
Cóisir - A Group Collaboration Application Built Using Ionic 3

Dara MacMeanmain

B.Sc.(Hons) in Software Development

APRIL 17, 2018

Final Year Project

Advised by: Daniel Cregg

Department of Computer Science and Applied Physics
Galway-Mayo Institute of Technology (GMIT)



Contents

1	Introduction	5
2	Context	6
2.1	Objectives	6
2.2	GitHub	6
3	Methodology	7
4	Technology Review	9
4.1	Visual Studio Code	9
4.2	TypeScript	9
4.3	JavaScript	10
4.4	Node.js	10
4.4.1	Express	10
4.5	Apache Cordova	11
4.6	Angular	11
4.7	Ionic Framework	11
4.8	Amazon Web Services	12
4.8.1	Amazon EC2	12
4.9	Apache CouchDB	12
4.10	PouchDB	13
4.11	GitHub	13
5	System Design	14
5.1	The Presentation Tier	14
5.1.1	Login	14
5.1.2	Create an Account	14
5.1.3	Home	15
5.2	The Logic Tier	15
5.2.1	Node.js	15
5.2.2	Angular	16

<i>CONTENTS</i>	3
5.3 The Data Tier	16
5.3.1 Amazon Web Services - EC2	16
5.3.2 CouchDB	16
6 System Evaluation	17
7 Conclusion	19
8 Appendix	20

About this project

Abstract The objective of this project is to create a platform that would allow users to create groups to coordinate and collaborate together. The main design is that of a multi-tiered architecture. The front end is a mobile application developed in Angular with the Ionic Framework. Here, the user is presented with the interface allowing them to either create an account with the system or log in using an existing one. The middle tier consists of the Angular back-end with TypeScript and Node.js as the server written in JavaScript. This layer processes the data input by the user and posts it to the database via HTTP. Finally the database consists of CouchDB being hosted on an Amazon Web Services virtual environment in the cloud. Here, the data input by the user is stored for later retrieval.

Authors This project has been developed by Dara MacMeanmain, a fourth year student studying B.Sc (Hons). Software Development in GMIT.

Chapter 1

Introduction

In the years since it's release, particularly in the past decade, Facebook, along with other social media platforms, has become a very large part of our day to day lives. In fact it is quite rare to find anyone that doesn't, at the very least, have an account created on one or more of these websites.

During the early weeks of college at the beginning of first year, one of my fellow classmates created a Facebook group in order to help us keep touch and share any necessary resources or information. Over the course of the following months the group grew and expanded to encompass the whole year and it eventually became our primary form of communication. As a result, we had become reliant on this platform.

After returning to college for my final year, I found myself wondering what I could do for my final year project. It was during this time, a conversation took place which helped me make that decision.

One classmate noted that he was no longer using Facebook, and that the only reason he had remained was for the group functionality. It was becoming apparent that more and more people had been seeking to abandon Facebook altogether. It occurred to me then, that a potential alternative to Facebook might be the idea I was looking for, at least in regards to the group functionality.

Chapter 2

Context

To create this project, having had some experience with the technology previously, I opted to build a mobile application in the form of a MEAN stack using the Ionic Framework. The four components of which being MongoDB, the database, Express, a web application framework for Node.js which handles routing and requests, Angular, the front-end of the application and finally Node.js, the server-side run-time environment.

I chose to modify the structure, however. As I wanted to make a mobile application I chose to include the Ionic Framework on top of Angular for the front-end of the application. The next change was made to the overall architecture. In order for the application to work, I needed to host the database on a remote server. For this, I used the Elastic Cloud Compute service from Amazon Web Services, which allowed me to create a virtual server-side environment.

2.1 Objectives

The primary objective of this project was to provide an alternative group collaboration service that resides outside of the realm of the larger social media platforms. It would be completely independent, thus requiring it's own user authentication service. To achieve this, a database system had to be established. In order for it to allow online user interaction, it would require a remote server to host the database. The end user would be able to connect to this server via the internet and create an account with the system.

2.2 GitHub

<https://github.com/daramacmeanmain/Year4Project>

Chapter 3

Methodology

The methodology behind the development of this project was primarily an Agile one. As such, the project was developed using iterative methods that adapted to any changes in the requirement, or the development process.

At the beginning of the project, I met with my supervisor to discuss my idea. From this meeting it became clear that the architecture behind the project would involve a MEAN stack spread out across three tiers.

