Cadence Health

Design Document

Mobile Application

Version 1, 7/10/14

Table of Contents

1. System Requirements Specification 3

1.1. Introduction 3

1.1.1. Purpose 3

1.1.2. Scope 3

1.2. Definitions, Abbreviations, Acronyms 4

1.3. References 4

1.4. Overview 4

1.5. Context Diagram 6

1.6. Overall Description 6

1.6.1. Product Perspective 6

1.6.2. Product Functions 7

1.7. User Classes / Characteristics 7

1.8. Operating Environment 8

1.9. User Documentation 8

1.10. Requirements 9

1.10.1. Functional Requirements 9

1.10.2. Performance Requirements 12

1.11. Design Implementation Requirements / Constraints 13

1.12. Other Non-functional Requirements 14

1.13. Usability Requirements 16

2. Use Case Diagram 17

3. Use Case Descriptions 18

4. System Design 21

4.1. System Architecture 22

4.2. Storage/Persistent Data Strategy 22

4.3. Trade-offs and Choices 22

4.4. Concurrent Processes 23

4.5. Package Diagram 23

5. User Interface Layouts 24

6. Program Navigation Diagram 25

7. Data Definitions 26

8. Analysis and Design Class Diagram 27

8.1. Sequence Diagram 28

8.2. State Diagram 30

9. Requirements Traceability Matrix 31

10. Design Assumptions 33

# System Requirements Specification

## Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) is to document and communicate the functional and non-functional requirements for the mobile APPLICATION. The SRS provides a structured collection of information to allow design and development on the APPLICATIONs requirements, constraints, functions, characteristics and overall perspective. Furthermore, this SRS defines the planned system features and capabilities, establishing an understanding between the product developers and the product clients of the new APPLICATION.

## Scope

As noted in the feasibility study, the developing team will develop a mobile APPLICATION for the smartphone market that will allow users to search, share and take snapshots of healthy meals.

The primary objective of developing a healthy food app is to promote a healthy lifestyle to the general public as well as providing the necessary tools for individuals to follow a correct healthy lifestyle. To achieve this, the APPLICATION will allow the functionality of an ORAC rating which will rate the health standard of the meal. A list of ingredients and the meals location can also be viewed.

By developing an app that promotes healthy living, it provides benefits towards not only its users but also the client organisation Cadence Health. Users are benefited by this app through cost savings, time savings and an overall better wellbeing to society. The client also benefits through this app as it provides a better communication medium towards its users and the general public. In return this can potentially improve the brand image of Cadence Health.

The goals and objective listed above will need to be achieved through communication between stakeholders such as the client, target audience and the group members.

This report outlines the planned initial release of the APPLICATION. Initially the app will be expected to grow in information overtime as users will be able to share their meals and data onto the app. Further updates can also be added and be built upon. Before the initial release, there will be two separate prototypes that will outline the basis and functionality of the app which will be further improved through testing. It is expected that the app will provide users with an increased understanding of health tips, meal information and health awareness which represents the foundations for achieving the above benefits and objectives.

## Definitions, Abbreviations, Acronyms

The following definitions apply to the system. Where these appear capitalised, the following is implied.

USER- A person who has downloaded the APPLICATION and is using it

ANDROID- Google Android operating system of at least version 4.1.1

APPLICATION- An ANDROID APPLICATION that is downloaded and run on the USERs device

TAG- A categorical name attached to a picture for the purpose of identification by a USER

GEOLOCATION- The technique of identifying the geographical location of where a USER had their meal, by means of information processed via the internet

PROXIMITY- The closeness of a USER to a GEOLOCATION

USER RATING- The rating given to a meal by USER in a scale from 1 to 10, with 0 being the worst and 10 being the best.

ORAC- Oxygen Radical Absorbance Capacity

ORAC Rating Method- A method to calculate the ORAC Rating of a meal, by the combining of ingredients within the meal.

## References

Android. (2014). *developer.android*. Retrieved from developer.android.com/distribute/essentials/quality/core.html

*developer.android*. (2014). Retrieved from developer.android.com/about/dashboards/index.html?utm\_source=ausdroid.net

## Overview

This report will demonstrate the proper requirements for software development, which will provide further understanding of the system when designing and developing the APPLICATION. It is divided into three clauses. Clause 1 introduces the new proposed APPLICATION and provides a detailed overview of the purpose of developing this new APPLICATION. This is then followed on with the overall scope of the system requirements specifications. The scope will provide relevant benefits, goals and objectives to business strategies. A list of definitions, abbreviations, acronyms and references will also be noted.

