

Institiúid Teicneolaíochta Cheatharlach



At the Heart of South Leinster

# 4<sup>th</sup> Year Final Project – Final Report

BSc (Honours) Software Development

**Name:** Alex Matthews

**Student ID:** C00208942

**Supervisor:** Dr. Oisín Cawley

**Date:** 12/04/2019

## Contents

Acknowledgements.....	3
Introduction .....	4
Project Description .....	4
Screenshots of the Application .....	6
Food Diary – Main Screen .....	6
Food Diary – Context Menu .....	7
Food Diary – Calendar.....	8
Food Diary – Add Entry .....	9
Food Diary – Edit Entry .....	10
Food Diary – View Entry.....	11
Recipes – Main Screen .....	12
Recipes – Context Menu .....	13
Recipes – Create Recipe.....	14
Recipes – Edit Recipe .....	15
Recipes – View Recipe .....	16
Conformance to Specification and Design.....	18
Description of Learning.....	19
Personal .....	19
Time Management.....	19
Communication.....	19
Teamwork .....	19
Technical .....	20
Python & MySQL.....	20
React JS & JavaScript.....	20
Cascading Style Sheets.....	20
Project Review .....	20
What Went Wrong.....	21
What Went Right .....	22
Future Improvements .....	24
Conclusion.....	24

## Acknowledgements

I would like to thank my classmates and friends for keeping me sane during this whole process. I'd like to thank the whole Erasmus+ team, especially Daire O'Broin and Joseph Kehoe. Finally, I would like to thank my supervisor, Oisín Cawley, for his guidance and help throughout the project.

## Introduction

This final report will document the process of developing a food and recipes diary web application for the Erasmus+ project for people with coeliac disease and for the final year Software Develop course at Institute of Technology, Carlow. The Erasmus+ project is a two-year project with the aim of increasing gluten free diet compliance rates among teenagers with coeliac disease.

The project team has members from Ireland, Amsterdam, Austria, and Spain and it is led by IT Carlow. The team is cross discipline with students from software development, games design, UX/UI design, and health courses. The software development students on the project are each building one part of the application each and those parts will be integrated together into a final product. The four parts are a home gluten test reader, educational quiz, symptom tracker, and of course, food diary and recipes.

This report will first go through a detailed project description which will outline the functionalities of the system and will show some screenshots of the application. The report will then outline how the final system conformed to the original design outlined in the functional specification and design document. There will then be a description of what was learned during the course of the project, and finally there will be a project review which will outline what went well and what didn't and what improvements could have been implemented if I had more time.

## Project Description

This project was built using React JS, CSS, Python, Flask, and MySQL. React JS is a JavaScript framework for building user interfaces. React is component based and makes it easy to break the application into separate, reusable components. CSS was used to style the application and was the source of a lot of problems when I couldn't make something look the way I wanted it to look, but I will discuss that later in this document. Python and Flask were used to build the API which interfaces with the MySQL database and serves data to the React front-end.

This project consists of two distinct parts: food diary and recipes. The food diary allows users to keep track of their diet by adding a new diary entry whenever they eat something. Each entry is added to a particular date categorized under a specific meal. These meal categories are breakfast, lunch, dinner, snacks, and drinks. Entries are stored for future reference. Entries can also be edited and deleted. In order to create an entry, the user must supply a photo, a title, an optional description, the meal, and the date/time. Although the meal and the date/time are automatically generated.

The main idea behind the food diary is that it would link be linked to another section of the application: The Gluten Immunogenic Peptides (GIP) home test reader, which is being developed by another development student. Making a correlation between the food diary and the GIP test reader will enable users to see what they ate around the time that they tested positive for gluten. This will teach users about what foods or restaurants to stay away from due to the risk of gluten ingestion. The symptom tracker will also be integrated with the food diary so that users can see a correlation between their symptoms, what they ate, and their test results. This is how the final application will provide value to the end user. This

functionality is out of the scope of this project, but it could be implemented during the second year of the project.

The other section of the application is for gluten free recipes. It allows users to create and store recipes in the “My Recipes” section. To create a recipe, the user must supply a photo, a title, an optional description, preparation time, the required ingredients, and the recipe steps. It also allows users to share recipes with the rest of the community. When a user shares one of their personal recipes it can then be seen by all users of the application in the “Community Recipes” section. Users can also search a list of recipes using the provided search bar. The list is filtered as the user types. The goal of the recipes section is to give users more choice of things to eat as that was one of the main complaints when the health students on the project interviewed coeliac patients during the requirements gathering phase.

## Screenshots of the Application

### Food Diary – Main Screen

This is the screen that the user sees when they first open the application. It is the food diary part of the application and from here, users can add, edit, and delete entries. They can also change the date. The screenshot on the left shows what happens when there is not entries to display. This screen also displays a loading animation as the list is loading.

