Software Requirements Specification

For

My Diet

Version 1.0 approved

Prepared by :

Ana Hernandez
Alaba Magallanes
Diego Torres
Julio Roa
Victor Rodriguez

27/09/13

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the My Diet v 1.0? application. It would explain what the application would do, the purpose and features of the application, interfaces and the constraints under which it must operate. This document describes the entire application.

1.2 Document Conventions

Same as SRS template

1.3 Intended Audience and Reading Suggestions

This document is intended for the client, developers, testers, project manager and documentation writers. In this SRS we will find all the specifications for the requirements and agreements between the client and the development team. The second chapter of this document is an overall description about the project followed by the project specific features and interfaces. The last part of this document explains the requirements, project managers, developers and the client should read the entire document.

1.4 Project Scope

There is a vision and scope document available.

1.5 References

- Vision and scope document

2. Overall Description

2.1 Product Perspective

This is the first version of My Diet App . The main parts are described in Figure 1 :

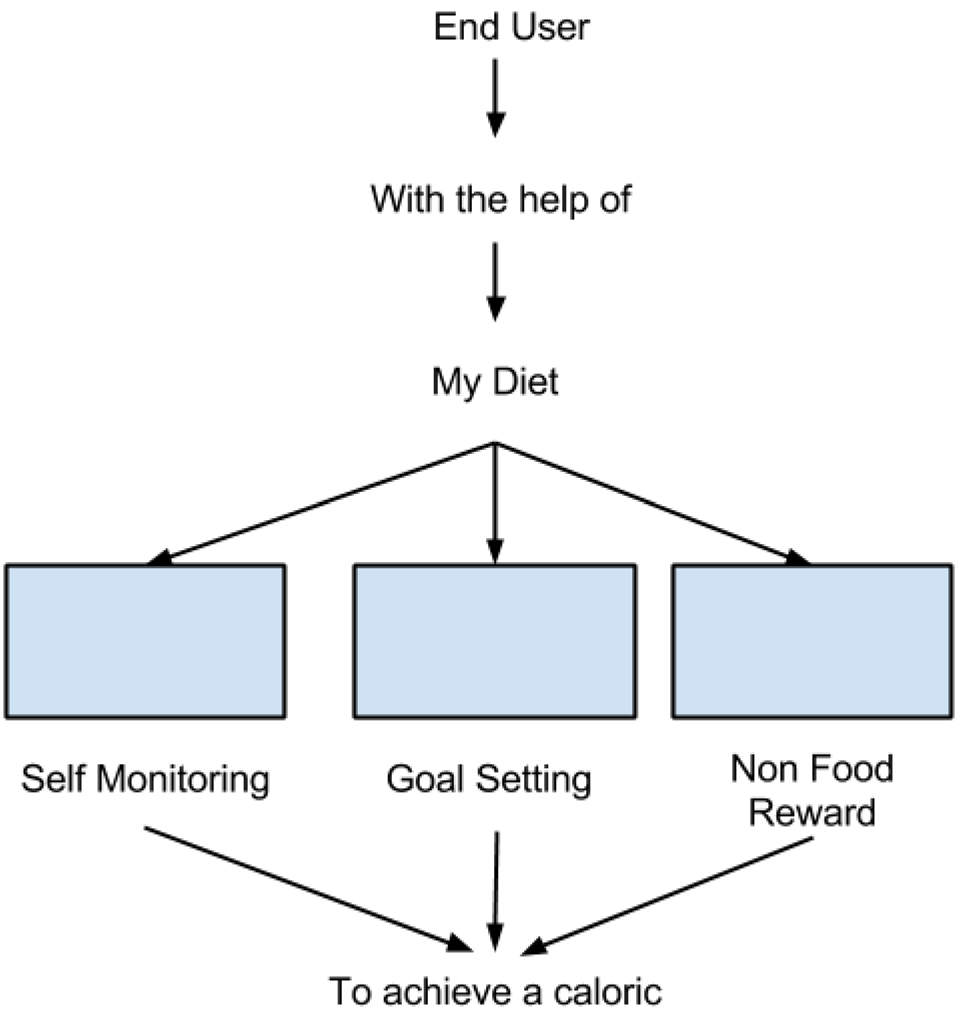

Figure 1 My Diet main parts

2.2 Product Features

Feature 1: Self-monitoring of caloric intake

- Track calories
- Add new meals

Feature 2: Goal setting

- Determine ideal caloric intake

Feature 3: Non food rewards

- Report balance of ingested meals
- Point-based system

2.3 User Classes and Characteristics

- Patients diagnosed with obesity or overweight: most of these users will have a nutritionist caloric intake recommendation. They will use the app on a daily basis and must enter all their meals. The app will be used as a tool for them to lose weight by tracking their caloric intake in each meal. This class will be the main market of the app.
- Patients with no weight problems who want to monitor their intake: the second class of users doesn't have weight problems, and use the app only to stay in shape and monitor their consumption. They could use the app on a daily basis or just as informative tool to know if they're eating correctly.

2.4 Operating Environment

- The main development of the application exist on the Application level.(Figure 2)
- It use a few parts of the existing Applications Framework (Content Provider -> My SQL database)
- It does not require the development of libraries or drivers