These tiers are the presentation, the end product itself that the user interacts with, the database, where user data is stored, and the application, the back-end processing tier that acts as an intermediary between the user and the database. In this case, the database is being hosted on a remote server.

After meeting with my supervisor, I began work by laying the foundation for the server side of the project with the use of Amazon Web Services. Having used this service for projects previously, I felt comfortable utilising it for this application. I chose an Elastic Compute Cloud server, a Windows Server to be specific, and launched the virtual environment. From there, I installed MongoDB on the remote server as this would be the database for the application.

The next step was to create a static IPv4 address in order to connect to the instance, otherwise a new one would be created each time it's restarted, requiring code alteration's each time. The next connection requirement was to create the necessary custom TCP (Transmission Control Protocol) rules to allow an external device to connect to MongoDB on the server.

For the client side of the application, I created an Ionic 3 project and, with the help of a tutorial online, attempted to make a connection to the database on the server. However, I found myself running into difficulties with MongoDB and sought alternative solutions. During this time I began working on another project which involved a remote database and for this project a

classmate recommended CouchDB. After using this database software for a time, I found it to be more user friendly, while also retaining the same capabilities as the alternative I had been using. I then made the decision to remove MongoDB from this application and sought to replace it with CouchDB.

Chapter 4

Technology Review

In this section I will discuss the various technologies that were used in the development of this application.

4.1 Visual Studio Code

Visual Studio Code, or simply VS Code, is an and open source code editor from Microsoft. It features built-in support for JavaScript and TypeScript, with support available for many more, such as Java, Python and Racket[1].

One of the major downfalls of the mainline Visual Studio is that while there are many options for development, from plugins to emulators to real-time GUI displays, it is an extremely intensive program to run. Another significant problem in relation to this is that the installation size, depending on the required tools, can be quite large.

In direct contrast to this, Visual Studio Code is far more lightweight to run, and has a much smaller installation size.

4.2 TypeScript

TypeScript is an open source programming language developed by Microsoft in 2012. It is a superset of JavaScript that adds optional types, classes and modules[2].

TypeScript 1.0 was first released in 2014, nearly two years after its first public release of build 0.8, and became a built-in language to Visual Studio[3].

As TypeScript is superset of JavaScript, the compiler interprets it as such and then emits it as JavaScript, allowing programs to immediately run in a wide range of execution environments[4].

Listing 4.1: TypeScript Example

```
let credentials = {  
    username: this.username,  
    password: this.password  
};
```

4.3 JavaScript

JavaScript is a high level programming language, and is one of the core technologies of the web alongside HTML and CSS.

First released in 1995 alongside NetScape 2.0, JavaScript is a multi-paradigm language, supporting even-driven, functional and object-oriented programming styles[5].

Listing 4.2: JavaScript Example

```
app.use(function(req, res, next) {  
    res.header("Access-Control-Allow-Origin", "*");  
    res.header('Access-Control-Allow-Methods', 'DELETE, PUT');  
    next();  
});
```

4.4 Node.js

Node.js is a JavaScript run-time environment for executing server-side JavaScript code. It was first released in 2009 and allows developers to run server-side scripts to produce dynamic web content before the page is received by the client.

Installed via the npm package manager, it is asynchronous and event-driven, designed to build scalable network applications[6].

4.4.1 Express

This is a web application framework for Node.js. Typically, Express handles the HTTP routing for Node.js when connecting to a server.

Listing 4.3: Express Example

```
var app = express();
app.set('port', process.env.PORT || 3000);
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: false }));
app.use(cors());
```

4.5 Apache Cordova

Cordova is an open-source mobile development framework which allows the use of standard web technologies, HTML, CSS and JavaScript, for cross-platform development. Cordova's plugins can access a device's capabilities such as sensors and network.

The majority of Cordova applications use CSS and HTML5 for rendering and JavaScript for the back-end logic. With HTML5, Cordova can be provided access to hardware such as the accelerometer, camera and GPS.

4.6 Angular

Angular, or Angular 2, is an open source web application platform and is a complete rewrite of the original AngularJS platform. Unlike the original platform, Angular makes use of Microsoft's TypeScript as the back-end language.

NgModules provide a compilation context for components, and collect related code into functional sets. Components define sets of screen elements called views, that Angular can modify according to the program logic. These components use services to provide specific functionality[7].