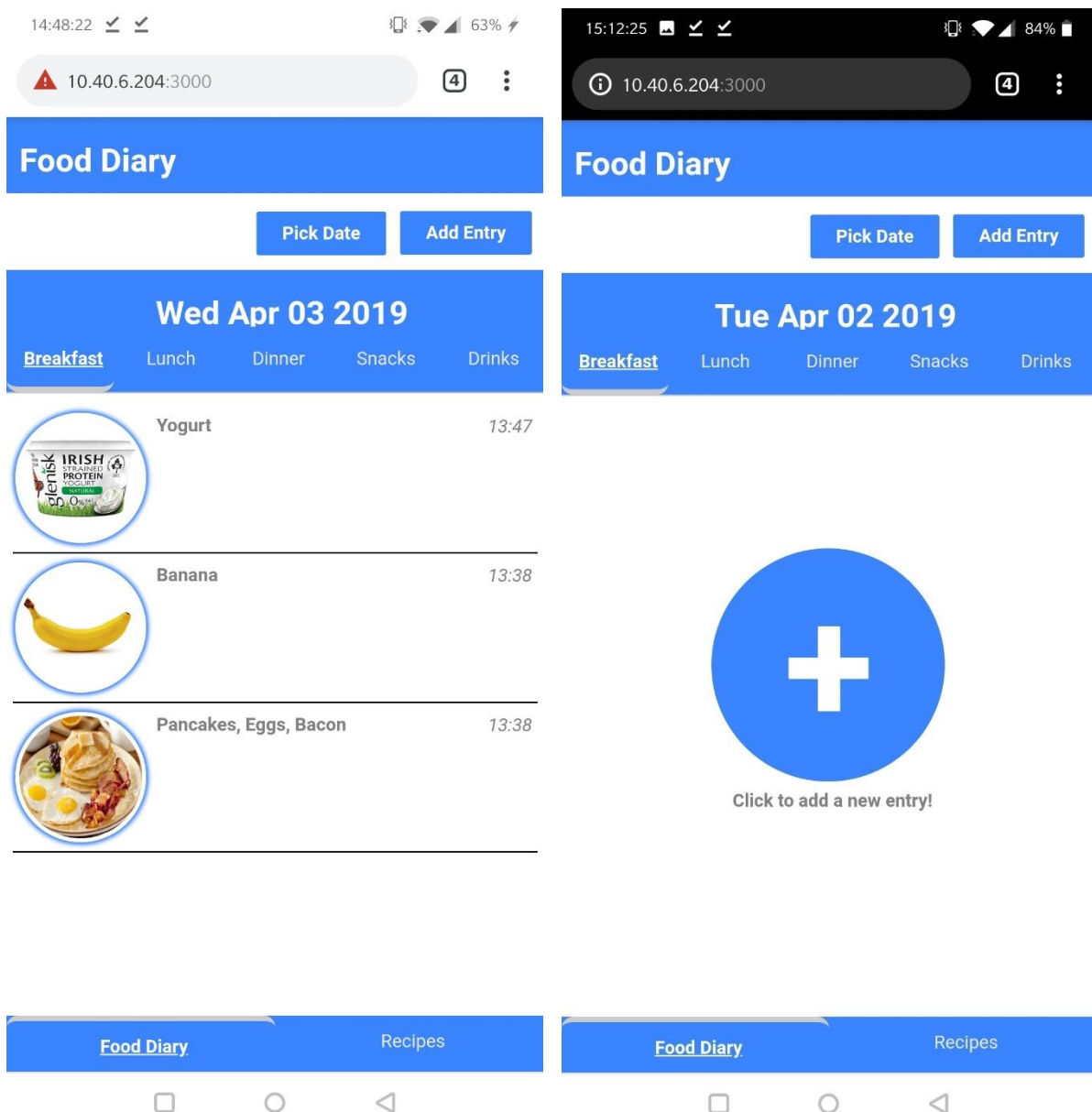


Figure 1 – Food Diary

## Food Diary – Context Menu

This is the context menu that appears when the user presses a food diary entry from the list. The menu has options for viewing, editing and deleting.

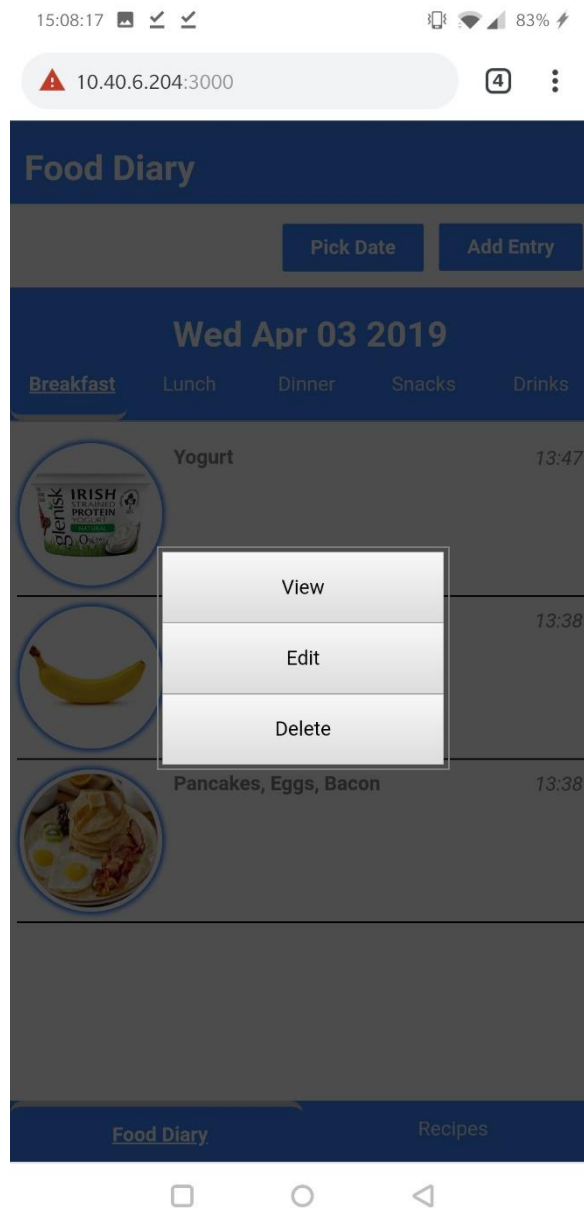
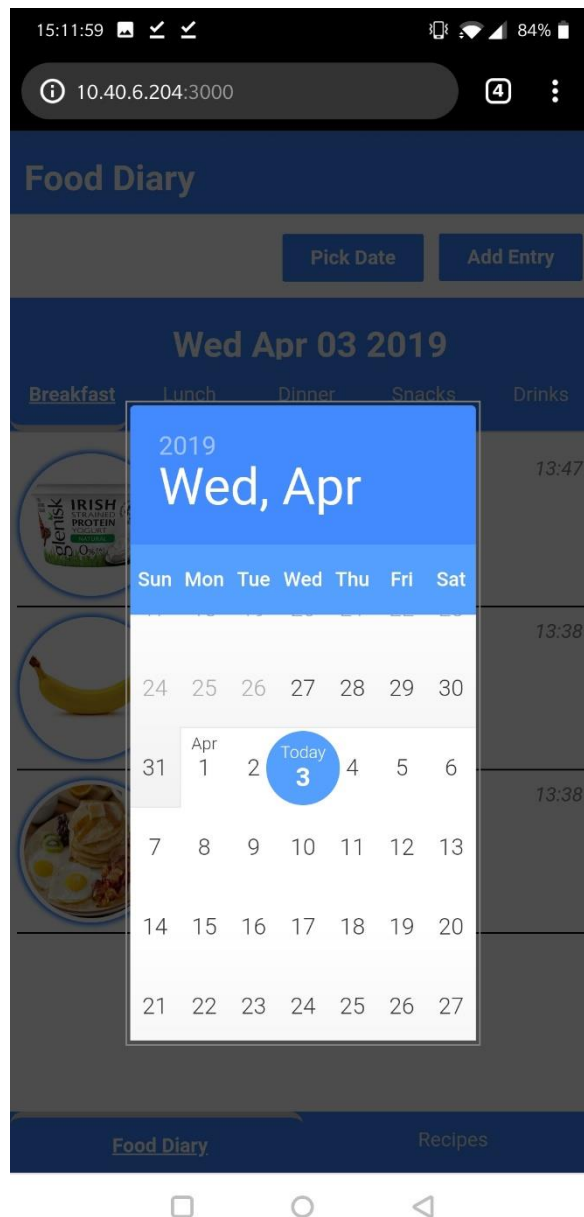


Figure 2 – Food Diary Context Menu

## Food Diary – Calendar

When the user wishes to change the date of the food diary and they press the calendar button, they are presented with a simple calendar which allows them to select the date.



