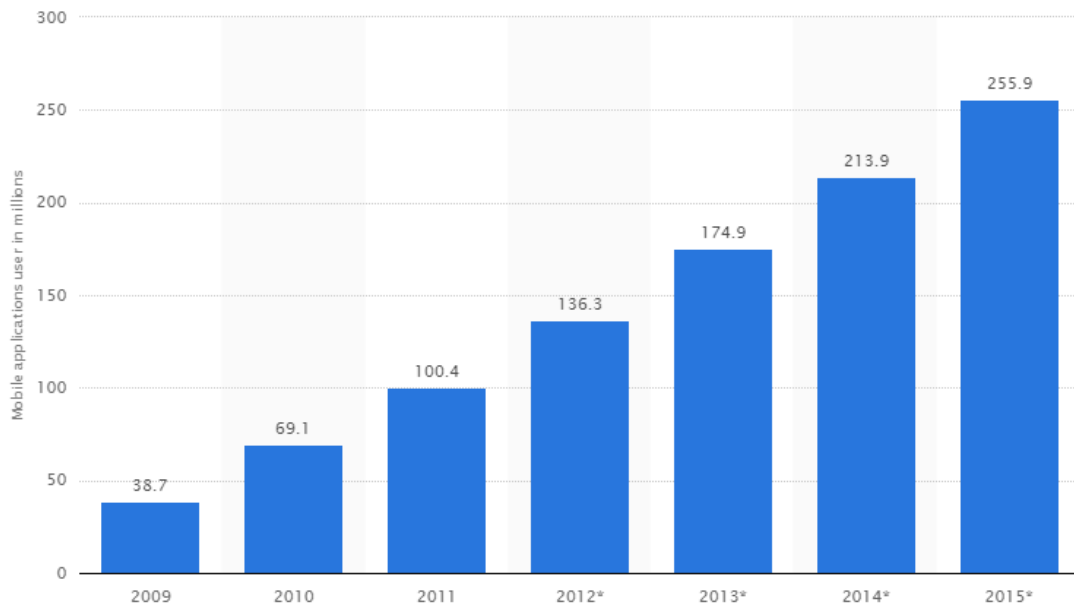


FIGURE 14. Global mobile applications user base from 2010-2015. [6]

People from all ages and all sex, genders utilize mobile handheld devices nowadays. The nature of WEi! does not ban any age range to not use the app. On the other hand, the nature of the app and the various researches found, makes us believe that bigger impact will be reached by focusing on an age range between 16 and 35.

In figure 15 we can see the usage of a location-based dating app, demonstrating ages between 16 and 35 are the main users for dating app.

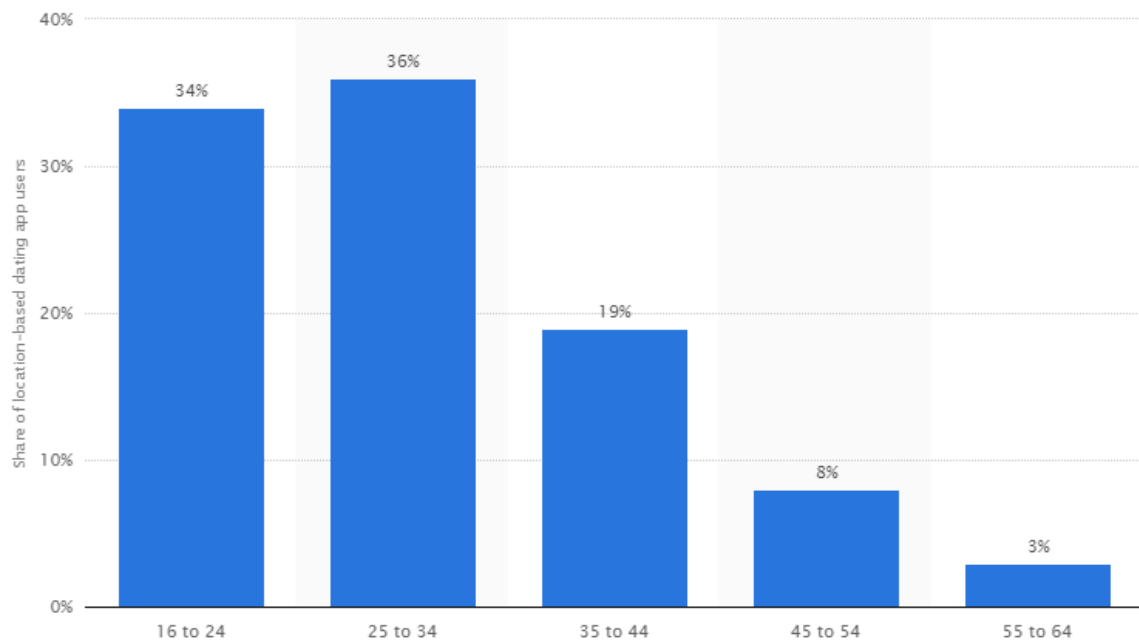


FIGURE 15. Location-based dating app usage among internet users worldwide as of 4th quarter 2014. [17]

