Professional Practice Project

Student Name: Ivan McGann

Reservation System

Contents

[Introduction 2](#_Toc512786611)

[System Requirements 2](#_Toc512786612)

[Technology Used and Why 2](#_Toc512786613)

[Ionic 3 3](#_Toc512786614)

[Firebase Realtime Database 3](#_Toc512786615)

[Node Js 4](#_Toc512786616)

[Visual Studio Code 5](#_Toc512786617)

[Architecture of the Solution 5](#_Toc512786618)

[App 6](#_Toc512786619)

[Models 6](#_Toc512786620)

[Pages 7](#_Toc512786621)

[Services 7](#_Toc512786622)

[Design Methodology 7](#_Toc512786623)

[Features of the Implementation 7](#_Toc512786624)

[Firebase 8](#_Toc512786625)

[Angular 8](#_Toc512786626)

[Create 8](#_Toc512786627)

[Read 8](#_Toc512786628)

[Update 8](#_Toc512786629)

[Delete 8](#_Toc512786630)

[Limitations 8](#_Toc512786631)

[Known Bugs 9](#_Toc512786632)

[Recommendations for Future Development 9](#_Toc512786633)

[Conclusions 10](#_Toc512786634)

[Appendix 11](#_Toc512786635)

# Introduction

My name is Ivan McGann; I am a student in the 3rd year of a B.Sc. in Computing and Software Development. This project was created as an appointment reservation system developed as a mobile application. This documentation is provided to my supervisors of the Professional Practice in IT module and can be used as a guide in understanding the finer details of the project. I will be using ionic to do the project which I have developed with before, but it is an ever-evolving platform, so it can be tricky to use. This document will give a summary of my project and what I planned to create under the constraints of time and ability. The project is not based for use by any working business practice in its current format.

The goal of this project was to create a reservation system using the ionic platform. The initial design uses a three-tier architecture which includes the application in conjunction of using firebase to receive and store information entered. The project is a system to allow for example a dentistry office to both store and receive appointments of a one-hour length. Before I began the project, I decided to do some research and decided to focus on several areas including:

* Updating all software required and creating a google firebase to use in the project.
* Develop a better understanding of the ionic platform since I last used it.
* Try to create a log in for dentist and or patients once the project is created.
* Organise a system timetable to book appointments and mark those booked so you cannot double book.

# System Requirements

This application is designed to use on android and iOS devices which I will discuss in the user guide for the Users. I developed the project on HP Notebook, using a Windows 10 Operating System and using an AMD Radeon 5 processor. I have also tested my app on an Android device (Samsung Galaxy S6) using the Nougat operating system. I do not have an iOS device to test the application on.

# Technology Used and Why

I have listed below the technologies I used to create and develop the project in. I chose to use systems and environments that I was familiar with. I have some experience creating Ionic application and using firebase to write and manipulate data.

### Ionic 3



Ionic is an open-source framework for mobile app development. The original version was released in 2013 and built on top of AngularJS and Apache Cordova. The more recent releases, known as Ionic 3 are built on Angular solely. Ionic provides tools and services for developing hybrid mobile apps using Web technologies like CSS, HTML, and typescript. Applications can be built with these web technologies and distributed through native app stores to be installed on devices. Ionic was created by Max Lynch, Ben Sperry, and Adam Bradley of Drifty Co. in 2013.

I am currently using ionic version 3.19.1 (version 3.2 was released during development. I have previously worked on Ionic version 1 in a module for my second year of Software Development. This played a part in my selection as I’ve only developed on the Universal Windows Platform (UWP) the last few months and am not overly familiar with using xaml and C# classes.

### Firebase Realtime Database



The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. When you build cross-platform apps with our iOS, Android, and JavaScript SDKs, all clients share one Realtime Database instance and automatically receive updates with the newest data.

I chose firebase as I felt doing an SQL database could cause issues in how I store the data as I had issues using MongoDB and MySQL with manipulating data in the Data Centric RAD module in the previous semester. To use a firebase to store data written to it you need to change the rules of the database from null to true.