**Figure 3 – Food Diary Calendar**



## Food Diary – Add Entry

This is the screen the user will see when they click the add entry button. It consists of a quick add carousel which scrolls horizontally. It displays recently added food diary entries for whichever meal is chosen and the user can click on them to quickly add them to the diary. The user can also choose to add a picture from their gallery and fill in the form if they wish to add a new entry.

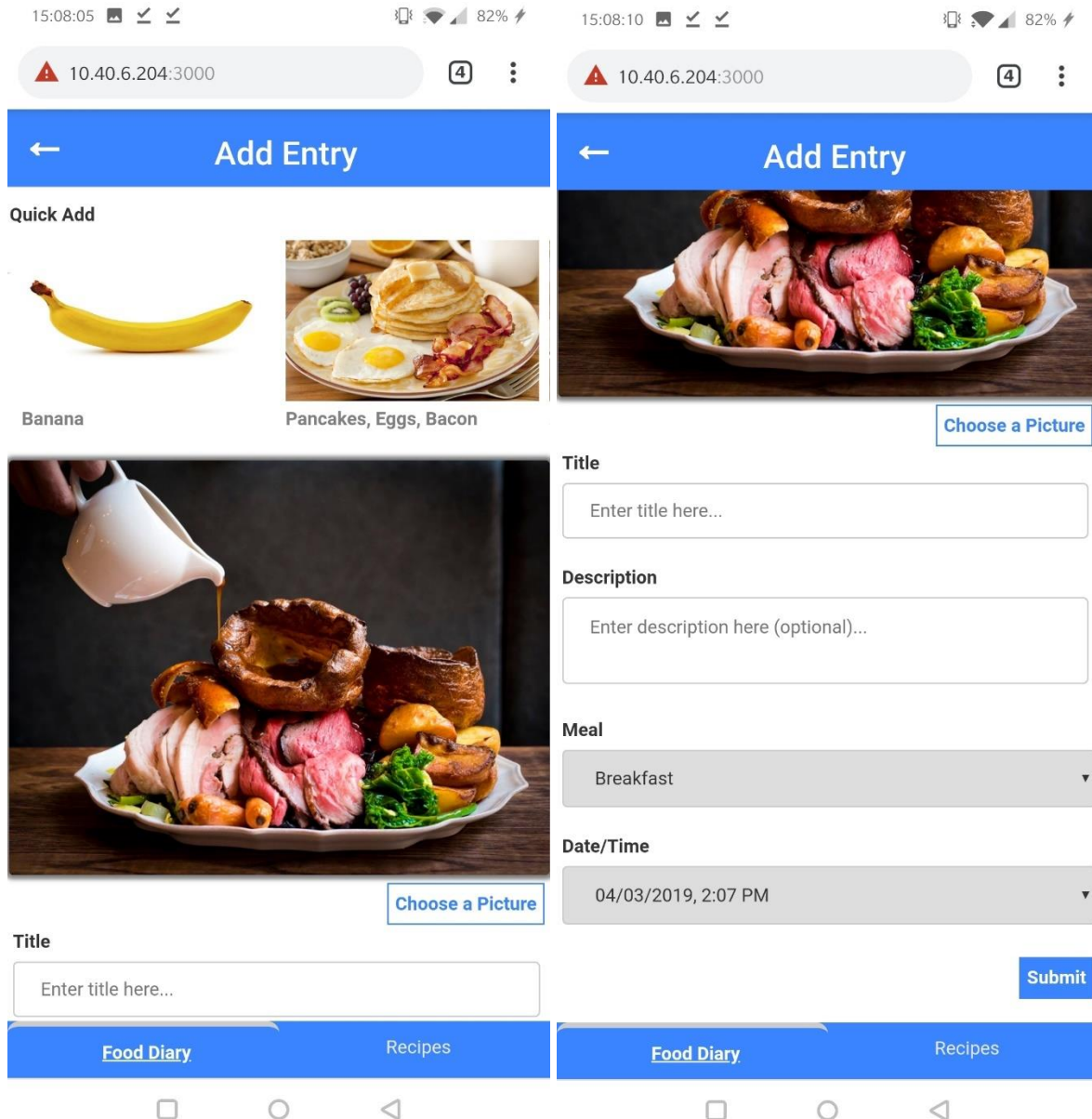


Figure 4 – Add Entry

## Food Diary – Edit Entry

This is the edit entry screen. It's very similar to the add entry screen as it is the exact same form. From here, the user can update an already existing diary entry.

15:08:25 10.40.6.204:3000 83%

### Edit Entry

Choose a New Picture

**Title**

**Description**

**Meal**

Breakfast

Food Diary Recipes

Figure 5– Edit Entry

## Food Diary – View Entry

This screen displays the information of a particular food diary entry.



**Figure 6 – View Entry**

## Recipes – Main Screen

This is the first screen the user sees when they open the recipes tab. From here, the user can choose to create, edit, delete, or search recipes. They can also access the community recipes from here. Community recipes displays any recipes that any user has shared with the community. This screen also displays a loading animation as the list is loading.