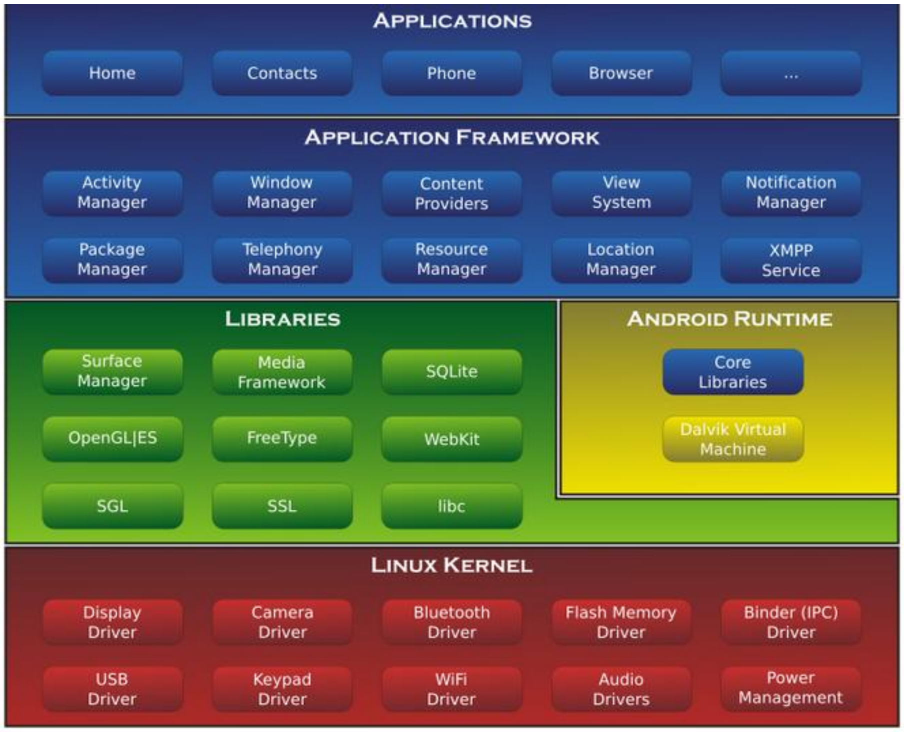

Figure 2 Android OS layaout

This application is supported from Android version 2.3 and up. Table 1 show the multiple versions

Version	*Code name*	*Release date*	*API level*	*Distribution*
4.4	*KitKat*	*TBA*	*TBA*	*0%*
4.3	*Jelly Bean*	*July 24, 2013*	*18*	*0%*
4.2.x	*Jelly Bean*	*November 13, 2012*	*17*	*8.5%*
4.1.x	*Jelly Bean*	*July 9, 2012*	*16*	*36.6%*
4.0.3–4.0.4	*Ice Cream Sandwich*	*December 16, 2011*	*15*	*21.7%*
3.2	*Honeycomb*	*July 15, 2011*	*13*	*0.1%*
3.1	*Honeycomb*	*May 10, 2011*	*12*	*0%*
2.3.3–2.3.7	*Gingerbread*	*February 9, 2011*	*10*	*30.7%*
2.3–2.3.2	*Gingerbread*	*December 6, 2010*	*9*	*0%*
2.2	*Froyo*	*May 20, 2010*	*8*	*2.4%*
2.0–2.1	*Eclair*	*October 26, 2009*	*7*	*0%*
1.6	*Donut*	*September 15, 2009*	*4*	*0%*
1.5	*Cupcake*	*April 30, 2009*	*3*	*0%*

Table 1 Android OS versions

2.5 Design and Implementation Constraints

Just Android phones from the OS version previously specified (2.3 and up)

2.6 User Documentation

- User manual
- Examples

2.7 Assumptions and Dependencies

- GPL 2 License
- All android apps are portable to other new Android OS versions

3. System Features

3.1 Self-monitoring caloric intake

3.1.1 Description and Priority

The user is able to track calories of the meals that has ingested. In order to do this, the app must have a local database (based on light my-sql) indicating caloric content on common foods. This feature includes two sub features:

- Track calories: the user is able to add a list of all the meals ingested during the day and the app calculates the number of calories ingested.

- Add new meals:

    - If an ingredient is not present in the default database, the user can add it by indicating the number of calories present in it.

    - If a meal is not present in the database, the user can add it by either indicating the total calories in it; or indicating the meal ingredients from the actual database.

    - End users can not share new meals or ingredients among each other

Priority: High

3.1.2 Stimulus/Response Sequences

- Add caloric intake

    - From existing

        - Ingredient:

      - Open a list view of existing ingredients
      - User must return to previous menu with back button
    - Meal
      - Open a list view of existing meals
      - User must return to previous menu with back button
  - New
    - Ingredient
      - User has the options:
        - Name : If repeated , send an error toast message and don’t let the user add the value
        - Caloric content
        - User must return to previous menu with back button if the ingredient was created successfully
        - If user returns without saving the value the ingredient is not added to the database
    - Meal
      - Indicate caloric content
        - User has the options:
          - Name : If repeated , send an error toast message and don’t let the user add the value
          - Caloric content
          - User must return to previous menu with back button if the ingredient was created successfully
          - If user returns without saving the value the meal is not added to the database
      - Indicate ingredients
        - User has the options:
          - Name : If repeated , send an error toast message and don’t let the user add the value
          - Add ingredient field (default is two ingredient fields)
            - Open a list view of existing ingredients

- User must go return to previous menu with back button
- User must return to previous menu with back button if the ingredient was created successfully
- If user returns without saving the value the ingredient is not added to the database

- Check records
  - Daily
    - Display last day entries
    - User must go return to previous screen using back button
  - Monthly
    - Display last month caloric intake
    - User must go return to previous screen using back button
- Modify database
  - Modify ingredient
    - Modify ingredient info
      - Modify name: If repeated , send an error toast message and don’t let the user modify the value
      - Modify caloric content
      - User must return to previous screen using back button
    - Delete ingredient
  - Modify meal
    - Modify meal info
      - Modify name
      - Modify ingredients
      - Modify caloric content
      - User must go return to previous screen using back button
    - Delete meal

3.1.3 Functional Requirements

REQ-1: There must be a Light my-SQL database according to the feature description with a well designed (portable) database interface to:

- Add element
- Modify element
- Delete element

REQ-2: If the end user commit a cast mistake, system should not crash and show a warning toast message regarding the error.

3.2 Goal setting

3.2.1 Description and Priority

The user is able to set a daily caloric intake range according to the instructions given to them by their nutritionist or calculated by the app. If the user does not set a goal, a 1800-2600 calorie-intake goal is set by default. This feature includes a sub-feature:

- Determine ideal caloric intake: if the user doesn’t have a nutritionist recommendation for caloric intake, the app will select an ideal based on gender and activity given the following values:
    - Men
        - Sedentary: 2000-2600
        - Moderately active: 2200-2800
        - Active: 2400-3000
    - Women
        - Sedentary: 1600-2000
        - Moderately active: 1800-2200
        - Active: 2000-2400

Priority: Medium

3.2.2 Stimulus/Response Sequences

- Set new goal
    - Nutritionist recommendation
        - User selects caloric inferior limit, given by the personal nutritionist
        - User selects caloric superior limit, given by the personal nutritionist
    - Calculate ideal goal

  - User selects gender (male/female)
  - User selects activity (sedentary/Moderate/Active)
    - App calculates ideal goal based on the previous information and default ranges. A message is displayed with the selected range
  - User must return to previous screen by using the back button

3.2.3 Functional requirements

REQ-1: There must be two fields on the Application database (based on light My SQL) .

