



# SOFTWARE DEVELOPMENT PROJECT

DELIVERABLE 6

NOVEMBER 2018

Sydney Twigg - M8C3XRSN8  
Imtiyaaz Samuels - MB2015-0261  
Andrea Cloete - Z46WWQH76  
Justin Bruce - BS36V9PD9

# CONTENTS

TABLE OF FIGURES.....	8
TABLE OF TABLES.....	12
INTRODUCTION .....	16
PROJECT INTRODUCTION.....	17
GROUP AND CUSTOMER INFORMATION.....	17
GROUP INFORMATION.....	17
CUSTOMER INFORMATION.....	18
PROJECT INTRODUCTION .....	18
BACKGROUND .....	18
PURPOSE .....	20
SCOPE .....	21
AIM .....	22
OBJECTIVES .....	22
REQUIREMENTS .....	23
FUNCTIONAL REQUIREMENTS .....	23
NON-FUNCTIONAL REQUIREMENTS .....	26
TECHNICAL REQUIREMENTS .....	27
SCHEDULE.....	28
PROJECT OVERVIEW.....	28

GANTT CHART FOR DELIVERABLE 1 .....	29
RISK MANAGEMENT .....	30
RISK REGISTER .....	32
METHOLODOLOGY AND DEVELOPMENT PLAN.....	36
METHODOLOGY .....	36
ITERATIVE INCREMENTAL DEVELOPMENT .....	36
DEVELOPMENT PLAN .....	40
REQUIREMENT INCREMENTS .....	40
UPDATED SCHEDULE.....	43
GANTT CHART FOR DELIVERABLE 2.....	43
UPDATED RISK REGISTER .....	44
USER REQUIREMENTS DOCUMENT .....	50
INTRODUCTION .....	50
PURPOSE .....	50
SCOPE .....	52
ACRONYMS AND ABBREVIATIONS .....	54
DEFINITIONS .....	55
OVERVIEW .....	55
SYSTEM REQUIREMENTS .....	56
SYSTEM PERSPECTIVE .....	56
FUNCTIONAL REQUIREMENTS .....	56

USE CASE DIAGRAM .....	59
NON-FUNCTIONAL REQUIREMENTS .....	61
TECHNICAL REQUIREMENTS .....	62
USER CHARACTERISTICS .....	63
OPERATIONAL ENVIRONMENT .....	64
INFORMATION SYSTEMS DESIGN .....	65
LOGICAL DESIGN .....	65
USE CASE DIAGRAMS.....	65
CONTEXT DIAGRAM .....	85
SYSTEMS LEVEL DFD .....	86
PRIMITIVE LEVEL DFD.....	87
ENTITY RELATIONSHIP DIAGRAM.....	88
PHYSICAL DESIGN .....	90
INVESTIGATION OF TECHNOLOGIES TO BE APPLIED .....	90
SYSTEM TESTING.....	95
UPDATED SCHEDULE.....	112
GANTT CHART FOR DELIVERABLE 3.....	112
RISK REGISTER .....	114
SYSTEM INTERFACE DESIGN .....	118
INCREMENT ONE: MOCKUP DESIGNS.....	120
LANDING PAGE .....	120

LOGIN PAGE.....	121
REGISTRATION PAGE .....	122
CLIENT INTERFACE.....	123
DIETICIAN INTERFACE.....	132
INCREMENT TWO: PROTOTYPE DESIGNS .....	134
LANDING PAGE.....	134
HOME PAGE .....	135
MEAL PLAN OVERVIEW.....	136
MEAL PLAN DAY VIEW .....	137
EMAIL MEAL PLAN.....	138
SHOPPING LIST .....	139
VIEW PROFILE .....	140
CONTACT PAGE.....	141
INCREMENT 3: FINAL DESIGN.....	142
LANDING PAGE.....	142
TESTING RESULTS.....	159
TEST RESULTS: INCREMENT TWO: PROTOTYPE DESIGNS .....	159
TEST PLAN .....	159
TEST TEMPLATE.....	161
TEST RESULTS: INCREMENT THREE: FINAL SYSTEM .....	170
TEST PLAN .....	170

TEST TEMPLATE.....	173
SCHEDULE .....	182
GANTT CHART FOR DELIVERABLE 4 .....	182
RISK REGISTER.....	183
USER MANUAL.....	190
INTRODUCTION .....	190
GETTING STARTED .....	191
HOW TO INSTALL EATFORLIFE .....	191
INSTALLING LOCALLY .....	191
ACCEESSING THE LANDING PAGE.....	192
LOGIN TO THE APPLICATION.....	192
REGISTER A NEW ACCOUNT.....	193
NAVIGATING THE APPLICATION .....	194
VIEW THE HOME PAGE.....	194
INTERACT WITH YOUR PERSONAL PROFILE.....	195
VIEW MEAL PLAN OVERVIEW.....	196
VIEW DAILY MEAL PLAN .....	197
EMAIL MEAL PLAN.....	198
PRINT MEAL PLAN.....	198
VIEW SHOPPING LIST .....	199
SEND A MESSAGE TO USERS .....	200

VIEW A BROADCAST MESSAGE .....	200
EVALUATION REPORT .....	201
INTRODUCTION .....	201
THE FINAL SYSTEM AND THE CUSTOMER'S REQUIREMENTS .....	201
THE METHODOLOGY (ITERATIVE INCREMENTAL DEVELOPMENT).....	204
THE CUSTOMER.....	205
GROUP DYNAMICS AND TEAM COLLABORATION .....	205
TIME MANAGEMENT .....	206
LESSONS LEARNED.....	206
JUSTIN BRUCE.....	206
SYDNEY TWIGG.....	207
ANDREA CLOETE.....	207
IMTIYAAZ SAMUELS .....	208
CONCLUSION .....	209
CUSTOMER SIGN-OFF .....	210
REFERENCES .....	211

## TABLE OF FIGURES

Figure 1 - Screenshot's showing the mobile view of the EatForLife system. Screenshots taken on an iPhone X using the Google Chrome web browser.....	20
Figure 2 - Project Timeline for EatForLife.....	28
Figure 3 - Timeline for Deliverable 1 .....	29
Figure 4 - The Agile Development Process. Source: (Rouse, 2017).....	36
Figure 5 -The Scrum process. Source (Scrum Alliance, 2018).....	38
Figure 6 - The Scrum Process. Source: (Thompson, 2012).....	39
Figure 7 - Timeline for Deliverable 2 .....	43
Figure 8 - Screenshot's showing the mobile view of the EatForLife system. Screenshots taken on an iPhone X using the Google Chrome web browser.....	51
Figure 9 - Use Case Diagram for the EatForLife Mobile Application.....	59
Figure 10 - Use Case Diagram for the EatForLife System.....	65
Figure 11 - Use Case Diagram for the EatForLife System.....	66
Figure 12 - Use Case Diagram for the registration event on the EatForLife Mobile Application.	67
Figure 13 - Use Case Diagram for the login event on the EatForLife Mobile Application....	68
Figure 14 - Use Case Diagram for the login event on the EatForLife Mobile Application ....	69
Figure 15 - Use Case Diagram for the view meal plan event on the EatForLife Mobile Application .....	70
Figure 16 - Use Case Diagram for the edit diet preferences event on the EatForLife Mobile Application .....	71

Figure 17 - Use Case Diagram for the change diet item event on the EatForLife Mobile Application .....	72
Figure 18 - Use Case Diagram for the mark item as eaten event on the EatForLife Mobile Application .....	73
Figure 19 - Use Case Diagram for the download or save meal plan event on the EatForLife Mobile Application .....	74
Figure 20 - Use Case Diagram for the share meal plan event on the EatForLife Mobile Application .....	75
Figure 21 - Use Case Diagram for the view personal information event on the EatForLife Mobile Application .....	76
Figure 22 - Use Case Diagram for change personal information event on the EatForLife Mobile Application .....	77
Figure 23 - Use Case Diagram for the view BMI progress event on the EatForLife Mobile Application .....	78
Figure 24 - Use Case Diagram for the view shopping list event on the EatForLife Mobile Application .....	79
Figure 25 - Use Case Diagram for the download or save shopping list event on the EatForLife Mobile Application .....	80
Figure 26 - Use Case Diagram for the share shopping list event on the EatForLife Mobile Application .....	81
Figure 27 - Use Case Diagram for the communicate with dietician event on the EatForLife Mobile Application .....	82
Figure 28 - Use Case Diagram for the reset password event on the EatForLife Mobile Application .....	83
Figure 29 - Context Diagram for the EatForLife system.....	85
Figure 30 - Systems Level Data Flow Diagram for the EatForLife System. ....	86

Figure 31 - Primitive Level Data Flow Diagram for the EatForLife System. ....	87
Figure 32 - Model of the software testing process. Source: (Sommerville, 2011). ....	95
Figure 33 - White, black, and grey box-based strategies and inputs. Source (Saleh, 2009).	96
Figure 34 - Summarised Gantt Chart for the EatForLife Project. ....	112
Figure 35 - Detailed Gantt Chart for Deliverable 3.....	113
Figure 36 - User Interface Design Process. Source: (Pressman & Maxim, 2015).....	118
Figure 37 - Mock-up Design for the EatForLife Application Landing Page.....	120
Figure 38- Mock-up Design for the EatForLife Application Login Page.....	121
Figure 39- Mock-up Design for the EatForLife Application Registration Page. ....	122
Figure 40- Mock-up Design for the EatForLife Application Home Page.....	123
Figure 41- Mock-up Design for the EatForLife Application Meal Plan Overview Page.....	124
Figure 42- Mock-up Design for the EatForLife Application Meal Plan Day View Page.....	125
Figure 43 - Mock-up Design for the EatForLife Application Shopping List Page.....	126
Figure 44 - Mock-up Design for the EatForLife Application Profile Page. ....	127
Figure 45 - Mock-up Design for the EatForLife Application Edit Profile Page. ....	128
Figure 46 - Mock-up Design for the EatForLife Application Statistics Page. ....	129
Figure 47 - Mock-up Design for the EatForLife Application Measurement Logs Page.....	130
Figure 48 - Mock-up Design for the EatForLife Application Contact Page. ....	131
Figure 49 - Mock-up Design for the EatForLife Application Client List Page.....	132
Figure 50 - Mock-up Design for the EatForLife Application Client Details Page.....	133
Figure 51 - Prototype of the EatForLife Application Landing Page. ....	134

Figure 52 - Prototype of the EatForLife Application Home Page.....	135
Figure 53 - Prototype of the EatForLife Application Meal Overview Page.....	136
Figure 54 - Prototype of the EatForLife Application Meal Day View Page.....	137
Figure 55 - Prototype of the EatForLife Application Email Meal Plan Page.....	138
Figure 56 - Prototype of the EatForLife Application Shopping List Page.....	139
Figure 57 - Prototype of the EatForLife Application Profile Page.....	140
Figure 58- Prototype of the EatForLife Application Contact Us Page.....	141
Figure 59 - Final Design for the EatForLife Mobile Application.....	142
Figure 60 - Gantt Chart for Deliverable 4.....	182
Figure 61 - EatForLife landing page.....	192
Figure 62 - Registration page.....	193
Figure 63 - Navigation bar.....	194
Figure 64 - EatForLife home page.....	194
Figure 65 - The Profile screen displays your Profile picture, designated Dietician and your personal details.....	195
Figure 66 - Upload a new profile picture.....	195
Figure 67 - Meal Plan Overview page.....	196
Figure 68 - Meal Plan Day View Page.....	197
Figure 69 - Email Meal Plan Page.....	198
Figure 70 - Food List Page.....	199
Figure 71 - Messages Page.....	200



## TABLE OF TABLES

Table 1 - Group Information .....	17
Table 2 - Customer Information.....	18
Table 3 - Functional Requirements for the EatForLife project.....	23
Table 4 - Non-Functional Requirements for the EatForLife project.....	26
Table 5 - Technical Requirements for the EatForLife project.....	27
Table 6 - Risk Register.....	32
Table 7 - Requirement increments .....	40
Table 8 - Updated Risk Register .....	44
Table 9 - Acronyms and abbreviations .....	54
Table 10 - Definitions used in the User Requirements Document .....	55
Table 11 - Functional Requirements for the EatForLife Mobile Application.....	56
Table 12 - Non-Functional Requirements for the EatForLife Mobile Application.....	61
Table 13 - Technical Requirements for the EatForLife Mobile Application.....	62
Table 14 - Users involved in the EatForLife system .....	63
Table 15 - Operational environment .....	64
Table 16: Categories of Test Types. Source: (Valacich, et al., 2015). .....	96
Table 17 - Tests conducted on an information system. Source: (Stair & Reynolds, 2016)...	98
Table 18 - Test Plan for the EatForLife System.....	99

Table 19 - Test Template for the EatForLife Home Page .....	103
Table 20 - Test Template for the EatForLife Home Page .....	103
Table 21 - Test Template for the EatForLife Registration Page.....	104
Table 22 - Test Template for the EatForLife Registration Page.....	104
Table 23 - Test Template for the EatForLife Login Page .....	105
Table 24 - Test Template for the EatForLife Login Page .....	105
Table 25 - Test Template for the EatForLife Home Page .....	106
Table 26 - Test Template for the EatForLife Home Page .....	106
Table 27 - Test Template for the EatForLife Meal Plan Page .....	107
Table 28 - Test Template for the EatForLife Meal Plan Page .....	107
Table 29 - Test Template for the EatForLife Shopping List Page .....	108
Table 30 - Test Template for the EatForLife Shopping List Page .....	108
Table 31 - Test Template for the EatForLife Profile Page.....	109
Table 32 - Test Template for the EatForLife Profile Page.....	109
Table 33 - Test Template for the EatForLife Communication Page .....	110
Table 34 - Test Template for the EatForLife Communication Page .....	110
Table 35 - Test Template for the EatForLife Admin Page.....	111
Table 36 - Test Template for the EatForLife Admin Page.....	111
Table 37 - Risk Register for the EatForLife application development. ....	114
Table 38 - User Interface Screens for the EatForLife Mobile Application.....	119
Table 39 - Test Plan for the EatForLife System Development, Increment Two.....	159

Table 40 - Test Template for the EatForLife Home Page .....	161
Table 41 - Test Template for the EatForLife Home Page .....	161
Table 42 - Test Template for the EatForLife Registration Page.....	162
Table 43 - Test Template for the EatForLife Registration Page.....	162
Table 44 - Test Template for the EatForLife Login Page .....	163
Table 45 - Test Template for the EatForLife Login Page .....	163
Table 46 - Test Template for the EatForLife Home Page .....	164
Table 47 - Test Template for the EatForLife Home Page .....	164
Table 48 - Test Template for the EatForLife Meal Plan Page .....	165
Table 49 - Test Template for the EatForLife Meal Plan Page .....	165
Table 50 - Test Template for the EatForLife Shopping List Page .....	166
Table 51 - Test Template for the EatForLife Shopping List Page .....	166
Table 52 - Test Template for the EatForLife Profile Page.....	167
Table 53 - Test Template for the EatForLife Profile Page.....	167
Table 54 - Test Template for the EatForLife Communication Page .....	168
Table 55 - Test Template for the EatForLife Communication Page .....	168
Table 56 - Test Template for the EatForLife Admin Page.....	169
Table 57 - Test Template for the EatForLife Admin Page.....	169
Table 58 - Test Plan for the EatForLife System Development, Increment Two.....	170
Table 59 - Test Template for the EatForLife Home Page .....	173
Table 60 - Test Template for the EatForLife Home Page .....	173

Table 61 - Test Template for the EatForLife Registration Page .....	174
Table 62 - Test Template for the EatForLife Registration Page .....	174
Table 63 - Test Template for the EatForLife Login Page .....	175
Table 64 - Test Template for the EatForLife Login Page .....	175
Table 65 - Test Template for the EatForLife Home Page .....	176
Table 66 - Test Template for the EatForLife Home Page .....	176
Table 67 - Test Template for the EatForLife Meal Plan Page .....	177
Table 68 - Test Template for the EatForLife Meal Plan Page .....	177
Table 69 - Test Template for the EatForLife Shopping List Page .....	178
Table 70 - Test Template for the EatForLife Shopping List Page .....	178
Table 71 - Test Template for the EatForLife Profile Page.....	179
Table 72 - Test Template for the EatForLife Profile Page.....	179
Table 73 - Test Template for the EatForLife Communication Page .....	180
Table 74 - Test Template for the EatForLife Communication Page .....	180
Table 75 - Test Template for the EatForLife Admin Page.....	181
Table 76 - Test Template for the EatForLife Admin Page.....	181
Table 77 - Risk Register for the EatForLife application development. ....	183

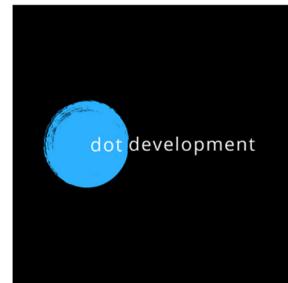
## INTRODUCTION

# PROJECT INTRODUCTION

## GROUP AND CUSTOMER INFORMATION

### GROUP INFORMATION

Dot Development was formed by CTI students Sydney Twigg, Imtiyaaz Samuels, Andrea Cloete & Justin Bruce as part of the 3<sup>rd</sup> year Software Development Project.



The table below provides more information on each team member and their skills:

*Table 1 - Group Information*

Student Number	Member Name	Skills
M8C3XRSN8	Sydney Twigg	Documentation, Java, HTML, JavaScript, CSS, SQL, Design, Management, Testing
MB2015-0261	Imtiyaaz Samuels	Microsoft Office, Java, HTML, SQL, VB, GML, design, research
Z46WWQH76	Andrea Cloete	Microsoft Office, Java, HTML, SQL, Documentation, Tester, Design
BS36V9PD9	Justin Bruce	Microsoft Office, Java, HTML, SQL, Documentation, Tester, Presenter

## CUSTOMER INFORMATION

As per our project scope, our team approached various clients with system proposals - ultimately choosing to develop a mobile application for EatForLife, a company that provides its clients with a nutrition programme specialising in changing people's lifestyle and eating habits, developed by dieticians (FutureLife, 2016).

The table below provides more information about our client:

*Table 2 - Customer Information*

Company Name	EatForLife
Owner Name	Matthew Grossett
Owner Contact Number	082 559 8151
Owner Email Address	matthew@rwfl.co.za
Industry	Dietetics

## PROJECT INTRODUCTION

### BACKGROUND

EatForLife was started by Matthew Grossett - the CEO of *Run/Walk for Life South Africa*, and is a subsidiary of *Run/Walk For Life*.

EatForLife is a nutrition programme developed by registered dieticians that focuses on making a change to people's lifestyle and eating habits (FutureLife, 2016). EatForLife is designed and administered by qualified dieticians; the meal plans are suited for men, woman, children, teenagers, and those suffering from high blood pressure, type 2 diabetes and cholesterol (FutureLife, 2016) (EatForLife, 2018).

The EatForLife programme is based on balanced nutrition and personal goal setting - EatForLife clients will set personal weight loss goals with the aid of the programme dieticians, and they will be provided with a customised eating plan based on the client's goals and health status (EatForLife, 2018). Progress towards the set goals is monitored, and any health risks for clients are screened and referred to a doctor or specialist.

The EatForLife programme is a web-based programme, users can log into the website through an online portal - once logged in users can view and edit their details; their weight loss chart; a 7-day eating plan and a shopping list for those 7 days; view and update their measurements and weight loss logs; and users may contact their dietitian with any queries through this site.

The following excerpt from the EatForLife website shows the services that are offered:

---

***List of services:***

- *Your risk status will be assessed and meal plans will be devised accordingly*
- *Meal plans will be devised according to individual requirements*
- *Meal plans can be tailor made according to members taste preferences*
- *Meal plans cater for vegetarians, members who are lactose or wheat intolerant*
- *Type 2 Diabetics, members with high cholesterol or high blood pressure receive suitable meal plans for their medical conditions*
- *The programme is suitable for people wanting to lose weight, gain weight or maintain their weight, as well as for increased energy levels.*
- *Suitable goal weights will be worked out*
- *Your BMI will be calculated*
- *Your progress will be monitored*
- *You will receive weekly nutrition education relevant to your weight loss journey*
- *You will be taught how to make healthy food choices and stay motivated and focused on your journey*
- *The scientific basis of the programme ensures that you lose weight in a healthy, sustainable way while receiving the optimum benefit from your diet.*

---

*Excerpt 1 - Services Offered by EatForLife. Source: (EatForLife, 2018).*

## PURPOSE

The main purpose of the proposed EatForLife mobile application is to enhance the current system and user experience.

The current EatForLife system is only web-based, and while users can log in to the site through their mobile web browsers the website is not optimised or ideal for mobile devices.

The following screenshots provide a view of the current mobile view on the EatForLife website:

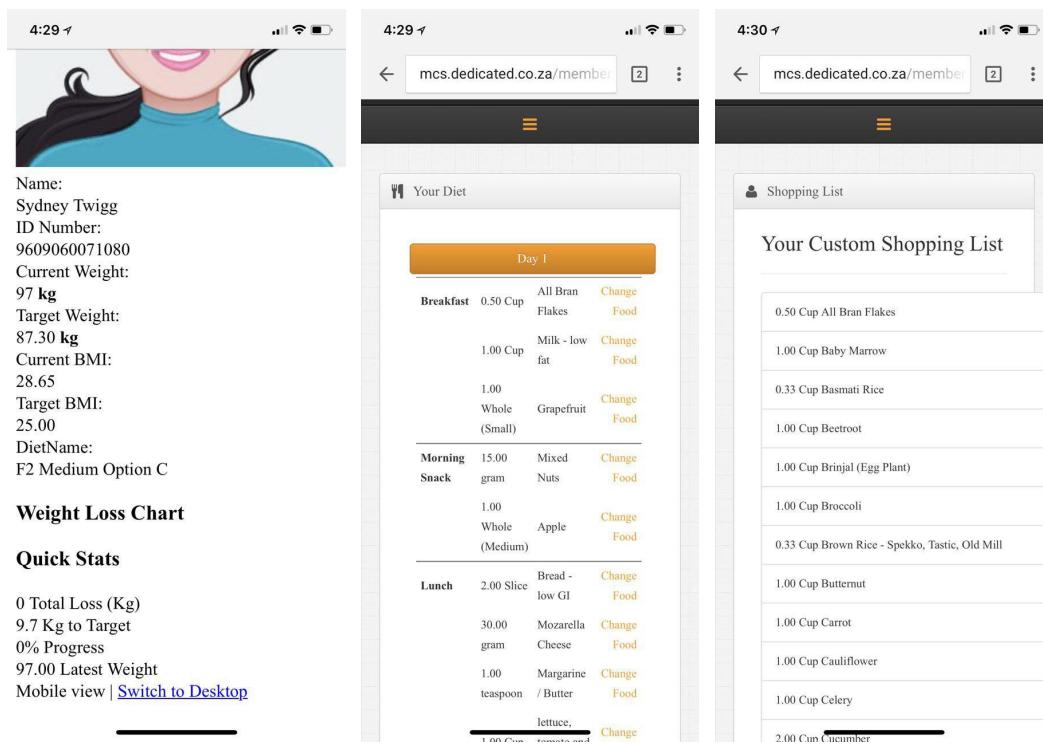


Figure 1 - Screenshot's showing the mobile view of the EatForLife system. Screenshots taken on an iPhone X using the Google Chrome web browser.

As shown above, the user experience is not ideal for a mobile device. A website acts as a form of marketing for a company, and many people would be put off by an inaccessible or hard to read the website - especially in the current times where most people access the internet via smartphones or tablets. South Africa has 21 million internet users - with a majority using mobile devices to access the internet (Shapshak, 2017).

In order to expand the current customer base and make the eating programme more accessible and easier to use for the South African market EatForLife requires a mobile application.

A mobile application provides many benefits to a business, such as the following outlined by (Haselmayr, 2018):

- Be visible to the customer at all times;
- Create a direct marketing channel;
- Build your brand and recognition;
- Improved customer engagement; and
- Stand out from the competition.

Thus, the purpose of this project is to provide a mobile application that is fully functional, easily accessible, user friendly, cross-platform, and well designed for EatForLife customers to access their personalised eating programmes.

## SCOPE

The scope of the system includes engagement and communication with EatForLife, integrating the existing system into the new system and communication over systems requirements; understanding and analysing the current system process - as well as making suggestions on how EatForLife can change current methodologies to enhance the user experience.

The EatForLife application will be made as a native mobile application as opposed to a mobile web application. According to (Pressman & Maxim, 2015) a mobile application is distinguished from a mobile web application as an application that can gain direct access to the hardware characteristics of a device - such as the accelerometer as well as access the internal storage and processing capabilities of the mobile device; while a mobile web application allows a mobile device to view web-based content on a browser that emulates a mobile application interface.

## AIM

The aim of the EatForLife mobile application project is to design and implement a cross-platform mobile application that will provide an easy way for members to access and manage their personalised eating plan.

## OBJECTIVES

The objectives of the project are:

- To develop a mobile application that allows EatForLife members to use on both iOS and Android devices.
- To ensure the system is developed according to the schedule, and that all requirements are met in the time frame.
- To design a fully functional, thoroughly tested, user-friendly system, that meets Nielson's Heuristics for User Interface Design (Nielson, 1995) and meets all of the client's requirements.

All of the above objectives should be complete by the October 2018, when the project is due.

## REQUIREMENTS

The following table contains the high-level requirements for the EatForLife project:

### FUNCTIONAL REQUIREMENTS

The table below represents the Function Requirements for the EatForLife project according to (EatForLife, 2018), (Somerville, 2011), and (Schwalbe, 2015):

*Table 3 - Functional Requirements for the EatForLife project.*

ID	Requirement	Priority
<b>Registration</b>		
FUR-1	The system allows users to register.	High
<b>Login</b>		
FUR-2	The system allows users to login.	High
<b>Data Input</b>		
FUR-3	The system allows a user to input their details.	Medium
<b>Data Update</b>		
FUR-4	The system allows a user to edit their details.	Medium
<b>Communication</b>		
FUR-5	The system allows users to send queries to and communicate with dietitians.	Medium
<b>Diet Plans</b>		
FUR-6	The system allows users to get customised diet plans based on specific user parameters.	Medium
FUR-7	The system provides information about diet plans to users.	Medium
FUR-8	The system provides information about each item in the eating plans to users.	High
FUR-9	The system allows users to have their health statuses analysed and a meal plan can be created accordingly.	High

Diet plan items awareness		
FUR-10	The system allows users to find alternative items to the substitute foods for in the diet plan.	High
FUR-11	The system provides users with a checklist to check off food once eaten on the diet plan.	High
Health checks		
FUR-12	The system can calculate the BMI for users.	Medium
FUR-13	The system can calculate the correct BMI for users based on user inputted data.	Medium
FUR-14	The system allows users to monitor their progress with reports and graphs.	Medium
FUR-15	The system allows users to monitor their measurement and weight history with reports and graphs.	Medium
Goals		
FUR-16	The system will allow the user to set weight loss goals.	Low
Downloads		
FUR-17	The system will allow users to download their customized 7-day meal plan directly through the mobile application.	High
FUR-18	The system will allow users to have their 7-day meal plan emailed to their provided email addresses.	High
Downloads		
FUR-19	The system will allow the user to download their 7-day meal plan shopping list directly through the mobile application.	High
FUR-20	The system will allow the user to have their 7-day meal plan shopping list directly through the mobile application.	High
Other		
FUR-21	The system allows the user to reset their password	High
FUR-22	The system allows the user to assign a profile picture to their accounts	Low
FUR-23	The system allows the user and dietitians to log out.	High

Admin Panel		
<b>FUR-24</b>	The system allows the dieticians to login to the application as an admin user	High
<b>FUR-25</b>	The system will direct the dietician to the admin panel on login	Medium
<b>FUR-26</b>	The system will allow the dietician to view a list of all registered clients and their details.	Medium
<b>FUR-27</b>	The system will allow the dietician to view a detailed view of each registered client.	Medium
<b>FUR-28</b>	The system allows dietitians to monitor the progress of users through the admin panel.	Medium
<b>FUR-29</b>	The system will allow an admin user to add another admin user through the admin page.	Low

## NON-FUNCTIONAL REQUIREMENTS

The table below contains the Non-Functional Requirements for the EatForLife project according to (EatForLife, 2018; Sommerville, 2011; Schwalbe, 2015).

*Table 4 - Non-Functional Requirements for the EatForLife project.*

Availability		
ID	Requirement	Priority
NFUR-1	The system must allow the user to have access to it at all times.	High
NFUR-2	The system must allow dietitians to have access to it at all times.	High
Security		
NFUR-3	The system should only allow the user access to the system if the password and username is correct	Medium
NFUR-4	The system should only allow the user dietitians to the system if the password and username is correct	Medium
Validation		
NFUR-5	The system should validate the data entry of the users.	Low
NFUR-6	The system should validate the data entry of the dietitians.	Low
Usability		
NFUR-7	The system should be easy to use.	Medium
Interface		
NFUR-8	The colour scheme should be consistent throughout the system.	Medium
NFUR-9	The styles should be consistent throughout the system.	Medium
NFUR-10	The themes should be consistent throughout the system.	Medium

## TECHNICAL REQUIREMENTS

The table below contains the Technical Requirements for the EatForLife project according to (Sommerville, 2011; Dave, 2017).

*Table 5 - Technical Requirements for the EatForLife project.*

Developer's Hardware		
ID	Requirement	Priority
THR-1	Laptop capable of running Android Studio and/or XCode if on an Apple Laptop.	High
THR-2	PC capable of running Android Studio and/or XCode if on an Apple Mac.	High
THR-3	Mobile device running Android OS.	High
THR-4	Mobile device running iOS.	High
Client Hardware		
THR-5	Mobile device running Android OS.	High
THR-6	Mobile device running iOS.	High
THR-7	PC and/or laptop capable of running emulation software.	High
Developer's Software		
THR-8	The system must be developed for Android and iOS.	High
THR-9	The system must be developed with JavaScript and CSS.	High
THR-10	The system must make use of a SQLite database.	High
THR-11	The system will be designed with NativeScript.	High
THR-12	The developers must have internet access.	High
Client Software		
THR-13	The user must have an Android or iOS device.	High
THR-14	The user must have internet access.	High

## SCHEDULE

### PROJECT OVERVIEW

The figure below illustrates the Gantt chart for the entirety of the EatForLife mobile application project:

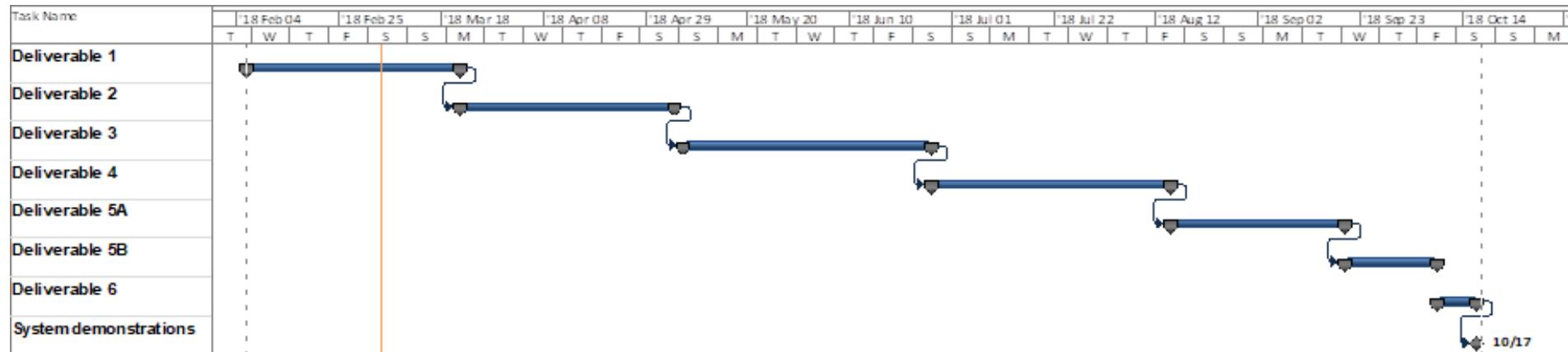


Figure 2 - Project Timeline for EatForLife

From Figure 2 - Project Timeline for EatForLife above we can see that the EatForLife project consists of seven deliverables and a system presentation, these deliverables start on Tuesday 6 February 2018 and are due to be finished by Wednesday 17 October 2018.

## GANTT CHART FOR DELIVERABLE 1

The figure below illustrates the Gantt chart for the first deliverable for the EatForLife mobile application project:

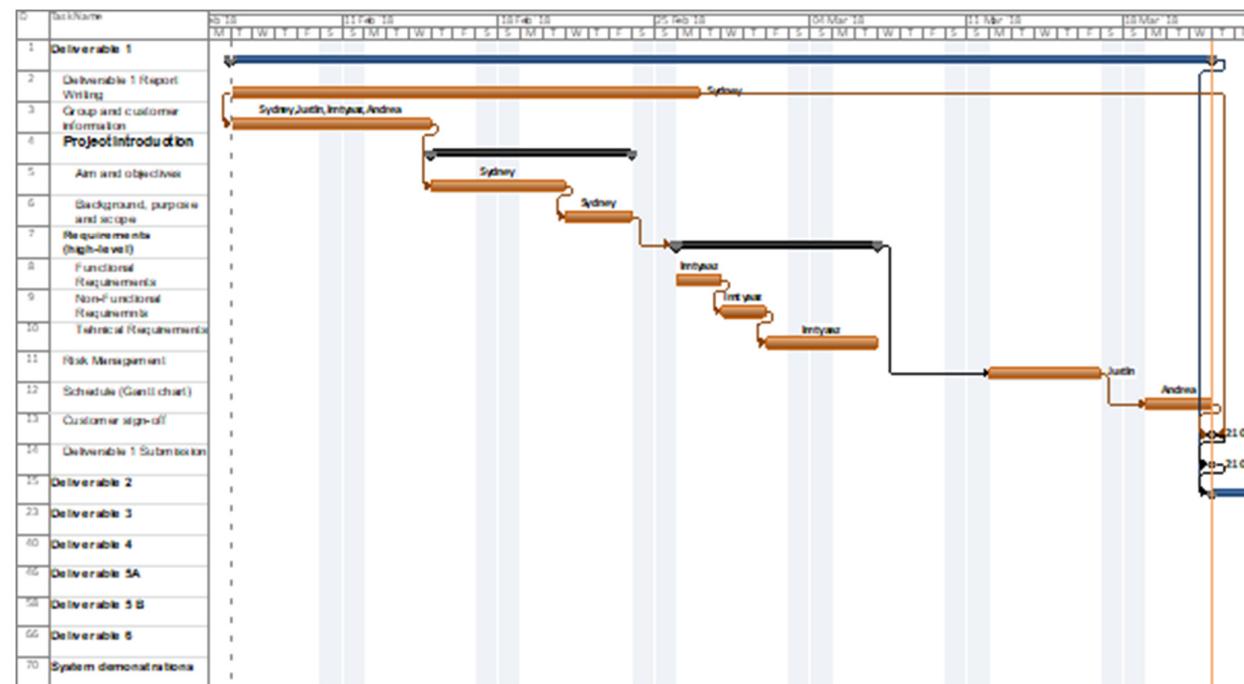


Figure 3 - Timeline for Deliverable 1

Figure 3 above - Timeline for Deliverable 1 shows the content planning for the first deliverable of the project; the deliverable dates, as well as resource names, can be seen in this figure.

## RISK MANAGEMENT

In order to give an introduction to risk management, one must know what a “risk” is. According to (WebFinance Inc, 2018), a risk is the presence of a probability of a negative occurrence that may be caused by external or internal vulnerabilities. These vulnerabilities may be acted upon through any pre-emptive actions. These pre-emptive actions may consist of risk management.

Risk management is the foretelling and evaluation of any possible risks that may impact the development of a system or a company’s business. This also includes the identification of how to avoid or minimize the impact of such incidents or disasters with the use of procedures (Rouse, 2016).

There are five steps to the risk management process, which are as follows (Kloosterman, 2016):

### **1. Identify the risk:**

In this stage, the development team discovers and describes what risks may affect the development of the system or project. By manipulating these risks, a Risk Register may be initiated.

### **2. Analyse the risk:**

After identifying all the risks, the development team determines the likelihood and affect (consequence if the risk were to occur) of each risk. This information is added to the Risk Register.

### **3. Evaluate or rank the risk:**

To evaluate or rank a risk the development team determines the risk magnitude. The risk magnitude consists of the combination of the likelihood (probability) and consequence (impact) of the risk. This list of ranking is added to the Risk Register.

### **4. Treat the risk:**

Treating the risk is known as responding to the risk, i.e. risk response planning. Risk response planning contains risk mitigation strategies, contingency plans, and preventive plans. The highest-ranking risks treatment is added to the Risk Register.

## **5. Monitor and review the risk:**

Monitoring and reviewing risk is to simply keep track of, review and update the risks in the risk register.

The risk register for the EatForLife system development is shown on the following table:

## RISK REGISTER

*Table 6 - Risk Register*

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R06	1	Cross-over incompatibility	The application may work well on Android but not on iOS	Technology Risk	The chosen coding platform does not support the needed language conversion	The chosen platform is incorrect for development of both Android and iOS	Focus on the functional platform to deliver the best full working project, the other platform can be integrated further down the line.	DOT Development	Medium	High
R02	2	Delay of required client information	The client takes a long time to respond to the group's questions via email or does not answer calls consistently	People Risk	The client does not regularly read their email, is too busy to reply or answer the phone or is impeded by another variable	The client isn't as invested as they should be in developing their system	Try complete information needed through other research methods - online etc.	DOT Development	Medium	Medium
R03	3	Uncertainty of full client requirements	The client gives the team a vague list of what he/she wants the system to be and what it should entail	People Risk	The team does not aggressively ask for the requirements	Identified during meetings with the client	Send the client a system proposal with requirement ideas for their feedback	DOT Development	Medium	Medium

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R11	4	Update of requirements	The client decides to change requirements for the system mid-development	People Risk	The client no longer wants to have specific features in the system or wants to add additional features to the system	The client's business develops new requirements for their future success	The team will ensure that a meeting takes place in order to document the new requirements appropriately	DOT Development	Medium	Medium
R04	5	Bad communication skills among team members	Team members do not communicate appropriately and/or frequently enough	People Risk	Due to team members not owning a phone, not using discord (our team's communication application) or google drive	Team members don't know how to work well with each other	Discuss it with the person, if there is no improvement escalate the issue with your lecturer.	DOT Development	Low	High
R05	6	The resignation of a team member	Team member decides to resign mid-way during the development process	People Risk	An unexpected turn of events or too much stress for the team member to handle	The team member can no longer take the pressure of developing the system	The team will have to accept the decision for the resignation of the member and continue development	DOT Development	Low	High
R01	7	Loss of business	The client decides to no longer work with our team	Market Risk	The client has lost interest in the system or no longer wants to invest the time	The client's business no longer needs a new system to further their success	Find a new client, or if too late, continue without the client	Client	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R07	8	Fail to choose a suitable coding language	The team does not choose a suitable coding language the is required for the chosen/designed system	People Risk	Team members may be unsure of which language(s) best suit their skills and the requirements for development	There are multiple languages to choose from and team members have their own unique skills in their respective languages	Use a familiar well documented language	DOT Development	Low	High
R12	9	Hardware failure	The developers have an unforeseen hardware failure, resulting in critical data loss.	Technology Risk	Poor system maintenance, faulty computer components, untrustworthy software installed on the system	Team members did not ensure their hardware was functional and that critical data was backed up	Ensure that all critical data is backed up through Google Drive, GitHub, and other cloud storage, or offsite storage.	DOT Development	Low	High
R10	10	Quality of the system is not satisfactory for the client	The client decides that they do not like the developed system and no longer want to use it	Market Risk	The developed system is not up to the client's standards nor meets their expectations	The developed system is invalid for the use of the client	As a group ensures that all requirements are met. If the client does not like a system that meets the requirements, terminate the project.	Client	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R08	11	Chosen IDE is inappropriate for development	The chosen IDE may not be compatible with required plugins or not contain much needed tools for development	Technology Risk	Team members do not know of various IDE's that may be suitable for the development of the system	The chosen IDE does not support the needed tools for the input of needed plugins or modules	Find a new IDE that suits the project	DOT Development	Low	Medium
R09	12	Poor time management	Increments are put together late	People Risk	Due to team members delaying input due to incompetence or personal issues	Team members do not know how to schedule their time well enough to get work done	Discuss with the individual why their work is not timely, assist them if needed to finish the deliverables in the schedule. If the team members do not improve their time management, escalate it to the lecturer.	DOT Development	Low	Low

# METHODOLOGY AND DEVELOPMENT PLAN

## METHODOLOGY

### ITERATIVE INCREMENTAL DEVELOPMENT

Agile software development is a methodology that is based on iterative incremental development - within an Agile approach requirements and solutions to the requirements can evolve over time through the collaboration of self-organising, multifunctional teams (Thompson, 2012). Agile is based on the mentality of a disciplined project management process whereby the work is done is frequently reviewed and changed to fit the outcome of the review process (Thompson, 2012) Agile development is based on the principles of teamwork, self-organisation and accountability, rapid delivery of high quality software, and a business approach that aligns the customer needs, company goals with the development process (Thompson, 2012). Agile methodology is aimed at small teams working interdependently towards a common goal and allows for flexibility in the delivery of a project, focusing on breaking code into small pieces, testing the code often, and delivering prototypes of the system as often as possible (Rouse, 2017).