Clause 2 describes the overall description of the proposed APPLICATION. This includes a brief of the product perspective, the product functions, the user classes & characteristics, operating environment and user documentation. The product perspective describes the context of system including the overall use of the APPLICATION, the components of the APPLICATION and how it connects with the system objectives. Product functions will summarise the major functions of the APPLICATION. User classes will allow the readers to understand the system APPLICATION in clear context. Furthermore, the operating environment will note the necessary hardware and software requirements. Finally the user documentation discusses how APPLICATION users will be guided through the use of tutorials via the system.

Clause 3 discusses the requirements of the system and goes through the functional and non-functional requirements. The functional requirements will be a dot point list describing what the APPLICATION must be able to do. Performance requirements and design implementation requirements will also be noted, constraints will also be followed. Clause 3 will also go through any non-functional requirements describing how the proposed system will be able achieve system objectives.

## Context Diagram

Users Smart Phone

<<Interface>>

USER interface

Camera

Web Server

App

Web Storage

<<Interface>>

Web Browser

Storage

## Overall Description

## Product Perspective

The mobile APPLICATION is designed to be a standalone APPLICATION running on an Android smart phone, which will allow users to better track and manage their meals so they can become healthier and improve their wellbeing.

It will be able to have users simply and easily take a photo of their meal, tag ingredients, and upload it to social media.

From data inputted it will generate and ORAC from these TAGS, and generate a shopping list from the TAGGED ingredients from a selectable amount of meals saved.

The images uploaded to social media will also have the following information: location, personal user rating, and the type of food. The data can be queried and sorted by information uploaded with the image.

## Product Functions

The APPLICATION shall positively change people’s eating patterns using the ORAC rating method. The APPLICATION shall also allow better tracking of their meals they are eating, or would like to eat by using the ORAC rating method. It shall automatically generate an ORAC rating, by reading the information input by the USER.

The APPLICATION shall use publicly available information on the ORAC rating system, using a database populated with ingredients and their ORAC RATING. The APPLICATION will provide consistent information based on this method, ensuring it is better known within the health industry. The USER generated data will populate the web server database with relevant information about healthy meals.

The APPLICATION shall allow the USER to capture images of their meals and TAG the meal. The USER may enter the Geo-location and personal rating to further provide better information about the meal. It will allow users to upload their saved meals to social media.

The APPLICATION shall provide the option to generate a list of ingredients for user selected meals. The APPLICATION shall allow users to group their meals in various ways, including TAG, GEOLOCATION, PROXIMITY, ORAC RATING, and USER RATING.

The APPLICATION shall allow users to search through the database of meals by the following categories; TAG, GEOLOCATION, PROXIMITY, ORAC RATING, USER RATING.

The APPLICATION shall also positively change people’s eating patterns by providing health tips and provisions for further health and wellbeing advice.

## User Classes / Characteristics

There are three main types of users involved with the APPLICATION.

Type 1 - Administrators: Administrators are responsible for maintaining the backend of the APPLICATION. The main focus will be on moderating the types of photos that are being shared using the APPLICATION and whether they are relevant types of food and/or drink.

Type 2 - Users Interested in Improving Their Health: Anyone that purchases the APPLICATION and is interested in improving their health. The majority of functionality within the APPLICATION is focused towards improving the health of end users.

Type 3 - Health Industry Representatives: Health industry representatives such as personal trainers. This user type uses the APPLICATION to benefit other non-users.

## Operating Environment

The APPLICATION must be supported by Android and IOS. The software for the APPLICATION must be written in a language that allows for it to be compatible for possible web browser use.

The APPLICATION must be downloadable by Android's 'Play Store' at all times. A direct link through the Clients website will be supplied for easy access to the downloadable APPLICATION.

This allows for all software and hardware platforms to access the system without compatibility issues and will also prevent vendor lock in and support future upgrades.

Future implementation for an APPLICATION based interface may also be considered, with the possibility of running on Windows or IOS devices.

The APPLICATION data-storage and server-side components will require a web server to store and access data of the APPLICATION’s users. This will allow information to be stored from any user and populate a searchable and sortable database for ALL users.

It is not anticipated that the APPLICATION will place an onerous strain on server resources and therefore the use of a web server hosting the database will be adequate until such a time that the strain deems it necessary to upgrade the server-side components.

The APPLICATION must be available for 24 hour use, 7 days a week with 100% certainty. Updates will be released and notify the users to download at their leisure.

The APPLICATION must be available to use both offline and online, noting that full functionality will only be supported while online.