- Caloric inferior limit
- Caloric superior limit

These values must be stored in a none volatile memory.

There database should have an interface to easily

- Read element
- Modify element

3.3 Non food rewards

3.3.1 Description and priority

The app provides a feedback system to the user promoting positive reinforcement by a point system in which the app gives points to the user everyday that the goal is met; on the other hand it removes points when the opposite happens. If the caloric intake exceeds the goal, more points are removed from the user; if the caloric intake doesn't reach the inferior limit, lesser points are removed.

This features includes two sub-features:

- Report balance of daily intake: the app classifies a meal entry as balanced if its caloric content is within the result of the following operation caloricGoal/NumberOfMeals; it classifies the meal as unbalanced otherwise.
- Point-based system: the user gets points or looses points based on a goal previously set on a weekly basis. Points reset every week, and recorded in a database so that the user can compete against himself each week.

Example 1)

“I´m a sedentary woman, today I woke up and had a banana and yogurt for breakfast(300kcal), later i ate a doughnut(500kcal). For lunch I had a big mac(1500kcal) and i skipped dinner.(0kcal)”

- Total caloric intake of day: 2300
- Caloric Goal: 1600-2000
- Balance inferior = 1600/3 = 533kcal
- Balance superior = 2000/3 = 666kcal
- Ideal meal: 533-666kcal

Daily intake was not balanced because:

- Banana and yogurt breakfast had less than 533 kcal
- Doughnut snack had less than 533kcal
- Big Mac lunch had more that 666kcal
- Dinner was skipped

Priority: Low

3.3.2 Stimulus/Response Sequences

Balance meals

- Display if day was balanced
- User must return to previous screen by using the back button

Points

- Displays actual points up until last completed day
- Displays points summary on the last completed day
- User must return to previous screen by using the back button

3.3.3 Functional Requirements

REQ-1 There must be a table(or tables) in the App Database (Based on light My SQL) for storage all the points of the user within the time (weeks) . This table (or tables) should have views of the most commons queries required for this feature.

4. External Interface Requirements

In the Table 2 we show the main user interfaces for My Diet App

ID Item	Item Description
UI-1	My diet app screen displays shall conform to the *Android Design Patterns* (http://developer.android.com/design/patterns/index.html)
UI-2	Profile view: The user can configure his profile.

UI-3	Food/Ingredients view: The user can add, modify and delete food/ingredients.
UI-4	Custom meals view: The user can add, modify and delete the custom meals.
UI-5	Records view: The user can add, modify and delte the records of the calories or melas consumed.
UI-6	Meals view: The user can viewa several meals with the calorie information.
UI-7	Calorie indicator view: The user can view the quantity of calories consumed in the current day.
UI-8	Reports view: The user can show the daily and weely report for the consumed calories.
UI-9	Help view: The user can find the user guide in order to involve with the application in a right way.
UI-10	About view: The user can review the version information from the application, the authors, license and name company.

Table 2 Main user interfaces

GUI standards

UI Component	Description	Value
Themes	To promote greater cohesion between all apps on the platform, Android provides three system themes that you can choose from when building apps	Holo Light Holo Dark Holo Light with dark action bars
Touch Feedback	Use color and illumination to respond to touches, reinforce the resulting behaviors of gestures, and indicate what actions are enabled and disabled.	48dp Rhythm
Mind the gaps	Spacing between each UI element	8dp
Typographic Scale	Size of the typographic	micro 12p small 14p medium 18p large 22p
Iconography	to create an icon for different densities, you should follow the 2:3:4:6 scaling ratio between the four primary densities (medium, high, x-high, and xx-high, respectively	For example, consider that the size for a launcher icon is specified to be 48x48 dp. This means the baseline (MDPI) asset is 48x48 px, and the high density (HDPI) asset should be 1.5x the baseline at 72x72 px, and the x-high density (XHDPI) asset should be 2x the baseline at 96x96 px, and so on.
Launcher	The launcher icon is the visual representation of your app on the Home or All Apps screen.	48x48 dp
Action Bar	Action bar icons are graphic buttons that represent the most important actions people can take within your app.	Action bar icons for phones should be 32x32 dp. Full asset, 32x32 dp Optical square, 24x24 dp

Table 3 Android patterns design

Naming and structure files convection

Icons

Asset Type	Prefix	Example
Icons	ic_	ic_star.png
Launcher icons	ic_launcher	ic_launcher_calendar.png
Menu icons and Action Bar icons	ic_menu	ic_menu_archive.png
Status bar icons	ic_stat_notify	ic_stat_notify_msg.png
Tab icons	ic_tab	ic_tab_recent.png
Dialog icons	ic_dialog	ic_dialog_info.png

Table 4 Naming and structure convention

Working Space

It organizes files by density in order to supporting multiple screen densities.

For example:

```
art/...
  mdpi/...
    _pre_production/...
      working_file.psd
    finished_asset.png
  hdpi/...
    _pre_production/...
      working_file.psd
    finished_asset.png
  xhdpi/...
    _pre_production/...
      working_file.psd
    finished_asset.png
```

The resources directory structure.

```
res/...
  drawable-ldpi/...
     finished_asset.png
  drawable-mdpi/...
     finished_asset.png
  drawable-hdpi/...
     finished_asset.png
  drawable-xhdpi/...
     finished_asset.png
```

General App UI Structure

A typical Android app consists of top level and detail/edit views. If the navigation hierarchy is deep and complex, category views connect top level and detail views.