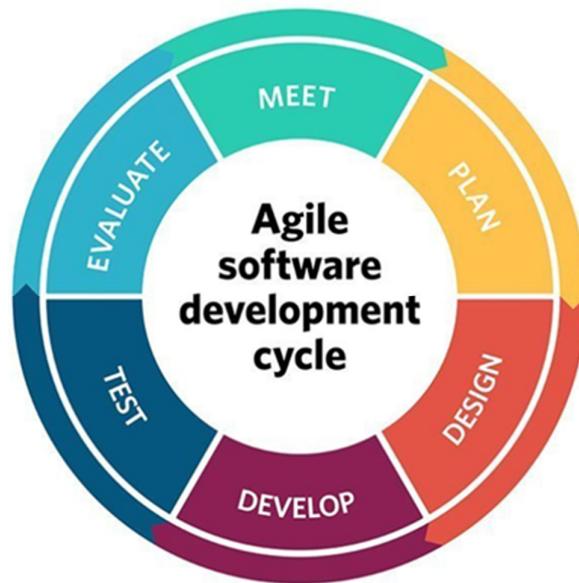


Figure 4 - The Agile Development Process. Source: (Rouse, 2017)

Scrum is an Agile framework that is a collection of practices and concepts to build process; Scrum involves very few elements, there it is a lightweight framework to use (Thompson, 2012). These elements include three roles - Scrum Master, Product Owner, and Team - three meetings - Daily Scrum, Sprint Planning, and Retrospective - and three artefacts - Product Backlog, Sprint Backlog, and Burndown chart (Thompson, 2012). Scrum is based on iterative, cyclical development - where processes are repeated every few weeks until the product backlog is complete - processes can overlap in Scrum, where the Team and Scrum Master can start planning the next Sprint while they are still busy with the current one (Thompson, 2012).

The Scrum process is as follows:

1. The product owner creates a product backlog, a prioritised list of requirements to be implemented in the system;
2. Sprint planning takes place, wherein the team takes a requirement from the product backlog list and decides how to implement it into the system;
3. The team sets an amount of time to complete the chosen requirement - often 2-4 weeks - and have daily meetings to assess the progress towards finishing the requirement;
4. The Scrum master - who is the appointed project manager - keeps the team focused on completing the project and reaching their goals;
5. At the end of each sprint the product should be functional and ready to hand over a functional prototype to the customer;
6. The sprint ends with a review process and a team retrospective - a meeting where the whole team can reflect on the sprint;
7. The next sprint begins with the team repeating the process, picking a requirement from the product backlog and begin working on it (Scrum Alliance, 2018).

The cycle is continued until the entire product backlog has been cleared, or when budget and deadlines limit the development team (Scrum Alliance, 2018). The requirements should be implemented in a rank order of highest to lowest, one at a time (Thompson, 2012).

The benefits of Scrum are that it increases team productivity and reduces development time in comparison to the Waterfall model. Scrum allows the development team to easily adjust the project in the event of rapidly changing requirements and develop a project that fits to evolve with a business's goals (Thompson, 2012). Scrum helps a business to produce high quality deliverables, expect and manage changing requirements, provide accurate estimates for deliverables, and to have more control over the project schedule (Thompson, 2012).

Figure 2 below illustrates the scrum process:

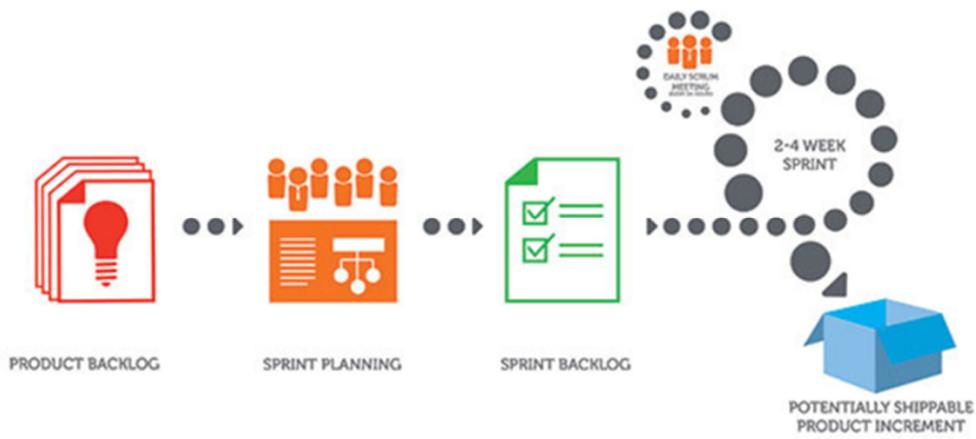


Figure 5 -The Scrum process. Source (Scrum Alliance, 2018).

Scrum is best used to manage complex software projects where the requirements have a high chance of changing, and for projects that do not fit a traditional SDLC such as the Waterfall process (Thompson, 2012).

Scrum should ideally be used for projects where:

- Requirements change frequently;
- Feedback is required often;
- Stakeholder feedback is needed to prioritise the requirements;
- Working prototypes are required for client feedback as they are completed;
- Fast turnaround time for the high priority requirements;
- The entire project's steps aren't known and can't be accurately estimated;

The project's increments are not long and don't require specialised resources (Thompson, 2012).

We have decided to use the Scrum methodology in our project as it is most suited to our project, where we may have changing requirements and a complex system.

The figure below illustrates a detailed diagram of the scrum process:

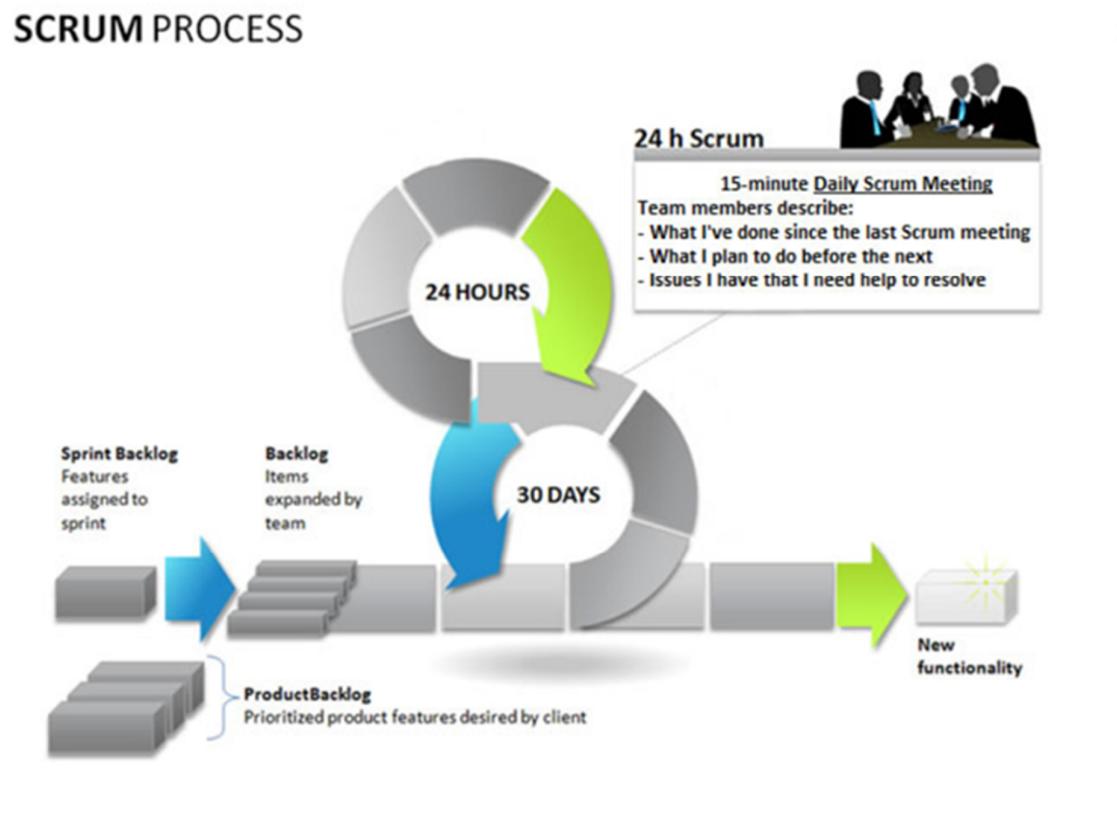


Figure 6 - The Scrum Process. Source: (Thompson, 2012).

## DEVELOPMENT PLAN

### REQUIREMENT INCREMENTS

The purpose of this development plan is to define and outline the increments that will make up the EatForLife system. Our estimated timeframe for development is 3 months, from 06/06/2018-30/09/2018, in order to integrate the system and test all functionalities before the November due date. All of the below increments are to be completed between Deliverable 4, and Deliverable 5.

The table below illustrates the increments for the EatForLife system.

*Table 7 - Requirement increments*

Increment	Description	Time	Testing	Requirement
		allocated	time	
1	<b>Develop prototype user interfaces</b> - this will be a working shell of the graphical user interface and navigation methods through the application.	7 days	3 days.	NFUR-7, NFUR-8, NFUR-9, NFUR-10
2	<b>Creation and linking of SQL database</b> - this will be the testing environment database, as the system will be integrated into EatForLife's current database. The database should be created according to the existing specifications and linked to the application.	7 days	3 days	THR-10, NFUR-5, NFUR-6
3	<b>Login, registration and user authentication functionality</b> - users should be able to login and register through the application.	7 days	3 days	FUR-1, FUR-2, FUR-3, NFUR-3, NFUR-4

4	<p><b>Profile page, measurements and update details functionality -</b></p> <p>implement all functionality for user profiles to be added, including:</p> <ul style="list-style-type: none"> <li>Updating personal details - including password change.</li> <li>Updating progress and measurements.</li> <li>Providing graphs and reports based on measurements.</li> <li>Updating BMI.</li> <li>Displaying user badges.</li> <li>Updating user goals and goal progress.</li> </ul>	14 days	3 days	FUR-3, FUR-4, FUR-12, FUR-13, FUR-14, FUR-15, FUR-16, FUR-21, FUR-22, FUR-23
5	<p><b>Diet plan functionality -</b></p> <p>implement all the functionality to the diet plan page.</p> <p>This includes:</p> <ul style="list-style-type: none"> <li>Generate weekly meal plan.</li> <li>Information about each food on the meal plan.</li> <li>Checklist functionality.</li> <li>Alternative substitutes for foods.</li> <li>Download/email plan functionality.</li> <li>Generating a weekly shopping list based on the meal plan.</li> <li>Download/email shopping list functionality.</li> </ul>	14 days	3 days	FUR-6, FUR-7, FUR-8, FUR-9, FUR-10, FUR-11, FUR-19, FUR-20

6	<b>Communications functionality</b> - implement all functionality for communications between the user and dieticians.	7 days	3 days	FUR-5
7	<b>Contact us page functionality</b> - implement functionality for the user to contact the developers and EatForLife	7 days	3 days	NFUR-7

The above table illustrates the requirement increments that will make up the EatForLife system. The time frame for the development can also be obtained in these tables.

## UPDATED SCHEDULE

### GANTT CHART FOR DELIVERABLE 2

The figure below illustrates the Gantt chart for the second deliverable for the EatForLife mobile application project:

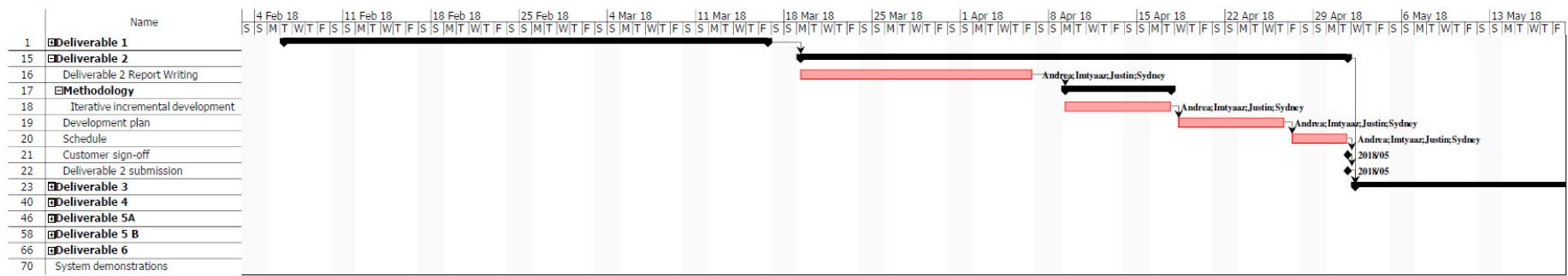


Figure 7 - Timeline for Deliverable 2

Figure 7 - Timeline for Deliverable 2 shows the content planning for the second deliverable of the project; the deliverable dates, as well as resource names, can be seen in this figure.

## UPDATED RISK REGISTER

The updated risk register for the EatForLife system development is shown in the table below:

*Table 8 - Updated Risk Register*

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R06	1	Cross-over incompatibility	The application may work well on Android but not on iOS	Technology Risk	The chosen coding platform does not support the needed language conversion	The chosen platform is incorrect for development of both Android and iOS	Focus on the functional platform to deliver the best full working project, the other platform can be integrated further down the line.	DOT Development	Medium	High
R02	2	Delay of required client information	The client takes a long time to respond to the group's questions via email or does	People Risk	The client does not regularly read their email, is too busy to reply or answer the phone or	The client isn't as invested as they should be in developing their system	Try complete information needed through other research methods - online etc.	DOT Development	Medium	Medium

			not answer calls consistently		is impeded by another variable					
R03	3	Uncertainty of full client requirements	The client gives the team a vague list of what he/she wants the system to be and what it should entail	People Risk	The team does not aggressively ask for the requirements	Identified during meetings with the client	Send the client a system proposal with requirement ideas for their feedback	DOT Development	Medium	Medium
R11	4	Update of requirements	The client decides to change requirements for the system mid-development	People Risk	The client no longer wants to have specific features in the system or wants to add additional features to the system	The client's business develops new requirements for their future success	The team will ensure that a meeting takes place in order to document the new requirements appropriately	DOT Development	Medium	Medium

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R04	5	Bad communication skills among team members	Team members do not communicate appropriately and/or frequently enough	People Risk	Due to team members not owning a phone, not using discord (our team's communication application) or google drive	Team members don't know how to work well with each other	Discuss it with the person, if there is no improvement escalate the issue with your lecturer.	DOT Development	Low	High
R05	6	The resignation of a team member	Team member decides to resign mid-way during the development process	People Risk	An unexpected turn of events or too much stress for the team member to handle	The team member can no longer take the pressure of developing the system	The team will have to accept the decision for the resignation of the member and continue development	DOT Development	Low	High
R01	7	Loss of business	The client decides to no longer work with our team	Market Risk	The client has lost interest in the system or no longer	The client's business no longer	Find a new client, or if too late, continue without the client	Client	Low	High

					wants to invest the time	needs a new system to further their success				
R07	8	Fail to choose a suitable coding language	The team does not choose a suitable coding language the is required for the chosen/designed system	People Risk	Team members may be unsure of which language(s) best suit their skills and the requirements for development	There are multiple languages to choose from and team members have their own unique skills in their respective languages	Use a familiar well documented language	DOT Development	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R12	9	Hardware failure	The developers have an unforeseen hardware failure, resulting in critical data loss.	Technology Risk	Poor system maintenance, faulty computer components, untrustworthy software installed on the system	Team members did not ensure their hardware was functional and that critical data was backed up	Ensure that all critical data is backed up through Google Drive, GitHub, and other cloud storage, or offsite storage.	DOT Development	Low	High
R10	10	Quality of the system is not satisfactory for the client	The client decides that they do not like the developed system and no longer want to use it	Market Risk	The developed system is not up to the client's standards nor meets their expectations	The developed system is invalid for the use of the client	As a group ensures that all requirements are met. If the client does not like a system that meets the requirements, terminate the project.	Client	Low	High

R08	11	Chosen IDE is inappropriate for development	The chosen IDE may not be compatible with required plugins or not contain much needed tools for development	Technology Risk	Team members do not know of various IDE's that may be suitable for the development of the system	The chosen IDE does not support the needed tools for the input of needed plugins or modules	Find a new IDE that suits the project	DOT Development	Low	Medium
R09	12	Poor time management	Increments are put together late	People Risk	Due to team members delaying input due to incompetence or personal issues	Team members do not know how to schedule their time well enough to get work done	Discuss with the individual why their work is not timely, assist them if needed to finish the deliverables in the schedule. If the team members do not improve their time management, escalate it to the lecturer.	DOT Development	Low	Low

# USER REQUIREMENTS DOCUMENT

## INTRODUCTION

Since the user requirements document allows a user to know what to expect the software to do, aspects such as the purpose of the EatForLife app and the scope of the system will be discussed in this document. Sections such as definitions, acronyms, and abbreviations are included, to provide a simplistic understanding of the document. Along with the aspects mentioned above the system requirements will also be included. This section will provide an inside to the functionalities of the system and the user characteristics.

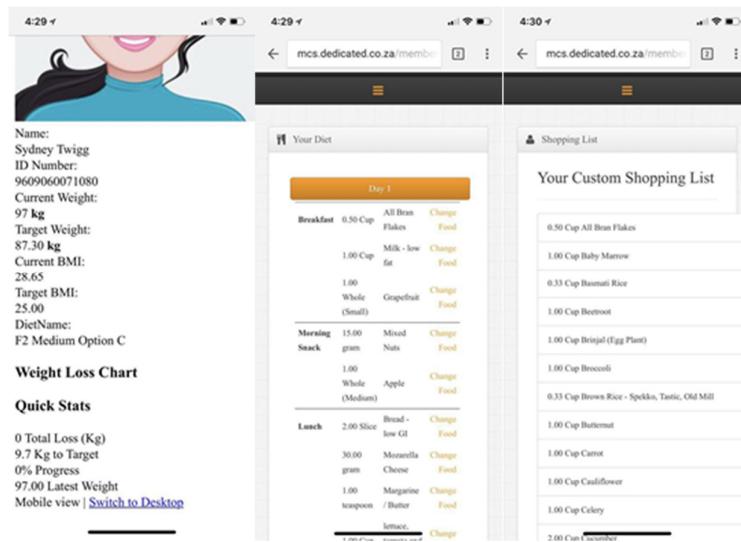
## PURPOSE

The EatForLife system is categorised as a transaction system since the system will be able to collect, process, store, modify, display and cancel transactions (Cress, 2009). The system is also classified because the system stored the data that is collected in a database (Cress, 2009). Along with this the fact that the stored data will be used to generate feedback to the user (Cress, 2009)

This document is constructed for the use of our customer Matthew Grossett, the CEO of the EatForLife company (Robertson & Robertson, 2006). This document is constructed to lay out precisely what the system will consist of and what it will do (Robertson & Robertson, 2006). The reason as to why we are construction a user requirement document is that as the developers cannot claim that the system is complete and ready if not all of the requirements are met. The documents are also constructed to ensure that the customer cannot demand requirements for the system that are not mentioned in this document (Robertson & Robertson, 2006).

The primary purpose of the proposed EatForLife mobile app is to enhance the current system and user experience. The current EatForLife system is only web-based, and while users can log in to the site through their mobile web browsers, the website is not optimised or ideal for mobile devices.

The following screenshots provide a view of the current mobile view on the EatForLife website:



*Figure 8 - Screenshot's showing the mobile view of the EatForLife system. Screenshots taken on an iPhone X using the Google Chrome web browser.*

As shown above, the user experience is not ideal for a mobile device. A website acts as a form of marketing for a company, and many people would be put off by an inaccessible or hard to read the website - especially in the current times where most people access the internet via smartphones or tablets. South Africa has 21 million internet users - with a majority using mobile devices to access the internet (Shapshak, 2017). To expand the current customer base and make the eating programme more accessible and easier to use for the South African market EatForLife requires a mobile app.

A mobile app provides many benefits to a business, such as the following outlined by (Haselmayr, 2018):

- Be visible to the customer at all times;
- Create a direct marketing channel;
- Build your brand and recognition;
- Improved customer engagement; and
- Stand out from the competition.

Thus, the purpose of this project is to provide a mobile app that is fully functional, easily accessible, user-friendly, cross-platform, and well designed for EatForLife customers to access their personalised eating programmes.

## SCOPE

The system which the dot development team will develop is the EatForLife system. The app allows a member to log in to their personalised account. The app will provide the registered menu to have access to a customisable eating plan. A specialised shopping list based on the selected eating plan is generated for the guidance of the system. The user will be able to capture and update personal details. Based on these details the user will receive feedback charts. The registered member will be able to have direct feedback and communication services with a dietician.

The scope of the system includes engagement and communication with EatForLife, integrating the existing system into the new system and communication over systems requirements; understanding and analysing the current system process - as well as making suggestions on how EatForLife can change current methodologies to enhance the user experience.

Benefits that the system will provide are:

- The system will be functional on both Android and iOS mobile devices.
- A user-friendly app will be delivered.
- It will provide a direct marketing network.
- The system will allow easy access.
- Around the clock, access will be provided for the customers.
- EatForLife will build their brand recognition.
- User engagement will be enhanced.
- The app will allow EatForLife to stand out.
- Communication between customer and client will be improved.
- Customers experience will be improved.

Goals set for the EatForLife app:

- Improve customer experience from the current system.
- Allow the app to run both on Android and iOS mobile devices.
- Design a user-friendly app for EatForLife.
- Allow users of the app to have a comfortable communication platform.

Goals set for the development of the system:

- Develop the system according to schedule.
- Ensure that all the requirements are met.
- Development of a user-friendly interface that meets the Nielson's Heuristics.
- Ensure the system is tested thoroughly.
- Hand over a fully functional system for our client EatForLife.

## ACRONYMS AND ABBREVIATIONS

The table below indicates the acronyms and abbreviations used through the development of this user requirement document.

*Table 9 - Acronyms and abbreviations*

Abbreviation and Acronyms	Expanded
°C	Celsius
App	Application
DDR	Double Data Rate
Etc	et cetera
GB	Gigabyte
GHz	Gigahertz
OS	Operating system
RAM	Random-access memory
SQL	Structured Query Language
TB	Terabyte
URD	User Requirement Document

From the table above all the abbreviations and acronyms used in this document can be identified.

## DEFINITIONS

The table below indicates terms that will be used in this user requirements.

*Table 10 - Definitions used in the User Requirements Document*

Term	Definition
<b>Application</b>	It is software that runs on a computer (Campbell, 2015).
<b>Android OS</b>	Is an operating system developed by Google design for mobiles cell phones to quickly provide real actions as swiping, tapping and pinching to manipulate objects on screen (Rouse, 2017).
<b>Functionality</b>	A capability in which a device it can perform (Bass, 2012).
<b>Interface</b>	Computer hardware or software designed to communicate information between hardware devices, between software programs, between devices and programs or between a device and a user (Sonmez, 2010).
<b>Mobile Application</b>	A portable computer program used for a particular type of job or purpose (Rouse, 2017).
<b>Mobile Device</b>	A portable, wireless computing device that is small enough to be used while held in the hand/handheld (Viswana, 2017).

The table above explains some of the uncertain terms used through this user requirement document.

## OVERVIEW

The system requirements for the EatForLife app will be explained. More detail for the functional, non-functional and technical requirements are provided. The user characteristics will of the system will be identified as well as the operating system for the system.

## SYSTEM REQUIREMENTS

### SYSTEM PERSPECTIVE

The EatForLife mobile application is related to the current existing EatForLife website; these will be linked together in the implementation stage of the development. Registered users for the EatForLife program will be able to access their accounts on both the mobile application and the website, and all information will be synced between the two in real time. At the implementation stage, the mobile application and existing website will share a hosted database, however, during the development and testing stages, the mobile application will have its testing environment database that is based on the existing website's database structure.

### FUNCTIONAL REQUIREMENTS

#### *LIST OF FUNCTIONAL REQUIREMENTS*

All the functional requirements for the EatForLife system are listed below. An identification code is allocated to all the requirements. A short description of each requirement is also provided. The requirements are arranged according to priority (EatForLife, 2017; Schwalbe, 2015).

*Table 11 - Functional Requirements for the EatForLife Mobile Application.*

ID	Requirement	Priority
<b>FUR-1</b>	The system allows users to register.	High
<b>FUR-2</b>	The system allows users to login.	High
<b>FUR-8</b>	The system provides information about each item in the eating plans to users.	High
<b>FUR-9</b>	The system allows users to have their health statuses analysed and a meal plan can be created accordingly.	High
<b>FUR-10</b>	The system allows users to find alternative items to the substitute foods for in the diet plan.	High
<b>FUR-11</b>	The system provides users with a checklist to check off food once eaten on the diet plan.	High

ID	Requirement	Priority
<b>FUR-17</b>	The system will allow users to download their customized 7-day meal plan directly through the mobile application.	High
<b>FUR-18</b>	The system will allow users to have their 7-day meal plan emailed to their provided email addresses.	High
<b>FUR-19</b>	The system will allow the user to download their 7-day meal plan shopping list directly through the mobile application.	High
<b>FUR-20</b>	The system will allow the user to have their 7-day meal plan shopping list directly through the mobile application.	High
<b>FUR-21</b>	The system allows the user to reset their password	High
<b>FUR-23</b>	The system allows the user and dietitians to log out.	High
<b>FUR-24</b>	The system allows the dieticians to login to the application as an admin user	High
<b>FUR-3</b>	The system allows a user to input their details.	Medium
<b>FUR-4</b>	The system allows a user to edit their details.	Medium
<b>FUR-5</b>	The system allows users to send queries to and communicate with dietitians.	Medium
<b>FUR-6</b>	The system allows users to get customised diet plans based on specific user parameters.	Medium
<b>FUR-7</b>	The system provides information about diet plans to users.	Medium
<b>FUR-12</b>	The system can calculate the BMI for users.	Medium
<b>FUR-13</b>	The system can calculate the correct BMI for users based on user inputted data.	Medium
<b>FUR-14</b>	The system allows users to monitor their progress with reports and graphs.	Medium
<b>FUR-15</b>	The system allows users to monitor their measurement and weight history with reports and graphs.	Medium
<b>FUR-25</b>	The system will direct the dietician to the admin panel on login	Medium
<b>FUR-26</b>	The system will allow the dietician to view a list of all registered clients and their details.	Medium

ID	Requirement	Priority
<b>FUR-27</b>	The system will allow the dietician to view a detailed view of each registered client.	Medium
<b>FUR-28</b>	The system allows dietitians to monitor the progress of users through the admin panel.	Medium
<b>FUR-29</b>	The system will allow an admin user to add another admin user through the admin page.	Low
<b>FUR-16</b>	The system will allow the user to set weight loss goals.	Low
<b>FUR-22</b>	The system allows the user to assign a profile picture to their accounts	Low

## USE CASE DIAGRAM

The diagrams below illustrate the high-level use case diagram for the EatForLife system. The actors involved in the system are identified as well as the different interactions that these users have with the system.

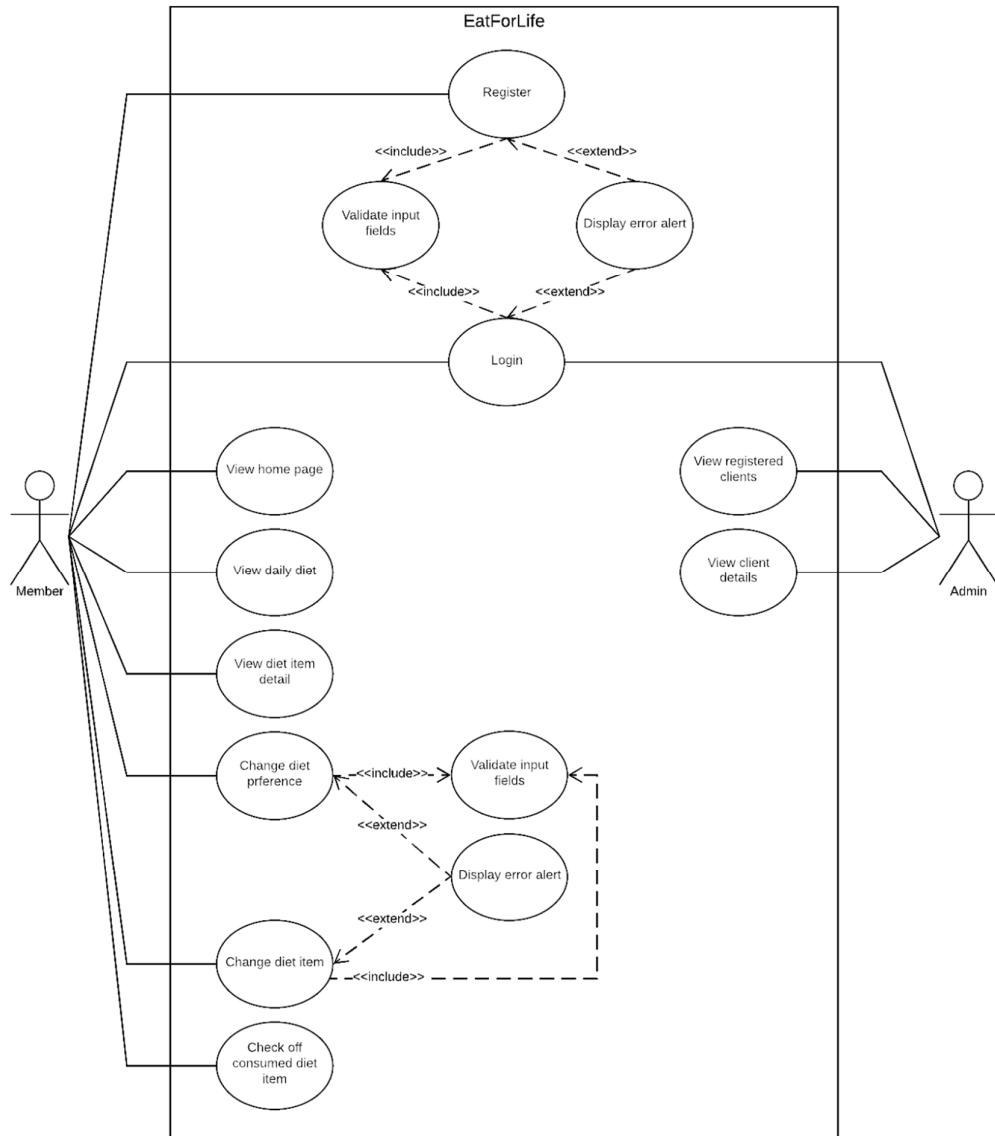


Figure 9 - Use Case Diagram for the EatForLife Mobile Application.

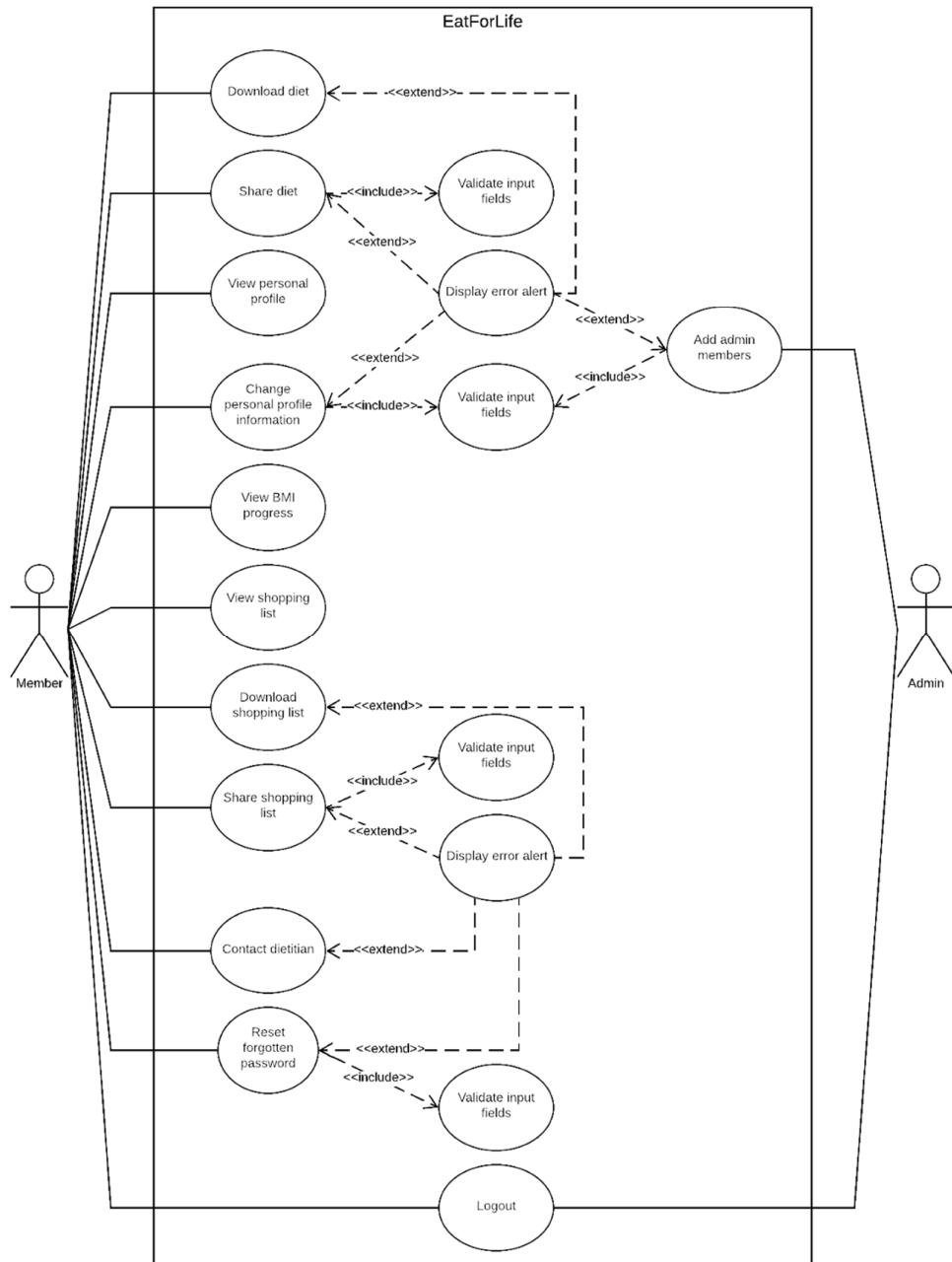


Figure 10 - Use Case Diagram for the EatForLife Mobile Application.

## NON-FUNCTIONAL REQUIREMENTS

All the non-functional requirements for the EatForLife system are listed below. An identification code is allocated to all the requirements. A short description of each requirement is also provided. The requirements are arranged according to priority (EatForLife, 2017; Schwalbe, 2015).

*Table 12 - Non-Functional Requirements for the EatForLife Mobile Application.*

ID	Requirement	Priority
<b>NFUR-1</b>	The system must allow the user to have access to it at all times.	High
<b>NFUR-2</b>	The system must allow dietitians to have access to it at all times.	High
<b>NFUR-3</b>	The system should only allow the user access to the system if the password and username is correct	Medium
<b>NFUR-4</b>	The system should only allow the user dietitians to the system if the password and username is correct	Medium
<b>NFUR-7</b>	The system should be easy to use.	Medium
<b>NFUR-8</b>	The colour scheme should be consistent throughout the system.	Medium
<b>NFUR-9</b>	The styles should be consistent throughout the system.	Medium
<b>NFUR-10</b>	The themes should be consistent throughout the system.	Medium
<b>NFUR-5</b>	The system should validate the data entry of the users.	Low
<b>NFUR-6</b>	The system should validate the data entry of the dietitians.	Low

## TECHNICAL REQUIREMENTS

All the technical requirements for the EatForLife system are listed below. An identification code is allocated to all the requirements. A short description of each requirement is also provided. The requirements are arranged according to priority (EatForLife, 2017; Schwalbe, 2015).

*Table 13 - Technical Requirements for the EatForLife Mobile Application.*

ID	Requirement	Priority
<b>Developer's Hardware</b>		
<b>THR-1</b>	Laptop capable of running Android Studio and/or XCode if on an Apple Laptop.	High
<b>THR-2</b>	PC capable of running Android Studio and/or XCode if on an Apple Mac.	High
<b>THR-3</b>	Mobile device running Android OS.	High
<b>THR-4</b>	Mobile device running iOS.	High
<b>Client Hardware</b>		
<b>THR-5</b>	Mobile device running Android OS.	High
<b>THR-6</b>	Mobile device running iOS.	High
<b>THR-7</b>	PC and/or laptop capable of running emulation software.	High
<b>Developer's Software</b>		
<b>THR-8</b>	The system must be developed for Android and iOS.	High
<b>THR-9</b>	The system must be developed with JavaScript and CSS.	High
<b>THR-10</b>	The system must make use of an SQLite database.	High
<b>THR-11</b>	The system will be designed with NativeScript.	High
<b>THR-12</b>	The developers must have internet access.	High
<b>Client Software</b>		
<b>THR-13</b>	The user must have an Android or iOS device.	High
<b>THR-14</b>	The user must have internet access.	High

## USER CHARACTERISTICS

The table below identifies the users of the EatForLife system as well as information about the users of the system.

*Table 14 - Users involved in the EatForLife system*

User Role	User Type	Level of Experience	Language Preference	Technical Expertise
<b>Client/User</b>	Novice	Low	English	Able to view and edit account and personal data.
<b>Dietician</b>	Novice	Moderate to High	English	Able to create, upload, edit and delete data on the system.
<b>Database Manager</b>	Expert	High	English	Able to maintain and update the database.
<b>Server Manager</b>	Expert	High	English	Able to maintain the server on which the database is present.

The users of the system are identified, along with the type of user, the level of experience required

## OPERATIONAL ENVIRONMENT

The table below indicates the operational environment for the EatForLife system.

*Table 15 - Operational environment*

System Section	Environment	Specification
<b>Core System</b>	Mobile device	<u><b>Operating System (OS):</b></u> iOS and Android
<b>Server</b>	Server Room	<u><b>Operating System:</b></u> Windows Server 2008 or better <u><b>Processor:</b></u> Intel Core i5 2.2GHz or better <u><b>RAM:</b></u> 4GB DDR3 <u><b>Hard Drive Allocation Size:</b></u> 1TB or more <u><b>Temperature:</b></u> Constantly air conditioned between 10°C or 28°C <u><b>Condition:</b></u> Abides by health & safety regulation
<b>Database</b>	Server	<u><b>Database:</b></u> SQL (Latest Version), <u><b>Database GUI:</b></u> SQL Workbench (Latest Version)

Specifications of the environments required for operation of the EatForLife system are specified in the table above.

# INFORMATION SYSTEMS DESIGN

## LOGICAL DESIGN

### USE CASE DIAGRAMS

#### OVERALL SYSTEM USE CASE

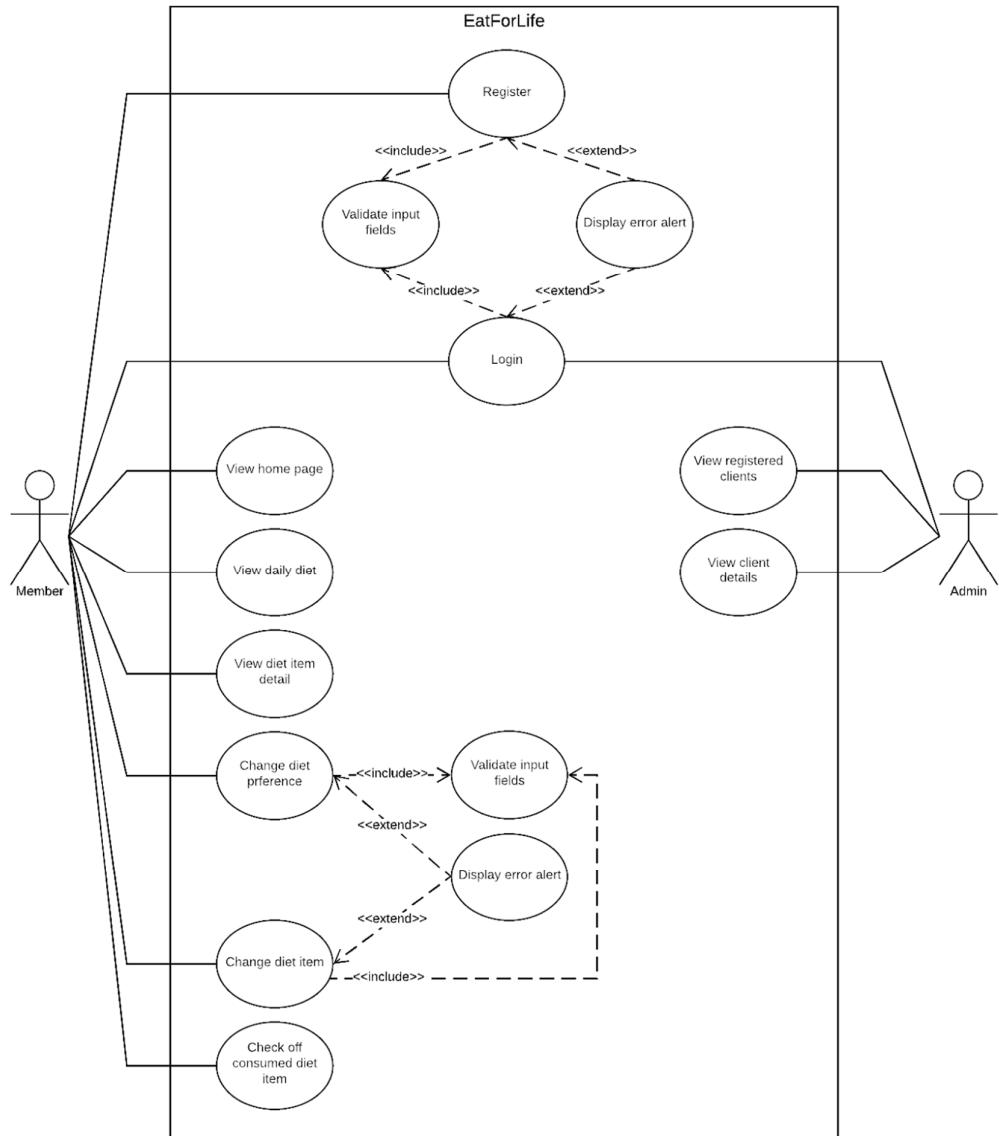


Figure 11 - Use Case Diagram for the EatForLife System.

