

Nguyen Thong

**DEVELOPING GEOLOCATION CHAT BASE APPLICATION WITH IONIC  
FRAMEWORK**

**DEVEVELOPING GEOLOCATION CHAT BASE APPLICATION WITH IONIC  
FRAMEWORK**

Nguyen Thong  
Bachelor's thesis  
Spring 2016  
Business Information Technology  
Oulu University of Applied Science

## ABSTRACT

Oulu University of Applied Sciences  
Business Information Technology

---

Author(s): Nguyen Thong

Title of Bachelor's thesis: Developing geolocation chat base application with Ionic framework.

Supervisor(s): Tapani Alakiuttu

Term and year of completion: Autumn 2015

Number of pages: 42

---

In this thesis, a mobile application was developed based on web technology to demonstrate the power of flexible ability of the web platform. It is the chat application in which user can attend their chat room based on their current location and communicate with people near them.

The good mobile application can be achieved with the web platform instead of developing the native application which will required more development and maintain time for the owner. A web developer can reuse their web technology knowledge to develop mobile application. Ionic framework which was based on the popular JavaScript framework Angular provide infrastructure for developer when developing a large application with JavaScript language. The thesis will explain how Ionic and Angular providing the feature lacking in JavaScript which can make developing mobile application faster and easier. Furthermore, it will prove the flexibility and power of web technology on the mobile platform.

The project source code is developed under distributed version control system Git and Git Flow branching model to manage the development process and communication between developers. The thesis will also show how use Github as the host server for the project source code and discussion room.

---

Keywords: Ionic, AngularJS, Firebase, mobile development

# CONTENTS

1	INTRODUCTION.....	4
1.1	Background of the study .....	4
1.2	Objectives of the thesis.....	5
2	RESEARCH METHOD.....	6
3	THEORETICAL BACKGROUND .....	8
3.1	HTML5 and CSS3.....	8
3.2	JavaScript (AngularJS).....	10
3.3	Cordova .....	12
3.4	Firebase.....	14
3.5	Git and GitHub.....	15
4	DESIGN AND SYSTEM REQUIREMENTS .....	16
4.1	User Requirements Specification .....	17
4.2	Design of User Interface .....	17
5	IMPLEMENTATION .....	20
5.1	Project setup.....	20
5.2	Installing boilerplate and needed libraries .....	22
5.3	Structuring source code.....	24
5.4	Database modelling (Firebase).....	28
5.4.1	Database design .....	28
5.4.2	Querying data based on Geolocation location .....	30
5.5	Debugging .....	32
5.6	Chrome developer tool (with Batarang extension).....	33
5.7	Debugging on mobile device .....	34
6	CONCLUSION .....	37
7	DISCUSSION.....	38
	<b>REFERENCES.....</b>	<b>41</b>

# 1 INTRODUCTION

## 1.1 Background of the study

The mobile platform has been the fastest growing platform in the last decade since the first iPhone was introduced in 2007. Apple, Google and Microsoft have continuously created mobile devices and mobile operating systems that can compete directly with desktop operating systems. More and more Tech companies desire to compete in Mobile application market with a view to gaining profit as well as earning market position. Corpi Oy is no exception. Corpi Oy is a start-up company in Oulu region. The company has come up with the idea of creating a chat application on mobile platform. The special feature of this application is that it conjugates users based on their geographical locations. Users of this application own the capability of chatting and exchanging information with each other near them. Owing to the fact that chat application is clarifying its increasing popularity nowadays as can be seen from the success of Zalo, Whatapps, Viber, Kakao talk, etc, Corpi is raising hope for its own product. After reaching consensus on the idea of the product, the company has been searching for developers to implement the idea in terms of coding, designing and maintaining the application, thus, exposing opportunity for the author who is a last year Business Information Technology student. The author has been introduced to the company through his previous training supervisor, Pint Please Oy. After signing the agreement, the author has developed the application throughout a three-month period. This thesis work will discuss the development process as well as the outcome of the application. The subject choice of the thesis reflects the interest of the author in web and mobile application development. The purpose of the thesis is to give a full description on how a mobile application (specifically, chat application) is created and tested by a junior developer – the author. The work will also discuss function of the application and methods used. The core technologies used to develop the application combined HTML, CSS, Angular JS, JavaScript (as both backend and frontend tool), Ionic and Firebase as core databases. In terms of methodology, both quantitative and qualitative methods have been used. The author carried out multiple interviews including both direct and email interviews in order to discuss with the client company on modifying and improving the application throughout the process. In terms of desk research, a large amount of secondary data has been collected (especially through Information Technology Documentations) to support the project work. The project delivers clear benefit to the commissioner since it is their

purpose from the start to develop a chat application and push it into the market. The author has made attempt to fulfil all the requirements of the client, thus, helping them with building a desired application. On the aspect of the author, the thesis work is a great opportunity for him to apply the knowledge he acquired as a Business Information Technology student, to practice in a real-life situation through creating a mobile application, to understand better the main concepts in app development as well as to sharpen his skills throughout the project (coding skill, designing skill as well as communication skill).

## **1.2 Objectives of the thesis**

The main objective of this thesis is to provide readers with the in-depth knowledge on reusing web technology skills, concurrently, applying it on modern day of mobile development software. It includes the explanations on several topics of modern software development process, particularly, utilizing Git and Github as prominent version control system for keeping track of development process as well as the scrum method for dividing tasks and estimate the time of completion. Moreover, the main focus is going to be deeply explanation of organizing a large JavaScript application using Ionic and AngularJS framework as well as comprehensive description of how AngularJS framework works.

The outcome of the thesis work is a properly functional geolocation base chat application that owns the capability of running on both website and mobile platform (Android and iOS). On the client perspective, they will have the testing application for fundraising purposes to start a new company. Two main features of the application, first is the ability to show all other available users whom inside the searching radius based on user location and second is the chat notification system which shows the incoming and unread indicator of the latest messages.

## 2 RESEARCH METHOD



*TABLE 1: The research process*

In terms of collecting data, both qualitative and quantitative research methods have been applied. In consideration of secondary research, desk research has been carried out. Desk research is among the most common method nowadays. It involves processing data that has been collected by another party (DJS Research Ltd, date of retrieval 07.12.2015). In the scope of this thesis, multiple technological documentations have been consulted by the author in order to support his process of developing the application. Specifically, the most updated version of AngularJS, JavaScript, Ionic, Firebase, CSS, SASS documentations have been reviewed to support the author in coding, designing and testing the application.

In terms of qualitative research method, interviews have been carried out throughout the process. Interviews through direct meetings and through emails act as regular communication between the thesis worker and the commissioner. Interviews allow the client company to express their specific needs and desired modifications toward the application. Both structured and unstructured interviews have been implemented. Structured interviews (designed by the author and sent to the client company) serve the purpose of helping the author with developing the application based on certain rules and limitation in terms of his own technical background. Structured interviews often happen when modifications emerge as the client company attempt to change the application based on their requirements. Therefore, the author has knowledge on what the commissioner really wants and expects. After that, the author started modifying the application and its feature based on the client's requirements mapped out in the interviews.

In terms of quantitative research, pilot tests have been used among a certain number of possible users of the application. The application trials have been sent to a number of users. In return, the users submit their feedbacks to the author. The chosen trial users are mostly students from various universities in Finland (Oulu and Helsinki). By observing the test and the users' behaviours, the author achieve some valuable information to improve the application.



### **3 THEORETICAL BACKGROUND**

The main technologies used for building this application include mark-up tools, namely, HTML5, CSS3 and necessarily the use of JavaScript language. It was developed based on the Ionic framework and the AngularJS framework as the core. For styling application, instead of CSS, SASS technology was compiled to CSS during the building process. In addition, Cordova is used to provide the APIS (application programming interface) of the native mobile platforms. Firebase was used with the view to saving and maintaining the consistency of data as well as controlling the interaction between the application and the users. Git and Git repository hosting service (Github) used as the distributed version control tools for managing changes in the source code.

#### **3.1 HTML5 and CSS3**

HTML is the most widely used mark-up language for structuring and presenting content on a website. HTML5 is the latest official version of HTML language which is completed on 28 October 2014 by World Wide Web Consortium (W3C). The newest version has also added new geolocation API feature which was utilized to build this application.

“Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in mark-up languages HTML or SVG” (Mozilla Network Developer, date of retrieval 23.11.2015). The language adds the ability to separate the styling presentation from the document structure. Therefore, it improves the accessibility and flexibility in controlling the appearance of the web application.

