Software Requirements Specification

for

My Diet

Version 1.0 approved

Prepared by :

Ana Hernandez
Alba Magallanes
Diego Torres
Julio Roa
Victor Rodriguez

08/11/13

Table of Contents

Revision History

1. Introduction
- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Project Scope
- 1.5 References

2. Overall Description
- 2.1 Product Perspective
- 2.2 Product Features
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

3. System Features
- 3.1 System Feature 1
- 3.2 System Feature 2 (and so on)

4. External Interface Requirements
- 4.1 User Interfaces
- 4.2 Hardware Interfaces
- 4.3 Software Interfaces
- 4.4 Communications Interfaces

5. Other Nonfunctional Requirements
- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: Issues List

Revision History

Name	Date	Reason For Changes	Version
Team	08/11/13	Feedback sent by the client.	2.0

1. Introduction

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.

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.

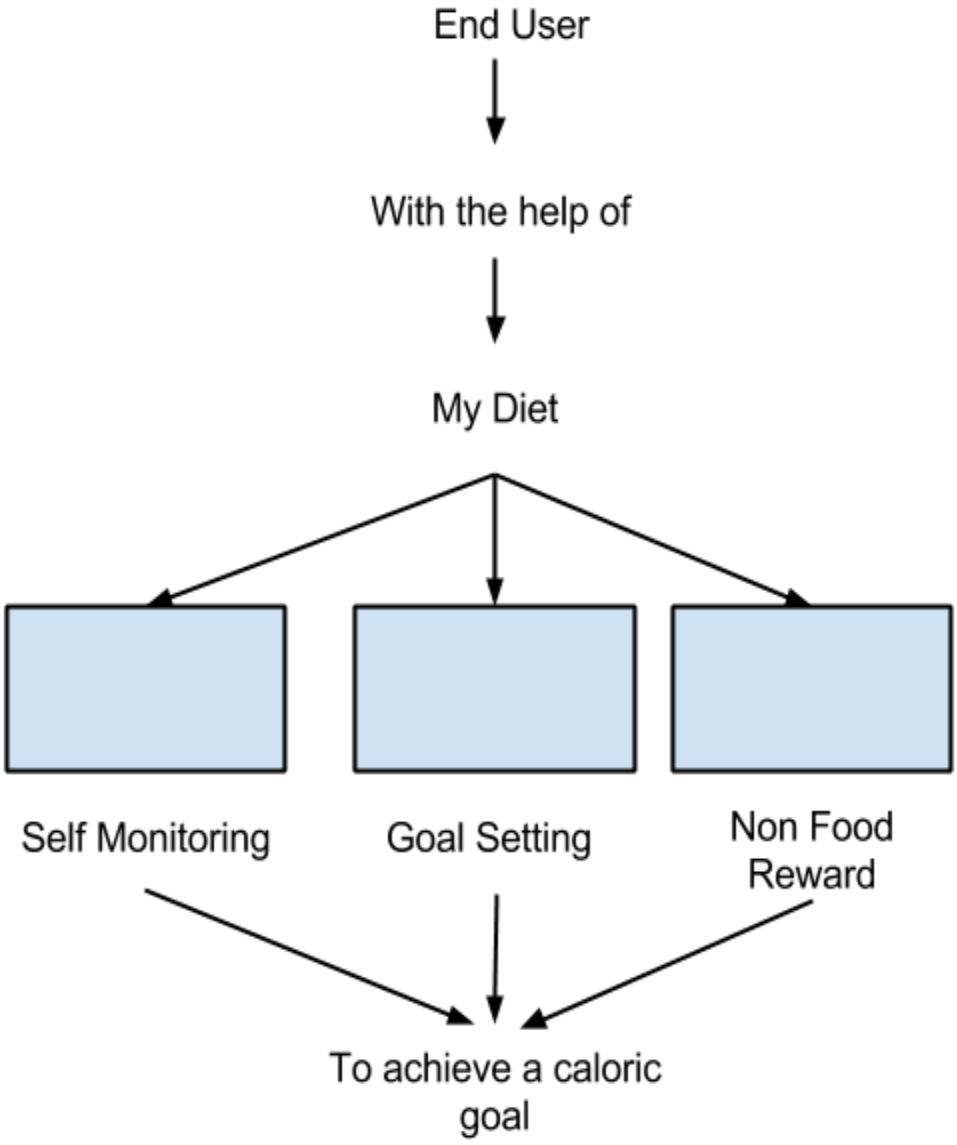
Project Scope

There is a vision and scope document available.