Note that future versions of the APPLICATION may demand enhanced hardware and/or the upgrade of the server-side components.

## User Documentation

The APPLICATION shall include an in-app interactive tutorial page that walks the user through on how to use the apps features. This tutorial will also run automatically on the user’s first run of the APLICATION.

## Requirements

## Functional Requirements

APPLICATION v1.0 Function Requirements

* R1.1: The APPLICATION shall allow a user to be able to capture images and TAG images.
* Purpose: To allow for a database of meals acquired by the user.
* Test: There must be at least 5 different test cases per user.

These users will be spread out on various devices.

%100 of images must be able to be captured, TAGGED per user.

* R1.1.1: The APPLICATION shall allow a user to be able to use gallery images and TAG images.
* Purpose: To allow for a database of meals acquired by the user.
* Test: There must be at least 5 different test cases per user.

These users will be spread out on various devices.

%100 of images must be able to be captured, TAGGED per user.

* R1.2: The APPLICATION shall recognise either a common TAG, or new TAG by a user.
* Purpose: To allow for auto filling, and pre-generate information about the meal.
* Test: There must be at least 200 different common test cases, 10 unique cases.

Of which all unique cases must be added to the APPLICATIONs online database.

All common cases must be recognised by the APPLICATION.

* R1.3: The APPLICATION shall allow a user to add the GEOLOCATION of the meal.
* Purpose: To provide a location for the captured meals for future visitation by the user, or other users.
* Test: There must be at least 5 different test cases per user.

These users will be spread out on various devices.

%100 of images must be able to have the correct GEOLOCATION added to its details.

* R1.4: The APPLICATION shall be able to access Google's Maps and GEOLOCATION function
* Purpose: To provide an accurate GEOLOCATION for captured meals that can be accessed through Google Maps.
* Test: There must be at least 5 different test cases per user.

These users will be spread out on various devices.

100% of images must be able to have correct GEOLOCATION added to its details.

* Assumption: If the internet is not accessible then GEOLOCATION will not work. Therefore it is assumed that internet is on for this functionally to work correctly.
* R1.5: The APPLICATION shall allow a user to edit an ingredients list in relation to the meal after identifying the meal.
* Purpose: To generate a more accurate list of ingredients a more accurate ORAC RATING.
* Test: There must be at least 50 test cases of unedited meals.

100% of ingredient lists must be accurate and consistent to the TAGGED meals.

There must be at least 50 test cases of edited meals.

100% of ingredient list must be editable.

* Assumption: There is a general database that handles the pairing of meals and ingredient lists.

* R1.6: The APPLICATION shall generate an ORAC RATING based upon a meal and/or ingredients.
* Purpose: To give an appropriate and accurate ORAC RATING along a certain scale to provide helpful information to the user.
* Test: There must be at least 200 test cases.

There must be at least 20 duplicate cases.

There must be at least 100 similar test cases.

* R1.7: The APPLICATION shall allow the user to search through a database populated by their own meals.
* Purpose: The user must be easily able access earlier meals for future use.
* Test: There must be at least 20 different test cases per user.

These users will be spread out on various devices.

100% of meals must be accessible through the inbuilt file system.

* R1.8: The APPLICATION shall allow the user to sort through meals based on the user’s GEOLOCATION.
* Purpose: To provide the user with meals that is close to their current GEOLOCATION.
* R1.9: The APPLICATION shall allow the user to sort through meals based on the ORAC RATING.
* Purpose: To provide the user with meals that is high on the ORAC RATING for healthy eating.
* R1.10: The APPLICATION shall allow the user to sort through meals based on a given GEOLOCATION.
* Purpose: To provide the user with meals that are close to a given GEOLOCATION.

* R1.11: The APPLICATION shall allow the user to sort through meals based on the TAG name.
* Purpose: To provide the user with meals that are of the same variety as their input TAG.

* R1.12: The APPLICATION shall allow the user to search/sort through a database of all meals that are within the online database
* Purpose: The user must easily be able to access all meals shared across other users for future use, to create meal plans, and find meals base on sort requirements easily.
* Test: There must be at least 5 users.

There must be at least 100 meals that vary according to ORAC RATING, USER RATING, TAGs, and GEOLOCATION.

100% of meals must be viewable and accessible through all users.

* Assumption: There is an online database that manages the uploading and downloading of meals either TAGGED by users of the APPLICATION of added by the moderator of the APPLICATION.
* R1.13: The APPLICATION shall allow the user to group together meals and export either individual or groups of ingredient lists for meals.
* Purpose: The user will be able to use the list to shop for ingredients to maintain a healthy diet based on the APPLICATIONs functions.
* Test: There must be at least 5 group test cases.