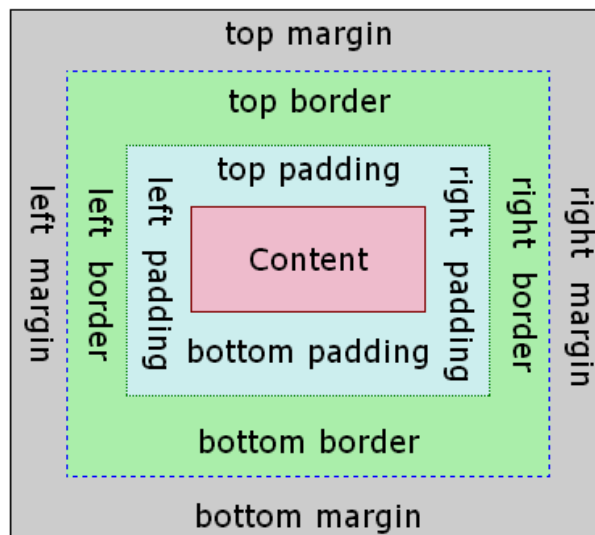


TABLE 2: CSS box model (CSS Box Model and Positioning, Code project, date of retrieval: 06.12.2015).

In consideration of styling for the web, CSS treats every element as they were in a box. The border separates the edge of one box from other surrounding boxes. The margin defines the distance between the border of a box and the box next to it. Padding is the space between the content of the box and its border. CSS is used to control the border, margin and padding on each side of a box along with all properties paired with them. Margin and padding are the two most important properties in creating the “white space” as designers refer to – the space between varieties of parts of the page (Duckett, Jon 2013, 232). For example, when both the border and the text (elements) are in black colour, the padding is in need of using to avoid the text touching the border and overlapping each other. There is one special feature in terms of margin. When a bottom margin of one element meets the top margin of another, only the larger of the two is showed (Duckett 2013, 233).

SASS is the CSS extension language which combines new features in order to make CSS more programmable. SASS is "transpiled" back to CSS in the end. Hence, the browser is able to parse it correctly. Those important features are variables, inheritances, operators, mixins and module system which are applied to develop the application. SASS's transpiler was originally written in Ruby language. However, it has gained a lot of popularity in the web development community recently. Therefore, it is also available with C/C++ compiler version (LIBSASS) for the purpose of integrating with other languages outside the Ruby ecosystem.

### 3.2 JavaScript (AngularJS)

JavaScript is the light weight programming that can run on the browsers so it was used to control the logic of user interaction on the websites. JavaScript was created by Brendan Eich at Netscape in 1995 even though it shares the common word Java with Java language, those 2 languages are barely have anything in common. JavaScript is mainly constructed more like functional language instead of Object Oriented language like Java. The latest version of JavaScript that compatible with all the browsers is ECMAScript 5 used to develop this application instead of ECMAScript 6 which is just partial support by some browsers.

AngularJS is modern JavaScript framework invented in 2009 by Misko Hevery at Google. The framework is popular in JavaScript community due to its declarative template as well as the heavily support from Google to actively maintain the project. The version use in the thesis is the stable version 1.3.20 released on September 30 2015. AngularJS support MVC or MVVM architectures, due to the size of this application, MVC pattern was chosen because of simplicity than MVVM.

JavaScript itself doesn't have a modules system instead every JavaScript files are loaded into the browsers by using a <script> tag. Hence, web developers are resolving their JavaScript application dependencies manually by paying attention to the order of <script> tag in the html file that is going to request those JavaScript files. Due to this missing feature in JavaScript language has created a lot of most of variables and functions available at global scope. However, a large application needs encapsulation to ensure the data flow doesn't collide or overwrite accidentally. AngularJS has introduced to its own module system to add this feature to the language.

```
var app = angular.module('app', [  
    uiBootstrap  
])
```

*FIGURE 1: AngularJS module system*

AngularJS module system is actually JavaScript Object which contain module function that 2 parameters, first is the string of your app name and second parameter is the array of your dependencies in this example it is uiBootstrap.

Among the module system AngularJS also providing the \$scope object to bind between view and the controller and model in our MVC application. Scope object in AngularJS again is just a plain JavaScript object. The binding feature is loop that can check every properties of the Scope object, if the changes has been detected the value of properties is going to be updated. This loop in AngularJS has a special name called “dirty checking”.

```
angular.module('scopeExample', [])
.controller('MyController', ['$scope', function($scope) {
  $scope.username = 'World';

  $scope.sayHello = function() {
    $scope.greeting = 'Hello ' + $scope.username + '!';
  };
}]);
```

FIGURE 2: AngularJS \$scope object (source: AngularJS documentation Scope section)

Controller utilizes this \$scope object to reveal the method to view of application which is AngularJS template. The template is html page that has special AngularJS attributes such as ng-app, ng-controller..., these templates will be evaluated by AngularJS to product html pages that the browsers can understand. AngularJS Template promotes the declaration when developing software application which is its strongest point compare with other frameworks such as EmberJS or ReactJS.

```
<body ng-app="app">
<div ng-controller="MyController">
  Your name:
  <input type="text" ng-model="username">
  <button ng-click='sayHello()'>greet</button>
  <hr>
  {{greeting}}
</div>
</body>
```

FIGURE 3: AngularJS template engine (AngularJS documentation section)

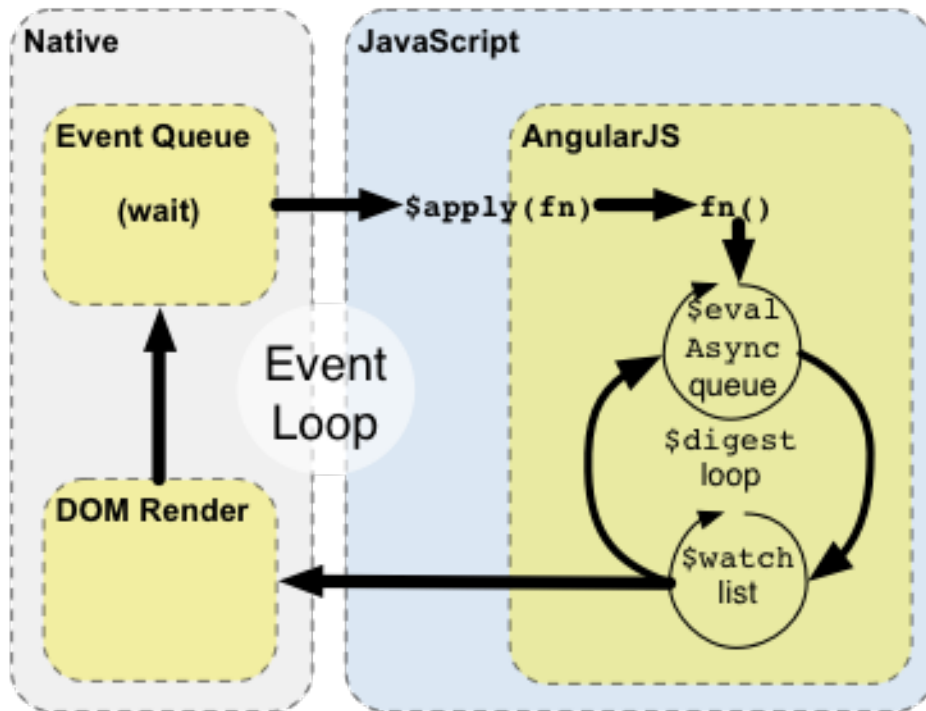


TABLE 3: Integration with browser event loop (Source: <https://docs.AngularJS.org/guide/scope>)

Browsers are event processing software, JavaScript on the browsers has single-threaded environment with event loop for processing all the events that happened inside the browser. When a browser receive HTML and CSS files it is going to parse and render the site on the computer screen. The DOM is the Document Object Model is the API of the user interface browser just render which is available to the JavaScript to access its components. Because of its single-threaded nature of JavaScript runtime, browsers implement the event queue to serialize the call back functions it needs to be executed when events happen. The Event Loop picks these call back functions to execute in order until there nothing left in the queue. So if those function also register in the AngularJS context it is automatically running its \$digest loop to update the properties of all \$scope object.

### 3.3 Cordova

Cordova is core engine of PhoneGap own by Adobe, the framework providing JavaScript interface for accessing data from the native apis on mobile platforms like Android, iOS and Windows phone. Cordova itself was donated by Adobe to the Apache Software Foundation therefore it is free to use for commercial purposes. The plugins ecosystem of Cordova has growth

sufficient that able to provide almost any anything native functionalities throughout the JavaScript interface. The environment would be almost the same as the browser on the desktop but the developer is having more data from mobile device's sensors.