2. Overall Description

Product Perspective

This is the first version of My Diet App . The main parts are :



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.

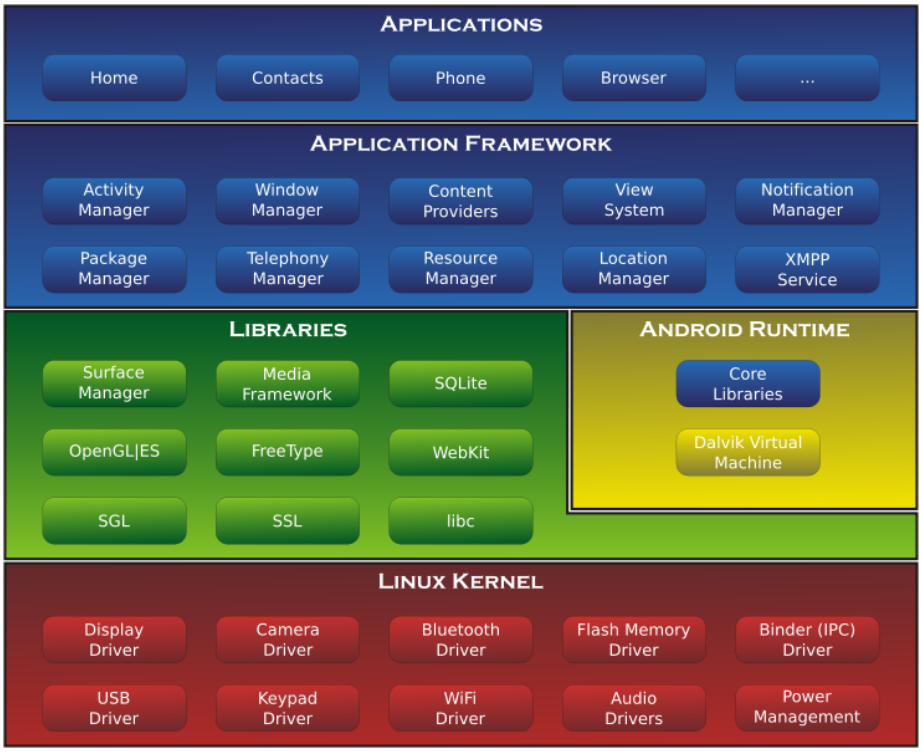
User Classes and Characteristics

- 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 loose 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.

Operating Environment

- The main development of the application exist on the Application level.
- 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



This application is supported from Android version 2.3 and up.

Version	*Code name*	*Release date*	*API level*	*Distribution*
4.4	*KitKat*	*TBA*	*TBA*	*0%*
4.3	*Jelly Bean*	*July 24, 2013*	*18*	*0%*
4.0.3–4.0.4	*Ice Cream Sandwich*	*December 16, 2011*	*15*	*21.7%*
3.2	*Honeycomb*	*July 15, 2011*	*13*	*0.1%*
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%*

User Documentation

- User manual
- Examples

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 MySQL) 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 on it.
    - If a meal is not present in the database, the user can add it by either indicating the total calories on it, or by indicating the meal ingredients from the actual database.
    - End users cannot 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.
    - From existing 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 text 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.
    - New meal
        - Indicate caloric content
            - User has the options:
                - Name: If repeated, send an error text 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 text 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 return to previous screen using back button.
    - Monthly
        - Display last month caloric intake.
        - User must return to previous screen using back button.

- Modify database
    - Modify ingredient
        - Modify ingredient info
            - Modify name: If repeated, send an error text 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 return to previous screen using back button.
        - Delete meal

3.1.3 Functional Requirements

REQ-1: There must be a Light MySQL 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 text 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 MySQL).