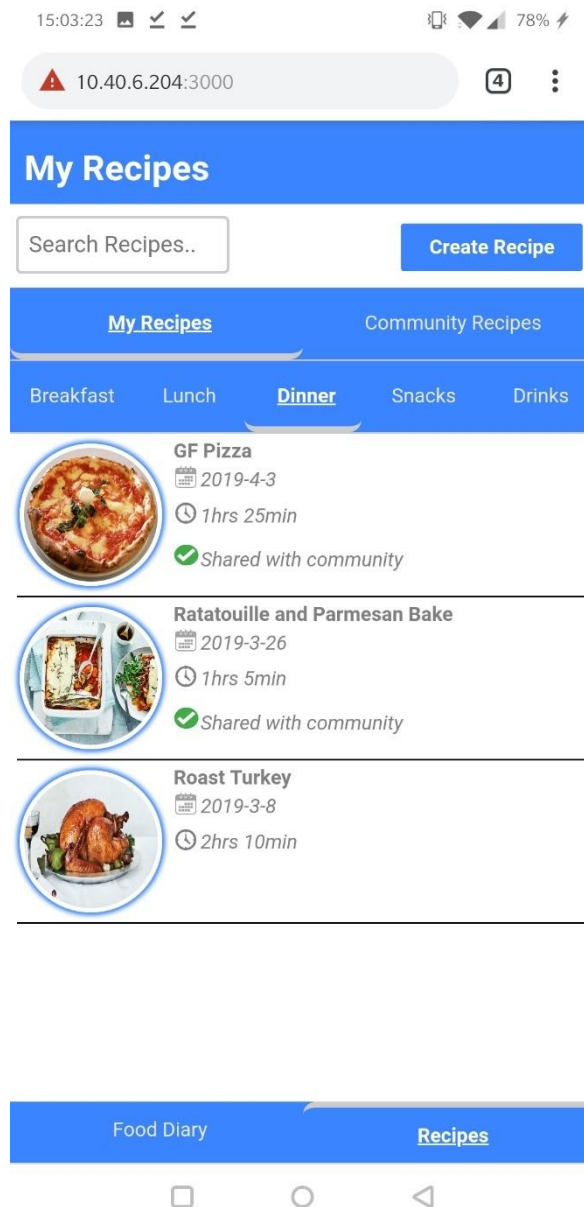
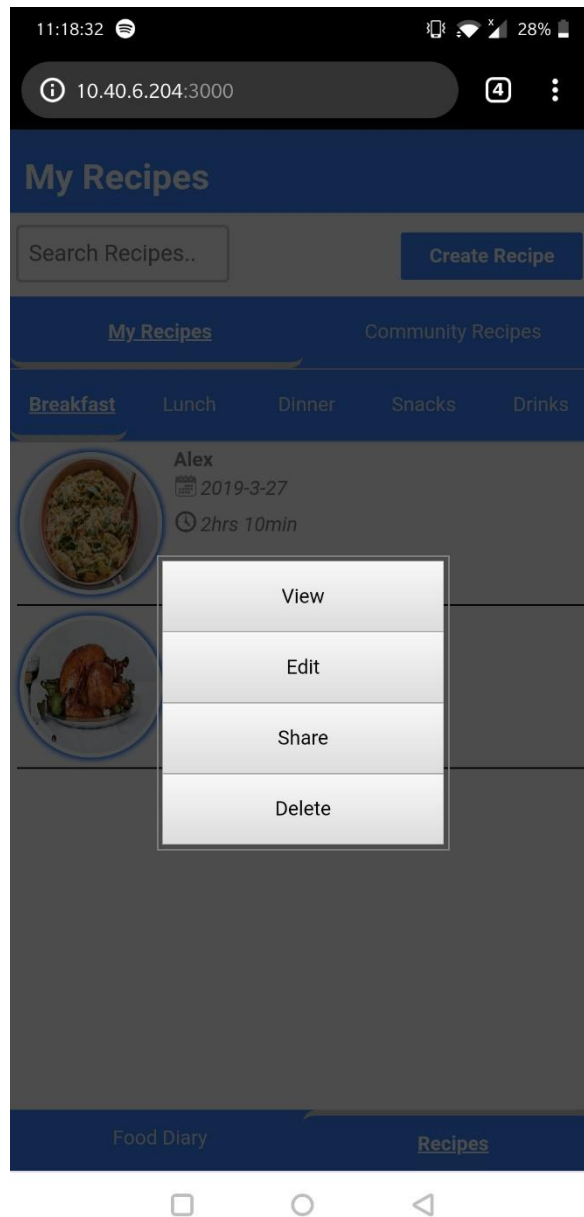


Figure 7 – Recipes Main Screen

## Recipes – Context Menu

This is what the user sees when they press a recipe from the list in the “My recipes” section.



**Figure 8 – Recipes Context Menu**

## Recipes – Create Recipe

This is the screen where users create a new recipe. They just need to provide a photo, title, description, preparation time, how many the recipe serves, the ingredients of the recipe, and the recipe steps.