There must be at least 20 individual test cases.

These must be exported into each format.

* R1.14: The APPLICATION shall allow the user to edit/remove meals/groups of meals from the APPLICATION.
* Purpose: To give control of the file system to the user.
* Test: There must be at least 5 group test cases of editing.

There must be at least 5 group test cases of removing.

There must be at least 20 individual test cases of editing.

There must be at least 20 individual test cases of removing.

100% must show either the edit of removal was completed correctly.

* R1.15: The APPLICATION shall provide healthy tips to the user.
* Purpose: To provide information to the user to better maintain a healthy lifestyle.

## Performance Requirements

APPLICATION v1.0 Performance Requirements

* R1.16: The APPLICATION must load the home page in less than 4 seconds
* Purpose: To minimise User waiting times
* Test: There must be at least 5 test cases per user.

100% of test cases must load in under 4 seconds.

60% of test cases must load under 2 seconds.

* R1.17: The APPLICATION does not freeze, crash, or function in an abnormal way on any targeted device.
* Purpose: To give the user a positive experience with the APPLICATION.

98% of devices must not crash or function in an abnormal way.

* R1.18: The APPLICATION displays all UI elements without noticeable blurring, or any other forms of distortion.
* Purpose: To leave a good visual impression on the User.

98% of the UI must not have blurring or other noticeable distortions.

* R1.19: The content that this app delivers must be displayed proportionately on all targeted devices, including tablets.
* Purpose: To ensure that all users can utilise the functionality of this APPLICATION, regardless of the Android device they are using.

98% of devices must be able to access the app.

* R1.20: The APPLICATION must be responsive to all button presses within 1 second of being pressed.
* Purpose: To give the user a positive experience using the APPLICATION.
* R1.21: The generation of an ORAC score should not exceed 4 seconds.
* Purpose: To deliver information promptly,
* R1.22: Search results should load in less than 3 seconds.
* Purpose: To promptly assist with the users search.
* R1.23: Camera should launch within 2 seconds an APPLICATION request.
* Purpose: To allow user access to the camera seamlessly without delay.
* R1.24: Generation and exportation of shopping list should take no longer than 3 seconds.
* Purpose: To allow user quick access to information.

## Design Implementation Requirements / Constraints

* R1.25: The APPLICATION must be delivered to the client no later than the 4th December 2014
* R1.26: The APPLICATION must be compatible with Android devices running Android 4.1.1 and above
* Purpose: To eliminate compatibility issues, as 75% of Android devices run on 4.1.1 or above.
* R1.27: The APPLICATION shall attempt to be compatible with Android devices running older operating systems.
* Purpose: To allow users of older devices to access the APPLICATION
* R1.28: The APPLICATION must be compatible with a variety of screen sizes, and scale all elements accordingly.
* Purpose: To cater to users who are using tablets or small devices
* R1.29: The APPLICATION colour scheme must match that of the Cadence health website
* Purpose: To identify the APPLICATION with Cadence health, and ensure continuity across their product range
* R1.30: The APPLICATION must follow the Android Design guidelines and conform to common UI patterns
* Purpose: To conform with guidelines set out by Google
* R1.31: The APPLICATION does not redefine the expected function of a system icon (such as the "home" or "back " buttons)
* Purpose: To avoid confusion with the users.
* R1.32: The APPLICATION does not misuse Android UI patterns, which would make a standard user become confused.
* Purpose: To avoid confusion with the users.
* R1.33: The APPLICATION must function normally when installed on an SD card
* Purpose: To allow users to store the APPLICATION on an SD card.
* R1.34: The APPLICATION must request the absolute minimum permissions that it needs to operate normally.
* Purpose: To ensure that the APPLICATION does not accidentally access unauthorised system resources
* R1.35: The APPLICATION should support both landscape and portrait orientations
* Purpose: To provide an adaptable view of the APPLICATIONs content
* R1.36: The APPLICATION must adhere to the terms of the Google play developer content policy found at https://play.google.com/about/developer-content-policy.html
* Purpose: To allow the APPLICATION to be sold on the Google Play store.

## Other Non-functional Requirements

APPLICATION v1.0 Non-Functional Requirements

* R1.37: The APPLICATION shall improve user health
* Purpose: To prove the need for this APPLICATION
* Test: Regular examination of database traffic and reviews of the APPLICATION will determine if it is providing assistance.