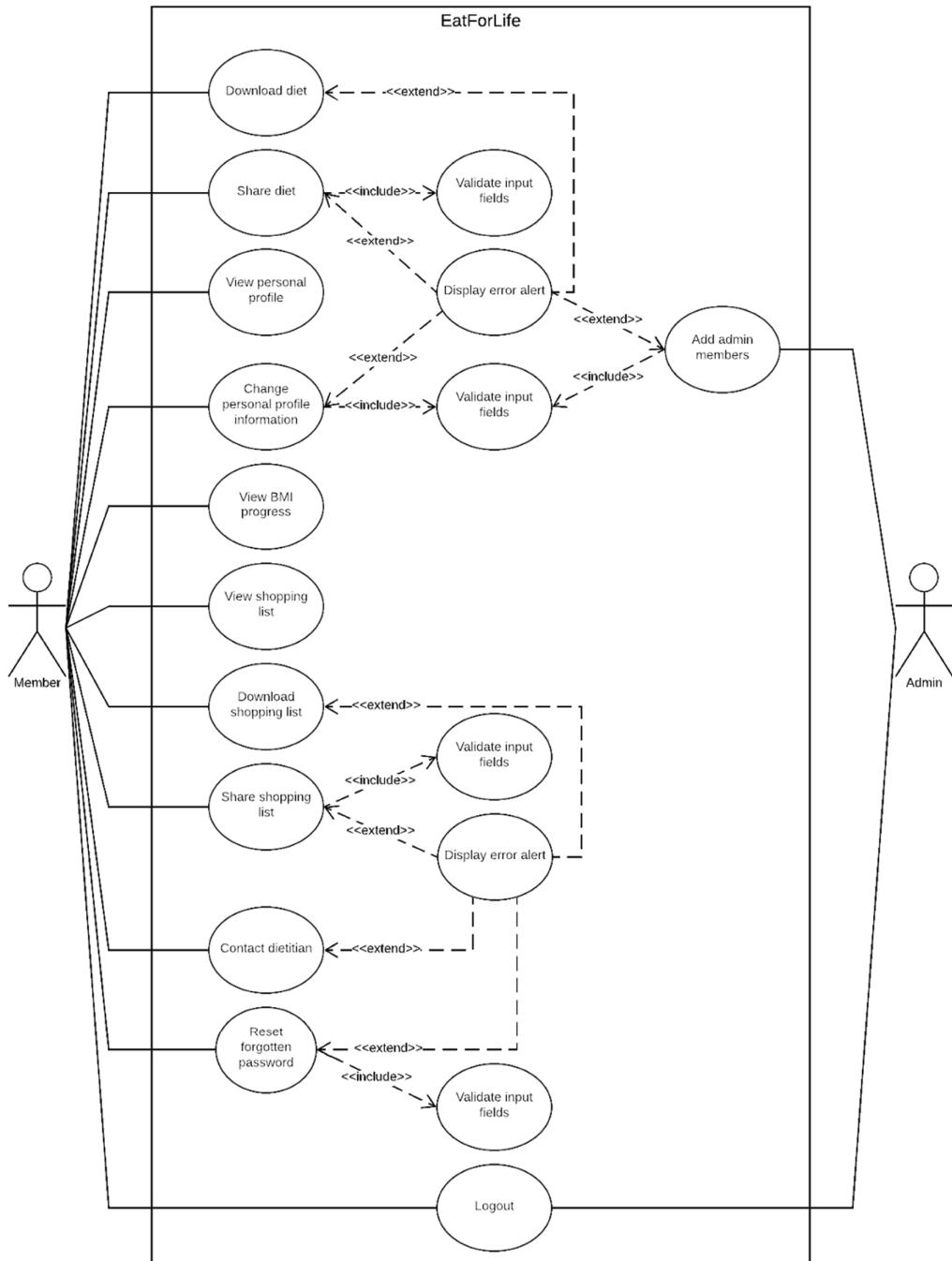


Figure 12 - Use Case Diagram for the EatForLife System.

## REGISTRATION USE CASE

The following figure illustrates the use case diagram for the registration event of the EatForLife system (LucidChart, 2018).

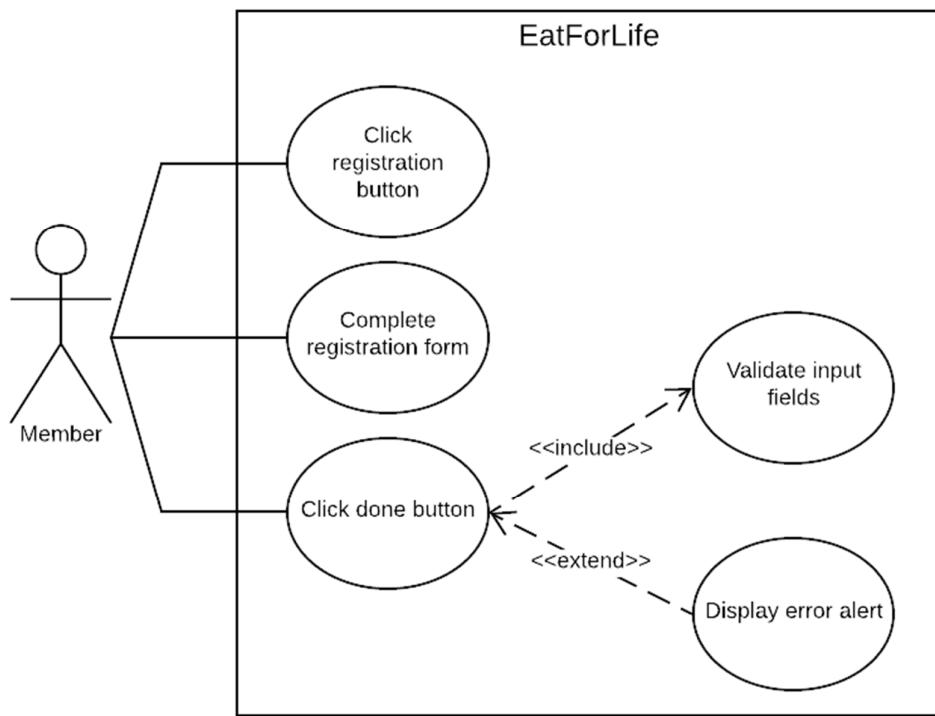


Figure 13 - Use Case Diagram for the registration event on the EatForLife Mobile Application.

## LOGIN USE CASE

The following figure illustrates the use case diagram for the login event of the EatForLife system.

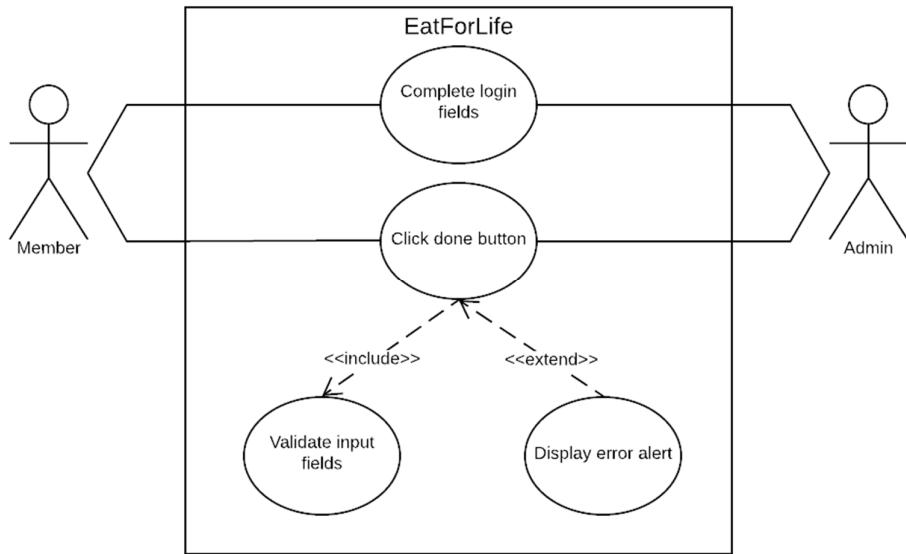


Figure 14 - Use Case Diagram for the login event on the EatForLife Mobile Application.

## VIEW PROGRESS USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to view their weight loss progress.

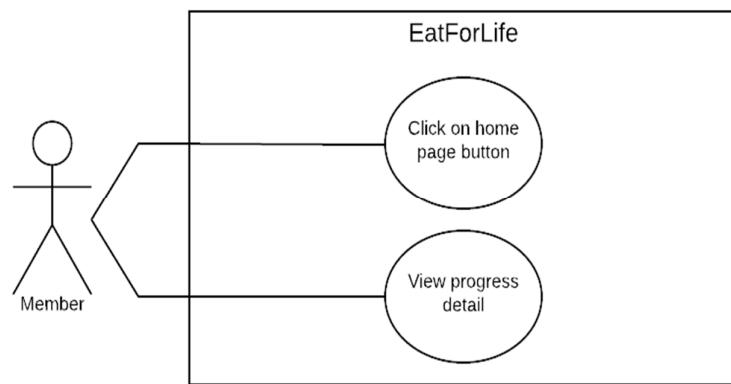


Figure 15 - Use Case Diagram for the login event on the EatForLife Mobile Application.

## VIEW MEAL PLAN USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to view their personal meal plan.

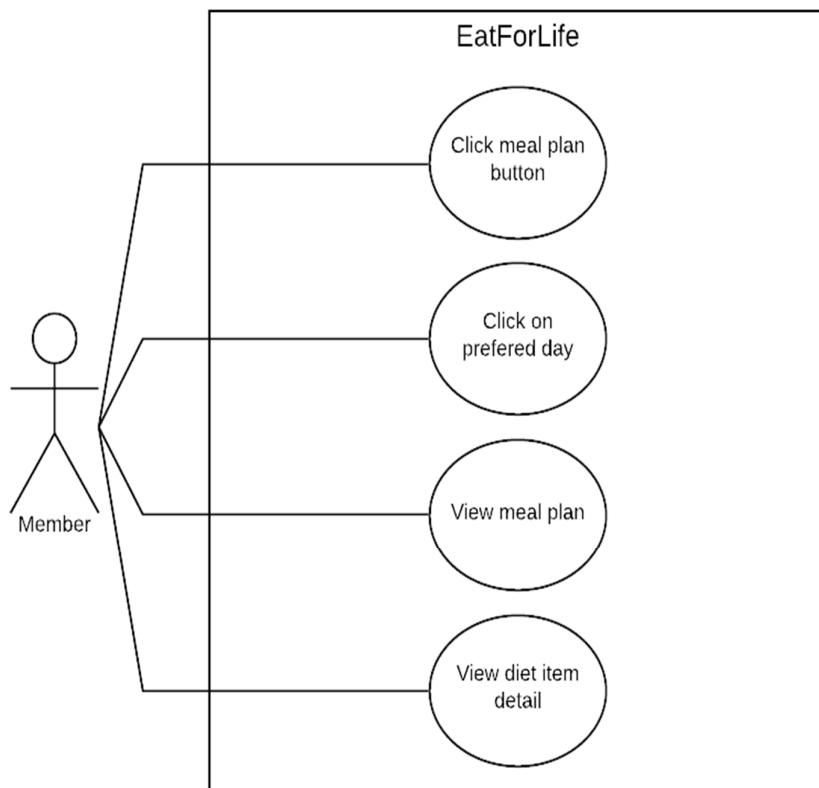


Figure 16 - Use Case Diagram for the view meal plan event on the EatForLife Mobile Application

## CHANGE DIET PREFERENCES USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to change the food options of their personal meal plan.

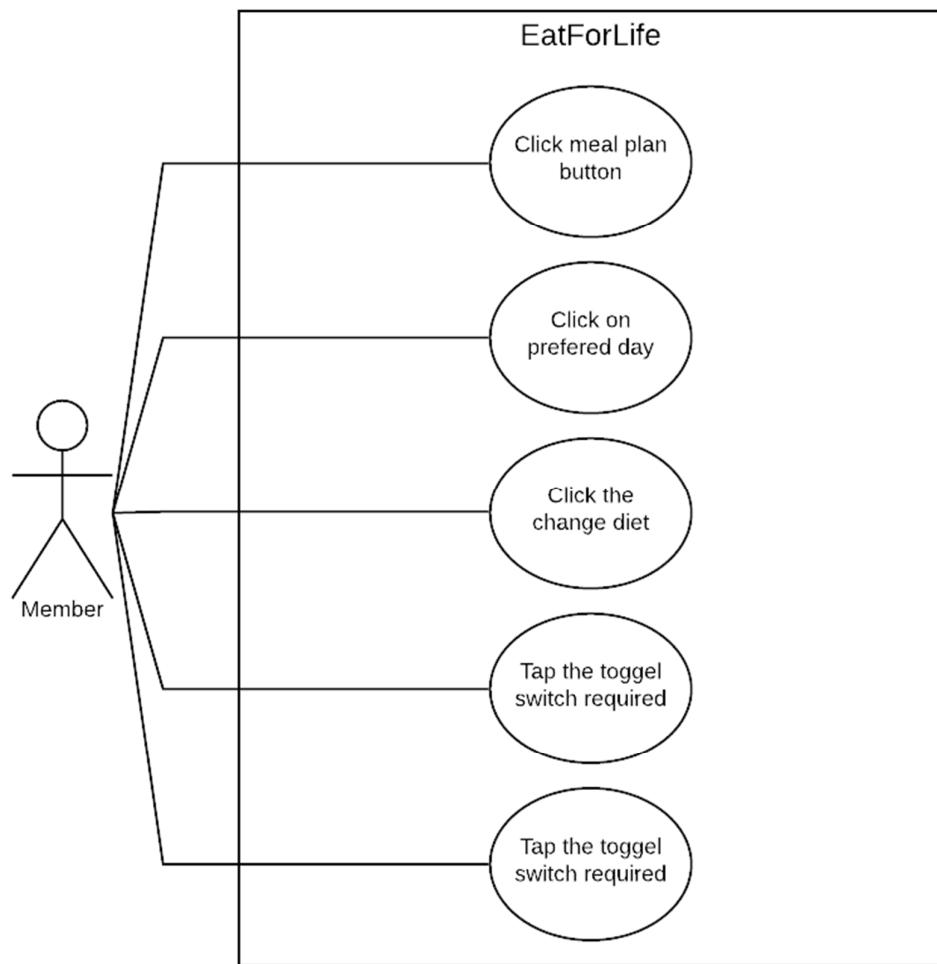


Figure 17 - Use Case Diagram for the edit diet preferences event on the EatForLife Mobile Application

## CHANGE DIET ITEM USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to substitute a food item on their meal plan.

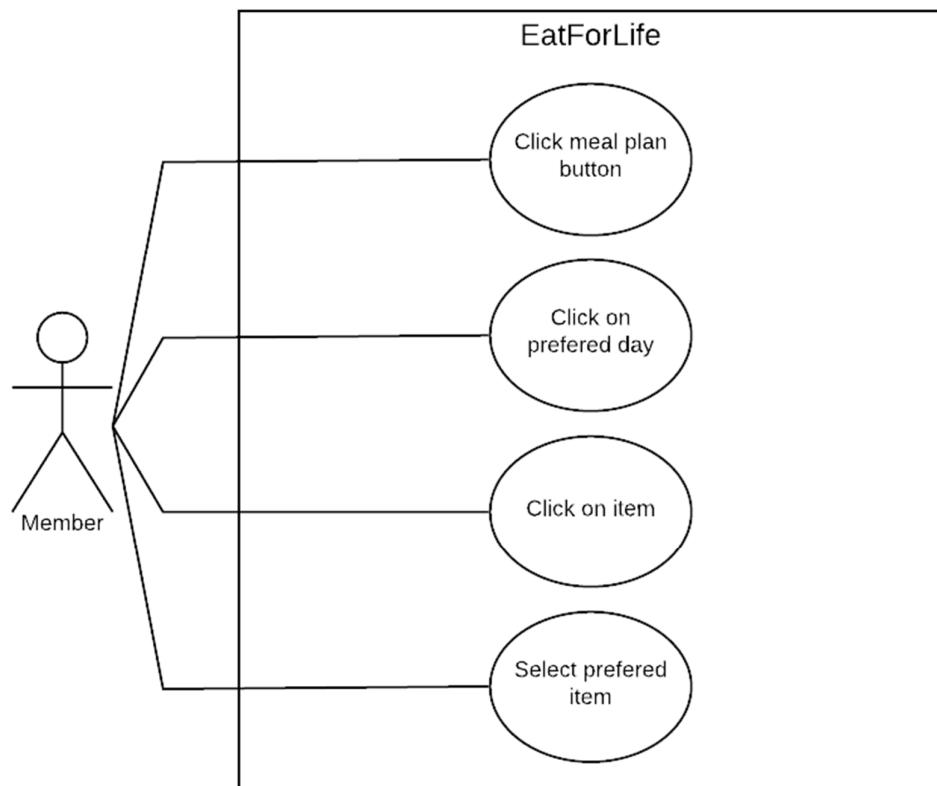


Figure 18 - Use Case Diagram for the change diet item event on the EatForLife Mobile Application

## MARK ITEM AS EATEN USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to indicate that an item on the food list has been eaten.

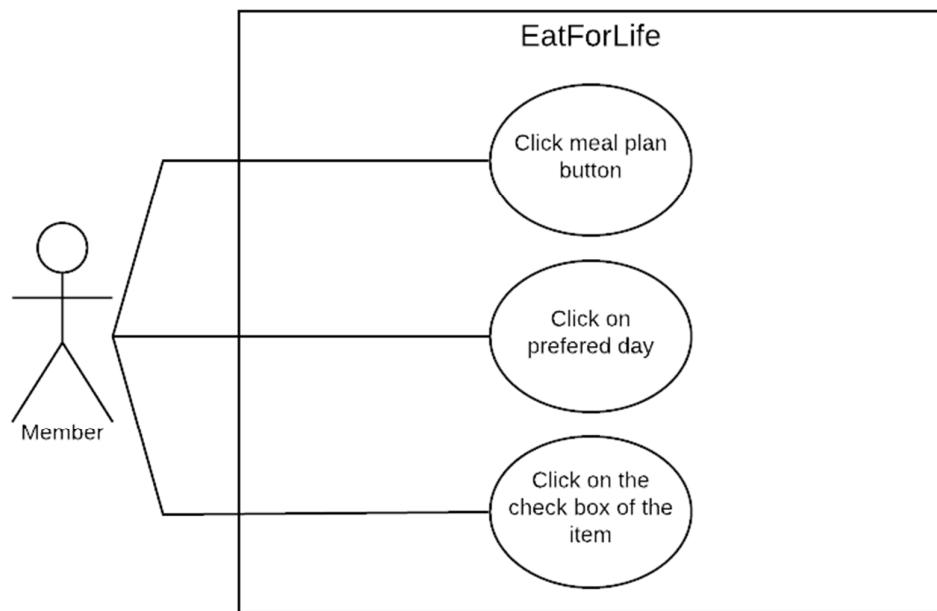


Figure 19 - Use Case Diagram for the mark item as eaten event on the EatForLife Mobile Application

## DOWNLOAD/SAVE MEAL PLAN USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to download or save their meal plan to local storage.

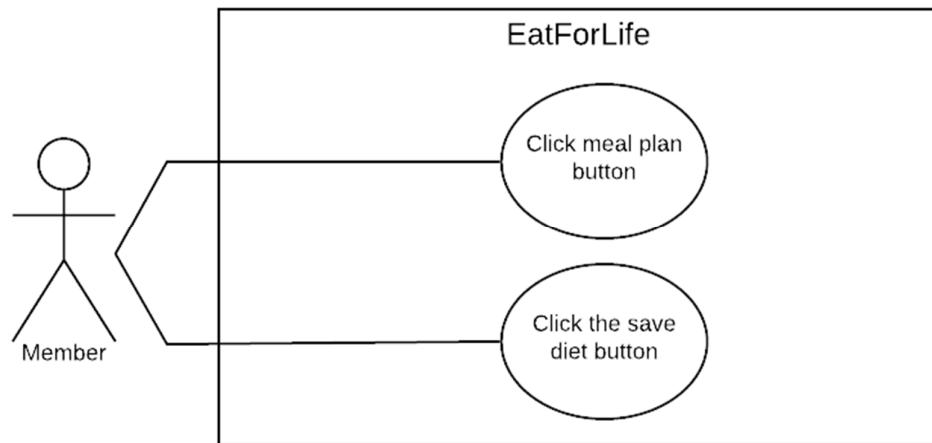


Figure 20 - Use Case Diagram for the download or save meal plan event on the EatForLife Mobile Application

## SHARE MEAL PLAN USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to share their meal plan via email.

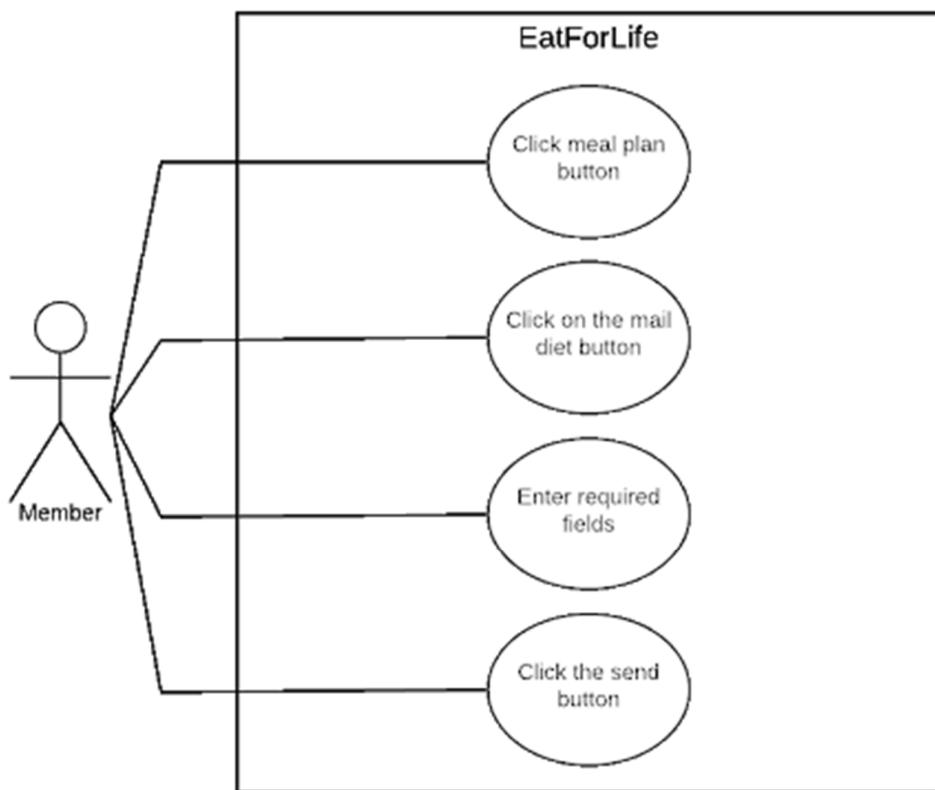


Figure 21 - Use Case Diagram for the share meal plan event on the EatForLife Mobile Application

## VIEW PERSONAL INFORMATION USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to view their personal information stored in the EatForLife mobile application.

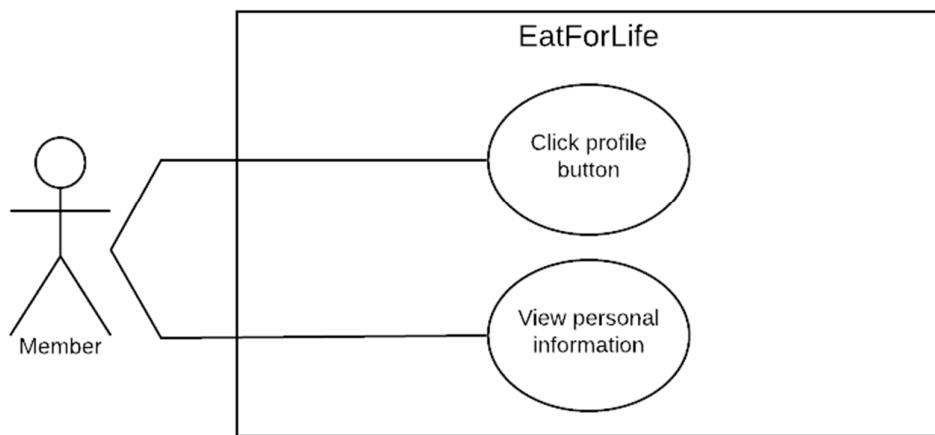


Figure 22 - Use Case Diagram for the view personal information event on the EatForLife Mobile Application

## CHANGE PERSONAL INFORMATION USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to update their stored personal information.

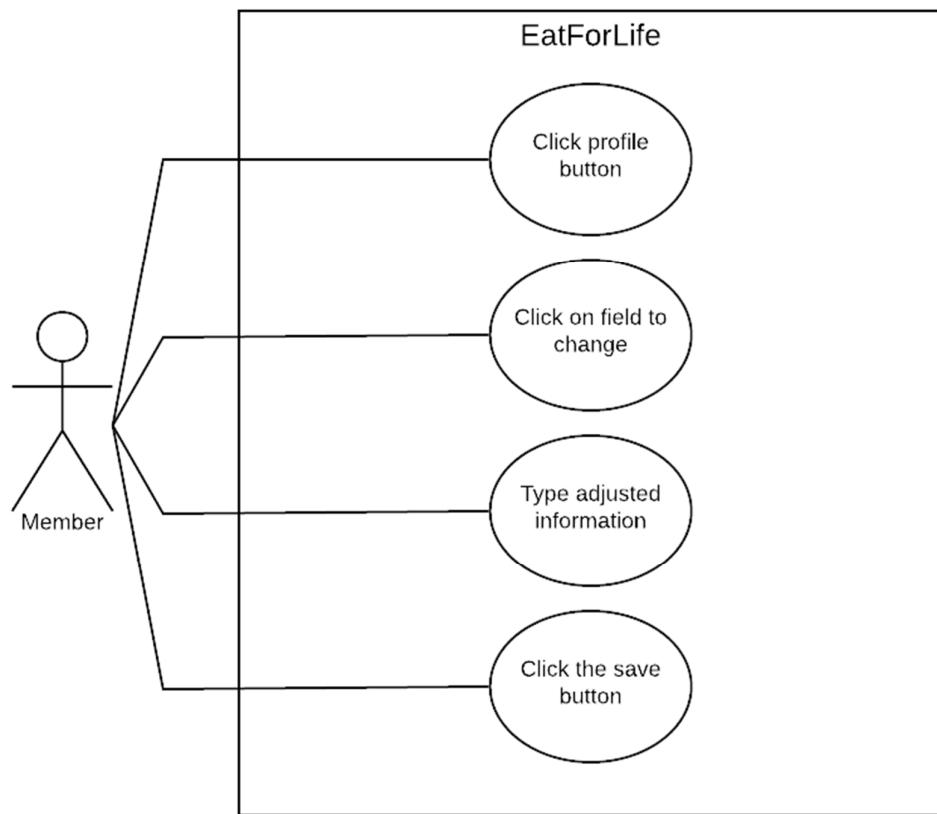
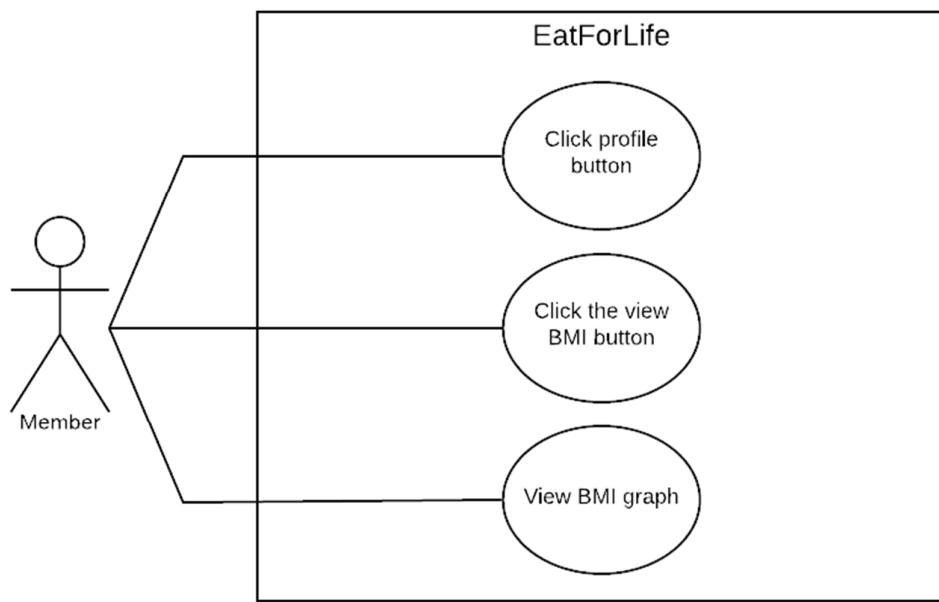


Figure 23 - Use Case Diagram for change personal information event on the EatForLife Mobile Application

## *VIEW BMI PROGRESS USE CASE*

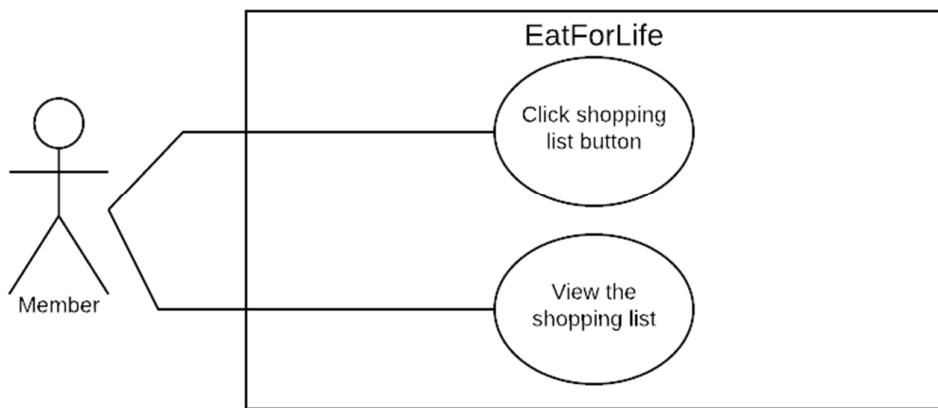
The following figure illustrates the use case diagram for a member of EatForLife to view their BMI history from their previously entered measurements.



*Figure 24 - Use Case Diagram for the view BMI progress event on the EatForLife Mobile Application*

## [VIEW SHOPPING LIST USE CASE](#)

The following figure illustrates the use case diagram for a member of EatForLife to view their generated personal 7-day shopping list.



*Figure 25 - Use Case Diagram for the view shopping list event on the EatForLife Mobile Application*

## DOWNLOAD/SAVE SHOPPING LIST USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to download or save their 7-day shopping list to local storage.

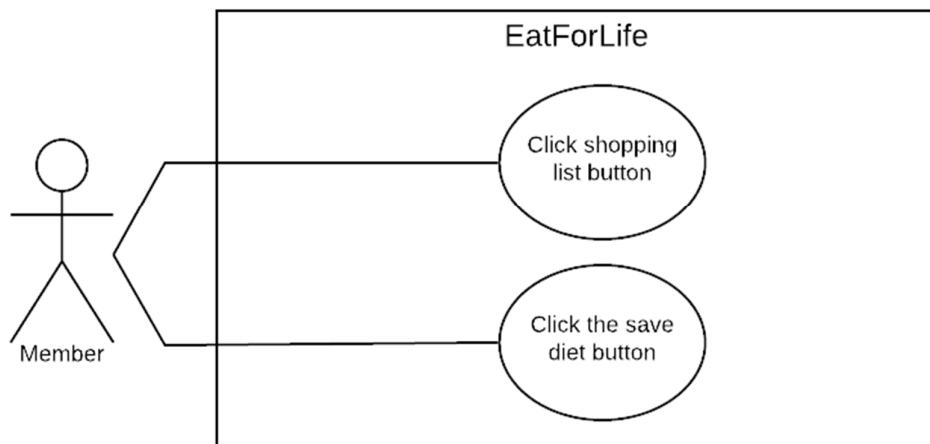


Figure 26 - Use Case Diagram for the download or save shopping list event on the EatForLife Mobile Application

## SHARE SHOPPING LIST USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to share their shopping list via email.

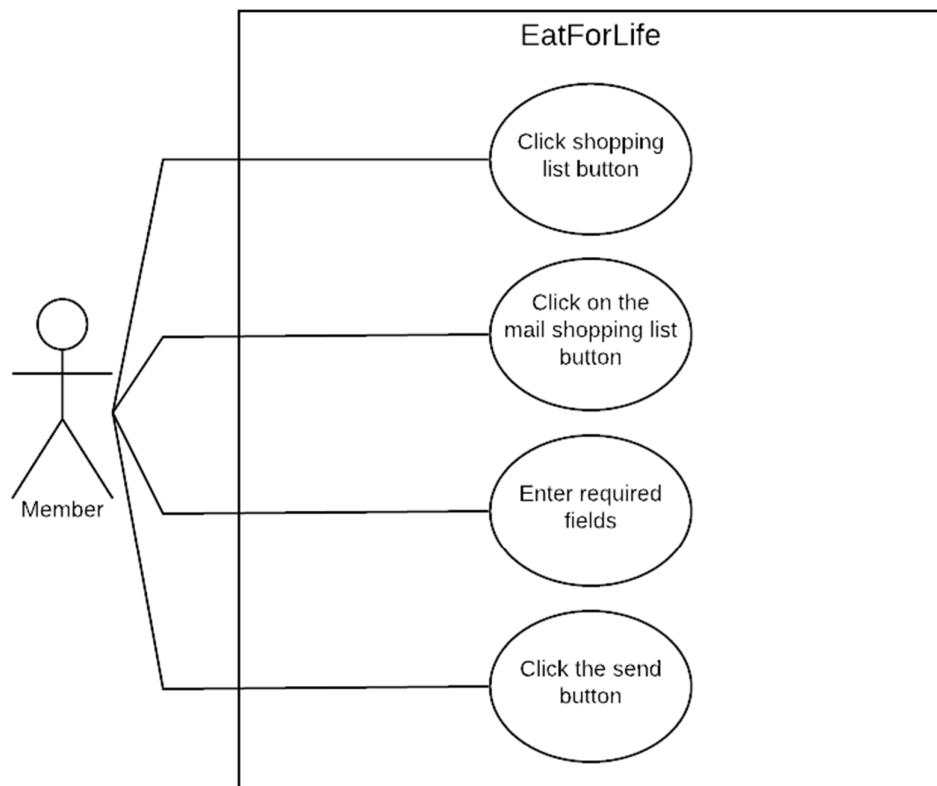


Figure 27 - Use Case Diagram for the share shopping list event on the EatForLife Mobile Application

## COMMUNICATE WITH DIETICIAN USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to communicate with their dietitian via email.

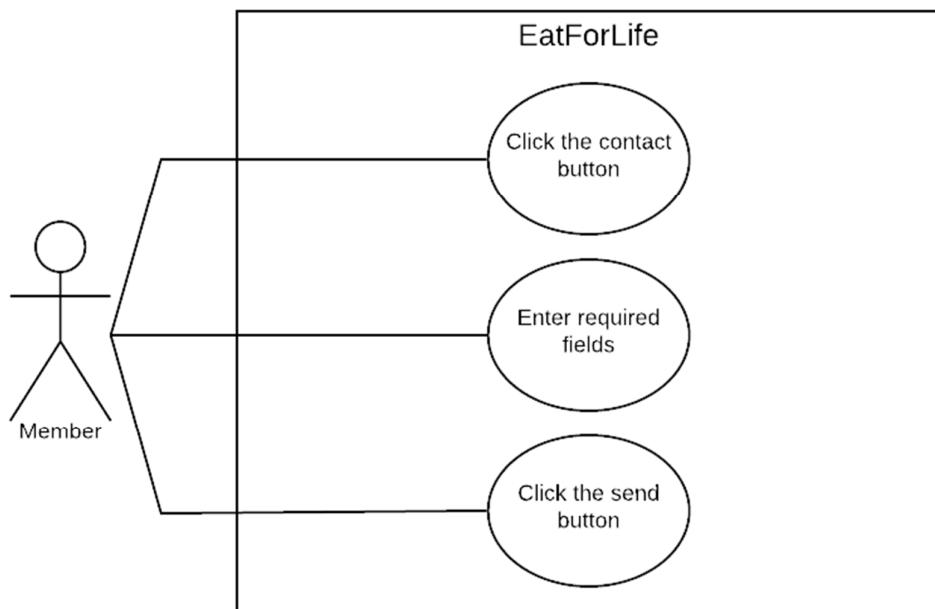


Figure 28 - Use Case Diagram for the communicate with dietician event on the EatForLife Mobile Application

## RESET FORGOTTEN PASSWORD USE CASE

The following figure illustrates the use case diagram for a member of EatForLife to reset their account password.

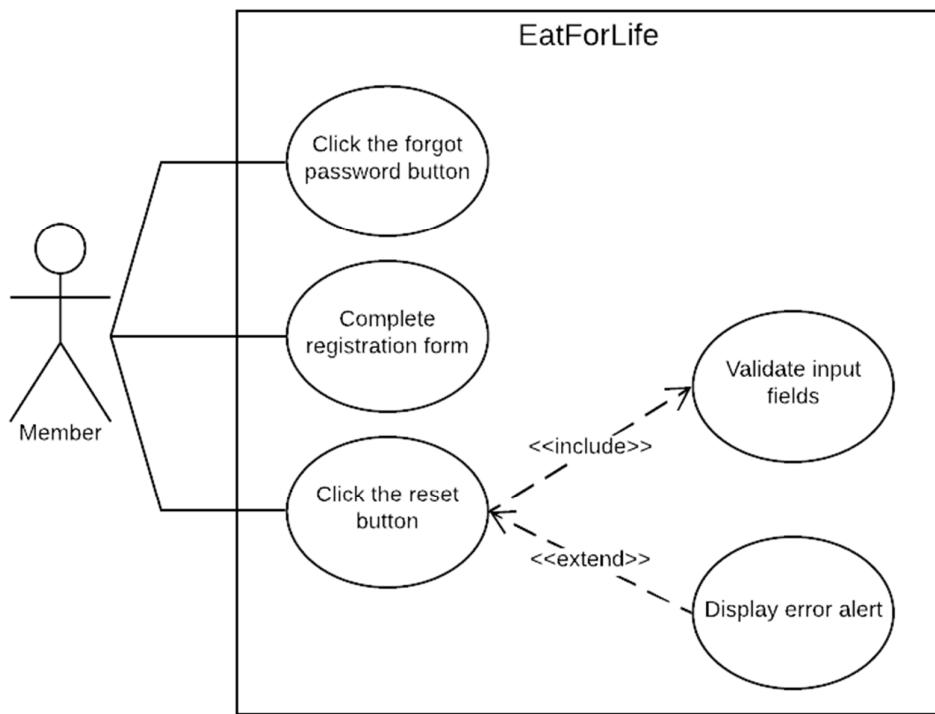


Figure 29 - Use Case Diagram for the reset password event on the EatForLife Mobile Application.

## LOG OUT OF APPLICATION

The following figure illustrates the use case diagram for a member of EatForLife log out of the EatForLife Application.

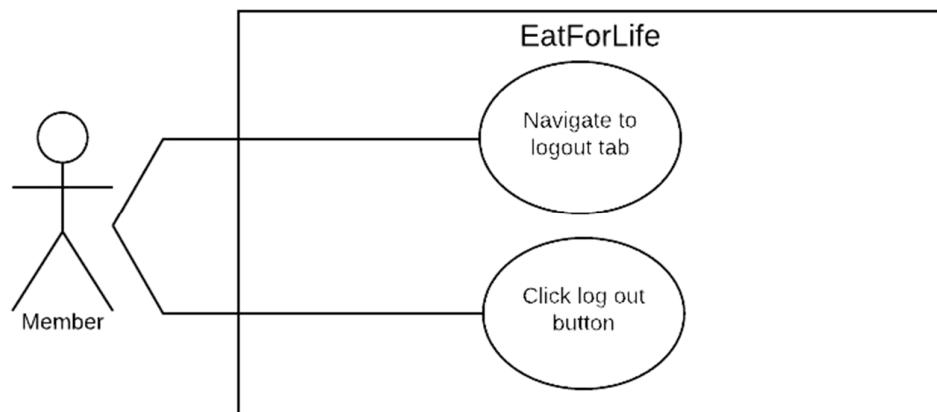


Figure 30 - Use Case Diagram for the log out event on the EatForLife Mobile Application.

