
Food Ordering System for the GMIT Catering Company

Ronan Connolly

Vladislav Marisevs

B.Sc.(Hons) in Software Development

APRIL 13, 2016

Final Year Project

Advised by: Dr John Healy

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



Contents

1	Introduction	7
2	Context	9
2.1	Web App	9
2.2	Mobile App	9
2.2.1	Cross Platform Frameworks	10
2.2.2	Tooling	10
2.3	Push Server	11
3	Methodology	12
4	Technology Review	14
4.1	Web App	14
4.1.1	PHP	14
4.1.2	Zend Framework	14
4.2	Mobile App	16
4.2.1	Ionic Framework	16
4.2.2	Tools	16
4.2.3	Alternatives	16
4.3	Push Server	16
4.3.1	CouchDB/Cloudant	16
4.3.2	NodeJS	16
4.3.3	ExpressJS	16
4.4	REST Architecture	16
4.4.1	HTTP Requests	16
4.4.2	Custom API	16
4.5	JSON	16
5	System Design	17
6	System Evaluation	18

<i>CONTENTS</i>	3
-----------------	---

7 Conclusion	19
---------------------	-----------

About this project

Abstract This project sets out to create a food ordering system for a local company. The systems primary components are a mobile application that the user interacts with and a web application that the staff interact with.

The need for such a system stems from two problems, firstly the issue of rush hour times during business hours where there is a vast number of customers to service, and secondly to bring more presence and promotion to the business, as they are finding it hard to reach out to their current customer base and would be customers.

We aim to solve the first problem by having a system in which customers can pre-order sandwiches and other products via a mobile application. Users will be able to top up their account, order products, pick a collection time, and view their balance, products and past order history.

The second problem will be solved by implementing push notifications into the application so that the company can let customers know about menus, events and various other updates. Another way to increase promotion and presence is by having various information about the company on the application; for instance: opening times, contact details and general information.

All of the information on the web application can be updated; this includes the menus, opening times, user and staff details, and much more. This information is reflected in the web application. Interactions from within the mobile application including: topping up, logging in, registration and ordering go through the web application.

We plan to create a cohesive, thoughtfully designed, robust system that solves these two problems.

Authors This project was created by two fourth year software development students: Ronan Connolly & Vladislav Marisevs, as part of our Bachelors of Science honours degree in Software Development.

Ronan was in charge of creating all aspects of the user facing mobile app. Vladislav was in charge of creating all aspects of the staff facing web app.

We spent most of our shared time coming up with the overall architecture we would implement, and an interface to be used between mobile and web app for transfer of data.

Acknowledgements We would like to acknowledge and thank our supervisor Dr John Healy for all the time and effort he has put into helping us throughout this project, he gave us a good structure and set milestones for us in order to keep on top of things.

We'd also like to thank the GMIT Catering Company staff for the time spent meeting with us in order to continuously improve and adapt the project.

Chapter 1

Introduction

We set out to create a food ordering system for the GMIT Catering Company (known henceforth as the company). The basic structure is a mobile application (henceforth known as mobile app) for the Android and iOS systems that the user interacts with, and a server (henceforth known as web app) that the staff can log into in order to view transactions, user details and to update the mobile app.

The reason such a system is needed is that queues during peak times tend to be enormous and currently it's hard to service all the customers.

Another reason is to encourage customers to get into a habit of repeat ordering, if it is an easy process then it should increase purchases.

Lastly, the company wants to increase presence and promotion in the college, in order to achieve this end we have implemented push notifications where staff can send a notification to all users. On top of this the mobile application itself serves as a promotional device, containing details of various aspects of the company.

The components contained within the mobile app include pages for login, registration, about the company and user details. There is also a way to top up and order sandwiches. A huge emphasis is put on design for this project, using the company's colour theme and creating a nice icon. This mobile app was created using the Ionic Framework which is programmed primarily using the AngularJS framework.

The components contained within the web app include many pages such as the login system, orders, stock, user details, vouchers (for adding credit to your account), settings (collection and opening times) and accounts (staff) pages. This web app was created in PHP using Zend Framework 2. Most of the information on the web app is reflected on the mobile app.

The two applications talk to each other via JSON over HTTP Get and

Post requests.

We set out to create a well thought out, carefully designed, robust food ordering system using modern technologies. This project could be extended in the future to be used

We used an agile structure where we had certain components we needed complete by specific dates. We had various meetings each month with our supervisor and several members of the company.

Chapter 2

Context

- Our project consists of creating a food ordering system for the GMIT Catering Company.
- Our basic objectices were to create a mobile and web application to deal with the above item.

Below we will:

- list what each chapter contains below.
- list out the various elements in our Github repositories below.
- explain the various components of our project.

2.1 Web App

2.2 Mobile App

The mobile application is created using bleeding edge technologies such as the Ionic framework, which utilizes the AngularMVW framework, which in turn is programmed using JavaScript. HTML and CSS were also heavily utilized.

Initially I had no idea about JavaScript, web development or cross platform development. I spent much time studying JavaScript, Angular, Ionic, and the MEAN Stack (MongoDB, ExpressJS, AngularJS, and NodeJS). I then spent a lot fo time trying out various cross platform development frameworks.

2.2.1 Cross Platform Frameworks

I first tried out Cordova, which I found had the capabilities to do pretty much anything, but the community and framework is very sparse, it's hard to get anything up and running, and the stylings are horrid. [?]

Next I tried out JQeuryMobile which had better styling but again I came across many issues. [?]