- Caloric inferior limit.
- Caloric superior limit.

These values mus 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 get 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 then I skipped dinner (0kcal)."

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

Daily intake was not balanced because:

1. Banana and yogurt breakfast had less than 533 kcal.
2. Doughnut snack had less than 533 kcal.
3. Big Mac lunch had more than 666 kcal.
4. Dinner was skipped.

Priority: Low

3.3.2 Stimulus/Response Sequences

- Balance meals
  - Display if the 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(s) in the App Database (based on light MySQL) for storage all the points of the user within the time (weeks. This table(s) should have views of the most commons queries required for this feature.

4. External Interface Requirements

User Interfaces

In the following table, 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 search food/ingredients
UI-4	Custom meals view: The user can add, modify, search and delete the custom meals.
UI-5	Records view: The user can add, modify and delete the records of the calories or meals consumed.
UI-6	Meals view: The user can view a several meals with the calorie information.
UI-7	Calorie indicator view: The user can view the quantity of calories consumed i the current day.
UI-8	Reports view: The user can show the daily and weely report for the consume 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.

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.

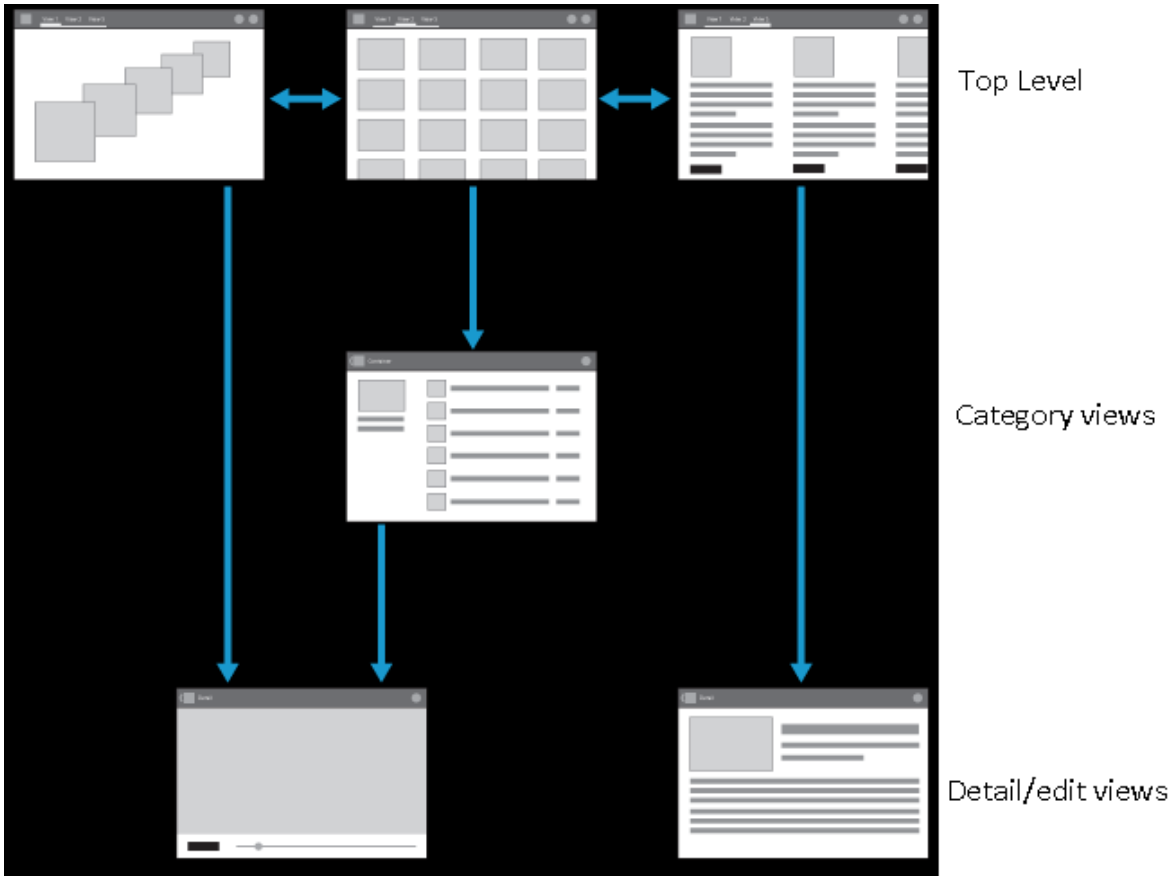
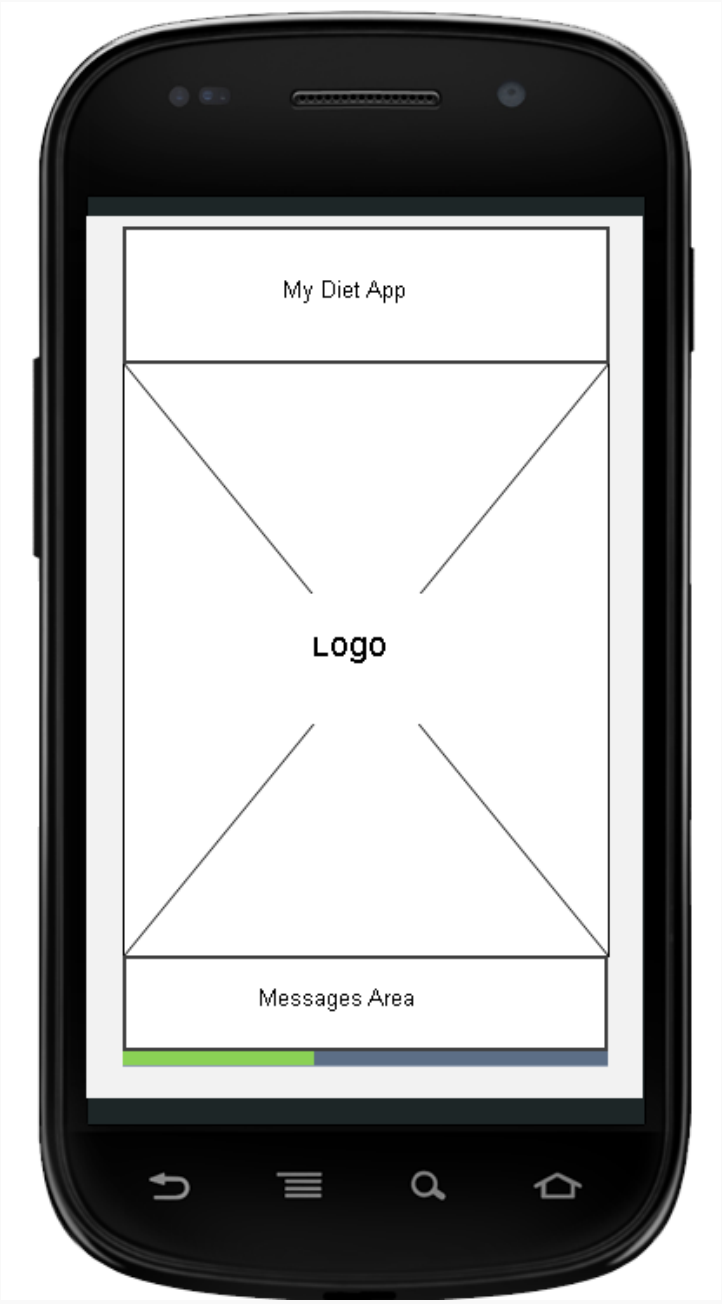


Figure 3: Taken from (http://developer.android.com/design/patterns/app-structure.html)

In general, My Diet App is going to follow the Android Design Patterns (http://developer.android.com/design/patterns/index.html) in order to develop quality user interfaces.