## CONTEXT DIAGRAM

This diagram shows a brief overview of the data flow between the client and dietitians with the EatForLife mobile application.

In this diagram, all data input is grouped as *data input*. The interaction between the user and the clients with the application is grouped as a *data request*. The confirmation, feedback, and verification from the system to the client and dietitians are grouped as *feedback*. All data that is displayed is grouped as *data display*.

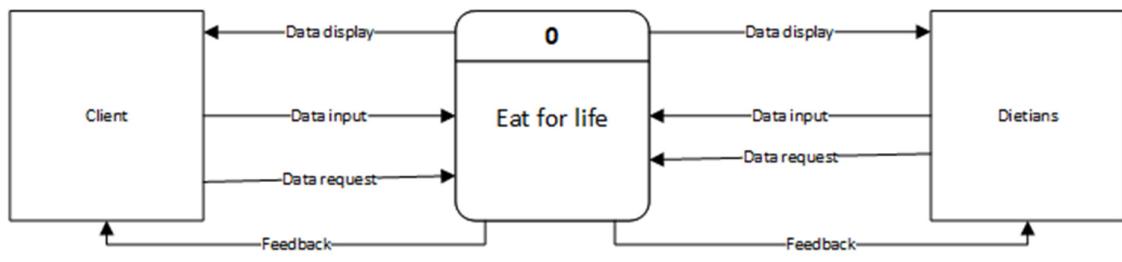


Figure 31 - Context Diagram for the EatForLife system.

## SYSTEMS LEVEL DFD

This diagram shows an overview of the data flow between the user and dietitians with each sub-system within the EatForLife mobile application.

In this diagram, all data input is grouped as *user data*, *client data*, and *dietitian data*. Confirmation and verification from the system to the client and dietitians is grouped as *feedback*.

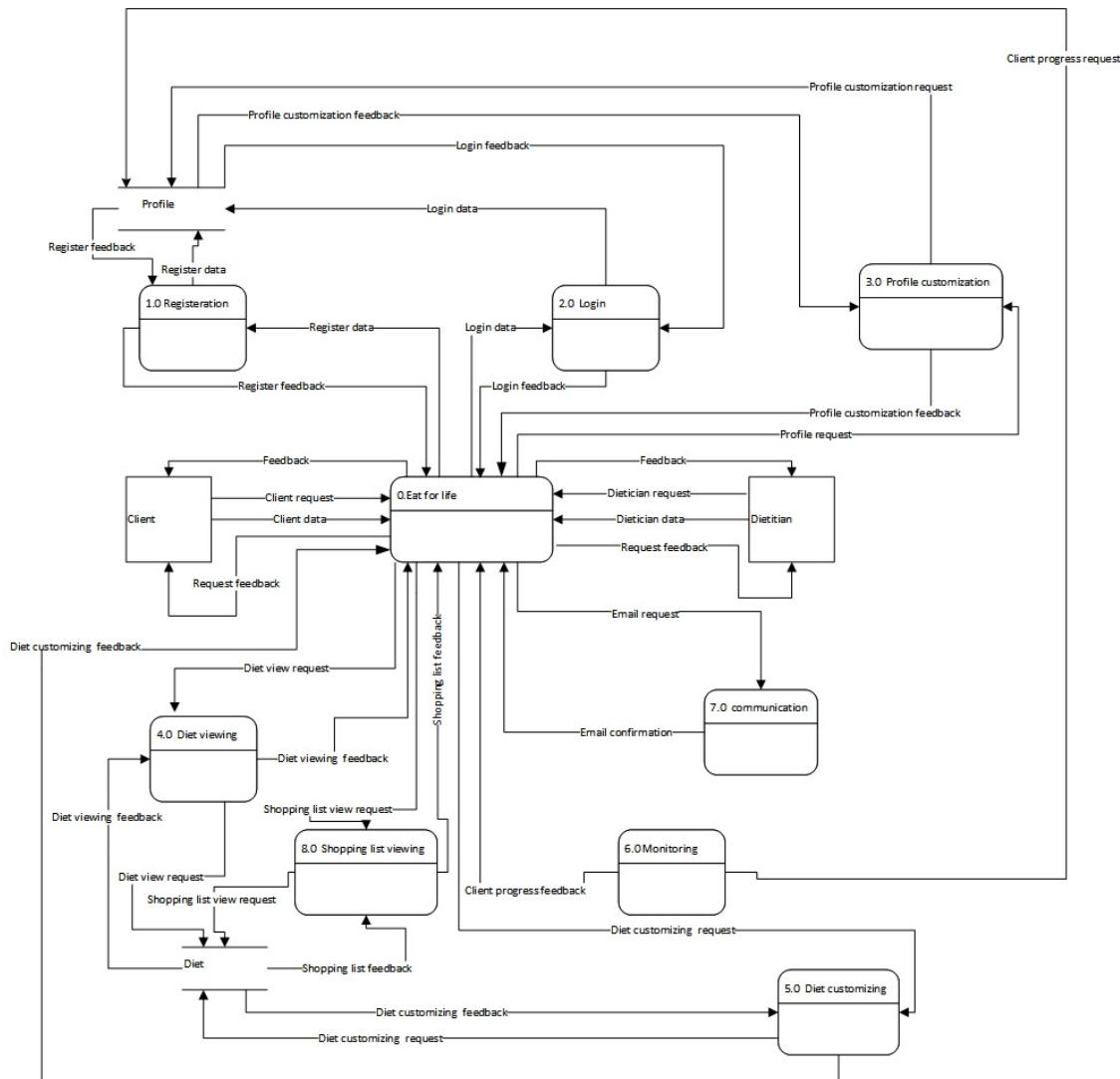


Figure 32 - Systems Level Data Flow Diagram for the EatForLife System.

## PRIMITIVE LEVEL DFD

This diagram shows all data flow between the user and dietitians and the sub-systems of the EatForLife mobile application.

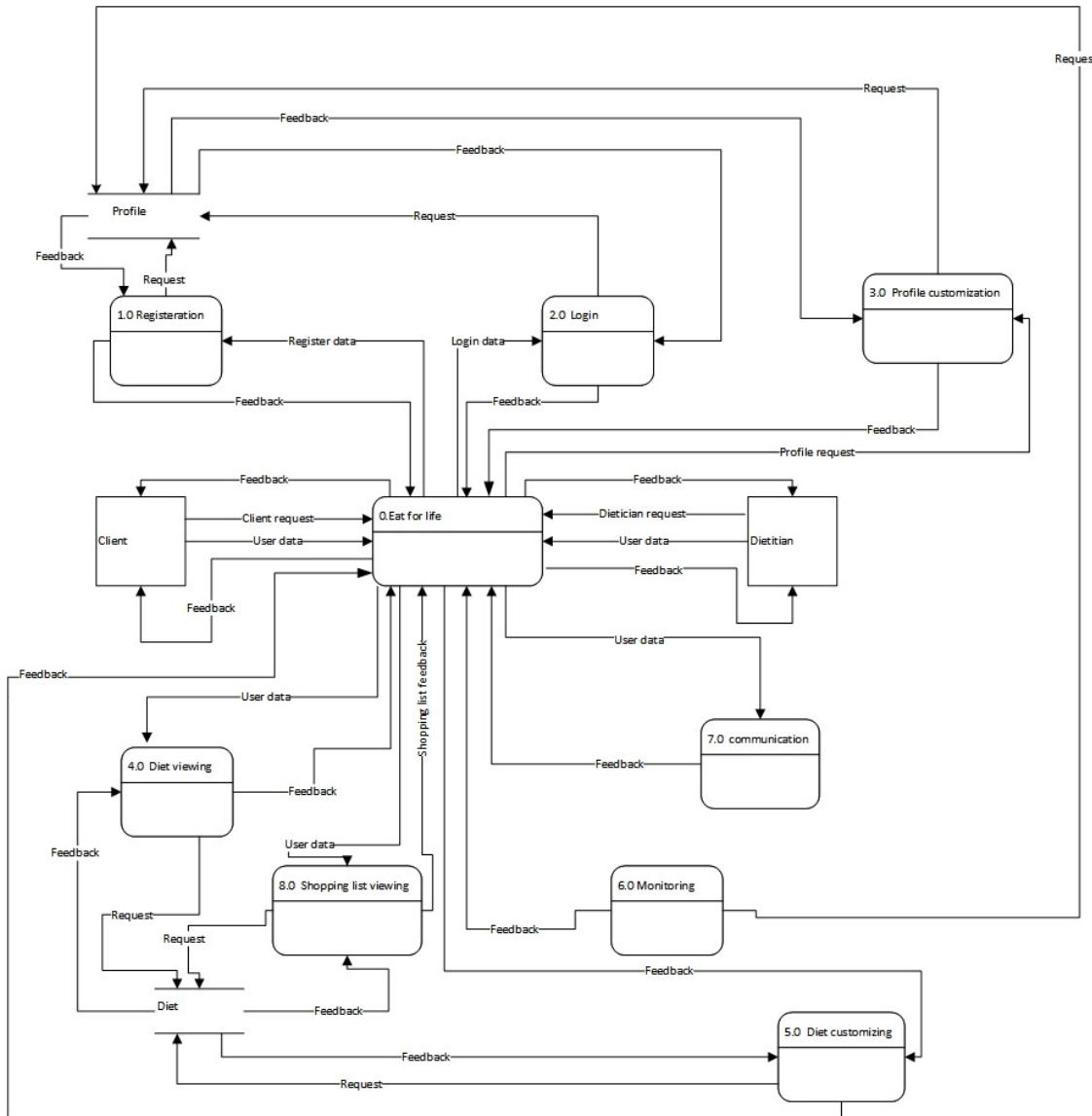


Figure 33 - Primitive Level Data Flow Diagram for the EatForLife System.

## ENTITY RELATIONSHIP DIAGRAM

An entity relationship diagram (ERD) is a logical, detailed graphical representation of all the entities, associations and data for the system (Valacich, et al., 2015). The diagram below shows the ERD for the EatForLife system, outlining all entities and their relationships.

### CONCEPTUAL MODEL

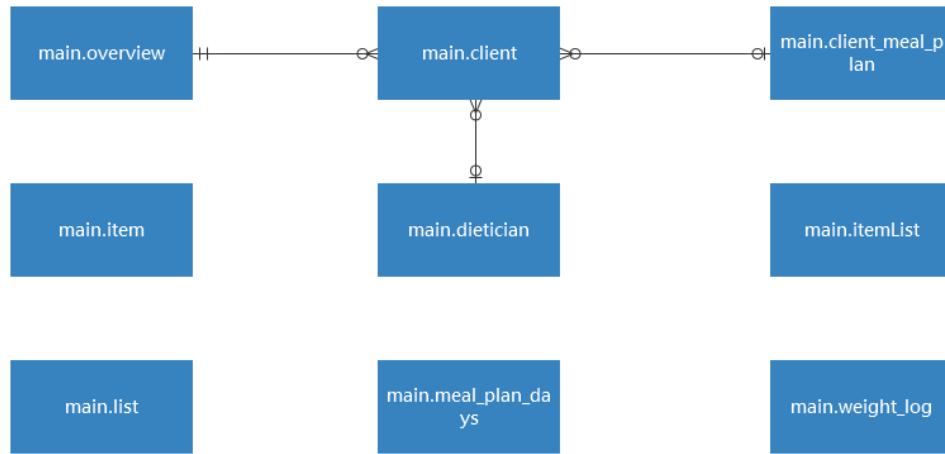


Figure 34 - Conceptual Model of the EatForLife Database.

### PHYSICAL MODEL

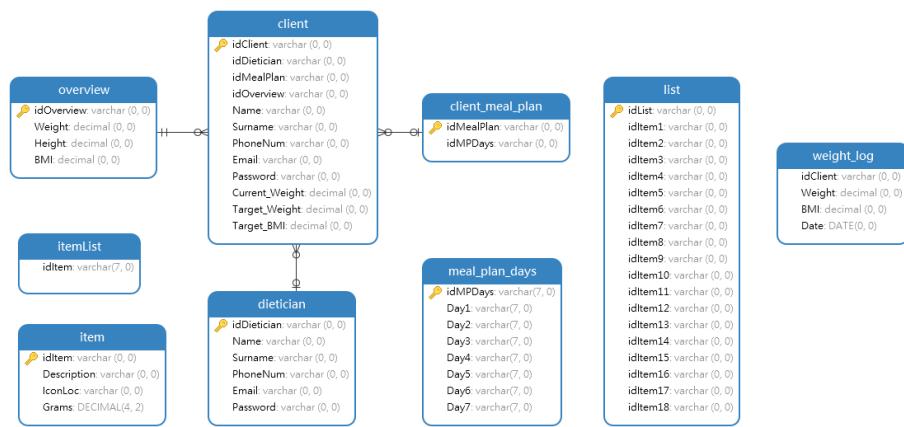


Figure 35 - Physical Model of the EatForLife Database

## LOGICAL MODEL

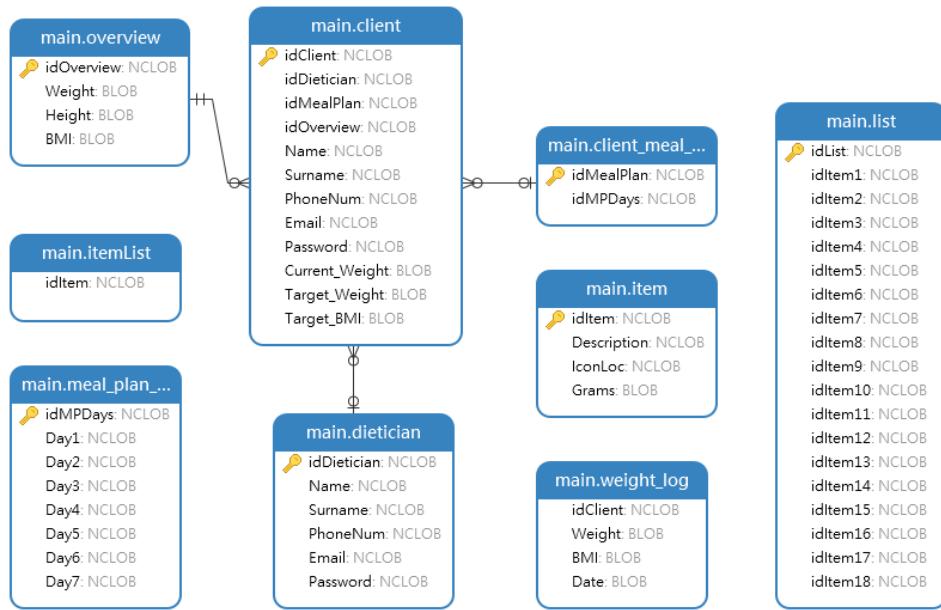


Figure 36 - Logical Model of the EatForLife Database

## PHYSICAL DESIGN

### INVESTIGATION OF TECHNOLOGIES TO BE APPLIED

All computer systems are made up of hardware and systems software that works in harmony in order to run applications programs and software, through which a development team can create new applications software (Bryant & O'Hallaron, 2011).

### *SOFTWARE DEVELOPMENT TECHNOLOGIES*

#### Hardware

*Computer hardware is a collection of the physical parts of a computer's system, including both the peripheral devices - such as a monitor, keyboard, and mouse - as well as the components within the computer such as a motherboard, hard disk drive and graphics card (Zandbergen, 2018).*

The hardware to be used by the developers to develop and test the system is:

- A PC or laptop running a Windows operation system.
- An iMac or MacBook running a Mac OSX operating system.
- A smartphone running iOS.
- A smartphone running Android OS.

The hardware to be used by the users of the system in order to run the system is:

- A smartphone running iOS, or,
- A smartphone running Android OS.

## Software

*Computer software is a set of instructions to instruct the hardware to perform specific tasks and operations for the user, the software includes both operating systems software as well as applications software (Zandbergen, 2018).*

### **NativeScript Development Environment**

NativeScript can be used on Windows, Linux and MacOS systems - we will be using Windows and MacOS to develop the EatForLife application:

The NativeScript command line interface (CLI) is required to build and run applications on both Windows and MacOS, the following software is used to run the NativeScript CLI:

- Windows
  - Chocolatey;
  - Google Chrome;
  - Node.js;
  - JDK 8;
  - Android SDK;
  - Android Studio (NativeScript, 2018).
- MacOS
  - Homebrew;
  - Node.js
  - Xcode 9;
  - Xcodeproj ruby gem;
  - CocoaPods;
  - Python ‘six’ package;
  - JDK 8 (NativeScript, 2018).

### **Integrated Development Environment (IDE)**

NativeScript can be developed in any IDE, however, a NativeScript plugin is available for Visual Studio Code, thus this is the recommended IDE for development (NativeScript, 2018). NativeScript can also be used in Android Studio and Xcode.

## **Database Management System**

The software used to manage and run the SQL database is MySQL Workbench, which is compatible with both Windows and Mac OSX (MySQL, 2018).

### **GitHub**

GitHub is a web-based hosting service that allows for version control for code repositories. All code relating to the EatForLife project will be uploaded onto a private GitHub repository only accessible to the group members.

### Programming Languages

### **NativeScript**

NativeScript is an open source transcription/translation framework that allows native mobile apps to be built for both Android and iOS using Angular, Vue.js, TypeScript or JavaScript (NativeScript, 2018). NativeScript applications are built primarily using JavaScript, or languages that can be compiled into JavaScript. NativeScript has become an increasingly popular technology since its initial release in 2014, with companies such as SAP utilising it (Haag, 2017). The mobile applications built using this framework are fully cross platform, however instead of the standard practice of wrapping applications in a web view that looks like an application in order to achieve a code-once cross platform mobile application NativeScript uses the native API's to build the mobile application (Shore, 2016). Mobile applications built in NativeScript run as if they have been programmed using Xcode (iOS) and Android Studio (Android) in their respective languages as opposed to writing a single piece of software that can be ported to both mobile operating systems with high performance levels (Shore, 2016).

Using NativeScript the mobile application's front end can be built with HTML-like syntax and styled using CSS, while the application logic is written in JavaScript or Angular (NativeScript, 2018).

## **SQL**

SQL is a standard language used in relational database programming (Sommerville, 2011). SQL allows the programmer to:

- Create the database and table structures;
- Perform data management tasks - such as insert, update, and delete;
- Perform simple and complex queries (Connolly & Begg, 2004).

## **Git**

Git is a version control language that is used in conjunction with GitHub.

## **Prototyping Software**

The initial prototype designs will be designed using MockingBot, an online wireframing tool that allows users to design an interactive user interface using drag and drop features (MockingBot, 2018).

### Documentation

## **Microsoft Word & Google Docs**

All documentation will be compiled into the final deliverable documents using both Microsoft Word and Google Docs. Google Docs enables live collaboration between the group members, while Microsoft Word allows for the document to be compiled into a final compiled formatted document for submission.

## **Microsoft Excel & Google Sheets**

Microsoft Excel and Google Sheets will be used in collaboration to create any required spreadsheets - such as the risk register.

## **Google Drive**

All documentation and files relating to the EatForLife are uploaded and stored on a Google Drive folder that is accessible to all team members from any of their devices. This acts as a central document repository.

## **Microsoft Visio & Draw.io**

Diagrams will be designed using Microsoft Visio, or the web-based equivalent, Draw.io.

## **Microsoft Project & Project Libre**

The Gantt chart for each deliverable will be created and updated using Microsoft Project, or its open source equivalent Project Libre.

### Communication Software

#### **WhatsApp**

The group will communicate using a WhatsApp group to coordinate meetings and keep updated on the project progress.

#### **Email**

All communication with the client will be conducted through email, as well as all deliverables will be sent through email to each group member to review.

#### **Discord**

Discord is a voice over internet protocol (VoIP) application that allows both text and voice chat. A Discord server has been created for group members to share any important links, documents, and have voice meetings.

#### **Google Docs**

Google Docs allows for comments to be made on documents, as well as allowing users to assign tasks through these comments. Each deliverable task will be assigned to one or more members of the team to complete.

## SYSTEM TESTING

Software application testing is the process of ensuring that the software is error free, fully functional and that all requirements are met. Testing is performed throughout the systems development life cycle, the testing process involves various activities; from creating an overall test plan in the analysis phase, to unit, integration, and system test plans during the design phase (Valacich, et al., 2015). These tests are all carried out on the system during the implementation of the system to ensure that it is fully functional and meets all defined requirements (Valacich, et al., 2015).

According to Sommerville (2011), system testing has two goals:

1. To demonstrate to both the developer and the client that the software meets the outlined requirements; and
2. To discover situations where the behaviour of the software is incorrect, erroneous, or does not perform according to its specification.

The following diagram outlines an abstract model of the software testing process:

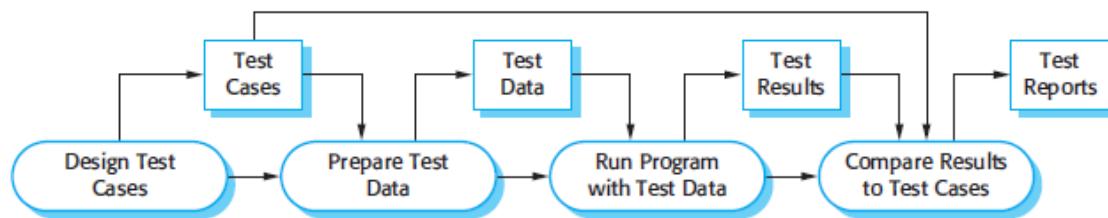


Figure 37 - Model of the software testing process. Source: (Sommerville, 2011).

## TESTING TYPES

Software application testing is an umbrella term covering various types of tests, testing can be performed with code execution and without code execution and can be manual or automated (Valacich, et al., 2015).

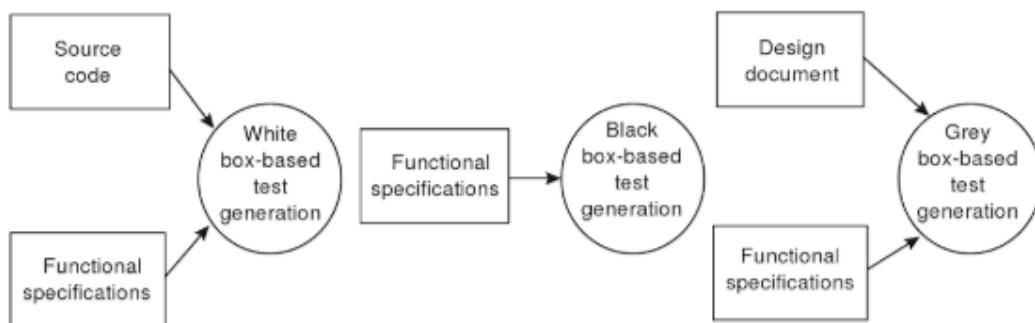
The following table outlines the different categories of testing:

*Table 16: Categories of Test Types. Source: (Valacich, et al., 2015).*

	Manual	Automated
Without code execution	Inspections	Syntax checking
With code execution	Walkthroughs Desk checking	Unit testing Integration testing System testing Stub testing

There are various types of testing that can be performed on the system, these fall under three general categories - black box testing, grey box testing and white box testing (Saleh, 2009).

The following diagram outlines the three categories of testing and what they test:



*Figure 38 - White, black, and grey box-based strategies and inputs. Source (Saleh, 2009).*

The following will provide a brief outline of the main types of testing used in software development.

### **Inspection**

Inspection is a testing technique where a software tester manually examines the code for any well-known errors that may not be picked up by automated system inspection software (Valacich, et al., 2015).

### **Walkthroughs**

A walkthrough checks what the code does and is done to detect errors in the code - these are often informal and should be done often to check the code for any errors in order to correct them (Valacich, et al., 2015).

### **Desk checking**

Desk checking is the process in which the code is manually sequentially executed by the software tester - this is an informal process done with a pen and paper, where the logic of the system is tested step by step (Valacich, et al., 2015).

### **Syntax checking**

Syntax checking ensures that all syntax is correct and error free, this is typically performed by the compiler (Valacich, et al., 2015).

### **Unit testing**

Unit testing is the process whereby each individual unit - i.e. a set of code that performs a single function - is tested alone to discover any errors in that unit's code (Valacich, et al., 2015).

### **Integration testing**

This is the process of compiling units together and testing their integration together for any errors when running together (Valacich, et al., 2015; Stair & Reynolds, 2016).

## System testing

System testing is the process of testing all components of the system in an integrated complete system to determine that all requirements are (Valacich, et al., 2015; Stair & Reynolds, 2016).

## Volume testing

Volume testing is the process of testing the system performance under various loads of work volumes to determine the volume at which systems performance degrades (Stair & Reynolds, 2016).

## User acceptance testing

This is the process whereby the users who will be using the system test the completed integrated system to verify that the system meets requirements and is satisfactory to their needs (Stair & Reynolds, 2016).

The following table outlines a selection of the tests that can be conducted on an information system:

*Table 17 - Tests conducted on an information system. Source: (Stair & Reynolds, 2016).*

Form of Test	What Is Tested	Purpose of Test	Who Does It
User Acceptance	Test the complete, integrated system (hardware, software, databases, people, and procedures).	Verify the information system can complete required tasks in a real-world operating environment and do this according to the system design specifications.	Trained users of the system
Volume	Evaluate the performance of the information system under realistic and varying work volume and operating conditions.	Determine the work load at which systems performance begins to degrade and identify and eliminate any issues that prevent the system from reaching its required service-level performance.	System development team and members of the operations organization
System	Test the complete, integrated system (hardware, software, databases, people, and procedures).	Validate that the information system meets all specified requirements.	Independent test team separate from the software development team
Integration	Test all of the individual units of the information system linked together.	Uncover any defects between individual components of the information system.	Software developers or independent software testers using black box testing measures
Unit	Test individual units of the system.	Verify that each unit performs as designed.	Software developers

## TEST PLAN

A test plan is a collection of written documents that outline the various tests to be performed on the system, allowing for improved communication between all members involved in testing the software (Valacich, et al., 2015). The test plan specifies the role's that each team member will fulfil, as well as serving as a checklist to ensure that all steps in the testing process have been completed (Valacich, et al., 2015).

*Table 18 - Test Plan for the EatForLife System.*

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
1	Prototype user interfaces	Functional Testing			
1	Prototype user interfaces	Usability Testing			
1	Prototype user interfaces	User Acceptance Testing			
1	Prototype user interfaces	Unit Testing			
2	Creation of database	Unit Testing			
2	Linking of the database to the application	Unit Testing			
2	Database functions	Functional Testing			
3	Registration page	Functional Testing			
3	User registration	Unit testing			
3	User login	Unit testing			
3	Dietician login	Unit testing			
4	Profile Page	Functional testing			
4	Update personal details	Unit testing			
4	Update progress and measurements	Unit testing			
4	Generate graphs	Unit testing			

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
4	Generate reports	Unit testing			
4	Update goals	Unit testing			
5	Diet plan	Functional testing			
5	Generate weekly meal plan	Unit testing			
5	Display information on each food item on the meal plan	Unit testing			
5	Checklist functionality	Unit testing			
5	Display alternative substitutes for foods	Unit testing			
5	Download meal plan	Unit testing			
5	Email meal plan	Unit testing			
5	Generate shopping list	Unit testing			
5	Download shopping list	Unit testing			
5	Email shopping list	Unit testing			
6	Communications	Functional testing			
6	Communication between user and dietician	Unit testing			
6	Contact us page	Functional testing			
6	Users are able to contact the development team	Unit testing			
6	Users are able to contact EatForLife	Unit testing			
7	Admin page	Functional testing			
7	Admin users are able to access the admin page	Unit testing			

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
7	Admin users are able to add a new admin user	Unit testing			
-	System integration	Integration testing			
-	System integration	System testing			
-	Customer handover - user acceptance	User acceptance testing			

## TESTING TEMPLATES

A testing template is a document that outlines each individual requirement to be tested, and whether or not the functionality is acceptable. The test template provides a guide for the development team to test all requirements, as well as serving as evidence that the system has been tested before customer handover.

The functionality of each requirement will be scored using a Likert-type scale; this allows the tester to select a broader, more descriptive response to measure the system functionality (Krosnick & Presser, 2010). The Likert scale model allows users to indicate a level of disagreement or agreement - usually on a 5-point scale ranging from:

1. Strongly disagree
2. Disagree
3. Neutral
4. Agree
5. Strongly agree (Krosnick & Presser, 2010).

The scale used to measure the functionality of each requirement in our test template is adapted from the Likert scale into a similar style, based on a 7-point level of acceptability scale:

1. Totally unacceptable
2. Unacceptable
3. Slightly unacceptable
4. Neutral
5. Slightly acceptable
6. Acceptable
7. Perfectly acceptable (Vagias, 2006).

The tester can rate each requirements functionality acceptability using this scale on the test template.

## HOME PAGE

The home page is split into two components dependant on if the user is logged in or not. If the user is not logged in they will be presented with an unlogged-in home page that gives them the option to log in or register for the EatForLife program. If the user is logged in their home page will show their details and progress.

*Table 19 - Test Template for the EatForLife Home Page*

Page: Home Page (User Not Logged In)	
<b>Description:</b> This is the opening page of the EatForLife application, allowing users to either Login or Register for an account.	<b>Version:</b> _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b> _____ _____ _____	

*Table 20 - Test Template for the EatForLife Home Page*

Test ID	Requirement <b>Home Page (User Not Logged In)</b>	Acceptability	Comments
		(1-7)	
HM001	The user is able to view the Home Page.		
HM002	The user is able to view and select the Login button.		
HM003	The user is able to view and select the Register button.		

## REGISTRATION PAGE

The registration page allows users to create a new account with EatForLife.

*Table 21 - Test Template for the EatForLife Registration Page*

Page: Registration Page	
<b>Description:</b> The system allows users to enter their details to create an account	<b>Version:</b> _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b> _____ _____	

*Table 22 - Test Template for the EatForLife Registration Page*

Test ID	Requirement <b>Registration</b>	Acceptability (1-7)		Comments
RG001	The user is able to view the Registration Page.			
RG002	The user is able to access navigation options (return to Home Page) from the Registration Page.			
RG003	The user is able to view and enter their details.			
RG004	The user is able to view and select buttons.			
RG005	The user is given feedback if their submission was successful once the Submit button has been selected.			
RG006	Data validation is performed on all fields.			
RG007	The user is redirected to the login screen once they have successfully registered for a new account.			

## LOGIN PAGE

The login page allows a user - both dietician and customer - log into their existing EatForLife account.

*Table 23 - Test Template for the EatForLife Login Page*

Page: Login Page	
<b>Description:</b> The system allows a user to login to an existing account	<b>Version:</b> _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b> _____ _____ _____	

*Table 24 - Test Template for the EatForLife Login Page*

Test ID	Requirement	Acceptability (1-7)	Comments	Login Page
LG001	The user is able to view the login page.			
LG002	The user is able to access navigation options (return to Home Page) from the Login Page.			
LG003	The user is able to view and enter their details.			
LG004	The user is able to view and select buttons.			
LG005	Data validation is performed on all fields.			
LG006	The user is given feedback if their submission was successful once the Submit button has been selected.			
LG007	The user is redirected to their Home Page if successfully logged in.			

## HOME PAGE

The home page displays a summary of the user's details and statistics.

*Table 25 - Test Template for the EatForLife Home Page*

Page: Home Page (User Logged In)	
<b>Description:</b> Once a user is successfully logged in, their home page provides a quick overview of their information, weight loss statistics, and a weight loss graph.	<b>Version:</b> _____ _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____ _____
<b>Signature:</b> _____	<b>Date:</b> _____ _____
<b>Additional Comments:</b> <hr/> <hr/> <hr/>	

*Table 26 - Test Template for the EatForLife Home Page*

Test ID	Requirement	Acceptability (1-7) Home Page (User Logged In)	Comments
HM004	The user is able to view the Home Page.		
HM005	The user is able to view their Member Details.		
HM006	The user is able to view their statistics.		
HM007	The user is able to view a graph of their weight loss.		

## MEAL PLAN PAGE

The meal plan page displays a seven-day meal plan for the user based on their diet preferences.

*Table 27 - Test Template for the EatForLife Meal Plan Page*

Page: Meal Plan Page	
<b>Description:</b> This page allows a user to view their seven-day meal plan, as well as edit any foods on their list, reset their meal plan, find more information on their foods, print or email the meal plan.	<b>Version:</b> _____ _____
<b>Type of Test:</b> _____ <b>Signature:</b> _____	<b>Tested By:</b> _____ <b>Date:</b> _____ _____
<b>Additional Comments:</b> _____ _____	

*Table 28 - Test Template for the EatForLife Meal Plan Page*

Test ID	Requirement Meal Plan Page	Acceptability (1-7)	Comments
MP001	The user is able to view the Meal Plan Page.		
MP002	The user is able to select which day of their meal plan they would like to view.		
MP003	The user is able to view items on their meal plan.		
MP005	The user is able to change a food to a replacement food by clicking the Edit Food button.		
MP006	The user is able to view the foods separated by different meal times.		
MP007	The user is able to reset their food choices back to the default - undoing any substitutions - by clicking the Reset Diet button.		
MP008	The user is able to view and print a printable version of their meal plan by clicking the Print button.		
MP009	The user is able to view and receive an email version of their meal plan by clicking the Email button.		

## SHOPPING LIST PAGE

The shopping list page allows a user to see what items they will need to buy for the next seven days for their generated meal plans.

*Table 29 - Test Template for the EatForLife Shopping List Page*

Page: Shopping List Page	
<b>Description:</b> This page allows a user to view all items they will need to purchase for their weekly meal plan, as well as allowing users to print or email the shopping list.	<b>Version:</b> _____ _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b>  _____ _____ _____	

*Table 30 - Test Template for the EatForLife Shopping List Page*

Test ID	Requirement  <b>Shopping List Page</b>	Acceptability (1-7)	Comments
<b>SL001</b>	The user is able to view the Shopping List page.		
<b>SL002</b>	The user is able to view all items and their measurements on the Shopping List.		
<b>SL003</b>	The user is able to view and print a printable version of their shopping list by clicking the Print button.		
<b>SL004</b>	The user is able to view and receive an email version of their shopping list by clicking the Email button.		
<b>SL005</b>	The shopping list is automatically updated when a user changes a food in their meal plan.		

## PROFILE PAGE

The user's profile page allows the user to view their details stored on the EatForLife system, update their measurements, update their personal details, update their health details, as well as change their email and password. The user is able to view their weight logs from the profile page.

*Table 31 - Test Template for the EatForLife Profile Page*

Page: Profile Page	
<b>Description:</b> The user is able to view and update all of their details from their profile page, as well as view measurement logs from the profile page.	<b>Version:</b> _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b> _____ _____ _____	

*Table 32 - Test Template for the EatForLife Profile Page*

Test ID	Requirement <b>Profile Page</b>	Acceptability (1-7)	Comments
PR001	The user is able to view the Profile Page.		
PR002	The user is able to view their personal details.		
PR003	The user is able to view and update their weight loss details.		
PR004	The user is able to view their historical measurement logs.		
PR005	The user is able to log out of the application.		

..

## COMMUNICATION PAGE

The communication page allows for the user to communicate with their dietitian directly, allowing for any queries or comments to be sent to their dietitian.

*Table 33 - Test Template for the EatForLife Communication Page*

Page: Communication Page	
<b>Description:</b> The user is able to directly contact their dietitian through the application.	<b>Version:</b> _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b> _____ _____ _____	

*Table 34 - Test Template for the EatForLife Communication Page*

Test ID	Requirement	Acceptability (1-7)	Comments
		Communication Page	
CM001	The user is able to view the Communication Page.		
CM002	The user is able to view and enter their details.		
CM003	The user is able to view and select buttons.		
CM004	Data validation is performed on all fields.		
CM005	The user is given feedback if their submission was successful once the Submit button has been selected.		
CM006	The user is able to send messages through the application.		

## ADMIN PAGE

The admin page allows admin users to view all EatForLife clients and their details, as well as allowing the admin user to create a new admin user.

*Table 35 - Test Template for the EatForLife Admin Page*

Page: Admin Page	
<b>Description:</b> The system allows admin users to view all EatForLife clients and their details, as well as allowing for new admin users to be created.	<b>Version:</b> _____
<b>Type of Test:</b> _____	<b>Tested By:</b> _____
<b>Signature:</b> _____	<b>Date:</b> _____
<b>Additional Comments:</b> _____ _____	

*Table 36 - Test Template for the EatForLife Admin Page*

Test ID	Requirement  Admin	Acceptability	Comments
		(1-7)	
AP001	The admin user is able to view the Admin Page.		
AP002	The admin user is redirected to the admin page when logging in with their admin credentials		
AP003	The user is able to view a list of all clients.		
AP004	The user is able to click on a client name and be redirected to the user detail page.		
AP005	The user is able to view the client details.		
AP006	The user is able to view and select buttons.		
AP007	The user is able to select the “Add Admin User” button and be redirected to a new user form.		
AP008	The user is able to view and enter the details for a new admin user.		
AP009	The user is able to view and select the “Submit” button.		
AP010	Data validation is performed on all fields.		
AP011	The user is given feedback if their submission was successful once the Submit button has been selected.		
AP012	The user is able to log out.		

## UPDATED SCHEDULE

### GANTT CHART FOR DELIVERABLE 3

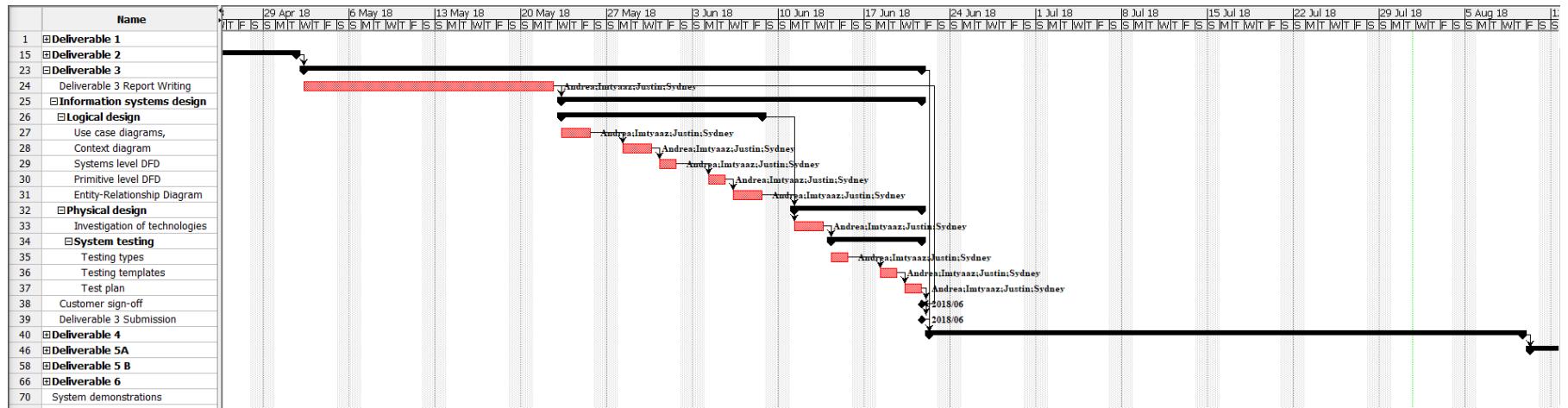


Figure 39 - Summarised Gantt Chart for the EatForLife Project.

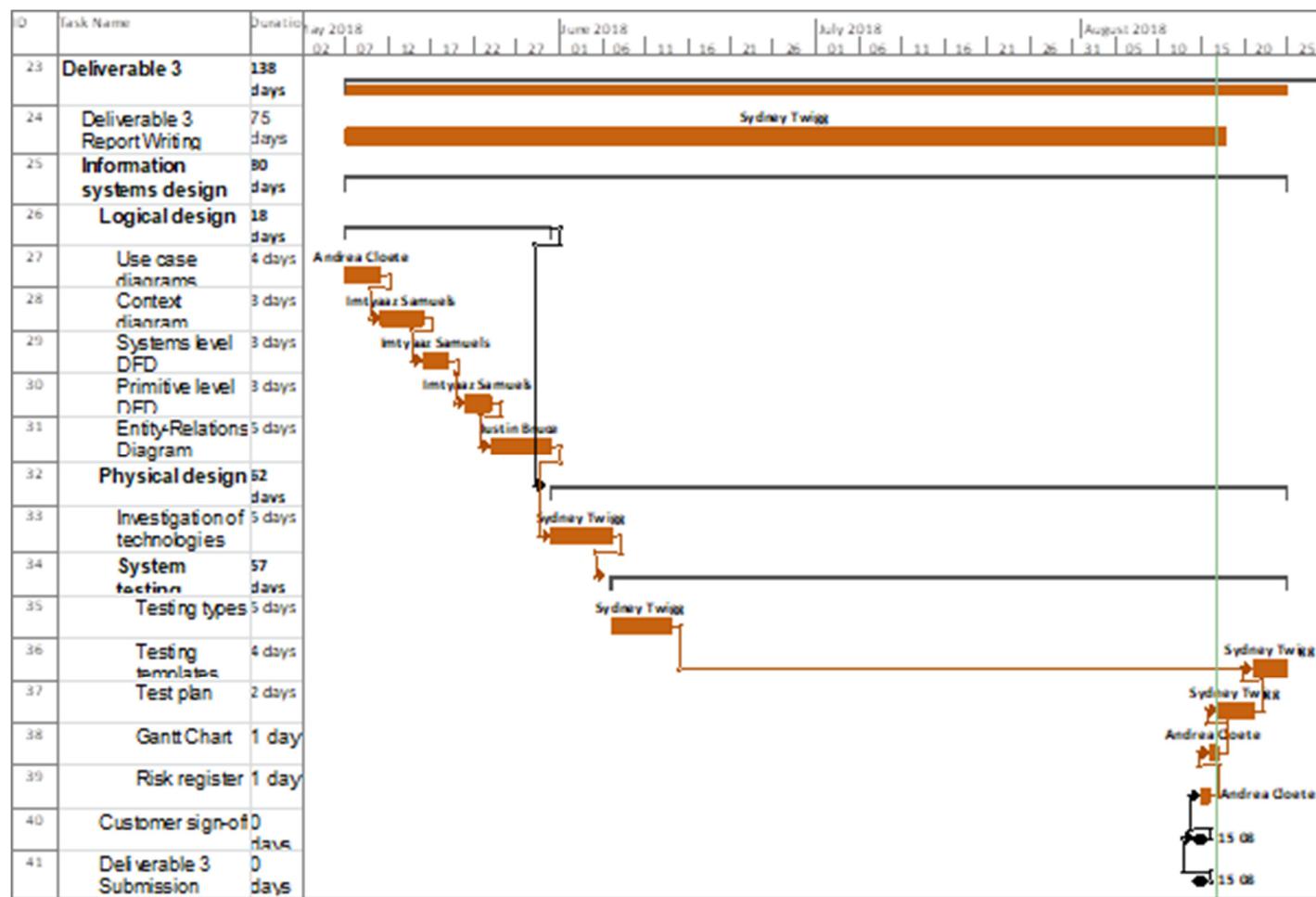


Figure 40 - Detailed Gantt Chart for Deliverable 3.