4.7 Ionic Framework

Ionic is an SDK, developed by Drifty, which is used for building hybrid mobile applications. Originally built on top of AngularJS, its most recent iteration, Ionic 3, is built on top of Angular 2.

Ionic applications are created and developed primarily through the Ionic command line interface and utilises Cordova to build and deploy as a native application. The following commands, respectively, are used to create an

Ionic application using the command line, and then to run the application in a web browser via localhost on port number 8100[8].

Listing 4.4: Ionic Command Line Example

```
ionic start coisir blank
ionic serve
```

4.8 Amazon Web Services

First launched in 2002, Amazon Web Services provides on-demand cloud computing platforms on a paid subscription basis. These platforms range from cloud storage, to virtual machines, to databases. This technology allows developers to have a full cluster of computers at their disposal at all times via the internet.

Amazon's technology is implemented using server farm located all over the world. Fees are based on the usage and hardware requirements[9].

4.8.1 Amazon EC2

Amazon Elastic Compute Cloud provides scalable computing capacity in the AWS cloud. EC2 provides virtual environments, called instances, with various options for operating systems, processing power and storage depending on the user's needs[10].

4.9 Apache CouchDB

CouchDB is an open source database that uses a document-oriented NoSQL architecture[11]. A NoSQL database differs from MySQL in that it's non-relational thus providing more flexible methods of retrieving data[12]. In CouchDB, data is stored in the form of JSON (JavaScript Object Notation), which stores data in the form of objects made up of name/value pairs[13].

Listing 4.5: CouchDB JSON Example

```
{
  "_id": "17b0be69-0d38-4786-a8f4-d27038994ef8",
  "_rev": "1-ef5895230b614215974f4b32c99092e8",
  "title": "new"
}
```

4.10 PouchDB

PouchDB is an open-source JavaScript database designed to interact with CouchDB on the client side. It enables an application to store data locally before syncing with CouchDB on a server

Listing 4.6: PouchDB TypeScript Initialisation

```
this.db = new PouchDB('coisir');
```

4.11 GitHub

GitHub is a web-based hosting service using git, a version-control system for tracking changes in files. This allows developers to collaborate and coordinate work among multiple people[14].

Some notable examples of git commands are as follows.

- git init - Create a new repository
- git add - Add the file contents to the index
- git commit - Record changes to the repository
- git push - Update the repository with the saved changes

Chapter 5

System Design

The overall architecture behind the application consists of three tiers: The presentation tier, the logic tier and the database.

5.1 The Presentation Tier

This layer describes the front end of the application, in other words the part of the application that the user interacts with. It is comprised primarily of Ionic 3, a framework built on top of Angular which uses HTML code to display the user interface.

5.1.1 Login

This page give the user the ability to log into their account if they have registered with the system. To do so they would enter their chosen user name and password before pressing the "Login" button, which would then navigate them to the Home page, assuming the user's credentials are correct. However, if the user has not registered with the system, they can do so by selecting the "Create an Account" which will bring them to the relevant page.

5.1.2 Create an Account

In this page, the user is prompted with a form that will allow them to register with the system. This forms consists of a name, a username, an e-mail address, a field for the user to enter their desired password and a field to confirm said password. Once entered, the user can then select "Register" to have their account created with the system.

5.1.3 Home

Once logged in, the user can then interact with the Home screen. If there is data available to be displayed, it will appear here, consisting of a post along with the username of the person that posted it. The user then can add a post of their own to the database by clicking the "Create Post" option in the navigation bar at the top of the screen, or they can log out of the application by selecting the "Logout" button.

Create Post

With this selected, the user is prompted with an alert that allows them to enter the body of the post they wish to send to the database. Once the user has entered their desired text, they can select "Save" to post it, alternatively they can return from the prompt by selecting "Cancel".

Logout

This button will end the session for the user and return them to the Login screen.

5.2 The Logic Tier

This tier contains the logic side of the application, other-wise referred to as the back-end, the purpose of which being to bridge the presentation layer and the data layer together. The technology behind this is primarily Node.js with JavaScript, and Angular with TypeScript.

5.2.1 Node.js

This is used to run the server-side code, written in a JavaScript file, that will interact with the database. In this file the routing configuration is set up using Express, which allows the application to make HTTP requests to the database on the remote server.

This file also initialises the package for a user authentication service called SuperLogin. It contains the details of server and the IPv4 address for which it can be accessed, the definitions for the user databases, and a security system to lock a user out after too many failed attempts.