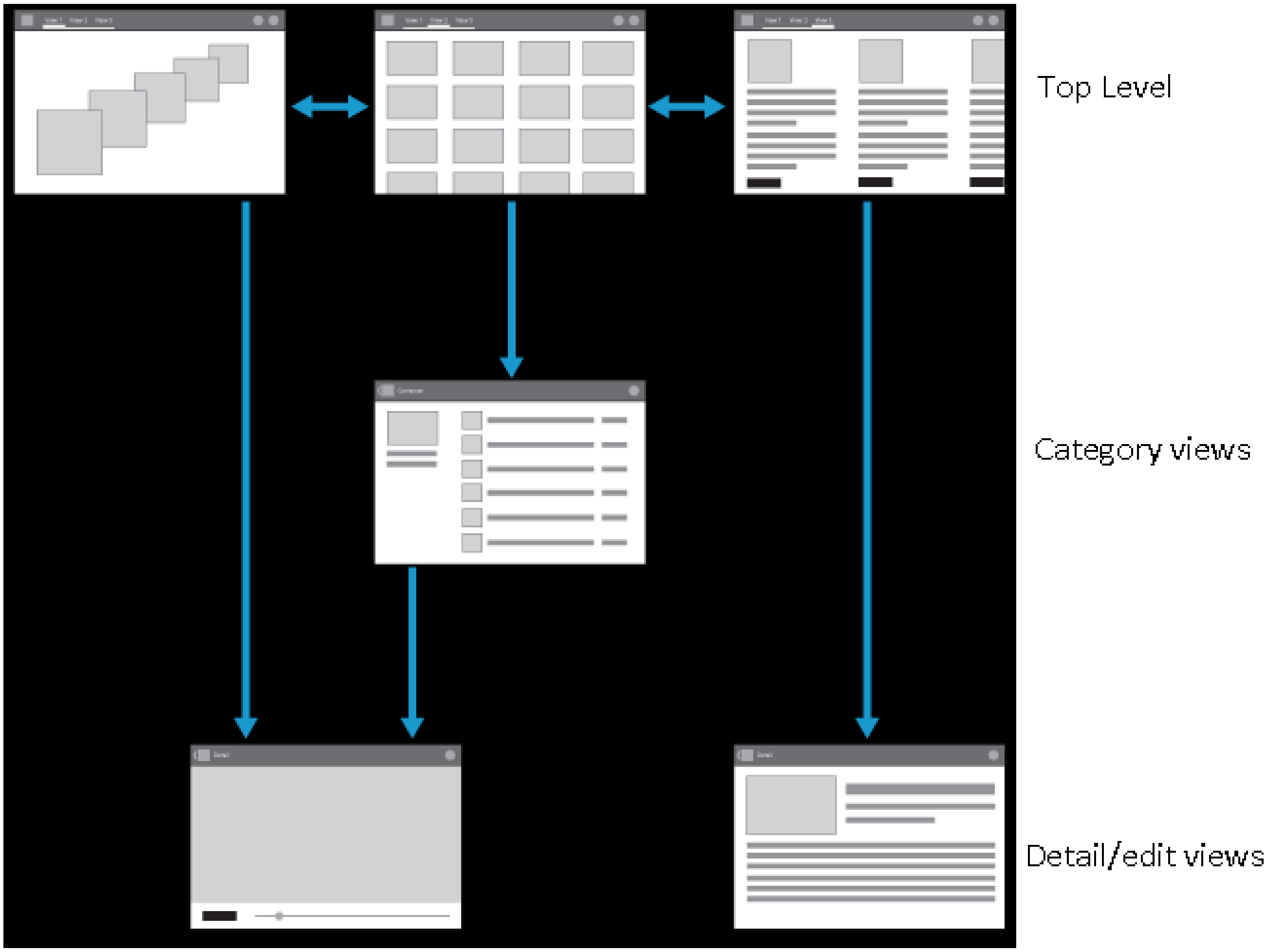

Figure 3 Top Level and detail view

In general way My Diet App is going to follow the [Android Design Patterns] in order to develop a quality user interfaces.

Prototypes

1- Start template: This is the template when My Diet App is loading.

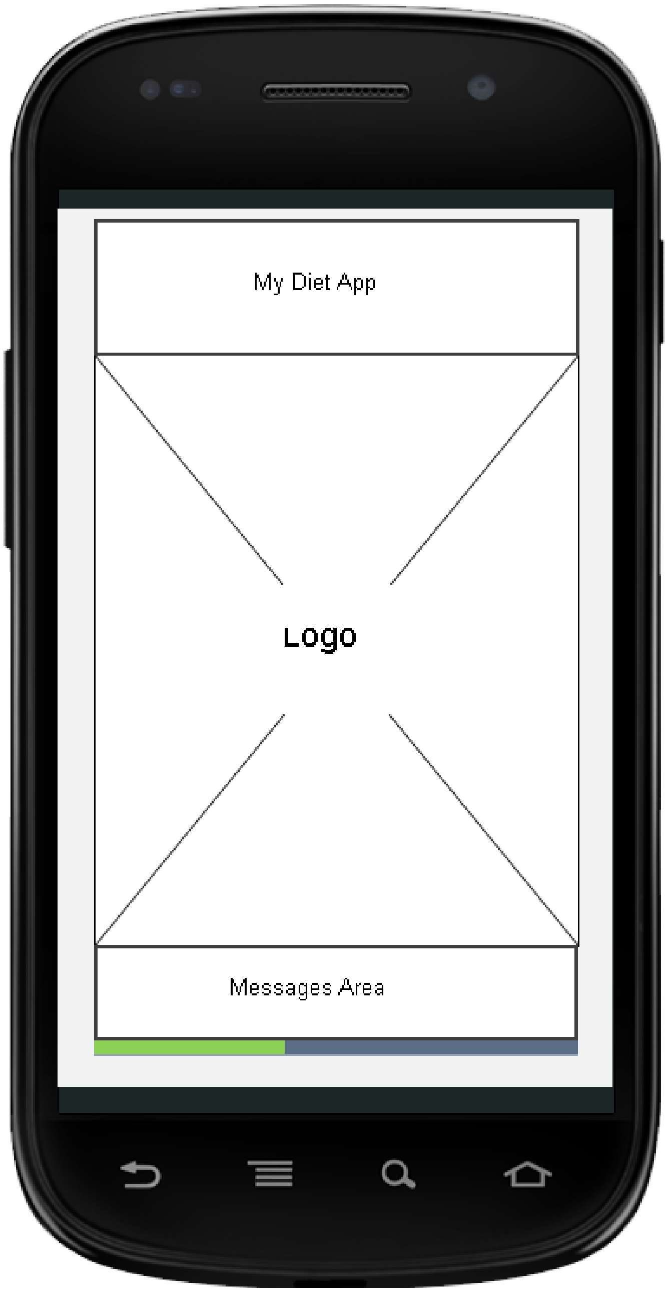

Figure 4 Start template

2- Main Menu Template: This is the template when My Diet App is loaded, this the top level UI. We can view the main features and the calorie indicator for the app.