### Node Js



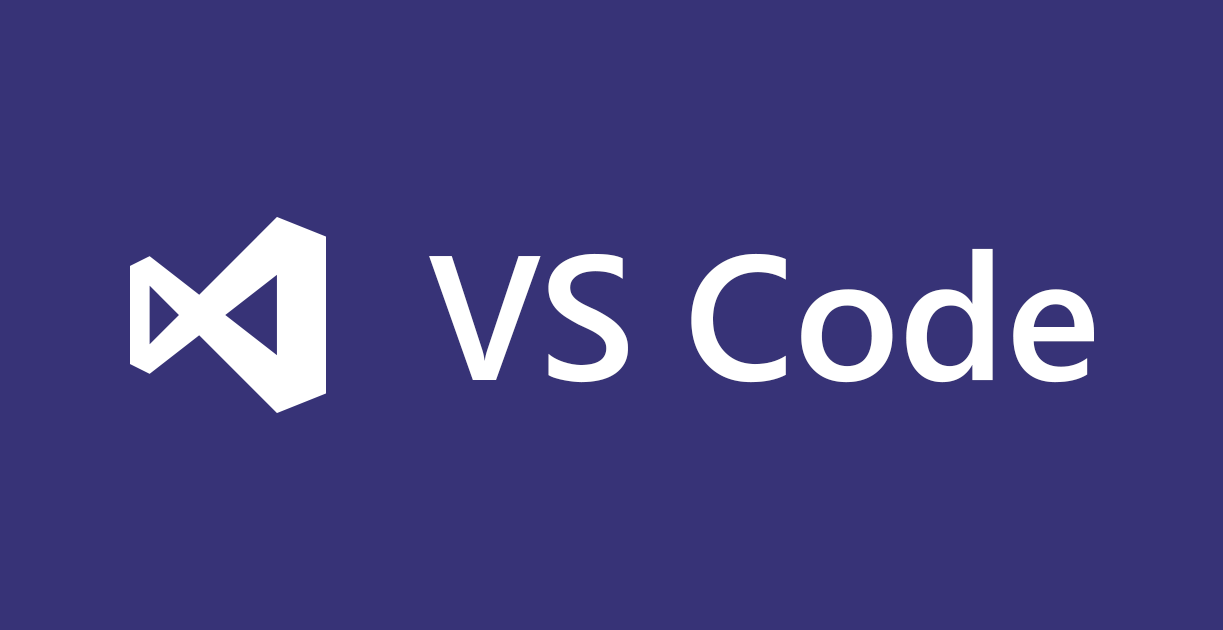
Node.js is an open-source, cross-platform JavaScript run-time environment that executes JavaScript code server-side. Node.js lets developers use JavaScript for server-side scripting, running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js unifies web application development around a single programming language, rather than different languages for server side and client side scripts.

Node Packet Manager (npm) is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry. The registry is accessed via the client, and the available packages can be browsed and searched via the npm website. The package manager and the registry are managed by npm, Inc.

Similar to Ionic and Visual Studio Code I chose Node JS as I am familiar with using the environment. It has a calendar package which I thought would help in the development of my project. I am also familiar with using Async which I used on the project.

<http://www.damirscorner.com/blog/posts/20170127-Angular2TutorialWithAsyncAndAwait.html>

### Visual Studio Code

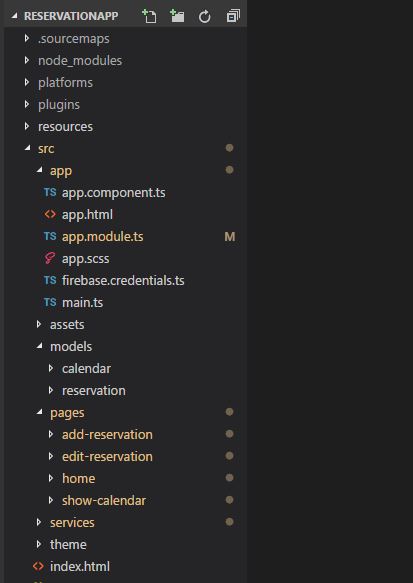


Visual Studio Code is a source code editor developed by Microsoft for Windows, Linux and Mac Operating Systems. It supports several programming languages and sets of features. It includes debugging, embedded Git control, syntax highlighting and refactoring support. It is also customizable, so users can change the editor's theme, keyboard shortcuts, and preferences. It is free and open-source, although the official download is under a proprietary license. Visual Studio Code is based on Electron, a framework which is used to deploy Node.js applications for the desktop running on the Blink layout engine. In the Stack Overflow 2018 Developer Survey, Visual Studio code was ranked the most popular developer environment tool, with 34.9% respondents claiming to use it.