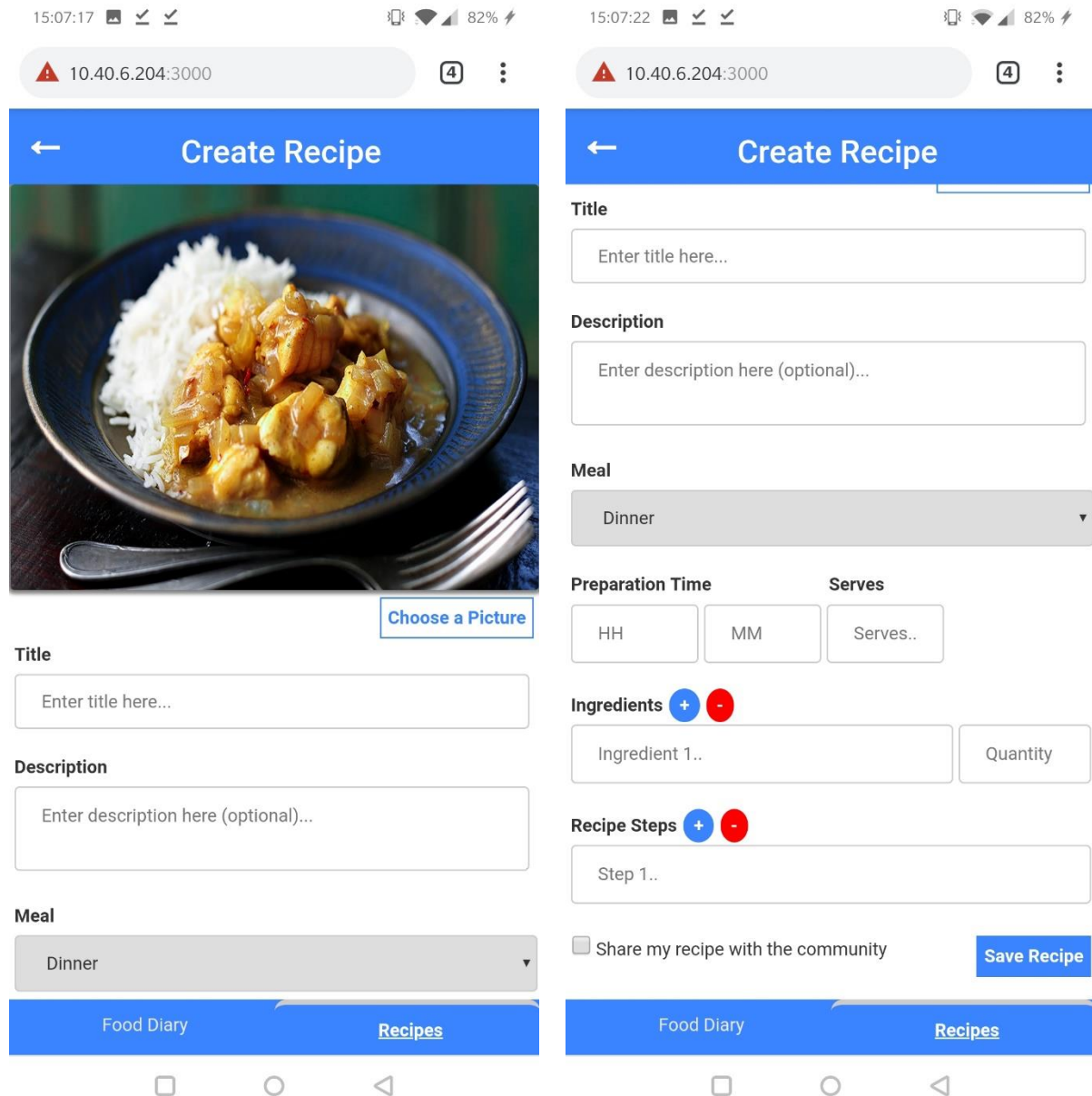


Figure 9 – Create Recipe Screen

## Recipes – Edit Recipe

This is the edit recipe screen. It's very similar to the create recipe screen as it is the exact same form. From here, the user can update an already existing recipe.

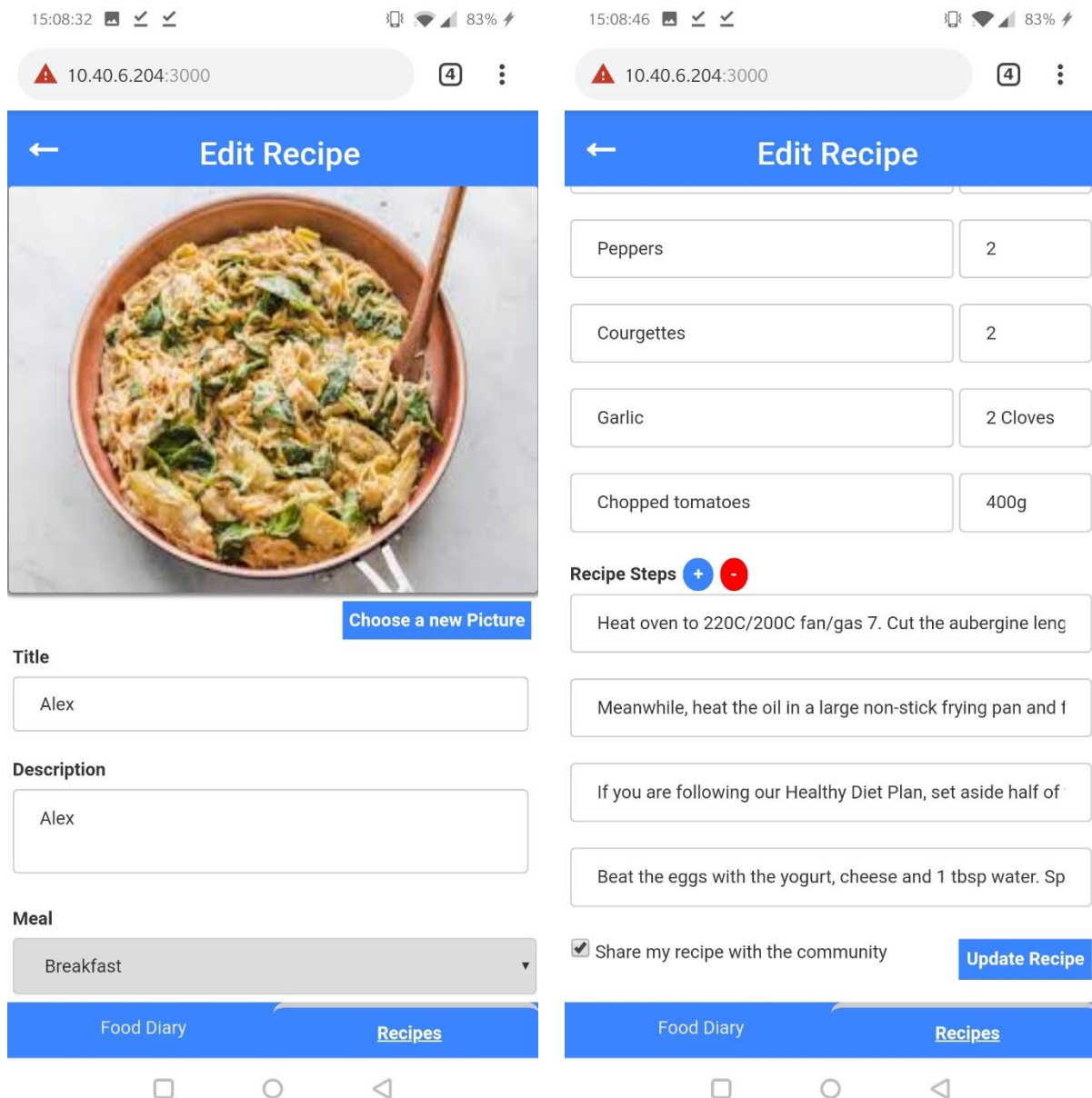


Figure 10 – Edit Recipe Screen

## Recipes – View Recipe

This screen displays the information of a particular recipe.