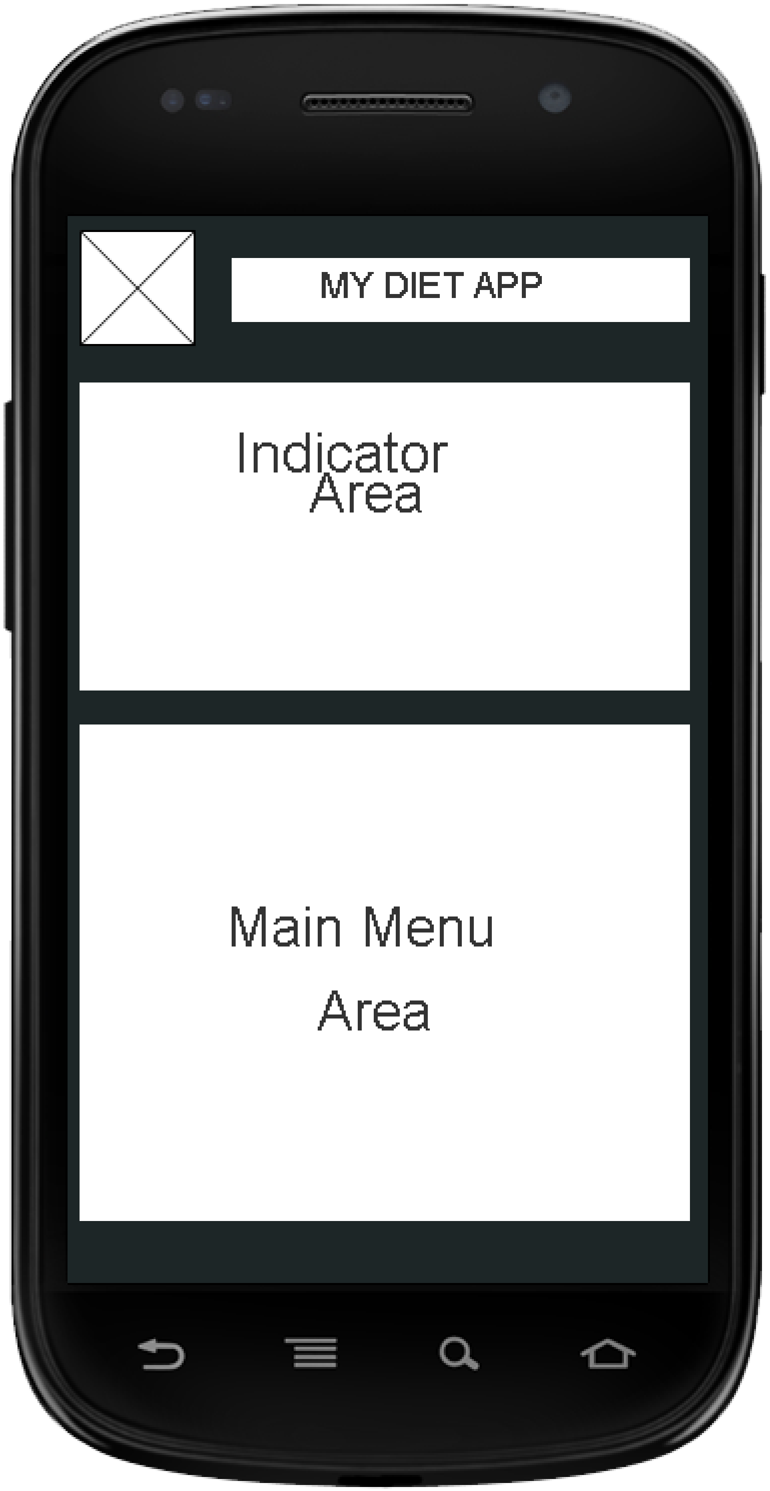

Figure 5 Main Menu Template

3- Submains template: This is the template when My Diet App is loaded and one option in the main menu was selected, this the category level UI.