I had the choice to edit the code in notepad++ or VS Code and it was an easy choice. Using VS Code, it is easier to go between folders to edit files and it suggests fixes for any errors such as importing supported and installed plugins and suggests which models to use. Being able to go between the typescript and html files is very useful as I was using those files the most. I had minimal experience of using VS Code for one semester, so it was an easy decision to use it.

# Architecture of the Solution

I decided here to explain the file structure as the architecture. In the next topic I will explain the systems fundamentals and how the application writes and retrieves from the database. Below I will explain what the classes and files contain.



### App

In the source folder (src/app), App.Module.ts works as a way of organising your application. In your declarations you declare any components of your app, I just chose to leave that default (I chose to lazy load the page). After importing the Angular fire plugin at the top of page I initialise it in my imports and create a Firebase\_Config file. This is in the app folder and contains a constant which is how you connect your application to a firebase Realtime database which you access on the homepage of a created firebase (Add Firebase to your Web App). Providers contain your services which allow the app to use the CRUD architecture for the reservation list and Toast which displays messages to the user when they interact with the application.

### Models

Next you have the model’s folder which contain an interface model to define what a reservation will look like, what the variable types are and their names. I also created one for the calendar when trying to search between two date parameters, but I was unable to correctly implement that with the calendar page.

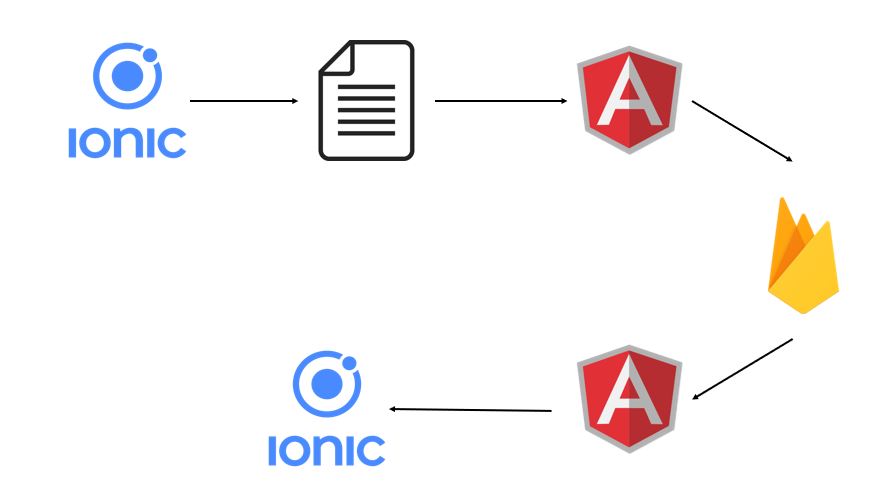
### Pages

Below this in the app/pages folder contains the application main files for the home page, add or edit reservation and show calendar. Each page contains a html, typescript, SCCS and module file. The SCSS works like CSS files as it allows the user put style on their pages, I left this until the end and regrettably used very little styling. The html contains ion- content which is the content of the page this can be labels or using the ngModel to communicate with the models created previously created and binds it to the value on the html page. The .ts or Typescript file creates a version of your model to use in the page. E.g. reservation: Reservation is a new reservation based off the Reservation Model template and you can set the default values to undefined.

### Services

I use these to inject angular fire database into the application using the constructor, you create a reference to tell AngularFire which part of the firebase you want to reference (‘reservation-list’). We then make this reference from the model equal to the type model <Reservation>. This makes sure what is written to the database is a type of reservation.

# Design Methodology



# Features of the Implementation

The key features are the navigation between pages and CRUD implementation to and from the Firebase Realtime Database. Navigation through pages is automatically implemented but the create, read, update and delete obviously are not. All changes to firebase are handled in the reservation-list.services.ts file in the app folder.

### Firebase

First, I created a blank application in ionic 3. Then I installed firebase and angular fire dependencies in app via the command line. The application will connect to Firebase using Firebase.credentials.ts in app folder. This contains a constant FIREBASE\_CONFIG (copy from your Firebase project as seen in the Install-Config-Guide).