## PhoneGap Architecture

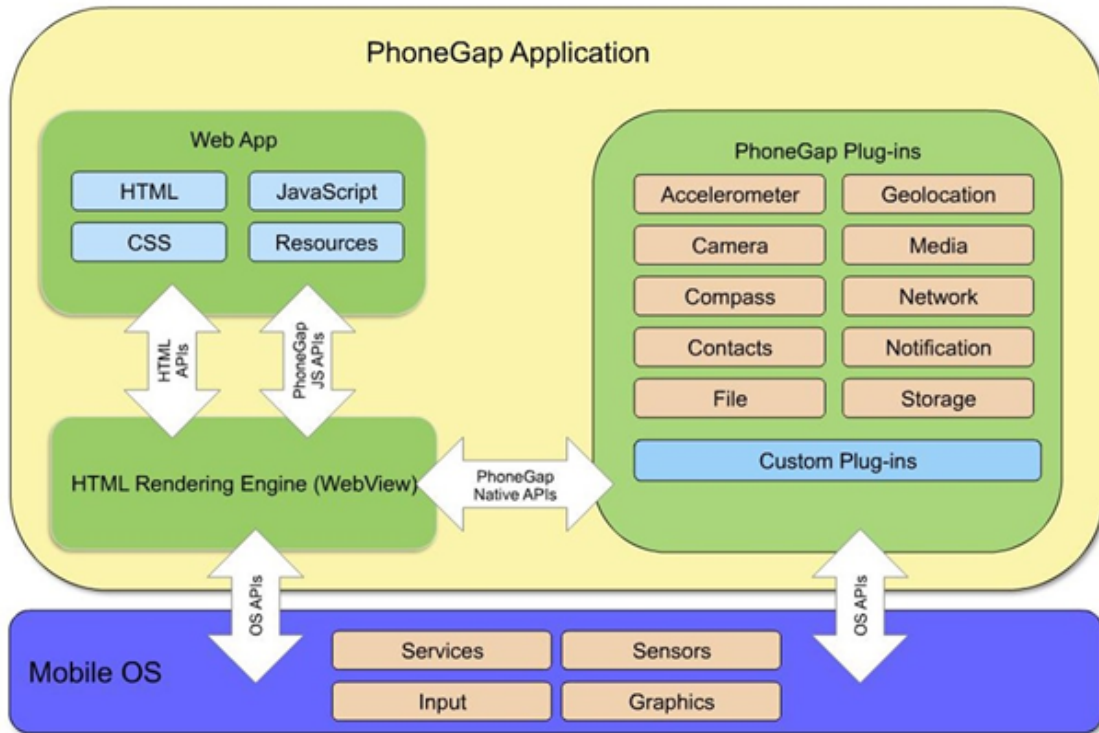


TABLE 4: PhoneGap architecture

Ionic is hybrid mobile framework that built on top of Cordova and AngularJS. The goals of Ionic is providing the mobile framework for web developers so that they can develop mobile application easily using web technology. In addition, it included a lot of components to bridge the gap between native application and hybrid application running on different mobile platform such as UI components (navigation bar, notification system ...). Moreover, the framework also provides the build process for Android and iOS with their Ionic Package service.

### 3.4 Firebase

Firebase is real-time database platform that support syncing database across multiple connected devices at the same time. The database has AngularJS wrapper module called AngularFire that expose the database apis and integrates with the internal system of AngularJS framework. The data-binding in AngularJS which is implementation of observer pattern is also being observed any changes by the Firebase. Therefore if there is one change of a property on the Scope object in variable it is noticed by the database and the data stay in the property will also be changed. In the scenario like the chat application we are building in this thesis, multiple people are connected to the same room, let assume our chat conversation is a JavaScript Array built-in data structure and in the AngularJS context, when there is a message available this modified array will be modified all across the devices that connects to the chat room. And that is call 3 ways data-binding in Firebase database terminology.

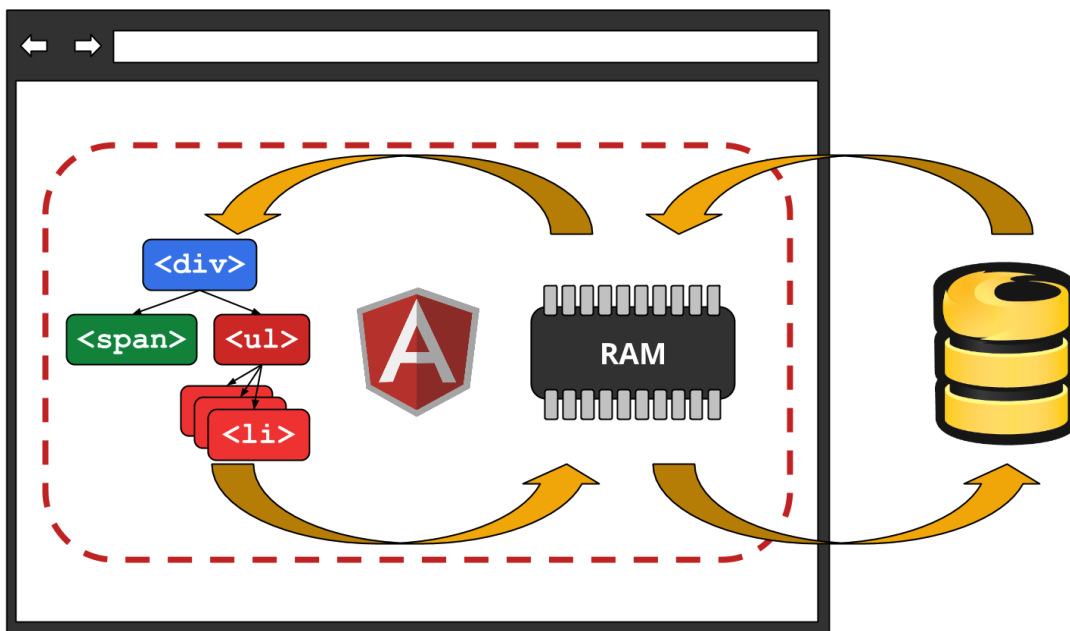


TABLE 4: Three ways data-binding with AngularFire. (Firebase company blog, date of retrieval 29.11.2015)

Firebase is a NoSQL database therefore JavaScript developer can use the same JavaScript data structure (Array, Object) to store their data directly.

### 3.5 Git and GitHub

Git is free and open source distributed version control system that assist developers keeping tracks of changes when they are developing the application. Git was developed for the purpose of managing Linux Kernel development then it has become one of the main stream version control system due to the support of GitHub. To install Git on the system, users can download the UI installer from <https://git-scm.com/> , Git can run on 3 main operating system MacOS, Linux as well as Window.

GitHub is a free hosting open source git repositories with many features to support communication between contributors of the repository. Nowadays most of the open source projects are living on GitHub this makes it become the largest social network website for developers.



## 4 DESIGN AND SYSTEM REQUIREMENTS

Because of this application will be running on mostly Android due the limitation access to the resources of the author. Developing iOS's application needs the iOS stimulator which is only running on Mac OS, on the other hand Android platform was built with JVM technology therefore developer can run the stimulator in any desktop operating systems.

This application is going to implementation the general functionalities of chat application and special one is searching all users based on their locations to join the chat room, each chat room will have to specify the radiation of user location from the place it was created and all the users inside that chat room can join the conversation.

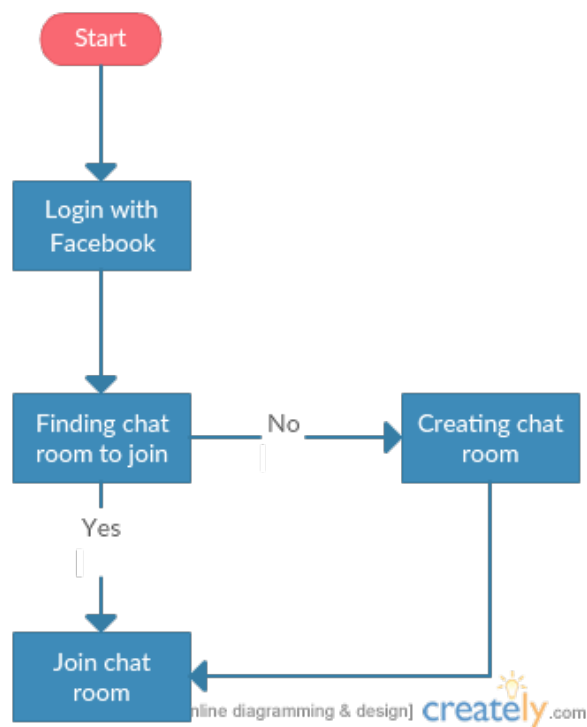


FIGURE 4: Use's process flow

## 4.1 User Requirements Specification

Features list:

- The chat application has ability to integration with Facebook authentication system and use their profile picture as the chat picture profile.
- The chat application able to find all the chat rooms inside the user's desired radius as well as color-coded chat room user can join.
- All the users ensure to see the latest message if they are online.
- The chat application has ability to visualize the user's location and chat room locations.
- The chat application has ability to notify the user which chat room they are in and receive the notification from that chat room for unread message.