Examining the general health of state could possibly be used at a later stage.

* R1.38: The APPLICATION shall be available to users 24/7 with 100% reliability

Purpose: This is to ensure that all users will have access to the APPLICATIONs functionality at all times

* R1.39: The APPLICATION shall be available to users without the requirement of installing additional software.
* Purpose: This is to ensure that all users may easily access the APPLICATION.
* R1.40: The APPLICATION shall provide accurate an ORAC RATING to users
* Purpose: To provide accurate statistics of meals to the users which will help the user make informed choices.
* Test: R1.6 will provide these tests.
* R1.41: The APPLICATION shall provide accurate ratings to users

Purpose: To provide accurate reviews of meals to help the user make informed choices.

* Test: There must be at least 10 meals created.

There must be at least 200 reviews on each meal generated

* R1.42: The APPLICATION shall retain user anonymity
* Purpose: To provide anonymity to the user, as personal details being shared or access are not necessary.
* R1.43: The APPLICATION shall record all stored meals for other users use.
* Purpose: To provide a largely populated database of meals for the user to search and sort through at any time.
* Test: There must be at least 20 users

These users will be spread over multiple devices

There must be at least 10 meals submitted by each user

These meals must have attached GEOLOCATION

These users must be able to search, sort and view all meals by all users.

* R1.44: The APPLICATION shall update its database to further identify information being input by the user
* Purpose: To allow for quicker
* R1.45: The APPLICATION shall allow for future functionality to be added.
* Purpose: To allow for development on other platforms and to add additional functionality

## Usability Requirements

* R1.46: The APPLICATION shall be simple for anyone to use

Purpose: To provide an easy and useful application to the user

* R1.47: The APPLICATION shall provide a help tool within the application

Purpose: To provide the user with the means to learn how to utilise the application

* R1.48: The APPLICATION shall utilise similar layouts to common Android icons

Purpose: To provide a familiar application for the user

# Use Case Diagram

Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:Use Case Model.pdf

# Use Case Descriptions

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **Tag New Meal** | |
| **System** |  | |
| **Actors** | **User** | |
| **Goal**  <a longer statement of the goal in context if needed> | To Create a meal | |
| **Trigger**  <the action upon the system that starts use case> | The user selects Tag New Meal | |
| **Preconditions**  <what we expect is already the state of the world> | 1. The user has sufficient storage 2. The user has a phone with a functional camera 3. The user is in front of a meal | |
| **Success End Condition**  <the state of the world upon successful completion> | A new meal is saved | |
| **Failed End Condition**  <the state of the world if goal abandoned> | A meal is not fully created or is aborted before saving of meal | |
| **Primary Actors;**  **Secondary Actors** | User | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | App is open |
| 1.b | User selects Tag new meal |
|  |  |
|  | |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** |  |
| 1 | **Branching** |
|  | A1 | Camera application opens |
|  |  | Image is taken |
|  | A2 | User selects meal from gallery |
|  | A3 | **Branching** |
|  |  | Image is saved |
|  |  | Image is tagged by user |
|  | 1.c | Tagged image is saved |
|  | 1.d | Meal is created |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **View Local Meals** | |
| **System** |  | |
| **Actors** | **User** | |
| **Goal**  <a longer statement of the goal in context if needed> | User is able to view their locally saved meals | |
| **Trigger**  <the action upon the system that starts use case> | The user selects View Meals | |
| **Preconditions**  <what we expect is already the state of the world> | 1. The user has 1 or more locally saved meal(s) | |
| **Success End Condition**  <the state of the world upon successful completion> | The user views locally saved meals on their mobile | |
| **Failed End Condition**  <the state of the world if goal abandoned> | The user is unable to view locally saved meals | |
| **Primary Actors;**  **Secondary Actors** | User | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | The User selects View Meals |
| 1.b | The application then searches the local device |
| 1.c | The application then returns all locally saved meals |
| 1.d | The locally saved meals are viewed |
|  |  |
|  |  |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** | **Branching** |
| a | The user selects edit meal on an open meal |
| A1 | The user then uploads and saves to web database |
| A1.1 | The application gets the geo-location for the image and ads it to the tags |
| A2 | The user generates the ORAC rating for the meal selected |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Project** |  | |
| **Use Case** | **Searches Meals** | |
| **System** |  | |
| **Actors** | **Application** | |
| **Goal**  <a longer statement of the goal in context if needed> | The user searches the web database for Meals according to User selected tags | |
| **Trigger**  <the action upon the system that starts use case> | The user selects Search Meal | |
| **Preconditions**  <what we expect is already the state of the world> | 1. The mobile must have internet connection 2. The user must have selected to let the application use mobile data | |
| **Success End Condition**  <the state of the world upon successful completion> | The user, according to their preferences, views a list of meals, along with individual meals. | |
| **Failed End Condition**  <the state of the world if goal abandoned> | No meals are viewed by the user | |
| **Primary Actors;**  **Secondary Actors** | Application  User | |
| **Description / Main Success Scenario**  <the steps of the scenario from trigger to goal delivery and any clean up after. Indicate sub steps using numbering> | **Step** | **Action** |
| 1.a | The user selects Search Meal |
| 1.b | User then selects what they want to filter by |
| 1.c | The application sends a query to the web database to view all matches |
| 1.d | The SQL database then returns the query |
| 1.e | The query results are displayed by the application |
|  |  |
|  |  |
| **Alternative Flows**  <a: condition causing branching>  <a1: action or name of sub use case> | **Step** |  |
|  |  |