I create a button to take me to the form to complete and a button to push the saved data to firebase. It does this in the following way. I create a model Reservation which I will bind to the html and import the model into the typescript file.

### Angular

In services I create a constructor which injects Angular into the application and allows the use of AngularFireDatabase. I create a reference to allow the model <Reservation> to be bound to firebase to write data via a reservation-list model. I then create a function to call this reference to allow it to be displayed to the user.

### Create

Next, I create a method to add a reservation. I import the model and the service to allow access these parameters. I create reservations by filling in the form and pressing a button (using an on click event) that calls a function which uses an Angular method called .push which pushes the reservation model on to firebase.

### Read

I display this using \*ngFor that helps display the list as a string and async retrieves the latest value of the model from firebase. To help this happen use an Observable that gets the list from the database and any changes to it by retrieving its key and values which returns an object/s of reservation/s from the reservation-list.

### Update

I edit a reservation by creating a function which uses the Angular method .update. I use navPush from where the reservation list is displayed to go to the edit-reservation page which displays the selected model and allows you to change the reservation. Like the add reservation it makes changes to the reservation-list using the key and values.

### Delete

Deleting uses an Angular method .remove which calls a function in services. Like the add and edit methods it uses an onClick event contained in a button. The reservation model selected redirects you to the edit reservation page which contains the button to remove.

# Limitations

There are obviously several limitations to the final design. As described in my recommendations for future development I had tried several limitations of the current design. I am missing a search function that correctly organises the appointments in an appropriate manner. I initially tried to display them in a calendar style timetable to show the appointments and that could not display properly. Then I tried to search between two date fields but that did not work correctly either too.

I also wanted to do a login for customers to make and see appointments and maybe a separate login then for the business owner to see what appointments are made and when, also allow them certain privileges not available to the customers. I’ll go into more detail in my conclusion about what I would have done if I were able to do it all over again.

# Known Bugs

There are not many bugs in the project as the design is simplistic in design rather than overly complex as there were areas of the project that were not working as desired that I had to remove. With Ionic if you have blatant errors in the code it will affect any functionality of the app and it won’t run as required from the boot up. The only known bug in this code is occasionally the back button used to navigate between pages won’t appear. In previous versions of ionic you would have to create a button specifically to traverse between pages but in Ionic version 3 these are automatically generated once called in the code.

# Recommendations for Future Development

As seen in the introduction of this document I initially had grander plans for this project but a lack of knowledge and skill lead to me having to slowly pare back on the ambition of the project once I hit a wall in how I would display the data written to firebase. This was a constant issue from the moment I had a CRUD system working on the application.

At first, I would have liked to have integrated a calendar in one of the button links in the top left corner. The calendar would have displayed a week format with a timetable that shows the timetable for the days with separate hour slot during the day that displays the appointment and who is booked into that time. I was unable to implement this as I could not get the calendar to display in the format I wanted to implement. I would get the headers of the calendar but not the underlying hour slots to differentiate the appointments. I’m sure this is possible to do but I was unable to integrate it into my Ionic 3 reservation application. I’m sure it is possible, but I panicked and quickly had to abandon it once I could not get it to work correctly.

I also tried to create a search between two date parameters to find appointments booked between those dates with the use of a search method and a button. I was unable to get this function working and bringing back the dates in a correct order. Both are more organised and aesthetically pleasing display formats than the display I had to end up using. If I had to make any changes to build upon the current display I would prefer to display the appointments in a different format.

If I had time at the end of the project I wanted to create a separate login for a customer who can view appointments and see where there was space to book an appointment. Then as a business you can log in to view their appointments or delete/update appointments.

# Conclusions

I found this project a lot more challenging than I imagined. I felt I struggled t control the concept after my initial outline of what it would be. I had intended to write the appointments to a calendar which would display them in a 5 day week format, but I struggled to implement it correctly. This lead me down a proverbial rabbit hole of concepts that lead to me trying several display formats but would not solve the biggest issue which was the initial limiting core design.

I struggled to deal with the ionic development environment as its one of the more punishing applications to use as any error can lead to the designed apps not functioning on any level. I considered using a MySQL database, but I figured I would struggle to design it to work as effectively as I wanted as I have experienced how complex they can be in previous projects on separate course.