## RISK REGISTER

*Table 37 - Risk Register for the EatForLife application development.*

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R06	1	Cross-over incompatibility	The application may work well on Android but not on iOS	Technology Risk	The chosen coding platform does not support the needed language conversion	The chosen platform is incorrect for the development of both Android and iOS	Focus on the functional platform to deliver the best full working project, the other platform can be integrated further down the line.	DOT Development	Medium	High
R02	2	Delay of required client information	The client takes a long time to respond to the group's questions via email or does not answer calls consistently	People Risk	The client does not regularly read their email, is too busy to reply or answer the phone or is impeded by another variable	The client isn't as invested as they should be in developing their system	Try complete information needed through other research methods - online etc.	DOT Development	Low	Low
R03	3	Uncertainty of full client requirements	The client gives the team a vague list of what he/she wants the system to be and	People Risk	The team does not aggressively ask for the requirements	Identified during meetings with the client	Send the client a system proposal with requirement	DOT Development	Medium	Medium

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
			what it should entail				ideas for their feedback			
R11	4	Update of requirements	The client decides to change requirements for the system mid-development	People Risk	The client no longer wants to have specific features in the system or wants to add additional features to the system	The client's business develops new requirements for their future success	The team will ensure that a meeting takes place in order to document the new requirements appropriately	DOT Development	Medium	Medium
R04	5	Bad communication skills between team members	Team members do not communicate appropriately and/or frequently enough	People Risk	Due to team members not owning a phone, not using discord (our team's communication application) or google drive	Team members don't know how to work well with each other	Discuss it with the person, if there is no improvement escalate the issue with your lecturer.	DOT Development	Low	High
R05	6	Resignation of a team member	Team member decides to resign mid-way during the development process	People Risk	Unexpected turn of events or too much stress for the team member to handle	The team member can no longer take the pressure of developing the system	The team will have to accept the decision for the resignation of the member and continue development	DOT Development	Low	High
R01	7	Loss of business	The client decides to no	Market Risk	The client has lost interest in the system or	The client's business no longer needs	Find a new client, or if too late, continue	Client	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
			longer work with our team		no longer wants to invest the time	a new system to further their success	without the client			
R07	8	Fail to choose a suitable coding language	The team does not choose a suitable coding language the is required for the chosen/designed system	People Risk	Team members may be unsure for which language(s) best suit their skills and the requirements for development	There are multiple languages to choose from and team members have their own unique skills in their respective languages	Use a familiar well documented language	DOT Development	Low	High
R12	9	Hardware failure	The developers have an unforeseen hardware failure, resulting in critical data loss.	Technology Risk	Poor system maintenance, faulty computer components, untrustworthy software installed on the system	Team members did not ensure their hardware was functional and that critical data was backed up	Ensure that all critical data is backed up through Google Drive, GitHub and other cloud storage, or offsite storage.	DOT Development	Medium	High
R10	10	Quality of the system is not satisfactory for the client	The client decides that they do not like the developed system and no	Market Risk	The developed system is not up the client's standards nor meets their expectations	The developed system is invalid for the use of the client	As a group ensure that all requirements are met. If the client does not like	Client	Medium	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
			longer want to use it				system that meets the requirements, terminate the project.		High	Very High
R08	11	Chosen IDE is inappropriate for development	The chosen IDE may not be compatible with required plugins or not contain much needed tools for development	Technology Risk	Team members do not know of various IDE's that may be suitable for the development of the system	The chosen IDE does not support the needed tools for the input of needed plugins or modules	Find a new IDE that suits the project	DOT Development	Low	Medium
R09	12	Poor time management	Increments are put together late	People Risk	Due to team members delaying input due to incompetence or personal issues	Team members do not know how to schedule their time well enough to get work done	Discuss with the individual why their work is not timely, assist them if needed to finish the deliverables in schedule. If the team members do not improve their time management, escalate it to the lecturer.	DOT Development	High	Very High

## SYSTEM INTERFACE DESIGN

System interface design is a vitally important factor in systems design, as this is the face of what users will see of the system and how they interact with the system, thus it must make a good impression on the user (Pressman & Maxim, 2015). In order to create the user interface for the EatForLife application, our group followed the User Interface Design Process as outlined by Pressman & Maxim (2015):

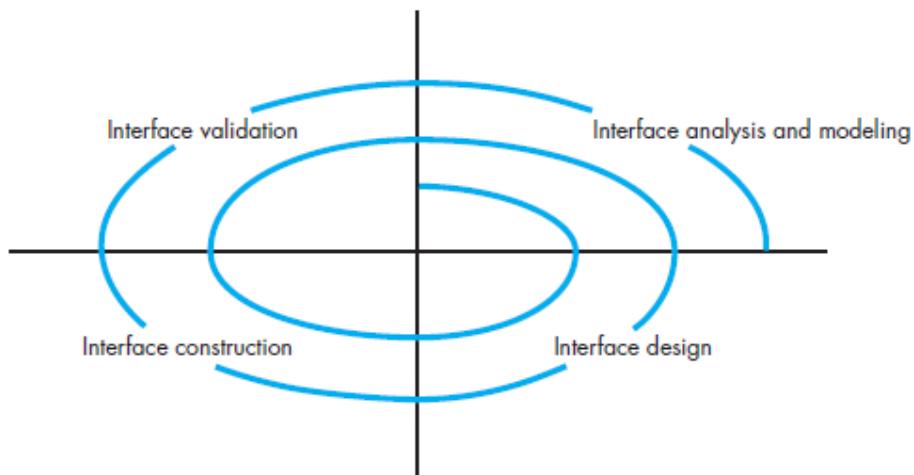


Figure 41 - User Interface Design Process. Source: (Pressman & Maxim, 2015).

The User Interface Design Process outlines an incremental model whereby the interface designers incrementally design models based on interface analysis and modelling, designing the interface, programming the interface and validating the user interface with user acceptance testing (Pressman & Maxim, 2015).

In order to design the interface, the designers need to perform interface analysis. The key point of interface analysis is to *understand the problem before you attempt to design a solution* - this means to understand the end users who will interact with the system, the tasks the end users will perform on the system, the content that is presented to the users, and the environment in which these tasks are to be conducted (Pressman & Maxim, 2015).

Our group used existing interfaces from the EatForLife desktop application to determine the tasks and content that will be present on the system, as well as research into various health and wellness mobile applications to determine the design principles to be used (So, 2017).

The EatForLife application will have two user interfaces, dependant on if the user is a dietitian or a client.

The following table outlines the views that each user interface will have:

*Table 38 - User Interface Screens for the EatForLife Mobile Application.*

Main Screen	Sub-Screen	User
<b>Landing Page</b>	Landing Page	Dietician and Client
	Login Page	Dietician and Client
	Register Page	Client
<b>Client Interface</b>		
<b>Home Page</b>	Home Page	Client
<b>Meal Plan</b>	Meal Plan Overview	Client
	Meal Plan Day View	Client
	Edit Food Item	Client
	Share/Print Meal Plan	Client
<b>Shopping List</b>	Shopping List	Client
	Share/Print Shopping List	Client
<b>Profile</b>	Profile Overview	Client
	Edit Profile	Client
	Statistics	Client
	Measurement Logs	Client
	Add New Measurements	Client
<b>Contact</b>	Send Message	Client
<b>Dietitian Interface</b>		
<b>Client List</b>	Client List	Dietitian
	Client Details	Dietitian
<b>Add New Admin</b>	Add Admin Details	Dietitian

## INCREMENT ONE: MOCKUP DESIGNS

The first increment of interface design involved designing an interface mock-up of what the final design will look like using CASE tools. Using these mock-ups, the group could base their programmed interfaces on the designs to ensure consistency. Pressman & Maxim (2015) state that consistency in the user interface is one of the golden rules of interface design, maintaining that all visual information should be the same on each page to ensure the best user experience.

### LANDING PAGE

When a user opens the application for the first time, this is the screen they will be shown. It gives them the option of logging into an existing EatForLife account or creating a new EatForLife account.

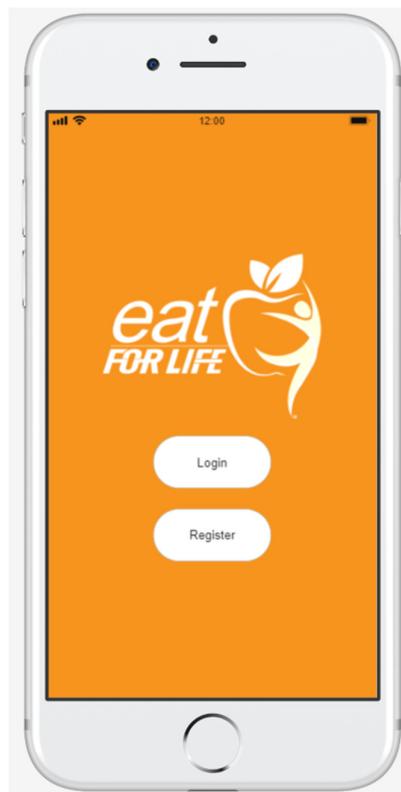


Figure 42 - Mock-up Design for the EatForLife Application Landing Page.

## LOGIN PAGE

This page allows users to input their email and password linked to their EatForLife account allowing them to login. If the user has forgotten their password, they can reset it using the forgot password feature.

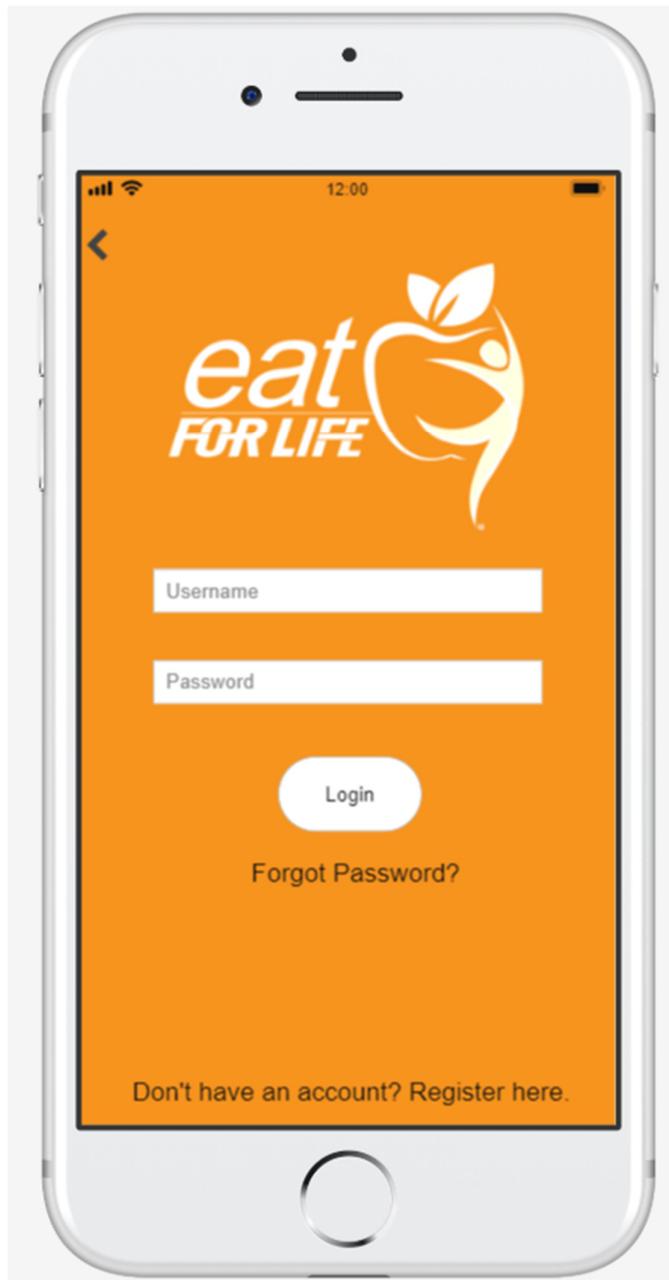


Figure 43- Mock-up Design for the EatForLife Application Login Page.

## REGISTRATION PAGE

The registration page allows users to enter their details to create a new EatForLife account, from here the registration will be sent to the EatForLife administrators to approve the account.

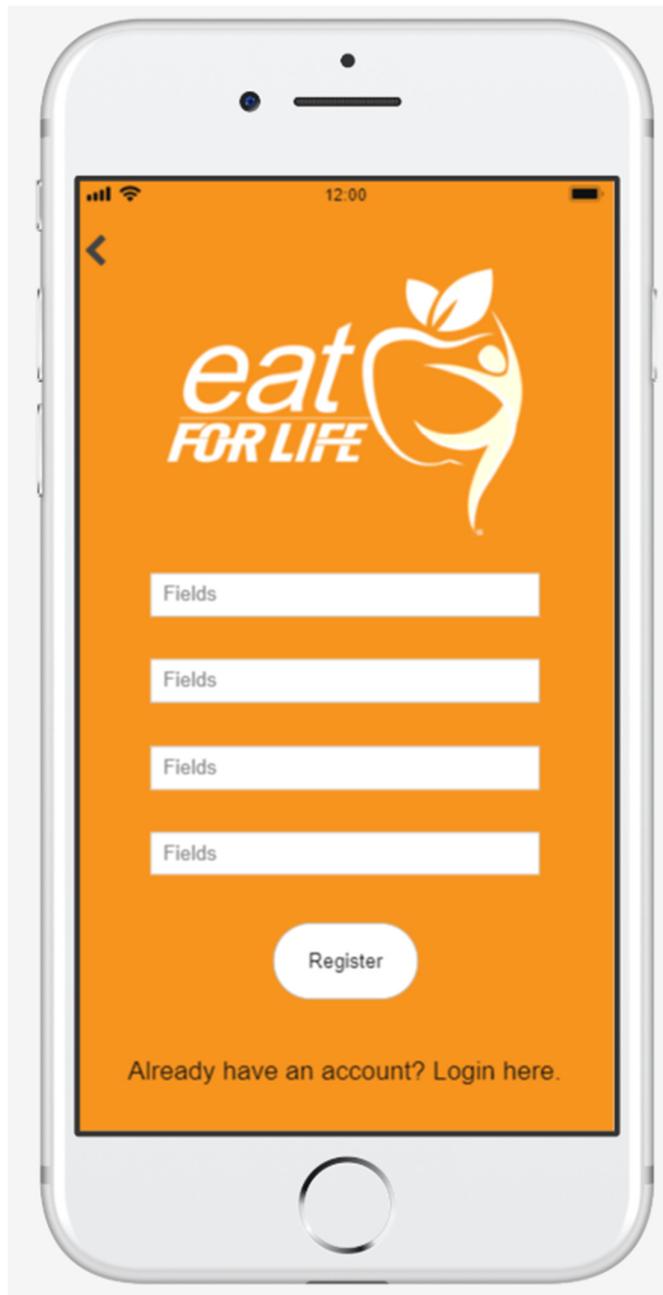


Figure 44- Mock-up Design for the EatForLife Application Registration Page.

## CLIENT INTERFACE

The following screenshots show the interface that is displayed when a client account logs into the application.

### HOME PAGE

The home page displays an overview of the client's weight loss progress.

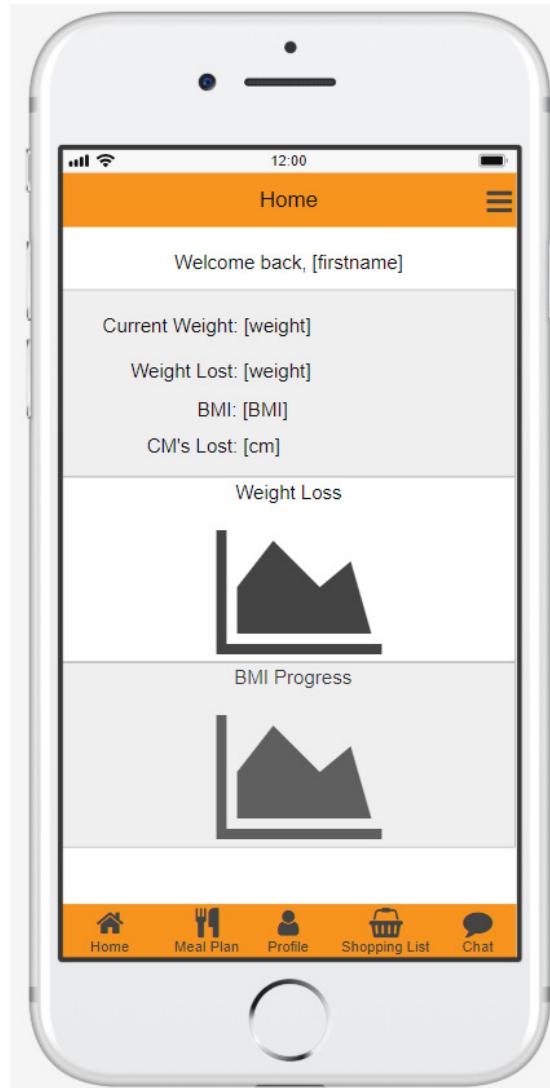


Figure 45- Mock-up Design for the EatForLife Application Home Page.

## MEAL PLAN OVERVIEW

The meal plan page allows the client to select which day of their meal plan they would like to view, as well as offering them options to print and email the full 7-day meal plan.

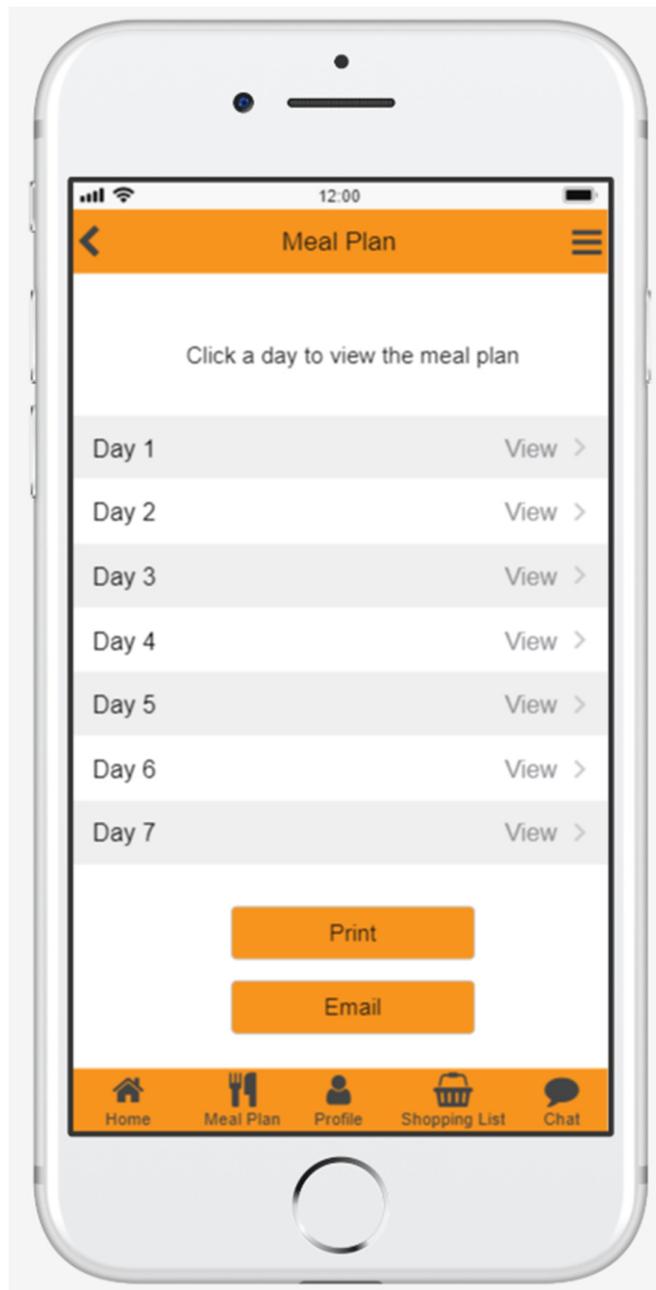


Figure 46- Mock-up Design for the EatForLife Application Meal Plan Overview Page.

## MEAL PLAN DAY VIEW

The meal plan day view page allows the user to view their daily meal plan, with options to edit the food and add a substitute and reset back to default. The meal plan is split into meal times, and users are able to check off which foods they have had so far using a completion checkbox.

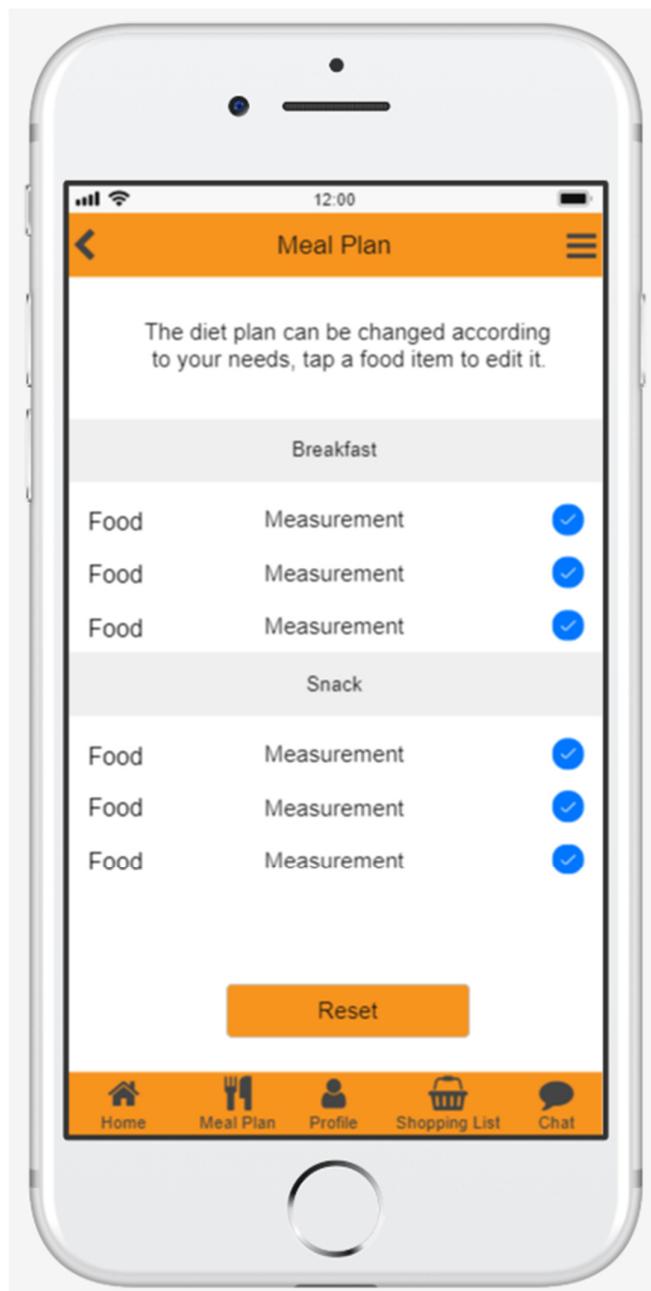


Figure 47- Mock-up Design for the EatForLife Application Meal Plan Day View Page.

## SHOPPING LIST

The shopping list page shows users a shopping list that corresponds with their 7-day meal plan, allowing them to see all the foods they will need for the week. Users are able to print and email the shopping list to themselves. Users are able to check off which foods they have using the completion checkbox.

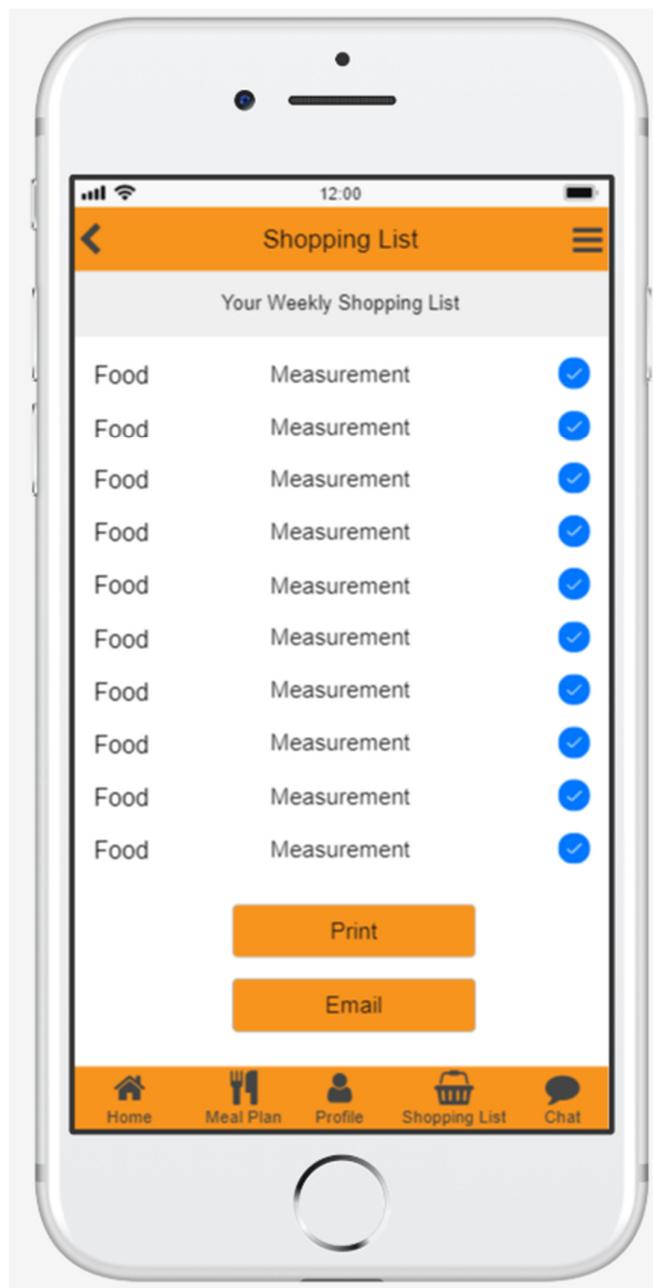


Figure 48 - Mock-up Design for the EatForLife Application Shopping List Page.

## PROFILE OVERVIEW

The user is able to view their information, measurements and total progress on the profile page. The user is also able to edit their details and add new measurements from their profile page.

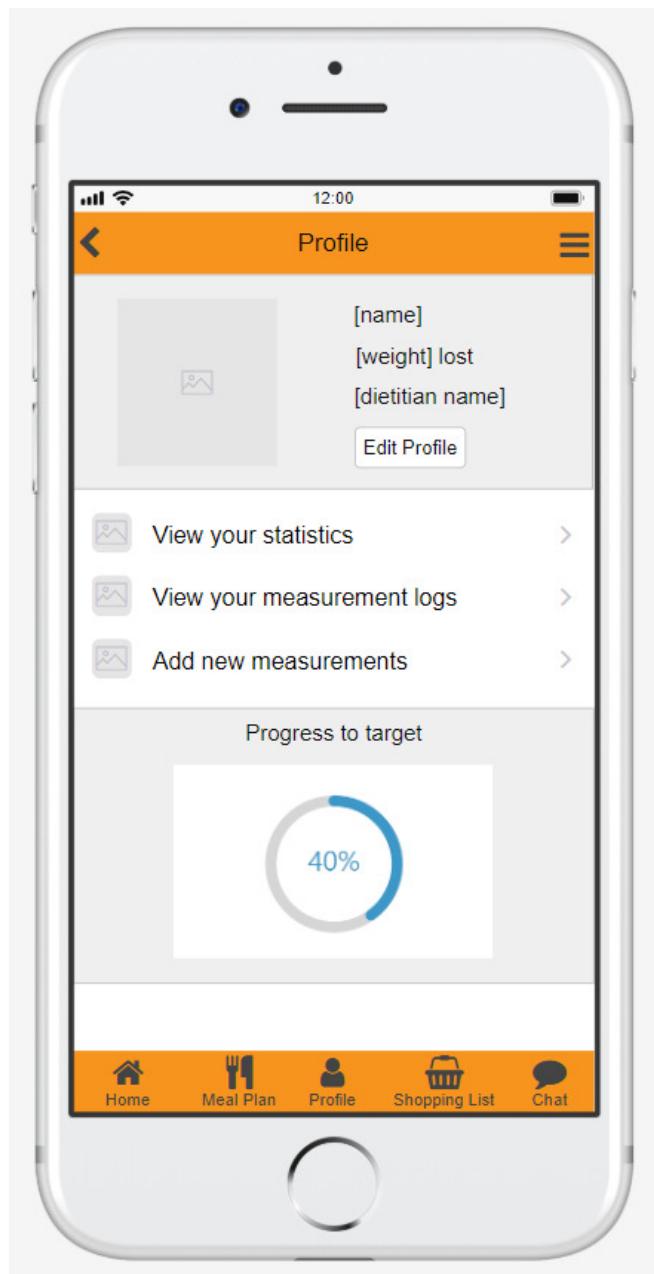


Figure 49 - Mock-up Design for the EatForLife Application Profile Page.

## EDIT PROFILE

The user is able to edit their personal details and password using the edit profile page. This page is accessed through the profile page.

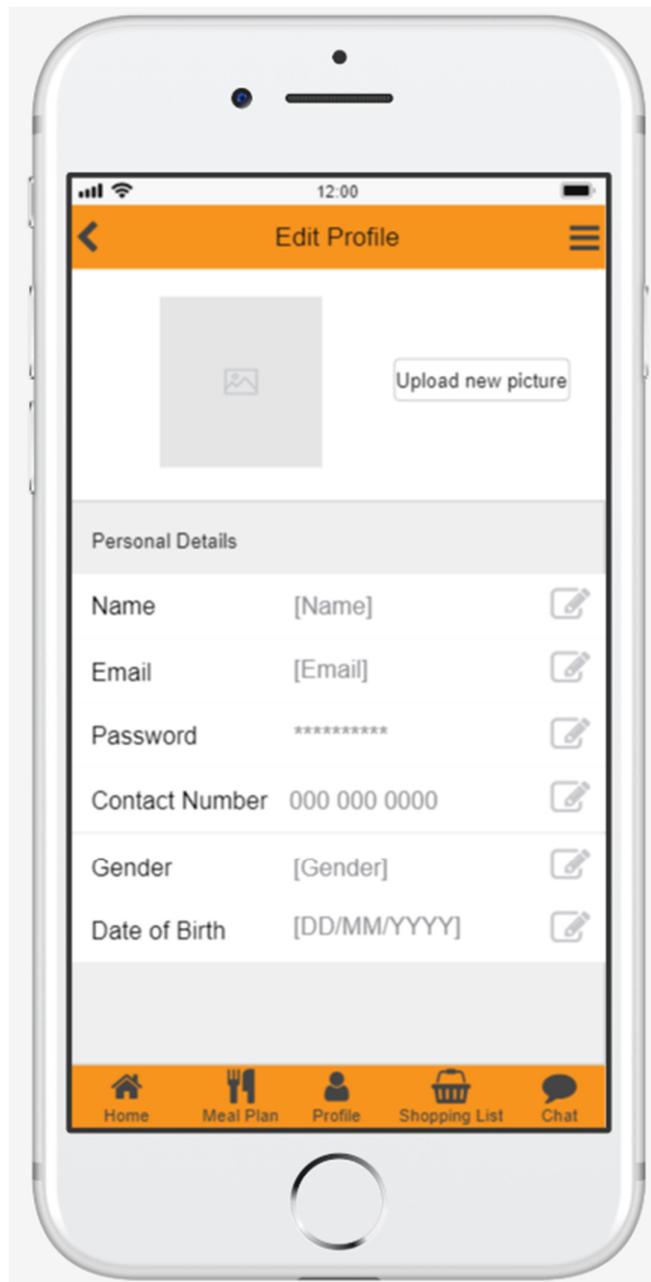


Figure 50 - Mock-up Design for the EatForLife Application Edit Profile Page.

## STATISTICS

The user is able to view their weight loss and measurement statistics through this page, with graphical representations of the progress they have made. This page is accessed through the profile page.

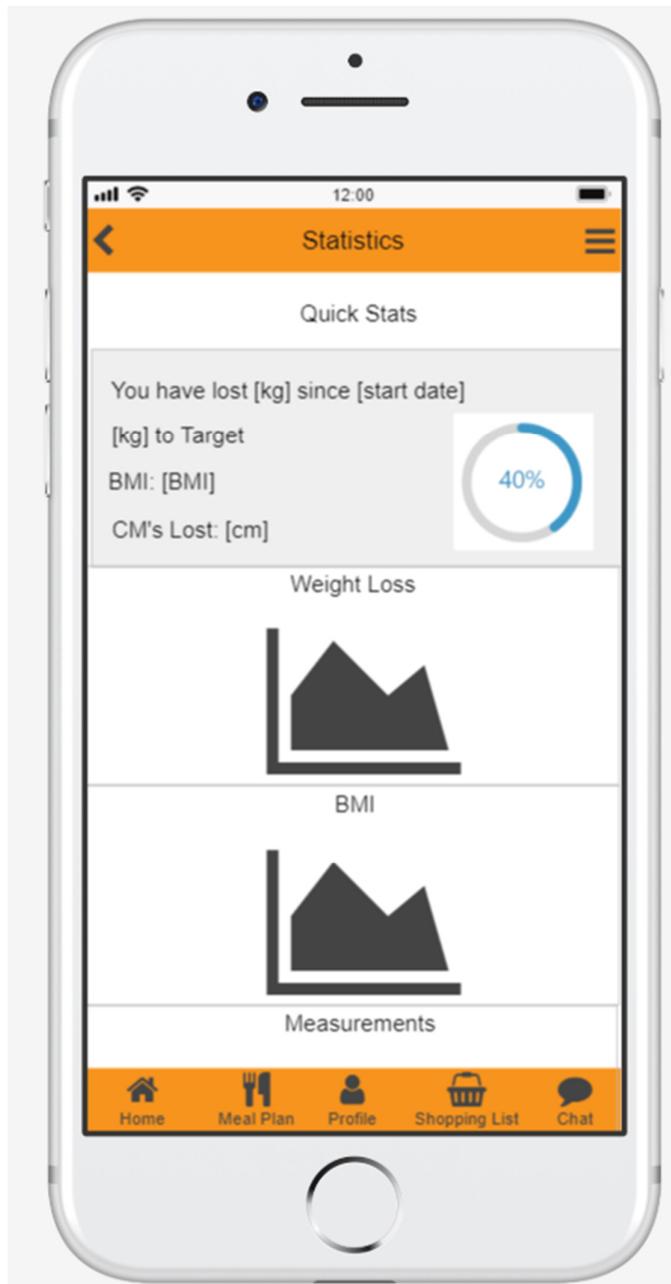


Figure 51 - Mock-up Design for the EatForLife Application Statistics Page.

## MEASUREMENT LOGS

The user is able to view all their previously entered measurements and weight logs in a tabular form. This page is accessed through the profile page.

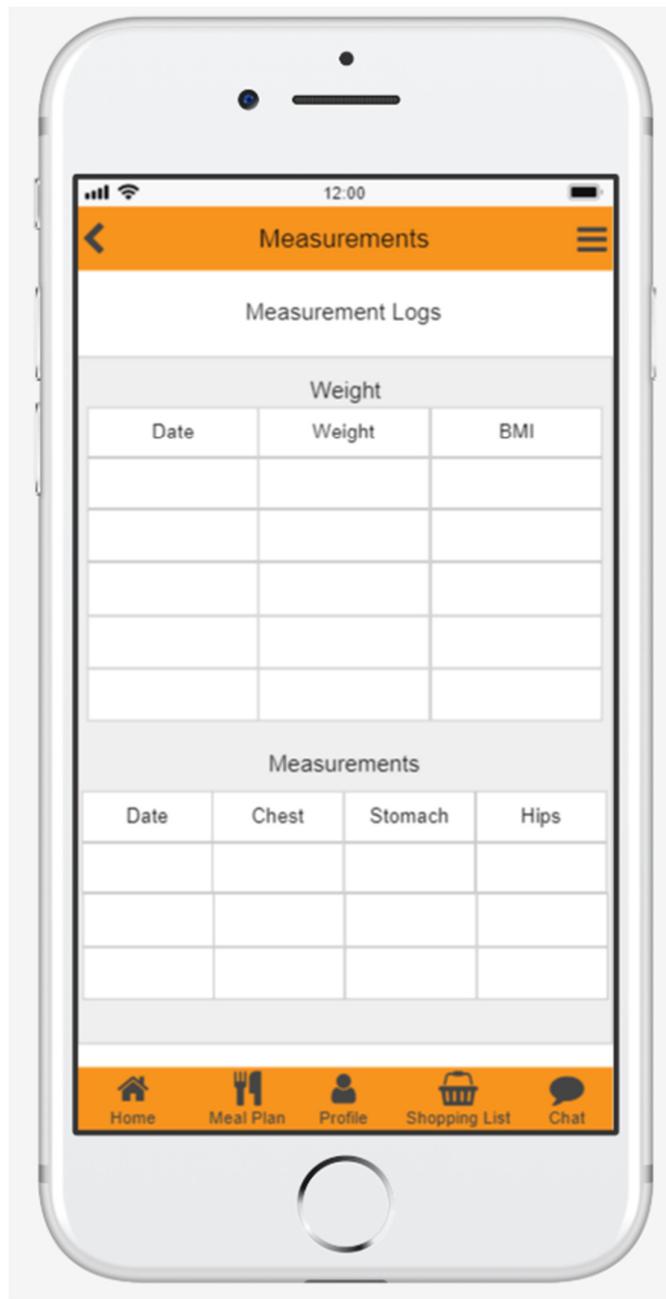


Figure 52 - Mock-up Design for the EatForLife Application Measurement Logs Page.

## CONTACT PAGE

The user is able to contact their dietician via email by using the contact form.

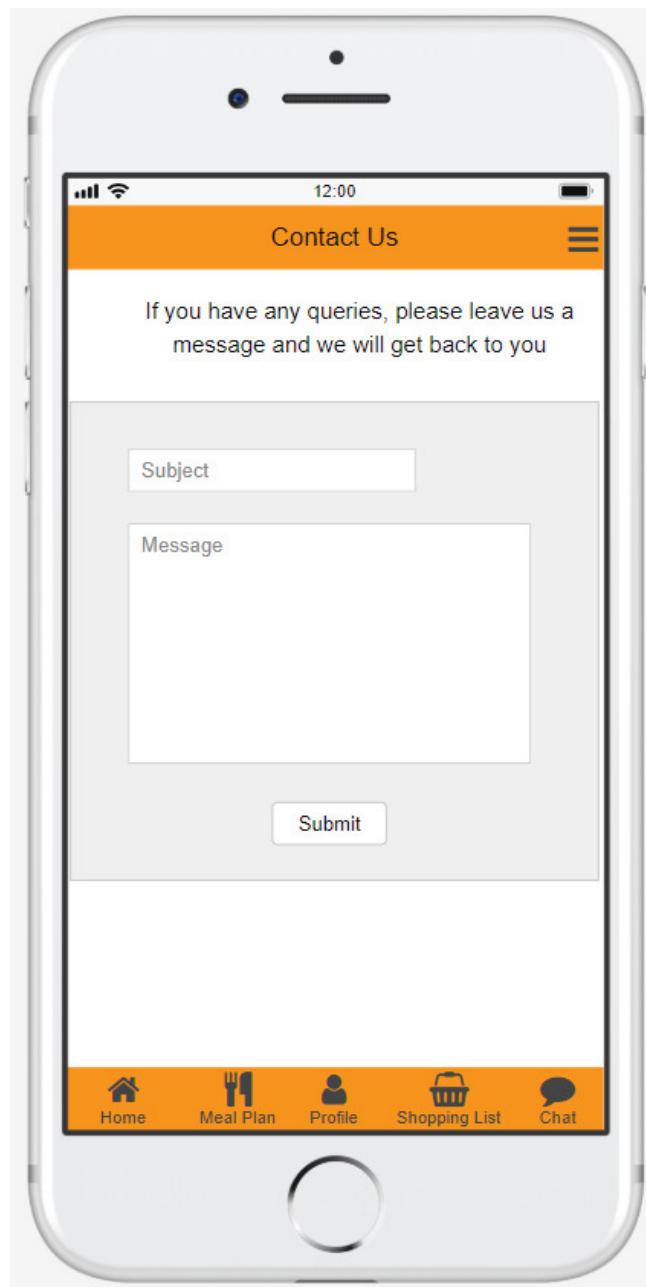


Figure 53 - Mock-up Design for the EatForLife Application Contact Page.