This chart was chosen because dating apps are location based, they partially utilize the same system as WEi! and share some of the business ideas, which would be fetching information from people without the need of displacing anywhere. This involves at the end digital socialization.

6.3 SWOT Analysis

The structured planning method that evaluates the four elements of a project or a business venture as strengths, weaknesses, opportunities and threats is shown in the figure 16.



FIGURE 16: SWOT analysis of the application.

7 CONCLUSION

Implementing the WEi! Application in the iOS environment was an utterly ineffable experience and gave me a lot of new information on how to use the technology to make an application. The project was carried out according to the plan. The general aim to develop an iOS application was successfully met. The application provides the same functionality as the Android version, which was confirmed by testing. A little difference can be found in the user interface part. The switch button and a slider for age differ from Android in the colour combination. However, it works as the same way as in the Android version. I was flexible to make my own user interface, which can have a separate business logic from the UI of the Android version.

The front-end of the application was developed with the GUI in Xcode, which gave a very smooth and user-friendly look. The performance of this newly built application is high. While testing with the simulator, it works completely fine. This app was developed on top of the single view application. Later on, different view controller pages and navigation system were implemented. Since the focus of this work was the development of a secure application, using the swift programming language provided extra security. Swift took a lot of features from other languages, like the data structure declarations from C# and Java. It was easy to build or design interfaces because Cocoa and Cocoa Touch are built using the Model-View-Controller pattern. Swift is a major improvement from Objective-C. Swift is a modern safe programming language with modern features to make programming easier, more flexible, and more fun.

The Swift compiler enforces more rules on how to write a program so that it allows us not to make more mistakes nowadays than previously were made by a lot of developers. I found the Swift language to be a very friendly language for new programmers. There are a lot of constructs in the language that will take some work to understand and utilize. At the same time, it is a very approachable language, there is a playground in Xcode, which helps to understand the language in deep.

Swift is an industrial language. Many things were not present in Objective-C. Some of the things in Swift are borrowed from Ruby on Rails, Python, Java, and Scala. There are also features of Swift that are present exclusively to help with the interoperation between Swift and Objective-C. The compiler catches every possible error therefore it is easy for a developer to see where they did the mistake. Although it catches the errors or bugs, it can be hard to find out the solution. Swift helps programmers to catch the errors before runtime. While building WEi!, I was very grateful with the safety of the Swift programming language.

Being an interdisciplinary, the newest field of this technology intrigued me, fascinated me. I am glad that this project helped me to incur my knowledge to the next level since the let go.

REFERENCES

1. Ios Layers. Date of retrieval 17.02.2016

<https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>

2. Swift Programming Language. Date of retrieval 15.02.2016

[https://en.wikipedia.org/wiki/Swift_\(programming_language\)](https://en.wikipedia.org/wiki/Swift_(programming_language))

3. Swift Programming Language. Date of retrieval 22.02.2016

<https://developer.apple.com/swift/>

4. Git. Date of retrieval 19.02.2016

<https://git-scm.com/>

5. GitHub. Date of retrieval 19.02.2016

<https://github.com/>

6. Global Mobile Application Statistics. Date of retrieval 25.02.2016

<http://www.statista.com/statistics/219959/global-mobile-applications-user-base-forecast/>

7. Agile Software Development. Date of retrieval 27.02.2016

https://en.wikipedia.org/wiki/Agile_software_development

8. iOS SDK. Date of retrieval 15.03.2016

https://en.wikipedia.org/wiki/IOS_SDK

9. Apple iOS. Date of retrieval 18.03.2016

<https://en.wikipedia.org/wiki/IOS>

10. Xcode. Date of retrieval 15.02.2016

<https://developer.apple.com/xcode/>

11. Native Application. Date of retrieval 25.03.2016

<http://searchsoftwarequality.techtarget.com/definition/native-application-native-app>