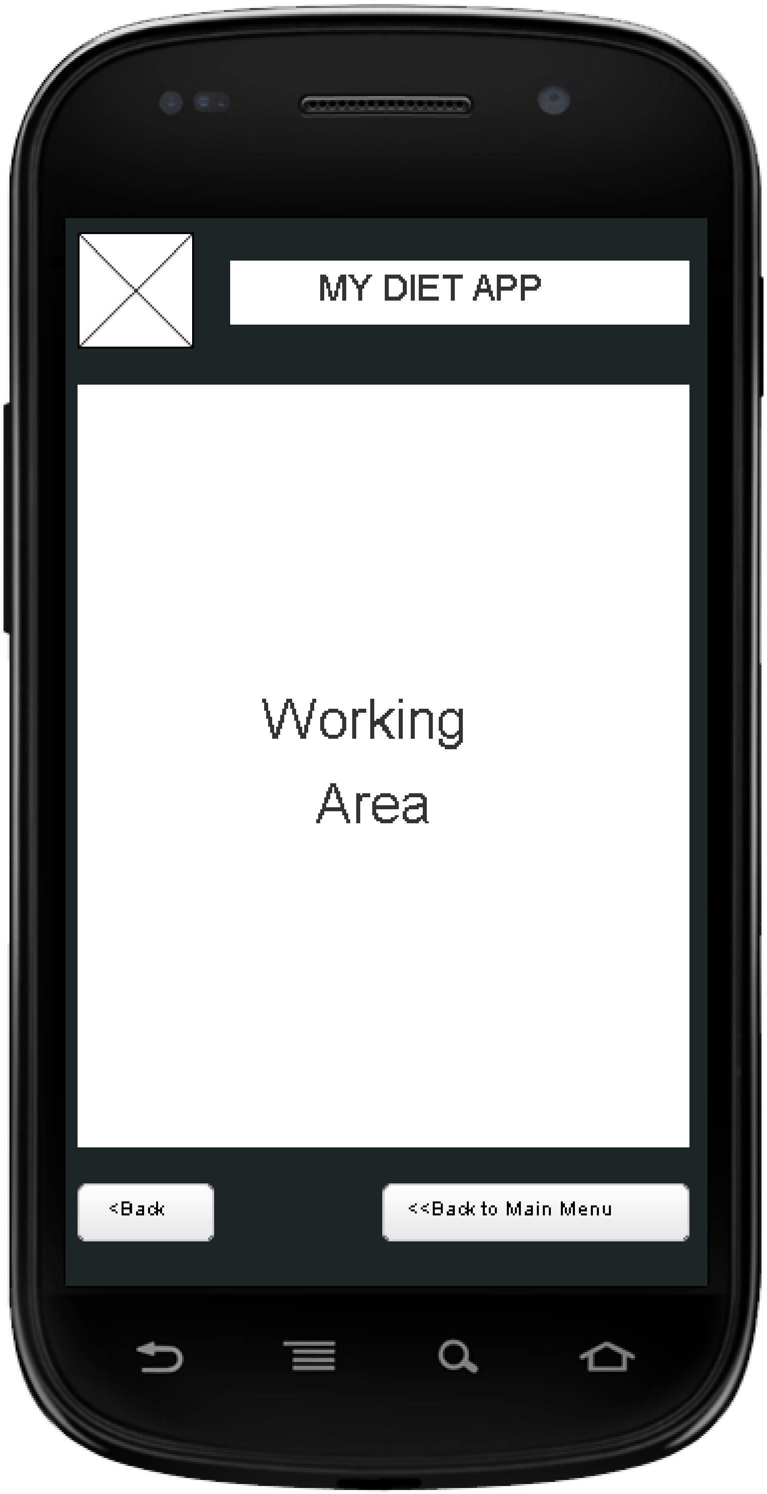

Figure 6 Sub mains template

4- Main menu (category view)

