



Project Log

Menu Solutions

Jack Corbett



Contents

Analysis	3
Background	3
Statement of Problem	4
Users	5
Research	6
End User Questionnaire	6
Online Survey	7
Current Solutions	11
Programming Research	11
Table of Analysis	12
Objectives	13
Solution Description	13
Core Objectives	13
Extension Objectives	14
Modelling	15
Flow Diagrams	15
Navigation	15
Login System	16
Database Structure	17
Design	18
Overall System Design	18
Complex Algorithms	20
Recipe Selection	20
Display Recipe	23
Shopping List	25
Add Recipe	28
Validation Required	29
Registration Form Validation	29
Login Form Validation	31
Web Server File Structure	31
Database	33
Definition of Data Requirements	33
Querying the Database	38
Programming	39
Language Choice	39



Libraries	39
User Interface Design (HCI)	39
Desktop UI Diagrams	40
Mobile UI Diagrams	42
Data Integrity and System Security	44
Encryption	44
Security	44
Prototype	45
System Testing	47
Overall	47
Account	49
Family	53
Plan	56
Calendar	61
List	63
Admin	65
Technical Solution	71
Overview	71
Code Contents	71
Includes (PHP Scripts)	81
User	96
Family	102
Plan	107
Calendar	112
List	117
Assets	120
SQL	125
Evaluation	127
Overall Effectiveness of System	127
Completion of Objectives	127
Extension Objectives	130
User Feedback	131
System Improvements	133
Bibliography	134



Analysis

Background

Many families face the same problems each week, deciding what to eat that will suit everyone's dietary needs and preferences. It is becoming a real issue in our modern world where time to cook is in short supply and people's health is being compromised because of it.

Families often resort to fast food or waste food which they purchase and never actually use. This needs to stop as there simply is not enough food globally to maintain our current habits. "*795 million people in the world do not have enough food to lead a healthy active life*"¹ because we as a culture overeat. If we want to eradicate hunger we need to take control of our portions and wastage.

Obesity is another side effect of our overeating and it's on the rise. In the UK *61.7% of adults are currently classified as overweight or obese*². This not only causes many health issues but has a knock on effect on our health services that are treating avoidable illnesses and conditions.

These issues cover the bigger picture but in order to fix them we must first fix what is happening in the home. That is why I believe it is so important to build a solution that can take the stress and time commitment out of eating healthily, allowing users to find meals that even the fussiest of kids will enjoy, making it more accessible than ever before.

¹ World Food Programme, Hunger statistics, <<https://www.wfp.org/hunger/stats>> [accessed 13 September 2016].

² Public Health England 2016, UK and Ireland prevalence and trends: Public health England obesity knowledge and intelligence team (2016), <https://www.noo.org.uk/NOO_about_obesity/adult_obesity/UK_prevalence_and_trends> [accessed 13 September 2016].



Statement of Problem

I am going to solve the problem of not knowing what to eat, eating too much, eating an unhealthy diet and wasting food. In order to do this I need to provide a tool to help plan healthy easy meals without the stress.

The barrier between eating quick and easy unhealthy food and a healthy diet is simply time. Not everyone has the time to plan out a large and complex agenda for what they are going to eat (or money to pay someone else to) each week and many that do, would still rather spend that time doing anything else. In order to fix this I need to deliver a solution that takes minimal time to set up, to fit into people's daily lives.

With everyone leading such busy lives it is not everyday that everyone is able to eat the same thing together; leading to people compromising the quality of their food to have something quickly. I can solve this by allowing the user to input who will be eating each day and how long they have available to prepare the meal, meaning the plan is fully adaptable.

In order to overcome overeating issues the tool takes control of portion sizes by calculating the amount of ingredients depending on how many people are eating. This also helps deal with the wastage issue as it means the user will only purchase and cook what they need.



Users

My primary user, despite there being a real demand for a service like this in the market, is my Mum. Deciding what to cook for the week has forever been a frustration for her. Cooking many different meals to suit everyone's preferences and resorting back to very similar things week in week out makes it a seemingly impossible task to try anything new.

Many an hour is wasted trying to come up with something different and exciting to eat which puts a lot of stress on my Mum who does the shopping each week. At the start of every week she will write out a plan on a calendar for each day so she knows what to get when shopping. This is a very efficient way of doing it as it minimises wastage and means it can be adapted with relative ease if plans change. However, to get to this stage it will have wasted her a large amount of time thinking and planning, when on many days she may revert back to something we eat all the time anyway as she cannot think of anything else. I will be simplifying this process further by creating an algorithm to make the decisions for her.

I mentioned at the start that this tool is made predominantly with my Mum in mind but she does a good job on her own, it will just save her a lot of time and effort and enable us to try new things more often.

The people this will really benefit are those who simply skip the planning step and just buy what they like the look of from the supermarket. They then end up only using half of what they bought and having to buy in more during the week anyway to make up the meals. To get people out of this habit my solution needs to be just as simple as walking into the shop and filling up your trolley. Only this time not buying more than they needed and making the correct choices to provide them with a healthy balanced diet.

This will also help in families with picky eaters to encourage them to try new things and lead to only one meal being cooked each night rather than multiple for different people.



Research

Once I had decided to create a meal planning tool I looked to many places for inspiration. Simplicity was something I knew would be really important for my end users as the whole purpose of using my tool was to take the hassle out of the task. This meant it was essential I was able to hide the complexity behind a well designed UI. I really liked the clean design of websites like Apple³, Tesla⁴ and Intel⁵ who all stuck to a basic but effective look and decided I wanted to implement this into my project. Each had a large background image or video, navigation bar at the top of the window and a call to action button/text in the centre.

I then moved on to planning the functionality which would ultimately be the success or failure of my project. To get into the mind set of menu planning I decided to produce a questionnaire for my end user, my Mum. As I had never created a family meal plan myself, I needed to understand her frustrations. By picking my questions carefully I could tap into her process for selecting recipes and begin to understand how it could be converted into an algorithm.

Another resource was a family friend who is a trained nutritionist. Although she no longer practices it she has helped many clients improve their diet by building menu plans for their specific requirements. As my Computer Science based solution was effectively attempting to act as a virtual nutritionist her input would be invaluable.

So let's begin with my Mum's response and demonstrate how this affected the scope and direction of my project.

End User Questionnaire

Notes I made summarising her response

1. Describe the process of writing a meal plan

I start by working out who is eating on each day so I know who I am cooking for and what time they are going to want it ready. I also need to work out what I can cook quickly on Tuesdays and Fridays as I start work at 6 so need to do it early. To make sure we don't eat the same things each week I look at what we ate the week before and then start to fill in each day, which is the bit that takes ages to try to come up with things to eat.

³ Inc, Apple, *Apple (United Kingdom)* (Apple (United Kingdom), 2017), <<http://www.apple.com/uk/>> [accessed 3 April 2016]

⁴ *Tesla UK* (2017), <https://www.tesla.com/en_GB/> [accessed 3 April 2016]

⁵ Corporation, Intel, *Data center solutions, IoT, and PC innovation* (Intel, [n.d.]), <<http://www.intel.co.uk/content/www/uk/en/homepage.html>> [accessed 3 April 2016]



Jack Corbett

2. What do you find most frustrating about making the plan?

The most frustrating part is definitely choosing the meal itself. Scheduling who will be eating and working out how long I have is easy but coming up with new things is impossible when everyone likes different things. There are so many different foods I would like to try but can't because not everyone would like it and I don't want to cook 4 different meals every evening.

3. How long does it normally take you?

I do it each Sunday and it normally takes a couple of hours of thinking while I get on with other things and if I still don't know I get the family to help decide. It can take longer though if people change their minds about if they are going to be eating and when they want it ready.

4. What would a solution have to include to make it useable for you?

I would need to chose who will be eating on what days as it changes every week depending on when everyone is working. It needs to suggest simple, regular recipes that people will actually like and cater to everyone's preferences taking into account a wide range of information to make decisions. It also needs to be really easy to use and simple looking so I can actually use it! I need to be able to get easy access to my meal plan for the week and it needs to take into account what we had before so we don't keep getting the same recipes.

Online Survey

To take my research a step further I decided to take what I learned from my end user interview and produce an online questionnaire. Google forms was perfect for this as it offers a wide range of question types and integrates seamlessly into a google sheet for analysis. I shared the questionnaire on social media and sent it directly via email to family friends.

The form includes a header image of various vegetables (tomato, potato, mushroom) and a brief introduction about the researcher's project. The first question asks if the user enjoys meal planning, with a rating scale from 1 (I despise it) to 5 (I love it). A note at the bottom of the question says: "If you rated 5 please press next to proceed, if not please answer the questions below". The second question asks what the user dislikes about meal planning, with four options: "Coming up with new things to eat", "Finding recipes that everyone in the family likes", "Planning who is going to be eating each day", and "Writing a shopping list".

Menu Solutions

I am currently performing research for my Computer Science A Level Project. I have decided to create a web based menu planning tool to make it easier to eat delicious home cooked food for the whole family every night of the week. To understand what my solution needs to deliver to facilitate this, I need your help.

* Required

Do you enjoy meal planning? *

1 2 3 4 5

I despise it I love it

If you rated 5 please press next to proceed, if not please answer the questions below

What do you dislike about meal planning?

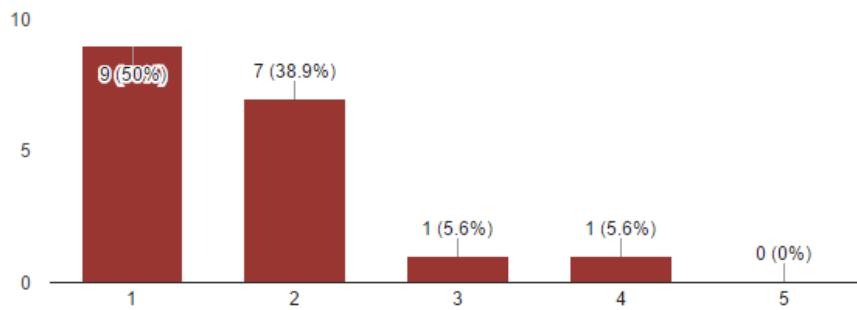
Coming up with new things to eat
 Finding recipes that everyone in the family likes
 Planning who is going to be eating each day
 Writing a shopping list



Jack Corbett

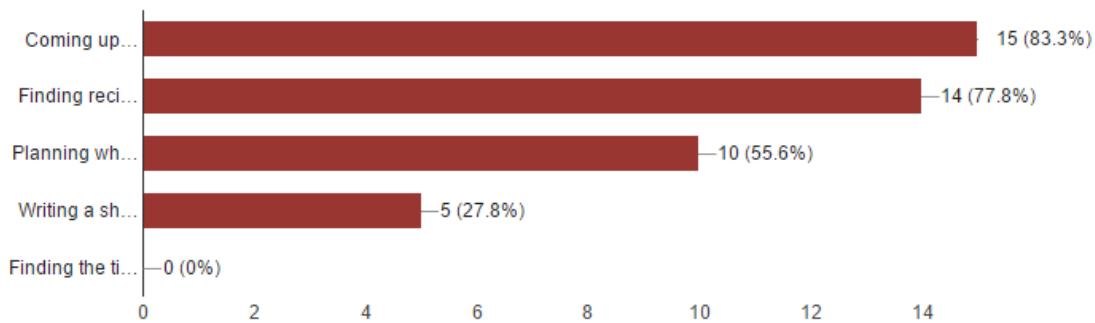
As expected I received very similar results with a universal hatred of meal planning being apparent. Having never been responsible for family meal planning I hadn't realised just how frustrating and difficult it was. 50% of the 18 families to complete my survey voted that they despised planning meals.

Do you enjoy meal planning? (18 responses)



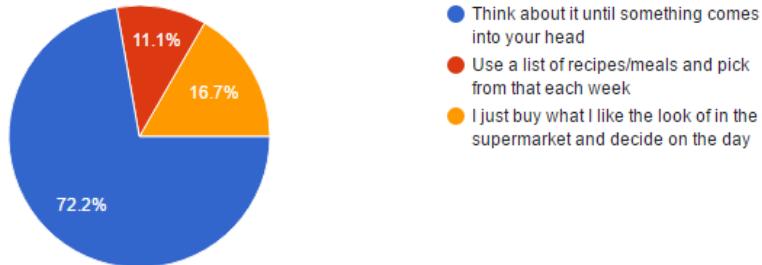
I then asked, for anyone that did not rate 5 saying they loved meal planning, what it was they disliked about creating the plan. Coming up with new things to eat and finding recipes that everyone likes came out on top, followed by planning who is eating each day and writing a shopping list. This shows that for my program to be a viable solutions it needs to make finding new recipes easy and the recipes need to be simple 'regular' meals that are family friendly. It must also be flexible to allow different family members to eat depending on the week and day.

What do you dislike about meal planning? (18 responses)



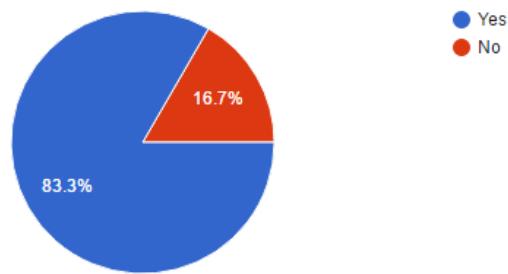


How do you currently decide what to eat? (18 responses)

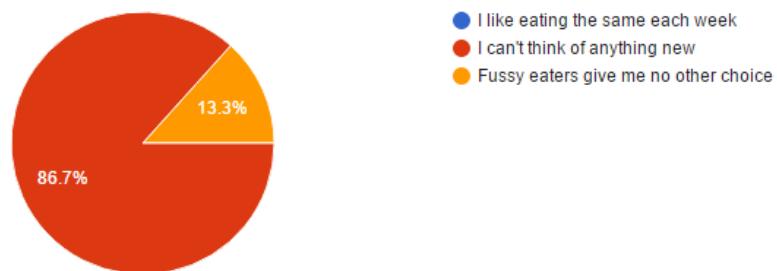


When I asked how people currently make their plans I was surprised to see that most just relied on their memory to come up with recipes and more people just bought what they felt like from the supermarket than building up a list of all the recipes they liked and choosing from that. It is clear that people need something that is easy and does most of the task for them.

Do you feel like you eat the same thing each week? (18 responses)



If you answered yes why? (15 responses)

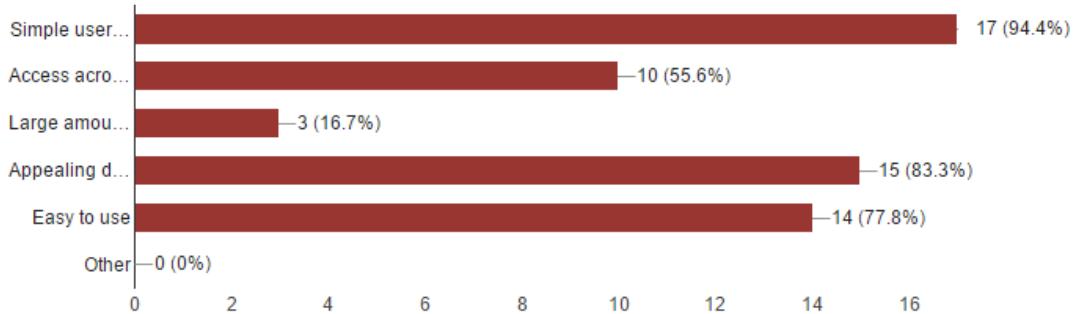




Jack Corbett

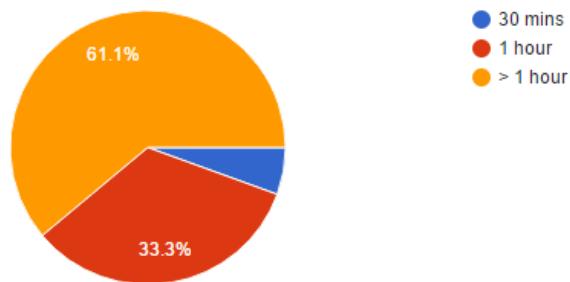
Almost all the participants said they felt like they ate the same thing week in week out and put it down to the fact that they couldn't think of any new recipes. I need to take this out of their hands by picking random recipes but also ensuring the same recipes are not used two weeks in a row.

What do you look for in an app? (18 responses)



When I asked what users looked for in an app, the need for something simple and easy really showed. I need to make sure I hide the complexity involved with generating a plan, creating the shopping list and managing a user's family behind a well designed, scalable UI.

How long do you spend each week planning meals? (18 responses)



This was probably the most shocking response as well over half of the people that filled in my survey spent over an hour a week trying to decide what to eat. This shows just how much of a demand there is for a tool to make it easier as it will save families a lot of time and stress.

I also left an extra suggestion question at the bottom and although only two of the eighteen participants gave answers they elaborated on two important aspects of my program. The first was about the ability to view the meal plan/recipes on a mobile device. This makes a lot of sense as the thought of turning on a computer or laptop just to get up a recipe is ridiculous so it's essential when I come to designing my UI I consider how each page will scale on smaller devices. The



second was around which family members were eating and the ability to only include members on certain days or not include someone at all one week. The example they gave was their husband eating at work and a girlfriend that ate every other week.

Current Solutions

There are already many different online options to try to help users decide what to eat. Both Tesco⁶ and Sainsbury's⁷ have a tool built into their website which each give a range of options, the difficulty is that many of the recipes suggested are not family orientated meals and trying to get the correct portion sizes is challenging. Meaning it takes almost as long as coming up with it yourself by looking in a recipe book, once you have made enough changes to cater to fussy eaters that won't eat Camembert Fondue, Serrano Ham and Manchego Salad or Salmon and Asparagus Fajitas.

The other option available is to basically do the hard work yourself and find the recipes online but this takes a serious amount of time so why not let a website do it for you? A website that can store all your recipes together in an organised plan you can access from any device. The Love Food Hate Waste⁸ page does a very good job of showing easy to cook recipes the difficulty is the UI is not that easy to navigate and it still requires a lot of input from the user, which I am looking to avoid.

The closest app currently available to what I am looking to achieve is Menu Planner⁹ which is available on iOS and Watch OS. It brings a lot of great features but users must still spend hours choosing what they are going to eat each day as it will not auto generate the plan. It also, to the frustration of users who have spent large amount of time adding their own categories, removed the feature in a recent update. With a very complex larder system it overcomplicates the process that needs to be as simple and easy as possible for the users.

Programming Research

At first I wanted to make an iOS app using Apple's Swift programming language but realised it would limit the user base too much. This is why I decided to build a website that could be accessed across all the users devices and scale to each. This means if the user would prefer to generate the plan and set up their family on a computer they can but they can just check their phone or tablet for the recipes each day.

⁶ Meal planner | weekly meal plans | Tesco real food (Tesco Real Food, [n.d.]), <<https://realfood.tesco.com/meal-planner.html>> [accessed 17 October 2016]

⁷ Sainsbury plc J 2016, *Healthier family recipes* (Sainsbury's Inspiration, 2016), <<https://livewellforless.sainsburys.co.uk/healthier-family-recipes/>> [accessed 18 October 2016].

⁸ WRAP, *About food waste*, <<http://www.lovefoodhatewaste.com/content/about-food-waste-1>> [accessed 18 October 2016].

⁹ InnovaDev, *Home - menu planner* (2016), <<http://mp2.menu-planner.com/>> [accessed 18 October 2016]



To deliver this I decided to use a combination of HTML, PHP, SQL and CSS. I would be coding my algorithms in PHP and it was the language I had the least experience with so I started by taking the codecademy PHP course and taking notes in quizlet to turn learning the syntax into a game.¹⁰ Once I had completed the course I felt I understood the syntax but I also needed to learn PHP specific functions and how to use things like super globals and cookies. I took the recommendation of Eli The Computer Guy¹¹ in his introduction to PHP on Youtube and purchased PHP and MySQL for Dynamic Web Sites by Larry Ullman.¹² This takes you from the basics of setting up a local apache server to protecting your site from SQL injection. Used alongside various Youtube playlists and many Stack Overflow¹³ threads I was able to reach the level of understanding to start coding my website.

I had experience with HTML and CSS in the past, having put a couple of very basic sites together. However W3 CSS¹⁴ will be extremely helpful by building on one of their templates I can concentrate on the functionality and design rather than simple formatting. Although I'm sure I will need to make some changes it's an excellent starting point with tutorials to go along with it.

Table of Analysis

Date	Analysis
10/08/2016	Questionnaire for primary end user.
21/08/2016	Published the online Google form. Sending it out on social media and via email to a range of families.
05/09/2016	Reviewed survey results and reflected upon them with my end user.
07/10/2016	Researched current solutions by testing their capabilities and reading reviews.
03/12/2016	End user tested navigation from registration to meal plan generation on a laptop and phone to collect feedback on HCI.
25/02/2017	Reached out to families that participated in my survey to test the completed site and give feedback.

¹⁰ "PHP", Codecademy, 2017 <<https://www.codecademy.com/learn/php>> [accessed 8 July 2016].

¹¹ "PHP Programming Part 1: Introduction To PHP Programming", Youtube, 2017

<https://www.youtube.com/watch?v=27dR_sLaM74&t=891s> [accessed 8 June 2016].

¹² Larry E Ullman, *PHP And Mysql For Dynamic Web Sites*, 4th edn (Berkeley, Calif.: Peachpit Press, 2012).

¹³ "Stack Overflow", Stackoverflow.Com, 2017 <<http://stackoverflow.com/>> [accessed 7 June 2016].

¹⁴ "W3.CSS Home", W3schools.Com, 2017 <<https://www.w3schools.com/w3css/default.asp>> [accessed 8 August 2016].



Objectives

Solution Description

Menu Solutions is a web based menu planning tool enabling the modern family to eat a home cooked healthy diet.

It allows users to add their family members along with their dietary requirements. The user then inputs each week who will be eating on each day and the time they have available to prepare a meal. My algorithm considers all of this information to generate a meal plan for the user with healthy simple recipes that suit everyone's needs. It also generates a shopping list containing all the required ingredients.

Core Objectives

Complexity Rating: 1 (Simple) - 5 (Very complex)

1. Account

1. Each user is able to create an account using an email and password encrypted using SHA2 encryption: **3**
2. Access to user specific pages is blocked if not logged in to a valid account: **2**
3. The login and register form is validated to limit errors before the information is submitted to the database using RegEx and the user is presented a specific error message to inform them of the issue: **4**
4. Users remain logged in between web pages giving them access to all of their information until they log out: **1**

2. Family

1. The user can add members to their family: **3**
2. Members information can be edited to keep it up to date: **3**
3. Members can be deleted from the family entirely: **2**

3. Plan

1. The family members that are eating each day can be selected: **2**
2. The time available to cook each day can be inputted: **2**
3. This data can all be formatted and passed through to the generation script: **3**

4. Generation

1. Recipes are selected at random ensuring that there is enough time available to cook it (*view recipe selection algorithm for more details*): **5**
2. Recipes used in last weeks plan must be discounted to avoid repetition: **5**



3. Ingredient measurements calculated individually depending on the number of members eating: **4**
5. **Calendar**
 1. Table containing summary of each days recipes and the date they are to be cooked: **3**
 2. Ability to overwrite the plan for the coming week with warning that it will delete the current meal plan: **5**
 3. A link for each day to a automatically populated page containing cooking instructions and ingredients: **4**
6. **List**
 1. Display ingredients needed for each recipe: **3**
 2. Show the amount of each ingredient needed, adding values if used in multiple recipes that week: **5**
 3. Categorise ingredients to make it easier to find when shopping: **5**

Extension Objectives

7. Improve encryption of user accounts using salting and key stretching algorithms like bcrypt to protect passwords: **4**
8. Allow users to add their own recipes through a form: **5**
9. Tutorial system modal to teach the user how to make the most of the tool: **2**
10. Members can be ticked in the plan page or unticked by default to compensate for family members that eat less often: **4**
11. Separate php scripts using include files: **3**
12. Make the site mobile friendly by changing the display format depending on the size of the users device: **4**



Modelling

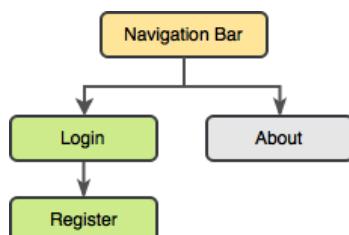
Flow Diagrams

Navigation

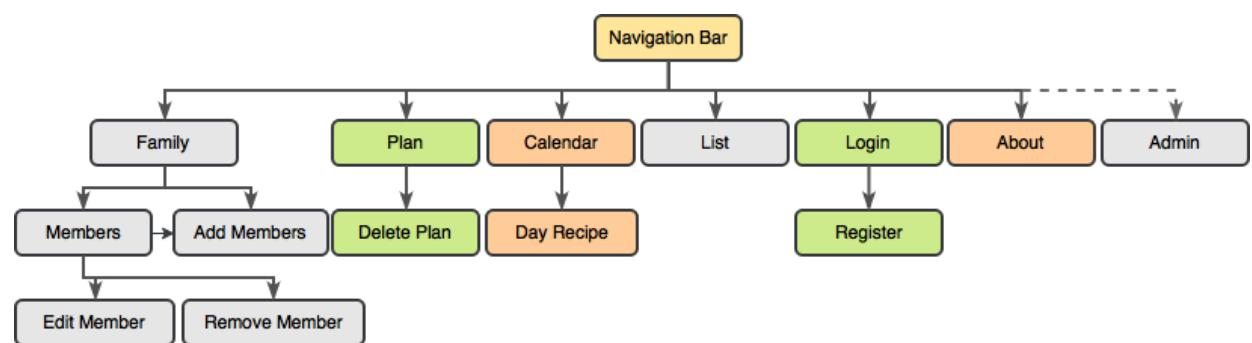
To make using my web app a fluid experience I need to ensure I have my page hierarchy perfected before I start coding. Each main page can be accessed through the navigation bar which remains at the top of browser window. If the user is logged out the bar will show limited options but once logged in they can access every page. Access to the other pages will also be blocked even if the URL is used without a valid session. This will present the user with an error message and ask them to login.

To begin the user must login or register for an account. They can then add their family members through the family pages and select who is eating each day and the time they have to cook on the plan page. The calendar page shows the recipes for each day and the admin page is only available with special permissions. The full features of these pages will be laid out in the design phase.