Non-functional feature:

- 
- The application is designed in MVC pattern and module pattern for easier to maintain and organize functionality.
- The asynchronous code is written in promises instead of call backs to ensure the readability and being chainable.
- Each feature of the application is developed in its own git branch before merge back to develop for testing than merge with the master branch. Master branch is always stable of the application.
- Licences of libraries using to develop application must be free and open source due to the limitation of resources providing for the project.

## 4.2 Design of User Interface

The user interface of application is designed following the guidelines of Material Design which is Google's visual design framework. The guideline aim is going to help improve the overall interaction between user and the application. The aim of Material Design framework is focusing on the meaningful communication between users and the application hence the users don't have to think how to navigate around the application. Every inputs of user to the application, it can provide the feedback back to user therefore they can feel the software as a physical thing. The

location of user is always being update to the header text of application. Each components overlap each other always showing shadow to indicate it has different layers.

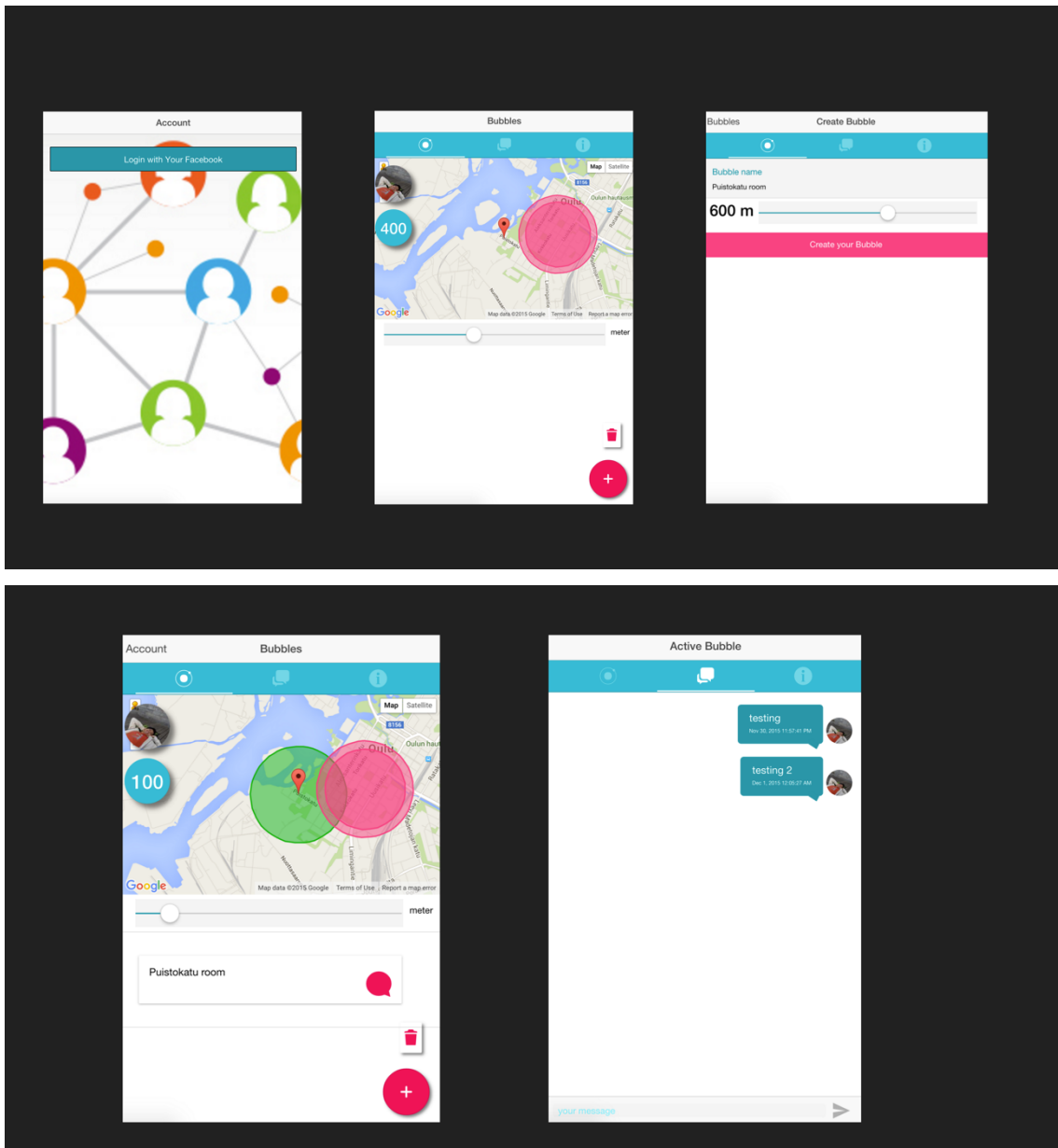


FIGURE 5: Application design prototype

When user first use the application they will be asked to log in with Facebook authentication system. With this Facebook authentication token the application is able to update user profile picture automatically. After the login page user is navigated to the main page where all the rooms will be listed if it is available, those rooms are being visualized on Google Maps as well as the

current user location. If the room is accessible for user it is drawn as green circle following its radius, in contrast those rooms are red coded.

## 5 IMPLEMENTATION

### 5.1 Project setup

Most of the project dependencies and build tools are built on top of the Node.js platform, Node.js at the core is the V8 JavaScript run time engine which is also the same run time developed for Chrome by Google. It enables to run JavaScript outside the browsers environment. To install Node.js we just need to download it from <https://nodejs.org/en/download/>.

When Node.js was installed it will come NPM which is the Node package manager like PIP or GEM in the python and Ruby ecosystem. Cordova and Ionic are available in NPM system therefore to install these tools, typing below command to the command line system is needed.

```
npm install -g cordova
npm install -g ionic
```

*FIGURE 6: Command for installing Cordova and Ionic*

Now Ionic command line tool is available globally in the system. For the purpose of writing this thesis the author already has built the application and hosted it GitHub repository ([https://github.com/nguyenthong/Geo\\_chat.git](https://github.com/nguyenthong/Geo_chat.git)). The project is downloadable using clone command providing by Git.

```
git clone https://github.com/nguyenthong/Geo_chat.git
```

*FIGURE 7: Command for cloning the project repository*

After the running the command finish, Git pull the project from remote GitHub repository to the location the command was ran. The name of the project as well as the folder is Geo\_chat. The project folder contains several directories.

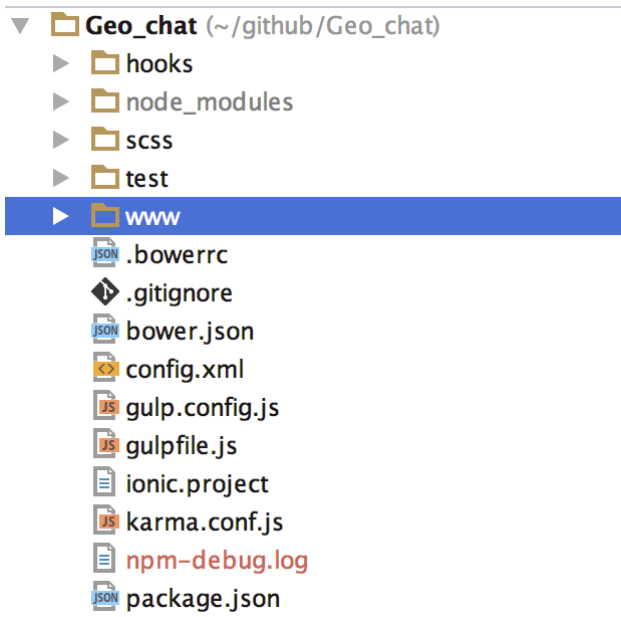


FIGURE 8: Application folder organization

The www directory is main directory where all the application HTML, SASS and JavaScript files are placed. The package.json and bower.json are files in which all our application dependencies are listed and managed by NPM and Bower. Gulpfile.js is written down all the automation build tasks by Ionic. Config.xml is the Cordova configuration file.