If I could restart the project again I would have changed the foundation of the project keeping firebase but not using the model functionality as I felt it struggled to allow me manipulate date that I used in conjunction with the AngularFireDatabase. There were several things I wanted to implement that I was unable to which I discuss above in the future recommendations section of this document.

# Appendix

Firebase: <https://console.firebase.google.com/u/0/project/fir-project18/overview>

Update npm: <https://stackoverflow.com/questions/18412129/how-can-i-update-npm-on-windows/27464115>

Update Cordova: <https://cordova.apache.org/docs/en/3.1.0/guide/cli/index.html#updating-cordova>

Uninstall Ionic: <https://forum.ionicframework.com/t/uninstall-ionic/96968/3>

Ionic starter: <https://ionicframework.com/getting-started/>

Ionic 3 Firebase: <https://blog.paulhalliday.io/2017/07/06/firebase-cloud-functions-with-ionic-3/>

Fetch server data: <https://www.joshmorony.com/using-http-to-fetch-remote-data-from-a-server-in-ionic-2/>

Ionic 3 Integrate firebase: <https://www.youtube.com/watch?v=LHDfRCyApP4>

Import AngularJS: <https://stackoverflow.com/questions/42714284/how-to-import-angular-httpmodule-to-the-root-ngmodule>

Ionic research: <http://masteringionic.com/blog/2016-12-15-using-php-and-mysql-with-ionic/>

Ionic firebase: <https://www.pluralsight.com/guides/ionic-2-and-firebase>

Ionic Issue: <https://forum.ionicframework.com/t/ionic-start-fails-due-to-missing-ionic-angular-module/116946/10>

Paul Halliday: <https://www.youtube.com/channel/UCYJ9O6X1oFt7YGXpfRwrcWg/videos>

Ionic DateTime: <https://ionicframework.com/docs/api/components/datetime/DateTime/>

Ionic Timepicker: <https://gitlab.tubit.tu-berlin.de/stapps/ionic-timepicker/tree/master/src>

Ionic default time: <https://forum.ionicframework.com/t/datetime-default-to-todays-date/53178/2>

Ionic forum: <https://forum.ionicframework.com>

Ionic calendar: <https://ionicframework.com/docs/native/calendar/>

Firebase LINK: <https://console.firebase.google.com/u/0/project/fir-project18/database/fir-project18/data>

Ionic calendar 2: <https://devdactic.com/ionic-calendar-angular-calendar/>

Github project via command line: <https://help.github.com/articles/adding-an-existing-project-to-github-using-the-command-line/>

Angular cloud: <https://angularfirebase.com/lessons/secure-firebase-cloud-functions/>

Ionic reorder: <https://ionicframework.com/docs/api/components/item/ItemReorder/>

Ionic UI integration: <https://www.djamware.com/post/5a0bb8f780aca75eadc12d6b/build-ionic-3-angular-5-calendar-ui-with-event-integration>

Firebase sorting: <https://stackoverflow.com/questions/41552884/firebase-sort-by-points-depending-on-date>

Ionic order a list: <https://forum.ionicframework.com/t/order-a-list/55389/2>

Angular pipe: <https://stackoverflow.com/questions/42697441/angular-2-pipe-argument-fails/42699338#42699338>

Timestamp order: <https://stackoverflow.com/questions/37883620/how-to-order-by-timestamp-using-firebase?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa>

Child property: <https://stackoverflow.com/questions/33573615/how-to-create-an-index-on-a-child-property-of-an-array-object-in-firebase>

Fetch datetime: <https://forum.ionicframework.com/t/fetching-datetime-from-firebase-database-and-showing-them-in-ionic-datetimepicker/108564/4>

Order list 2: <https://forum.ionicframework.com/t/order-a-list/55389>

Sort date firebase: <https://firebase.google.com/docs/database/web/lists-of-data#sort_data>

Ionic generate pipe: <https://ionicframework.com/docs/cli/generate/>

Firebase indexing: <https://firebase.google.com/docs/database/security/indexing-data>

Firebase duplicates: <https://stackoverflow.com/questions/23177437/preventing-duplicate-records-in-firebase>

Firebase query: <https://firebase.google.com/docs/reference/js/firebase.database.Query>

Firebase database reference: <https://firebase.google.com/docs/reference/js/firebase.database.Reference>