Navigation Bar Logged Out



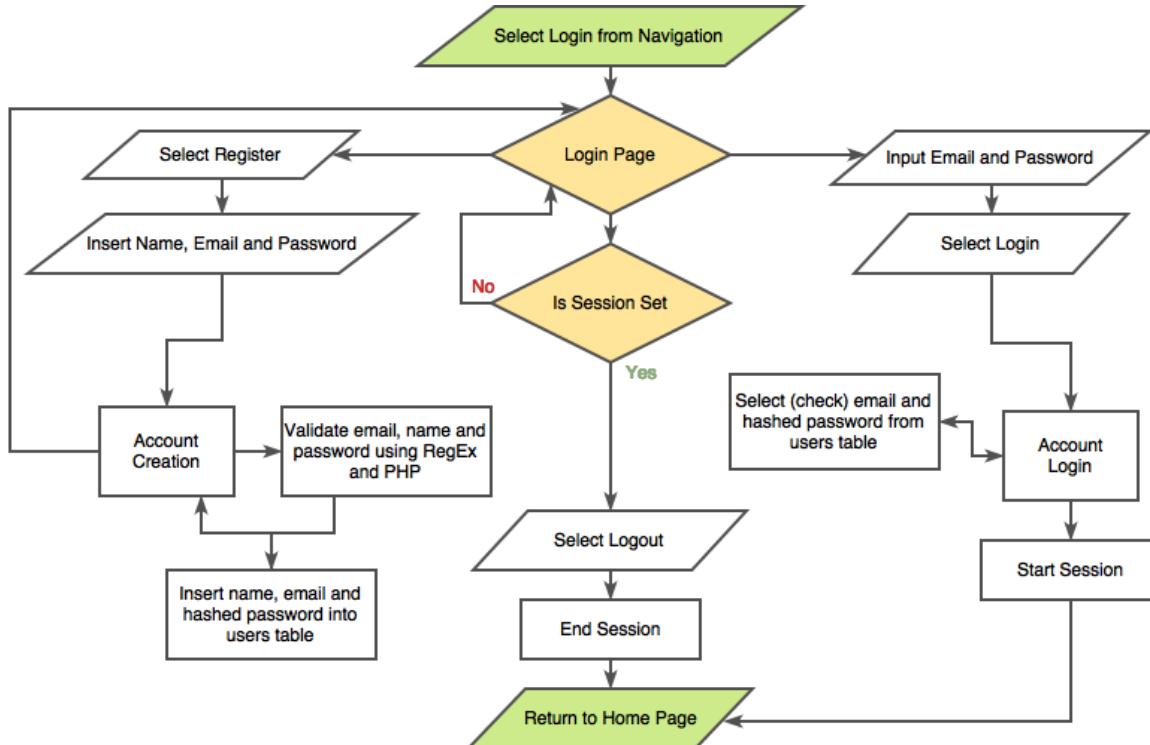
Navigation Bar Logged In





Login System

When a user first accesses www.menu-solutions.com they will be greeted by the home page displaying the logged out navigation bar. In order to progress an account must first be created to keep users meal plans and family members separate and protect their information.



When you first view the login page only the input fields, login button and register button will appear. The logout button is hidden until a session is created. When the user inputs their details and presses login the backend PHP scripts will handle checking the database and starting the user's session before linking back to the home page. To register, the user will click the register button and be taken to a separate page to fill in more of their details. This data will be validated and then submitted to the database. The user can then login to their newly created account.

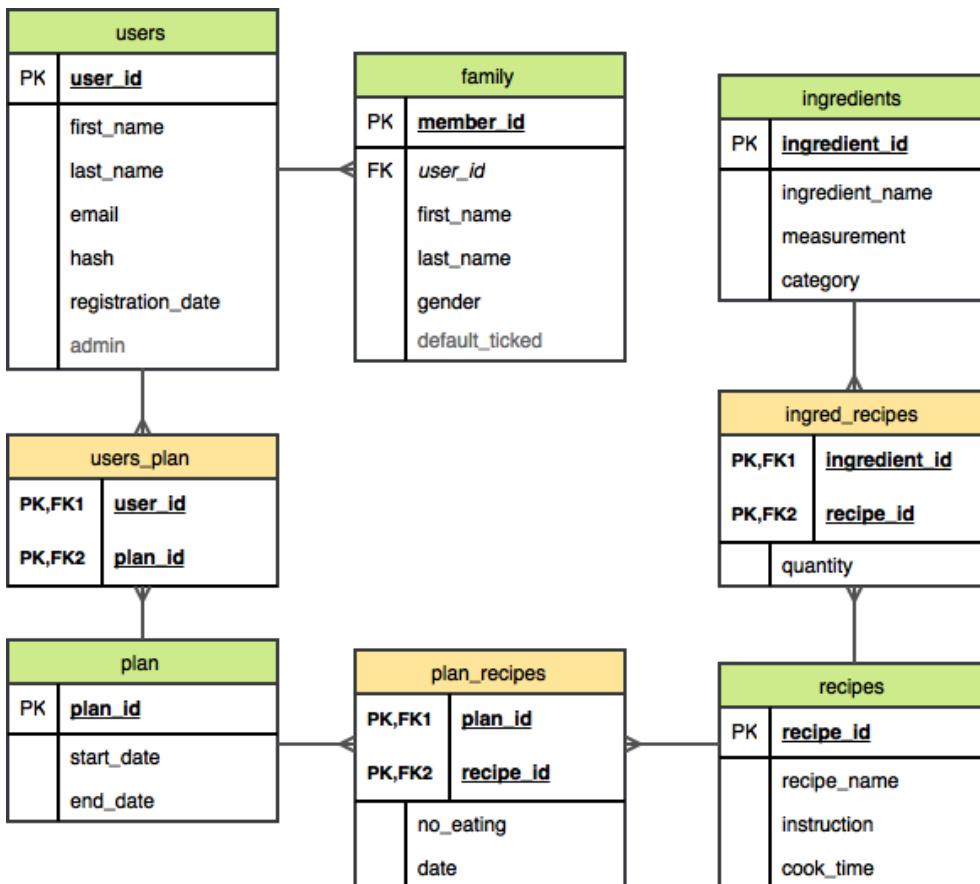
This is essential for my site to ensure sessions are managed correctly in order to display the right data to the user.



Database Structure

In order to store the user's information, family members and recipe plans I am going to build a relational database using MySQL. This way I can hold all the data for each user account and it will make running queries easy by using SQL statements through PHP. Although I will be writing the database code by hand in SQL, PHPmyAdmin will make it easy to visualise my database and check for issues once each table is set up on my web server. To fully normalise the database I used three link tables containing composite primary keys shown in orange.

Database ER Diagram



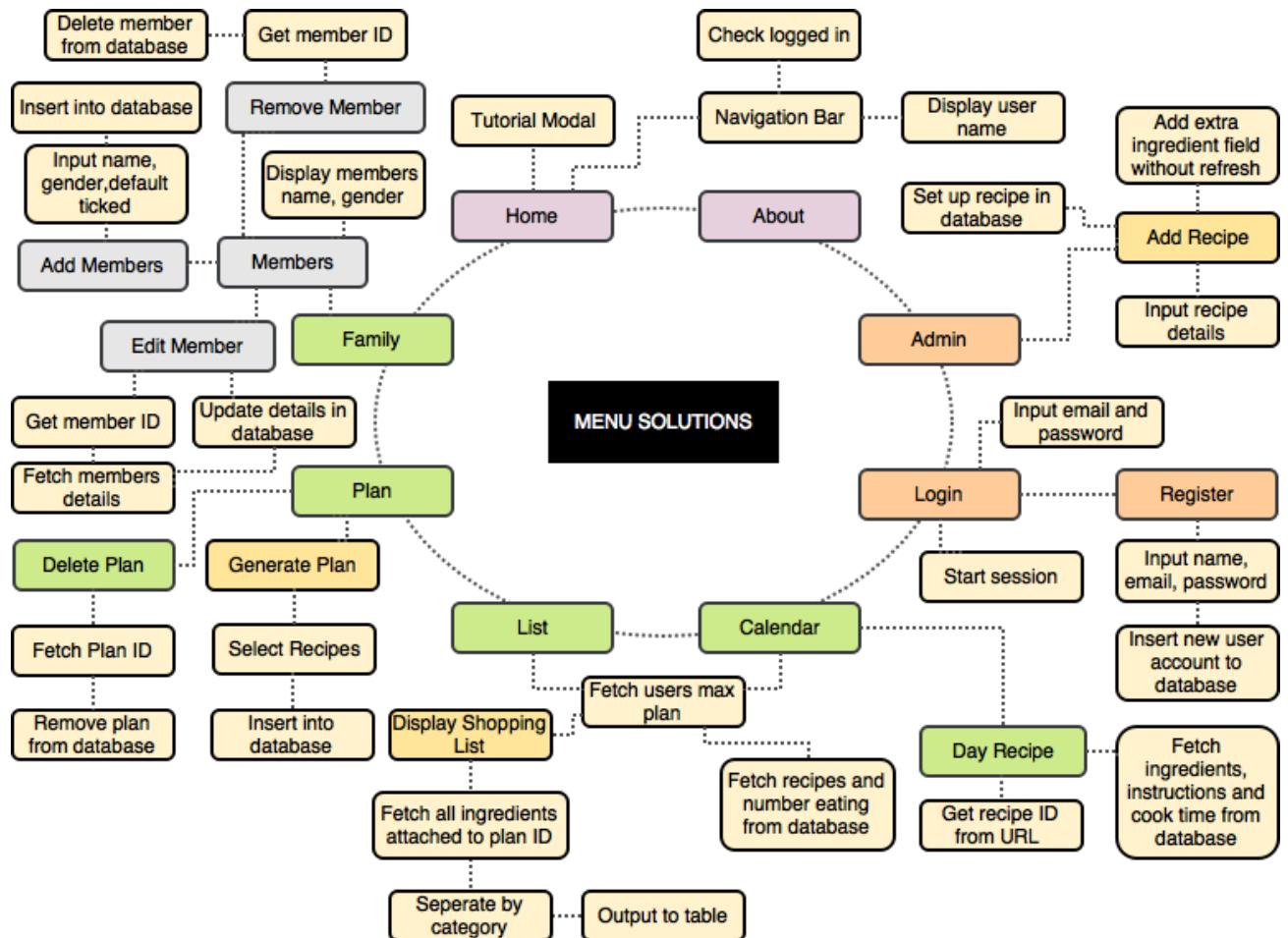
The fields in grey rely on completion of extension objectives



Design

Overall System Design

My program will use a traditional website client server model. This means users will be delivered web pages from the server in HTML, Javascript and jQuery. The contents of these pages will be influenced by PHP which will manage querying the MySQL database and sessions. Below is a system overview diagram which demonstrates the functions of each page and how it fits within the hierarchy of the website.





Inputs

- Name, Email and Password *Create account*
- Email and Password *Login*
- Firstname, Lastname, Gender *Add Family Members*
- Members, Time Available *Create Meal Plan*

Processes

- Register
- Login
- Add family members
- Edit family members
- Delete family members
- Generate meal plan
- Delete current meal plan
- View recipe
- Calculate shopping list
- Add recipe (*extension objective*)

Storage

- SSD storage on webserver
- Session cookie stored on user's device

Outputs

- Day, Date, Recipe Name *Meal Plan*
- Ingredient, Category, Quantity *Shopping List*

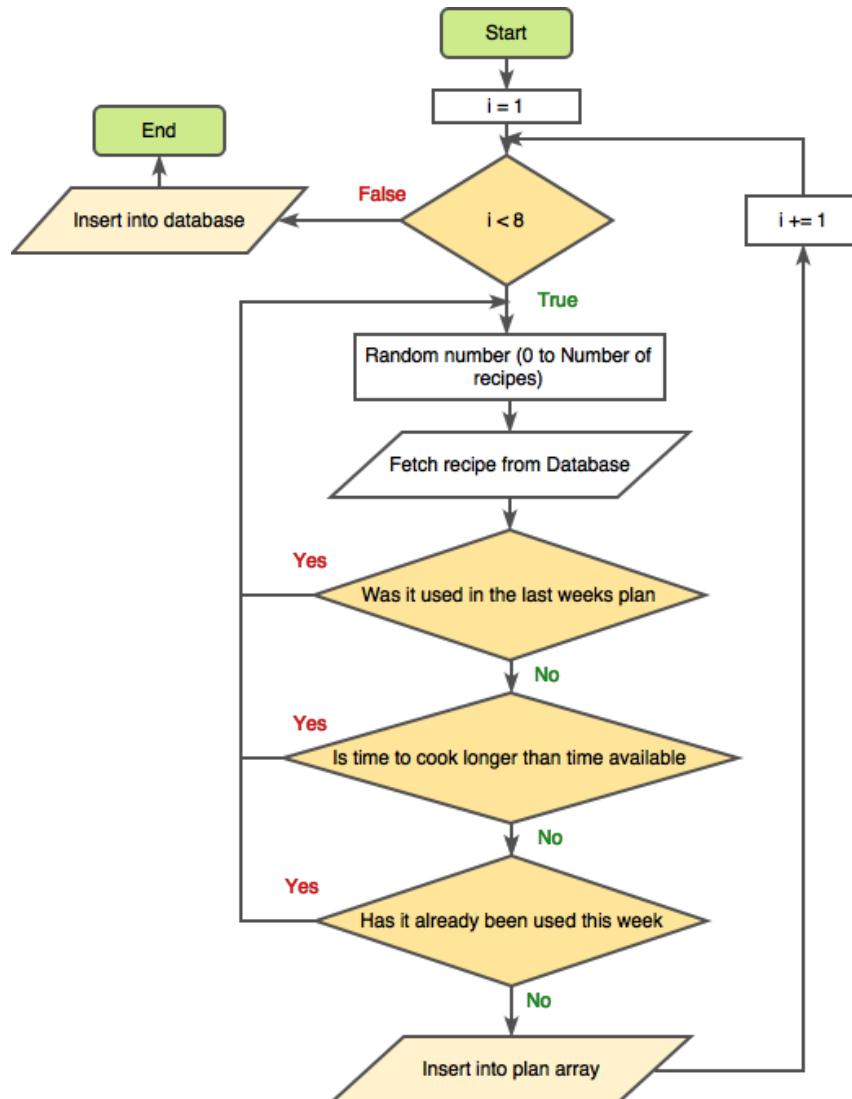
Now I have laid out the functions/operation that each page will perform and highlighted the inputs the system will be required to handle, I need to design my complex algorithms which are shown highlighted dark yellow in the diagram above.



Complex Algorithms

Recipe Selection

The most important aspect of my program is the algorithm tasked with choosing the recipe for each day of the meal plan. I need to make sure it is efficient and all requirements are checked in accordance to my generation objectives. This will be triggered each time the user clicks on the generate plan button on the plan page (page layout diagrams under HCI).



In order to improve efficiency rather than running a query on the entire table, which eventually could contain hundreds of records, I select a recipe at random to begin. This means I am only looking at one record at a time and if it does not fit the criteria I look at another random recipe.



The diagram below demonstrates how I can make use of a for loop to select a recipe for each day and store it in the plan table of my database.

This is an extremely simplified version of the algorithm as I will need to fetch the dates of each day of the week to add to the database, run multiple queries to fetch the relevant information and manage adding the plan to the database across three tables to set up the relationships. Below I have represented my algorithm in pseudo code to demonstrate some of the extra tasks it will need to perform, not shown in the flowchart representation.

Pseudo Code:

```
user_id = session ID
day = array(0 = monday, 1 = tuesday, 2 = wednesday ... for each day)

for i = 0 to 7
    Store user submitted values from plan page
    (day[i]) = array(members eating that day)
    time[i] = time for that day
    no_eating[i] = number of people eating that day → count((day[i]))
End
```

```
Set up plan dates
date_eating = array(next monday to sunday)
```

```
Get the details of users last plan
Fetch largest plan_ID from DB
If plan_ID > 0 then
    Fetch start_date from DB
    If start_date = date_eating[0]
        If there is already a plan for that week check if the user wants to override it
        Jump to removeplan.php
    End if
    recipe_history = array(Fetch recipes from plan)
End if
```

```
Set upper bound for random number generation
no_recipes = count of recipes from DB
```

```
for i = 0 to 7
    accept = false
    While accept = false do
        accept = true
        recipe = random(1, no_recipes)
        For x = 0 to x < (i+1)
            Check the recipe was not in last weeks plan
            If recipe = recipe_history[x]
                accept = false
```

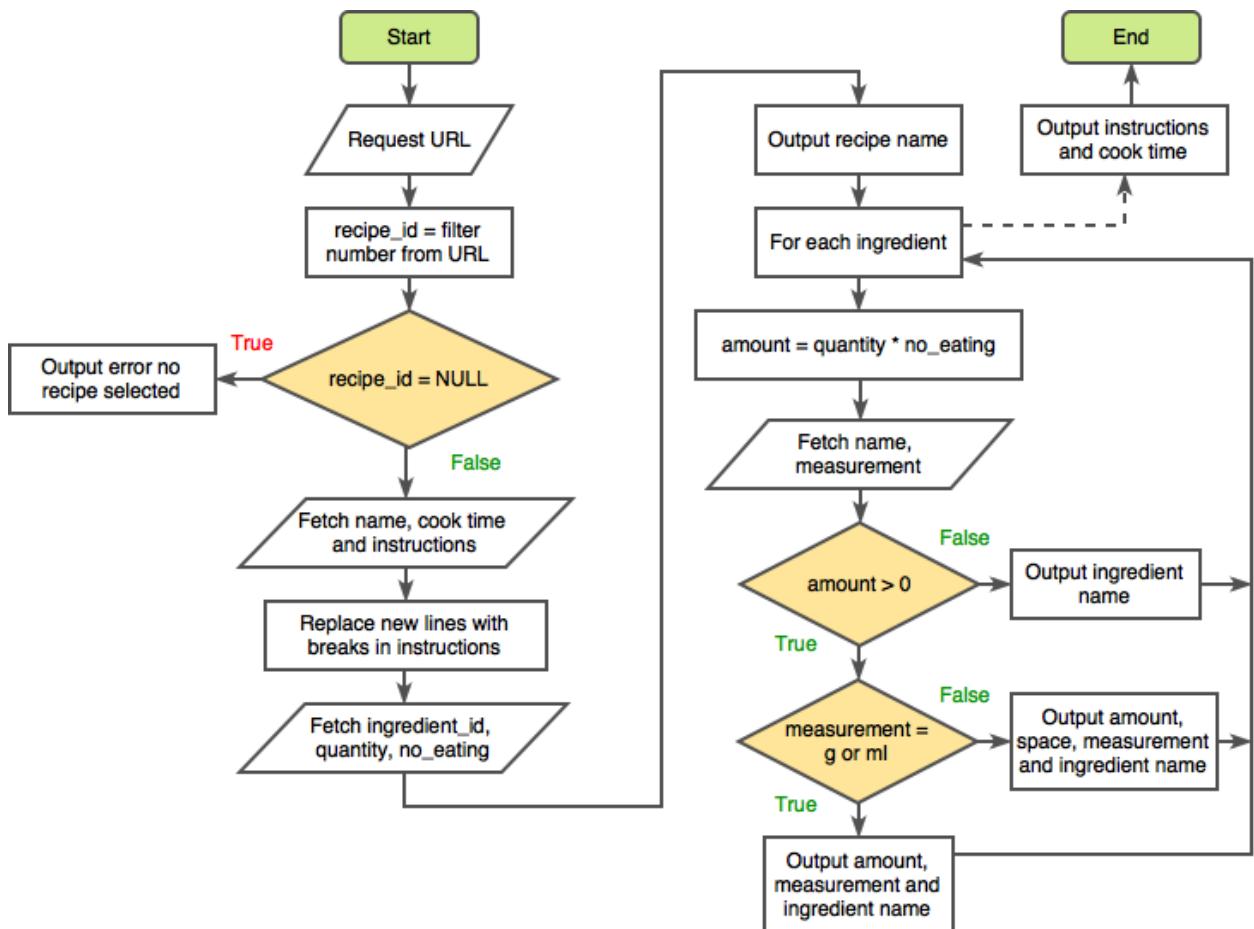


```
End if
End
Check the user has long enough to cook it
Fetch cook_time from DB
If time[i] < cook_time
    accept = false
End if
For q = 0 to q < (i+1)
    Check if the recipe has already been used that week
    If recipe = plan[q]
        accept = false
    End if
End
Add recipe to the plan array
plan[i] = recipe
Loop
End
Insert into the database
Insert start_date, end_date into plan table
Insert user_id, plan_id into users_plan table
For x = 0 to 7
    Insert plan_id, plan[x], no_eating[x], date_eating[x] into plan_recipes
End
```



Display Recipe

Although most of my algorithms are separated into my include folder (shown in web server file structure) however I decided to keep the PHP responsible for formatting the recipe page inside dayrecipe.php. This is because there are so many small queries and php calculations intertwined with the HTML it would be unnecessary to spread them across many extra files. The flowchart for the process is shown including the effect it has on the output.





Pseudo Code:

Fetch the recipe ID from the URL

url = request URL

recipe_id = filter(url, integer)

If recipe_id = NULL

 Output 'Error select the recipe you want to view on the calendar page'

Else

 Fetch name, instructions, cook_time, ingredient_id[], quantity[], no_eating[] from DB

 Replace new line with break(instructions)

 Output recipe_name

 amount = array

For i = 0 to count(ingredient_id)

 amount[i] = quantity[i] * no_eating

 Fetch ingredient_name, measurement from DB

If amount[i] > 0

If measurement = 'g' or 'ml'

 Output amount[i] . measurement[i] . ' ' . ingredient_name

Else

 Output amount[i] . ' ' . measurement[i] . ' ' . ingredient_name

End if

End if

 Output ingredient_name

End

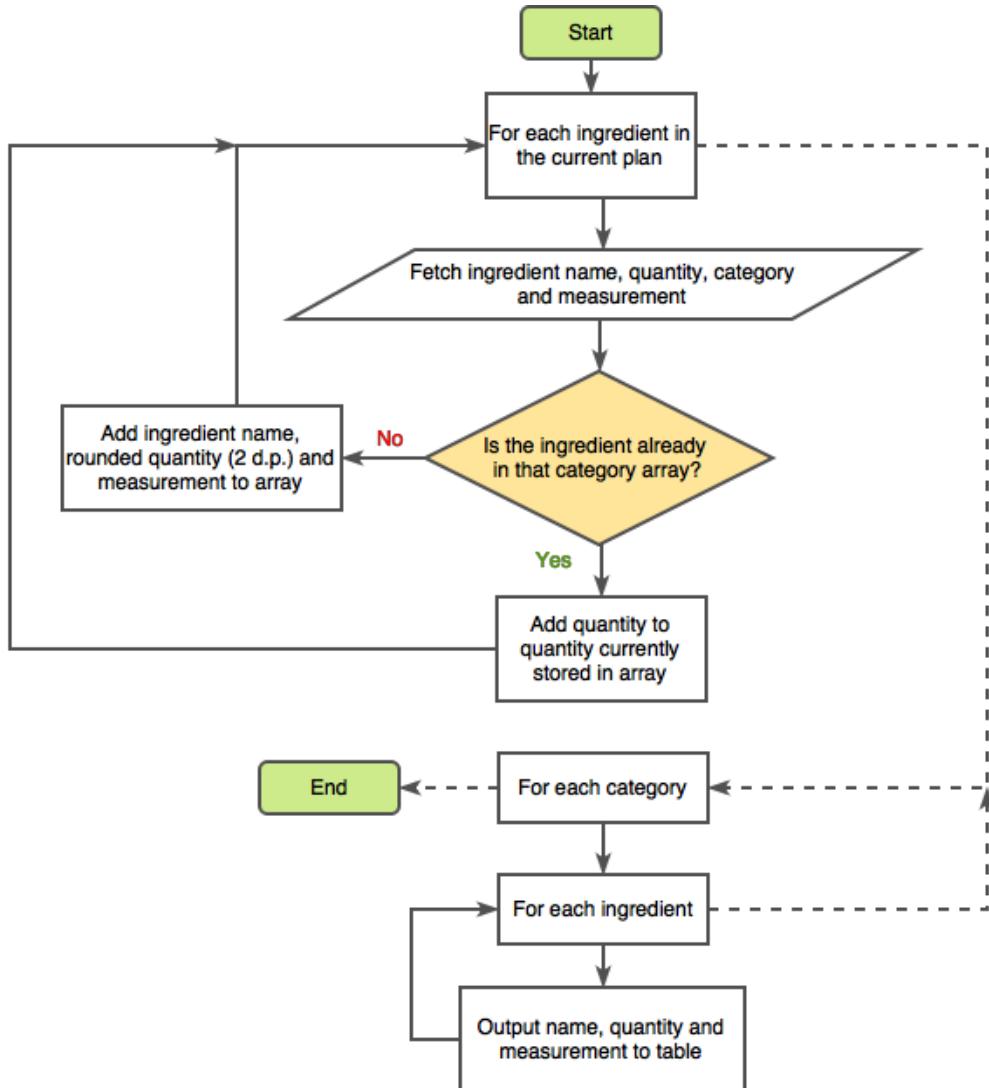
 Output instructions, cook_time

End if



Shopping List

Another important algorithm is tasked with setting up the user's shopping list. This needs to be able to find the user's latest plan. It then needs to find every ingredient included in each recipe for that plan, including the quantity and display the ingredients categorised to the user. In order for this to be possible they need to be separated into arrays for each category.