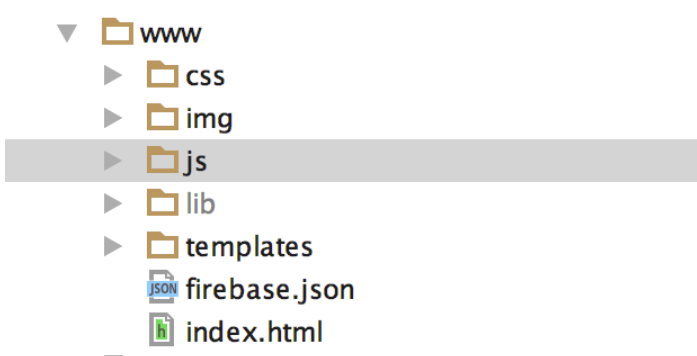


FIGURE 9: Application folder organization, www folder

Inside the www folder files are categorized by its extensions, template folder holds the .html template files, js folder hold all the .js files and css folder hold .css file. The lib directory is where

all of the libraries defined in bower.json are downloaded and loaded through the index.html. Bower is popular package manager for front end library.

## 5.2 Installing boilerplate and needed libraries

In the bower.json all of the libraries the application is utilizing are declared in the JavaScript object notation data structure. The application is depend on AngularFire, the firebase library provided for AngularJS developer for integrating with the database. Angular-google-maps is the google maps library wrapper for Angular, it is used to the display the location of user among with visualization of the chat rooms around user location. NgCordova is also the library wrapper around most used Cordova APIs, the library provide these APIs in the form of AngularJS service therefore it assist developer write the code cleaner with AngularJS architect.

```
{
  "name": "Geo_chat",
  "private": "true",
  "devDependencies": {
    "ionic": "driftyco/ionic-bower#1.0.0-beta.14"
  },
  "dependencies": {
    "angularfire": "1.0.0",
    "angular-google-maps": "2.1.0-X.10",
    "geofire": "~3.2.1",
    "ngCordova": "~0.1.12-alpha",
    "firebase": "~2.2.3",
    "angular-elastic": "~2.4.2",
    "rsvp": "~3.0.17"
  }
}
```

FIGURE 10: Application dependency using bower dependency

To install all the dependencies of our application, typing the command below to in the project root directory where the file bower.json is placed.

```
bower install
```

FIGURE 10: Application dependency using bower package manager

And then for all the NPM package which the application is going to need to run the development server and the build process for the application.

```
{
  "name": "geo_chat",
  "version": "1.0.0",
  "description": "Geo_chat: An Ionic project",
  "dependencies": {
    "gulp": "^3.5.6",
    "gulp-concat": "^2.2.0",
    "gulp-minify-css": "^0.3.0",
    "gulp-rename": "^1.2.0",
    "gulp-sass": "^0.7.1",
    "q": "^1.2.0"
  },
  "devDependencies": {
    "bower": "^1.3.3",
    "gulp-connect": "^2.2.0",
    "gulp-inject": "^1.2.0",
    "gulp-load-plugins": "^0.8.1",
    "gulp-ruby-sass": "^1.0.0-alpha.3",
    "gulp-util": "^2.2.14",
    "shelljs": "^0.3.0",
    "wiredep": "~2.2.2",
    "plumber": "~0.4.8",
    "karma": "~0.12.31",
    "karma-chrome-launcher": "~0.1.7",
    "jasmine-core": "~2.2.0",
    "karma-jasmine": "~0.3.5",
    "karma-firefox-launcher": "~0.1.4",
    "karma-phantomjs-launcher": "~0.1.4",
    "karma-browserify": "~4.0.0"
  },
  "cordovaPlugins": [
    {
      "id": "org.crosswalk.engine",
      "locator": "./engine/cordova-crosswalk-engine-c0.6.2"
    },
    "org.apache.cordova.geolocation"
  ]
}
```

FIGURE 11: Application dependency using NPM package manager

The application Cordova plugin install by Ionic is also listed under package.json. The Crosswalk engine plugin is the web runtime environment, instead of using the web view in Android devices, your application has its own web runtime. This will ensure your application run across multiple android seamlessly with consistent APIs and performance. The trade-off of this plugin is the size of application, generally the application size is about 30mb that will make the users wait longer to



download the application from the app store. To install all these dependencies developer just need to type below command in the root directory. NPM automatically resolve all these dependencies and download them from NPM server.

```
npm install
```

### 5.3 Structuring source code

The source code is organized following the module pattern therefore each page user navigate in the application has its own folder. These folder are different from AngularJS module because it sits inside our geo\_chat module, these folder can be imagined as mini module. This set up allow developer easily finds files needed as well as scale the application size effortlessly compared to the traditional MVC structure folder in which every controllers and views have to be under controller and view folder.

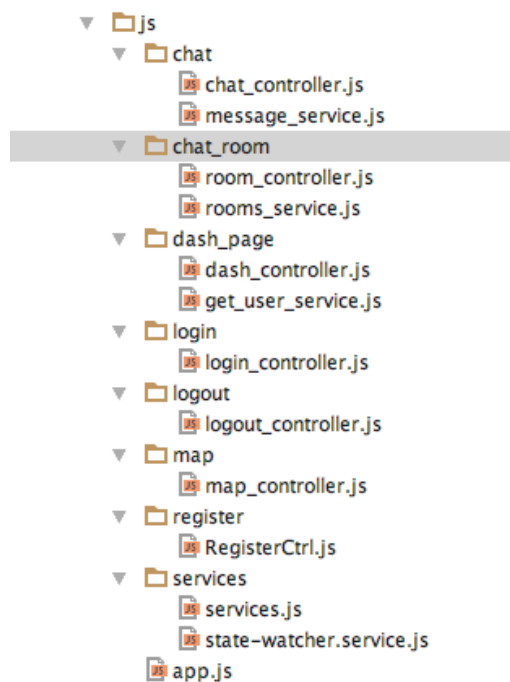


FIGURE 12: Structure of the source code

Each component has its own controller and services, unfortunately the view template of these modules are placed under the template folder because it is Ionic folder structure categorized its file according to its extension. However, it is not a problem if developer want to move the template file to its own module.

Ionic framework routing utilizes the UI-Router component instead of AngularJS routing system. Ionic framework routing system added their own directive suited for mobile routing scenario such as `<ion-tabs>` to present all nested tabs element inside the application, the `<ion-tab>` to update the state title to show where users in the application.

```
<ion-tabs class="tabs-striped tabs-top tabs-background-balanced tabs-color-light">
  <!-- Dashboard Tab -->
  <ion-tab title="Status" icon-off="ion-ionic" icon-on="ion-ionic" href="#/tab/dash">
    <ion-nav-view name="tab-dash">

      </ion-nav-view>
    </ion-tab>

  <!-- Chats Tab -->
  <ion-tab title="Chats" icon-off="ion-ios7-chatboxes-outline" icon-on="ion-ios7-
chatboxes" href="#/tab/chats">
    <ion-nav-view name="tab-chats"></ion-nav-view>
  </ion-tab>
</ion-tabs>
```

FIGURE 13: Application source code in `www/template/tabs.html`

This UI-Router empowers developer model the application into state machine and can be nested to each other. Moreover, the URL can be used to navigate around the application as same as normal website, this feature is useful if the application is published under developer domain. Ionic fork some of directives of UI-router and added new specific features required for basic navigating around mobile app. These directives has its own controller, model and services defined by Ionic. The source code of these component is living in the `ionic-angular.js` file.

```

IonicModule
.directive('ionNavView', [
  '$state',
  '$ionicConfig',
function($state, $ionicConfig) {
  // IONIC's fork of Angular UI Router, v0.2.10
  // the navView handles registering views in the history and how to transition between
  them
  return {
    restrict: 'E',
    terminal: true,
    priority: 2000,
    transclude: true,
    controller: '$ionicNavView',
    compile: function(tElement, tAttrs, transclude) {

```

FIGURE 14: Ionic-framework source code

In AngularJS the directive name is the case is ionNavView is going to be normalize when it goes through the internal framework and become the new html tag <ion-nav-view></ion-nav-view>. When the browsers encounter these invalid html it is going to skip it however because it is in the AngularJS context, these tags are recognized by AngularJS and start to compile into valid html tag. This feature of AngularJS allow to reuse these directives as UI components and can be easily moving these tag around.