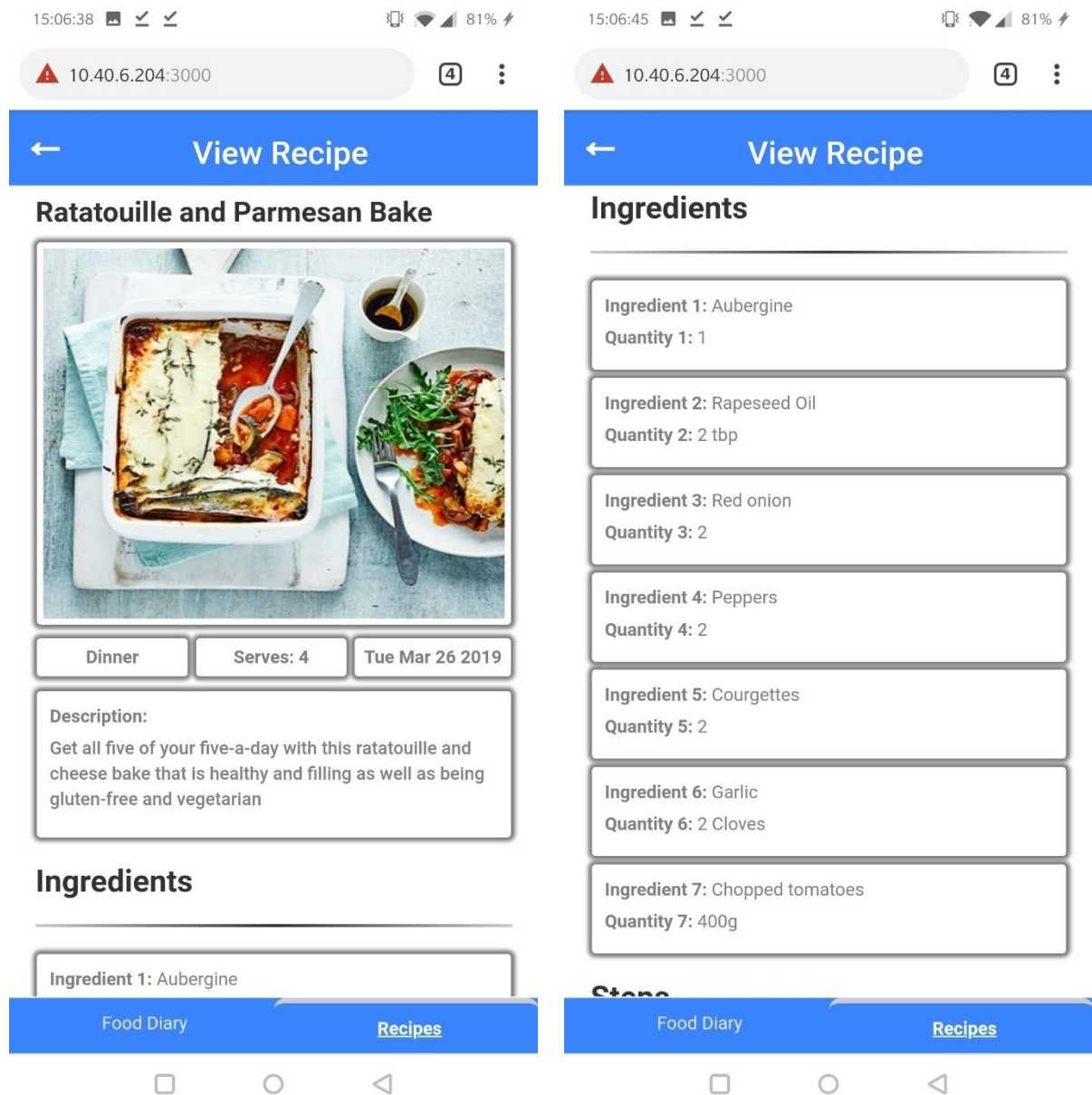
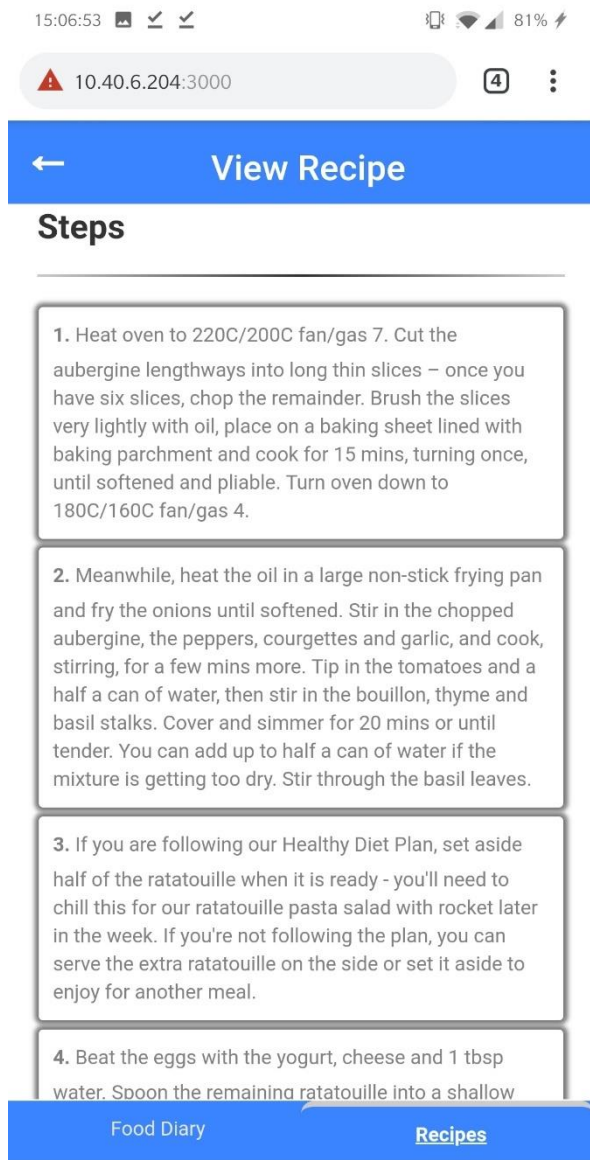


Figure 11 – View Recipe – 1 & 2 of 3





**Figure 12 – View Recipe – 3 of 3**

## Conformance to Specification and Design

The majority of functionality has been implemented, with the exception of favouriting recipes and being able to take a photo with the device camera directly from the application. The favouriting was not implemented because of two reasons. The first reason is that because there is no login functionality it was hard to design the functionality as it made no sense to only have one user be able to favourite recipes. The second reason is that I ran out of time to find a suitable solution. I tried to implement the functionality but everything I tried had to be dropped due to a very poor-quality implementation. The camera functionality didn't get implemented because I was not able to save the photo after it was taken. This is due to the plugin I was using used a different data format to that of a HTML form and this prevented me from being able to send the data to the API. If I had more time, I might have been able to find another solution. Implemented functionalities are as follows:

- Create Diary Entry
- Edit Diary Entry
- Delete Diary Entry
- Create Recipe
- Edit Recipe
- Delete Recipe
- Search Recipes
- Share Recipe with Community

The share recipe functionality shares a recipe with the rest of the users of the application. The functional specification states that it can be shared with outside applications but since the application was changed from a mobile application to a web application, it doesn't make sense to implement the share functionality in this way.