Pseudo Code:

```
Set up arrays
Fruit and Veg = 1
fv_name = array
fv_amount = array
fv_measure = array
Protein = 2
pr_name = array
pr_amount = array
pr_measure = array
Carbohydrates = 3
ca_name = array
ca_amount = array
ca_measure = array
Dairy = 4
da_name = array
da_amount = array
da_measure = array
Ignore = 5 ➔ Used for ingredients like water that need to be included in the recipe but not the shopping list
Uncategorised (Other) = 6
un_name = array
un_amount = array
un_measure = array
```

Complex SQL query detailed in example queries

Fetch ingredient_name, category, SUM(no_eating * quantity) as amount, measurement group by ingredient_name from DB

While returning query results

```
If category = 1
    push(fv_name, result['ingredient_name'])
    push(fv_name, round(result['amount'], 2))
    push(fv_name, result['measurement'])

Elseif category = 2
    push(pr_name, result['ingredient_name'])
    push(pr_name, round(result['amount']))
    push(pr_name, result['measurement'])

Elseif category = 3
    push(ca_name, result['ingredient_name'])
    push(ca_name, round(result['amount']))
    push(ca_name, result['measurement'])

Elseif category = 4
    push(da_name, result['ingredient_name'])
    push(da_name, round(result['amount']))
    push(da_name, result['measurement'])

Elseif category = 6
```



```
    push(un_name, result['ingredient_name'])
    push(un_name, round(result['amount']))
    push(un_name, result['measurement'])

End if
End
```

OutputArray is used to output the ingredients in each cell of the HTML table on list.php by passing through different arrays

Function OutputArray (amount, measurement, name, half)

For i = 0 to count(name)

Half is only passed through for the uncategorised field on mobile as it splits it over two fields to make it mobile friendly

If half > 0 and i = half

 Add an extra cell to the table

End If

If amount[i] > 0

 Conditional formatting to remove unnecessary spaces for grams and ml

If measurement[i] = 'g' or 'ml'

 Output amount[i] . measurement[i] . ' ' . name[i]

Else

 Output amount[i] . ' ' . measurement[i] . ' ' . name[i]

End if

Else

 Output name[i]

End if

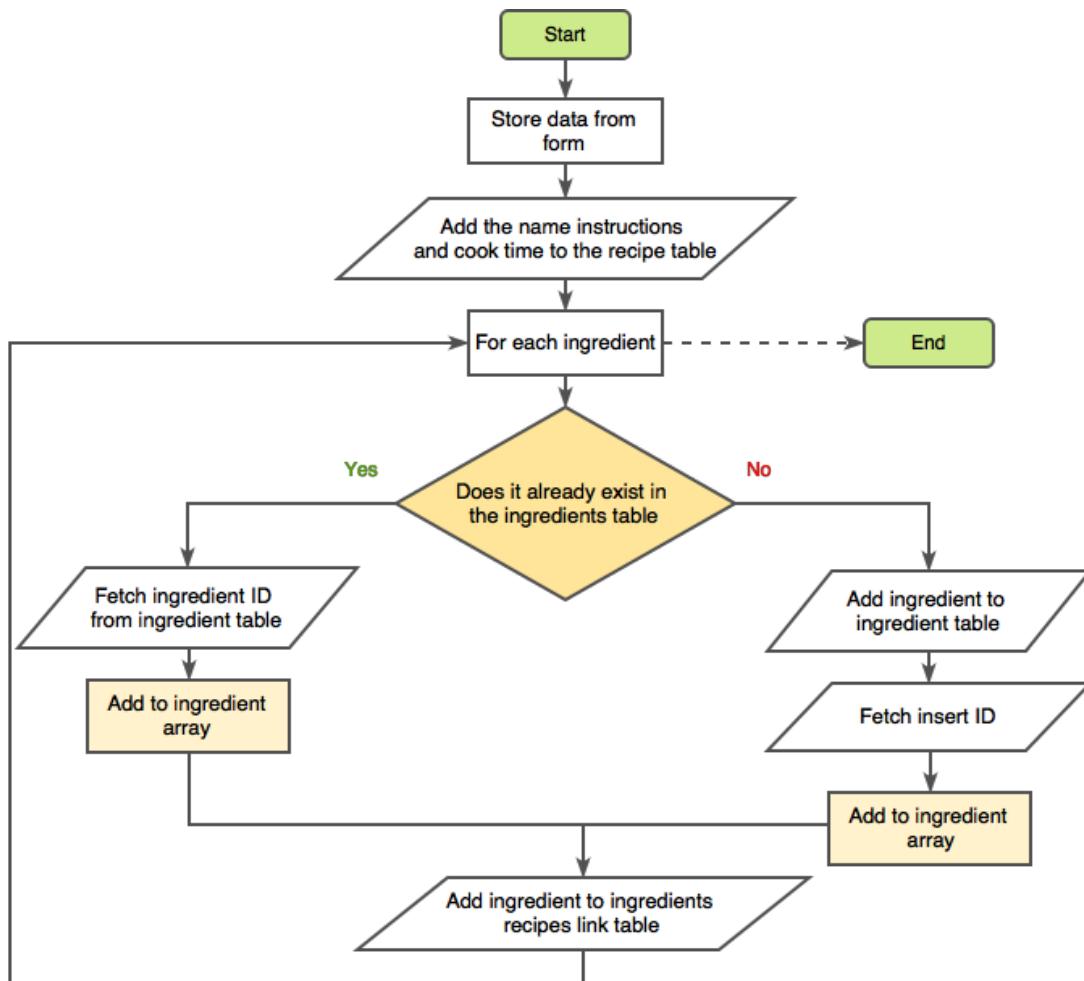
End



Add Recipe

Although an extension objective, the ability to add recipes is really important for my program. This will enable me to give others an easy way to add their own recipes which will help grow my recipe selection. It will also make it easier for me to add recipes rather than running many separate SQL queries to insert the information manually.

Giving any user this ability could cause major issues as it is very hard to validate the information submitted which could be selected in people's meal plans. The aim of this algorithm is to set up the relationships correctly in the database and avoid repeated ingredients being added to the ingredient table. This page will also only be available to admin users which are given account permissions through the admin column in the user table.





Pseudo Code:

Store user submitted values from admin page form

name = recipe name

cook_time = recipe cook time

instructions = recipe instructions

ingredient_name = array(ingredient names)

measurement = array(ingredient measurements)

quantity = array(ingredient quantities)

category = array(ingredient category)

Add the recipe information that doesn't need validation to the database

Insert name, instructions, cook time into recipes

For i = 0 to count(**ingredient_name**)

Check if there is already an ingredient with the same name in the database

Fetch **ingredient_id** where name LIKE **ingredient_name[i]**

If result returned

ingredient_id = result

Else

 Insert **ingredient_name[i]**, **measurement[i]**, **category[i]** into ingredients

ingredient_id[i] = insert id

End if

 Insert **ingredient_id[i]**, **recipe_id**, **quantity[i]** into ingredients_recipes

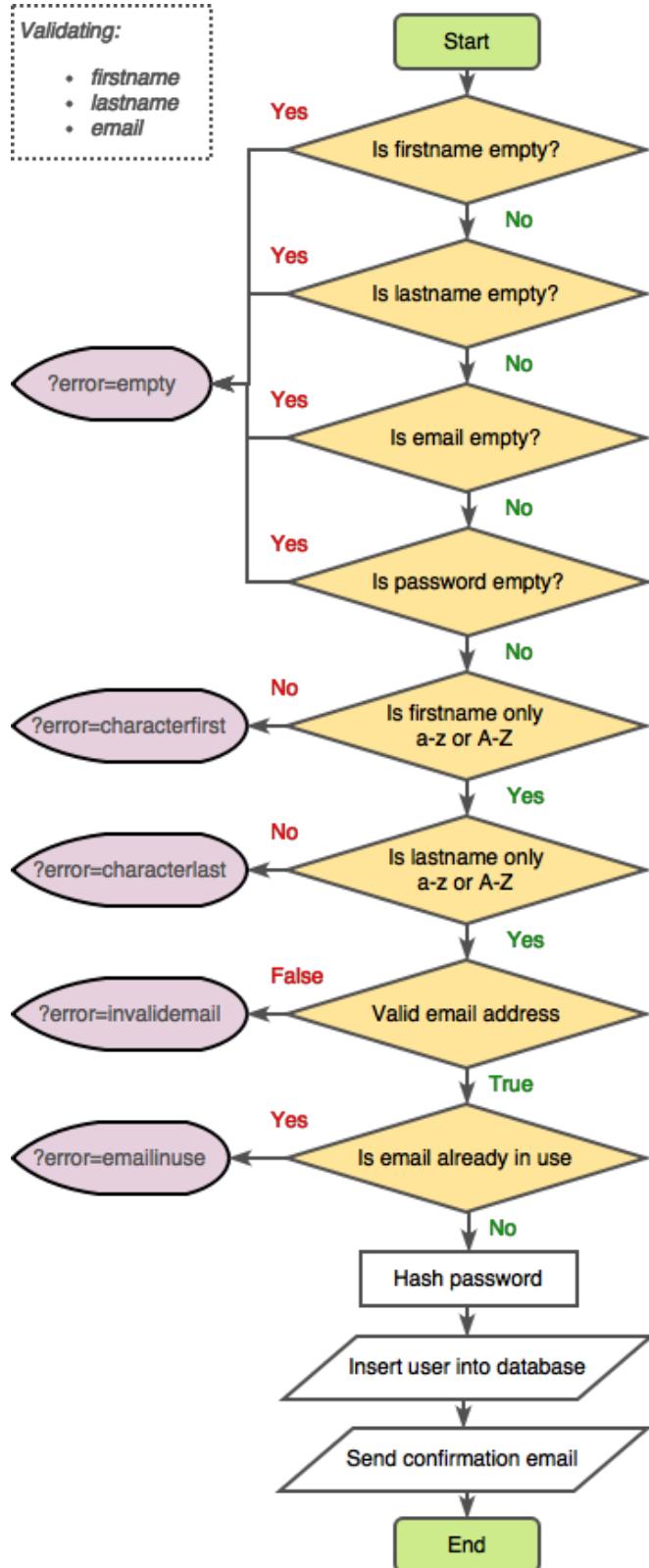
End

Validation Required

Registration Form Validation

It is essential that account creation is simple and that all forms are validated to stop incorrect data being submitted to the database. To do this a chain of if statements will check the data using in built in PHP functions and RegEx. This is important not just for the database to check there is no information missing but also the user who could have mistyped details such as their password or email which would make them unable to access their new account.

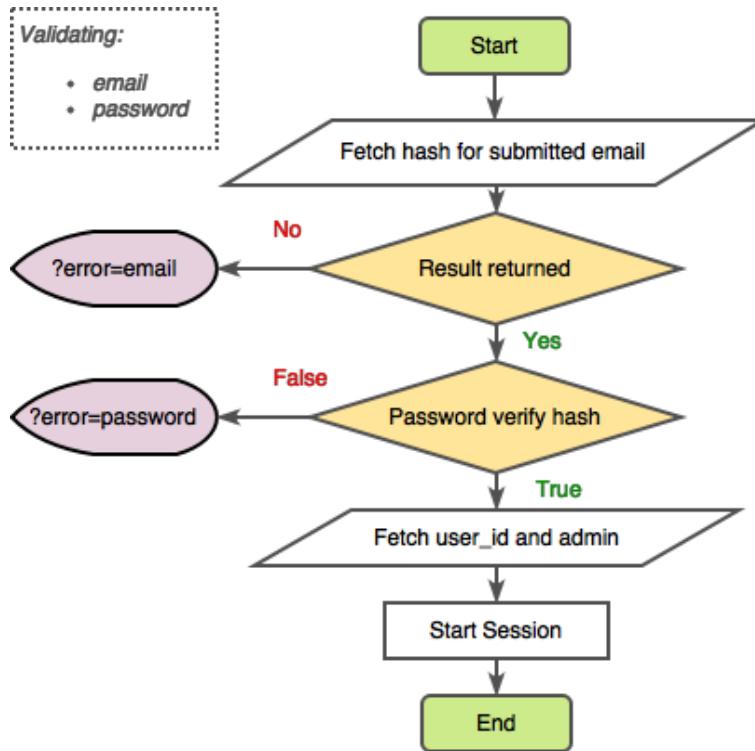
Below a flowchart shows the order in which I will be validating the submitted data and the error code which will be returned through the URL and trigger a specific error message on the register page.





Login Form Validation

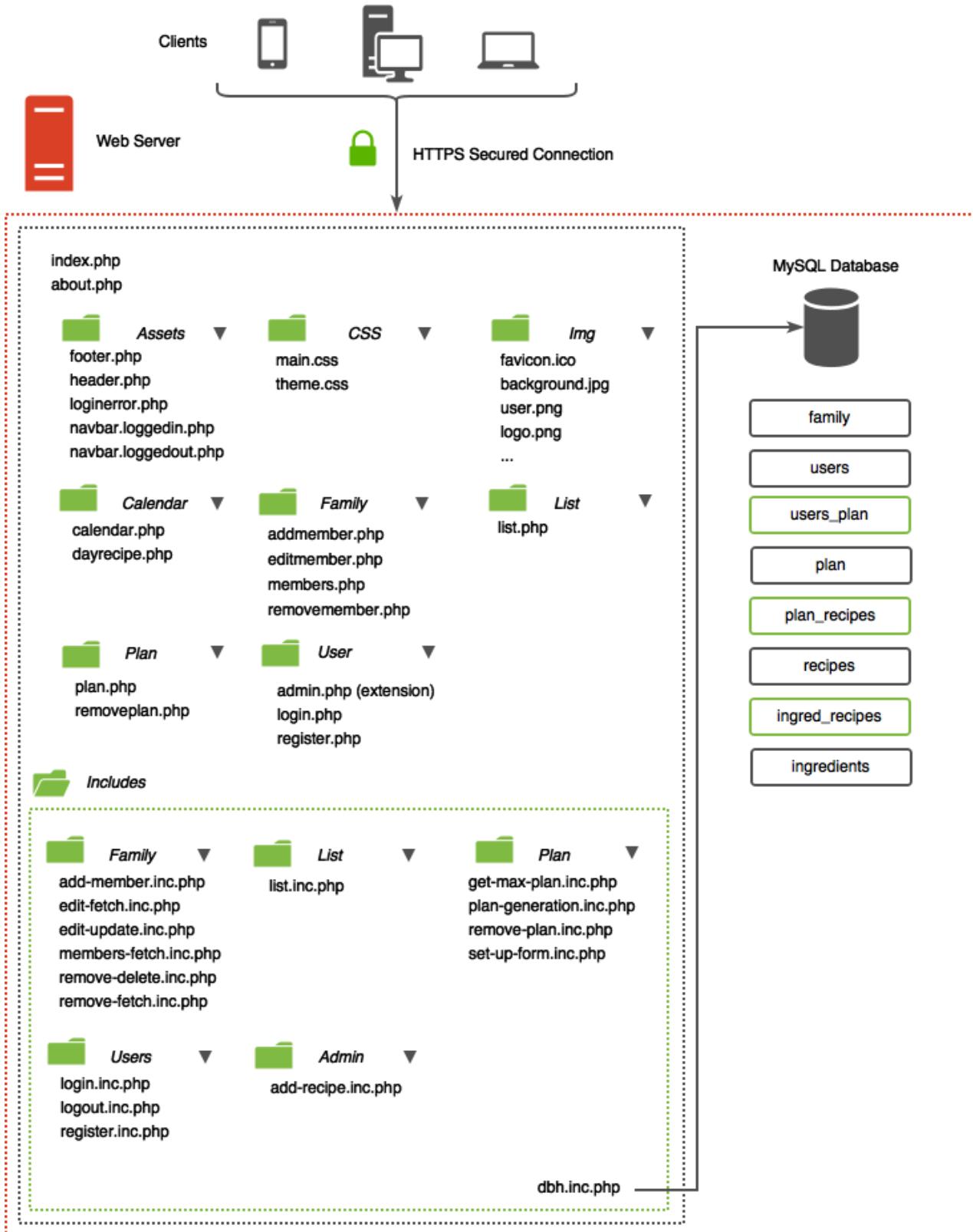
It isn't quite as important to validate the data submitted to a login form as it is when registering as the data will only be used to query the database. However, telling the user whether their password was incorrect or their email was the issue is extremely helpful as most people have multiple email addresses and password combinations they use across different sites.



Web Server File Structure

To ensure my website is expandable in the future and separate out my code to make it easy to manage and troubleshoot, I spent a long time designing my web server file structure. This enabled me to separate out, where possible, my PHP scripts from the bulk of my HTML. Meaning I can find specific scripts more easily and reuse them on other pages.

I also decided to separate out the files into folders which reflect the navigation bar. Keeping all the pages for the user's family eg: Edit Family Member, Add Family Member in one folder meant only the home page (index.php) and the about page will actually be stored in my root directory. This has the positive of keeping everything organised but does create some difficulties in linking files together and including scripts. I need to ensure I finalise files locations as if their file path changes it will break my links and cause issues.





Database

Definition of Data Requirements

A database is essential to support my site to make it easy to manage multiple users, families, plans and recipes and link them all together. Before I start building the database I need to ensure my design is capable of handling the volumes and types of data I will need to store. It is really important I store each element correctly as this will affect my ability to create indexes and foreign key constraints if types do not match. Below I have laid out the structure of each table including the datatype of each field. I have also included examples of the data that will be stored in these tables to make it easier to understand the role each table plays.

An entity relationship diagram for my fully normalised design can be found under the modelling section.

Key:
<u>Primary Key</u>
<u>Foreign Key</u>

Users

Structure

Name	Type	Auto Increment
<u>user_id</u>	mediumint(8)	True
first_name	varchar(20)	
last_name	varchar(40)	
email	varchar(60)	
hash	varchar(255)	
registration_date	datetime	
admin	<i>tinyint(1)</i>	



Jack Corbett

Data Example

<u>user_id</u>	first_name	last_name	email	hash	registration_date	admin
1	Jack	Corbett	jack@example.com	\$2y\$10\$ozTf23s7/e...	1/12/2016 10:05:25	1
2	Zoe	Hunt	example@email.com	\$2y\$10\$9hsab...	4/12/2016 13:50:20	0

Family

Structure

Name	Type	Auto Increment
<u>member_id</u>	mediumint(8)	True
<u>user_id</u>	mediumint(8)	
first_name	varchar(20)	
last_name	varchar(40)	
gender	enum(m, f, o)	
default_ticked	<i>tinyint(1)</i>	

Data Example

<u>member_id</u>	<u>user_id</u>	first_name	last_name	gender	default_ticked
1	1	Jack	Corbett	m	1
2	1	Beverley	Corbett	f	1
3	1	Stephen	Corbett	m	0
4	2	Zoe	Hunt	f	1



Users-Plan

Structure

Name	Type	Auto Increment
<u>user_id</u>	mediumint(8)	
<u>plan_id</u>	mediumint(8)	

Data Example

<u>user_id</u>	<u>plan_id</u>
1	1
2	1

Plan

Structure

Name	Type	Auto Increment
<u>plan_id</u>	mediumint(8)	True
start_date	date	
end_date	date	

Data Example

<u>plan_id</u>	start_date	end_date
1	5/12/2016	11/12/2016
2	5/12/2016	11/12/2016
3	12/12/2016	18/12/2016

Plan-Recipes

Structure

Name	Type	Auto Increment
<u>plan_id</u>	mediumint(8)	



<u>recipe_id</u>	mediumint(8)	
no_eating	int(8)	
date_eating	date	

Data Example

<u>plan_id</u>	<u>recipe_id</u>	no_eating	date_eating
1	5	2	5/12/2016
1	2	4	6/12/2016
1	7	1	7/12/2016
1	10	4	8/12/2016
1	9	4	9/12/2016
1	4	3	10/12/2016
1	15	4	11/12/2016
2	4	6	5/12/2016

Recipes

Structure

Name	Type	Auto Increment
<u>recipe_id</u>	mediumint(8)	True
name	mediumint(8)	
instructions	int(8)	
cook_time	int(11)	

Data Example

<u>recipe_id</u>	name	instructions	cook_time
1	Tomato Sauce with Penne	1. Heat the oil in a pan...	25
2	Chicken Noodles	1. First heat the oil in...	60



Ingredients-Recipes

Structure

Name	Type	Auto Increment
<u>ingredient_id</u>	mediumint(8)	
<u>recipe_id</u>	mediumint(8)	
quantity	decimal(5,2)	

Data Example

<u>ingredient_id</u>	<u>recipe_id</u>	quantity
1	1	0.50
2	4	0.50
3	8	0.25
3	10	1.50

Ingredients

Structure

Name	Type	Auto Increment
<u>ingredient_id</u>	mediumint(8)	True
ingredient_name	varchar(25)	
measurement	varchar(20)	
category	tinyint(4)	

Data Example

<u>ingredient_id</u>	ingredient_name	measurement	category
1	Coconut Oil	tbsp	6
2	Onion(s)		1
3	Garlic	cloves	1



Querying the Database

To fulfil the responsive design of my webpage I will need to run many basic queries to fetch information about everything from family members names to the latest meal plan for a user's account. I will also need to insert information for new users and family members as well as being able to update information stored about family members. To do this I will need to use a range of SQL commands.

I will run these queries using PHP meaning all will contain variables which contain the result of calculations or have been submitted by the user. Below is a basic query used to add a family member. Member_id is not inserted as this is an auto incrementing field.

```
INSERT INTO family (user_id, first_name, last_name, gender, default_ticked)
VALUES ('$id', '$firstname', '$lastname', '$gender', '$default_ticked')
```

Aggregate functions will also be extremely helpful when running queries to ensure I only return the information required and limit the amount of PHP code which is needed to extract it from the results. An example of this is using MAX to check a user's latest plan. This enables me to treat plans like a stack by using a FIFO approach.

```
SELECT MAX(plan_id) plan_id FROM users_plan WHERE user_id = '$id'
```

Finally, some features of the site will require much more complex queries that combine aggregate functions, grouping and joins to fetch the required data. Below is an example query to be used to construct the user's shopping list.

```
SELECT ingredients.category, ingredients.ingredient_name,
SUM(plan_recipes.no_eating * ingredients_recipes.quantity) AS amount,
ingredients.measurement FROM plan_recipes
JOIN recipes ON recipes.recipe_id = plan_recipes.recipe_id
JOIN ingredients_recipes ON ingredients_recipes.recipe_id = recipes.recipe_id
JOIN ingredients ON ingredients.ingredient_id = ingredients_recipes.ingredient_id
WHERE plan_recipes.plan_id = '$plan_id'
GROUP BY ingredients.ingredient_name
ORDER BY ingredients.category
```



Programming

Language Choice

As I have talked about in my research I did initially consider creating an app written in Swift but decided against it due to having limited device support which was why I settled on creating a web app. To achieve this HTML, CSS, PHP and SQL were the obvious choices when it came to creating a dynamic website.

Javascript will also help me achieve the scalability I need. I plan to use this to create modals, tab linked lists and allow an interactive navigation bar. I also plan to add jQuery to act as a high level interface to perform AJAX requests to enable the user to interact with the page without it needing the refresh. This will be especially helpful when adding recipes to let the user add extra fields to the recipe form, depending on how many ingredients are needed.

Libraries

I only plan to use one library in my project which is w3schools CSS template. This will save me a huge amount of time which I can instead dedicate to growing the functionality of my website.

User Interface Design (HCI)

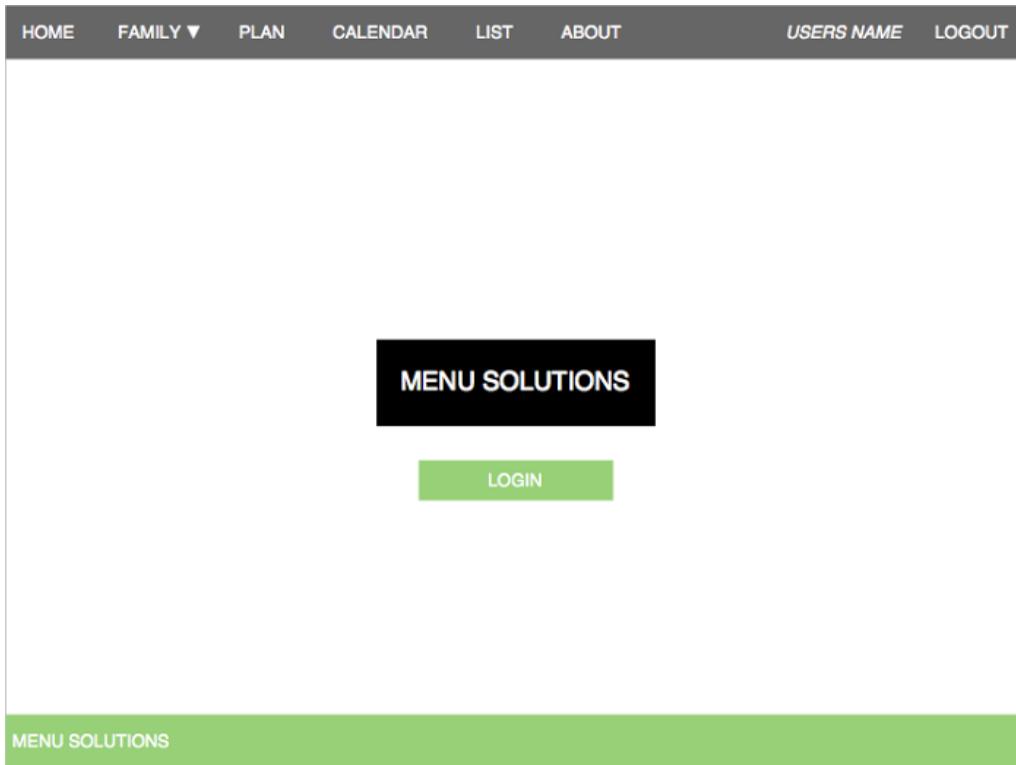
When designing the user interface of my web app it was important I considered how it would scale to mobile and tablet aspect ratio devices. To ensure I got the experience right on both I designed them separately to maximise the screen real estate of the users device.



Jack Corbett

Desktop UI Diagrams

The home page reflects the web designs I found during my research, with a navigation bar across the top of the window, full size background image (demonstrated by the white space as the image is yet to be decided), call to action in the centre along with the name of the web app and a simple footer at the bottom to frame the page. In this mockup the logged in navigation bar is displayed giving the user access to their family, plan, calendar and shopping list. It also displays the user's name in the top right so they can see that they are logged in along with the logout button. The admin page is hidden from the navigation bar in this example as the user does not have admin privileges. If they do an ADMIN options will appear alongside the about page.



If a user hasn't logged in yet they will only have access to the home and about pages. The logout button will also be replaced with a link to the login page where the user can either login to their previously created account or register for one.

Below is a diagram for the list page. I chose this to demonstrate my UI as the way the table is displayed changes dramatically depending on the size of the client's device. I wanted to keep all the pages simple, especially the calendar and list pages which will be used most. That is why for the list page I structured it as a simple table, allowing the user to see all the information in one place with helpful categories to make sure they don't miss anything in the supermarket.



Ingredient quantities will be included on the actual page

HOME FAMILY ▾ PLAN CALENDAR LIST ABOUT USERS NAME LOGOUT

LIST

Page Description.

Fruit and Vegetables	Protein	Carbohydrates	Dairy	Other
Beansprouts Peas Carrots Yellow Peppers Dried Parsley Baby Corn Spinach	Chicken Breasts Turkey Mince Beef Mince	Fast Action Yeast Brown Penne Pasta Burger Bun Plain Flour Noodles Wholemeal Wrap	Yoghurt Cheddar Cheese Parmesan Cheese	Oregano Tomato Puree Coconut Oil Oyster Sauce Garlic Butter Mixed herbs Caster Sauce Tabasco Sauce

MENU SOLUTIONS

Although I am going to apply a similar design approach to my other pages they will involve added complexity as more functionality is required to set up your meal plan than display a shopping list table.