All the routes of the application is defined in the app.js file placed under www/js folder . Tab state is the root state of all states are children of tab state that why it is needed to declare “abstract: true”. Each state has its own template and controller for defining the behaviours except some directive components the application took advantage from provided Ionic framework.

```

.state('tab', {
  url: "/tab",
  abstract: true,
  templateUrl: "templates/tabs.html"
})

// Each tab has its own nav history stack:

.state('tab.dash', {
  url: '/dash',
  cache: false,
  views: {
    'tab-dash': {
      templateUrl: 'templates/tab-dash.html',
      controller: 'DashCtrl'
    }
  }
})

.state('tab.createRoom', {
  url: '/createRoom',
  views: {
    'tab-dash': {
      templateUrl: 'templates/createRoom.html',
      controller: 'CreateRoomCtrl'
    }
  }
})

.state('tab.chats', {
  url: '/chats/:roomId',
  views: {
    'tab-chats': {
      templateUrl: 'templates/chat-detail.html',
      controller: 'ChatCtrl'
    }
  }
})

```

FIGURE 15: Application routing source code in app.js

The main state is tab and its URL is under “/tab” and the template is tab-dash.html under them the templates folder. The state tab.dash is child state of tab state as well as tab.createRoom and tab.chats. In the state tab.chats developer can pass the parameter required for the next state through the url: ‘/chats/:roomId’, the parameter is the roomId.

## **5.4 Database modelling (Firebase)**

### **5.4.1 Database design**

Firestore database is NoSQL database which mean it cannot be modelling as normal relational database such as MySQL or PostgreSQL. Firestore can store JSON (JavaScript Object Notion) and it is capable of store to 32 levels deep of the object. However because of it limitation in tern of querying the database, the JOIN method in SQL language is not available to the developer. One of the strategy of solving this problem is flattening the data and use the common key to join the data after querying it. The database design for this Geo\_chat application utilizes the same approach.

UML diagram is not compatible when developer visualizes the database design for NoSQL database. Fortunately, Firestore utilizes JSON format, the branching visualization can be used to present the design.

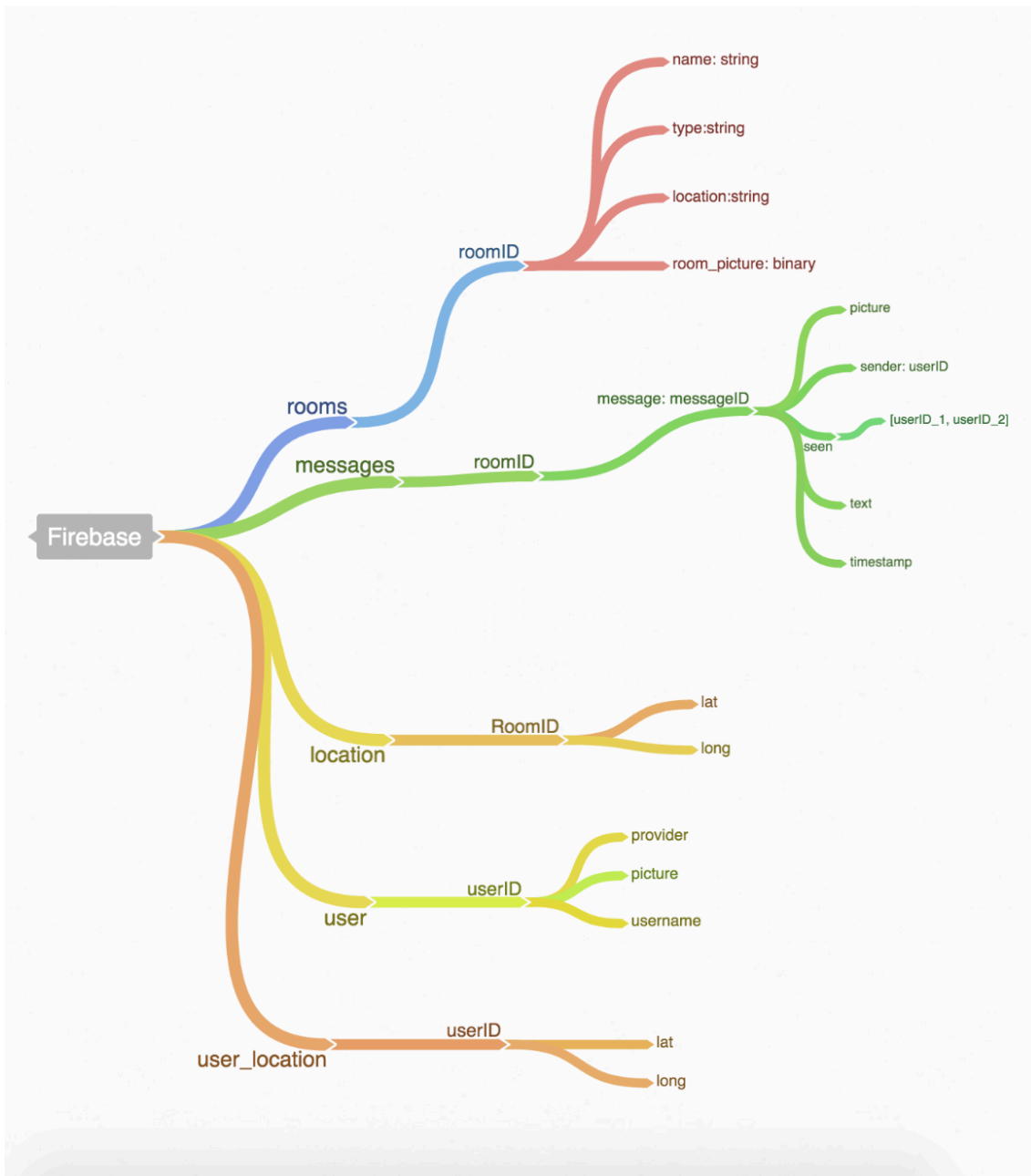


FIGURE 16: Database design visualisation

The database has 4 main keys at level 1, each of these containing the array of the room information related the key. The roomID is the unique key which is generated by Firebase to ensure the uniqueness of the key. The data is being divided into smaller section, this design can improve the speed of querying our database. When developer need to ask for the members inside specific room, they only have to send the roomID and will get back the only needed information instead of big JavaScript object contain every information about the room which can

slow down the responsiveness of the application. This operation is named flattening the data which turn large nested data into 2-3 level deep.

In the messages section each room contains its own messages marked by messageID in which these messageID will specify the picture of the sender, senderID and the text contain as well as the timestamp. In the seen field, it will have array of all the users have seen this message based on userID.

#### **5.4.2 Querying data based on Geolocation location**

Firebase has support for geolocation querying which mean the application can query specific given geographic area based on Firebase geolocation object. For creating this Firebase geolocation object Firebase is providing a GeoFire library in JavaScript language as well as Java and Object-C. Below is the code sample from the application source code utilizing the GeoFire library to creating geolocation object for the room.

```

function RoomsService( ROOMURL, MSGURL, MEMBERURL, LOCATIONURL, $q,
$firebaseArray, uiGmapGoogleMapApi) {
  var roomRef = new Firestore(ROOMURL);

  var geoRoomRef = new Firestore(LOCATIONURL);
  var geoFire = new GeoFirestore(geoRoomRef);
  var geoRef = geoFire.ref();

  var messageRef = new Firestore(MSGURL);

  var memberRef = new Firestore(MEMBERURL);
  var fireMember = $firebaseArray(memberRef);

  return {
    createRoom: function CreateRoom(newRoom) {
      var deferred = $q.defer();
      //todo fix the memberRef and MessageRef creation base on newRoomRef.key()
      var newRoomRef = roomRef.push(newRoom, function(error) {
        if(error === null){
          var successMessage = "Your room is created";
          deferred.resolve(successMessage);
        }
      });
      var newRoomID = newRoomRef.key();
      //using newRoomID to set ID for messages, locations, members
      memberRef.child(newRoomRef.key()).set({
        empty: true,
        info: {}
      });
      messageRef.child(newRoomRef.key()).set({
        empty: true,
        info: {}
      });
      //setting location for the room
      geoFire.set(newRoomRef.key(), newRoom.location);
      return deferred.promise;
    },
  },

```

FIGURE 17: Application creating room based on location source code in room\_service.js

The Firebase geolocation object must be the Firebase object therefore at first the new Firebase Object is defined as “var geoRoomRef = new Firestore(LOCATIONURL);”. This object contains all the references needed for the location in which the geolocation object required to be stored. Then, developer can pass this references to the constructor method of GeoFirestore to create the object “ new GeoFirestore(geoRoomRef) ”. After all these step now developer can set the location for their room using the latitude and longitude information from the GPS sensor.

When developer want to query the database, geoFirestore object just need to set the parameters of the center and the radius from that center in kilometres. Then, developer can listen to different events sending from Firestore such as ready, key\_entered, key\_exited. In this application, it is



listened to the event `key_entered` to know if the user in rooms specified radius.

```
var geoQuery = geoFire.query({
  center: location,
  radius: distance + DISTANCE_ADDED
  //radius in kilometers
});

geoQuery.on("key_entered", function(key, location, distance) {})
```

*FIGURE 18: Listen for event `key_entered` , source code in `room_service.js`*

## 5.5 Debugging

Debugging is unavoidable step when developers developing software, even considering how careful a developer can be. Fortunately, hybrid application can be debug easily and iteration quickly compare to the native application development. The reason for this fast process is developer doesn't have to wait until the source code to compile to see the result.