## DIETICIAN INTERFACE

The following screenshots show the interface that is displayed when a client account logs into the application.

### CLIENT LIST

The client list displays all clients assigned to the dietician, allowing them to view more details of the client and search through the client list.

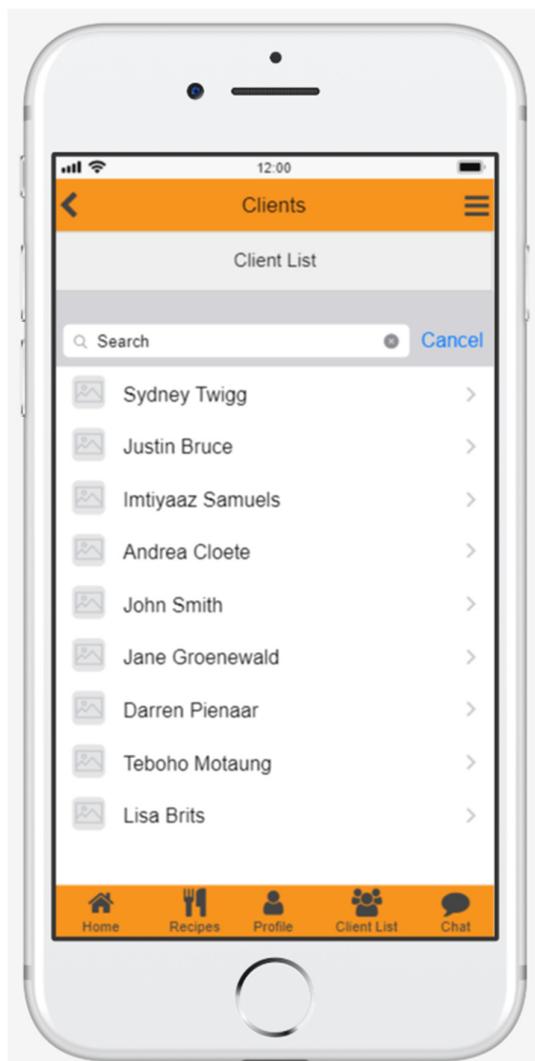


Figure 54 - Mock-up Design for the EatForLife Application Client List Page.

## CLIENT DETAILS

This page shows an overview of a client's progress with their weight loss progress, meal plan type and allows the dietician to send them a message directly from the page.

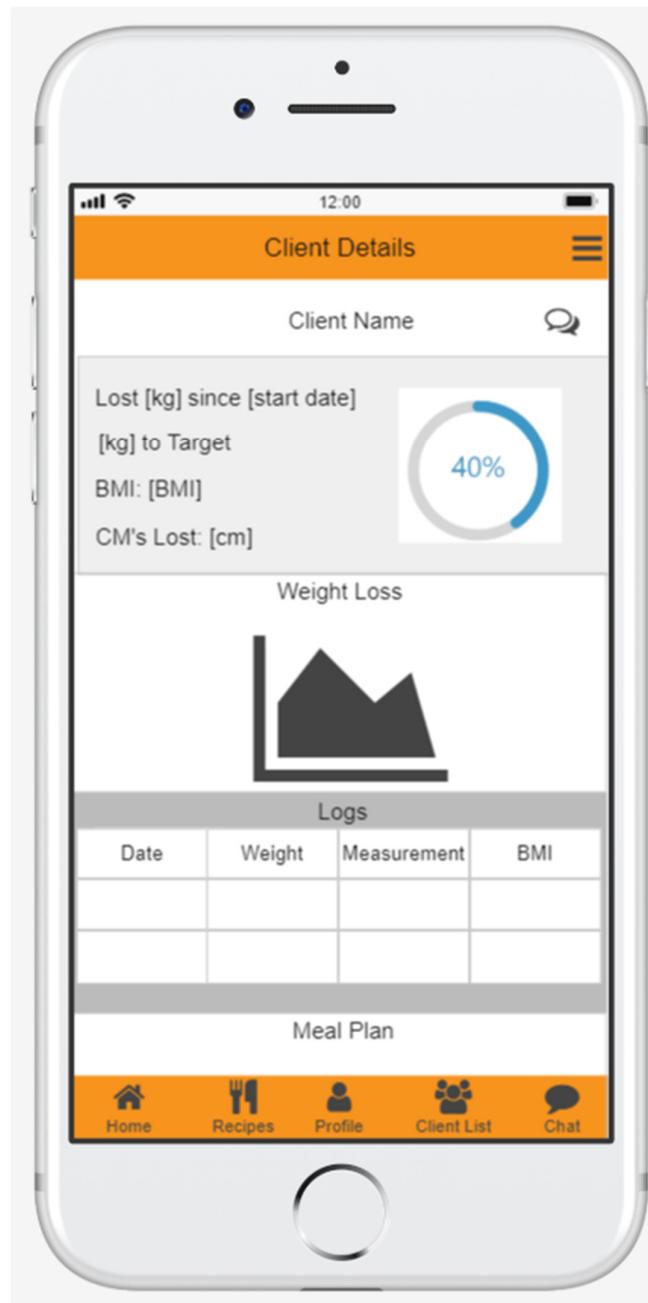


Figure 55 - Mock-up Design for the EatForLife Application Client Details Page.

## INCREMENT TWO: PROTOTYPE DESIGNS

The second increment of the system development involved programming the interfaces for the application and integration of the pages into one application for user acceptance testing of the navigation and functionality. Only the client interface has been created in this increment.

### LANDING PAGE

The landing page now includes the login page, to decrease the amount of user actions to take. Users are able to login from the landing page, as well as reset their password and access the registration page.

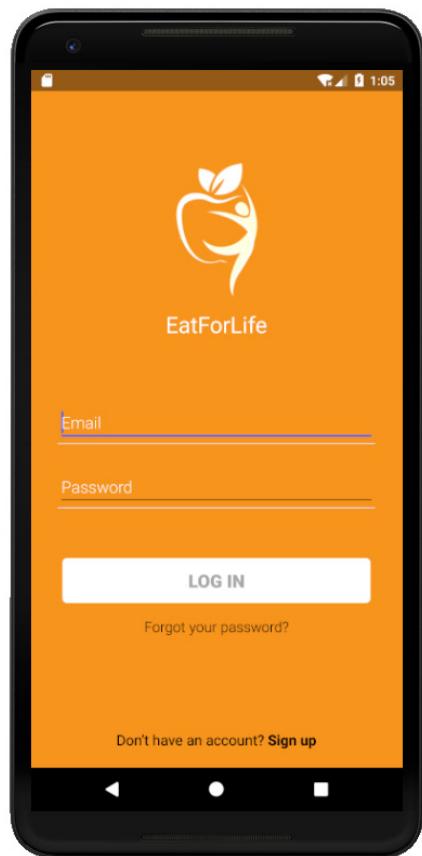


Figure 56 - Prototype of the EatForLife Application Landing Page.

## HOME PAGE

The home page displays an overview of the client's weight loss progress, as well as the badges they have unlocked so far. The home interface is consistent with increment one.

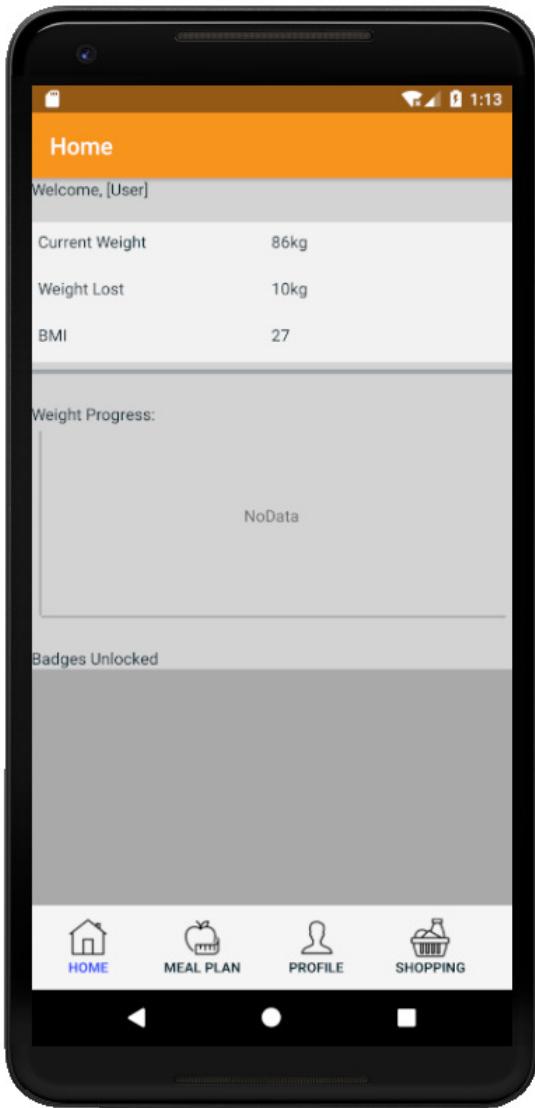


Figure 57 - Prototype of the EatForLife Application Home Page.

## MEAL PLAN OVERVIEW

The meal plan page allows the client to select which day of their meal plan they would like to view, as well as offering them options to print and email the full 7-day meal plan. The interface is consistent with the increment one design.

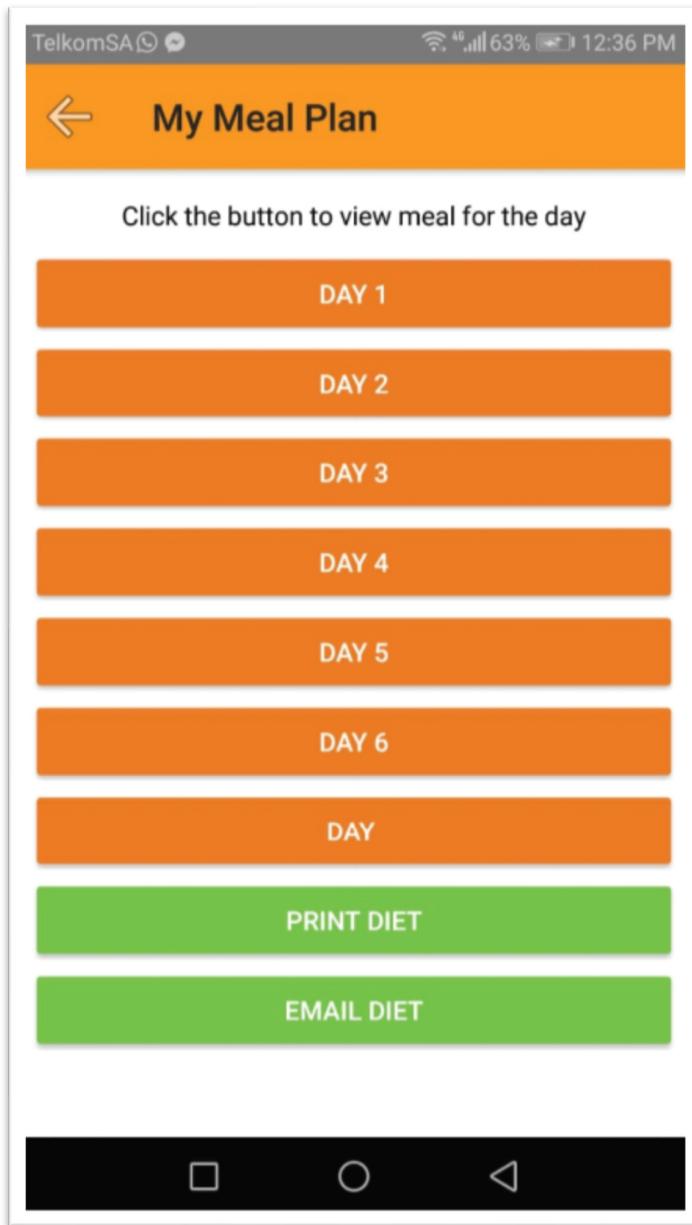


Figure 58 - Prototype of the EatForLife Application Meal Overview Page.

## MEAL PLAN DAY VIEW

The meal plan day view page allows the user to view their daily meal plan, with options to edit the food and add a substitute and reset back to default.

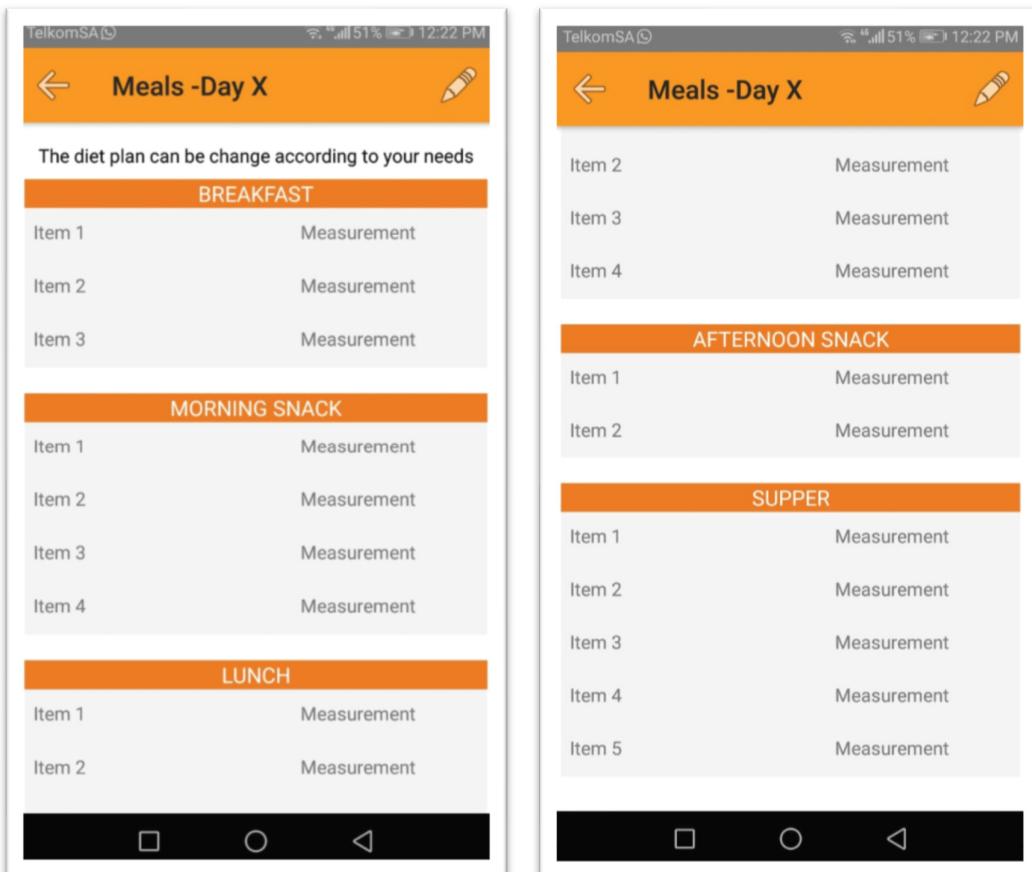


Figure 59 - Prototype of the EatForLife Application Meal Day View Page.

## EMAIL MEAL PLAN

The user is able to enter their email address to get their 7-day meal plan emailed to them.

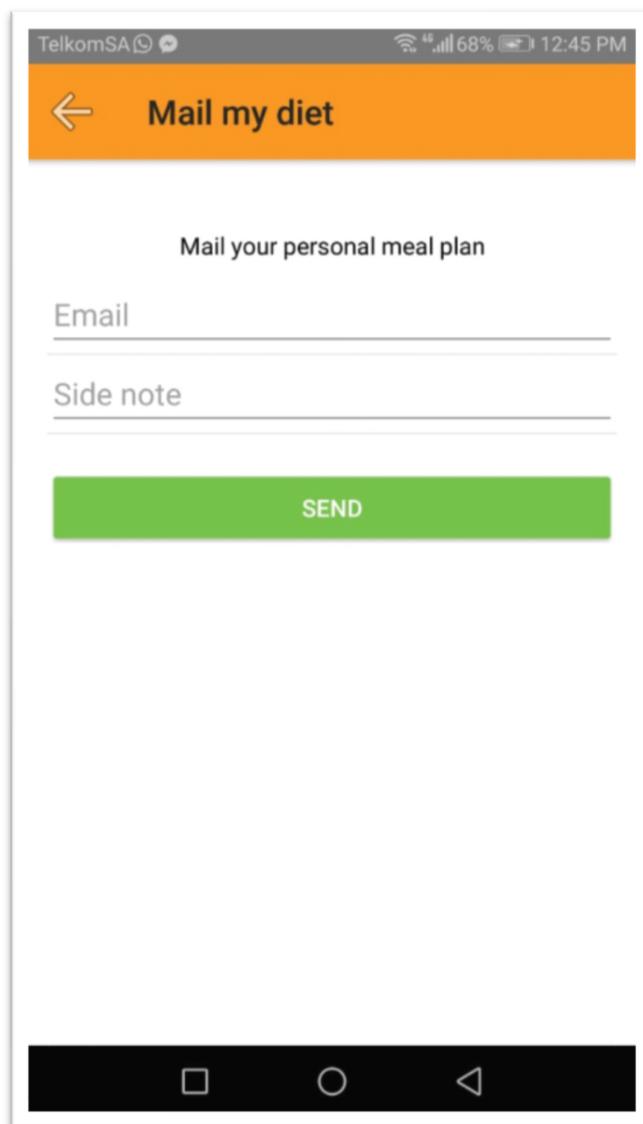


Figure 60 - Prototype of the EatForLife Application Email Meal Plan Page.

## SHOPPING LIST

The shopping list page shows users a shopping list that corresponds with their 7-day meal plan, allowing them to see all the foods they will need for the week.

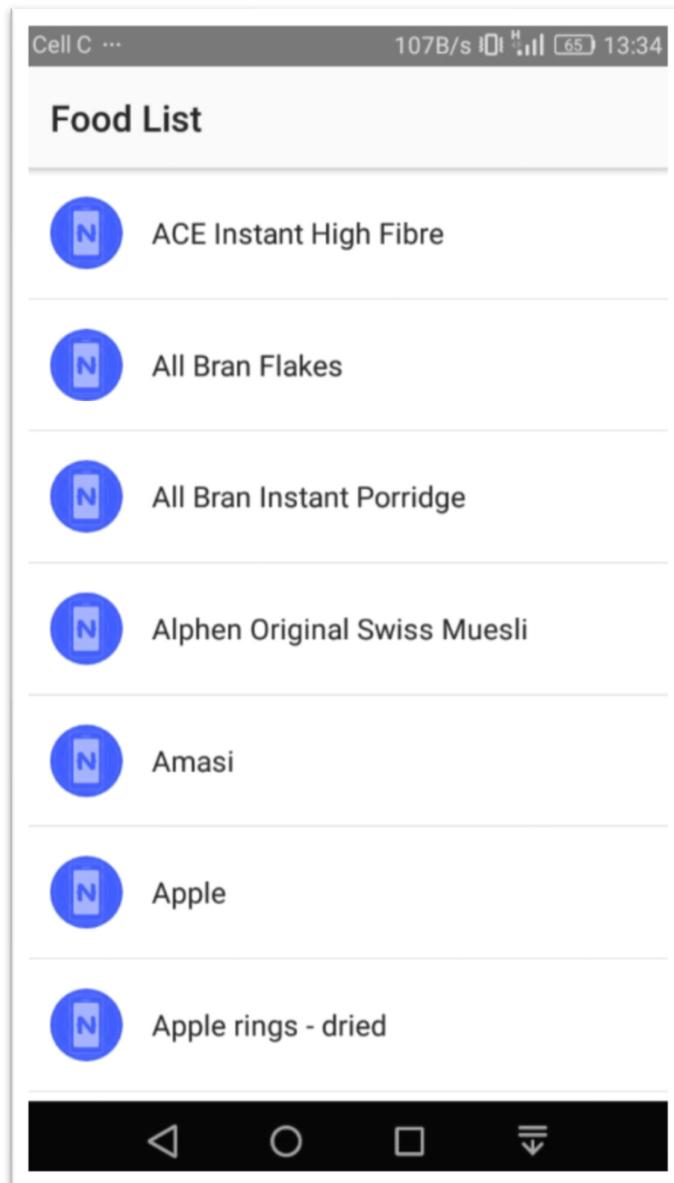


Figure 61 - Prototype of the EatForLife Application Shopping List Page.

## VIEW PROFILE

The user is able to view and edit their personal information on the profile page.

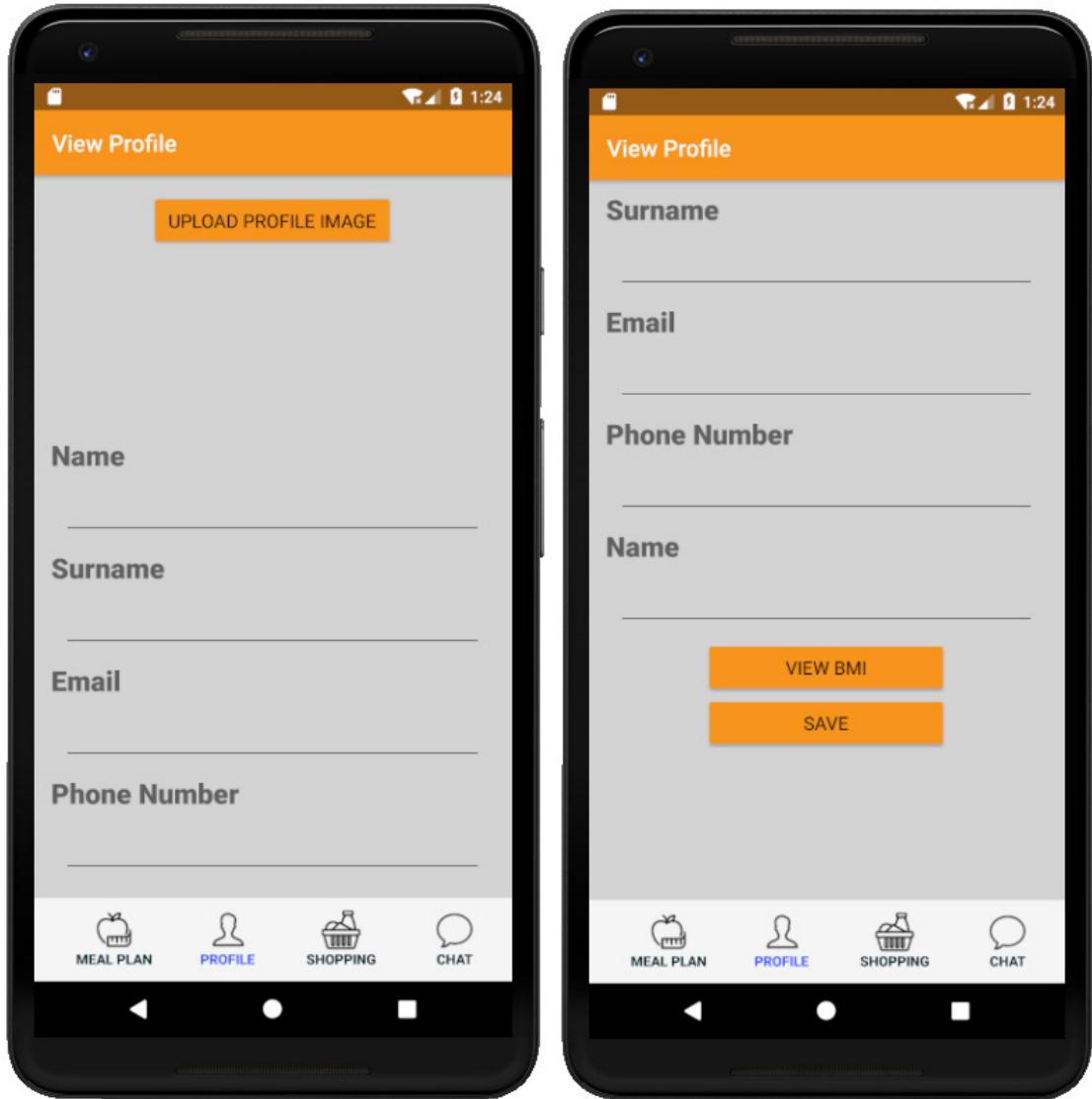


Figure 62 - Prototype of the EatForLife Application Profile Page.

## CONTACT PAGE

This page shows a form the user can complete in order to directly contact their dietician via email.

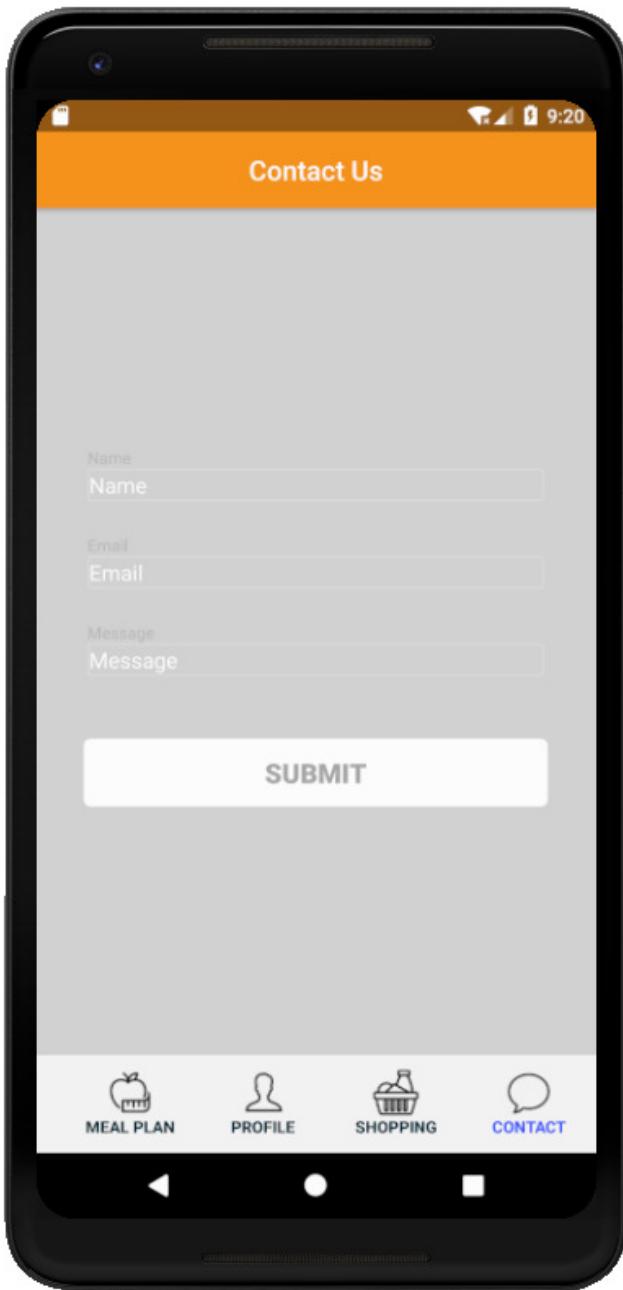


Figure 63- Prototype of the EatForLife Application Contact Us Page.

## INCREMENT 3: FINAL DESIGN

The following screenshots show the final design for the mobile application of EatForLife.

### LANDING PAGE

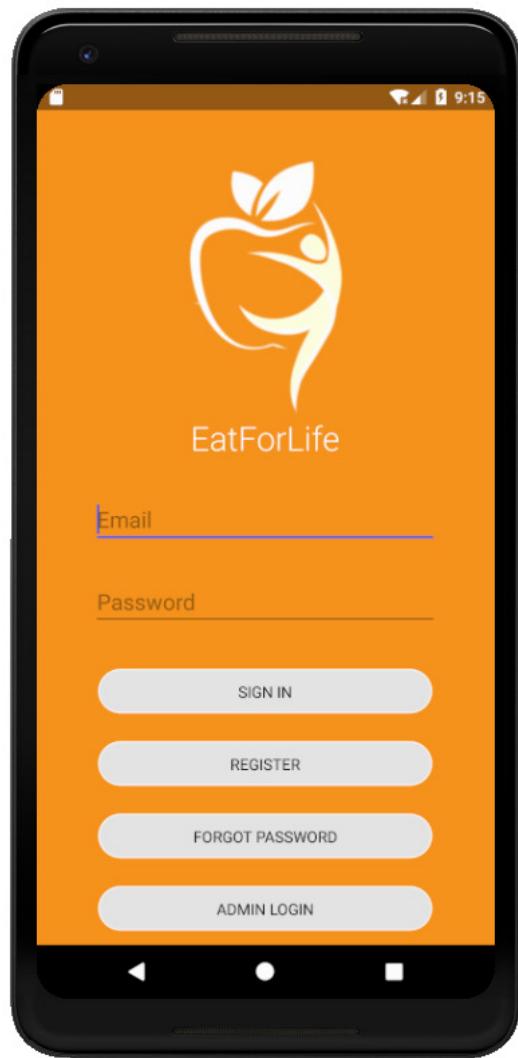


Figure 64 - Final Design for the EatForLife Mobile Application Landing Page.

## REGISTRATION PAGE

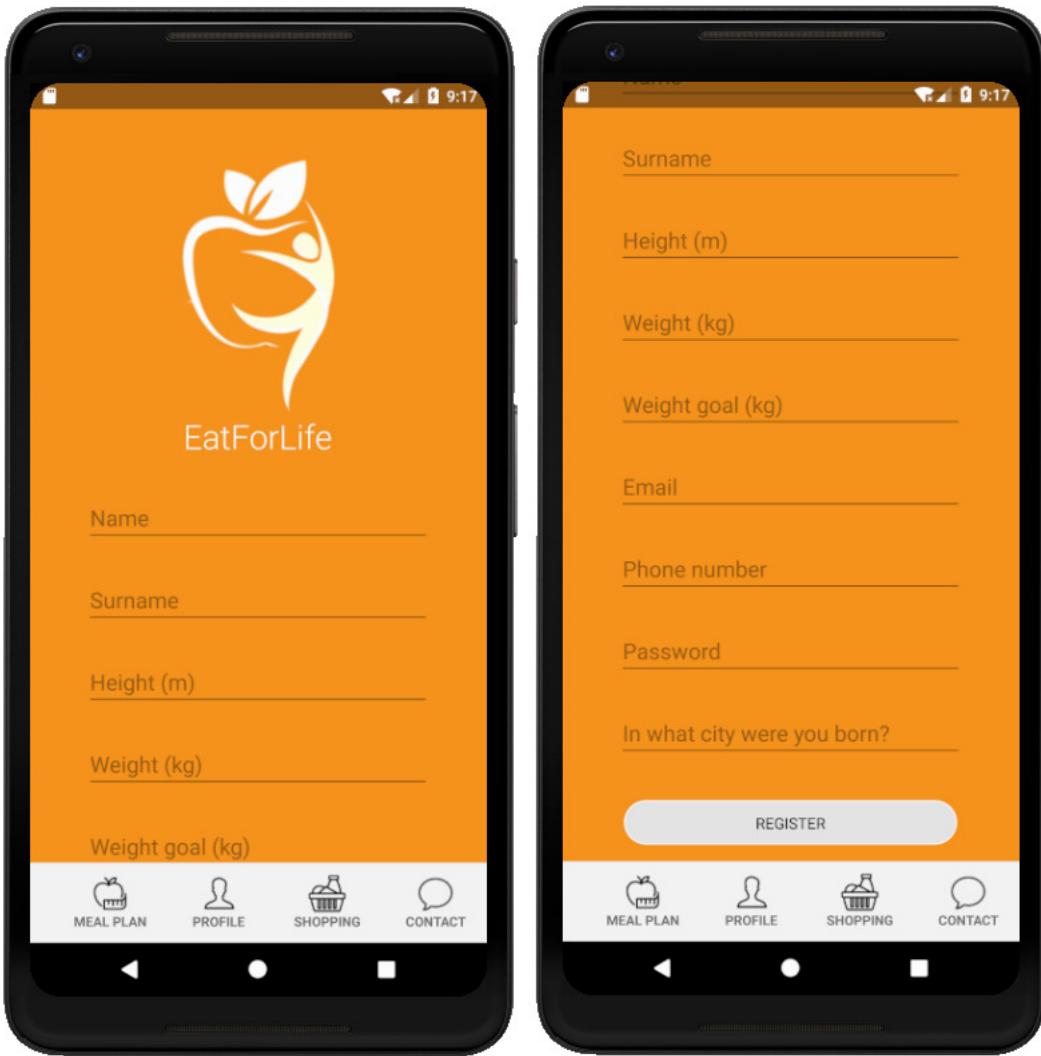


Figure 65 - Final Design for the EatForLife Mobile Application Registration Page.

## FORGOT PASSWORD PAGE

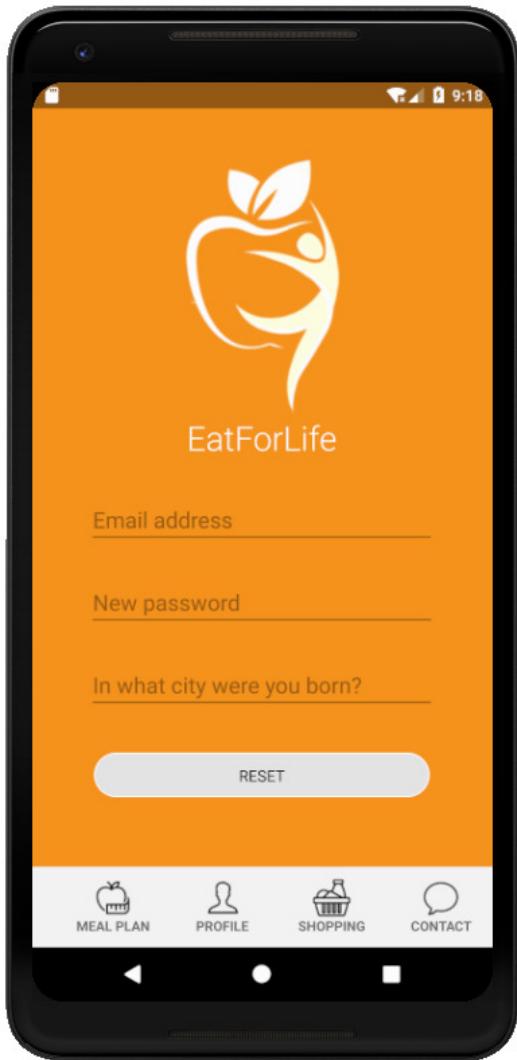


Figure 66 - Final Design for the EatForLife Mobile Application Forgot Password Page.

## HOME PAGE

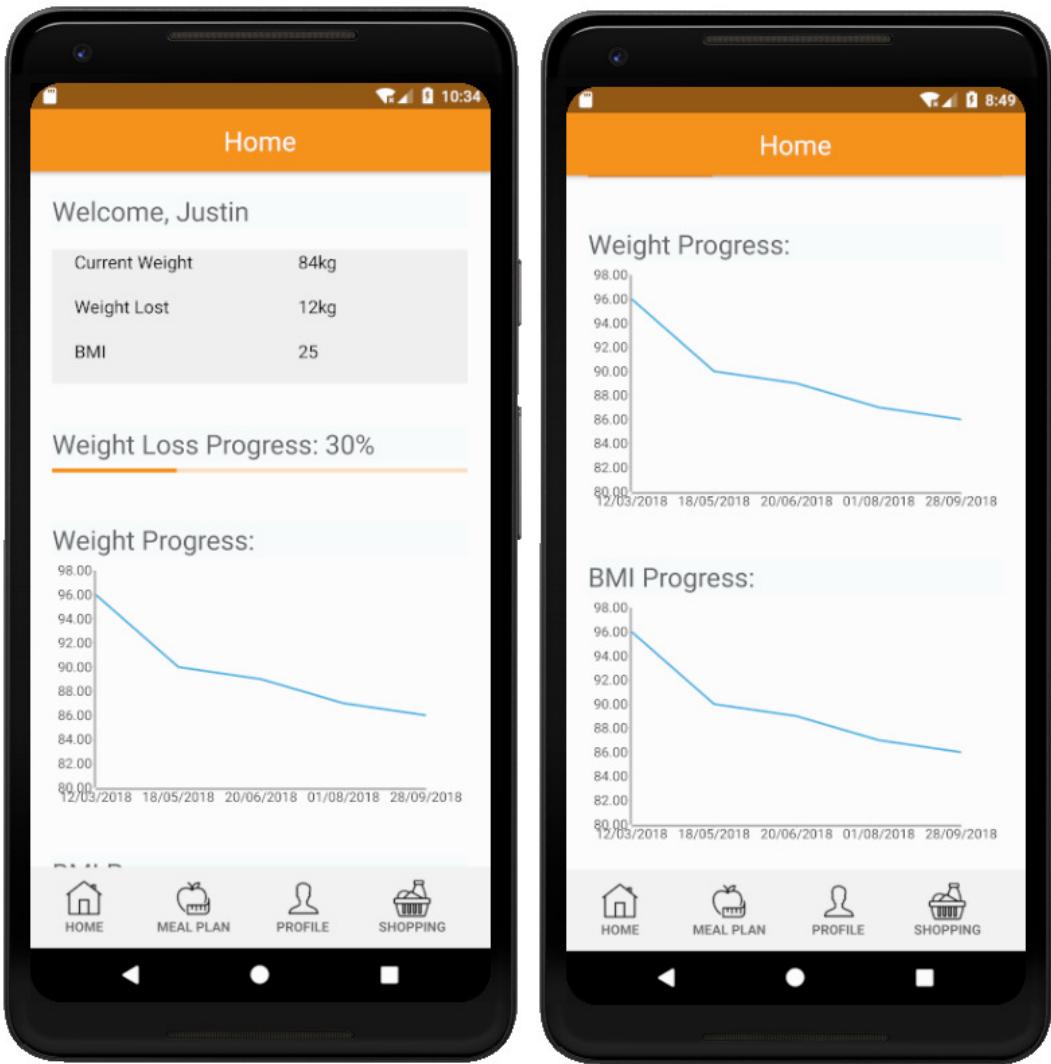


Figure 67 - Final Design for the EatForLife Mobile Application Home Page

## MEAL PLAN OVERVIEW PAGE

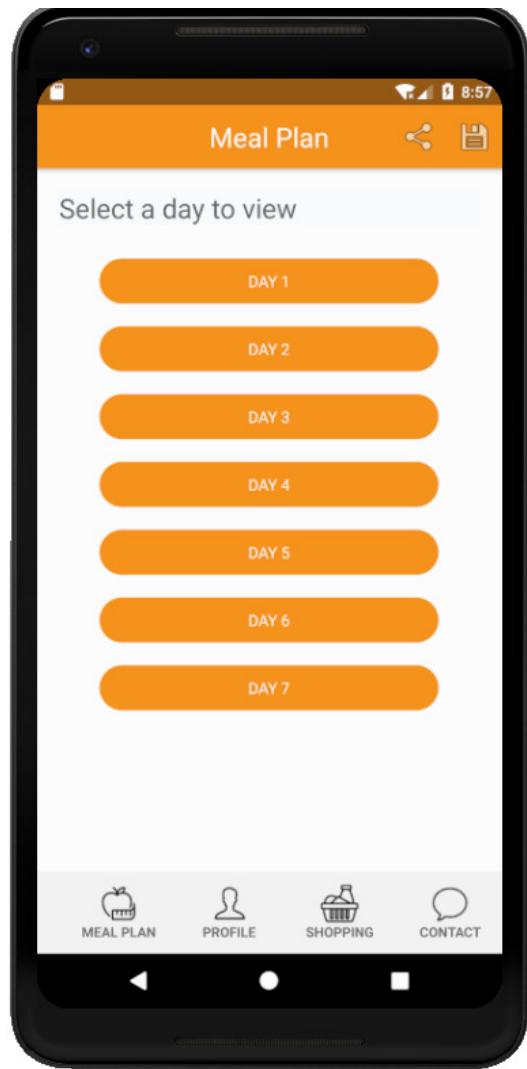


Figure 68 - Final Design for the EatForLife Mobile Application Meal Plan Overview Page