# System Design

Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:System Architecture.pdf

## System Architecture

Application – The mobile Application was developed using the model view controller (MVC) approach, which is a software architectural pattern for implementing user interfaces. Especially useful when using JavaScript as the added cascading style sheets (CSS) help to make the interface more attractive and user friendly.

Server Side Database– The server side database stores all information relevant to meal capture sharing.

Client Side Database – The client side database stores majority of the information that a user may need while using the application.

## Storage/Persistent Data Strategy

The users smart phone will be used as the primary storage device. With the server side database providing the ability to store data relevant to the sharing of meals between users.

Android smart phones provide SQLite as a means of storing data on the device itself. This means that data needed for basic functionality such as the ORAC rating calculation values and a list of predetermined meals can be saved locally. The user can also save their captured meals on their local device.

The server side database will only store information that a user allows to be stored, such as a captured meal and rating. This provides the infrastructure for users to search and share captured meals.

## Trade-offs and Choices

There were two options available to the development team. Store all data on a server side database, or store majority of data on a client side database and only use the server for information sharing capabilities.

Storing all data on a server side database was the least favorable option of the two analysed. This would mean that the user’s smart phone was constantly connecting to the server to retrieve information, resulting in slow application speed. This also posed potential risks, as the server would be storing possibly sensitive information about each user.

Storing majority of the data on a client side database such as the smartphone itself was the best option analysed. This would mean that the user had control over all of their information and could choose to upload their meal captures to the server so that other users are able to view them.