Finally I came across the Ionic Framework which uses Cordova underneath. This framework has a huge community of developers, it has the support of Google, Microsoft and many other large companies. They closely work with the Angular team at Google and the TypeScript team at Microsoft. They have a 24/7 group chat system set up with various sub rooms. They have amazing documentation, regular blog posts, quick response to questions on the blog, forums and chat. With Ionic you get:

- All the capabilities of Cordova, which allows you to access the Mobiles native APIs easily
- A slick native UI experience. The app changes design depending on the platform it's running on
- Extensive tooling. Starting an app, templates, app store image creation, logo creator, etc
- Rapid development cycle, there are constant updates (which can cause issues, but is usually great)

[?]

2.2.2 Tooling

Once I decided on using the Ionic framework I spent my time completing various JavaScript, Angular and Ionic tutorials. This was a steep learning curve as there is so much tooling for all the JavaScript frameworks. I installed NodeJS in order to use NPM (Node Package Manager) in order install Ionic.

Then Ionic came with it's own tools for various tasks, including SASS for programmatic CSS, Bower (Like NPM or Maven) for adding in new components, Grunt for running tasks (like Ant) and some others like Gulp (similar again to NPM).

Each of these tools takes time to learn. I read documentation and completed at least one tutorial for each.

2.3 Push Server

In order to facilitate push notifications I needed to get their device token and save it in a database. I created a controller in the mobile application, once the app starts it gets the device token along with some other information and sends it to the push server.

The push server is always listening for incoming requests, once it receives one it evaluates it, adds a timestamp and saves the user object (JSON) to a CouchDB server (Using IBM's Cloudant Web Server).

The push server is a MEAN Stack, Yeoman scaffolded project that uses NodeJS as the environment, Express as the Web Framework and Angular as the MVC (Model View Controller) framework in order to build the web application.

The reasons we chose to have the push notification server separate is that:

- We did not realise we needed to save the device tokens initially and to implement this new table into the SQL schema is a big effort
- We wanted to try out a MEAN Stack application
- We felt there was no big disadvantage on having the push server separate
- The mobile app is so tightly integrated with the push server (everything is written in JavaScript), that it made creating the web application extremely simple
- The GMIT Catering company may assign the task of push notifications to somebody that they do not wish to have access to the food ordering system, such as a social media expert.

Users can:

- Login/Logout via username and password
- View how many devices are registered for push notifications
- View previously sent push notifications
- Send new push notifications, and see if they were sent successfully.

Chapter 3

Methodology

About one to two Pages

Describe the way you went about your project:

- We used an Agile / incremental and iterative approach to development.
- There were plenty of meeting and planning with our supervisor and the companies manager.
- We did user testing on the app, and constantly made sure all the routes were working.
- We designed a RESTful style route interface. This let us work on our own components without the web and mobile app waiting for each other.
- We used Github during the development process, tracking tasks via the Github Issue tracker.
- We tested out various technologies at each step of the way and discussed why we were choosing each one.

Check out the nice graphs in Figure 3.2, and the nice diagram in Figure ??.

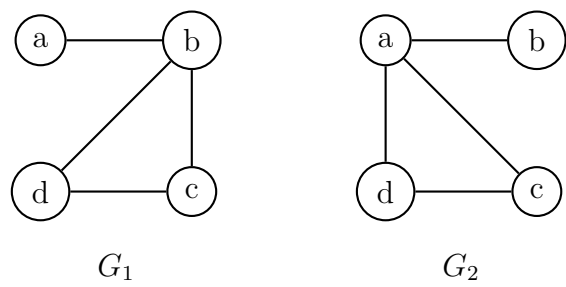


Figure 3.1: Nice pictures

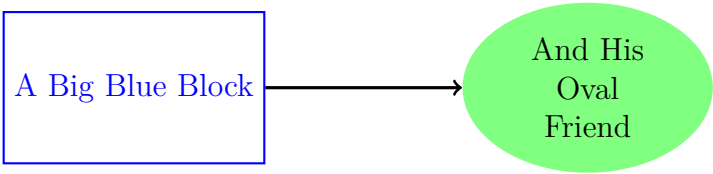


Figure 3.2: Nice pictures

Chapter 4

Technology Review

About seven to ten pages.

- Describe each of the technologies you used at a conceptual level. Standards, Database Model (e.g. MongoDB, CouchDB), XML, WSDL, JSON, JAXP.
- Use references (IEEE format, e.g. [1]), Books, Papers, URLs (timestamp) – sources should be authoritative.

4.1 Web App

I decided to use PHP since I was already proficient at it and it works really well as a web app because....

4.1.1 PHP

4.1.2 Zend Framework

4.2 Mobile App

4.2.1 Ionic Framework

HTML/CSS

JavaScript

AngularJS

4.2.2 Tools

Yeoman

Grunt

Jasmine

Jade

Bower

NPM

BASH

4.2.3 Alternatives

Cordova

PhoneGap

JQueryMobile

Xamarin

4.3 Push Server

4.3.1 CouchDB/Cloudant

4.3.2 NodeJS

4.3.3 ExpressJS

4.4 REST Architecture

4.4.1 HTTP Requests

4.4.2 Custom API

4.5 JSON

Chapter 5

System Design

As many pages as needed.

- Architecture, UML etc. An overview of the different components of the system. Diagrams etc... Screen shots etc.

Column 1	Column 2
Rows 2.1	Row 2.2

Table 5.1: A table.

Chapter 6

System Evaluation

As many pages as needed.

- Prove that your software is robust. How? Testing etc.
- Use performance benchmarks (space and time) if algorithmic.
- Measure the outcomes / outputs of your system / software against the objectives from the Introduction.
- Highlight any limitations or opportunities in your approach or technologies used.

Chapter 7

Conclusion

About three pages.

- Briefly summarise your context and ob-jectives (a few lines).
- Highlight your findings from the evalua-tion section / chapter and any opportuni-ties identified.