5.2.2 Angular

This handles the core back-end of the application using TypeScript files that interact with both the front-end HTML pages and the database on the server.

The TypeScript files process the input from the user and send them to the database via the Node.js server file. Likewise, data is retrieved back to the application in the same fashion, processed by TypeScript and displayed to the HTML page.

5.3 The Data Tier

This tier contains the database side of the application, in which all of the data that the user inputs is stored. For the purpose of this project, this database is hosted on a remote cloud server using Amazon Web Services, to accommodate user connectivity via the internet. The database in question is a document-oriented NoSQL database called CouchDB.

5.3.1 Amazon Web Services - EC2

The Elastic Cloud Compute from Amazon Web Services acts as the remote server for the project, and is the host for the database. CouchDB is installed on the server and is accessed using the given IPv4 address (13.58.176.103) via the port number 5984.

5.3.2 CouchDB

This database software stores the user data for the project, which includes posts that have been saved, and account data such as usernames and hashed versions of passwords for security purposes. This data is stored in documents in the form of JSON (JavaScript Object Notation), which can then be sent via HTTP back to client.

Chapter 6

System Evaluation

The main objective behind this project was to provide an alternative platform for group collaboration outside of the more mainstream social media platforms such as Facebook. In order to achieve this, the application would require a method of allowing users to create a group, using a group identifier, and invite other members into that group to allow communication.

The overall outline of this would be as follows:

- Allow a user to register an account with the system
- Allow the user to use that account to log into the system.
- Allow the user to create and send a post to the database, and also retrieve other posts from the database.
- Separate posts by their user ID thus showing the usernames belonging to the displayed posts.
- Allow the user to create a group and assign the group an ID.
- Allow the user to invite other members to join by sharing the group ID with them
- Show posts only to user that resides within the group.
- Allow users to upload media files via the post option to display in the application.

As of writing this, much of the later objectives have not been met. The most recent successful objective was that of separating the user submitted posts based on the user ID. As such, the application did not reach the target objective of allowing group creation, and subsequently, allowing group posting. This could potentially be added to the application at a later date.

Testing

While the application works in the browser via the "ionic serve" command, running tests on an Android device has lead me to encounter problems with online connectivity.

Chapter 7

Conclusion

To summarise, the objectives behind this project were to establish a platform to enable group collaboration outside of the usual means, such as Facebook groups.

Unfortunately, many of these objectives were not met in time due to setbacks in the project, such as the decision to switch to a new database system after Christmas, as well as difficulties implementing the ideas behind the group creation aspect of the project.

Of the current objectives, those that have been successfully implemented are as follows:

- An account system to allow the user to register online
- A login system for the user to access the system via the created account
- The ability to create posts and retrieve them from the database on the remote server
- Initial attempts at user separation with the use of adding the user ID with the post when sending it to the database

For future improvements, I'd like to implement the original objectives I set out at the beginning of the project.

Chapter 8

Appendix

GitHub Repository

- <https://github.com/daramacmeanmain/year4project>

Bibliography

- [1] “Documentation for visual studio code, <https://code.visualstudio.com/docs>.”
- [2] “Typescript - wikipedia, <https://en.wikipedia.org/wiki/typescript>.”
- [3] “Announcing typescript 1.0, <https://blogs.msdn.microsoft.com/typescript/2014/04/02/announcing-typescript-1-0/>.”
- [4] G. Bierman, M. Abadi, and M. Torgersen, “Understanding typescript,” pp. 257–281, 2014.
- [5] “Javascript - wikipedia, <https://en.wikipedia.org/wiki/javascript>.”
- [6] “Node.js, <https://nodejs.org/en/>.”
- [7] “Angular - architecture overview, <https://angular.io/guide/architecture>.”
- [8] “Installing ionic, <https://ionicframework.com/docs/intro/installation/>.”
- [9] “Amazon web services - wikipedia, https://en.wikipedia.org/wiki/amazon_web_services.”
- [10] “What is amazon ec2? - amazon elastic compute cloud.”
- [11] “Couchdb wikipedia, <https://en.wikipedia.org/wiki/couchdb>.”
- [12] “Nosql - wikipedia, <https://en.wikipedia.org/wiki/nosql>.”
- [13] “Json, <https://www.json.org/>.”
- [14] “Git - reference, <https://git-scm.com/docs>.”