## MEAL PLAN DAY PAGE

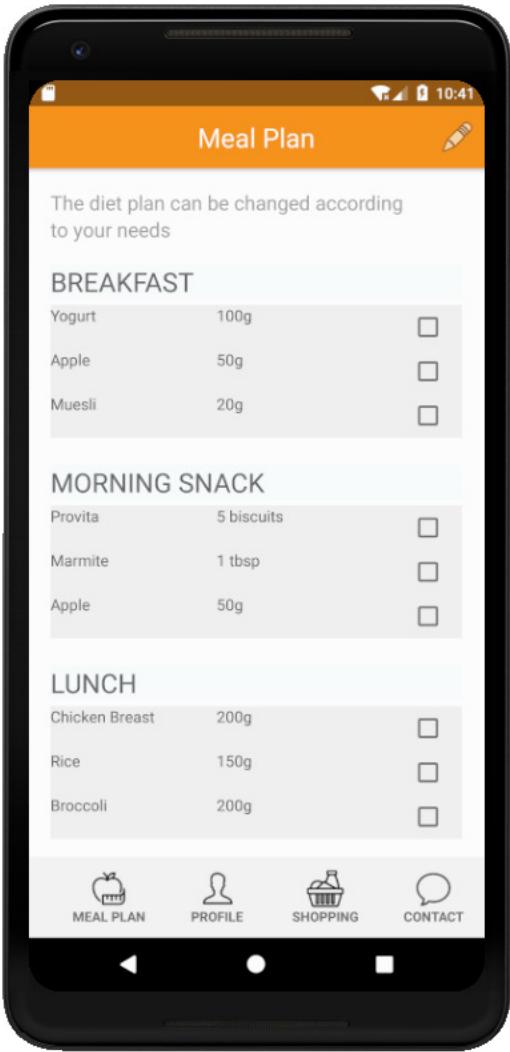


Figure 69 - Final Design for the EatForLife Mobile Application Meal Plan Day View Page

## EDIT MEAL PLAN PAGE

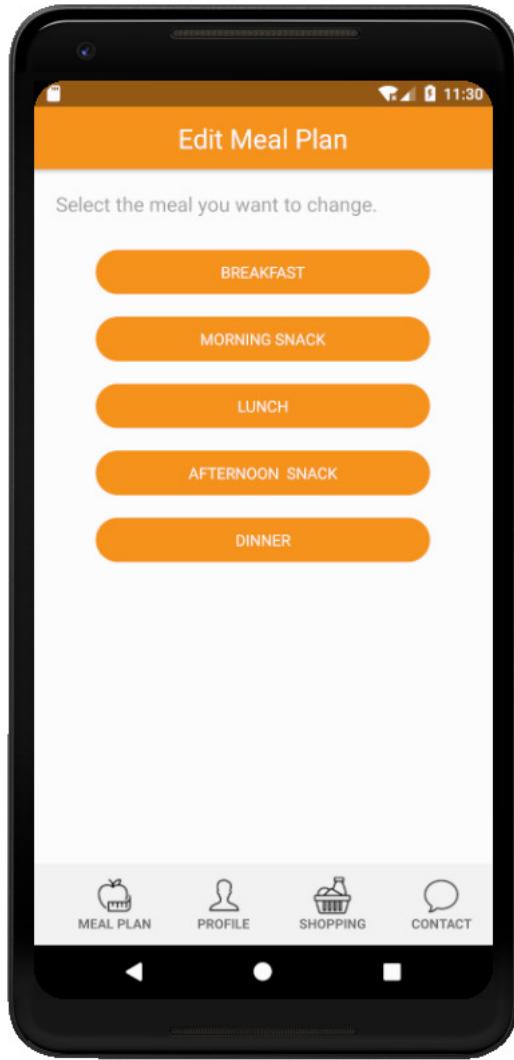


Figure 70 - Final Design for the EatForLife Mobile Application Edit Meal Plan Page

## EDIT MEAL ITEMS PAGE

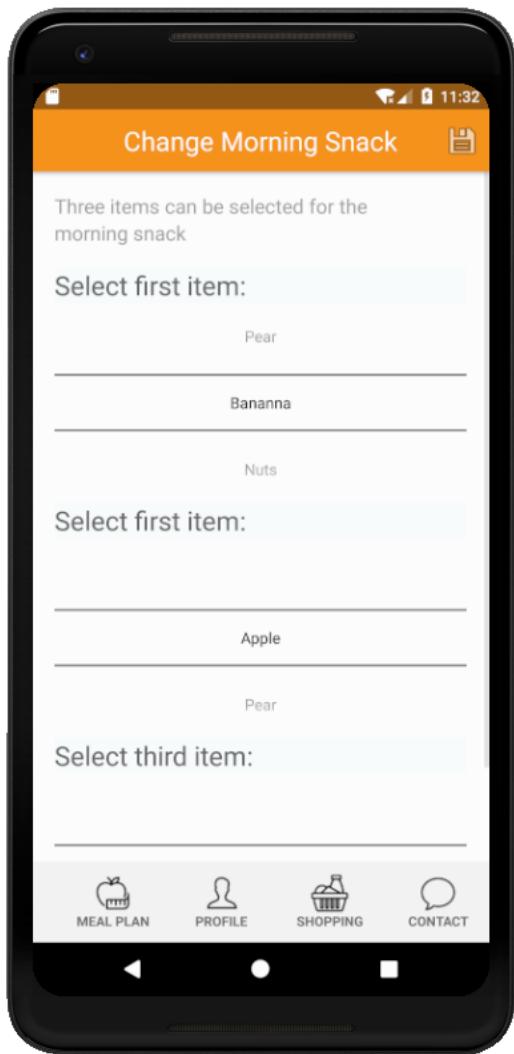


Figure 71- Final Design for the EatForLife Mobile Application Edit Meal Plan Page

Figure 72

## EMAIL MEAL PLAN PAGE

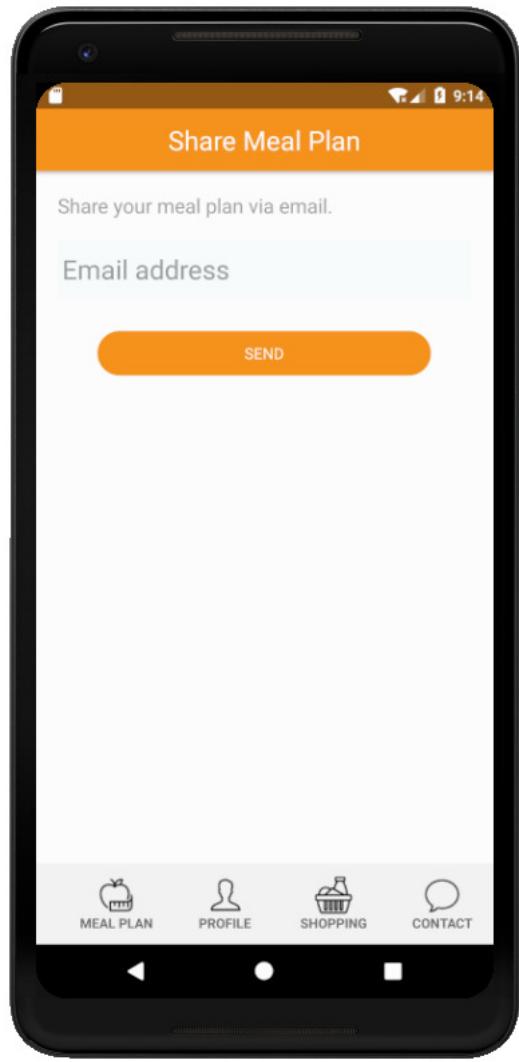


Figure 73- Final Design for the EatForLife Mobile Application Email Meal Page

## VIEW PROFILE PAGE

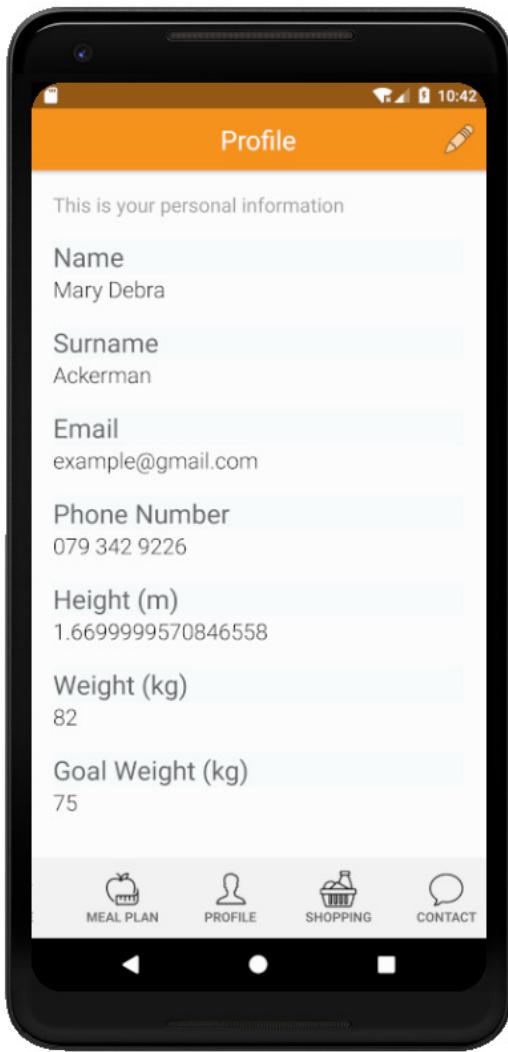


Figure 74 - Final Design for the EatForLife Mobile Application View Profile Page

## EDIT PROFILE PAGE

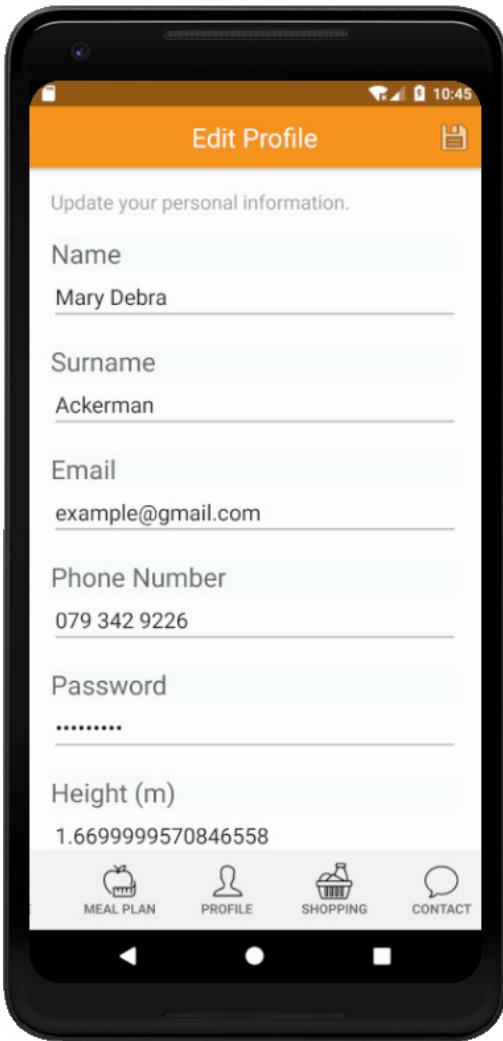


Figure 75 - Final Design for the EatForLife Mobile Application Edit Profile Page

## SHOPPING LIST PAGE

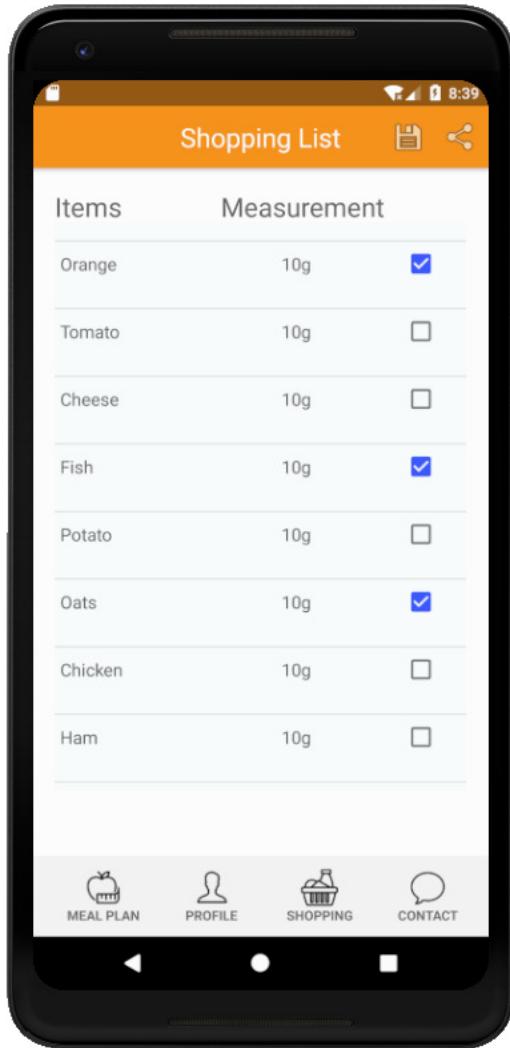


Figure 76 - Final Design for the EatForLife Mobile Application Shopping List Page

## EMAIL SHOPPING LIST PAGE

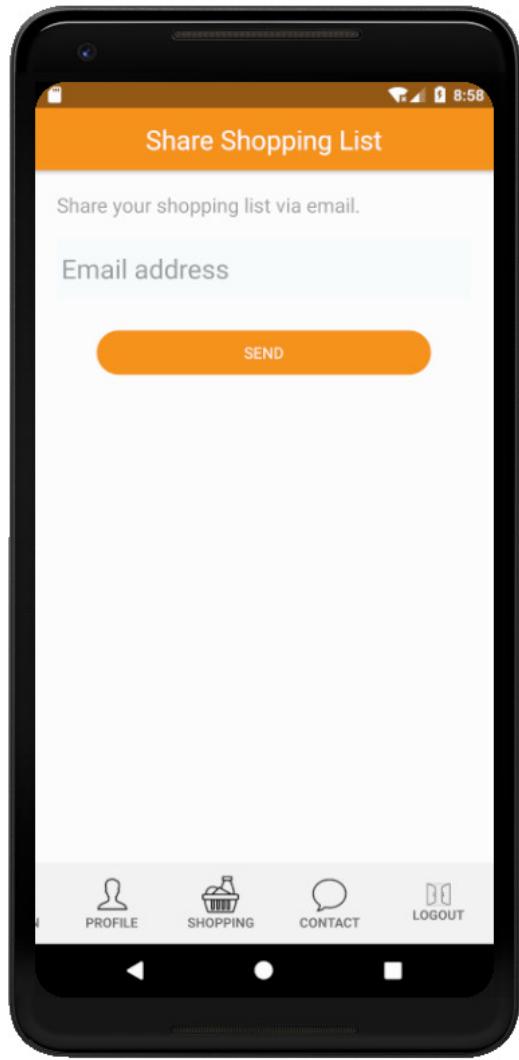


Figure 77 - Final Design for the EatForLife Mobile Application Email Shopping List Page

## CONTACT US PAGE

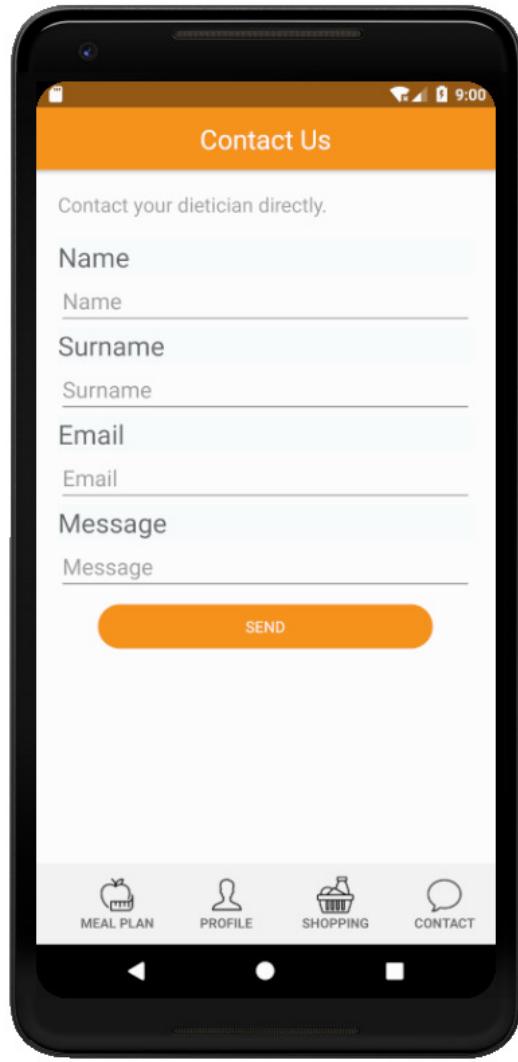


Figure 78 - Final Design for the EatForLife Mobile Application Contact Us Page

## ADMIN PANEL - VIEW CLIENT LIST PAGE

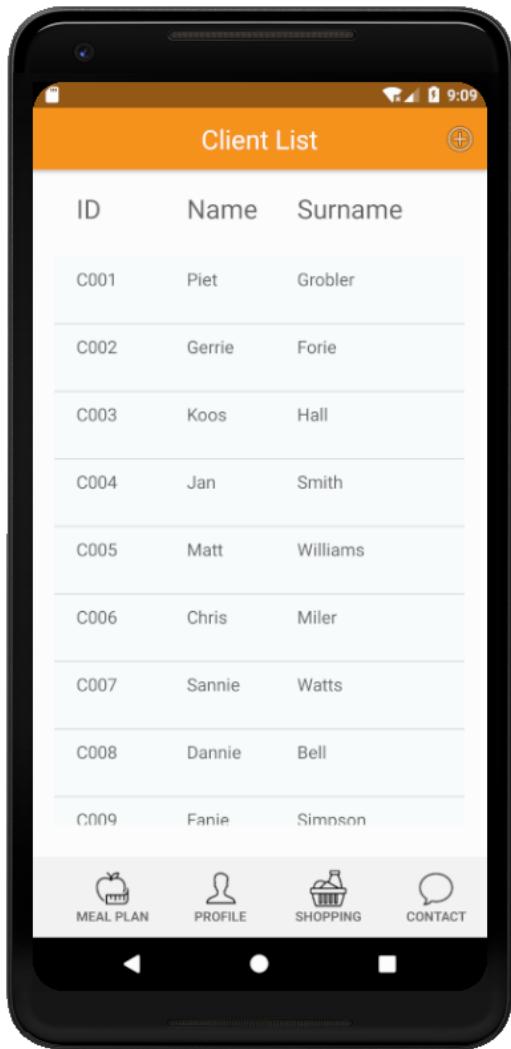


Figure 79 - Final Design for the EatForLife Mobile Application Admin Panel Client List Page

## ADMIN PANEL - VIEW CLIENT DETAIL PAGE

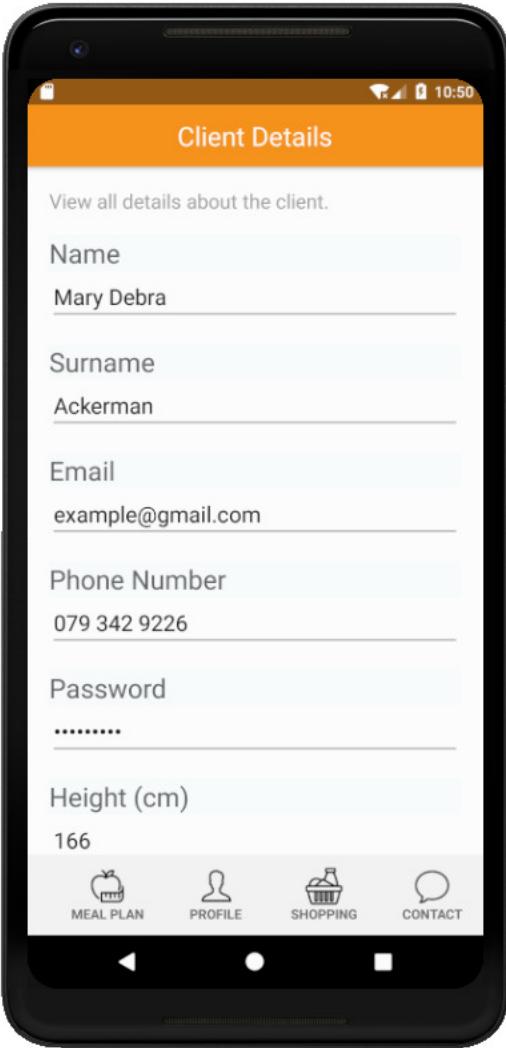


Figure 80 - Final Design for the EatForLife Mobile Application Admin Panel Client Detail Page

## ADMIN PANEL - ADD A NEW ADMIN MEMBER

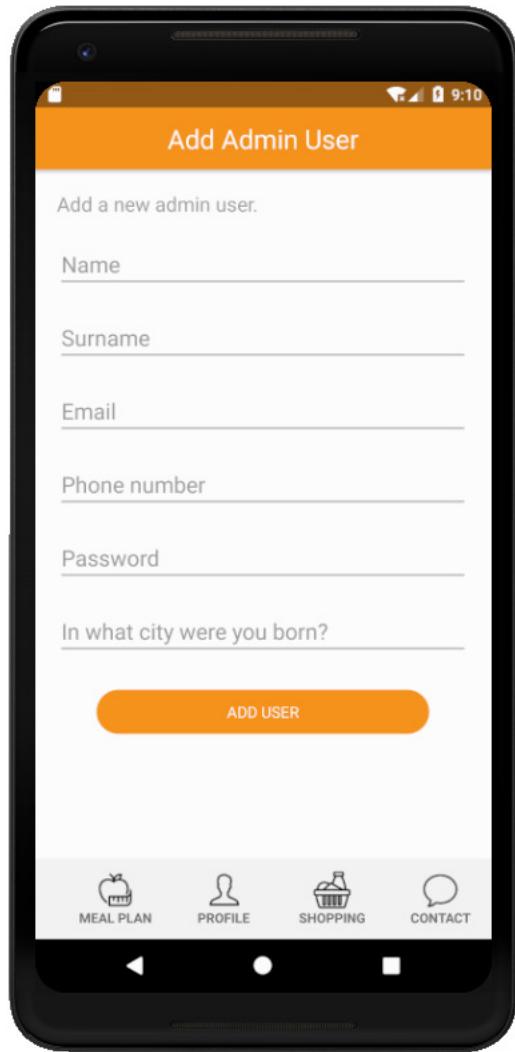


Figure 81 - Final Design for the EatForLife Mobile Application Add New Admin Member Page

# TESTING RESULTS

The group is following an Agile, incremental approach to development and testing. After each increment, the system is to be tested according to the outlined test plan and test templates. In the event that any features are added or removed, the test templates are to be edited to reflect the changes.

## TEST RESULTS: INCREMENT TWO: PROTOTYPE DESIGNS

The following test results were concluded after the prototypes of the application were developed in increment two.

### TEST PLAN

The test plan outlines which tests have been performed on the system.

*Table 39 - Test Plan for the EatForLife System Development, Increment Two.*

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
1	Prototype user interfaces	Functional Testing	04/09/2018	Sydney, Justin, Imtiyaaz, Andrea	Y
1	Prototype user interfaces	Usability Testing	04/09/2018	Sydney, Justin, Imtiyaaz, Andrea	Y
1	Prototype user interfaces	User Acceptance Testing	04/09/2018	Sydney, Justin, Imtiyaaz, Andrea	Y
1	Prototype user interfaces	Unit Testing	04/09/2018	Sydney, Justin, Imtiyaaz, Andrea	Y

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
2	Creation of database	Unit Testing	04/09/2018	Sydney, Justin, Imtiyaaz, Andrea	Y
2	Linking of the database to the application	Unit Testing			N
2	Database functions	Functional Testing	04/09/2018	Sydney, Justin, Imtiyaaz, Andrea	Y

## TEST TEMPLATE

### HOME PAGE

*Table 40 - Test Template for the EatForLife Home Page*

Page: Home Page (User Not Logged In)	
<b>Description:</b> This is the opening page of the EatForLife application, allowing users to either Login or Register for an account.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Test	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b>	

*Table 41 - Test Template for the EatForLife Home Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Home Page (User Not Logged In)
HM001	The user is able to view the Home Page.	7	
HM002	The user is able to view and select the Login button.	7	
HM003	The user is able to view and select the Register button.	7	

## REGISTRATION PAGE

*Table 42 - Test Template for the EatForLife Registration Page*

Page: Registration Page	
<b>Description:</b> The system allows users to enter their details to create an account	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Testing	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b> Not yet completed.	

*Table 43 - Test Template for the EatForLife Registration Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Registration
RG001	The user is able to view the Registration Page.	7	
RG002	The user is able to access navigation options (return to Home Page) from the Registration Page.	-	Removed
RG003	The user is able to view and enter their details.	1	Not Yet Completed
RG004	The user is able to view and select buttons.	1	Not Yet Completed
RG005	The user is given feedback if their submission was successful once the Submit button has been selected.	1	Not Yet Completed
RG006	Data validation is performed on all fields.	1	Not Yet Completed
RG007	The user is redirected to the login screen once they have successfully registered for a new account.	1	Not Yet Completed

## LOGIN PAGE

*Table 44 - Test Template for the EatForLife Login Page*

Page: Login Page	
<b>Description:</b> The system allows a user to login to an existing account	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Testing	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b> _____	

*Table 45 - Test Template for the EatForLife Login Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Login Page
<b>LG001</b>	The user is able to view the login page.	7	
<b>LG002</b>	The user is able to access navigation options (return to Home Page) from the Login Page.	-	Removed
<b>LG003</b>	The user is able to view and enter their details.	7	
<b>LG004</b>	The user is able to view and select buttons.	7	
<b>LG005</b>	Data validation is performed on all fields.	1	Not yet completed
<b>LG006</b>	The user is given feedback if their submission was successful once the Submit button has been selected.	1	Not yet completed
<b>LG007</b>	The user is redirected to their Home Page if successfully logged in.	5	Not yet integrated with the database - button functionality works.

## HOME PAGE

*Table 46 - Test Template for the EatForLife Home Page*

Page: Home Page (User Logged In)	
<b>Description:</b> Once a user is successfully logged in, their home page provides a quick overview of their information, weight loss statistics, and a weight loss graph.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Testing.	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b> _____	

*Table 47 - Test Template for the EatForLife Home Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Home Page (User Logged In)
HM004	The user is able to view the Home Page.	7	
HM005	The user is able to view their Member Details.	1	Not yet completed.
HM006	The user is able to view their statistics.	7	
HM007	The user is able to view a graph of their weight loss.	4	Works with dummy data - not integrated with database yet.

## MEAL PLAN PAGE

*Table 48 - Test Template for the EatForLife Meal Plan Page*

Page: Meal Plan Page	
<b>Description:</b> This page allows a user to view their seven-day meal plan, as well as edit any foods on their list, reset their meal plan, find more information on their foods, print or email the meal plan.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Testing.	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b>	

*Table 49 - Test Template for the EatForLife Meal Plan Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Meal Plan Page
MP001	The user is able to view the Meal Plan Page.	7	
MP002	The user is able to select which day of their meal plan they would like to view.	7	
MP003	The user is able to view items on their meal plan.	7	
MP005	The user is able to change a food to a replacement food by clicking the Edit Food button.	1	Not yet completed.
MP006	The user is able to view the foods separated by different meal times.	1	Not yet completed.
MP007	The user is able to reset their food choices back to the default - undoing any substitutions - by clicking the Reset Diet button.	7	
MP008	The user is able to view and print a printable version of their meal plan by clicking the Print button.	2	No functionality on the button.
MP009	The user is able to view and receive an email version of their meal plan by clicking the Email button.	1	No functionality on the button.

## SHOPPING LIST PAGE

*Table 50 - Test Template for the EatForLife Shopping List Page*

Page: Shopping List Page	
<b>Description:</b> This page allows a user to view all items they will need to purchase for their weekly meal plan, as well as allowing users to print or email the shopping list.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b>	

*Table 51 - Test Template for the EatForLife Shopping List Page*

Test ID	Requirement Shopping List Page	Acceptability (1-7)	Comments
SL001	The user is able to view the Shopping List page.	7	
SL002	The user is able to view all items and their measurements on the Shopping List.	7	
SL003	The user is able to view and print a printable version of their shopping list by clicking the Print button.	1	No functionality on the button
SL004	The user is able to view and receive an email version of their shopping list by clicking the Email button.	1	No functionality on the button
SL005	The shopping list is automatically updated when a user changes a food in their meal plan.	1	Not yet completed

## PROFILE PAGE

*Table 52 - Test Template for the EatForLife Profile Page*

Page: Profile Page	
<b>Description:</b> The user is able to view and update all of their details from their profile page, as well as view measurement logs from the profile page.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b>	

*Table 53 - Test Template for the EatForLife Profile Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Profile Page
PR001	The user is able to view the Profile Page.	7	
PR002	The user is able to view their personal details.	7	
PR003	The user is able to view and update their weight loss details.	1	Not yet completed
PR004	The user is able to view their historical measurement logs.	3	Not yet completed
PR005	The user is able to log out of the application.	1	Not yet completed

## COMMUNICATION PAGE

*Table 54 - Test Template for the EatForLife Communication Page*

Page: Communication Page	
<b>Description:</b> The user is able to directly contact their dietitian through the application.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b> Not yet integrated with the main application.	

*Table 55 - Test Template for the EatForLife Communication Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Communication Page
CM001	The user is able to view the Communication Page.	7	
CM002	The user is able to view and enter their details.	7	
CM003	The user is able to view and select buttons.	7	
CM004	Data validation is performed on all fields.	1	Not yet completed.
CM005	The user is given feedback if their submission was successful once the Submit button has been selected.	1	Not yet completed.
CM006	The user is able to send messages through the application.	1	Not yet completed.

## ADMIN PAGE

The admin page allows admin users to view all EatForLife clients and their details, as well as allowing the admin user to create a new admin user.

*Table 56 - Test Template for the EatForLife Admin Page*

Page: Admin Page	
<b>Description:</b> The system allows admin users to view all EatForLife clients and their details, as well as allowing for new admin users to be created.	<b>Version:</b> 1.0
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Sydney, Justin, Andrea, Imtiyaaz
<b>Signature:</b> _____	<b>Date:</b> 04/09/2018
<b>Additional Comments:</b>	

*Table 57 - Test Template for the EatForLife Admin Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Admin
AP001	The admin user is able to view the Admin Page.	7	
AP002	The admin user is redirected to the admin page when logging in with their admin credentials	3	
AP003	The user is able to view a list of all clients.	1	Not yet completed.
AP004	The user is able to click on a client name and be redirected to the user detail page.	1	Not yet completed.
AP005	The user is able to view the client details.	1	Not yet completed.
AP006	The user is able to view and select buttons.	5	
AP007	The user is able to select the “Add Admin User” button and be redirected to a new user form.	3	
AP008	The user is able to view and enter the details for a new admin user.	1	Not yet completed.
AP009	The user is able to view and select the “Submit” button.	1	Not yet completed.
AP010	Data validation is performed on all fields.	1	Not yet completed.
AP011	The user is given feedback if their submission was successful once the Submit button has been selected.	1	Not yet completed.
AP012	The user is able to log out.	1	Not yet completed.

## TEST RESULTS: INCREMENT THREE: FINAL SYSTEM

The following test results were concluded after the mobile application was deemed fully complete.

### TEST PLAN

The test plan outlines which tests have been performed on the system.

*Table 58 - Test Plan for the EatForLife System Development, Increment Two.*

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
1	Prototype user interfaces	Functional Testing			
1	Prototype user interfaces	Usability Testing			
1	Prototype user interfaces	User Acceptance Testing			
1	Prototype user interfaces	Unit Testing			
2	Creation of database	Unit Testing			
2	Linking of the database to the application	Unit Testing			
2	Database functions	Functional Testing			
3	Registration page	Functional Testing			
3	User registration	Unit testing			
3	User login	Unit testing			
3	Dietician login	Unit testing			
4	Profile Page	Functional testing			
4	Update personal details	Unit testing			
4	Update progress and measurements	Unit testing			

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
4	Generate graphs	Unit testing			
4	Generate reports	Unit testing			
4	Update goals	Unit testing			
5	Diet plan	Functional testing			
5	Generate weekly meal plan	Unit testing			
5	Display information on each food item on the meal plan	Unit testing			
5	Checklist functionality	Unit testing			
5	Display alternative substitutes for foods	Unit testing			
5	Download meal plan	Unit testing			
5	Email meal plan	Unit testing			
5	Generate shopping list	Unit testing			
5	Download shopping list	Unit testing			
5	Email shopping list	Unit testing			
6	Communications	Functional testing			
6	Communication between user and dietician	Unit testing			
6	Contact us page	Functional testing			
6	Users are able to contact the development team	Unit testing			
6	Users are able to contact EatForLife	Unit testing			
7	Admin page	Functional testing			
7	Admin users are able to access the admin page	Unit testing			

Increment	Description	Test Type	Test Date	Team Members	Complete (Y/N)
7	Admin users are able to add a new admin user	Unit testing			
-	System integration	Integration testing			
-	System integration	System testing			
-	Customer handover - user acceptance	User acceptance testing			

## TEST TEMPLATE

### HOME PAGE

*Table 59 - Test Template for the EatForLife Home Page*

Page: Home Page (User Not Logged In)	
<b>Description:</b> This is the opening page of the EatForLife application, allowing users to either Login or Register for an account.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Test	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 60 - Test Template for the EatForLife Home Page*

Test ID	Requirement Home Page (User Not Logged In)	Acceptability	Comments
		(1-7)	
HM001	The user is able to view the Home Page.	7	
HM002	The user is able to view and select the Login button.	7	
HM003	The user is able to view and select the Register button.	7	

## REGISTRATION PAGE

*Table 61 - Test Template for the EatForLife Registration Page*

Page: Registration Page	
<b>Description:</b> The system allows users to enter their details to create an account	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Testing	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 62 - Test Template for the EatForLife Registration Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Registration
RG001	The user is able to view the Registration Page.	7	
RG002	The user is able to access navigation options (return to Home Page) from the Registration Page.	7	
RG003	The user is able to view and enter their details.	7	
RG004	The user is able to view and select buttons.	7	
RG005	The user is given feedback if their submission was successful once the Submit button has been selected.	7	
RG006	Data validation is performed on all fields.	5	Not all fields
RG007	The user is redirected to the login screen once they have successfully registered for a new account.	7	

## LOGIN PAGE

*Table 63 - Test Template for the EatForLife Login Page*

Page: Login Page	
<b>Description:</b> The system allows a user to login to an existing account	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Testing	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 64 - Test Template for the EatForLife Login Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Login Page
<b>LG001</b>	The user is able to view the login page.	7	
<b>LG002</b>	The user is able to access navigation options (return to Home Page) from the Login Page.	7	
<b>LG003</b>	The user is able to view and enter their details.	7	
<b>LG004</b>	The user is able to view and select buttons.	7	
<b>LG005</b>	Data validation is performed on all fields.	7	
<b>LG006</b>	The user is given feedback if their submission was successful once the Submit button has been selected.	7	
<b>LG007</b>	The user is redirected to their Home Page if successfully logged in.	7	

## HOME PAGE

*Table 65 - Test Template for the EatForLife Home Page*

Page: Home Page (User Logged In)	
<b>Description:</b> Once a user is successfully logged in, their home page provides a quick overview of their information, weight loss statistics, and a weight loss graph.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Testing.	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 66 - Test Template for the EatForLife Home Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Home Page (User Logged In)
HM004	The user is able to view the Home Page.	7	
HM005	The user is able to view their Member Details.	7	
HM006	The user is able to view their statistics.	7	
HM007	The user is able to view a graph of their weight loss.	7	
HM008	The user is able to view a graph of their BMI.	7	

## MEAL PLAN PAGE

*Table 67 - Test Template for the EatForLife Meal Plan Page*

Page: Meal Plan Page	
<b>Description:</b> This page allows a user to view their seven-day meal plan, as well as edit any foods on their list, reset their meal plan, find more information on their foods, print or email the meal plan.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Testing.	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 68 - Test Template for the EatForLife Meal Plan Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Meal Plan Page
MP001	The user is able to view the Meal Plan Page.	7	
MP002	The user is able to select which day of their meal plan they would like to view.	7	
MP003	The user is able to view items on their meal plan.	6	Hard coded data
MP005	The user is able to change a food to a replacement food by clicking the Edit Food button.	4	No actual database save
MP006	The user is able to view the foods separated by different meal times.	7	
MP007	The user is able to reset their food choices back to the default - undoing any substitutions - by clicking the Reset Diet button.		
MP008	The user is able to view and print a printable version of their meal plan by clicking the Print button.	1	Print changed to save to device - button but no functionality
MP009	The user is able to view and receive an email version of their meal plan by clicking the Email button.	7	

## SHOPPING LIST PAGE

*Table 69 - Test Template for the EatForLife Shopping List Page*

Page: Shopping List Page	
<b>Description:</b> This page allows a user to view all items they will need to purchase for their weekly meal plan, as well as allowing users to print or email the shopping list.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 70 - Test Template for the EatForLife Shopping List Page*

Test ID	Requirement  Shopping List Page	Acceptability (1-7)	Comments
SL001	The user is able to view the Shopping List page.	7	
SL002	The user is able to view all items and their measurements on the Shopping List.	6	Hardcoded data
SL003	The user is able to view and print a printable version of their shopping list by clicking the Print button.	1	Print changed to save to device - button but no functionality
SL004	The user is able to view and receive an email version of their shopping list by clicking the Email button.	7	
SL005	The shopping list is automatically updated when a user changes a food in their meal plan.	1	Fixed shopping list for overall diet

## PROFILE PAGE

*Table 71 - Test Template for the EatForLife Profile Page*

Page: Profile Page	
<b>Description:</b> The user is able to view and update all of their details from their profile page, as well as view measurement logs from the profile page.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 72 - Test Template for the EatForLife Profile Page*

Test ID	Requirement Profile Page	Acceptability (1-7)	Comments
PR001	The user is able to view the Profile Page.	7	
PR002	The user is able to view their personal details.	7	
PR003	The user is able to view and update their weight loss details.	6	Able to edit - not saved to database
PR004	The user is able to view their historical measurement logs.	6	Hardcoded data
PR005	The user is able to log out of the application.	6	Back end functionality

## COMMUNICATION PAGE

*Table 73 - Test Template for the EatForLife Communication Page*

Page: Communication Page	
<b>Description:</b> The user is able to directly contact their dietitian through the application.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 74 - Test Template for the EatForLife Communication Page*

Test ID	Requirement	Acceptability (1-7)	Comments
			Communication Page
CM001	The user is able to view the Communication Page.	7	
CM002	The user is able to view and enter their details.	7	
CM003	The user is able to view and select buttons.	7	
CM004	Data validation is performed on all fields.	7	
CM005	The user is given feedback if their submission was successful once the Submit button has been selected.	1	Send with GMail app and no such feature supported
CM006	The user is able to send messages through the application.	7	Access GMail through app

## ADMIN PAGE

The admin page allows admin users to view all EatForLife clients and their details, as well as allowing the admin user to create a new admin user.

*Table 75 - Test Template for the EatForLife Admin Page*

Page: Admin Page	
<b>Description:</b> The system allows admin users to view all EatForLife clients and their details, as well as allowing for new admin users to be created.	<b>Version:</b> 1.1
<b>Type of Test:</b> Usability, User Acceptance Test.	<b>Tested By:</b> Andrea
<b>Signature:</b> _____	<b>Date:</b> 06/11/2018
<b>Additional Comments:</b> Testing was done while project was still in development	

*Table 76 - Test Template for the EatForLife Admin Page*

Test ID	Requirement	Acceptability (1-7)		Comments
		Admin		
AP001	The admin user is able to view the Admin Page.	7		
AP002	The admin user is redirected to the admin page when logging in with their admin credentials	7		
AP003	The user is able to view a list of all clients.	6		Hardcoded data
AP004	The user is able to click on a client name and be redirected to the user detail page.	7		
AP005	The user is able to view the client details.	6		Hardcoded
AP006	The user is able to view and select buttons.	7		
AP007	The user is able to select the “Add Admin User” button and be redirected to a new user form.	7		
AP008	The user is able to view and enter the details for a new admin user.	7		
AP009	The user is able to view and select the “Submit” button.	7		
AP010	Data validation is performed on all fields.	1		No validation
AP011	The user is given feedback if their submission was successful once the Submit button has been selected.	1		No feedback
AP012	The user is able to log out.	1		No log out

## SCHEDULE

### GANTT CHART FOR DELIVERABLE 4

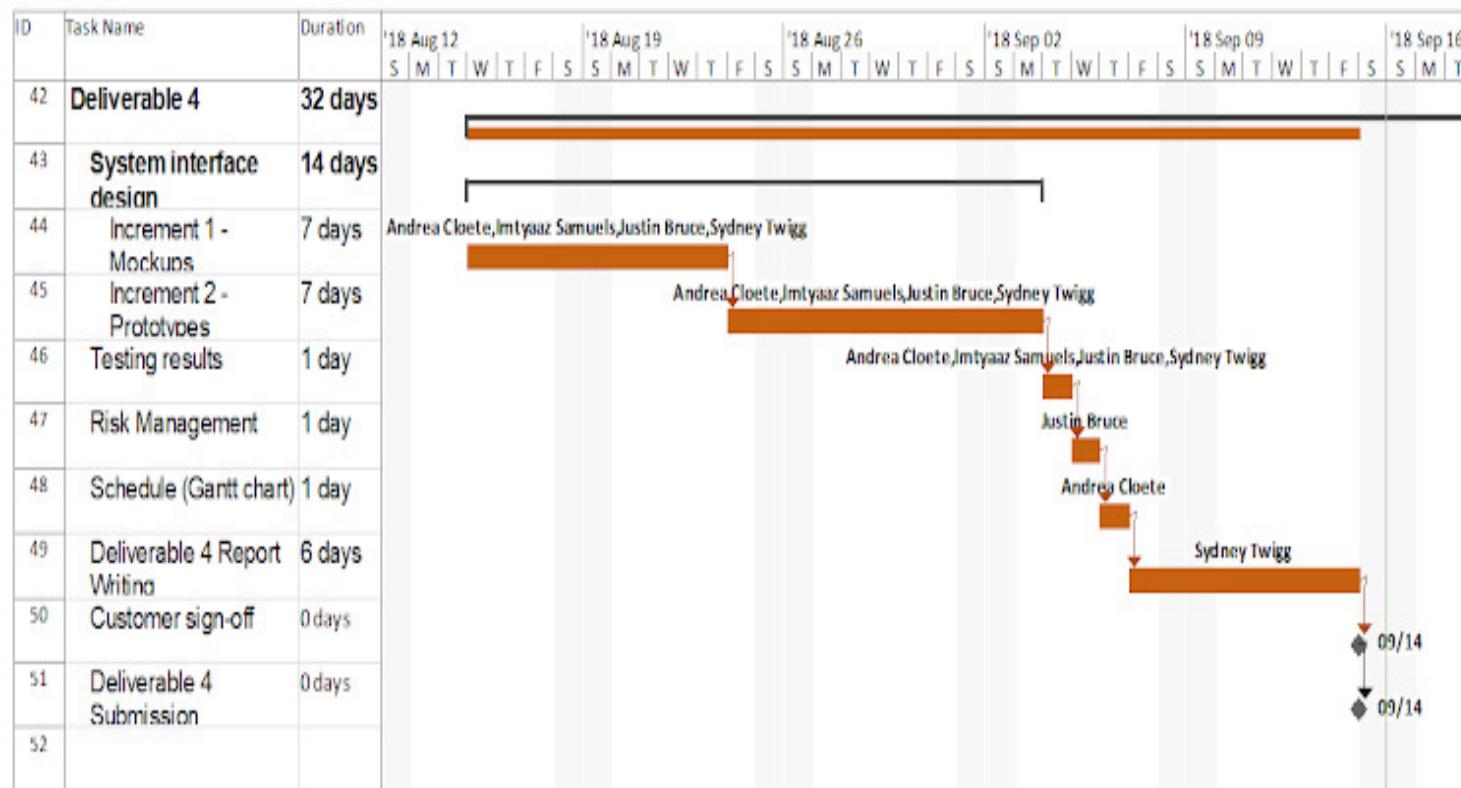


Figure 82 - Gantt Chart for Deliverable 4.