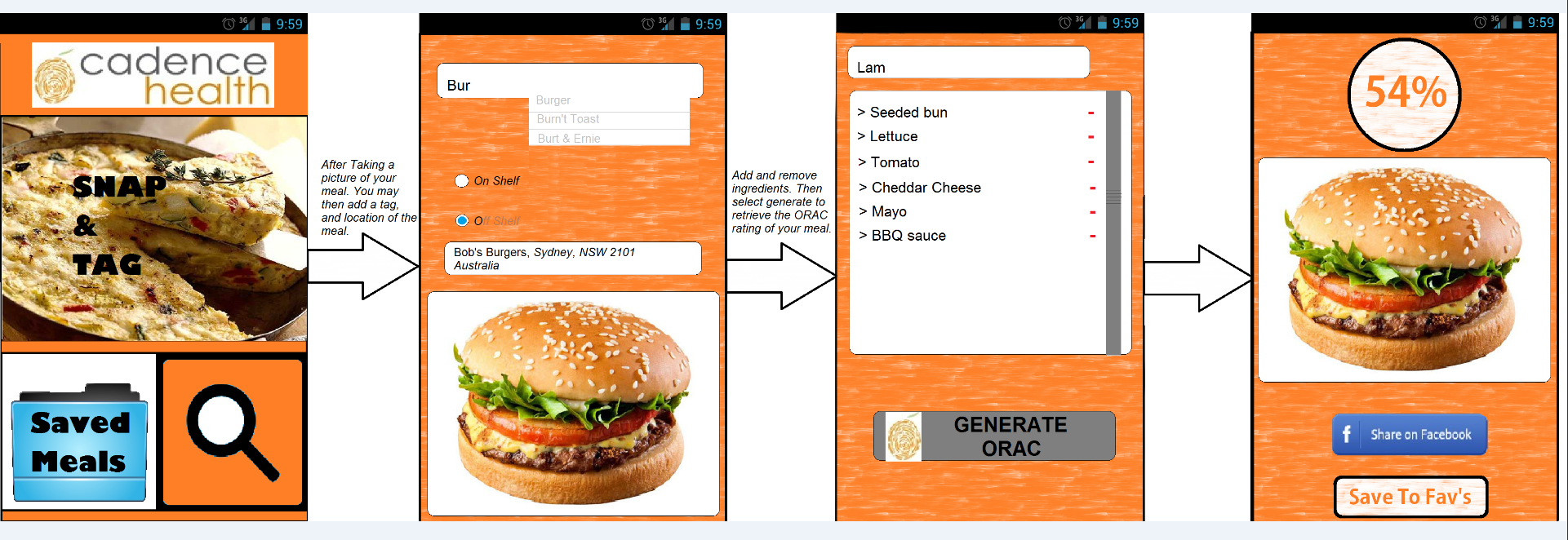
## Concurrent Processes

The mobile application does not contain any concurrent processes. This is due to the type of database management system used for storage. SQLite allows for multiple read operations at the same time, although the entire database is locked during writes. Meaning that all processes are sequential.

## Package Diagram

## Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:Package Model.pdf

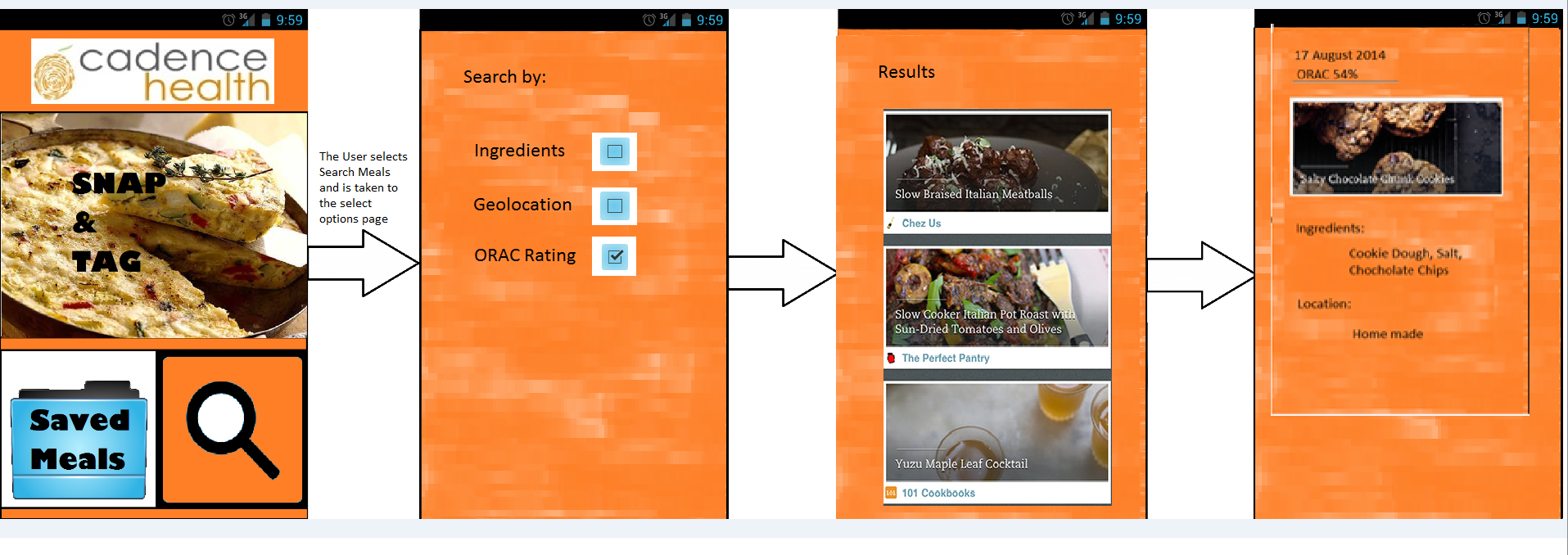
# User Interface Layouts

i. From Home Page – Tagging – Saved Meals

ii. From Home Page – Saved Meals – Individual Meal



iii. From Home Page – Search Meal



# Program Navigation Diagram

i. User going from Home Page through to a completed meal being saved

ii. User going from Home page through to a saved meal being viewed

iii. User going from Home page through to a saved meal on webserver

# Data Definitions

All data stored locally on the user’s smartphone and on the server side database will be stored using SQLite. Which is a relational database management system contained in a C programming library. SQLite differs from regular database management systems in that it’s server-less.

