[Get Fit]

# Project Description

This is a fitness tracker that will record calorie intake as well as calories burned. Users may input meals eaten, which will return an estimate of calories taken in. Workouts will also be recorded and an estimate of calories burned will be shown. The target audience is for those looking to lose weight, and those who are trying to have a stricter diet. This is a health tracking application for mobile devices.

# Part 1: Non-Functional Requirements

1. The system is to be an Android mobile application
2. Recommended specs:
   1. Android OS 4.4 (Kit Kat)
   2. Screen Resolution: 1080 x 1920
   3. Screen Size: 5.0in
   4. Available storage space of at least 500MB
3. The system will be thouroughly tested prior to delivery.
   1. The system must be shown to have functional errors in no more that 15% of delivered requirements.

# Part 2: Functional Requirements

## Glossary

* “System” – the delivered software and any additional resources required to make it function.
* “User” – any person who wishes to user/interact with the application.
* “Meal” – a recipe created from the ingredients already included in the database.
* “Ingredient” – All the base entries to the database. Users may not create these.
* “Workouts” – a set of directions created by a user or already entered into the database.

Each requirement has a priority level of [1], [2], or [3].

1. Must-have functionality critical to the problem solution.
2. Highly desirable feature that should be included.
3. Optional requirements that will be completed if time allows.

## Account creation

1. Users must create an account in order to use the application[1]
2. The following required information is stored:
   1. Unique Username[1]
   2. Password[1]
   3. Unique Email[1]
3. The following optional information is stored:
   1. Age
   2. Weight
   3. Height
   4. Goals
      1. Muscle loss/gain/maintain (Goal Muscle)
      2. Weight loss/gain/maintain (Goal Weight)
      3. Calorie Quota is created if current weight and goal weight are entered.
   5. Gender
      1. Disclaimer: These are recommednations. This is not guaranteed to work for everyone.

## Ingredients

1. A list of base ingredients – users cannot add to this table[1]
   1. The system stores the following required information about the ingredient:
      1. Calories per Amount(ounce, tsp, etc)[1]
      2. Category(vegetable, meat, etc)[3]
      3. Ingredient name[1]

## Meals

1. A user can add a meal to their own library[1]
   1. The system stores the following required information about a meal:
      1. Ingredients [1]
         1. User searches for these ingredients in a search bar and adds them to the Meal
      2. Calories [1]
      3. Servings [1]
      4. Publication Date [1]
      5. Meal name[1]
   2. The following meta-data may be searched for by users:
      1. Recommended for (weight loss, weight gain, etc) [2]
      2. Dietary Restrictions[2]
      3. Meal type (breakfast, lunch, dinner, snack) [1]
2. A user can add/delete their meal to the system for other users to see[2]
   1. Required stored information:
      1. Creator[2]
   2. To delete, user attempting action must match Creator [2]
3. A user can rate a meal [3].
   1. Stars 1-5 on meals created by other users

## Meal Search

1. Meals will be filtered based on calorie quota [2]
   1. If calorie quota is 2000, only meals 2000 and under are shown.
   2. As meals are eaten, calories are deducted from quota, thus narrowing search results
2. May filter based on Recommendations for weight loss, muscle gain, etc, as well as Dietary Restrictions, and Meal type. (All meta data previously mentioned) [2]

## Workouts

1. Users can add workouts to their own library[1]
   1. The system stores the following required information about a workout:
      1. Workout type (strength training, cardio)[1]
      2. Length [1]
      3. Workout name[1]
2. Users can add workouts to the system for other users to see[2]
   1. Required stored information:
      1. Creator[2]

## Timer

1. There is a timer that users can use [3]
   1. Will make a noise every 5 minutes [3]

# Part 3: Basic Design

Before developing any major project, you should first have a design and a plan. For this section, consider your logical architecture and produce a reasonably complete design to express it. Most commonly, a UML diagram will suffice, but supplementing that with wireframe designs of the UI, flow chart diagrams, or other design documents is acceptable and encouraged.

Your design should encapsulate the logical architecture (classes, interfaces, data members, methods, etc) and a component model as needed (these show the relationships between things like a server, mobile device, local application, etc).

# Part 4: Project Plan

Project length: May 8th – June 5th

Phase 1: Project Proposal/Training of Xamarin

Phase 2:

Set up database – create tables and begin entry of test data/use Entity framework.

begin structuring of code – have a basic skeleton of all the [1] priorities allowing us to easier flesh it out in the following phase.

allow login – User creation is now possible, making it possible to access the application.

Phase 3:

Have a working prototype – Users may create meals and workouts that will be saved in their own library.

Start on Priority [2] – Make it possible for users to add their meals/workouts to the database (rough)

Phase 4:

User may add to the database – this is now fully functioning. Make it so they may now delete their own items as well.

Phase 5:

Debug – Fix any issues that may be popping up.

add [3] priorities – if time allows, begin implementing [3] priorities such as the timer and rating of meals