## Description of Learning

Many skills, both technical and personal, were acquired and improved during the course of this project. I will discuss my progress in both areas below.

### Personal

#### Time Management

My time management skills have improved greatly since starting the project as I had to manage my time carefully in order to have enough time for the project and my other subjects. This proved a challenge in the first half of the year, but it started to come naturally in the second half of the year. I am now able to better estimate the amount of time will take to get a specific piece of work done or to get a specific piece of functionality implemented. This year has proved very tough on time management as there was such a large amount of work to get through. It took a lot of self-discipline in order to set aside enough time for everything and there were moments I thought I wouldn't have enough time to get everything done but it all worked out in the end.

#### Communication

Throughout the duration of the project my communication skills have improved greatly. My verbal communication skills were improved through regular meeting with my supervisor and with the rest of the Erasmus+ team. The requirements gathering trip to Amsterdam also helped improve my communication skills as I had to work with people from different backgrounds and people who speak different languages. I am now able to get my point across in a clearer manner, something I struggled with before. The two project presentations helped a lot in this respect as it forced me to leave my comfort zone and speak in front of students and supervisors. Public speaking was something that made me extremely uncomfortable and nervous but facing my fear seems to have remedied the situation somewhat.

#### Teamwork

As my project differed from most other people's projects in that I had to work with a team, my teamwork skills have come a long way since the beginning of the project. We had various meetings about different things such as how the project would be approached, what functionalities to include, what technologies to use, and feedback sessions where we would each give feedback about other team members work. Working with people from different disciplines (Software Development, Games Design, UX/UI Design, Health) was a great experience and really helped develop my teamwork skills and showed me how a well-balanced team can be very beneficial to how a project goes. There were many times when the team had conflicting opinions and suggestions and there were times when the team leads made decisions which the developers weren't happy with, but we overcame everything through discussion and compromise and eventually came to an agreement on the majority of issues that arose. This helped a lot in the development of my teamwork skills.

## Technical

### Python & MySQL

I was competent with Python before I began the project but building the API in Python has increased my skill level greatly. Solving problems I was having has led me to discover new features and quirks of the language. Many gaps in my Python knowledge have been filled in and my general programming skills have improved. Python was primarily used to interface with the MySQL database by retrieving data and creating/updating/deleting records. My SQL skills were improved through the development of the API.

### React JS & JavaScript

React JS is a JavaScript framework for building user interfaces for the web and I had no knowledge of the framework before the beginning of the project. I also had extremely limited experience with JavaScript prior to the project. Before I started developing the project, I did a few React JS tutorials and built some small applications in order to become more familiar with React and JavaScript. Once I had a grasp of some of the basic concepts, I began coding the application. It proved to be a challenge at the start and progress was slow. Once I started becoming more proficient in both React and JavaScript progress started accelerating greatly and I was able to build the functionalities that I set out to do. I have become proficient in JavaScript and I'm confident I'd be able to be a productive member of a professional development team working on a JavaScript application. React uses JavaScript Extensible Mark-up Language so this is something I also had to learn. JSX is used by React as a way of representing HTML. Despite the name, JSX is far closer to HTML than it is to XML. It is used in React code and it makes it easy to pass data into HTML templates. The React code generates HTML from the specified JSX.

### Cascading Style Sheets

My CSS skills were extremely basic prior to the project but I have become a lot better over the course of development. With over 1,300 lines of CSS to style the project, it is safe to say that it was an extremely large chunk of the project and it is the area that gave the most trouble overall. The application was designed from the start to be a mobile application but there was a late change to a web application. This caused me to have to recreate common built in mobile components from scratch. Things like tabs, buttons, and lists which would have been built into mobile development technologies, all had to be designed from the ground up in order to comply with the original design. This is the reason there is so much CSS in the project and it proved extremely tedious to build these standard components. I feel like my CSS skills have improved ten-fold since the start of the project.

### Project Review

This project was one of the hardest and most rewarding pieces of work I have undertaken during my time in college or work experience. It required me to utilise all of the knowledge I have acquired during my time at college. I have learned a lot and I have come out a much better software developer than I was when I started the project. Overall, I think the project went well and was a success but there were some things that didn't go right.

## What Went Wrong

The first thing that went wrong with the project was near the beginning. We had our first meeting with the Erasmus+ team in Carlow where we were filled in on the early details of the project and what the goal of the project was. At this meeting we were assigned a particular piece of functionality which would be our final project and part of the final Erasmus application. The functionality that was assigned to me was a map which displayed information about restaurants and if the restaurants were coeliac friendly. After the meeting I started to work on my research document which came out at nearly 7000 words once it was finished. In November, we flew to Amsterdam for a week for requirements gathering. By the end of this week I was told to change the focus of my project to a food and recipe diary as it was more in line with what the users of the application wanted. This was a set back and required me to carry out my research again and write a new research document. There were some things that I reused but the majority of it had to be re-written which wasted valuable time that could have been put into something worthwhile.

Another set back came after the functional specification and design document were written and before I began writing code. Up to that point, we had agreed we were building a mobile application using a hybrid mobile development technology such as Cordova or Xamarin. I wrote the functional specification and the design document with this assumption in mind, so the application was designed to be a mobile application. This meant that the user interface was designed to be that of a typical mobile application. Just before we began writing code, we had a team meeting where we were told we wouldn't be developing a mobile application and instead would be developing a web application using React JS. We were also told we would focus on a mobile browser as that is how most people would use the application. The reason this change was made was to make it easier to show the application to people. Instead of asking people to download and install a mobile application, they would be able to just access the website on their browser. I feel like this was a bad trade off and we should have continued with the original plan. This made the development of the UI unnecessarily difficult. I spent half my time on the project writing CSS and trying to make the application look and feel like a mobile application as it was designed in the functional specification. This proved very difficult as my CSS skills were not that good, and I had to design and build common mobile elements such as tabs. The whole application would have looked a lot better had we been using a mobile development technology where the tabs, buttons etc. are built in and can be drag and dropped into the application. If the technology was set in stone from the beginning of the project, the final result would have been much better in terms of usability.