The family page will have a list of members complete with a small image displaying their gender to make it easy to identify members and a pencil icon for editing their details and an x for deleting them. At the bottom of this list there will be a + button which will let them add new members or they can achieve this through the dropdown menu in the navigation bar.

For the plan page it is going to be impossible to fit an identical form for who is eating and how much time the user has available to cook without cluttering up the interface. To get around this I am going to have a separate tab link for each day. Each member will then get a box with their name, gender image and a tick box. Below this will be a drop down the choose from a range of available times.

To view the calendar I will employ a similar list design to the family page with the date, day and name of the recipe on the left side and on the right a small image of a platter repeated for the number of members eating. This makes it easy to see on each day how many people the recipe

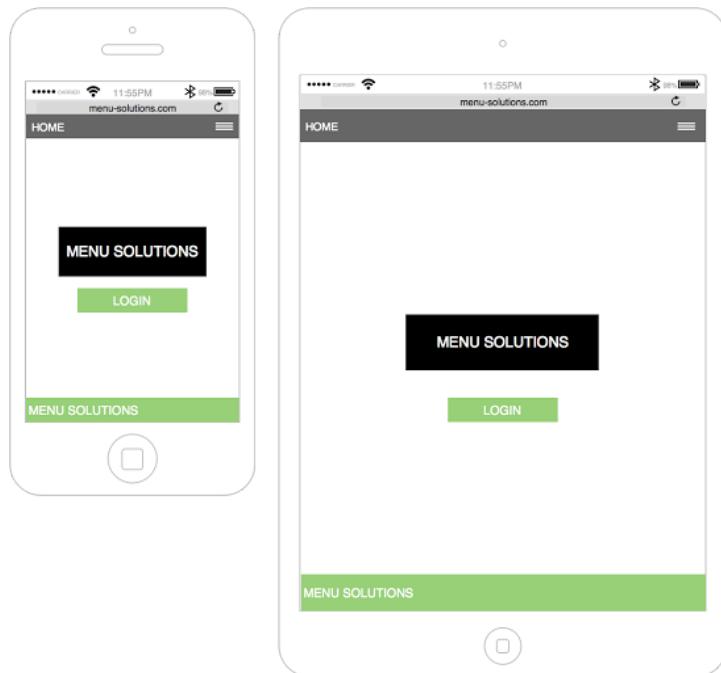


Jack Corbett

has been calculated for. By clicking the recipe name the user then reaches the recipe itself with the name at the top followed by ingredients, instructions and the cook time.

Finally the admin page (extension objective) will be much longer as it holds the largest form on the site. At the top it will provide guidance on how to submit your recipe followed by a simple form for inputting recipe name, cook time and instructions. Below this is a box for the first ingredient with inputs for it's name, measurement, quantity (per person) and category. To add more there is a button above it to duplicate the box allowing the user to add as many ingredients as they need.

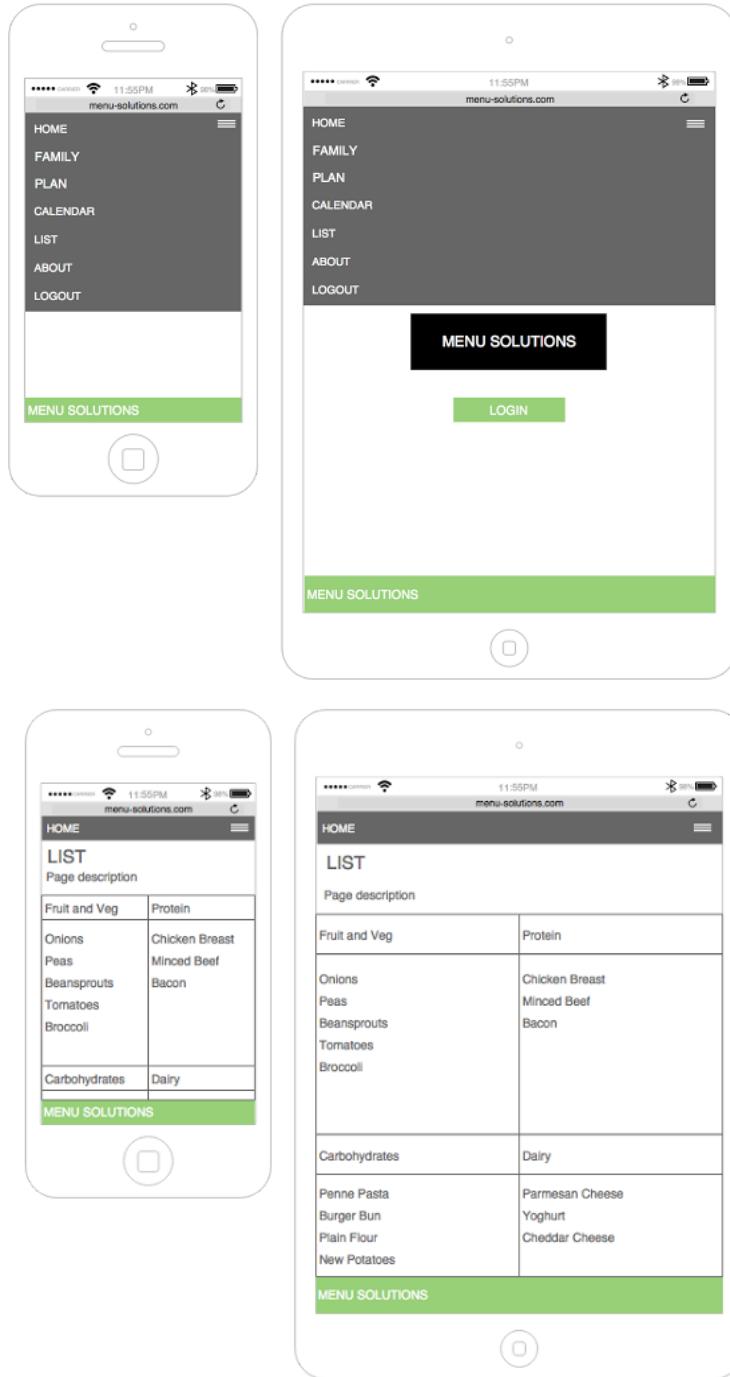
Mobile UI Diagrams



The mobile user interface for the home screen is very similar with the same background image with centralised logo and call to action. The big change on mobile and tablet devices in the navigation bar. As there simply isn't the space to span the pages across I opted for a drop down menu which is triggered by selecting the three bars in the top right of the window. This does mean the drop down option is not available on the family page, meaning the link takes the user directly to the members page where their family is listed and they can then use the + button to add members.



Jack Corbett



Most pages will be formatted very similarly to its desktop counterpart as I designed it to have very clutter on one page at a time. The tab links will simply spread down the page rather than across it while the lists will have minor tweaks such as only 5 platters per row on the calendar page to avoid getting in the way of the day and recipe name. The list page sees the biggest



changes as it will be formatted in two columns which is why I create an extra field in the OutputArray function to ensure there is an even number.

Data Integrity and System Security

Encryption

To encrypt the traffic between the web server and my clients I will use the HTTPS protocol which I will enforce using .htaccess. As my website is hosted an SSL certificate is provided by my hosting service. This not only help protect users data from practices like packet sniffing on public networks but Google's algorithm favours SSL secured sites so will make my site appear higher in search results.

I spent a long time researching hashing algorithms to protect users passwords that would be stored in my database. I considered just relying on Google and Facebook's sign in SDK's but decided that many of the users of my site may not have accounts/want to use them on an unknown site. I may add the option to use them in the future but it is important to have a standard sign in option as well. Originally I planned to use SHA256 encryption on the passwords submitted by users but after some research realised it would be possible for passwords to be compromised through dictionary attacks and eventually through brute force as every algorithm is eventually cracked. That is why I opted to use PHP's default encryption, this way if in the future bcrypt is compromised PHP will, in theory, change it's default standard hashing algorithm. This did mean I had to leave some extra space in my database to store the hash incase it's length increased in future but this is a worthy compromise to improve password protection.

Bcrypt is also a key stretching encryption algorithm meaning it would take a hacker much longer to test each key. Increasing its resistance to brute force attacks, where a computer will try many combinations to look for similar results.

Security

Besides protecting users passwords and securing their connection to my webserver I have also escaped user submitted strings from forms to avoid SQL injection. This is really important as my site is so reliant on the database and holds private information such as names of family members that need to be protected.



Prototype

As each system within my site follows on from the other I am going to need to take a linear approach to programming my project. However, to ensure I was able to achieve my project goals I decided to lay out some code to test how recipes could be selected. As this is the main focus of my site it was really important I was able to deliver it. This did require me to add the recipes to the database and set up some of the tables, but meant I could focus on this before creating my login system, family management, plan input and shopping list.

In order to test my code I set the time to cook to 30 minutes and stated that 2 people were eating each day. I did this in the arrays at the top of the code below. I then outputted each recipe that was randomly selected. This shows which recipes were not accepted into the final plan causing the code to loop again and select a new one. It then uses the chain of if statements to check if the recipe fits the requirements and outputs the final plan. After a while of fixing errors I was able to create the working prototype below which picked a recipe for each day of the week. I will still need to add a large amount of functionality to the script to allow me to check past plans, generate dates and add the plan to the database but this proves that my algorithm works as planned and allowed me to begin building the rest of my project.

Selected recipe_id: 3, 16, 10, 17, 6, 8, 4, 10, 14, 11, 7,

Weeks Plan:

3, 16, 10, 6, 8, 14, 7,

```
<?php  
  
include '../dbh.inc.php';  
  
$day = array(0 => 'monday', 1 => 'tuesday', 2 => 'wednesday', 3 => 'thursday', 4  
=> 'friday', 5 => 'saturday', 6 => 'sunday');  
$time = array(30, 30, 30, 30, 30, 30, 30);  
$no_eating = array(2, 2, 2, 2, 2, 2, 2);  
  
$sql = "SELECT COUNT(1) FROM recipes";  
$result = @mysqli_query($conn, $sql);  
$row = mysqli_fetch_array($result);  
$no_recipes = $row[0];
```



```
$plan = array();

echo "Selected recipe_id: ";

for($i=0; $i<7; $i++) {

$accept = false;

while ($accept == false) {
    $accept = true;

$recipe = rand(1,$no_recipes);
echo $recipe . ", ";

for ($a=0; $a<7; $a++) {
    if ($recipe == $recipe_history[$a]) {
        $accept = false;
    }
}

$sql = "SELECT cook_time FROM recipes WHERE recipe_id = '$recipe'";
$result = @mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
$cook_time = $row['cook_time'];

if ($time[$i] < $cook_time) {
    $accept = false;
}

for ($q=0; $q<$i+1; $q++) {
    if ($recipe == $plan[$q]) {
        $accept = false;
    }
}

$plan[$i] = $recipe;
unset($recipe);
}
}

?>
<br />
<h1>Weeks Plan: </h1>
<p><?php
for ($i=0; $i<7; $i++) {
    echo $plan[$i] . ", ";
}
?>
```



System Testing

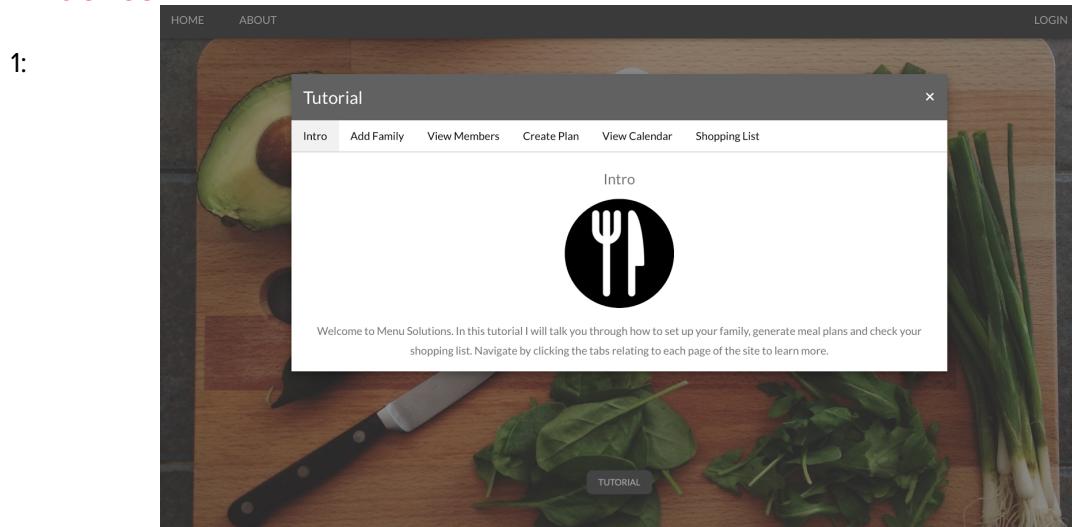
Testing overview and demonstration of functionality: <https://youtu.be/Wu6x59nYJoU>

Concludes with end user testing, where I set tasks for my user to carry out on her tablet.

Overall

Test No.	Description	Expected Result	Pass/Fail
Normal			
1	Tutorial modal pop up	When the tutorial button is selected a modal pops up with tabs containing instructions for each page.	Pass
2	Navigation bar when the user is logged out	The logged out navigation bar with limited options is displayed.	Pass
3	Navigation bar when the user is logged in	The logged in navigation bar displays with full page selection.	Pass
4	Access is blocked to family, plan, calendar and admin pages if not logged in	Error message you need to login to access this page with link to login page.	Pass
5	Home page shows extra buttons when user logged in	Button links appear under page descriptions on home page when logged in.	Pass

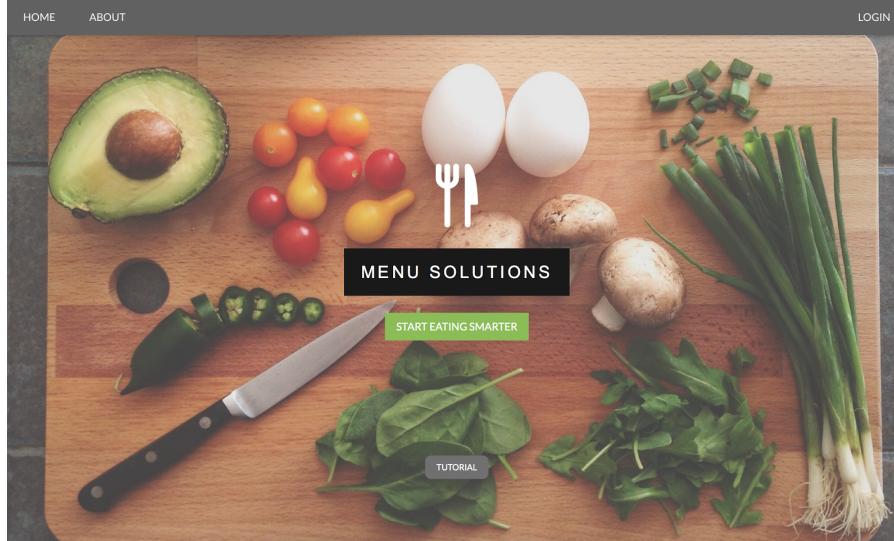
Evidence



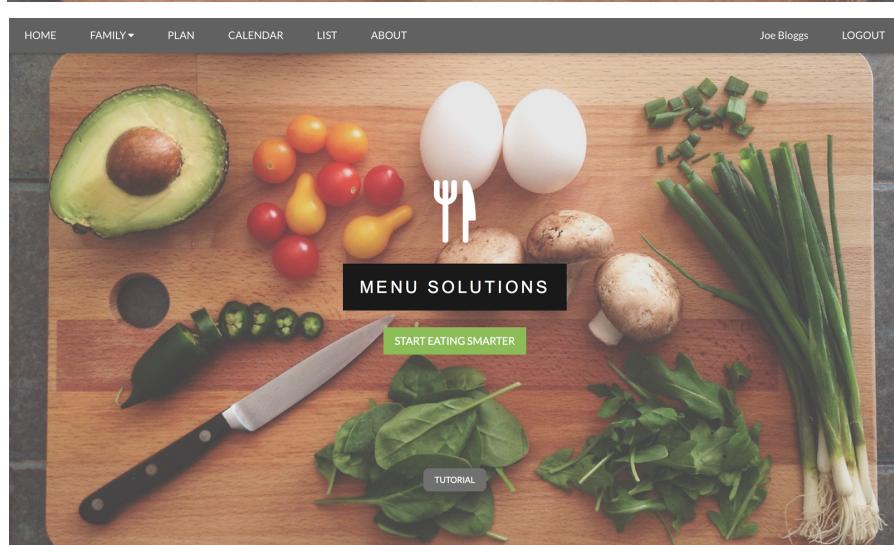


Jack Corbett

2:



3:



4:



You need to login to access this page

[Login](#)



Jack Corbett

5:

Simple Steps to a Healthy Diet

Add Your Family
Tell Us When
Generate Calendar
Shopping List

Add each family members details
Tell us how much time you have to cook each night and who will be eating
Review your calendar of meal plans for the week
Use your generated shopping list to fetch the ingredients and enjoy a week of stress free cooking

MENU SOLUTIONS - JACK CORBETT

Simple Steps to a Healthy Diet

Add Your Family
Tell Us When
Generate Calendar
Shopping List

Add each family members details
Tell us how much time you have to cook each night and who will be eating
Review your calendar of meal plans for the week
Use your generated shopping list to fetch the ingredients and enjoy a week of stress free cooking

FAMILY
 PLAN
 CALENDAR
 LIST

MENU SOLUTIONS - JACK CORBETT

Account

Register

Test No.	Description	Test Data	Expected Result	Pass/Fail
Normal				
6	Complete all fields with valid data	Joe, Bloggs, joe@example.com , 1234	Return to login page, new user created in database	Pass
Erroneous				
7	All fields empty	-	Error please fill in every field	Pass
8	Firstname empty	-, Bloggs, joe@example.com ,	"	Pass



Jack Corbett

		1234		
9	Surname empty	Joe, -, joe@example.com , 1234	"	Pass
10	Email empty	Joe, Bloggs, -, 1234	"	Pass
11	Password empty	Joe, Bloggs, joe@example.com , -	"	Pass
12	Numeric character in firstname	1Joe, Bloggs, joe@example.com , 1234	Error only letters and spaces allowed	Pass
13	Numeric character in lastname	Joe, 1Bloggs, joe@example.com , 1234	"	Pass
14	No @ in email	Joe, Bloggs, joeexample.com , 1234	Invalid email address	Pass
15	No email domain	Joe, Bloggs, joe@ , 1234	"	Pass

Evidence

6:

Joe

Bloggs

joe@example.com

[Sign Up](#)

Email

Password

[Login](#)

If you don't already have an account you can register for one below

[Register](#)

7,8,9,10,11:

Register

50



Jack Corbett

12:

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Sign Up

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Sign Up

Only letters and spaces allowed in first name

13:

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Sign Up

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Sign Up

Only letters and spaces allowed in surname

14:

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Sign Up

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Sign Up

Invalid email address



15:

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Register

Here you can register for your Menu Solutions account, simply fill in the form below and press sign up

Invalid email address

Login

Test No.	Description	Test Data	Expected Result	Pass/Fail
Normal				
16	Complete fields with account details	joe@example.com , 1234	Return to home page users name in the navigation bar to show login successful.	Pass
Erroneous				
17	Email empty	-, 1234	Invalid email address	Pass
18	Password empty	joe@example.com , -	Password entered was incorrect	Pass
19	Check submitted strings are escaped to avoid SQL injection	joe@example.com AND 0 = sleep(10), 1234	Invalid email address	Pass



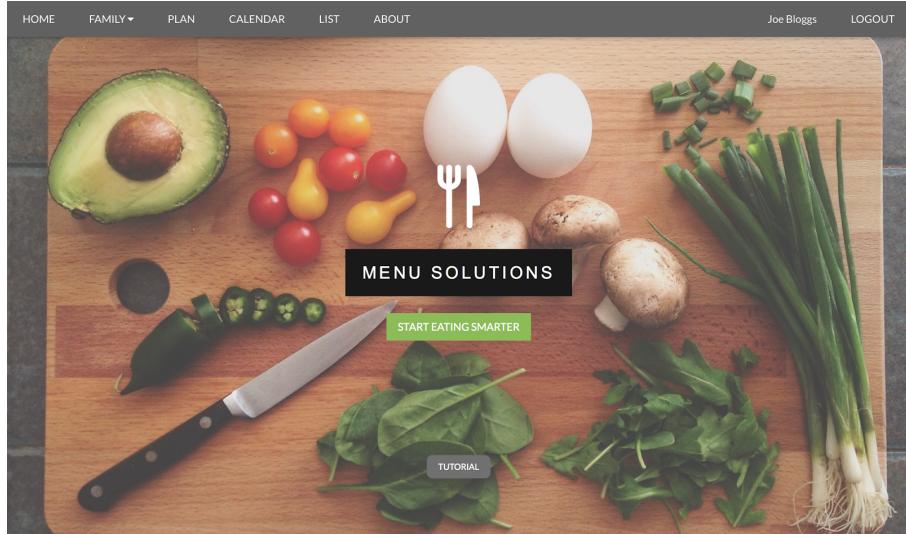
Jack Corbett

Evidence

16:

 joe@example.com

 ••••



17, 19:

Login



Email

18:

Email

This email has not been registered for an account

Login

The password you entered was incorrect

Login

Family

Test No.	Description	Expected Result	Pass/Fail
Normal			
20	Selecting add member from navbar family drop down	Taken to add member page to input a family member's details	Pass
21	Pressing + button on family page	"	Pass
22	Member is added to the database by submitting add member form	Members name appears on family page	Pass
23	Selecting the pencil on the family	Populates a form with the	Pass



	page	members details to edit.	
24	When the edit member form is submitted the details in the family page are updated	The information for that member stored in the database is changed.	Pass
25	Selecting the x for a family member	Asks if you want to delete the member and gives their name.	Pass
26	Confirming deletion of a family member	The member is removed from the family page list	Pass
Boundary			
27	Adding a family member with the same details	Adds the member to the family listing	Pass
Erroneous			
28	Leaving add member fields empty	Error please complete both first name and surname	Pass

Evidence

20,21:

Add

22:

Add

<input type="text" value="First Name"/>	<input type="text" value="Joe"/>
<input type="text" value="Surname"/>	<input type="text" value="Bloggs"/>
<input type="radio"/> Male <input type="radio"/> Female <input checked="" type="radio"/> Other	<input checked="" type="radio"/> Male <input type="radio"/> Female <input type="radio"/> Other
Selected by default: <input checked="" type="radio"/> Yes <input type="radio"/> No	Selected by default: <input checked="" type="radio"/> Yes <input type="radio"/> No

member_id	user_id	first_name	last_name	gender	default_ticked
49	16	Joe	Bloggs	m	1



Jack Corbett

Family

Here you can see the family members that have been added and their individual profiles. You can edit your submitted information for each person or add a new family member by pressing the button below. This information will be used on the plan page to tell us who will be eating each day.

Joe Bloggs



23:

Edit

Male Female Other

Selected by default:

Yes No



Male Female Other

Selected by default:

Yes No



member_id	user_id	first_name	last_name	gender	default_ticked
49	16	Jane	Bloggs	f	1

25:

Remove

Do you want to remove Jane Bloggs?



26:

Family

Here you can see the family members that have been added and their individual profiles. You can edit your submitted information for each person or add a new family member by pressing the button below. This information will be used on the plan page to tell us who will be eating each day.





Jack Corbett

27:

Joe Bloggs

Joe Bloggs

+

28:

Add

First Name

Surname

 Male Female OtherSelected by default:
 Yes No

Please complete both first name and surname

Plan

Test No.	Description	Expected Result	Pass/Fail
Normal			
29	When a member is selected	Their name is highlighted green.	Pass
30	Default ticked dictates which members are selected to begin with (Jess is not default ticked)	Any member who is default ticked should be ticked each day.	Pass
31	Delete current plan cannot be selected when no plans exist for that user	Delete current plan button unable to be selected.	Pass
32	Generate meal plan selected	The user is taken to the calendar page displaying their newly generated meal plan.	Pass
33	Generate meal plan selected when a plan for the coming week has already been generated	The user is asked if they want to delete the current plan.	Pass
34	Delete current plan can be pressed when a plan exist	The button is dark grey to show it is active.	Pass
35	Select delete current plan button	The user is taken to a page to confirm deletion of that weeks meal plan. Once deleted they are	Pass



Jack Corbett

		returned to the plan page with a confirmation message.	
36	No recipes are repeated in the same week	All recipes in the calendar page are different.	Pass
37	No recipes are used from the user's last plan	All the recipes in the new weeks plan are different	Pass
38	30 minute cook time on Monday	The chosen recipe for Monday take less than 30 minutes to cook.	Pass
39	60 minute cook time on Monday	The chosen recipe for Monday take less than 60 minutes to cook.	Pass
Boundary			
40	Trial plan generation to look at recipes before adding family members	Plan generated showing recipes but not calculating quantities of ingredients	Pass

Evidence

29:

A screenshot of a user interface element. It features a green circular profile icon followed by the name "Joe" and a checked checkbox icon to its left.A screenshot of a user interface element. It features a green circular profile icon followed by the name "Joe" and an unchecked checkbox icon to its left.

30:

Select who will be eating:

A screenshot of a user interface element. It shows two rows of user profiles. The first row has a checked checkbox next to "Joe". The second row has a checked checkbox next to "Jane".
A screenshot of a user interface element. It shows two rows of user profiles. The first row has a checked checkbox next to "Joe". The second row has an unchecked checkbox next to "Jess".

31:

A screenshot of a meal planning interface. At the top is a dark grey button labeled "Generate Meal Plan". Below it is a light grey button labeled "Delete Current Plan".



Jack Corbett

32,36:

	Monday - 13/03	Chicken Noodles	
	Tuesday - 14/03	Chilli Con Carnie	
	Wednesday - 15/03	Spicy Turkey Burgers	
	Thursday - 16/03	Chicken Kiev	
	Friday - 17/03	Quick Healthy Pizza	
	Saturday - 18/03	Devil Chicken	
	Sunday - 19/03	Chicken Stir Fry	

33:

Delete Plan

You already have a plan for the coming week. Do you wish to delete it?



34:

Generate Meal Plan

Delete Current Plan

35:

Delete Plan

Wednesday Thursday Friday Saturday Sunday

Do you want to delete your latest plan?



