

**NELSON MANDELA UNIVERSITY**

**School of Information Technology**

**IT Project 3 (ITPV302)**

**Bachelor of Information Technology (BIT)**

**Date**

***Implementation Documentation***

**Compiled by**

|  |  |  |  |
| --- | --- | --- | --- |
| **Lecturer** | **Student Name** | **Student Number** | **Date Submitted** |
|  |  |  |  |

Table of Contents

[Introduction 2](#_Toc180066606)

[Choice of Tools 3](#_Toc180066607)

[Firebase 3](#_Toc180066608)

[Visual Studio Code 3](#_Toc180066609)

[Api 3](#_Toc180066610)

[Extracts of Complex Code 4](#_Toc180066611)

[Source Code References 5](#_Toc180066612)

[Flutter and Firebase Tutorial 5](#_Toc180066613)

[Problems Encountered 6](#_Toc180066614)

[Creating a Flutter App 6](#_Toc180066615)

[Using Firebase 6](#_Toc180066616)

[Bottom Overflow Error 6](#_Toc180066617)

# Introduction

# Choice of Tools

## Firebase

We selected Firebase because it has features that we can make use of and provide adequate security for the app.

Firebase handles the Authentication for the login and register section of the app and ensures that the details are protected. Firebase also will handle the recipe information via the Firestore database and provides the tools to enable syncing of changes when a device regains an internet connection.

Firebase also hosts a console that can be used to monitor the app usage and other metrics that will ensure the app can run smoothly.

## Visual Studio Code

Visual studio code (VS code)was selected as it works well with Flutter and Firebase , VS code also provides a useful way of performing work in a group with an easy to use source control tool.

The benefit of VS Code is that the project can easily be developed within a isolated environment away from external packages on the system.

## Api

For the recipe data we made use of the Api provided by edumam. This is due to the Api allowing use for free as well as having a large number of recipes that have the field we need such as diet, nutrition information etc.

# Extracts of Complex Code

# Source Code References

## Flutter and Firebase Tutorial

Description

To learn more about the dart programming language, flutter and the fundamentals of firebase we followed a course created by Vandad Nahavandipoor.

Usage

* Flutter setup
* Login View
* Email Verification
* Firebase backend setup
* Auth service
* Migration to Firestore Service
* Bloc for routing and dialogs

Reference

freeCodeCamp.org (Director). (2022, February 24). *Flutter Course for Beginners – 37-hour Cross Platform App Development Tutorial* [Video recording]. https://www.youtube.com/watch?v=VPvVD8t02U8

## Grid view UI

Description

Implementing the grid view of the screen and structuring UI elements by using GridView.builder.

Usage

UI elements of Saved Recipe screen

Reference

Mitch Koko (Director). (2022, November 4). *Donut App UI • Flutter Tutorial* [Video recording]. https://www.youtube.com/watch?v=OmYL-VK75-o

# Problems Encountered

## Creating a Flutter App

Our goal to create a cross-platform app meant that we needed a framework that allowed for simple cross platform development which was Flutter. We hadn’t used Flutter and the language dart which it uses before and needed to learn how to make use of it in order to begin creating the app.

We found a video on YouTube that helped us to gain insight on how to use Flutter and code an app with dart. The tutorial allowed us to learn and use that knowledge to start developing the app.

## Using Firebase

For authentication we did not want to have to handle an authentication for the app as it would add unnecessary complexity to the app. Within the same YouTube video that taught us flutter, Firebase was used to handle the Authentication and provide storage which we decided to make use of for the app.

Firebase also solves the problem of syncing changes to the database when a user returns to an area of internet connectivity.

## Bottom Overflow Error

Occasionally we encountered an error that would pop up whenever a text field was pressed and brought up the keyboard which was the bottom overflow by x pixels.

This error comes about when the screen is made to be static and then an element such as the keyboard will push the content below it off the screen. To fix this we wrapped the widget that is shown under a Single Child Scroll View which allows the screen to be scrollable.

## A screenshot of a recipe Description automatically generatedRight Overflow Error

Figure : Right Overflow Error

When designing the UI elements for the saved recipe screen there was a right overflow error as seen in Figure 1. This occurred when using the GridView.builder widget to allow two items per row. Causing an overflow due to the items overlapping.

To fix the error we needed to ensure the width of the grid items do not overflow by using a combination of crossAxisSpacing and padding. To avoid a bottom overflow error for the grid items we used mainAxisSpacing as well. The aspect ratio of the grid items was also changed so that all contents fitted inside.

## Stack Overflow Error

Figure : Stack Overflow on Search Screen

On the search screen where we were testing a placeholder model for displaying data, we got a stack overflow error. There were no errors highlighted in the code, but the debug console showed it was occurring because of the model class.

The problem occurred because it was calling the getRecipe() method, which caused an infinite loop of the data being displayed. To fix the issue the recipe list itself was returned.