One problem that arose from writing the CSS from scratch is that the UI can become messed up depending on the device the application is running on. This is because I developed the application using my phone and laptop, so it looks perfect on any device which has the same resolution as those. In order to fix this issue, I would have had to use CSS media queries. Media queries are a way of specifying different CSS depending on the device being used. I already had to use media queries to develop for both my laptop and my phone so it wasn't feasible to write media queries for every possible device as it would have required me to test every resolution using an emulator which would have taken up too much time. It was hard enough to develop the CSS for just two devices so I wouldn't have been able to do it

well for more than that. In hindsight, I would have used a responsive CSS library such as Bootstrap.

One of the biggest problems with the project was the time constraints. It took nearly the whole of my time on the project in order to get the functionalities that were outlined in the functional specification implemented. I didn't have enough time to properly act on most of the feedback I have received from the various team meetings. I think I should have reduced the number of functionalities when I was designing the application with my team during the Amsterdam meeting. That way I would have been able to refine each functionality more as I would have had more time to work on each one. For these reasons, it was hard to implement feedback as I was too preoccupied with creating a functioning application. I was able to act on some feedback such as some user interface improvements. I also got some feedback saying there should be a way of quickly adding common entries to the food diary, so I implemented a quick add feature. This feature displays a list of entries that were recently added by the user and can be clicked and added to the diary immediately. The feature vastly improves usability by making it less tedious when adding the same items every day.

Another issue I encountered during development was to do with system pop-up messages such as "Are you sure you want to exit? All changes will be lost." or "Recipe successfully saved". For most of development I used a JavaScript system alert box as a placeholder with the intention of replacing them with a more custom solution. This proved to be more difficult than I had anticipated, and I was unable to replicate the confirm functionality. I was able to reduce the number of alert boxes by using a React plugin which displays pop-up messages so when a recipe is added a pop-up is displayed at the top of the screen. If I had more time, I might have been able to come up with a solution for all of the pop-up boxes.

### What Went Right

Overall, I think the whole project mostly went right. There were no serious blockers that prevented me from completing any of the base functionality. I think the process of learning React went well. When I first started to learn React, I had a bit of a hard time getting my head around it and how data was handled but once I completed a few tutorials and had sufficient time with React I was able to start programming at a more efficient and faster pace. Once I became comfortable with React, I started to enjoy developing the application and actually looked forward to having the time to work on it again. I just wish I had more time to improve the existing functionalities and add new ones.

I think the development of the API also went relatively well and I learned a lot in the process. There was a small hitch, but it was quickly overcome. At the start of the project when we were discussing the API, I suggested we use Flask as it is simple and easier to use than a lot of API frameworks out there. We had agreed to use Flask and we were told an API skeleton will be put on GitHub and we could add our API calls to the skeleton. A lot of time passed and there was still no sign of the API skeleton being uploaded to GitHub so I just began working on my part of the API locally using Flask and Python. When I had the majority of my API developed locally, the shared API was put onto GitHub. There was just one problem, the API was written in a Python technology called Tornado. Tornado is a scalable, non-blocking web server and web application framework. I'm not sure why the decision was

made to change from Flask to Tornado, but I explained to the team that I wouldn't have enough time to convert my nearly finished Flask API to a Tornado API. The team lead set up a session to help me and another of the team convert the application so that it uses Tornado. When I showed up to this meeting, we were told that we would just use Flask, like we had agreed upon at the beginning. This was a big relief as I didn't want to have to learn a new framework that late in the development process.

We also discussed how we would structure the API at this meeting, and I suggested using Flask Blueprints as I had used them before during my work placement and thought they were a good way of structuring an API which has several distinct parts. It was agreed that we would use the Blueprints and I was asked to set it up and push it to the API repository. I was also tasked with writing a small tutorial on how to add a new blueprint to the API which could be used by the rest of the team when they were adding their part of the API. Shortly after this, I completed my section of the API and pushed it to the repository so it would be ready to be hosted on a server.

Early in the project, we travelled to Amsterdam for a need finding trip in the Amsterdam University of Applied Sciences. While there, we met the team from Austria and Amsterdam and tried to discover what coeliac patients wanted out of the application we were creating. We achieved this through various team activities. Through these activities we put ourselves in the shoes of a coeliac patient in order to better understand the struggles they face in everyday life. One of the activities was to step through the day of a coeliac patient and try to figure out which parts of their life was affected the most from their condition. Another activity was to cook a gluten free meal. We were split into 3 teams, one each for starter, main and dessert. My team was on dessert, so we decided to make a chocolate cake. The first thing we did was go to the supermarket to buy the ingredients which was not easy due to the language barrier and due to gluten free products being extremely hard to find. There were even products that said gluten free on the front of the packet and said there may be traces of gluten on the back of the packet. Once we had the ingredients we went to a kitchen in the college and cooked a meal. We then sat and ate the meal together. Some of the other activities included brainstorming and prototyping. Once we had a clickable prototype we were then tasked with walking around the college and asking people to test the it. This went well and people responded well to some of our ideas. We also got to go on a tour of Zaanse Schans where we saw old windmills and dwellings, and we got to see how Dutch clogs are made. The whole Amsterdam trip was an extremely rewarding experience and it was definitely the highlight of my final year project.

## Future Improvements

Some improvements for the application:

- Integration with the GIP test kit reader.
- Improved user interface and usability.
- Proper logos and symbols.
- Better use of colour.
- Better forms.
- List of common food items for the food diary and for recipe ingredients.
- Community recipes favouriting system.
- Community recipes comments.
- Improved calendar for the food diary – Display progress information.
- Better error handling when connection drops during certain operations.
- Better information messages.
- Login functionality.
- Use of camera from the application to take pictures when creating a diary entry or recipe.

## Conclusion

I'm very happy with how the project went and once I learned React, I really enjoyed developing the project. It's not perfect by any means but I think I achieved the vast majority of what I set out to do in the first place. I'm very happy to have been picked to be a part of the Erasmus+ project and I really enjoyed working with the team. I really hope that my work will be used in the future by people with coeliac disease. There are some things I would have done differently but I think I did my best with the time I had. It was a struggle trying to balance all of my other subjects, but I think I did a decent job in the end.