Your plan for the coming week has been deleted. You can now generate a new one.

Generate Meal Plan

Delete Current Plan



Jack Corbett

37:

Monday - 06/03
Chicken Kiev

Tuesday - 07/03
Spicy Turkey Burgers

Wednesday - 08/03
Devil Chicken

Thursday - 09/03
Red Lasagne

Friday - 10/03
Chicken Noodles

Saturday - 11/03
Quick Healthy Pizza

Sunday - 12/03
Sweet Potato Cottage Pie

Monday - 13/03
Homemade Pizza

Tuesday - 14/03
Tomato Sauce with Penne

Wednesday - 15/03
Spanish Chicken

Thursday - 16/03
Crispy Chicken and Bacon Salad

Friday - 17/03
Beef Stir-Fry in Hoisin Sauce

Saturday - 18/03
Chicken Stir Fry

Sunday - 19/03
Chicken Fajita Salad

38:

Select the number of minutes you have available to cook:

30



Crispy Chicken and Bacon Salad

Ingredients:

3 tsp Olive Oil
3 Chicken Breast(s)
9 New Potato(es)
Leafy Salad
6 Bacon Rasher(s)
45g Cajun Spice
3 Slice(s) Granary Bread

Instructions:

1. Dry fry the chicken and bacon with the cajun spice.
2. In the meantime slice the bread into small chunks, drizzle with olive oil and bake in the oven at 180C for 10-15 mins.
3. Boil the new potatoes.
4. Place the salad, chicken, bacon, croutons and potatoes in a bowl and serve with a dressing of choice.

Cook Time: 20 minutes



Jack Corbett

39:

Select the number of minutes you have available to cook:

Spicy Turkey Burgers

Ingredients:

3 tbsp Worcestershire Sauce
6 Spring Onion(s)
450g Turkey Mince
3 tbsp Yoghurt
30g Cayenne Pepper
30g Paprika
3 tsp Dried Parsley
3 tsp Tabasco Sauce
3 Burger Bun(s)

Instructions:

1. Mix all the ingredients in a bowl.
2. Split into balls (depending on the amount of burgers you wish to make) and shape into a burger shape on a floured board.
3. Chill in the fridge for half an hour before frying in a pan for about 10 mins on a low heat.

Cook Time: 45 minutes

40:

Monday - 13/03
Pork with Apple

Tuesday - 14/03
Devil Chicken

Wednesday - 15/03
Sweet Potato Cottage Pie

Thursday - 16/03
Haddock with Mustard Sauce

Friday - 17/03
Spicy Turkey Burgers

Saturday - 18/03
Chicken Kiev

Sunday - 19/03
Tomato Sauce with Penne



Calendar

Test No.	Description	Expected Result	Pass/Fail
Normal			
41	When no plan has been generated or all have been deleted	Display message to user to inform them that they must generate a meal plan first, from the plan page.	Pass
42	Display platters depending on how many people are eating that day	Three platters on Monday.	Pass
43	Select a recipe	The user is taken to a page with calculated ingredients depending on the number of people eating, instructions and cook time.	Pass
44	Ingredients are calculated on the number of people eating	The ingredients for Monday's recipe are three times the quantity stored in the database.	Pass
45	If more than 5 people are eating on a day	The platter images are spread over multiple lines to avoid conflicting with the recipe name on mobile.	Pass

Evidence

41:

Calendar

Here you can view your meal plan for the week with the number of people eating each day on the side. Simply click the recipe for today and get cooking.

To view your meal plan you must first visit the plan page and generate one.

42:

Monday - 13/03
Spanish Chicken





Jack Corbett

43: Spanish Chicken

Ingredients:

0.75 Onion(s)
3 Chicken Breast(s)
450g Canned Chopped Tomatoes
Baby Corn
1.5 Yellow Pepper(s)
Mixed Herbs
Baby Carrots

Instructions:

Steam corn and carrots.

Oven:

1. Mix together a bowl of tomatoes, mixed herbs, onions and peppers.
2. Slice chicken and place it in an ovenproof bowl and cover with mix.
3. Cook at 200C for 45 to 60 minutes.

Or

Hob (shorter cooking time):

1. Brown chicken in frying pan.
2. Soften chopped onion and pepper.
3. Add mixed herbs and tomatoes.
4. Stir and cook for 10-15 mins.

Cook Time: 20 minutes

44: Ingredients:

0.75 Onion(s)
3 Chicken Breast(s)
450g Canned Chopped Tomatoes
Baby Corn
1.5 Yellow Pepper(s)
Mixed Herbs
Baby Carrots

ingredient_id	recipe_id	quantity
2	14	0.25
17	14	1.00
31	14	150.00
54	14	0.00
60	14	0.50
71	14	0.00
78	14	0.00

45: HOME



Calendar

Here you can view your meal plan for the week with the number of people eating each day on the side. Simply click the recipe for today and get cooking.

Monday - 13/03

Beef Stir-Fry in Hoisin Sauce





List

Test No.	Description	Expected Result	Pass/Fail
Normal			
46	When no plan has been generated or all have been deleted	Display message to user to inform them that they must generate a meal plan first, from the plan page before they can view their shopping list.	Pass
47	Ingredients are separated by category	Shopping list is split by ingredient category to make it easy to find in the supermarket.	Pass
48	Repeated ingredients are summed	Chicken is being eaten three times with one breast used for each recipe per person. Three people are eating each time so there should be 9 in the list.	Pass
49	Mobile view	The shopping list is split into two columns.	Pass
50	Other (uncategorised) column on mobile	The uncategorized ingredients are split into two columns on mobile to keep columns even.	Pass

Evidence

46:

List

Here you can see all the ingredients for the recipes for your current plan.

To view your shopping list you must first visit the plan page and generate a meal plan.



Jack Corbett

47:

List

Here you can see all the ingredients for the recipes for your current plan.

Fruit and Vegetables	Protein	Carbohydrates	Dairy	Other
3 cloves Garlic	750g Minced Beef	450g Noodles	90g Margarine	5 tsp Curry Powder
12 Spring Onion(s)	6 Pork Chop(s)	2 Potato(es)	Parmesan Cheese	Mixed Herbs
1170g Canned Chopped Tomatoes	9 Chicken Breast(s)	270g Brown Rice	3 tsp Honey	2 tbsp Worcestershire Sauce
Broccoli	300g Kidney Beans	9 New Potato(es)		6 tbsp Tomato Puree
195g Sweetcorn		270g Rice		4 tbsp Tomato Ketchup
2.25 Onion(s)		2 Sweet Potato(es)		3 tsp Mustard Powder
2.25 Red Pepper(s)		240g Brown Penne Pasta		1 tsp Chinese 5 Spices
75g Peas		2g Plain Flour		2 tbsp Sweet Chutney
3 tsp Mustard				3 tbsp Soy Sauce
375g Beansprout(s)				1 tbsp Oregano
3 Carrot(s)				2 tsp Caster Sugar
				3 cube(s) Beef Stock
				5 tbsp Coconut Oil
				5 tbsp Oyster Sauce
				6 tbsp Apple Sauce

48:



Monday - 13/03

Pork with Apple



Tuesday - 14/03

Tomato Sauce with Penne



Wednesday - 15/03

Sweet Potato Cottage Pie



Thursday - 16/03

Chicken Stir Fry



Friday - 17/03

Devil Chicken



Saturday - 18/03

Chilli Con Carnie



Sunday - 19/03

Chicken Noodles

49:

HOME



List

Here you can see all the ingredients for the recipes for your current plan.

Fruit and Vegetables	Protein	Carbohydrates	Dairy
3 cloves Garlic	750g Minced Beef	450g Noodles	90g Margarine
12 Spring Onion(s)	6 Pork Chop(s)	2 Potato(es)	Parmesan Cheese
1170g Canned Chopped Tomatoes	9 Chicken Breast(s)	270g Brown Rice	3 tsp Honey
Broccoli	300g Kidney Beans	9 New Potato(es)	
195g Sweetcorn		270g Rice	
2.25 Onion(s)		2 Sweet Potato(es)	
2.25 Red Pepper(s)		240g Brown Penne Pasta	
75g Peas			
3 tsp Mustard			
375g Beansprout(s)			
3 Carrot(s)			



50:

Other

5 tsp Curry Powder	3 tbsp Soy Sauce
Mixed Herbs	1 tbsp Oregano
2 tbsp Worcestershire Sauce	2 tsp Caster Sugar
6 tbsp Tomato Puree	3 cube(s) Beef Stock
4 tbsp Tomato Ketchup	5 tbsp Coconut Oil
3 tsp Mustard Powder	5 tbsp Oyster Sauce
1 tsp Chinese 5 Spices	6 tbsp Apple Sauce
2 tbsp Sweet Chutney	



MENU SOLUTIONS - JACK CORBETT

Admin

Test No.	Description	Test Data	Expected Result	Pass/Fail
Normal				
51	Admin page can only be accessed by accounts with permission	-	Joe Bloggs cannot access the admin page but Jack Corbett can.	Pass
52	Clicking add a new ingredient	-	Adds another group of inputs for another ingredient.	Pass
53	Recipe added with all new ingredients	Test, 30, 1. This is a test 2. This is a test, Test 1, g, 10, Fruit and Veg, Test 2, ml, 10, Other	Once the form is submitted the user is given a green message confirming the recipe has been added. Both ingredients have also been added to the ingredients table and the recipe is in the recipe table.	Pass
54	Recipe added using ingredients already in the database	Test, 30, 1. This is a test 2. This is a test Coconut Oil, tbsp, 1, Other Onion(s), - , 1, Fruit and Veg	Once the form is submitted the user is given a green message confirming the recipe has been added. The ingredients that were already in the ingredient table have been used and linked to the new recipe.	Pass
55	Recipe added where	Test, 30, 1. This is	Once the form is submitted	Pass



	ingredient has no measurement or quantity	a test 2. This is a test Test 1, -, -, Other, Test 2, g, 1, Fruit and Veg	the user is given a green message confirming the recipe has been added. The ingredient is still added to the recipe table as it is possible not to specify and leave the decision up to the user.	
Boundary				
56	Submit recipe with only one ingredient	Test, 30, 1. This a test 2. This is a test Test 1, g, 1, Fruit and Veg	The recipe is still added to the database. This is to allow future expansion into meal kits.	Pass
Erroneous				
57	Submit empty form	-	Red error message that all the information was not completed so the recipe could not be added.	Pass
58	Submit without recipe name	-, 30, 1. This is a test 2. This is a test, Test 1, g, 10, Fruit and Veg	"	Pass
59	Submit without cook time	Test, -, 1. This is a test 2. This is a test, Test 1, g, 10, Fruit and Veg	"	Pass
60	Submit without instructions	Test, 30, -, Test 1, g, 10, Fruit and Veg	"	Pass
61	Submit with no ingredients	Test, 30, 1. This is a test, -	"	Pass



Jack Corbett

Evidence

51:

HOME FAMILY▼ PLAN CALENDAR LIST ABOUT Joe Bloggs LOGOUT

HOME FAMILY▼ PLAN CALENDAR LIST ABOUT Joe Bloggs LOGOUT

You do not have permission to access this page.

HOME FAMILY▼ PLAN CALENDAR LIST ABOUT ADMIN Jack Corbett LOGOUT

HOME FAMILY▼ PLAN CALENDAR LIST ABOUT ADMIN Jack Corbett LOGOUT

Admin

Welcome to the admin page. You can only access this if you have been granted permission. From here you can help me improve the Menu Solutions experience by adding new recipes which will be used to generate users weekly meal plans. Simply follow the instructions below to input the details.

1. Give your recipe a name and tell me how long it takes to cook in minutes.
2. Write out numbered instructions making a new line for each new step (the same format as these instructions). Do not press submit until all the recipes have been entered.
3. Fill in the ingredient name (always use the plural with brackets eg: Potato(es) or Carrot(s)), how it is measured, the quantity needed (this is per person and if there is no set portion eg: serve with carrots enter 0) and pick a category. The other category is for ingredients that do not fit in to the other categories and the hide category is used for ingredients the user doesn't need on their shopping list such as water.
4. To add the rest of the ingredients just click the Add an Ingredient button and another set of boxes will appear. Repeat this for all ingredients and finish by pressing submit.

Recipe Name Cook Time
Instructions

Ingredient Name	Measurement	Quantity
-----------------	-------------	----------

52:

Add an Ingredient

Ingredient Name	Measurement	Quantity
-----------------	-------------	----------

Choose a category



Jack Corbett

Add an Ingredient

Ingredient Name	Measurement	Quantity
Choose a category		
Ingredient Name	Measurement	Quantity
Choose a category		

53:

Test	30
1. This is a test 2. This is a test	Submit

Add an Ingredient

Test 1	g	10
Fruit and Veg	Submit	
Test 2	ml	10
Other	Submit	

ingredient_id	recipe_id	quantity	ingredient_id	ingredient_name	measurement	category
92	18	10.00	92	Test 2	ml	6
91	18	10.00	91	Test 1	g	1
84	17	150.00	90	Burger Bun(s)		3
86	17	10.00	89	Tabasco Sauce	tsp	6
			88	Dried Parsley	tsp	1
16	Haddock with Mustard Sauce	1. Place the fish in an ovenproof dish. 2. Pour a ...	30			
17	Spicy Turkey Burgers	1. Mix all the ingredients in a bowl. 2. Split in...	45			
18	Test	1. This is a test 2. This is a test	30			

pressing submit.

Your recipe has been added to the database and can now be chosen in plan generation.

Recipe Name	Cook Time
-------------	-----------



Jack Corbett

54:

Test 30

1. This is a test
2. This is a test

Submit

Add an Ingredient

Coconut Oil	tbsp	1
Other		

Onion(s)	Measurement	1
Fruit and Veg		

ingredient_id	recipe_id	quantity	ingredient_id	ingredient_name	measurement	category
1	18	1.00	1	Coconut Oil	tbsp	6
2	18	1.00	2	Onion(s)		1

15 Crispy Chicken and Bacon Salad	1. Dry fry the chicken and bacon with the cajun sp...	20
16 Haddock with Mustard Sauce	1. Place the fish in an ovenproof dish. 2. Pour a ...	30
17 Spicy Turkey Burgers	1. Mix all the ingredients in a bowl. 2. Split in...	45
18 Test	1. This is a test 2. This is a test	30

pressing submit.

Your recipe has been added to the database and can now be chosen in plan generation.

55:

Test 30

1. This is a test
2. This is a test

Submit

Add an Ingredient

Test 1	Measurement	Quantity
Other		

Test 2	g	1
Choose a category		



Jack Corbett

ingredient_id	recipe_id	quantity
92	18	1.00
91	18	0.00
86	17	150.00

ingredient_id	ingredient_name	measurement	category
92	Test 2	g	0
91	Test 1		6
90	Burger Bun(s)		3
89	Tabasco Sauce	tsn	6

17 Spicy Turkey
Burgers

1. Mix all the ingredients in a bowl.
2. Split in...

45
30

pressing submit.

Your recipe has been added to the database and can now be chosen in plan generation.

Recipe Name

Cook Time

56:

Test 30

1. This is a test
2. This is a test

Submit

Add an Ingredient

Test 1 g 1

Fruit and Veg

ingredient_id	recipe_id	quantity
91	18	1.00
86	17	10.00

ingredient_id	ingredient_name	measurement	category
91	Test 1	g	1
90	Burger Bun(s)		3

Burgers

bowl.
2. Split in...

30

pressing submit.

Your recipe has been added to the database and can now be chosen in plan generation.

Recipe Name

Cook Time

57,58,59,60,61:

Your recipe could not be added to the database as some of the information had not been completed.



Technical Solution

Overview

My code is split into the folders used on my webserver and commented throughout to explain its operation. Above each document the name of the file is shown, along with a list of it's complex components. I have included the most complex algorithms/pieces of code within the contents below to make them easy to locate.

Code Contents

Includes (PHP Scripts)	81
<i>addrecipe.inc.php</i>	81
<i>list.inc.php</i>	86
<i>plangeneration.inc.php</i>	88
<i>register.inc.php</i>	94
User	96
<i>admin.php</i>	99
Family	103
<i>members.php</i>	105
Plan	107
<i>plan.php</i>	107
Calendar	112
<i>calendar.php</i>	112
<i>dayrecipe.php</i>	114
List	117
<i>list.php</i>	117
Assets	120
SQL	125



index.php

- Javascript tutorial modal with tab system for each page
- Responsive display to user login

```
<?php
    include '_assets/header.php';
?>

<style>
/* Sets the font for the page */
body,h1,h2,h3,h4,h5,h6 {font-family: "Lato", sans-serif; }

body, html, h3 {
    height: 100%;
    color: #FFFFFF;
    line-height: 1.8;
}

h1,h2,h4,h5,h6 {
    height: 100%;
    color: #777;
    line-height: 1.8;
}

/* Create a Parallax Effect */
.bgimg-1 {
    opacity: 0.9;
    background-attachment: fixed;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
}

/* First image (Logo. Full height) */
.bgimg-1 {
    background-image: url(_img/choppingboard.jpg);
    min-height: 100%;
}

/* Turn off parallax scrolling for tablets and mobiles */
@media only screen and (max-width: 768px) {
    .bgimg-1, .bgimg-2, .bgimg-3 {
        background-attachment: scroll;
    }
}
.display {display:none}
</style>

<!-- Navigation Bar -->
<?php
```



```
if (isset($_SESSION['user_id'])) {
    include '_assets/navbar.loggedin.php';
} else {
    include '_assets/navbar.loggedout.php';
}
?>

<!-- First Parallax Image with logo and title -->
<div class="bgimg-1 w3-opacity w3-display-container" style="white-space:nowrap;">
    
    <span class="w3-display-middle w3-center w3-padding-large w3-black w3-xlarge w3-wide w3-animate-opacity">MENU SOLUTIONS</span>
    <a <?php
        if (isset($_SESSION['user_id'])) {
            echo 'href="family/members.php"';
        } else {
            echo 'href="user/login.php"';
        }
    ?> class="w3-display-middle-below w3-center w3-btn w3-animate-opacity w3-hover-light-grey w3-theme-d1">START EATING SMARTER</a>
    <div class="w3-display-bottommiddle-above">
        <a onclick="document.getElementById('tutorial').style.display='block'" class="w3-centre w3-btn w3-animate-opacity w3-dark-grey w3-round-large w3-small">TUTORIAL</a>
    </div>
</div>

<!-- Instructions Title -->
<div class="w3-container w3-center">
    <h2>Simple Steps to a Healthy Diet</h2>
</div>

<!-- Instructions -->
<div class="w3-row-padding w3-container w3-center w3-dark-grey">
    <div class="w3-quarter"><h3>Add Your Family</h3>
        
        <p>Add each family members details</p>
        <?php
            if (isset($_SESSION['user_id'])) {
                echo '<a href="family/members.php" class=" w3-btn w3-hover-white w3-theme-d1">FAMILY</a>';
            }
        ?>
        <br /><br />
    </div>
    <div class="w3-quarter w3-dark-grey w3-center"><h3>Tell Us When</h3>
        
        <p>Tell us how much time you have to cook each night and who will be eating</p>
        <?php
            if (isset($_SESSION['user_id'])) {
```



```
        echo '<a href="plan/plan.php" class=" w3-btn w3-hover-white w3-theme-d1">PLAN</a>';
    }
?>
<br /><br />
</div>

<div class="w3-quarter w3-dark-grey w3-center"><h3>Generate Calendar</h3>

<p>Review your calendar of meal plans for the week</p>
<?php
if (isset($_SESSION['user_id'])) {
    echo '<a href="calendar/calendar.php" class=" w3-btn w3-hover-white w3-theme-d1">CALENDAR</a>';
}
?>
<br /><br />
</div>

<div class="w3-quarter w3-dark-grey w3-center"><h3>Shopping List</h3>

<p>Use your generated shopping list to fetch the ingredients and enjoy a week of stress free cooking</p>
<?php
if (isset($_SESSION['user_id'])) {
    echo '<a href="list/list.php" class=" w3-btn w3-hover-white w3-theme-d1">LIST</a>';
}
?>
<br /><br />
</div>
</div>

<div class="site-footer">
<?php
    include '_assets/footer.php'
?>
</div>

<!-- Tutorial modal -->
<div id="tutorial" class="w3-modal">
<div class="w3-modal-content w3-card-4 w3-animate-zoom">
<header class="w3-container w3-dark-grey">
<span onclick="document.getElementById('tutorial').style.display='none'" class="w3-closebtn w3-padding-top w3-hover-text-red">&times;</span>
<h3>Tutorial</h3>
</header>

<ul class="w3-pagination w3-white w3-border-bottom" style="width:100%;">
<li><a href="#" class="tablink" onclick="openTab(event, 'Intro')">Intro</a></li>
<li><a href="#" class="tablink" onclick="openTab(event, 'Add Family')">Add
```



```
Family</a></li>
    <li><a href="#" class="tablink" onclick="openTab(event, 'Members')">View
Members</a></li>
    <li><a href="#" class="tablink" onclick="openTab(event, 'Plan')">Create
Plan</a></li>
    <li><a href="#" class="tablink" onclick="openTab(event, 'Calendar')">View
Calendar</a></li>
    <li><a href="#" class="tablink" onclick="openTab(event, 'List')">Shopping
List</a></li>
</ul>

<div id="Intro" class="w3-container w3-text-grey display">
    <div class="w3-center">
        <h4>Intro</h4>
        
        <p>Welcome to Menu Solutions. In this tutorial I will talk you through how
to set up your family,
            generate meal plans and check your shopping list. Navigate by clicking
the tabs relating to each
            page of the site to learn more.</p>
    </div>
</div>

<div id="Add Family" class="w3-container w3-text-grey display">
    <p>First you need to add your family members and fill in
        some basic information for each person. We will use this when generating
the meal plan.
        To add a member simply select add members from the drop down in the top bar
or press the round green plus button.
        This will take you to a page where you can fill in their first name,
surname and gender. Once that is filled in press the green add person
button.</p>
    <div class="w3-center">
        
    </div>
    <br />
</div>

<div id="Members" class="w3-container w3-text-grey display">
    <p>You can manage your family from the members page. To delete a family
member just press the x on the far right and the tick to confirm.
        To edit their information just press the pencil and when your finished
press the green tick to save your changes. </p>
    <div class="w3-center">
        
    </div>
    <br />
</div>

<div id="Plan" class="w3-container w3-text-grey display">
    <p>The plan page is where you input who will be eating each day
```



and tell us how long you will have to cook. Click on the tab for each day and use the checkboxes to select the available family members and the drop down for how many minutes you have.

Once you have pressed the generate button the calendar will be populated with your meal plan.

To overwrite a plan plan do simply press the delete current plan button or if you press generate meal plan again you will be asked if you would like to delete your current plan.