Web developer can reuse their debugging skill with hybrid application because under the hood JavaScript is still the main application logic layer. Using the chrome developer tool developer can access debugging tool for JavaScript and css, html rendering such as setting a break point, step into function to inspect what happened when their code doesn't behave as expected. Moreover, the Ionic framework in the end is built on top of AngularJS framework and there are good developer tool developed specifically by AngularJS team for debugging name Batarang. It helps AngularJS developers see all the scope and model of the application among profiling the performance. The tool can be download from the Chrome web store (<https://chrome.google.com/webstore/detail/AngularJS-batarang-stable/niopocochgahfkiccpjmmmpchncjoapek?hl=en>)

## 5.6 Chrome developer tool (with Batarang extension)

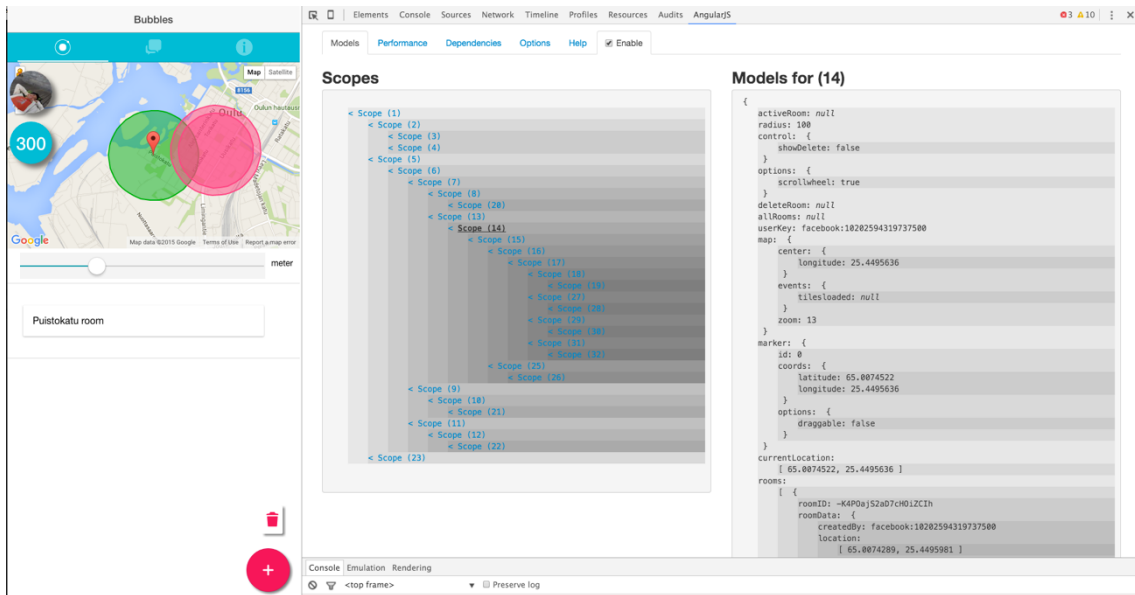
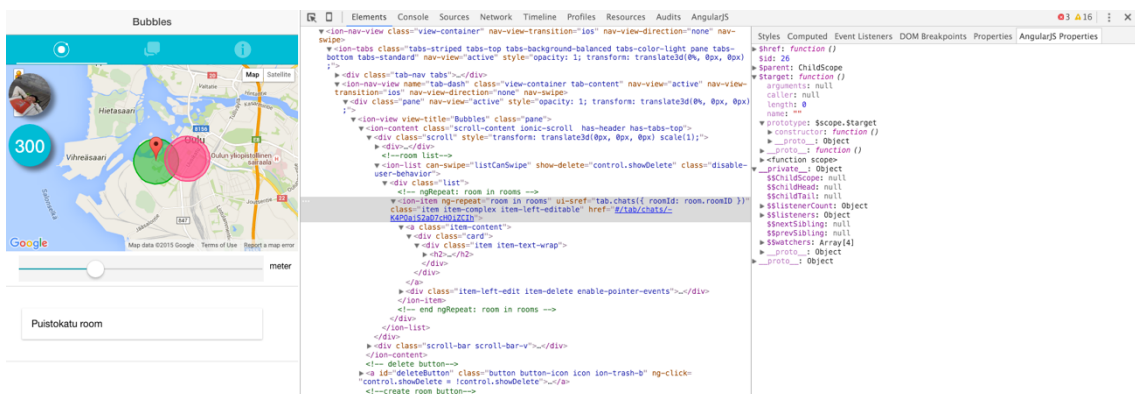


FIGURE 19: Batarang, AngularJS debugging tool

With Batarang developer have ability to see the structure of all the scope of the application as well as the model that is attached to the scope. The scope can have other nested scopes inside them therefore developers are easily losing tracking of the scope model especially with the application using a lot of directives from Ionic. Because directives can have 3 different type of settings for its scope, they can have inherited scope or shared scope as well as isolated scope.

Batarang is also providing the element inspector to see which scope is attached to specific Dom element on the application.



*FIGURE 20: Batarang, AngularJS debugging tool element inspector*

For example such as above, the <ion-item> directive has AngularJS properties attached, the scope's Id is 26 and has \_\_private object contains all its child scope and all its watchers. Following these tracing information developer can debug this directive.

### **5.7 Debugging on mobile device**

Sometime when the application using a mobile native api such as GPS, normal browser have the same api for location however this api is based on the IP addresses hence it is not as accurate as the GPS sensor on mobile phone. For examining the application developer need Android stimulator, in this thesis the author is going to use GenyMotion stimulator instead of the provided Android stimulator due to the better speed GenyMotion providing.

GenyMotion is free for downloading if it is used for personal development. Developer can download from <https://www.genymotion.com/#!/download>. After complete downloading and installing GenyMotion, developer has ability to examine the application running on different android devices provided by Genymotion. The commands below will automatic run the build process for the application on Android platform. After that, it install the application on the virtual device and automatically start the application.

```
ionic platform android  
ionic run android
```

*FIGURE 21: Command to run application in virtual mobile devices*

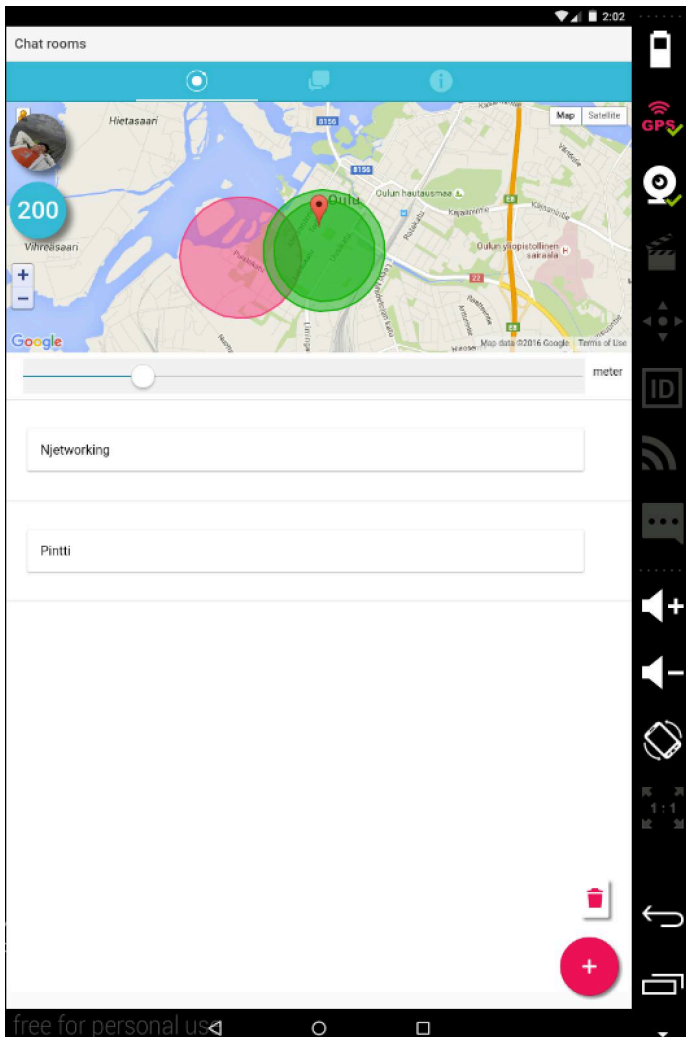


FIGURE 22: The application running on android virtual device.