Prototypes

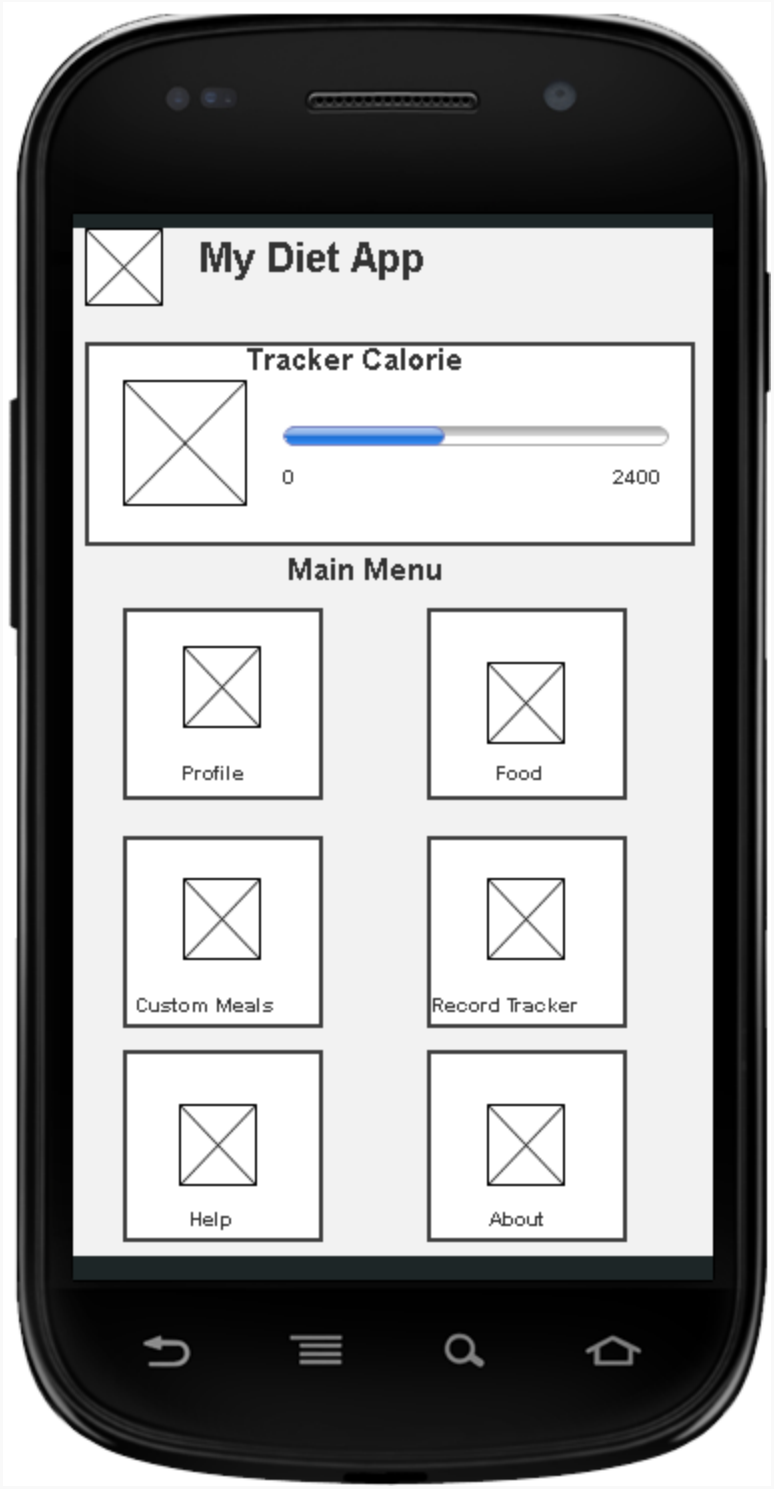
1. Start template: This is the template when My Diet App is loading.



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.