Always delete your current plan before entering the new values as they will not be saved.</p>

```
<div class="w3-center">
    
</div>
<br />
</div>
```

<div id="Calendar" class="w3-container w3-text-grey display">
 <p>The calendar page is where you can see your meal plan for the current week. It is also how

you will access your recipes each day when cooking. Each item in the list shows the day and date

to be cooked on the left along with the name of the recipe. On the right you can see how many people

are eating each day, represented by the number of heads. To see the instructions for a recipe just click

the recipe name and it will take you to a page with everything you need to get cooking. It even works out the

quantities based on how many are eating for you. Bon appetit!</p>

```
<div class="w3-center">
    
</div>
<br />
</div>
```

<div id="List" class="w3-container w3-text-grey display">
 <p>The list page contains the shopping list for your latest meal plan. You can see the ingredients split into categories

to make them easy to find in the supermarket. It also shows you exactly how much of each you need to feed the amount of people

who are eating that day! It even combines ingredients used on different days.</p>

```
<div class="w3-center">
    
</div>
<br />
</div>
</div>
```

<!-- End of modal tutorial -->

```
<script>
```



```
document.getElementsByClassName("tablink")[0].click();  
  
function openTab(evt, pageName) {  
  var i, x, tablinks;  
  x = document.getElementsByClassName("display");  
  for (i = 0; i < x.length; i++) {  
    x[i].style.display = "none";  
  }  
  tablinks = document.getElementsByClassName("tablink");  
  for (i = 0; i < x.length; i++) {  
    tablinks[i].classList.remove("w3-light-grey");  
  }  
  document.getElementById(pageName).style.display = "block";  
  evt.currentTarget.classList.add("w3-light-grey");  
}  
</script>
```

about.php

```
<?php  
  include '_assets/header.php';  
?  
  
<style>  
  body,h1,h2,h3,h4,h5,h6 {font-family: "Lato", sans-serif;}  
  
  h1,h2,h4,h5,h6 {  
    height: 100%;  
    color: #777;  
    line-height: 1.8;  
  }  
  body, html, h3 {  
    height: 100%;  
    color: #FFFFFF;  
    line-height: 1.8;  
  }  
  
  /* Create a Parallax Effect */  
  .bgimg-1, .bgimg-2, .bgimg-3 {  
    opacity: 0.8;  
    background-attachment: fixed;  
    background-position: center;  
    background-repeat: no-repeat;  
    background-size: cover;  
  }  
  
  /* First image */  
  .bgimg-1 {  
    background-image: url(_img/raspberries.jpeg);  
    min-height: 400px;  
  }
```



```
/* Second image */
.bgimg-2 {
    background-image: url(_img/blueberries.jpeg);
    min-height: 300px;
}

/* Third image */
.bgimg-3 {
    background-image: url(_img/strawberries.jpeg);
    min-height: 300px;
}

/* Turn off parallax scrolling for tablets and mobiles */
@media only screen and (max-width: 768px) {
    .bgimg-1, .bgimg-2, .bgimg-3 {
        background-attachment: scroll;
    }
}
</style>

<!-- Navigation Bar -->
<?php
if (isset($_SESSION['user_id'])) {
    include '_assets/navbar.loggedin.php';
} else {
    include '_assets/navbar.loggedout.php';
?
>

<!-- First Parallax Image -->
<div class="bgimg-1 w3-display-container" style="white-space:nowrap;">
    <span class="w3-display-middle w3-center w3-padding-xlarge w3-black w3-xlarge w3-wide w3-animate-opacity">MENU SOLUTIONS</span>
</div>

<!-- Container Introduction -->
<div class="w3-container w3-row-padding w3-dark-grey ">
    <h3 class="w3-center">How it Works</h3>
    <p class="w3-center">Menu Solutions provides an advanced meal planner which is accessible to all. It uses advanced algorithms based on the dietary requirements of each person to pick recipes that are suitable. It also caters for the modern family by allowing users to input who will be eating each day and how much time is available to cook to ensure everyone always gets a tasty, healthy meal. Another focus of Menu Solutions is to decrease the amount of food wastage and so it will build you a shopping list with everything you need to cook healthy meals every day of the week meaning you only buy what you need. This is better for the
```



```
planet and for your pocket.  
So what are you waiting for? Lets get started...</p>  
<br />  
</div>  
  
<!-- Second Parallax Image -->  
<div class="bgimg-2 w3-display-container">  
    <span class="w3-display-middle w3-xlarge w3-wide w3-black  
w3-padding-large">GETTING STARTED</span>  
</div>  
  
<!-- Container Instructions -->  
<div class="w3-container w3-row-padding w3-dark-grey">  
    <h3 class="w3-center">Creating an Account</h3>  
    <p class="w3-center">In order to start using this Menu Solutions please create an  
account by clicking on  
the login button at the top right of the screen. Once your account has been set  
up, by clicking register from the login page,  
you will notice five more tabs in your navigation bar: Family, Plan, Calendar,  
List and Account. If you  
do not see logout in the top right corner this is because you are not yet  
logged in and should  
press login and enter your username and password that you used when you  
registered.  
</p>  
    <h3 class="w3-center">Generating your first Meal Plan</h3>  
    <p class="w3-center"> Start with the family tab where you can add your family  
members and fill in  
some basic information for each person about their dietary requirements. We  
will use this when generating  
the meal plan to ensure everyone in the family gets what they need. The next  
step is to access  
the plan tab which enables you to input who will be eating each day and tell us  
how long you  
will have to cook. This means our meal plans can be very flexible and will fit  
in around your  
normal routine. Once you have pressed the generate button on this page your  
calendar page will  
be populated. Don't worry if plans change you can easily overwrite your current  
plan with a new one!  
</p>  
    <h3 class="w3-center">Cooking your Food</h3>  
    <p class="w3-center">The calendar tab is where you can see your meal plan for the  
current week. It is also how  
you will access your recipes each day when cooking. To see the instructions for  
that day just click the recipe name  
and it will take you to a page with everything you need to get cooking. It even  
works out the quantities based on  
how many are eating for you. Finally there is the list tab which generates a  
shopping list for your current meal plan,  
categorised to make picking up the required ingredients a breeze.  
</p>
```



```
<br />
</div>

<!-- Third Parallax Image -->
<div class="bgimg-3 w3-display-container">
  <span class="w3-display-middle w3-xlarge w3-wide w3-black w3-padding-large">ABOUT
ME</span>
</div>

<!-- Container About Me -->
<div class="w3-container w3-row-padding w3-dark-grey">
  <div class="w3-twothird">
    <h3>Jack Corbett</h3>
    <p>I am currently studying my A Levels at Barton Peveril College and this is my second year Computer Science Project.
    When given the task to code pretty much anything I wanted to I really wanted to solve a problem and create a product
      that I was proud of and believed actually warranted the time and effort put into it. I know that trying to decide what
        to eat each day is a struggle for many families and that is why often bad habbits are started. It is so important to
          eat healthy food as it affects everything you do and so I wanted to make it easy for everyone by providing a simple
            solution to a complex problem.</p>
    <p>Another issue I wanted to solve was to decrease food wastage which along with diet is the other biggest issue that we
      face. That is why I have built in an inventory system so that you only buy what you need and reuse ingredients rather than
        just getting rid of the excess. This site is still in very early development and many of the final features are not yet
          present but I would really appreciate anyone's support and suggestions on how I could improve it as I am working alone on
            this project. I hope you enjoy what I have created.
    </div>
    <div class="w3-third">
      <br />
      
      <p class="w3-text-theme">Presenting for Google at the BETT Show 2016</p>
    </div>
  </div>

<!-- Credits to sources -->
<div class="w3-container w3-row-padding w3-dark-grey w3-topbar">
  <h3 class=" w3-center">Credits</h3>
  <ul>
    <li type="square">W3 Schools for allowing open use of their CSS template which is excellent for beginners to web design and
      their wide range of tutorials</li>
    <li type="square">PHP and MySQL for Dynamic Websites by Larry Ulman is a brilliant book that takes you step by step
      through building your website and keeping it secure</li>
    <li type="square">Unsplash for being an amazing source for high quality

```



```
images</li>
<li type="square">Stack Overflow for helping me fix all of the issues I ran
into along the way</li>
<li type="square">Logo Makr for being a great tool for building the logos used
on the site</li>
</ul>
<br />
</div>

<?php
    include '_assets/footer.php'
?>
```

Includes (PHP Scripts)

dbh.inc.php

```
<?php
$username = "menuSolu_admin";
$password = "-----";

$conn = mysqli_connect("localhost", $username, $password,
"menuSolu_MenuSolutions");

if (!$conn) {
    die("Connection failed: ".mysqli_connect_error());
}
?>
```

Admin

addrecipe.inc.php

- Like comparison and retrieval of insert id
- Empty field check

```
<?php
include '../dbh.inc.php';

//Store the values passed through
$name = $_POST['name'];
$cook_time = $_POST['cookeTime'];
$instructions = $_POST['instructions'];

//Store the arrays passed through
$ingredient_name = $_POST['ingredient_name'];
$measurement = $_POST['ingredient_measurement'];
$quantity = $_POST['ingredient_quantity'];
$category = $_POST['ingredient_category'];
```



```
//Check all the information has been completed
if (empty($name) or empty($cook_time) or empty($instructions) or
empty($ingredient_name[0])) {
    header ('Location: ../../user/admin.php?failed');
    exit();
}

//Add the name, the instructions and cook time to the recipe table
$sql = "INSERT INTO recipes (name, instructions, cook_time) VALUES('$name',
'$instructions', '$cook_time')";
$result = @mysqli_query($conn, $sql);
$recipe_id = mysqli_insert_id($conn);

//For each ingredient check if it already exists in the ingredient table
for ($i=0; $i<count($ingredient_name); $i++) {
    $sql = "SELECT ingredient_id FROM ingredients WHERE ingredient_name LIKE
'$ingredient_name[$i]'";
    $result = @mysqli_query($conn, $sql);
//If it exists then get the ingredient ID and add it to the ID array
    if ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        $ingredient_id[$i] = $row['ingredient_id'];
    } else {
        //Add the ingredient to the ingredient table as it doesn't already exist
        $sql = "INSERT INTO ingredients (ingredient_name, measurement, category) VALUES
('$ingredient_name[$i]', '$measurement[$i]', '$category[$i]')";
        $result = @mysqli_query($conn, $sql);
        //Add the newly generated ingredient ID to the array
        $ingredient_id[$i] = mysqli_insert_id($conn);
    }
    //Add the information to the ingredients recipes link table
    $sql = "INSERT INTO ingredients_recipes VALUES ('$ingredient_id[$i]',
'$recipe_id', '$quantity[$i]')";
    $result = @mysqli_query($conn, $sql);
}

//Return to the admin page with a confirmation that the recipe has been added to
the database
header ('Location: ../../user/admin.php?added');
?>
```

Family

add-member.inc.php

- Escape user submitted strings
- Empty field check

```
<?php
session_start();
```



```
include '../dbh.inc.php';

//Pass through the members details and get the users ID from their session
$firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
$lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
$gender = mysqli_real_escape_string($conn, $_POST['gender']);
$default_ticked = mysqli_real_escape_string($conn, $_POST['default_ticked']);
$id = $_SESSION['user_id'];

//Double check that the name has not been left empty
if (empty($firstname)) {
    header("Location: ../../family/addmember.php?error=empty");
    exit();
} elseif (empty($lastname)) {
    header("Location: ../../family/addmember.php?error=empty");
    exit();
} else {
    //Insert the family members information into the database
    $sql = "INSERT INTO family (user_id, first_name, last_name, gender,
    default_ticked)
            VALUES ('$id', '$firstname', '$lastname', '$gender', '$default_ticked')";
    $result = mysqli_query($conn, $sql);

    header('Location: ../../family/members.php');
}
?>
```

edit-fetch.inc.php

```
<?php
include '../_includes/dbh.inc.php';

//Get the member ID from the URL
$member_id = mysqli_real_escape_string($conn, $_GET['member']);

//Select all the members information
$sql = "SELECT * FROM family WHERE member_id = '$member_id'";
$result = @mysqli_query($conn, $sql);

$row = mysqli_fetch_array($result, MYSQLI_ASSOC);

//Store the information in arrays to be used to edit
$first_name = $row['first_name'];
$last_name = $row['last_name'];
$gender = $row['gender'];
$default_ticked = $row['default_ticked'];
?>
```



edit-update.inc.php

- Escape user submitted strings

```
<?php
include '../dbh.inc.php';

//Pass through all the new values from the edit page
$member_id = $_POST['memberid'];
$firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
$lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
$gender = mysqli_real_escape_string($conn, $_POST['gender']);
$default_ticked = mysqli_real_escape_string($conn, $_POST['default_ticked']);

//Double check that the names are not empty
if (empty($firstname)) {
    header("Location: ../../family/editmember.php?error=empty");
    exit();
} elseif (empty($lastname)) {
    header("Location: ../../family/editmember.php?error=empty");
    exit();
} else {
    //Update the details for that member in the family table
    $sql = "UPDATE family SET first_name = '$firstname', last_name = '$lastname',
    gender = '$gender', default_ticked = '$default_ticked' WHERE member_id =
    '$member_id'";
    $result = mysqli_query($conn, $sql);

    header('Location: ../../family/members.php');
}
?>
```

members-fetch.inc.php

- Concatenate the user's name

```
<?php
$id = $_SESSION['user_id'];

//Select the members name, id and gender from the family table
$sql = "SELECT CONCAT(first_name, ' ', last_name) AS name, member_id, gender FROM
family WHERE user_id = '$id'";

$result = @mysqli_query($conn, $sql);
?>
```



remove-delete.inc.php

- Escape user submitted string

```
<?php
include '../dbh.inc.php';

//Get the member's ID from the URL
$member_id = mysqli_real_escape_string($conn, $_GET['member']);

//Delete the member from the family page
$sql = "DELETE FROM family WHERE member_id = '$member_id'";
$result = mysqli_query($conn, $sql);

header('Location: ../../family/members.php');
?>
```

remove-fetch.inc.php

- Escape user submitted strings
- Concatenate the user's name

```
<?php
include '../_includes/dbh.inc.php';

//Get the member ID from the URL
$member_id = mysqli_real_escape_string($conn, $_GET['member']);

//Select the member's name from the family table
$sql = "SELECT CONCAT(first_name, ' ', last_name) AS name FROM family WHERE
member_id = '$member_id'";
$result = @mysqli_query($conn, $sql);

$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
?>
```



List

list.inc.php

- SQL query using joins, aggregate functions and grouping to select ingredients
- Reusable function providing conditional formatting

```
<?php
//Ingredient Categories:
//Fruit and Veg = 1
$fv_name = array();
$fv_amount = array();
$fv_measure = array();
//Protein = 2
$pr_name = array();
$pr_amount = array();
$pr_measure = array();
//Carbohydrates = 3
$ca_name = array();
$ca_amount = array();
$ca_measure = array();
//Dairy = 4
$da_name = array();
$da_amount = array();
$da_measure = array();
//Uncategorised(Other) = 6
$un_name = array();
$un_amount = array();
$un_measure = array();
//Ignore = 5

//Find each ingredient involved in each recipe of the users latest plan. Add the
//totals if the ingredient is used in more than one recipe
$sql = "SELECT ingredients.category, ingredients.ingredient_name,
SUM(plan_recipes.no_eating * ingredients_recipes.quantity) As amount,
ingredients.measurement
FROM plan_recipes
JOIN recipes ON recipes.recipe_id = plan_recipes.recipe_id
JOIN ingredients_recipes ON ingredients_recipes.recipe_id = recipes.recipe_id
JOIN ingredients ON ingredients.ingredient_id = ingredients_recipes.ingredient_id
WHERE plan_recipes.plan_id = '$plan_id'
GROUP BY ingredients.ingredient_name
ORDER BY ingredients.category";
$result = @mysqli_query($conn, $sql);

//Split the ingredients into categories and add them to each corresponding array
while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
    if ($row['category'] == 1) {
        array_push($fv_name, $row['ingredient_name']);
        array_push($fv_amount, round($row['amount'], 2));
    }
}
```



```
array_push($fv_measure, $row['measurement']);
} elseif ($row['category'] == 2) {
array_push($pr_name, $row['ingredient_name']);
array_push($pr_amount, round($row['amount']));
array_push($pr_measure, $row['measurement']);
} elseif ($row['category'] == 3) {
array_push($ca_name, $row['ingredient_name']);
array_push($ca_amount, round($row['amount']));
array_push($ca_measure, $row['measurement']);
} elseif ($row['category'] == 4) {
array_push($da_name, $row['ingredient_name']);
array_push($da_amount, round($row['amount']));
array_push($da_measure, $row['measurement']);
} elseif ($row['category'] == 6) {
array_push($un_name, $row['ingredient_name']);
array_push($un_amount, round($row['amount']));
array_push($un_measure, $row['measurement']);
}

//Function to output each array
function OutputArray($amount, $measure, $name, $half) {
//Loop through each ingredient
for ($i=0; $i<count($name); $i++) {
//Half is only used on mobile devices to split the other column into two to
make it easier to read
if ($half > 0 AND $i == $half) {
echo "</td> <td>";
}
//If the amount is 0 only the name of the ingredient needs to be outputted
as it does not specify an amount in the recipe
if ($amount[$i] > 0) {
//If the unit is grams or millilitres there is no need to put a space
after the amount
if ($measure[$i] == "g" OR $measure[$i] == "ml") {
echo $amount[$i] . $measure[$i] . " " . $name[$i] . "<br />";
} else {
echo $amount[$i] . " " . $measure[$i] . " " . $name[$i] . "<br />";
}
} else {
echo $name[$i] . "<br />";
}
}
}
?>
```



Plan

get-max-plan.inc.php

- Use of an aggregate function

```
<?php
include '../_includes/dbh.inc.php';

//Get the users ID from their session
$id = $_SESSION['user_id'];

//Find the latest plan for that user
$sql = "SELECT MAX(plan_id) plan_id FROM users_plan WHERE user_id = '$id'";
$result = @mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
$plan_id = $row['plan_id'];
?>
```

plan-generation.inc.php

- Use of associative array and variable variables to iterate through each day of the week
- Concatenation of strings to fetch the time for each day
- Use of objects to select plan dates
- Use of aggregate functions to select the number of recipes and the user's old plan
- Fetching insert id to establish database relationships

```
<?php
session_start();

include '../dbh.inc.php';

//Get the users ID from their session
$id = $_SESSION['user_id'];
//Create an associative array for each day of the week to enable loop iteration
$day = array(0 => 'monday', 1 => 'tuesday', 2 => 'wednesday', 3 => 'thursday', 4 => 'friday', 5 => 'saturday', 6 => 'sunday');
$time = array();
$no_eating = array();

for ($x=0;$x<7;$x++) {
    //Store the members eating each day (this could be used in future to provide personalised recommendations rather than just using the number of people eating)
    ${$day[$x]} = $_POST[$day[$x]];
    // Place the time values submitted in an array
    $time[$x] = $_POST[$day[$x] . '_time'];
    //Store how many members are eating each day
    $no_eating[$x] = count(${$day[$x]});
}
```



```
//Declare the array to store the plan dates
$date_eating = array();
//OOP to calculate the dates of next week
//Create DateTime object containing the current time1
$date = new DateTime();
//Set the date object to Monday of next week
$date->setISODate($date->format('o'), $date->format('W') + 1);
//Calculate every days date of that week from Monday to Sunday by using the 6
following days
$periods = new DatePeriod($date, new DateInterval('P1D'), 6);
//Convert the DatePeriod object to the week_dates array
$week_dates = iterator_to_array($periods);
//Format the dates inside the week_dates array ready to be stored in the database
as strings
for($i = 0; $i < 7; $i++) {
    $date_eating[$i] = $week_dates[$i]->format('Y-m-d');
}

//A script to find the largest plan ID for that user so we can check last week's
recipes
include 'get-max-plan.inc.php';

//If they have a plan in the database
if ($plan_id > 0) {

    $sql = "SELECT start_date FROM plan WHERE plan_id = '$plan_id'";
    $result = @mysqli_query($conn, $sql);
    $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
    $current_start_date = $row['start_date'];

    //Check if the start date of their latest plan is for the current week. If so ask
the user if they want to delete their current plan
    if ($current_start_date == $date_eating[0]) {
        header ("Location: ../../plan/removeplan.php");
        exit();
    }

    //Stores the recipes eaten the week before
    $recipe_history = array();

    //Fetch all the recipes linked to their latest plan
    $sql = "SELECT recipe_id FROM plan_recipes WHERE plan_id = '$plan_id'";
    $result = @mysqli_query($conn, $sql);

    //Add all the recipes from last week's plan to the recipe_history array
    while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        array_push($recipe_history, $row['recipe_id']);
    }
}

//Checks how many recipes there are in the table to set the searching upper bound
```



```
$sql = "SELECT COUNT(1) FROM recipes";
$result = @mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result);
$no_recipes = $row[0];

//Declare the plan array which will store the recipes chosen for each day
$plan = array();

//PLAN GENERATION LOOP
for($i=0; $i<7; $i++) {

    //Initially set the accept boolean switch to false to ensure the while loop runs
    //at least once
    $accept = false;

    //Loop to pick a new recipe each time provided accept is still false
    while ($accept == false) {
        //Set the accept switch to true so we can then check for errors
        $accept = true;

        //Select a random recipe
        $recipe = rand(1,$no_recipes);

        //If the recipe matches a recipe used in the previous weeks plan set accept to
        //false
        for ($a=0; $a<7; $a++) {
            if ($recipe == $recipe_history[$a]) {
                $accept = false;
            }
        }

        //Get the time to cook of the recipe from the database to check if the user has
        //enough time available
        $sql = "SELECT cook_time FROM recipes WHERE recipe_id = '$recipe'";
        $result = @mysqli_query($conn, $sql);
        $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
        $cook_time = $row['cook_time'];

        if ($time[$i] < $cook_time) {
            $accept = false;
        }

        //Check the recipe has not already been selected for that week
        for ($q=0; $q<$i+1; $q++) {
            if ($recipe == $plan[$q]) {
                $accept = false;
            }
        }

        //Insert the recipe number into the plan array
        $plan[$i] = $recipe;
    }
}
```



```
//Unset the recipe variable ready for the next loop
unset($recipe);
}
}

//ADD THE PLAN TO DATABASE
//First add the start and end dates to the plan table as this will give us our AI
//value for the plan ID
$sql = "INSERT INTO plan (start_date, end_date) VALUES('$date_eating[0]',
'$date_eating[6]')";
$result = @mysqli_query($conn, $sql);
//Get the plan ID so we can use this in our linking tables
$new_plan = mysqli_insert_id($conn);

//Link the plan to the users account
$sql = "INSERT INTO users_plan (user_id, plan_id) VALUES ('$id', '$new_plan')";
$result = @mysqli_query($conn, $sql);

//Insert each day into the plan recipes table
for ($x=0;$x<7;$x++) {
    $sql = "INSERT INTO plan_recipes (plan_id, recipe_id, no_eating, date_eating)
VALUES ('$new_plan', '$plan[$x]', '$no_eating[$x]', '$date_eating[$x]')";
    $result = @mysqli_query($conn, $sql);
}

//Jump to the calendar page to see the newly generated plan
header ("Location: ../../calendar/calendar.php");
?>
```

remove-plan.inc.php

- Use of aggregate functions

```
<?php
session_start();

include '../dbh.inc.php';

//Get the users ID from their session
$id = $_SESSION['user_id'];

//Get the users latest plan to delete
$sql = "SELECT MAX(plan_id) plan_id FROM users_plan WHERE user_id = '$id'";
$result = @mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
$plan_id = $row['plan_id'];

//Delete that plan from the plan table which will cascade to delete it from the
//others
$sql = "DELETE FROM plan WHERE plan_id = '$plan_id'";
$result = @mysqli_query($conn, $sql);
```



```
//Link back to the plan page and display a message confirming deletion
header ("Location: ../../plan/plan.php?removed");
?>
```

set-up-form.inc.php

- Creation of an associative array to mirror plan-generation.inc.php

```
<?php
include '../_includes/dbh.inc.php';

//Get the users ID from their session
$id = $_SESSION['user_id'];
//Create an associative array for each day of the week to enable loop iteration
$day = array(0 => 'monday', 1 => 'tuesday', 2 => 'wednesday', 3 => 'thursday', 4
=> 'friday', 5 => 'saturday', 6 => 'sunday');
$name = array();
$member_id = array();
$default_ticked = array();
$gender = array();

//Fetch all the details for each member in the user's family
$sql = "SELECT first_name, member_id, default_ticked, gender FROM family WHERE
user_id = '$id'";
$result = @mysqli_query($conn, $sql);

while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
    $name[] = $row['first_name'];
    $member_id[] = $row['member_id'];
    $default_ticked[] = $row['default_ticked'];
    $gender[] = $row['gender'];
}
?>
```

User

login.inc.php

- Escape user submitted strings
- Hash and salt user's password using bcrypt

```
<?php
session_start();

include '../dbh.inc.php';

//Escape strings to avoid sql injection
$email = mysqli_real_escape_string($conn, $_POST['email']);
$password = mysqli_real_escape_string($conn, $_POST['password']);
```



```
//Query the database to fetch the user's hashed password
$sql = "SELECT hash FROM users WHERE email = '$email'";
$result = @mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
$hash = $row['hash'];

//If nothing is returned then the email must not be linked to a user account
if ($hash === NULL) {
    //Direct the user back to the login page and display an incorrect email error
    header("Location: ../../user/login.php?error=email");
} else {
    //If a hash is returned run the password verify function to check if the
    submitted password generates the same hash
    if (password_verify($password, $hash)) {
        //If the password matches log the user in by fetching the user's ID and
        checking if they have admin permissions
        $sql = "SELECT user_id, admin FROM users WHERE email = '$email' AND hash =
'$hash'";
        $result = @mysqli_query($conn, $sql);
        $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
        //Store the user ID and their admin status in their session so it can be
        accessed across the site
        $_SESSION['user_id'] = $row['user_id'];
        $_SESSION['admin'] = $row['admin'];
        //Jump back to the login page
        header("Location: ../../");
    } else {
        //If the password submitted did not create a matching hash show a password
        error on the login page
        header("Location: ../../user/login.php?error=password");
    }
}
?>
```

logout.inc.php

```
<?php
session_start();
session_destroy();
header("Location: ../../");
?>
```



register.inc.php

- Escape user submitted strings
- Use RegEx to validate submitted data
- Use of a PHP to validate the user's email
- Use of PHP mail to send an account confirmation email

```
<?php
include '../dbh.inc.php';

//Escape the submitted values to protect against SQL injection
$firstname = mysqli_real_escape_string($conn, $_POST['firstname']);
$lastname = mysqli_real_escape_string($conn, $_POST['lastname']);
$email = mysqli_real_escape_string($conn, $_POST['email']);

//FORM VALIDATION
//Each check returns a specific error message
//Double check that none of the required fields are empty despite being html
required fields
if (empty($firstname)) {
    header("Location: ../../user/register.php?error=empty");
    exit();
} elseif (empty($lastname)) {
    header("Location: ../../user/register.php?error=empty");
    exit();
} elseif (empty($email)) {
    header("Location: ../../user/register.php?error=empty");
    exit();
} elseif (empty($password)) {
    header("Location: ../../user/register.php?error=empty");
    exit();
} else {
    //Check using RegEx that the strings submitted for the users name only contain
    alphabetic characters
    if (!preg_match("/^[a-zA-Z ]*$/",$firstname)) {
        header("Location: ../../user/register.php?error=characterfirst");
        exit();
    } elseif (!preg_match("/^[a-zA-Z ]*$/",$lastname)) {
        header("Location: ../../user/register.php?error=characterlast");
        exit();
    } else {
        //Check using a php function the email submitted is valid
        if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
            header("Location: ../../user/register.php?error=invalidemail");
        } else {
            //Check the email has not already been used to create an account
            $sql = "SELECT email FROM users WHERE email='$email'";
            $result = mysqli_query($conn, $sql);
            $emailcheck = mysqli_num_rows($result);
            if ($emailcheck > 0) {
```



```
header("Location: ../../user/register.php?error=emailinuse");
exit();
} else {
    //Generate the hash using the default key stretching encryption method
    //currently bcrypt
    $hash = password_hash($_POST['password'], PASSWORD_DEFAULT);
    //Add the new user to the database
    $sql = "INSERT INTO users (first_name, last_name, email, hash,
registration_date)
        VALUES ('$firstname', '$lastname', '$email', '$hash', NOW())";
    $result = mysqli_query($conn, $sql);

    //Send an email from noreply@menu-solutions.com confirming account
creation
    $subject = "Welcome to Menu Solutions";
    $txt = "Dear $firstname $lastname, \r\n\r\nThank you for setting up a Menu
Solutions account.
        You can now get started by adding your family members on the family page.
        Then all that's left to do is jump onto the plan page to say who is
eating each day,
        how much time you have to cook and then press generate meal plan. You can
then use your list to collect
        all the ingredients you need to get cooking! For a more detailed tutorial
to show you how to get the most out
        of the site please read the about page or use the tutorial button on the
home page. \r\n\r\n
        I really appreciate your feedback to improve the site so please contact
me through the Menu Solutions page on Facebook if you have any suggestions.
        \r\n\r\nThank you again, \r\n\r\nKind regards \r\n\r\nJack Corbett
\r\nCreator of Menu Solutions";
    $headers = "From: noreply@menu-solutions.com";

    mail($email,$subject,$txt,$headers);

    //Return to the login page
    header("Location: ../../user/login.php");
}
}
}
?>
```



User

login.php

- Fetch error message from the URL using the server superglobal

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <!-- Navigation Bar -->
    <?php
        if (isset($_SESSION['user_id'])) {
            include '../_assets/navbar.loggedin.php';
        } else {
            include '../_assets/navbar.loggedout.php';
        }
    ?>

    <br /><br />

    <!-- Login Form -->
    <div class="w3-container w3-center">
        <h1>Login</h1>
        <div class="w3-card-4">
            <br />
            <form action="../_includes/user/login.inc.php" method="post">
                <div class="input-group margin-bottom-sm">
                    <span class="input-group-addon"><i class="fa fa-user fa-fw"></i></span>
                    <input class="form-control" type="text" name="email"
placeholder="Email" required>
                </div>
                <br />
                <div class="input-group">
                    <span class="input-group-addon"><i class="fa fa-lock fa-fw"></i></span>
                    <input class="form-control" type="password" name="password"
placeholder="Password" required>
                </div>
                <div class="w3-text-red">
                    <?php
                        $url = $_SERVER['REQUEST_URI'];
                        if (strpos($url, 'error=email') !== false) {
                            echo "This email has not been registered for an account";
                        } elseif (strpos($url, 'error=password') !== false) {
                            echo "The password you entered was incorrect";
                        } else {
                    
```



```
        echo "<br />";
    }
?>
</div>
<button type="submit" class="w3-btn w3-hover-dark-grey w3-animate-opacity w3-theme-d1">Login</button>
</form>
<br />

<?php
    if ($id !== NULL) {
?>
<form action="../_includes/user/logout.inc.php">
    <button type="submit" class="w3-btn w3-hover-red w3-animate-opacity w3-theme-d1">Log Out</button>
</form>
<?php
}
?>

<p>If you don't already have an account you can register for one below</p>
<a href="register.php" class="w3-center w3-btn w3-hover-blue w3-animate-opacity w3-theme-d1">Register</a>
<br /><br />

</div>
</div>
<br />
</div>

<?php
    include '../_assets/footer.php'
?>
```

register.php

- Fetch error message from the URL using the server superglobal

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <!-- Navigation Bar -->
<?php
    if (isset($_SESSION['user_id'])) {
        include '../_assets/navbar.loggedin.php';
    } else {
```



```
    include '../_assets/navbar.loggedout.php';
}
?>
<br /><br />

<!-- Login Form -->
<div class="w3-container w3-center">
    <h1>Register</h1>
    <div class="w3-card-4">

        <br />

        <p>Here you can register for your Menu Solutions account, simply fill in the form below and press sign up</p>
        <form action="../_includes/user/register.inc.php" method="post">
            <input type="text" name="firstname" placeholder="Firstname" required><br /><br />
            <input type="text" name="lastname" placeholder="Surname" required><br /><br />
            <input type="text" name="email" placeholder="Email" required><br /><br />
            <input type="password" name="password" placeholder="Password" required><br /><br />
            <button type="submit" class="w3-btn w3-hover-blue w3-animate-opacity w3-theme-d1">Sign Up</button>
        </form>

        <div class="w3-text-red">
            <?php
                $url = $_SERVER['REQUEST_URI'];
                if (strpos($url, 'error=empty') !== false) {
                    echo "Please fill in every field";
                }
                elseif (strpos($url, 'error=characterfirst') !== false) {
                    echo "Only letters and spaces allowed in first name";
                }
                elseif (strpos($url, 'error=characterlast') !== false) {
                    echo "Only letters and spaces allowed in surname";
                }
                elseif (strpos($url, 'error=invalidemail') !== false) {
                    echo "Invalid email address";
                }
                elseif (strpos($url, 'error=emailinuse') !== false) {
                    echo "Email address is already registered to another account, please recover your password or use another email";
                }
            ?>
        </div>
    <br />
</div>
</div>
</div>
```



```
<?php
    include '../_assets/footer.php'
?>
```

admin.php

- jQuery to create additional ingredient fields

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

<?php
    if (isset($_SESSION['user_id'])) {
        include '../_assets/navbar.loggedin.php';
    }
    <br /><br />
<?php
    if ($_SESSION['admin'] == 1) {
    }

        <div class="w3-container">
            <h1>Admin</h1>
            <p>Welcome to the admin page. You can only access this if you have been granted permission.