Because android web view is using mobile chrome to run the application, developer can utilize chrome developer tool for debugging. In desktop chrome browser, heading to <chrome://inspect/#devices>, chrome will list all the available application running through chrome web view on mobile device. Developer can click the inspect link which will open the developer tool for running application according to its title.

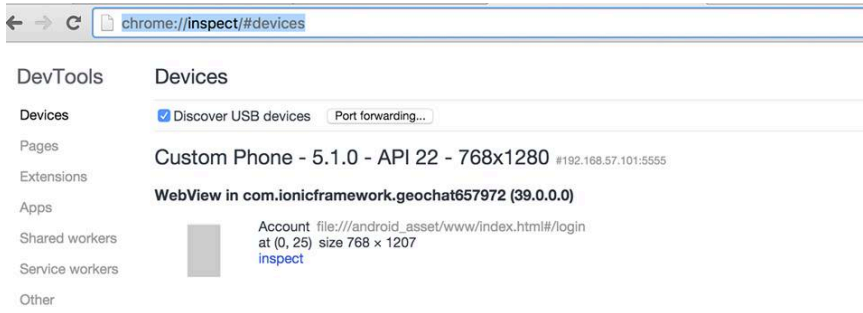


FIGURE 23: The application is listed on chrome inspect tab

## 6 CONCLUSION

With the application the client is now can ask users and investors to review the idea. Experiences has been built up during the time developing application with Ionic and AngularJS framework. Ionic has been proven that it is a sufficient framework, which can provide necessary components needed for most of development time. Developing a mobile application with web technologies that can have the same features sets like native application is doable task. Even though, Ionic at the time developing the application has not been reached 1.0 version, its apis has changed frequently which created a broken application several times. Therefore, managing and self-constructed source code for building software application is the important part of software developer in project. Developer should always use package control system as well as version control system to maintain it compatibility with all the dependencies. Firebase has helped shorten time for developing sever to serve as the backend for application. However, not all the backend operations can be done without the application backend such as indexing all the messages for searching and querying. For improving the performance and feature sets of the application is required in the next step to develop the backend to serve the application.

Finally, in the end of the project all the main requirements at the beginning are met and the processes for developing mobile application are explained in structured order. However, the development procedure has been missing writing unit tests to ensure the application logic's layer work properly. This step will guarantee the application working correctly after most of software development processes such as updating dependency packages or refactoring. Furthermore, it can reduce the development debugging time because all those test cases can guide developer to which part of the application is broken. AngularJS has good support for unit testing using angular-mocks module with Jasmine, AngularJS itself is tested using Jasmine. Its model using dependency inject ensure every part of application built with Ionic can be testable. From the technical perspective, this thesis offers its readers all the information needed for building mobile application with Ionic framework with modern JavaScript tools.

## 7 DISCUSSION

At the beginning of the thesis, it was considerably challenging in terms of schedule and application's feature issues. The commissioner and the author have met few of difficulties before reaching consensus. In particular, since the commissioner owns business oriented background, he has encountered some idea conflicts with the author who is more technical oriented. The client gave a relatively short time period (only three months) for the author to develop an application with many features. Thus, it was a bit pressured for the author to accomplish his task. The meeting schedule was tight in contradiction with the large size of the project (due to the commissioner's hectic working schedule). Direct meetings were held every week form one to two times, otherwise, the author had to discuss with the client through emails and phone calls. At every meeting, the CEO of the company reviewed the application, giving feedback and making request for modifications as well as expecting on improvement. The outcome of the meetings is more and more information on the product updated. The client company had required several modifications toward features of the chat application, in particular, how the chatroom should look like, how notifications should appear, how the users interact with the app, etc. All these changes led to changes in the code written by the author as well as required few updates in technology used.

The working process itself was one hand demanding but on the other hand was ultimately fascinating and motivating. Although at the beginning, the work pace was slow; there were difficulties in reaching agreement with the commissioner along with some delayed meetings, the author was very enthusiastic since he had the chance to gain some working experience. Besides, with extreme interest in coding, this project work was rather more fun to him than burdensome and exhausted.

The empirical research was conducted as personalized sets of interviews between the client and the author, together with pilot test among a number of possible users. The interviews were on a regular schedule every week on Wednesday and Friday. Exchanging information through emails also played an important role in the process. Pilot test was conducted at two campuses of Oulu University of Applied Sciences (Kotkantie Campus and Teuvo Pakkalan Campus) among mostly students (BIT and IB students). The chosen users of the test were friends and acquaintances of the author as well as some randomly picked. The test was implemented on IOS, Android as well

as Window phone platform. At the test, the application functioned properly on IOS and Android platform, remaining only some troubles and hindrances on Window phone platform. Particularly, on window phone platform, the users could not adjust the distance of the locations or could not view the address when entering the chat room. The test resulted in great modifications with a view to improving the application. Giving own assessment on the used method, the author highly values the pilot test since it had confirmed the performance of the application among real users on various mobile platform. This motivated a lot in terms of improving the application as well as helped the author to realize all the strength and weakness of his own work.

Prominent success factors were time management, the process of selecting users for the test and learning ability of the author. Since the given time was short and challenging, the author had to generate a good schedule to develop the application and fix the problems along the way. The pilot test was also indispensable. The author had to carefully select the possible users. The test included two groups of participant. The first group is composed of people with strong technical background who is familiar with giving feedback on mobile application development. This group has significant impact on modifying the product on technical side, giving comments and advices on possible modifications. The other group is “non-tech” users. These users are responsible for evaluating how user-friendly the product is, how easy to interact with the application and any difficulties met while using the app. Moreover, the author also made sure the presence of various mobile platforms at the test, meaning participants used different kinds of mobile phone including IOS, Android and Window phone. Learning skill and hard work of the author are also great contribution to the success of the work since the author had to learn a lot along the way to be able to develop the application.

At the end of project, it could not be denied that it had certainly brought benefit to both the commissioner and the author. The company now has its expected product with all required features. The author raises high hope that the work will be of good use to the company and welcomes any further development toward the application. The result of this thesis work is a sufficient application that can be tested in the public after three months of development. All the main features should be ready to use. Users are capable of browsing and seeing the chatrooms nearby. The application notification system takes responsibility of notifying user on unread messages if they are absent from the room. The Development plan has changed several times due to the requirement of the company. However, the whole development process is managed to complete in 3 months. The application is fully working product and meet the main requirements.



The most important part of this thesis proves a mobile application can be developed with web technologies. A web developer can reuse their knowledge to shorten to time of development as well as eliminate a need for learning new platform and language.

The author would like to thank the CEO of the case company for coming up with the idea, thus, create the thesis topic which greatly contributed to the author's completion of the author's Bachelor's degree. The author also would like to thank thesis supervisor Mr. Tapani Alakiuttu for his enthusiasm in supporting the project, being available when needed, positively giving valuable comments as well as guiding the author to accomplish the thesis. Special thank also to all participants at the pilot test who were played significant role in improving the application.

## REFERENCES

Jon Duckett, 2011 HTML and CSS: Design and Build Websites, Wrox Publishing.

David Herman, 2012 Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript (Effective Software Development Series) 1<sup>st</sup> edition, Addison-Wesley Publishing

Kate Dreyer. 1990. Mobile Internet Usage Skyrockets in Past 4 Years to Overtake Desktop as Most Used Digital Platform. Cited 07.08.2015.

<https://www.comscore.com/Insights/Blog/Mobile-Internet-Usage-Skyrockets-in-Past-4-Years-to-Overtake-Desktop-as-Most-Used-Digital-Platform>.

"CSS developer guide". Mozilla Developer Network. Cited 2015-010-24.

<https://developer.mozilla.org/en-US/docs/Web/Guide/CSS>.

"Build Your Own Angular". Cited 2015-11-01.

<http://teropa.info/build-your-own-angular>

Three-Way Data Binding with Firebase and Angular. Cited 2015-11-29.

<https://www.firebase.com/blog/2013-10-04-firebase-angular-data-binding.html>.

Ionic documentation. Cited 31.11.2015.

<http://ionicframework.com/docs/>.

Cross-walk documentation. Cited 31.11.2015.

<https://crosswalk-project.org/>.

AngularJS UI-Router. Cited 2.12.2015.

<https://github.com/angular-ui/ui-router/wiki>.

Firebase documentation. Cited 2.12.2015.

<https://www.firebase.com/docs/web/>.

Geofire documentation. Cited 2.12.2014.

<https://github.com/firebase/geofire-js>