12 Segue between view controllers. Date of retrieval 02.04.2016

<https://developer.apple.com/library/ios/featuredarticles/ViewControllerPGforiPhoneOS/UsingSegues.html>

13 Version Control. Date of retrieval 05.04.2016

<https://homes.cs.washington.edu/~mernst/advice/version-control.html>

14 Git File Status Lifecycle. Date of retrieval

05.04.2016 <https://www.launchacademy.com/codecabulary/learn-git/git-lifecycle>

15 OS X Segues. Date of retrieval 25.03.2016

https://developer.apple.com/library/ios/recipes/xcode_help-IB_storyboard/Chapters/StoryboardSegue.html

16 Marketing Strategy. Date of retrieval 15.04.2016

<http://www.inc.com/guides/writing-marketing-plan.html>

17 Location-based dating app usage among internet users worldwide as of 4th quarter 2014. Date of retrieval 25.03.2016

(<http://www.statista.com/statistics/388212/location-based-dating-app-worldwide-age/>).

Initial Design and development

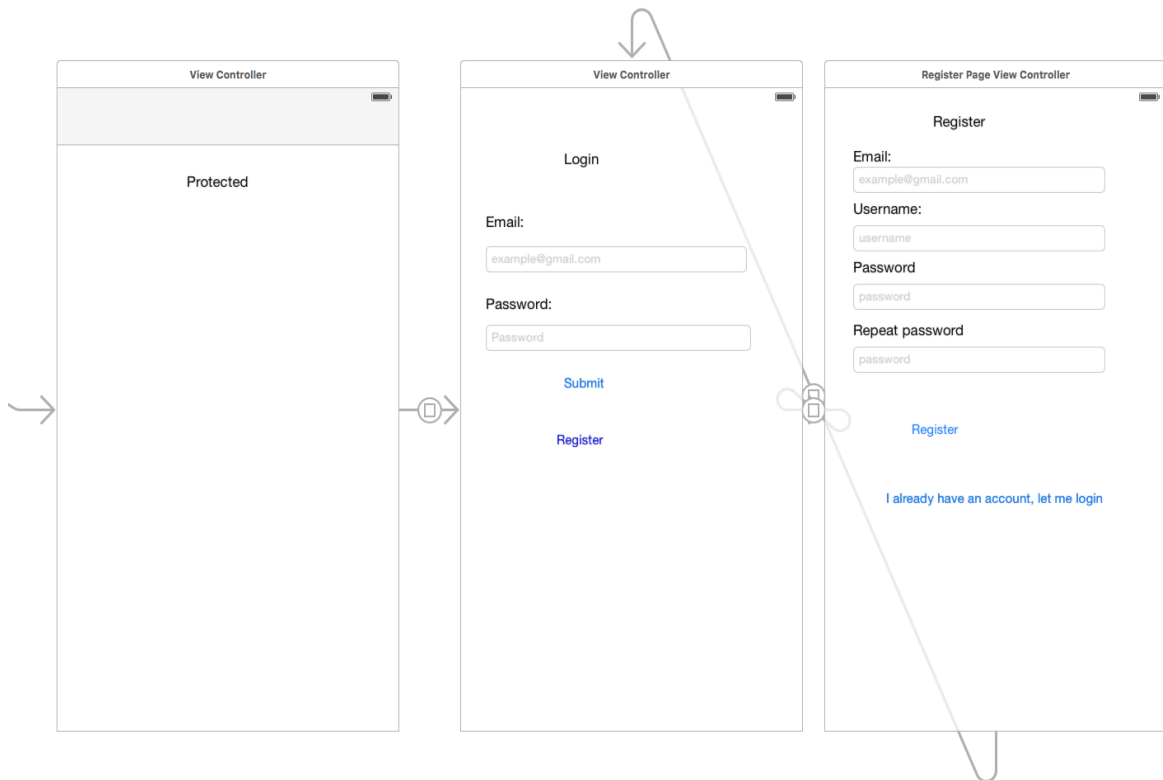


FIGURE 17. Initial User Authentication design.

At first I worked on an idea that a user needs an authentication system, for example as shown in the figure 17. The user needs to register and login to use an application. There is a text field for an email and a password to login, to register, the user has to submit their email address, username, password and repeat-password. If the user already has an account, they can simply click to a button “I already have an account, let me login”. After a successful login, the user can type their age and access their location to get started. However, it is a long process and the WEi! team wanted to have a direct access without this kind of authentication system.