From here you can help me improve the Menu Solutions experience by adding new recipes which will be used to generate users weekly meal plans.
Simply follow the instructions below to input the details.</p>
<p>1. Give your recipe a name and tell me how long it takes to cook in minutes.<br />
2. Write out numbered instructions making a new line for each new step (the same format as these instructions). Do not press submit until all the recipes have been entered.<br />
3. Fill in the ingredient name (always use the plural with brackets eg: Potato(es) or Carrot(s)), how it is measured, the quantity needed (this is per person and if there is no set portion eg: serve with carrots enter 0) and pick a category. The other category is for ingredients that do not fit into the other categories and the hide category is used for ingredients the user doesn't need on their shopping list such as water.<br />
4. To add the rest of the ingredients just click the Add an Ingredient button and another set of boxes will appear. Repeat this for all ingredients and finish by pressing submit.</p>
<?php
    $url = $_SERVER['REQUEST_URI'];
    if (strpos($url, 'added') !== false) {
        echo "<p class='w3-text-green w3-center'>Your recipe has been added to the
```



```
database and can now be chosen in plan generation.</p>";  
    } elseif (strpos($url, 'failed') !== false) {  
        echo "<p class='w3-text-red w3-center'>Your recipe could not be added to  
the database as some of the information had not been completed.</p>";  
    }  
?>  
</div>  
  
<div class="w3-card-4">  
    <br />  
    <form action="../../_includes/admin/addrecipe.inc.php" method="POST"  
class="w3-padding w3-center">  
        <!--Recipe Name-->  
        <input type="text" name="name" placeholder="Recipe Name" class="w3-half"  
required>  
        <!--Cook Time-->  
        <input type="text" name="cooktime" placeholder="Cook Time" class="w3-half"  
required>  
        <br /><br />  
        <!--Instructions-->  
        <textarea ROWS=3 COLS=30 name="instructions" placeholder="Instructions"  
style="width:100%" required></textarea><br /><br />  
        <button value="submit" class="w3-btn w3-hover-blue w3-animate-opacity  
w3-theme-d1">Submit</button>  
        <br /><br />  
        <button id="addFields" class="w3-btn w3-dark-grey" type="button">Add an  
Ingredient</button>  
        <br /><br />  
        <div class="w3-border w3-padding">  
            <!--Ingredient Name-->  
            <input class="w3-third" type="text" name="ingredient_name[]"  
placeholder="Ingredient Name" required>  
            <!--Measurement-->  
            <input class="w3-third" type="text" name="ingredient_measurement[]"  
placeholder="Measurement">  
            <!--Quantity-->  
            <input class="w3-third" type="text" name="ingredient_quantity[]"  
placeholder="Quantity">  
            <br /><br />  
            <!--Category-->  
            <select class="w3-select w3-border" name="ingredient_category[]" required>  
                <option value="" disabled selected>Choose a category</option>  
                <option value="1">Fruit and Veg</option>  
                <option value="2">Protein</option>  
                <option value="3">Carbohydrates</option>  
                <option value="4">Dairy</option>  
                <option value="6">Other</option>  
                <option value="5">Hide</option>  
            </select>  
        </div>  
        <br />  
</form>
```



```
</div>
<br />
</div>

<?php
} else {
    echo "<p class='w3-text-red w3-center'>You do not have permission to access
this page.</p>";
}
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>

</div>

<?php
    include '../_assets/footer.php';
?>

<script>
$(function ($) {
    $('body').on("click", '#addFields', function () {
        $('form').append('<div class="w3-border w3-padding"> \
            <input class="w3-third" type="text" name="ingredient_name[]" \
placeholder="Ingredient Name"> \
            <input class="w3-third" type="text" \
name="ingredient_measurement[]" placeholder="Measurement"> \
            <input class="w3-third" type="text" name="ingredient_quantity[]" \
placeholder="Quantity"> \
            <br /><br /> \
            <select class="w3-select w3-border" \
name="ingredient_category[]"> \
                <option value="" disabled selected>Choose a category</option> \
                <option value="1">Fruit and Veg</option> \
                <option value="2">Protein</option> \
                <option value="3">Carbohydrates</option> \
                <option value="4">Dairy</option> \
                <option value="6">Other</option> \
                <option value="5">Hide</option> \
            </select> \
        </div> \
        <br />');
    })
})(jQuery);
</script>
```



Family

addmember.php

- Fetch error message from the URL using the server superglobal

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <?php
        if (isset($_SESSION['user_id'])) {
            include '../_assets/navbar.loggedin.php';
        }
        <br /><br />

    <div class="w3-container w3-center">
        <h1>Add</h1>

        <div class="w3-animate-opacity w3-card-4">
            <br />
            <form action="../_includes/family/add-member.inc.php" method="post">
                <input type="text" name="firstname" placeholder="First Name" required><br /><br />
                <input type="text" name="lastname" placeholder="Surname" required><br /><br />
                <input type="radio" name="gender" value="m" checked> Male &nbsp;
                <input type="radio" name="gender" value="f" checked> Female &nbsp;
                <input type="radio" name="gender" value="o" checked> Other<br /><br />
                Selected by default:<br />
                <input type="radio" name="default_ticked" value="1" checked> Yes
&nbsp;
                <input type="radio" name="default_ticked" value="0" checked> No &nbsp;
            <br /><br />

            <button type="submit" class="w3-btn w3-hover-dark-grey w3-theme-d1 w3-round-xlarge"><i class="fa fa-user-plus" aria-hidden="true"></i></button>
            <a href="members.php" class="w3-btn w3-hover-dark-grey w3-red w3-round-xlarge"><i class="fa fa-times" aria-hidden="true"></i></a>
        </form>
        <br />
    </div>
    <br />
</div>
```



```
<div class="w3-text-red w3-center">
<?php
$url = $_SERVER['REQUEST_URI'];
if (strpos($url, 'error=empty') !== false) {
    echo "Please complete both first name and surname";
}
?>
</div>
<?php
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>
</div>

<?php
    include '../_assets/footer.php'
?>
```

editmember.php

- Use of PHP to populate edit member form

```
<?php
include '../_assets/header.php';
include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

<?php
if (isset($_SESSION['user_id'])) {
    include '../_assets/navbarloggedin.php';
}
?>
<br /><br />

<div class="w3-container w3-center">
<h1>Edit</h1>

<div class="w3-card-4 w3-animate-opacity">

<br />

<?php
    include '../_includes/family/edit-fetch.inc.php';
?>

<form action="../_includes/family/edit-update.inc.php" method="post">
<input autocomplete="off" value="<?php echo $first_name; ?>" type="text"
```



```
name="firstname" placeholder="First Name" required><br /><br />

    <input autocomplete="off" value="<?php echo $last_name; ?>" type="text"
name="lastname" placeholder="Surname" required><br /><br />

    <input type="radio" name="gender" value="m" <?php if ($gender == "m") {
echo "checked"; ?>> &emsp; Male &emsp;

    <input type="radio" name="gender" value="f" <?php if ($gender == "f") {
echo "checked"; ?>> &emsp; Female &emsp;

    <input type="radio" name="gender" value="o" <?php if ($gender == "o") {
echo "checked"; ?>> &emsp; Other<br /><br />

    Selected by default:<br />
    <input type="radio" name="default_ticked" value="1" <?php if
($default_ticked == 1) { echo "checked"; }?>> &emsp; Yes &emsp;
    <input type="radio" name="default_ticked" value="0" <?php if
($default_ticked == 0) { echo "checked"; }?>> &emsp; No &emsp;
    <br /><br />

    <input type="hidden" name="memberid" value="<?php echo $member_id; ?>">

    <button type="submit" class="w3-btn w3-hover-dark-grey w3-theme-d1
w3-round-xlarge"><i class="fa fa-check" aria-hidden="true"></i></button>
    <a href="members.php" class="w3-btn w3-hover-dark-grey w3-red
w3-round-xlarge"><i class="fa fa-times" aria-hidden="true"></i></a>
    </form>

    <br />

    </div>
</div>

<?php
    } else {
        include '../_assets/navbar.loggedout.php';
        include '../_assets/loginerror.php';
    }
?>

</div>

<?php
    include '../_assets/footer.php'
?>
```



members.php

- Output dependant on query, which generates links that use GET method to pass information

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

<?php
    if (isset($_SESSION['user_id'])) {
        include '../_assets/navbar.loggedin.php';
    }
    <br /><br />
    <div class="w3-container">
        <h1>Family</h1>
        <p>Here you can see the family members that have been added and their individual profiles. You can edit your submitted information for each person or add a new family member by pressing the button below. This information will be used on the plan page to tell us who will be eating each day.</p>
    </div>
    <!-- Members table -->
    <ul class="w3-ul w3-card-4 w3-animate-opacity">
        <?php
            include '../_includes/dbh.inc.php';
            include '../_includes/family/members-fetch.inc.php';

            while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {

                $name = $row['name'];
                $member_id = $row['member_id'];
                $gender = $row['gender'];
            }
            <li class="w3-padding-16">
                <a href="removemember.php?member=<?php echo $member_id; ?>">
                    <?php echo $member_id; ?>
                </a>
                <a href="editmember.php?member=<?php echo $member_id; ?>">
                    <?php echo $member_id; ?>
                </a>
                <i class="fa fa-pencil w3-hover-text-blue" aria-hidden="true"></i>
                <img src=
                    <?php
                        if ($gender == 'm') {
                            echo '../_img/male_user.png';
                        } elseif ($gender == 'f') {
                            echo '../_img/female_user.png';
                        } else {
                    
```



```
        echo '../_img/other_user.png';
    }
?>
    class="w3-left w3-circle w3-margin-right" style="width:30px">
        <span class="w3-large"><?php echo $name; ?></span>
    <br />
</li>

<?php
    }
?>
</ul>

<br />

<!-- Add family member button -->
<div class="w3-container w3-padding">
    <a class="w3-btn-floating w3-theme-d1 w3-hover-dark-grey"
style="text-decoration: none" href="addmember.php" >+</a>
</div>
<br />
<?php
    } else {
        include '../_assets/navbar.loggedout.php';
        include '../_assets/loginerror.php';
    }
?>
</div>

<?php
    include '../_assets/footer.php';
?>
```

removemember.php

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <?php
        if (isset($_SESSION['user_id'])) {
            include '../_assets/navbar.loggedin.php';
        }
    <br /><br />

    <div class="w3-container w3-center">
        <h1>Remove</h1>
```



```
<div class="w3-card-4">
<br />

<?php
    include '../_includes/family/remove-fetch.inc.php';
    echo "<p>Do you want to remove " . $row['name'] . "?</p>";
?>

    <a href="../_includes/family/remove-delete.inc.php?member=<?php echo
$member_id; ?>" class="w3-btn w3-hover-dark-grey w3-theme-d1 w3-round-xlarge"><i
class="fa fa-check" aria-hidden="true"></i></a>
    <a href="members.php" class="w3-btn w3-hover-dark-grey w3-red
w3-round-xlarge"><i class="fa fa-times" aria-hidden="true"></i></a>

    <br /><br />
</div>
</div>

<?php
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>

</div>

<?php
    include '../_assets/footer.php'
?>
```

Plan

plan.php

- Javascript to create tab bar for each day
- Default selection of members based on default_ticked

```
<?php
include '../_assets/header.php';
include '../_assets/standardstyle.php';
?>

<style>
    .day {display:none;}
</style>

<script>
    function openDay(evt, dayName) {
        var i, x, tablinks;
```



```
x = document.getElementsByClassName("day");
for (i = 0; i < x.length; i++) {
    x[i].style.display = "none";
}
tablinks = document.getElementsByClassName("tablink");
for (i = 0; i < x.length; i++) {
    tablinks[i].className = tablinks[i].className.replace(" w3-dark-grey",
 ""));
}
document.getElementById(dayName).style.display = "block";
evt.currentTarget.className += " w3-dark-grey";
}
</script>

<div class="page-wrap">\

<!-- Navigation Bar -->
<?php
if (isset($_SESSION['user_id'])) {
    include '../_assets/navbar.loggedin.php';
?>
<br /><br />

<div class="w3-container">
    <h1>Plan</h1>
    <p>Simply enter which family members are eating and how long you have
to cook by selecting each day from the bar below.
        We will use this to generate your calendar for the coming week.</p>
</div>

<div class="w3-card-4">
    <ul class="w3-navbar w3-theme-d1">
        <li><a href="#" class="tablink" onclick="openDay(event,
'monday');">Monday</a></li>
        <li><a href="#" class="tablink" onclick="openDay(event,
'tuesday');">Tuesday</a></li>
        <li><a href="#" class="tablink" onclick="openDay(event,
'wednesday');">Wednesday</a></li>
        <li><a href="#" class="tablink" onclick="openDay(event,
'thursday');">Thursday</a></li>
        <li><a href="#" class="tablink" onclick="openDay(event,
'friday');">Friday</a></li>
        <li><a href="#" class="tablink" onclick="openDay(event,
'saturday');">Saturday</a></li>
        <li><a href="#" class="tablink" onclick="openDay(event,
'sunday');">Sunday</a></li>
    </ul>

<form action=".../_includes/plan/plan-generation.inc.php" method="post">
    <?php
        include '../_includes/plan/set-up-form.inc.php';
```



```
//Loops through the days of the week
for ($i = 0; $i < 7; $i++) {
?>
<div id=<?php echo $day[$i]; ?>" class="w3-container w3-border day">

<p>Select who will be eating: </p>

    <div class="w3-row-padding">
<?php
//Find how many members there are in the users family
$no_members = count($name);

//Loop to write out a checkbox for each family member
for($x = 0; $x < $no_members; $x++) {
?>
    <div class="w3-third w3-card-2 w3-padding-8">
        <img src=
<?php if ($gender[$x] == 'm') {
    echo '../_img/male_user.png';
} elseif ($gender[$x] == 'f') {
    echo '../_img/female_user.png';
} else {
    echo '../_img/other_user.png';
}?>
        class="w3-left w3-circle w3-margin-right" style="width:30px">
            <input class="w3-check" name=<?php echo $day[$i] . '[]' ?>
?> type="checkbox"
        <?php if ($default_ticked[$x] == 1)
{ echo "checked"; }?> value=<?php echo $member_id[$x]; ?>">
            <label class="w3-validate">&nbsp;<?php echo $name[$x];
?></label>
    </div>
    <?php
    }
?>
</div>

<br />
<p>Select the number of minutes you have available to cook: </p>

    <div class="w3-half">
        <select class="w3-select w3-border" name=<?php echo $day[$i] .
'_time' ?>">
            <option selected value="90">90</option>
            <option value="60">60</option>
            <option value="30">30</option>
        </select>
    </div>

<br /><br />
</div>
```



```
<?php
    }
?
</div>

<br />

<div class="w3-text-red w3-center">
<?php
    $url = $_SERVER['REQUEST_URI'];
    if (strpos($url, 'removed') !== false) {
        echo "<p>Your plan for the coming week has been deleted. You can now
generate a new one.</p>";
    }
?
</div>

<div class="w3-center">
    <button class="w3-btn w3-hover-blue w3-dark-grey" type="submit">Generate Meal
Plan</button>
</div>

</form>

<br />

<div class="w3-center">
    <a href="
<?php
    include '../_includes/plan/get-max-plan.inc.php';
    if ($plan_id === NULL) {
        echo '#';
    } else {
        echo 'removeplan.php?delete';
    }
?
" class="w3-btn
<?php
    if ($plan_id === NULL) {
        echo 'w3-disabled';
    }
?
w3-hover-red w3-dark-grey">Delete Current Plan</a>
</div>

<br /><br />

<?php
}
if (isset($_SESSION['user_id']) == false) {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
```



```
}
```

```
?>
```

```
</div>
```

```
<?php
```

```
include '../_assets/footer.php';
```

```
?>
```

removeplan.php

- Fetch message from the URL using the server superglobal to provide context to the user

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

<?php
    if (isset($_SESSION['user_id'])) {
        include '../_assets/navbar.loggedin.php';
    }
    <br /><br />

<div class="w3-container w3-center">
    <h1>Delete Plan</h1>

    <div class="w3-card-4 w3-center">
        <br />
        <p>
            <?php
                $url = $_SERVER['REQUEST_URI'];
                if (strpos($url, 'delete') !== false) {
                    echo "Do you want to delete your latest plan?";
                } else {
                    echo "You already have a plan for the coming week. Do you wish to delete it?";
                }
            ?>
            </p>
            <a href="../_includes/plan/remove-plan.inc.php" class="w3-btn w3-hover-dark-grey w3-theme-d1 w3-round-xlarge"><i class="fa fa-check" aria-hidden="true"></i></a>
            <a href="plan.php" class="w3-btn w3-hover-dark-grey w3-red w3-round-xlarge"><i class="fa fa-times" aria-hidden="true"></i></a>
        <br /><br />
    </div>
</div>
```



```
<?php
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>
</div>

<?php
    include '../_assets/footer.php';
?>
```

Calendar

calendar.php

- Use of an associative array
- SQL aggregate functions
- Conditional formatting for platter images using MOD

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <!-- Navigation Bar -->
    <?php
        if (isset($_SESSION['user_id'])) {
            include '../_assets/navbar.loggedin.php';
        }
        <br /><br />

        <div class="w3-container">
            <h1>Calendar</h1>
            <p>Here you can view your meal plan for the week with the number of people
            eating each day on the side. Simply click the recipe for today and get cooking.</p>
        </div>

    <?php
        include '../_includes/dbh.inc.php';

        $day = array(0 => 'Monday', 1 => 'Tuesday', 2 => 'Wednesday', 3 => 'Thursday',
        4 => 'Friday', 5 => 'Saturday', 6 => 'Sunday');

        $id = $_SESSION['user_id'];

        $sql = "SELECT MAX(plan_id) plan_id FROM users_plan WHERE user_id = '$id'";
        $result = @mysqli_query($conn, $sql);
```



```
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
$plan_id = $row['plan_id'];

if ($plan_id === NULL) {
?>
<p class="w3-text-red w3-center">To view your meal plan you must first visit the
plan page and generate one.</p>

<?php
} else {
?>

<!-- Meal Plan table -->
<ul class="w3-ul w3-card-4 w3-animate-opacity">

<?php
$sql = "SELECT recipe_id, no_eating, date_eating FROM plan_recipes WHERE
plan_id = '$plan_id' ORDER BY date_eating ASC";
$result = @mysqli_query($conn, $sql);

while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
    $recipe_id[] = $row['recipe_id'];
    $no_eating[] = $row['no_eating'];
    $date_eating[] = date_create($row['date_eating']);
}

for ($i=0; $i<7; $i++) {
?>
<li class="w3-padding-16">

    
    <span class="w3-large"><?php echo $day[$i] . " - " .
date_format($date_eating[$i],"d/m"); ?></span>

    <br />

<?php
$sql = "SELECT name FROM recipes WHERE recipe_id = '$recipe_id[$i]'";
$result = @mysqli_query($conn, $sql);
$row = mysqli_fetch_array($result, MYSQLI_ASSOC);
$recipe_name = $row['name'];
?>

    <a href=<?php echo 'dayrecipe.php?' . $recipe_id[$i]; ?>" 
class="w3-medium" style="text-decoration: none"><?php echo $recipe_name; ?></a>

    <?php
for ($x=1; $x<$no_eating[$i]+1; $x++) {

        if (fmod($x,6) == 0) {
            echo "<br />";
```



```
        }

    ?>

        <?php
        }
    ?>
    <br />
</li>
<?php
}
?>
</ul>

<br />
<?php
}
?>
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>

</div>

<?php
    include '../_assets/footer.php'
?>
```

dayrecipe.php

- Filter the URL to retrieve the recipe_id

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <!-- Navigation Bar -->
    <?php
        if (isset($_SESSION['user_id'])) {
            include '../_assets/navbar.loggedin.php';
        }
    ?>
    <br /><br />

    <div class="w3-container w3-animate-opacity">
        <?php
```



```
include "../_includes/dbh.inc.php";

//Takes the URL
$url = $_SERVER['REQUEST_URI'];
//Picks out the numbers from the URL which dictates the recipe id
$recipe_id = filter_var($url, FILTER_SANITIZE_NUMBER_INT);

if ($recipe_id === NULL) {
    echo "<p class='w3-text-red w3-center'>Error</p><p class='w3-center'>Please
use the calendar page and click the recipe you want to view to access this
page</p>";
} else {

    $sql = "SELECT name, instructions, cook_time FROM recipes WHERE recipe_id =
'$recipe_id'";
    $result = @mysqli_query($conn, $sql);
    $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
    $recipe_name = $row['name'];
    $recipe_instructions = nl2br($row['instructions']);
    $cook_time = $row['cook_time'];

    $sql = "SELECT ingredient_id, quantity FROM ingredients_recipes WHERE
recipe_id = '$recipe_id'";
    $result = @mysqli_query($conn, $sql);
    while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        $ingredient_id[] = $row['ingredient_id'];
        $quantity[] = $row['quantity'];
    }

    $sql = "SELECT no_eating FROM plan_recipes WHERE recipe_id = '$recipe_id'";
    $result = @mysqli_query($conn, $sql);
    while ($row = mysqli_fetch_array($result, MYSQLI_ASSOC)) {
        $no_eating = $row['no_eating'];
    }
?>

<div class="w3-container w3-card-4 w3-bottombar">
    <h2><?php echo $recipe_name; ?></h2>
    </div>

    <h3>Ingredients:</h3>

    <p>
        <?php
            $amount = array();
            for ($i=0; $i<COUNT($ingredient_id); $i++) {
                $amount[$i] = $quantity[$i] * $no_eating;
                $sql = "SELECT ingredient_name, measurement FROM ingredients WHERE
'$ingredient_id[$i]' = ingredient_id";
                $result = @mysqli_query($conn, $sql);
                $row = mysqli_fetch_array($result, MYSQLI_ASSOC);
                $measurement = $row['measurement'];
            }
        <?php
    </p>
```



```
$name = $row['ingredient_name'];
if ($amount[$i] > 0) {
    if ($measurement == "g" OR $measurement == "ml") {
        echo $amount[$i] . $measurement . " " . $name;
    } else {
        echo $amount[$i] . " " . $measurement . " " . $name;
    }
} else {
    echo $name;
}
echo "<br />";
?
</p>

<h3>Instructions:</h3>
<p><?php echo $recipe_instructions; ?></p>
<h5>Cook Time: <?php echo $cook_time; ?> minutes</h5>
<?php
    ?
    ?
    ?
    ?
</div>

<?php
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>

</div>

<?php
    include '../_assets/footer.php';
?>
```



List

list.php

- Conditional formatting of mobile table for ‘other’ category using round and count

```
<?php
    include '../_assets/header.php';
    include '../_assets/standardstyle.php';
?>

<div class="page-wrap">

    <!-- Navigation Bar -->
    <?php
        if (isset($_SESSION['user_id'])) {
            include '../_assets/navbar.loggedin.php';
        }><br /><br />

        <!-- Title -->
        <div class="w3-container">
            <h1>List</h1>
            <p>Here you can see all the ingredients for the recipes for your current
plan.</p>
        </div>

    <?php
        include '../_includes/dbh.inc.php';
        $id = $_SESSION['user_id'];