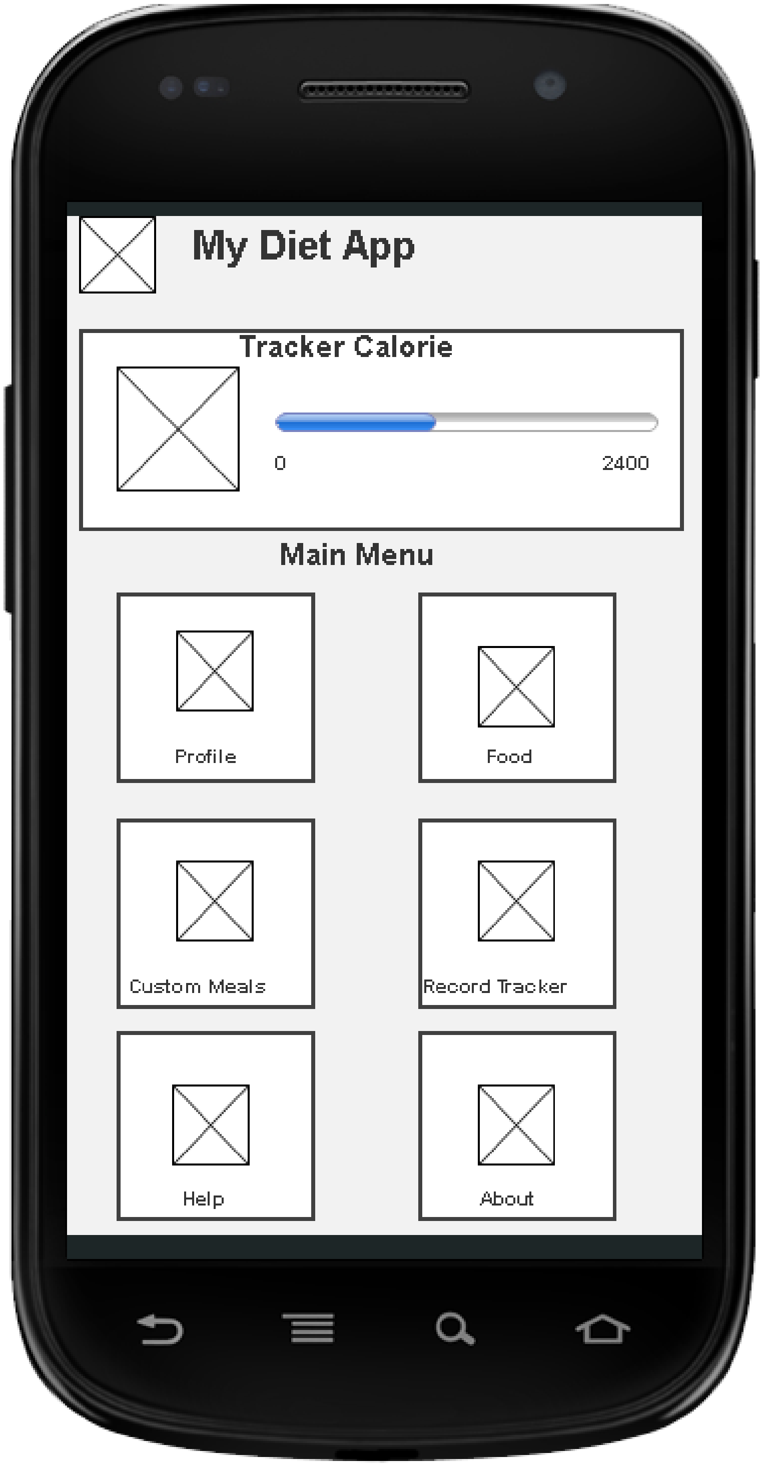

Figure 7 Category view

5- Profile

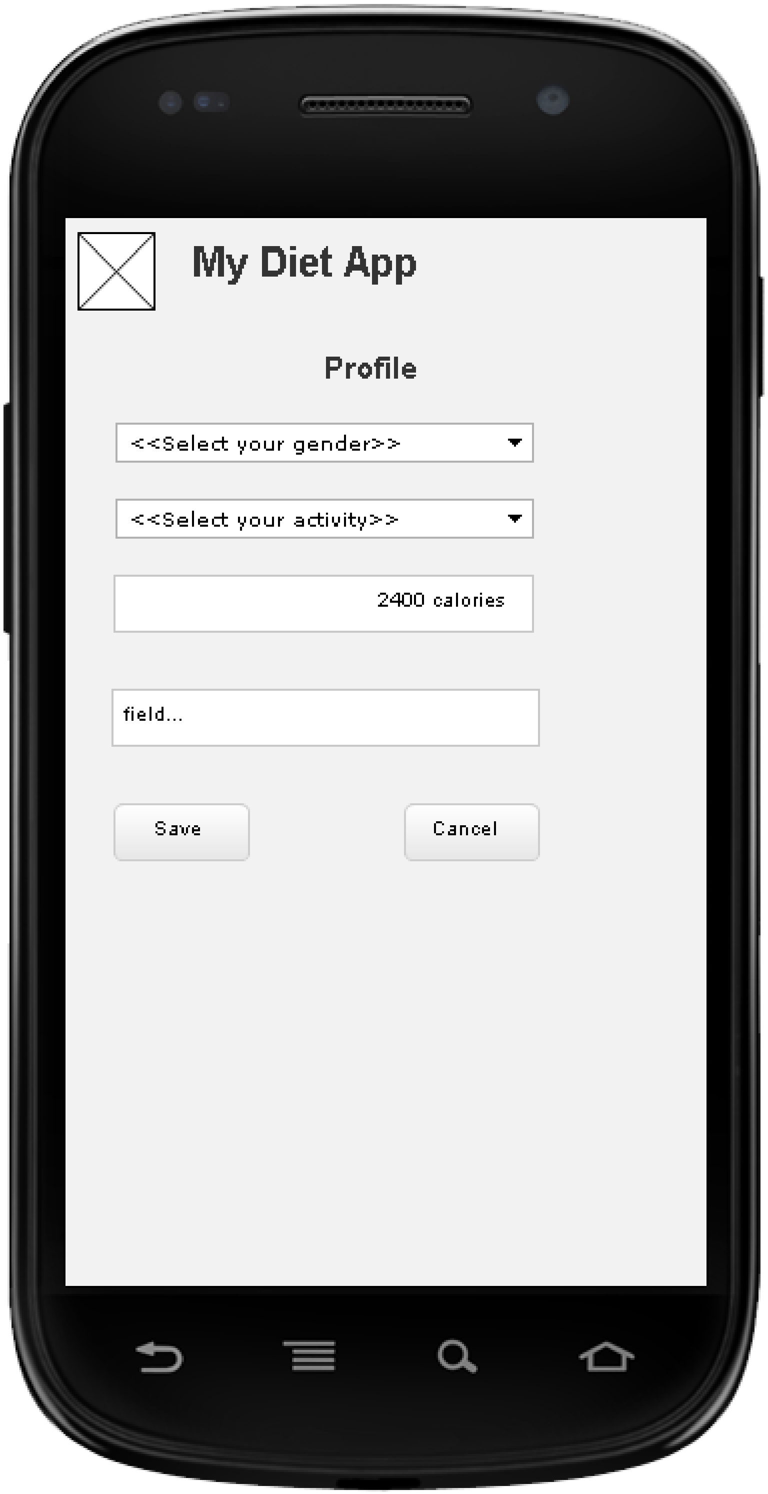

Figure 8 Profile

6- Add record. Detail View

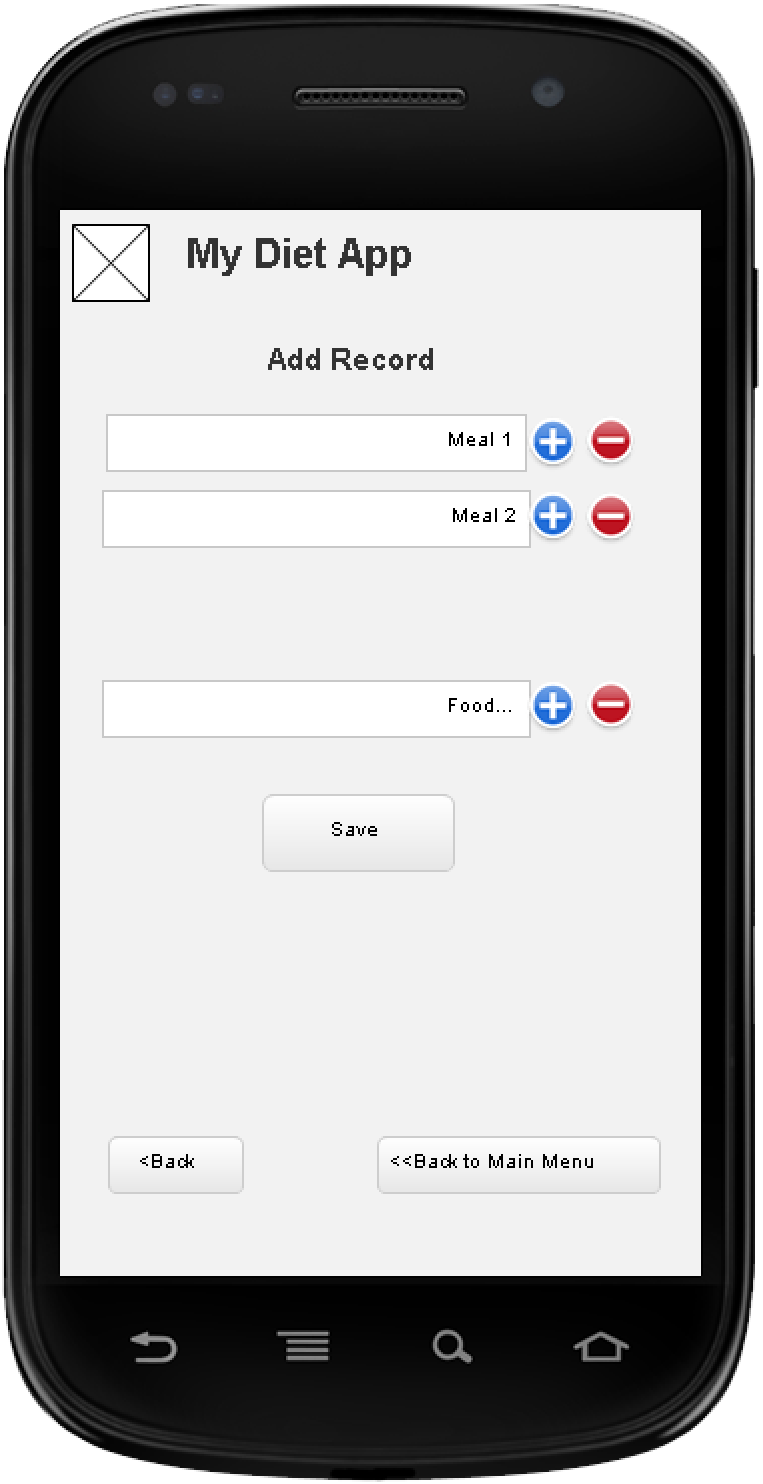

Figure 9 Add record. Detail View

4.1 Hardware Interfaces

Not apply to My Diet Project.

Maker	**Name**	**Android version**
Acer	Liquid Glow	4.0
Asus	Padfone	4.0
Asus	Padfone 2	4/01/2013
HTC	Desire C	4.0 with HTC Sense 4.0
HTC	Desire V	4.0 with HTC Sense 4.0
HTC	One V	4.0 with HTC Sense 4.0
HTC	One S	4.0 with HTC Sense 4.0, planned upgrade to 4.1
HTC	One X	4.0 with HTC Sense 4.0, upgradable to 4.1 (HTC Sense 4+)
HTC	One XL	4.0 with HTC Sense 4.0, planned upgrade to 4.1
HTC	Evo 4G LTE	4.0 with HTC Sense 4.0
HTC	Desire X	4.0 with HTC Sense 4.0
HTC	One X+	4.1 with HTC Sense 4+
HTC	HTC One	4.1.2 Jelly Bean upgradeable to 4.2.2 (Google Edition 4.2.2)
HTC	HTC One Mini	4.2.2 Jelly Bean
HTC	Butterfly	4.1.2 with HTC Sense 4+, upgradeable to 4.2.2
Karbonn	A15	4.0
LG	Optimus L5	4.0 with Optimus UI
LG	Optimus L7	4.0 with Optimus UI
LG	Optimus L9	4.0 with Optimus UI; upgradable to 4.1.2
LG	Optimus Vu	4.0 with Optimus UI
LG	Optimus 4X HD	4.0 with Optimus UI
LG	Optimus G	4.0 with Optimus UI 3.0, upgradable to 4.1
LG	G2	4.2.2 with Optimus UI 3.0
LG	Google Nexus 4	4/02/2013
Motorola	Atrix HD, MB886	4.0
Motorola	Droid RAZR I	4.0 with MOTOBLUR UI
Motorola	Droid RAZR HD	4.1 with MOTOBLUR UI
Motorola	Moto X	4.2.2 Jelly Bean
Panasonic	Eluga Power	4.0
Samsung	GT-B5330, Galaxy Chat	4.0 with TouchWiz Nature UX UI, upgradable to 4.1.2
Samsung	Galaxy Nexus	4.0, upgradable to 4.3
Samsung	S7562 Galaxy S Duos	4.0 with TouchWiz Nature UX UI
Samsung	i9300 Galaxy S III	4.0 with TouchWiz Nature UX UI, upgradable to 4.1.2 (February 2013) and 4.3 (October 2013)
Samsung	i9305 Galaxy S III	4.1.1/4.1.2 with TouchWiz Nature UX UI, upgradeable to 4.3 (October 2013)
Samsung	i9505 Galaxy S4(with LTE)	4.2.2 with TouchWiz Nature UX 2.0 UI, upgradeable to Android 4.3 (October 2013)
Samsung	N7100 Galaxy Note II	4.1.2 with TouchWiz Nature UX UI
Samsung	Galaxy S III mini	4.1.2 with TouchWiz Nature UX UI
Samsung	Galaxy S4 mini	4.2.2 with TouchWiz Nature UX 2.0 UI