# Analysis and Design Class Diagram

## Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:Analysis Class Diagram.pdf

## Sequence Diagram

i. Create Meal

Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:Sequence - Create Meal.pdf

ii. Search Meals

Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:Sequence - SearchMeals.pdf

iii. View Meals

Macintosh HD:Users:danielrichter:Documents:ISYS355 Project:Project_Documents:Sequence - View Meals.pdf

## State Diagram

Currently there are no objects that perform state transitions within the current design of the application. Because of this, at this point in time, no state diagrams have been included.

# Requirements Traceability Matrix

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Requirement ID | From Requirement ID | To Requirement ID | Type | Use Case/s | Build No | Remarks |
| R1.1 | R1.1 | R1.13 | Essential | Create Meal | V1.0 |  |
| R1.1.1 | R1.1 | R1.1.1 | Essential | Create Meal |  |  |
| R1.2 | R1.1 | R1.4 | Essential | Create Meal | V1.0 |  |
| R1.3 | R1.3 | R1.4 | Essential | Create Meal | V1.0 |  |
| R1.4 | R1.4 | R1.3, R1.2 | Essential | Create Meal | V1.0 |  |
| R1.5 | R1.2 |  | Essential | Create Meal | V1.0 |  |
| R1.6 | R1.2 | R1.5 | Essential | Create Meal | V1.0 |  |
| R1.7 | R1.7 |  | Essential | View Meal | V1.0 |  |
| R1.8 | R1.3 | R1.4 | Essential | View Meal | V1.0 |  |
| R1.9 | R1.5 | R1.6 | Essential | View Meal | V1.0 |  |
| R1.10 | R1.4 | R1.3 | Essential | View Meal | V1.0 |  |
| R1.11 | R1.2 |  | Essential | View Meal | V1.0 |  |
| R1.12 | R1.2 |  | Essential | Search Meal | V1.0 |  |
| R1.13 | R1.2 |  | Extensional | Search Meal | V1.0 |  |
| R1.14 |  |  | Essential | Search Meal | V1.0 |  |
| R1.15 |  |  | Essential | Search Meal | V1.0 |  |
| R1.16 |  |  | Essential |  | V1.0 |  |
| R1.17 |  |  | Essential |  | V1.0 |  |
| R1.18 |  |  | Essential |  | V1.0 |  |
| R1.19 |  |  | Extensional |  | V2.0 |  |
| R1.20 |  |  | Extensional |  | V2.0 |  |
| R1.21 |  |  | Extensional |  | V1.0 |  |
| R1.22 |  |  | Extensional |  | V2.0 |  |
| R1.23 |  |  | Extensional |  | V2.0 |  |
| R1.24 |  |  | Extensional |  | V2.0 |  |
| R1.25 |  |  | Essential |  | V2.0 |  |
| R1.26 |  |  | Essential |  | V2.0 |  |
| R1.27 |  |  | Extensional |  | V2.0 |  |
| R1.28 |  |  | Essential |  | V2.0 |  |
| R1.29 |  |  | Extensional |  | V2.0 |  |
| R1.30 |  |  | Essential |  | V2.0 |  |
| R1.31 |  |  | Essential |  | V1.0 |  |
| R1.32 |  | R1.46, R1.47, R1.48 | Essential |  | V1.0 |  |
| R1.33 |  |  | Essential |  | V1.0 |  |
| R1.34 |  |  | Essential |  | V1.0 |  |
| R1.35 |  |  | Essential |  | V1.0 |  |
| R1.36 |  |  | Essential |  | V1.0 |  |
| R1.37 |  |  | Extensional |  | V1.0 |  |
| R1.38 |  |  | Essential |  | V1.0 |  |
| R1.39 |  |  | Extensional |  | V1.0 |  |
| R1.40 |  | R1.6 | Essential | Create Meal | V1.0 |  |
| R1.41 | R1.40 |  | Essential | Create Meal | V1.0 |  |
| R1.42 |  |  | Essential |  | V1.0 |  |
| R1.43 |  |  | Essential |  | V1.0 |  |
| R1.44 |  |  | Essential |  | V1.0 |  |
| R1.45 |  |  | Extensional |  | V1.0 |  |
| R1.46 |  |  | Essential |  | V1.0 |  |
| R1.47 |  |  | Essential |  | V1.0 |  |
| R1.48 |  |  | Essential |  | V1.0 |  |

# Design Assumptions

The following assumptions are related to the development of the mobile Application:

* The current server provided by the client Cadence Health will be able to use SQLite
* All user smartphones containing Android will be using 4.1 and up