        //Finds the latest plan
        include '../_includes/plan/get-max-plan.inc.php';

        if ($plan_id === NULL) {
    ?>
        <p class="w3-text-red w3-center">To view your shopping list you must first visit
the plan page and generate a meal plan.</p>
        <?php
            } else {
                include '../_includes/list/list.inc.php';
            }>

        <!-- Table for desktops -->
        <table class="w3-table-all w3-card-4 w3-hide-small w3-hide-medium">
            <tr>
                <!-- Displays the full grid on desktops -->
                <th>Fruit and Vegetables</th>
                <th>Protein</th>
                <th>Carbohydrates</th>
                <th>Dairy</th>
```



```
<th>Other</th>
</tr>

<tr>
    <!--Output all the fruit and vegetables-->
    <td>
        <?php
            OutputArray($fv_amount, $fv_measure, $fv_name);
        ?>
    </td>

    <!--Outputs all the protein-->
    <td>
        <?php
            OutputArray($pr_amount, $pr_measure, $pr_name);
        ?>
    </td>

    <!--Outputs all the carbohydrates-->
    <td>
        <?php
            OutputArray($ca_amount, $ca_measure, $ca_name);
        ?>
    </td>

    <!--Outputs all the dairy-->
    <td>
        <?php
            OutputArray($da_amount, $da_measure, $da_name);
        ?>
    </td>

    <!--Outputs all the other ingredients needed-->
    <td>
        <?php
            OutputArray($un_amount, $un_measure, $un_name);
        ?>
    </td>
</tr>
</table>

<!-- Table for mobile devices -->


||
||
||


```



```
<?php
    OutputArray($fv_amount, $fv_measure, $fv_name);
?>
</td>

<!--Outputs all the protien--&gt;
&lt;td&gt;
    &lt;?php
        OutputArray($pr_amount, $pr_measure, $pr_name);
    ?&gt;
&lt;/td&gt;
&lt;/tr&gt;

&lt;tr&gt;
    &lt;th&gt;Carbohydrates&lt;/th&gt;
    &lt;th&gt;Dairy&lt;/th&gt;
&lt;/tr&gt;

&lt;tr&gt;
    <!--Outputs all the carbohydrates--&gt;
    &lt;td&gt;
        &lt;?php
            OutputArray($ca_amount, $ca_measure, $ca_name);
        ?&gt;
    &lt;/td&gt;

    &lt;!--Outputs all the dairy--&gt;
    &lt;td&gt;
        &lt;?php
            OutputArray($da_amount, $da_measure, $da_name);
        ?&gt;
    &lt;/td&gt;
&lt;/tr&gt;

&lt;tr&gt;
    &lt;th&gt;Other&lt;/th&gt;
    &lt;th&gt;&lt;/th&gt;
&lt;/tr&gt;

&lt;tr&gt;
    <!--Outputs all the other ingredients needed--&gt;
    &lt;td&gt;
        &lt;?php
            $half = round(count($un_name) / 2);
            OutputArray($un_amount, $un_measure, $un_name, $half)
        ?&gt;
    &lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;

&lt;br /&gt;</pre>
```



```
</div>
<?php
}
} else {
    include '../_assets/navbar.loggedout.php';
    include '../_assets/loginerror.php';
}
?>

</div>

<?php
    include '../_assets/footer.php'
?>
```

Assets

footer.php

```
<div class="site-footer">
    <footer class="w3-center w3-theme-d1 w3-topbar">
        <div class="w3-container">
            <div class="w3-left">
                <h3>
                    <a href="https://www.facebook.com/Jack.Corbett.47"
class="w3-hover-text-indigo"><i class="fa fa-facebook-official"></i></a>
                    <a href="https://www.twitter.com/JackCorbett7"
class="w3-hover-text-light-blue"><i class="fa fa-twitter"></i></a>
                    <a href="https://www.linkedin.com/in/jack-corbett"
class="w3-hover-text-indigo"><i class="fa fa-linkedin"></i></a>
                </h3>
            </div>
            <div class="w3-right">
                <p>MENU SOLUTIONS - JACK CORBETT</p>
            </div>
        </div>
    </footer>
</div>

</body>
</html>
```

header.php

```
<?php
    session_start();
?>

<!DOCTYPE html>
<html>
<head>
```



```
<title>Menu Solutions</title>
<meta charset="UTF-8">
<meta name="description" content="The meal planner for the modern family.
Generate simple plans that fit in around your busy life and enable
everyone to eat home cooked healthy meals. It will even make you a shopping
list, so you know exactly what to buy at the store.">
<meta name="keywords" content="meal, meals, planner, recipes, menu, solutions,
family, plan">
<meta name="author" content="Jack Corbett">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0, user-scalable=no"/>
<link rel="stylesheet" href="../../css/main.css">
<link rel="stylesheet" href="../../css/theme.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Lato">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.6.3/css/font-awesome.min.css">
<link rel="shortcut icon" href="https://www.menu-solutions.com/_img/favicon.ico">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
</head>
<body>
```

loginerror.php

```
<br /><br />
<div class="w3-text-red w3-center">
<p>You need to login to access this page</p>
<a href="../user/login.php" class=" w3-btn w3-hover-dark-grey
w3-theme-d1">Login</a>
</div>
```

navbar.loggedin.php

- Javascript to allow the user to view the mobile navigation bar
- Concatenate user's name to display in navigation bar

```
<!-- Navigation for desktops -->
<div class="w3-top">
<ul class="w3-navbar w3-card-2 w3-dark-grey w3-left-align">
<li class=" w3-hide-large w3-opennav w3-right">
<a class="w3-padding-large" href="javascript:void(0)" onclick="showMenu()"
title="Toggle Navigation Menu"><i class="fa fa-bars" aria-hidden="true"></i></a>
</li>
<li>
<a href="/" class="w3-padding-large w3-hover-none
w3-hover-text-light-grey">HOME</a>
</li>
<li class="w3-hide-small w3-hide-medium w3-dropdown-hover">
<a href="javascript:void(0)" class="w3-padding-large" title="More">FAMILY <i
class="fa fa-caret-down" aria-hidden="true"></i></a>
```



```
<div class="w3-dropdown-content w3-card-4">
    <a href="../family/members.php" class="w3-padding-large">MEMBERS</a>
        <a href="../family/addmember.php" class="w3-padding-large">ADD MEMBER</a>
    </div>
</li>
<li class="w3-hide-small w3-hide-medium">
    <a href="../plan/plan.php" class="w3-padding-large">PLAN</a>
</li>
<li class="w3-hide-small w3-hide-medium">
    <a href="../calendar/calendar.php" class="w3-padding-large">CALENDAR</a>
</li>
<li class="w3-hide-small w3-hide-medium">
    <a href="../list/list.php" class="w3-padding-large">LIST</a>
</li>
<li class="w3-hide-small w3-hide-medium">
    <a href="../about.php" class="w3-padding-large">ABOUT</a>
</li>
<?php
    if ($_SESSION['admin'] == 1) {
?>
    <li class="w3-hide-small w3-hide-medium">
        <a href="../user/admin.php" class="w3-padding-large w3-hover-blue">ADMIN</a>
    </li>
<?php
    }
?>
<div class="w3-right">
    <?php
        include $_SERVER['DOCUMENT_ROOT'] . '/_includes/dbh.inc.php';
        $id = $_SESSION['user_id'];

        $sql = "SELECT CONCAT(first_name, ' ', last_name) AS name FROM users WHERE
user_id = '$id'";
        $result = @mysqli_query($conn, $sql);
        $row = mysqli_fetch_array($result, MYSQLI_ASSOC);

        echo "<li class='w3-hide-small w3-hide-medium w3-padding-large'>" .
$row['name'] . "</li>";
    ?>
    <li class="w3-hide-small w3-hide-medium">
        <a href="/_includes/user/logout.inc.php" class="w3-padding-large
w3-hover-red">LOGOUT</a>
    </li>
    </div>
</ul>
</div>

<!-- Navigation Bar for small devices -->
<div id="smallNav" class="w3-hide w3-hide-large w3-top" style="margin-top:50px">
    <ul class="w3-navbar w3-left-align w3-dark-grey">
        <li><a href="../family/members.php" class="w3-padding-large">FAMILY</a></li>
        <li><a href="../plan/plan.php" class="w3-padding-large">PLAN</a></li>
```



```
<li><a href="../calendar/calendar.php" class="w3-padding-large">CALENDAR</a></li>
<li><a href="../list/list.php" class="w3-padding-large">LIST</a></li>
<li><a href="../about.php" class="w3-padding-large">ABOUT</a></li>
<?php
    if ($_SESSION['admin'] == 1) {
?
    <li><a href="../user/admin.php" class="w3-padding-large w3-hover-blue">ADMIN</a></li>
<?php
    }
?
    <li class="w3-right">
        <a href="../_includes/user/logout.inc.php" class="w3-padding-large w3-hover-red">LOGOUT</a>
    </li>
</ul>
</div>

<script>
    // Used to toggle the menu on small screens when clicking on the menu button
    // (three bars)
    function showMenu() {
        var x = document.getElementById("smallNav");
        if (x.className.indexOf("w3-show") == -1) {
            x.className += " w3-show";
        } else {
            x.className = x.className.replace(" w3-show", "");
        }
    }
</script>
```

navbar.loggedout.php

```
<div class="w3-top">
    <ul class="w3-navbar w3-card-2 w3-dark-grey w3-left-align">
        <li class="w3-hide-medium w3-hide-large w3-opennav w3-right">
            <a class="w3-padding-large" href="javascript:void(0)" onclick="showMenu()" title="Toggle Navigation Menu"><i class="fa fa-bars"></i></a>
        </li>
        <li>
            <a href="/" class="w3-padding-large w3-hover-none w3-hover-text-light-grey">HOME</a>
        </li>
        <li class="w3-hide-small">
            <a href="../about.php" class="w3-padding-large">ABOUT</a>
        </li>
        <li class="w3-right w3-hide-small">
            <a href="../user/login.php" class="w3-padding-large w3-hover-theme">LOGIN</a>
        </li>
    </ul>
```



```
</div>

<!-- Navigation Bar for small devices -->
<div id="smallNav" class="w3-hide w3-hide-large w3-hide-medium w3-top"
style="margin-top:50px">
    <ul class="w3-navbar w3-left-align w3-dark-grey">
        <li>
            <a href="../about.php" class="w3-padding-large">ABOUT</a>
        </li>
        <li class="w3-right">
            <a href="../user/login.php" class="w3-padding-large w3-hover-theme">LOGIN</a>
        </li>
    </ul>
</div>

<script>
// Used to toggle the menu on small screens when clicking on the menu button
function showMenu() {
    var x = document.getElementById("smallNav");
    if (x.className.indexOf("w3-show") == -1) {
        x.className += " w3-show";
    } else {
        x.className = x.className.replace(" w3-show", "");
    }
}
</script>
```

standardstyle.php

```
<style>
body,h1,h2,h3,h4,h5,h6 {font-family: "Lato", sans-serif;}
body, html {
    height: 100%;
    color: #777;
    line-height: 1.8;
}
</style>
```



SQL

- Composite primary keys in link tables
- Foreign key constraints to mirror deletion of members/plans/recipes

```
CREATE TABLE `family` (
  `member_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
  `user_id` mediumint(8) unsigned NOT NULL,
  `first_name` varchar(20) NOT NULL,
  `last_name` varchar(40) NOT NULL,
  `gender` enum('m','f','o') NOT NULL,
  `default_ticked` tinyint(1) NOT NULL,
  PRIMARY KEY (`member_id`),
  KEY `user_id` (`user_id`),
  CONSTRAINT `family_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users` (`user_id`)
ON DELETE CASCADE ON UPDATE CASCADE
)

CREATE TABLE `ingredients` (
  `ingredient_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
  `ingredient_name` varchar(25) NOT NULL,
  `measurement` varchar(20) NOT NULL,
  `category` tinyint(4) NOT NULL,
  PRIMARY KEY (`ingredient_id`)
)

CREATE TABLE `ingredients_recipes` (
  `ingredient_id` mediumint(8) unsigned NOT NULL,
  `recipe_id` mediumint(8) unsigned NOT NULL,
  `quantity` decimal(5,2) NOT NULL,
  PRIMARY KEY (`ingredient_id`,`recipe_id`),
  KEY `recipe_id` (`recipe_id`),
  CONSTRAINT `ingredients_recipes_ibfk_1` FOREIGN KEY (`ingredient_id`) REFERENCES
`ingredients` (`ingredient_id`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `ingredients_recipes_ibfk_2` FOREIGN KEY (`recipe_id`) REFERENCES
`recipes` (`recipe_id`) ON DELETE CASCADE ON UPDATE CASCADE
)

CREATE TABLE `plan` (
  `plan_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
  `start_date` date NOT NULL,
  `end_date` date NOT NULL,
  PRIMARY KEY (`plan_id`)
)

CREATE TABLE `plan_recipes` (
  `plan_id` mediumint(8) unsigned NOT NULL,
  `recipe_id` mediumint(8) unsigned NOT NULL,
  `no_eating` int(8) NOT NULL,
  `date_eating` date NOT NULL,
```



```
PRIMARY KEY (`plan_id`, `recipe_id`),
KEY `recipe_id` (`recipe_id`),
CONSTRAINT `plan_recipes_ibfk_1` FOREIGN KEY (`plan_id`) REFERENCES `plan`(`plan_id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `plan_recipes_ibfk_2` FOREIGN KEY (`recipe_id`) REFERENCES `recipes`(`recipe_id`) ON DELETE CASCADE ON UPDATE CASCADE
)

CREATE TABLE `recipes` (
`recipe_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
`name` varchar(30) NOT NULL,
`instructions` text NOT NULL,
`cook_time` int(11) NOT NULL,
PRIMARY KEY (`recipe_id`)
)

CREATE TABLE `users` (
`user_id` mediumint(8) unsigned NOT NULL AUTO_INCREMENT,
`first_name` varchar(20) NOT NULL,
`last_name` varchar(40) NOT NULL,
`email` varchar(60) NOT NULL,
`hash` varchar(255) NOT NULL,
`registration_date` datetime NOT NULL,
`admin` tinyint(1) NOT NULL,
PRIMARY KEY (`user_id`),
UNIQUE KEY `email` (`email`),
KEY `login` (`hash`, `email`),
KEY `full_name` (`first_name`, `last_name`)
)

CREATE TABLE `users_plan` (
`user_id` mediumint(8) unsigned NOT NULL,
`plan_id` mediumint(8) unsigned NOT NULL,
PRIMARY KEY (`user_id`, `plan_id`),
KEY `user_plan_ibfk_2` (`plan_id`),
CONSTRAINT `users_plan_ibfk_1` FOREIGN KEY (`user_id`) REFERENCES `users`(`user_id`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `users_plan_ibfk_2` FOREIGN KEY (`plan_id`) REFERENCES `plan`(`plan_id`) ON DELETE CASCADE ON UPDATE CASCADE
)
```



Evaluation

Overall Effectiveness of System

I believe my system provides an excellent solution to the issues of meal planning. I used all the feedback I collected from my end user and my online survey to construct a web app that is tailored to their needs. With a simple UI, wide device scalability and complex algorithms that automate the entire experience, I have delivered on all of their suggestions. During the sites growth it has attracted a lot of attention from potential customers and so I will be continuing development, with a look to creating a commercial product in the future. This way I can reach the people who need the help the most, to change their diets and eating habits to live healthier, easier lives.

Completion of Objectives

Account

Each user is able to create an account using an email and password encrypted using SHA2 encryption

I actually ended up surpassing this objective by implementing the bcrypt hashing algorithm, which is much stronger than SHA2 due to its key stretching nature. My account creation is really easy to use and a confirmation email is even sent to confirm the account has been set up.

Access to user specific pages is blocked if not logged in to a valid account

All pages containing users information not only rely on a valid session to fetch the data in the first place but display an error message if a session has not yet been created. The login form is also protected against SQL injection by escaping submitted strings to protect users accounts.

The login and register form is validated to limit errors before the information is submitted to the database using RegEx and the user is presented a specific error message to inform them of the issue

My form validation for the login and register pages performs according to my design flowcharts and uses the URL to return a specific error message to the user so they know what was incorrect.



Jack Corbett

Users remain logged in between web pages giving them access to all of their information until they log out

Sessions are managed across the site meaning the user will remain logged in moving between pages until they logout and the session is ended.

Family

The user can add members to their family

It is easy to add members to your family either through the add member link in the navigation bar or through the + button at the bottom of the members list. All the user has to do is fill in a simple form. The PHP scripts handle the rest to add the member to the database and link them with the user's account.

Members information can be edited to keep it up to date

All members details can be changed at any time through the members page by clicking on the pencil icon. They are then greeted with a form which populates with the member's current information. This can then be edited and submitted or cancelled to leave their details unchanged.

Members can be deleted from the family entirely

A member can be deleted by selecting the x icon on the members page. This uses the URL to pass through their ID allowing the confirmation page to show their name.

Plan

The family members that are eating each day can be selected

On the plan page there is an easy to use tab list with a form for each day to tick which members will be available to eat. It also changes the colour of the member's name when they have been selected to give the user a way to quickly check they have selected correctly.

The time available to cook each day can be inputted

On each tab there is a drop down box where the time the user has available can be input.

This data can all be formatted and passed through to the generation script

The inputs on each page are all part of the same form where arrays are used to store the information to be passed through to the PHP plan generation script.



Generation

Recipes are selected at random ensuring that there is enough time available to cook it

My plan generation script uses a random number generator to select a recipe from the database and a boolean flag is used to ensure its cook time is smaller than the user's available time that is passed through from the plan page.

Recipes used in last weeks plan must be discounted to avoid repetition

The same boolean flag is also used to check for repetition. The user's latest plan is fetched and all of the recipes included are compared to the randomly generated recipe. If they match it will not be accepted.

Ingredient measurements calculated individually depending on the number of members eating

The plans themselves are not stored with the calculated values in the database to avoid data repetition. Instead the number of members eating each day is stored and each time the recipe is displayed the calculation is performed. This approach means it is also easier to generate the shopping list.

Calendar

Table containing summary of each days recipes and the date they are to be cooked

I was able to create a simple list of recipes which display the day it will be cooked and the date. It also shows how many people are eating that day by the number of platter icons displayed. The recipe for each day links to the page with the instructions and ingredients.

Ability to overwrite the plan for the coming week with warning that it will delete the current meal plan

In order to handle multiple plans I have implemented a stack system (LIFO) meaning the latest plan is always removed first. The plan generation script checks to see if that start date is the same as the start date for the new plan. If it is it takes the user to the plan deletion page asking them if they want to remove their plan so they can replace it. The only way this could be improved is that once the user has chosen to delete their current plan the information in the plan form is lost meaning if they had not realised they already had a plan they would have to input the details again.



A link for each day to a automatically populated page containing cooking instructions and ingredients

The day recipe page fetches the recipe ID from the URL and checks the database to see how many users are eating to calculate the ingredient quantities. This means it can be reused for any recipe and it has conditional formatting to separate the ingredient names and measurements correctly.

List

Display ingredients needed for each recipe

A complex SQL query is used to fetch all the ingredients included in the user's latest plan and display them in a table. This table changes its formatting on mobile to make it easy to view from mobile devices when in the supermarket.

Show the amount of each ingredient needed, adding values if used in multiple recipes that week

The SQL query also handles summing ingredient quantities used in multiple recipes using the SUM aggregate function and grouping. This was a really successful approach as it limited the amount of PHP I needed to use to perform these calculations.

Categorise ingredients to make it easier to find when shopping

Each ingredient stored in the database is given a category allowing me to separate out my query results into separate arrays to output. This meant each column in the table could be for a different category of ingredients.

Extension Objectives

Improve encryption of user accounts using salting and key stretching algorithms like bcrypt to protect password

As I mentioned above under my core objective I was able to implement the bcrypt hashing algorithm and also escape all user submitted strings to help in securing my users account details.

Allow users to add their own recipes through a form

The admin page enables the user to add recipes directly to my database. It handles all of the formatting and performs a check to see if the ingredients submitted are already used in other recipes. If they are it won't record them again and will use the originals ID. This prevents data



Jack Corbett

from being repeated in my ingredients table and means the same ingredient will not appear separately on the user's shopping list.

Tutorial system modal to teach the user how to make the most of the tool

The tutorial modal can be accessed from the home page which provides screenshots of each page along with a description of how to use it. This will really help new users and it was really important to my end user that it was easy to get started with. There is also more tutorial information on the about page.

Members can be ticked in the plan page or unticked by default to compensate for family members that eat less often

When I first got my end users feedback on my original designs she said it was really important that it was easy to input who will be eating, especially for members that often don't eat with the family. With the default ticked radio buttons added to the family system the user can chose for each member whether they are always included in the plan but can be removed manually or they are always excluded and have to be selected manually.

Separate php scripts using include files

I felt the way in which I separated my code was successful in keeping it organised and avoiding large amounts of repetition by reusing scripts. I followed the exact structure laid out in my design section.

Make the site mobile friendly by changing the display format depending on the size of the users device

All aspects of the site scale to the users device including forms, tables, lists and the navigation bar to ensure the user has full control. This was really important to all the users I questioned so they can use my tool on the go.

User Feedback

Although I have touched on some of the ways in which I listened to my users when evaluating the completion of my objectives, I felt it was important to get my end users feedback on the final product. To do this I gave my Mum a week to test it without any instruction from me on how to use it and at the end of this time sat down with her to collect feedback about the experience and test her by setting tasks on her tablet which were recorded in my testing video.



Overall, her feedback was very positive. The user interface felt really simple and it was easy to pick up how everything worked. The family system was really easy to manage and understand thanks to the helpful icon buttons meaning you didn't need to read small text. I was really pleased about this as it meant the work I put in picking icons and designing logos did contribute to the user experience. She also felt the plan page worked really well to organise all the inputs to make them manageable and that it was really helpful to tick members by default to cater to our schedule. However, not realising that a plan had already been created she did end up losing what she had inputted when asked if she wanted to delete her current plan. This is something I definitely need to improve to avoid frustrating my users. Other than that minor complaint there were no issues from account setup to plan generation.

As someone who has been making weekly meal plans for the last 18 years for a family full of fussy eaters, there was some criticism of the generated plan. This was due in part to the lack of recipes currently in the database, as they were only added for testing purposes. The main issue she highlighted was the fact that no checks were performed on the recipe names themselves meaning you could potentially end up with similar recipes on adjacent days eg: chicken noodles, spanish chicken and chicken burgers or beef stir fry with hoisin sauce, chicken stir fry. This doesn't provide the varied diet I am looking to achieve. The only other issue she highlighted was the fact that the recipe selection algorithm picked from every recipe in the database as it meant some recipes could be selected that nobody would like. That aside she was still impressed that it could scale so easily to different numbers of people eating and that it could find her quick recipes when she was short on time.

The shopping list however, was a real success, she especially liked being able to see it easily on her phone while shopping. The categories made it really easy to find everything and made sure no ingredients were missed. The only thing that could be improved would be to allow the user to add their own items to the list for household items and extras like breakfast/lunch.

I was really pleased with the feedback as I knew it would need to tick a lot of boxes to convince my Mum to ditch the pen and paper and entrust an algorithm with selecting what we eat. Before she would be confident to just press the generate plan button and head into the shop with her digital list I need to add some more decisions to my recipe selection algorithm and give the user control over what recipes can be selected.



System Improvements

I have already highlighted some areas of improvement in my own evaluation of my objectives and my user feedback. There are many things I want to add to my web app although none of it will change the core functionality which I have already built, only adding extra user friendly changes which the time scale of the Computer Science A Level project did not allow with my other commitments.

The first needs to be to save the user's input from the plan form even if they then decide to delete their current plan. To do this all I need to do is either use a hidden form to pass the values back to the generation script once the plan has been deleted or just delete the plan from within the script itself. This will be simple to implement but will avoid frustrating users.

Another change to be made is to give the user access to a list of the recipes in the database allowing them to pick which they want to be included in their recipe selection. This way they can further cater it to their own tastes and needs. I could also improve the recipe selection by performing like queries using wildcards to check a similar recipe hasn't been selected on neighbouring days.

Finally, although not a programming change, when giving a talk about technology involving my web app I was asked if I would support vegetarian and vegan diets. As one of the main reasons people don't drop meat from their diet is the difficulty in finding recipes, this could really help people looking to make the switch and people who are finding it increasingly difficult to find meals that fit in with their diet choices. So this is definitely something I would look to add as it could also be a market opportunity.



Bibliography

Public Health England, "UK And Ireland Prevalence And Trends: Public Health England Obesity Knowledge And Intelligence Team", 2016
<https://www.noo.org.uk/NOO_about_obesity/adult_obesity/UK_prevalence_and_trends> [accessed 13 September 2016]

Sainsbury plc J, "Healthier Family Recipes", 2016
<<https://livewellforless.sainsburys.co.uk/healthier-family-recipes/>> [accessed 18 October 2016]

Corporation, Intel, "Data Center Solutions, IOT, And PC Innovation"
<<http://www.intel.co.uk/content/www/uk/en/homepage.html>> [accessed 3 April 2016]

Inc, Apple, "Apple (United Kingdom)", 2017 <<http://www.apple.com/uk/>> [accessed 3 April 2016]

InnovaDev, "Home - Menu Planner", 2016 <<http://mp2.menu-planner.com/>> [accessed 18 October 2016]

"Meal Planner | Weekly Meal Plans | Tesco Real Food"
<<https://realfood.tesco.com/meal-planner.html>> [accessed 17 October 2016]

"PHP", Codecademy, 2017 <<https://www.codecademy.com/learn/php>> [accessed 8 July 2016]

"PHP Programming Part 1: Introduction To PHP Programming", Youtube, 2017
<https://www.youtube.com/watch?v=27dR_sLaM74&t=891s> [accessed 8 June 2016]

Programme, World Food, "Hunger Statistics" <<https://www.wfp.org/hunger/stats>> [accessed 13 September 2016]

"Stack Overflow", Stackoverflow.Com, 2017 <<http://stackoverflow.com/>> [accessed 7 June 2016]

"Tesla UK", 2017 <https://www.tesla.com/en_GB> [accessed 3 April 2016]

Ullman, Larry E, *PHP And Mysql For Dynamic Web Sites*, 4th edn (Berkeley, Calif.: Peachpit Press, 2012)

"W3.CSS Home", W3schools.com, 2017 <<https://www.w3schools.com/w3css/default.asp>> [accessed 8 August 2016]

WRAP, "About Food Waste" <<http://www.lovefoodhatewaste.com/content/about-food-waste-1>> [accessed 18 October 2016]