Samsung	Galaxy Express	4.1.2 with TouchWiz Nature UX UI
Samsung	Galaxy Premier	4.1.2 with TouchWiz Nature UX UI
Sony	Xperia tipo	4.0
Sony	Xperia Miro	4.0
Sony	Xperia E	4/01/2013

Sony	Xperia Neo L	4.0
Sony	Xperia acro S	4.0
Sony	Xperia SL	4.0, upgradeable to 4.1.2
Sony	Xperia J	4.0, upgradable to 4.1.2
Sony	Xperia V	4.0, upgradeable to 4.1.2
Sony	Xperia T	4.0, upgradeable to 4.1.2
Sony	Xperia Z	4.1, upgradable to 4.2.2
Sony	Xperia ZL	4.1, upgradable to 4.2.2
Sony	Xperia ZR	4.1, upgradable to 4.2.2
Sony	Xperia Z Ultra	4.1, upgradable to 4.2.2
Sony	Xperia Z1	4/02/2002
Sony	Xperia SP	4.1, planned upgrade to 4.3
Sony	Xperia L	4/01/2002
Sony	Xperia M	4/01/2002
Spice	Spice stellar nhance mi-435	4.0
Cherry Mobile	Flare	4.0

Table 5 Support Devices

4.2 Software Interfaces

Database schema: The database will exist to store food, meals, recipes and daily records.
Continuous Integration: Jenkins server to make the builds for the application during development stage and GUI distributed integrating tool to keep the history for the source code.

4.3 Communications Interfaces

Connection to SQLite Database for configuration.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

As on every mobile application, having a good performance is one of the most important things to give a good user experience.

- Every app operation (database query, data storing, any calculation) should take less than 1 second to process.

5.2 Safety Requirements

- Meet every policy or regulation that an standard Android application requires.

- Since this application will let the user to establish goals for calories consumption, a warning message should appear if the goal that will be set could attempt to the users' health depending on their age, gender and weight.

5.3 Security Requirements

Because sensitive information such as user data (name, email account, etc) our application must eliminate any security hole. User data needs to be encrypted and accessible over the Internet only after authentication.

5.4 Software Quality Attributes

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: Issues List