## RISK REGISTER

*Table 77 - Risk Register for the EatForLife application development.*

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R06	1	Cross-over incompatibility	The application may work well on Android but not on iOS	Technology Risk	The chosen coding platform does not support the needed language conversion	The chosen platform is incorrect for development of both Android and iOS	Focus on the functional platform to deliver the best full working project, the other platform can be integrated further down the line.	DOT Development	Medium	High
R14	2	Software failure during presentation	The software that is used to simulate a mobile device fails to load the project system or the IDE does not run the code needed correctly	Technology Risk	The software used is faulty, causes too much stress on the platform's processor or poor system maintenance	The team members did not prepare a well-working platform that could handle the presentation of the project	Ensure that all components are working as intended before the presentation and have a backup platform as a fail-safe	DOT Development	Medium	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R13	3	Hardware failure during presentation	During the presentation of the system, the hardware that the presentation is supposed to be presented on suffers from technical difficulties such as crashes	Technology Risk	Poor system maintenance , faulty computer components, untrustworthy software installed on the system	The team members did not prepare a well-working platform that could handle the presentation of the project	Ensure that all components are working as intended before the presentation and have a backup platform as a fail-safe	DOT Development	Medium	High
R02	4	Delay of required client information	The client takes a long time to respond to the group's questions via email or does not answer calls consistently	People Risk	The client does not regularly read their email, is too busy to reply or answer the phone or is impeded by another variable	The client isn't as invested as they should be in developing their system	Try complete information needed through other research methods - online etc.	DOT Development	Medium	Medium

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R03	5	Uncertainty of full client requirements	The client gives the team a vague list of what he/she wants the system to be and what it should entail	People Risk	The team does not aggressively ask for the requirements	Identified during meetings with the client	Send the client a system proposal with requirement ideas for their feedback	DOT Development	Medium	Medium
R11	6	Update of requirements	The client decides to change requirements for the system mid-development	People Risk	The client no longer wants to have specific features in the system or wants to add additional features to the system	The client's business develops new requirements for their future success	The team will ensure that a meeting takes place in order to document the new requirements appropriately	DOT Development	Medium	Medium
R04	7	Bad communication skills between team members	Team members do not communicate appropriately and/or frequently enough	People Risk	Due to team members not owning a phone, not using discord (our team's communication application) or google drive	Team members don't know how to work well with each other	Discuss it with the person, if there is no improvement escalate the issue with your lecturer.	DOT Development	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R05	8	Resignation of a team member	Team member decides to resign mid-way during the development process	People Risk	Unexpected turn of events or too much stress for the team member to handle	The team member can no longer take the pressure of developing the system	The team will have to accept the decision for the resignation of the member and continue development	DOT Development	Low	High
R01	9	Loss of business	The client decides to no longer work with our team	Market Risk	The client has lost interest in the system or no longer wants to invest the time	The client's business no longer needs a new system to further their success	Find a new client, or if too late, continue without the client	Client	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R07	10	Fail to choose a suitable coding language	The team does not choose a suitable coding language the is required for the chosen/design ed system	People Risk	Team members may be unsure for which language(s) best suit their skills and the requirements for development	There are multiple languages to choose from and team members have their own unique skills in their respective languages	Use a familiar well documented language	DOT Development	Low	High
R12	11	Hardware failure	The developers have an unforeseen hardware failure, resulting in critical data loss.	Technology Risk	Poor system maintenance , faulty computer components, untrustworth y software installed on the system	Team members did not ensure their hardware was functional and that critical data was backed up	Ensure that all critical data is backed up through Google Drive, GitHub and other cloud storage, or offsite storage.	DOT Development	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R10	12	Quality of the system is not satisfactory for the client	The client decides that they do not like the developed system and no longer want to use it	Market Risk	The developed system is not up the client's standards nor meets their expectations	The developed system is invalid for the use of the client	As a group ensure that all requirements are met. If the client does not like system that meets the requirements, terminate the project.	Client	Low	High
R15	13	Failure to answer questions during presentation	Individual team members fail to answer questions about the system during presentation	People Risk	Lack of knowledge on how the system functions	Team members did not acknowledge nor learn what all the components of the system does	Ensure that all team members are fully briefed on how the system runs/operates	DOT Development	Low	High

No.	Rank	Risk	Description	Category	Root Cause	Triggers	Potential Responses	Risk Owner	Probability	Impact
R08	14	Chosen IDE is inappropriate for development	The chosen IDE may not be compatible with required plugins or not contain much needed tools for development	Technology Risk	Team members do not know of various IDE's that may be suitable for the development of the system	The chosen IDE does not support the needed tools for the input of needed plugins or modules	Find a new IDE that suits the project	DOT Development	Low	Medium
R09	15	Poor time management	Increments are put together late	People Risk	Due to team members delaying input due to incompetence or personal issues	Team members do not know how to schedule their time well enough to get work done	Discuss with the individual why their work is not timely, assist them if needed to finish the deliverables in schedule. If the team members do not improve their time management, escalate it to the lecturer.	DOT Development	Low	Low

## USER MANUAL

# USER MANUAL - EATFORLIFE MOBILE APPLICATION

### INTRODUCTION

The scope of the EatForLife mobile application project was to design and implement a cross-platform mobile application that will provide an easy way for members to access and manage their personalised eating plan.

The user manual for the EatForLife application is a practical guide to assist a user to understand how to operate within the app. It will also allow the user to understand how to make full use of the application. This user manual will assist the user from the installation process to leaving the app.

Dot Development are the proud developers of the EatForLife app, the team was formed by CTI students Sydney Twigg, Imtiyaaz Samuels, Andrea Cloete & Justin Bruce as part of the 3<sup>rd</sup> year Software Development Project. The Dot Development Team can be reached for additional assistance at the following email address: [dotdevelopment@gmail.com](mailto:dotdevelopment@gmail.com).

# GETTING STARTED

The following instructions will assist the user in the functional operation of the app.

## HOW TO INSTALL EATFORLIFE

*A user can install the EatForLife app from the Android Play Store.*

1. Open the **Play Store** 
2. Enter “*EatForLife*” into the search bar;
3. Download the application, the *EatForLife* app will install on your mobile device; and;
4. The *EatForLife* app can be found in the applications menu, or on the home screen of the mobile device.

*A user can install the EatForLife app from the Apple App Store:*

1. Open the App Store 
2. Enter “*EatForLife*” into the search bar;
3. Download the application, the *EatForLife* app will install on your mobile device; and;
4. The *EatForLife* app can be found in the applications menu, or on the home screen of the mobile device.

## INSTALLING LOCALLY

For testing purposes, the application can be run on the developer’s PC through an emulator:

1. Install NativeScript:
  - a. Open the command line;
  - b. Run the following command `npm install -g nativescript`
2. Navigate to the folder containing the EatForLife source code in the command line:
  - a. Open the command line
  - b. Run the following command `cd [file path to the project]`.
3. In the command line, run the command `tns run`
4. The *EatForLife* application will be launched on an emulator.

## ACCESSING THE LANDING PAGE

1. Click on the app icon in the application menu once it has been successfully installed.

The figure below shows the landing page for the EatForLife application.



*Figure 83 - EatForLife landing page.*

## LOGIN TO THE APPLICATION

1. Click the “Login” button on the landing page.
2. Enter your username and password.
3. Click the “Login” button.
  - a. If the login was successful, you will be redirected to the Home page.

## REGISTER A NEW ACCOUNT

1. Click the “Register” button on the landing page.
2. Complete all the required fields.
3. Click on the “Register” button.
  - a. If the registration is successful you will be redirected to the Login page.

The figure below shows the registration page.



*Figure 84 - Registration page.*

## NAVIGATING THE APPLICATION

The user is able to navigate using the Navigation bar at the bottom of the application:

1. Click on the button corresponding to which page you would like to view.

The following image shows the navigation bar for the EatForLife application.

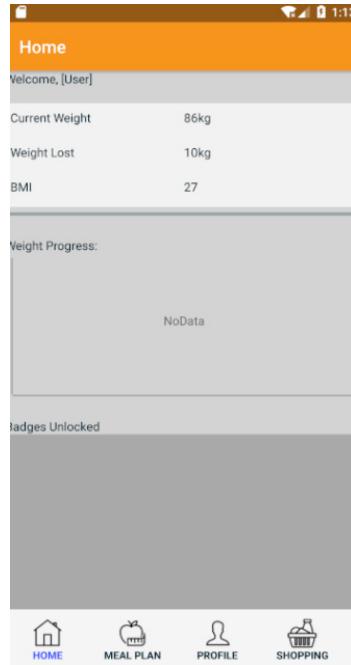


*Figure 85 - Navigation bar.*

## VIEW THE HOME PAGE

1. The user will automatically be directed to the Home Page after logging in to the application.
2. Click the “Home” button to navigate to the home page.

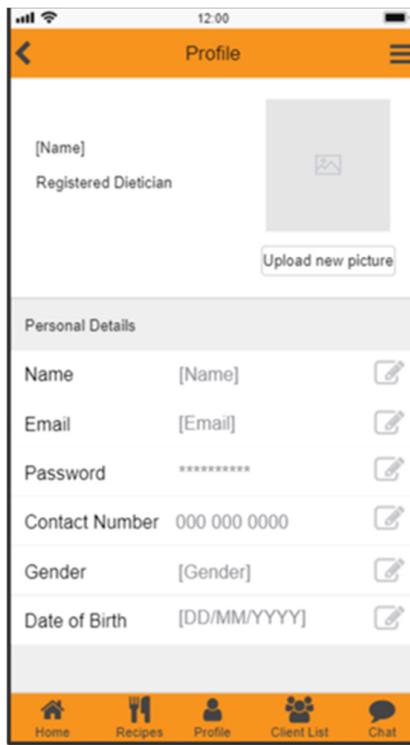
The following image shows the Home page.



*Figure 86 - EatForLife home page.*

## INTERACT WITH YOUR PERSONAL PROFILE

The image below displays the view of the Profile screen.



*Figure 87 - The Profile screen displays your Profile picture, designated Dietician and your personal details*

As a user, you are able to:

### **Set a Profile Picture:**

1. Tap the “Upload new picture” button
2. Search through the media in your device for a picture
3. Select the picture and upload it



*Figure 88 - Upload a new profile picture.*

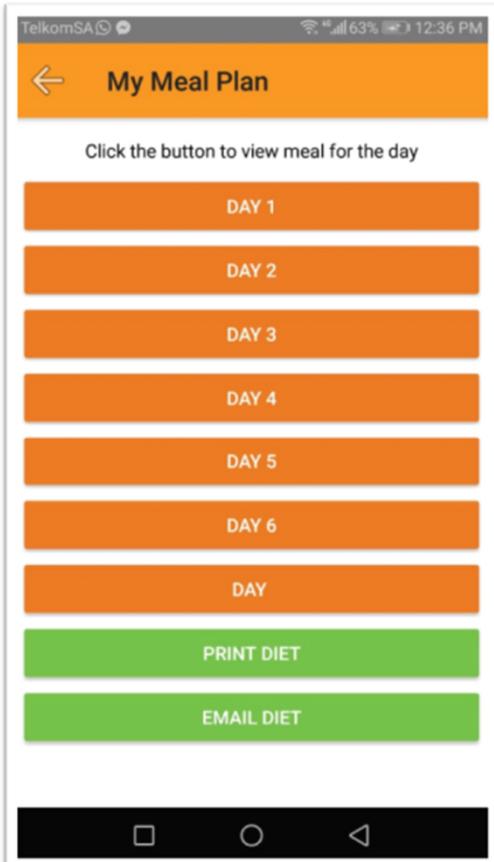
### **Update Personal Details:**

1. Choose which personal detail you want to update
2. Select the “Edit” button
3. The field will now become editable for you to enter your new details

### **VIEW MEAL PLAN OVERVIEW**

1. In the navigation bar select the “Meal plan” icon.
  - a. The “My meal plan” screen will appear.

The figure below shows the meal plan page.



*Figure 89 - Meal Plan Overview page.*

## VIEW DAILY MEAL PLAN

1. On the “My meal plan” screen click on the desired day you would like to view.
  - a. The “Meal - Day x” screen will appear.
2. Edit the meal by selecting the edit button .

The following images shows the individual day meal plan:

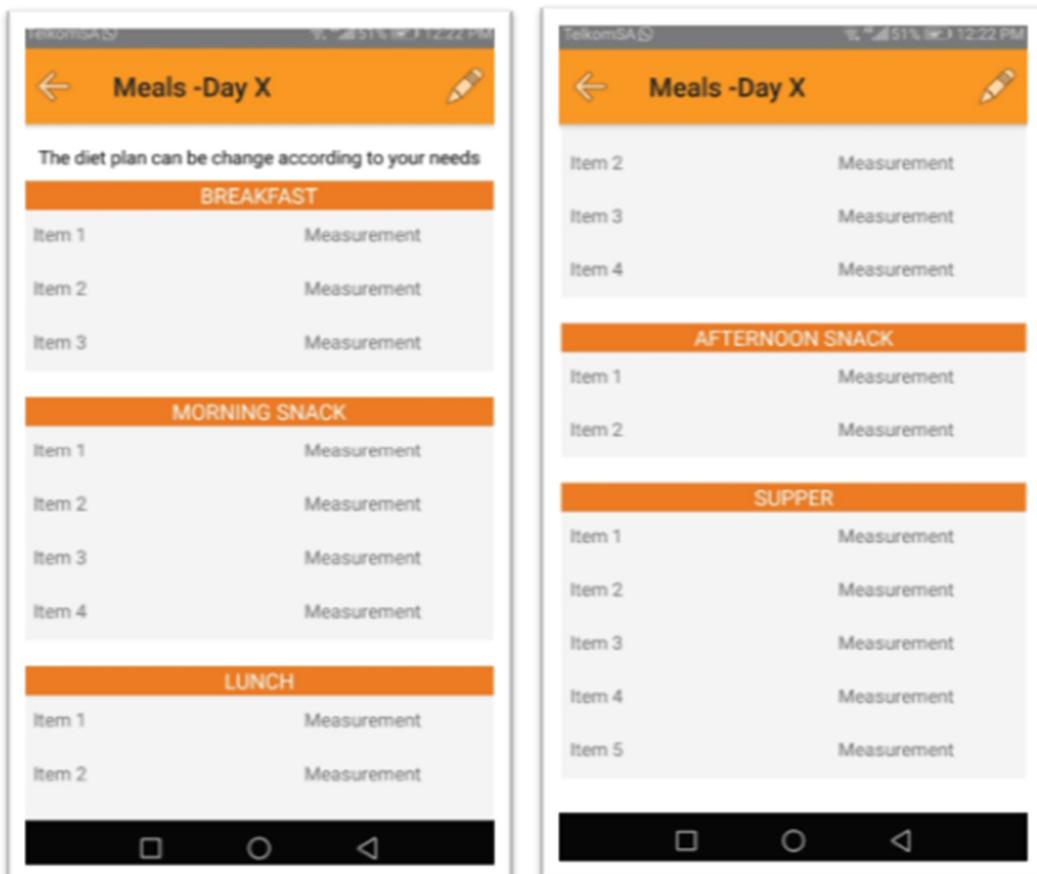


Figure 90 - Meal Plan Day View Page.

## EMAIL MEAL PLAN

1. The “My meal plan” screen click on the “Email diet” button.
  - a. The “Mail my diet” screen will appear.
2. Enter your desired email address.
3. Construct a side note for the email message.
4. Click the “Send” button.

The following figure shows the “Mail my diet” page.

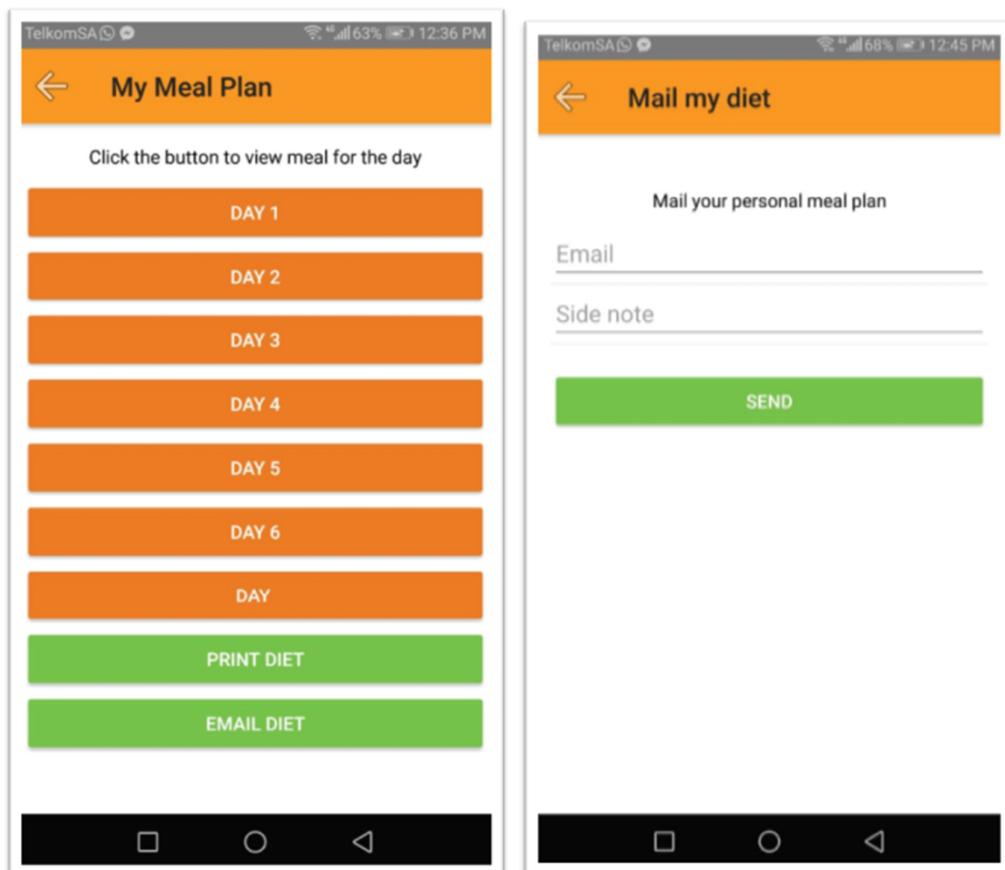


Figure 91 - Email Meal Plan Page.

## PRINT MEAL PLAN

1. In the “My meal plan” screen click on the “Print diet” button.
  - a. The “Print my diet” screen will appear.
2. Click the “Print” icon.

## VIEW SHOPPING LIST

1. In the navigation bar select the “Shopping list” icon.
  - a. The shopping list for the current user will be available.
2. The user can click the switch button when an item is bought.

The figure below shows the Food List page:

Food List	
ACE Instant High Fibre, 2.5 Kg	<input checked="" type="checkbox"/>
All Bran Flakes, 1 Kg	<input checked="" type="checkbox"/>
All Bran Instant Porridge, 1 Kg	<input checked="" type="checkbox"/>
Alphen Original Swiss Muesli, 1.5 Kg	<input checked="" type="checkbox"/>
Amasi, 200 g	<input checked="" type="checkbox"/>
Apple, 800g	<input checked="" type="checkbox"/>
Apple rings - dried, 12 pieces	<input checked="" type="checkbox"/>
Apple, Green, Raw, Unpeeled, 14	<input checked="" type="checkbox"/>

Figure 92 - Food List Page.

## SEND A MESSAGE TO USERS

1. Open the “Messages” page
2. Click on the user’s name who you would like to send a message to
3. Type your message in the message box
4. Click the “Send” button.

The following figures show the Messages page.

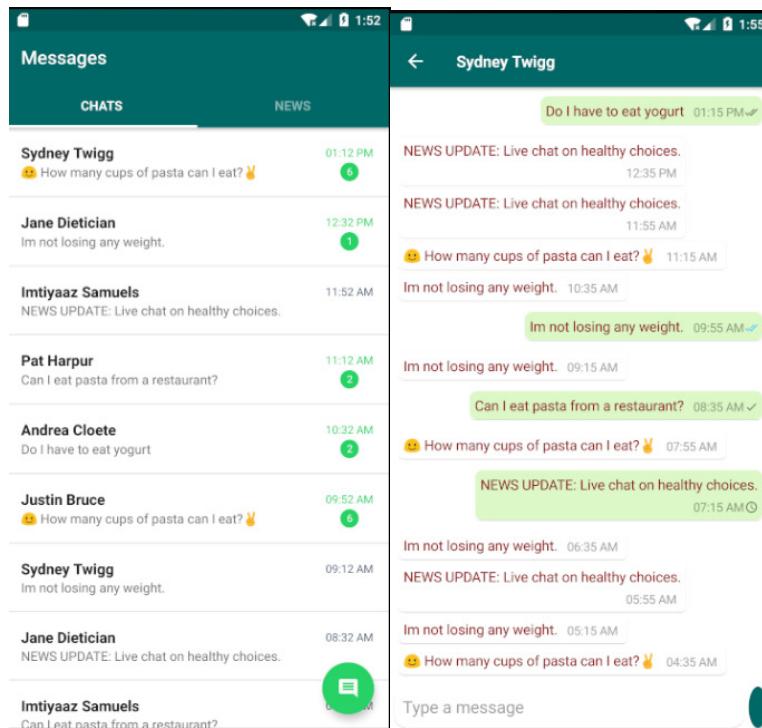


Figure 93 - Messages Page.

## VIEW A BROADCAST MESSAGE

1. Open the “Messages” page
2. Click the “News” tab
3. Click the Broadcast message to read it.

# EVALUATION REPORT

## INTRODUCTION

This document provides an evaluation of the EatForLife project. The outcome of the project is discussed by evaluating the requirements set at the beginning of the project. The effectiveness of the incremental methodology used through the development of the project is also explained along with any the changes made through the duration of the project. The relationship between the Dot Development Team and the EatForLife client are also explained. An inside to the collaboration, time management and lessons learned by the Dot Development Team are provided.

## THE FINAL SYSTEM AND THE CUSTOMER'S REQUIREMENTS

The following illustrates the functional requirements set out at the beginning of the project along with the status of the fulfillment of each functional requirement.

ID	Requirement	Priority	Fulfilment
	Registration		
FUR-1	The system allows users to register.	High	
	Login		
FUR-2	The system allows users to login.	High	
	Data Input		
FUR-3	The system allows a user to input their details.	Medium	
	Data Update		
FUR-4	The system allows a user to edit their details.	Medium	
	Communication		
FUR-5	The system allows users to send queries to and communicate with dietitians.	Medium	
	Diet Plans		
FUR-6	The system allows users to get customised diet plans based on specific user parameters.	Medium	

ID	Requirement	Priority	Fulfilment
FUR-7	The system provides information about diet plans to users.	Medium	
FUR-8	The system provides information about each item in the eating plans to users.	High	
FUR-9	The system allows users to have their health statuses analysed and a meal plan can be created accordingly.	High	
	Diet plan items awareness		
FUR-10	The system allows users to find alternative items to the substitute foods for in the diet plan.	High	
FUR-11	The system provides users with a checklist to check off food once eaten on the diet plan.	High	
	Health checks		
FUR-12	The system can calculate the BMI for users.	Medium	
FUR-13	The system can calculate the correct BMI for users based on user inputted data.	Medium	
FUR-14	The system allows users to monitor their progress with reports and graphs.	Medium	
FUR-15	The system allows users to monitor their measurement and weight history with reports and graphs.	Medium	
	Goals		
FUR-16	The system will allow the user to set weight loss goals.	Low	
	Downloads		
FUR-17	The system will allow users to download their customized 7-day meal plan directly through the mobile application.	High	
FUR-18	The system will allow users to have their 7-day meal plan emailed to their provided email addresses.	High	

ID	Requirement	Priority	Fulfilment
Downloads			
FUR-19	The system will allow the user to download their 7-day meal plan shopping list directly through the mobile application.	High	
FUR-20	The system will allow the user to have their 7-day meal plan shopping list directly through the mobile application.	High	
Other			
FUR-21	The system allows the user to reset their password	High	
FUR-22	The system allows the user to assign a profile picture to their accounts	Low	
FUR-23	The system allows the user and dietitians to log out.	High	
Admin Panel			
FUR-24	The system allows the dieticians to login to the application as an admin user	High	
FUR-25	The system will direct the dietitian to the admin panel on login	Medium	
FUR-26	The system will allow the dietician to view a list of all registered clients and their details.	Medium	
FUR-27	The system will allow the dietician to view a detailed view of each registered client.	Medium	
FUR-28	The system allows dietitians to monitor the progress of users through the admin panel.	Medium	
FUR-29	The system will allow an admin user to add another admin user through the admin page.	Low	

[REWRITE PARAGRAPH ON THE REQUIREMENTS]

In order for the EatForLife system to be successful, it is necessary for the client to provide the required resources requested by the Dot Development team. One of the most pertinent of these resources is the pre-existing database of EatForLife, as the app functionality is designed around the information present within.

Regretfully, the team have found it difficult to get the required information from the client, with repeated direct requests to be sent the database left unfulfilled. Consequently, the difficult choice was made to abandon hopes of receiving this, with the team instead deciding to create our own database from the ground up. Naturally, this created delays and pushed back our milestone dates, but without making this decision, we would not have made the progress we have since made

## THE METHODOLOGY (ITERATIVE INCREMENTAL DEVELOPMENT)

The effectiveness of the agile methodology that is being used for the development of the mobile application EatForLife is positive.

The positive aspect mainly consists of completion of deliverables and group dynamics. Completion of deliverables refers to all task, documentation, and coding are done rapidly while meeting certain standards and guidelines. Coding tasks is reviewed and updated to achieve better coding standards or the fixing of errors. Documentation tasks is reviewed and edited to account for any changes made to the guidelines by the clients, lectures or group members.

The improvement that could have been implemented using the agile methodology where the balancing of starting new task verses perfecting the previous task.

The changes that were made to the agile methodology project plan were mainly changes in due dates for tasks. The managing of those changes was to set task deadlines a few days before the new due date as a buffer. This buffer served as a period to complete, style, and perfect the task, as well as gather input from other members on the current task.

## THE CUSTOMER

As our customer was based in Johannesburg, all communication was done digitally, primarily via email. Communication between the group and customer was mainly facilitated by the group leader, Sydney. Our group approached the customer with the idea for the EatForLife mobile application of their existing web-based system, thus we had proposed the requirements to the customer. In terms of feedback on our requirements and deliverables, we were given a large amount of creative freedom over all aspects of the product, from determining the product requirements to user interface design and adding any features we felt necessary for the application. After the initial system proposal, we were mainly in communication with the developers of the existing system. It became evident that the EatForLife mobile application was a low priority project - and we had some hurdles in obtaining the relevant details we needed to proceed with development.

Overall, as a group, we found there were challenges with working with an out of town client - as communication is key to the success of any project.

## GROUP DYNAMICS AND TEAM COLLABORATION

The group collaborated well in allocating workloads and ensuring that we had adequate support. At the beginning of each increment, work was divided and assigned to each group member for them to complete, as well as each group member assigning themselves a set of functionalities to code within the application. The group worked efficiently to ensure all deliverables were completed and submitted on time and communicate any problems we had in our assigned work. Throughout the year, the group has remained conflict free with all members getting along and working well together. As a group, we took the approach of each member is responsible for their own work and responsible for completing it - however, we all offered each other support and help if it was needed. Our strength as a group lay in asking each other for help when it was needed in order to ensure that all work would be completed on time.

The group primarily communicated via a WhatsApp group, weekly meetings to check in with each other's progress, and work was assigned and completed on Google Docs and GitHub for easy collaboration.

## TIME MANAGEMENT

All documentation and every deliverable were handed in on time without any issues or delay. In terms of spreading the load across the entire team in order to do so, the team has been very efficient in order to minimise the time required to complete a deliverable. The team has done well in the aspect of balancing between the other modules' workload and the Software Project (SP). In the category of physical work or development for the project itself; the team has been lacking in time management, however, this may be contributed by the fact that the client the project is being made for has quite a delay on responding to any requests or communication. The delay in coding is due to a pause in connecting to the client's already existing database; for this required access to their private information. However, to make time management more efficient in this regard, the team could have made use of the temporary database that was created.

## LESSONS LEARNED

### JUSTIN BRUCE

I have learnt that after the experiences of last year's SP, a well-coordinated, communicative, working team is what best creates a well-produced project. When working with/for a client, the biggest liability or hurdle is the client themselves. Attempting to get information about the specifications or access to their current system alone took several weeks; which, in turn, delayed progress for the project. In conclusion, what I have learnt is that: A good team is needed to produce a good product; communication amongst team members and clients is required to efficiently develop a project, and I have learnt how to develop a mobile application for both iOS and Android using NativeScript and NativeScript Sidekick.

## SYDNEY TWIGG

This project was my first experience in mobile application development, I learnt a great deal on the processes that goes into planning, developing and testing a mobile application. I also gained a great deal of experience in using a new, lesser known frameworks with the use of NativeScript. This taught me how to rely on my own coding abilities and reading into documentation and debugging, rather than blindly following a tutorial. I had very little experience with using JavaScript and XML before this project, and I feel that this project has allowed me to improve my abilities in both these languages and in mobile development.

This was my first experience in being a group leader and managing people - I found that good communication helped a great deal in this project, with weekly check-ins to ensure that all members were on track with their allocated work. As the main point of contact between our group and our customer, I had to take on the responsibility of ensuring that all questions my group had were addressed to the client and to follow up with any queries.

Throughout this project, I have honed my ability to balance workloads and manage time effectively in both an individual and group setting. In particular, I feel that the experience of managing a group submission during a busy period for deliverables and submissions in other modules will stand me in good stead with my future studies.

## ANDREA CLOETE

I must admit the experience I have gained from the agile development approach for the development of the EatForLife software had a significant impact on my life. The agile approach which is flexible to adjustment as needed has become my mantra in dealing with the everyday stressful situations in my life. Even though it is necessary for a project to have an outline to work from and to have goals and expectations to aim towards, I have come to learn the value in being flexible, adaptable and to take situations as they come my way. I have learned software development is unpredictable since there are too many factors that cannot be predicted that come into play when having to state what will happen accurately. However, what I can embrace is that there are endless opportunities for growth, improvement, and success.

## IMTIYAAZ SAMUELS

The lessons learnt in the software project module this year are the requirements to complete a mobile application for a client within a group. The requirements are mainly team communication, mobile application development and time management. The lessons learnt for communication are each group member needs to have a certain amount of respect and understanding when communicating. Communication also needs to be effective, this refers to question that are asked is answered. Also, the asking of the question "Does anyone need help?" or the statement "I need help" can greatly decrease time need to complete tasks. The communication between group members and client is very important to completion of a successful project. The lessons learnt for mobile application development is the coding style to code for both IOS and android using NativeScript and NativeScript Sidekick. As well as all planning and documentation need for the development of the mobile application.

The lessons learnt for time management is how to use a minimum amount of time efficiently and effectively to archive a successful result.

## CONCLUSION

The development of the EatForLife application, as with any project, was met with some hiccups along the way - but ultimately the group worked well together in order to ensure that all deliverables were completed on time, and assist each other through any hurdles we faced. Both the Agile approach and NativeScript mobile application development were new to all members of the group, however, we took it in our stride and learnt to adapt to new situations as a group.

As a group, we learnt the importance of good communication and teamwork in a project - with both team members and clients. We have all learnt a great deal in working with real-world clients, as well as expanding our knowledge on mobile application development.

## CUSTOMER SIGN-OFF

X

---

Matthew Grossett  
CEO, EatForLife

X

---

Date

X

---

Sydney Twigg  
Group Leader, DOT Development

X

---

Date



dot development

## REFERENCES

- Bass, L., 2012. *Functionality / Understanding Quality Attributes in Software Architecture*. [Online]  
Available at: <http://www.informit.com/articles/article.aspx?p=1959673&seqNum=2>  
[Accessed 09 05 2018].
- Bryant, R. E. & O'Hallaron, D. R., 2011. *Computer Systems - A Programmers Perspective*. 2nd ed. s.l.:Pearson.
- Campbell, A., 2015. *What the Heck is an “App”?*. [Online]  
Available at: <https://smallbiztrends.com/2011/03/what-is-an-app.html>
- Connolly, T. M. & Begg, C. E., 2004. *Database Solutions A Step-by-Step Guide To Building Databases*. 2nd ed. Harlow: Pearson Education.
- Cress, K., 2009. *5 Types of Information Systems*. [Online]  
Available at: <https://www.mindmeister.com/37310006/5-types-of-information-systems>
- Dave, J., 2017. *How To Build A Mobile App Requirements Document*. [Online]  
Available at: <https://clearbridgemobile.com/how-to-build-a-mobile-app-requirements-document/>  
[Accessed March 2018].
- EatForLife, 2017. *EatForLife*. [Online]  
Available at: <http://www.eatforlife.co.za/home.html>  
[Accessed 06 03 2018].
- EatForLife, 2018. *About Us*. [Online]  
Available at: <http://www.eatforlife.co.za/About.html>  
[Accessed March 2018].
- EatForLife, 2018. *EatForLife*. [Online]  
Available at: <http://www.eatforlife.co.za/home.html>  
[Accessed March 2018].

FutureLife, 2016. Eat For Life. [Online] Available at: <http://futurelife.co.za/partner/eat-for-life/> [Accessed March 2018].

Haag, L., 2017. *Mobile Development Kit (formerly SAP Enterprise App Modeler)*. [Online] Available at: <https://blogs.sap.com/2017/05/24/sap-enterprise-app-modeler/> [Accessed July 2018].

Haselmayr, M., 2018. *7 Reasons Why Your Business Needs a Mobile App*. [Online] Available at: <https://www.allbusiness.com/7-reasons-business-needs-mobile-app-19179-1.html> [Accessed March 2018].

Kloosterman, V., 2016. *What are the 5 Risk Management Steps in a Sound Risk Management Process?*. [Online] Available at: <http://continuingprofessionaldevelopment.org/risk-management-steps-in-risk-management-process/> [Accessed March 2018].

Krosnick, J. A. & Presser, S., 2010. *Handbook of Survey Research*. 2nd ed. s.l.:Emerald Group Publishing Limited.

LucidChart, 2018. UML Use Case Diagram Tutorial. [Online] Available at: <https://www.youtube.com/watch?v=zid-MVo7M-E> [Accessed August 2018].

MockingBot, 2018. MockingBot - Features. [Online] Available at: <https://mockingbot.com/features> [Accessed August 2018].

MySQL, 2018. MySQL Workbench. [Online] Available at: <https://www.mysql.com/products/workbench/> [Accessed August 2018].

NativeScript, 2018. *Create native iOS and Android apps with JavaScript*. [Online] Available at: <https://www.nativescript.org/> [Accessed July 2018].

NativeScript, 2018. *NativeScript Advanced Setup: macOS.* [Online] Available at: <https://docs.nativescript.org/start/ns-setup-os-x> [Accessed July 2018].

NativeScript, 2018. *NativeScript Advanced Setup: Windows.* [Online] Available at: <https://docs.nativescript.org/start/ns-setup-win> [Accessed July 2018].

NativeScript, 2018. *NativeScript for Visual Studio Code.* [Online] Available at: <https://www.nativescript.org/nativescript-for-visual-studio-code> [Accessed August 2018].

Nielson, J., 1995. *10 Usability Heuristics for User Interface Design.* [Online] Available at: <https://www.nngroup.com/articles/ten-usability-heuristics/> [Accessed March 2018].

Pressman, R. S. & Maxim, B. R., 2015. *Software Engineering: A Practitioner's Approach.* 8th ed. New York: McGraw-Hill Education.

Robertson, S. & Robertson, J., 2006. *Mastering the Requirements Process.* s.l.:Pearson Education.

Rouse, M., 2016. *Risk Management.* [Online] Available at: <http://searchcompliance.techtarget.com/definition/risk-management> [Accessed March 2018].

Rouse, M., 2017. *TechTarget.* [Online] Available at: <https://searchsoftwarequality.techtarget.com/definition/agile-software-development> [Accessed 09 05 2018].

Rouse, M., 2017. *What is Android OS?.* [Online] Available at: <http://searchenterpriselinux.techtarget.com/definition/Android> [Accessed 09 05 2018].

Rouse, M., 2017. *What is mobile app?.* [Online] Available at: <http://whatis.techtarget.com/definition/mobile-app> [Accessed 09 05 2018].

- Saleh, K. A., 2009. *Software Engineering*. 1st ed. s.l.:J. Ross Publishing.
- Schwalbe, K., 2014. *Information Technology Project Management*. 7 ed. s.l.:Course Technology.
- Schwalbe, K., 2015. *Information Technology Project Management*. s.l.:Cengage Learning.
- Scrum Alliance, 2018. SCRUM ALLIANCE. [Online] Available at: <https://www.scrumalliance.org/learn-about-scrum> [Accessed 09 05 2018].
- Shapshak, T., 2017. *South Africa Has 21 Million Internet Users, Mostly On Mobile*. [Online] Available at: <https://www.forbes.com/sites/tobyshapshak/2017/07/19/south-africa-has-21m-internet-users-mostly-on-mobile/#719e4ea21b2d> [Accessed March 2018].
- Shore, J., 2016. *NativeScript framework eases cross-platform app development woes*. [Online] Available at: <https://searchcloudapplications.techtarget.com/podcast/NativeScript-framework-eases-cross-platform-app-development-woes> [Accessed July 2018].
- Sommerville, I., 2011. *Software Engineering*. 9th ed. Boston: Pearson.
- Sonmez, J., 2010. *Back to Basics: What is an Interface?*. [Online] Available at: <https://simpleprogrammer.com/2010/11/02/back-to-basics-what-is-an-interface/> [Accessed 09 05 2018].
- So, Y., 2017. *Designing for Mobile Apps: Overall Principles, Common Patterns, and Interface Guidelines*. [Online] Available at: <https://medium.com/blueprint-by-intuit/native-mobile-app-design-overall-principles-and-common-patterns-26edee8ced10> [Accessed September 2018].
- Stair, R. M. & Reynolds, G. W., 2016. *Principles of Information Systems*. Custom ed. Hampshire: Cengage Learning.

Thompson, K., 2012. *cPrime*. [Online]  
Available at: <https://www.cprime.com/2012/09/when-to-use-scrum/>  
[Accessed 09 05 2018].

Vagias, W. M., 2006. Likert-type Scale Response Anchors.. *Clemson International Institute for Tourism & Research Development, Department of Parks, Recreation and Tourism Management* .

Valacich, J. S., George, J. F. & Hoffer, J. A., 2015. *Essentials of Systems Analysis and Design*. 6th ed. Harlow: Pearson.

Viswana, N., 2017. *What Is a Mobile Device?*. [Online]  
Available at: <https://www.lifewire.com/what-is-a-mobile-device-2373355>  
[Accessed 09 05 2018].

WebFinance Inc, 2018. *Risk*. [Online]  
Available at: <http://www.businessdictionary.com/definition/risk.html>  
[Accessed March 2018].

Zandbergen, P., 2018. *What is Computer Hardware? - Components, Definition & Examples*. [Online]  
Available at: <https://study.com/academy/lesson/what-is-computer-hardware-components-definition-examples.html>  
[Accessed July 2018].