3. Sub-main 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.

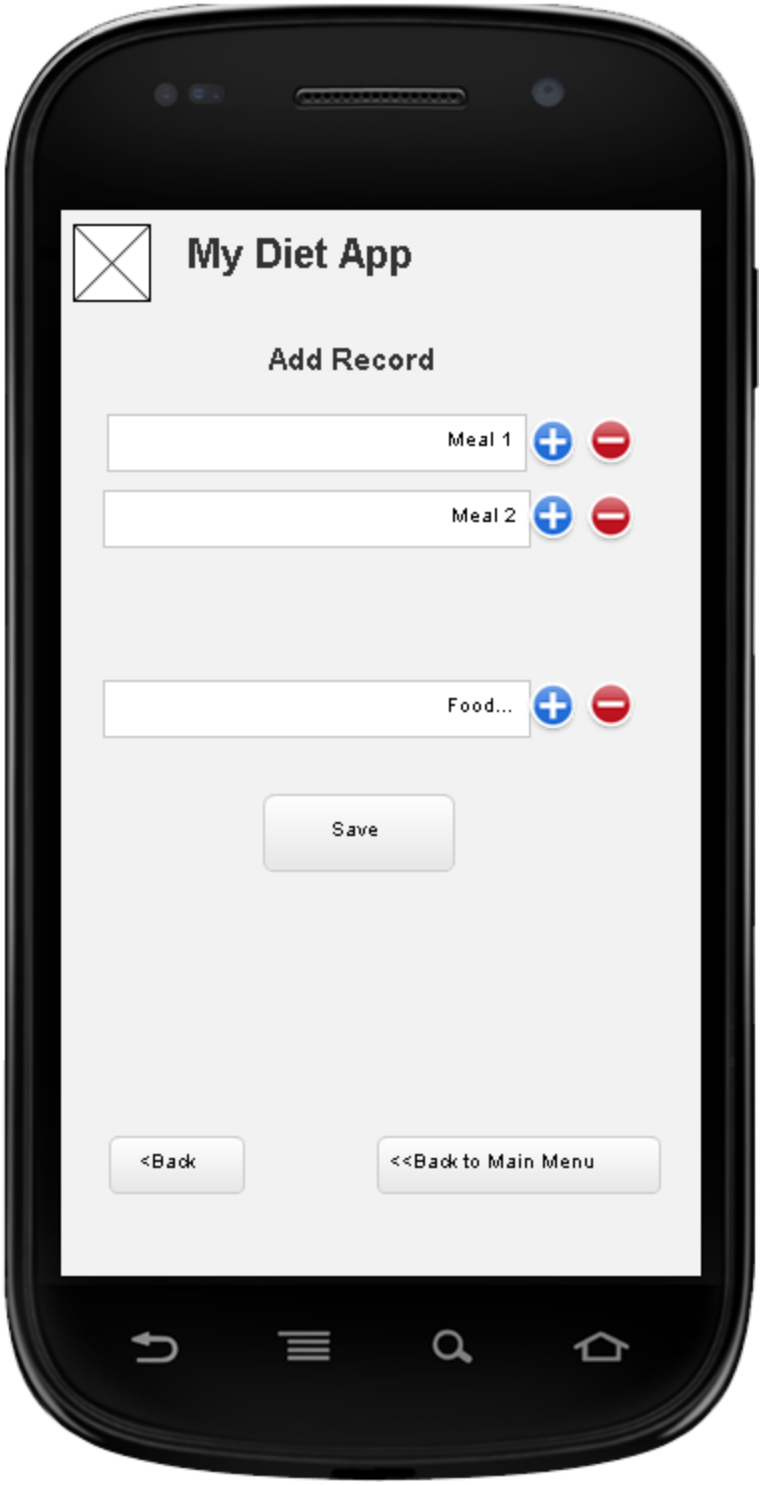
4. Main menu (category view)



5. Profile


My Diet App
Profile
<<Select your gender>>
<<Select your activity>>
2400 calories
field...
Save
Cancel

6. Add record. Detail View



Hardware Interfaces

Not apply to My Diet Project.

Supported Devices:

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 and 4.3
Samsung	i9305 Galaxy S III	4.1.1/4.1.2 with TouchWiz Nature UX UI, upgradeable to 4.3
Samsung	i9505 Galaxy S4(with LTE)	4.2.2 with TouchWiz Nature UX 2.0 UI, upgradeable to Android 4.3
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

Software Interfaces

Database schema: The database will exist to store food, meals, recipes and daily records.

Continuos 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.

Communications Interfaces

Connection to Light MySQL Database for configuration.

5. *Other Nonfunctional Requirements*

Performance Requirements

As on every mobile application, having a good performance is one of the most important aspects to care about in order to give a good user experience.

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

Safety Requirements

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

2. 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.

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.