Software Requirements Specification

for

ApolloFitness

**Version 1.0.0 approved**

**Prepared by Team Das Technik**

**Das Technik**

**Feb. 20, 2020**

**Table of Contents**

[**Introduction**](#_kccddjtfjbl3) **1**

[**Purpose**](#_3znysh7) **1**

[**Document Conventions**](#_xoc0dvpj1a0d) **1**

[**Intended Audience and Reading Suggestions**](#_7eaic7sfaopx) **1**

[**Product Scope**](#_9xlocgedurcy) **1**

[**References**](#_xixbjupj9cgl) **2**

[**Overall Description**](#_pvgs9w2i8n91) **2**

[**Product Perspective**](#_2s8eyo1) **2**

[**Product Functions**](#_1ssucfhf4xn7) **2**

[**User Classes and Characteristics**](#_73kfsyavl5ah) **2**

[**Operating Environment**](#_v0g6dmz266nn) **3**

[**Design and Implementation Constraints**](#_e8yd5lyyk26j) **3**

[**User Documentation**](#_wi7ze19012bl) **3**

[**Assumptions and Dependencies**](#_s05bp2od5std) **4**

[**External Interface Requirements**](#_9dfshi3j1plh) **4**

[**User Interfaces**](#_2jxsxqh) **4**

[**Hardware Interfaces**](#_pnilcs4otclb) **4**

[**Software Interfaces**](#_n6l0yxdqivq1) **5**

[**Communications Interfaces**](#_zft712a2q8xg) **5**

[**System Features**](#_tmq92eoitri5) **5**

[**Nutrition Tracking**](#_j028nupa7vsn) **5**

[**Hydration Tracking**](#_jq0bsjd03lgs) **6**

[**Profile/User Settings**](#_sv2mlib67jga) **7**

[**Other Nonfunctional Requirements**](#_3whwml4) **8**

[**Performance Requirements**](#_2bn6wsx) **8**

[**Safety Requirements**](#_8l15e4s4l9m6) **8**

[**Security Requirements**](#_bkrov362x8gy) **8**

[**Software Quality Attributes**](#_d0mo8zcqmap) **8**

[**Business Rules**](#_c7afsy7ehg9m) **9**

[**Other Requirements**](#_2538qk4isyyh) **9**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
|  |  |  |  |

# Introduction

## Purpose

This SRS document covers the software requirements specifications of Das Technik’s ApolloFitness Android application, version 1.0.0. The application is a single-user hydration, nutrition and mood tracking mobile application intended to help users track nutrition and mood in their day to day lives. This document covers the entire scope of the application and all of it’s systems and subsystems.

## Document Conventions

This document follows standardized conventions of Times New Roman size 11 font for all body paragraphs, size 18 bold for headers, and size 14 for sub headers. There are no special fonts or highlighting conventions. References to other sections of the document shall be written in italics. Each requirement statement will have it’s own specified priority written in full capitalization next to it.

## Intended Audience and Reading Suggestions

This document is intended for the development team, course instructors and the end user. It contains an introduction to the topics covered in this document, an overall description on the project, some external interface requirements, system features, and a list of nonfunctional requirements. To get a full scope of the project it’s best to read the entire document sequentially, however for information specific to your needs consult the following list.

Developers: In order to effectively navigate, the development team may focus on *Sections 3 - 5* in order to understand technical needs and requirements to work on the application.

Course Instructors: The course instructors and end users may want to view the document in its entirety to gain an initial high level view of the application. As the reader progresses to later sections of the document they will be exposed to lower level technical details of the project.

End users: A user of the ApolloFitness application may consider focusing their attention to the details provided in *Section 4: System Features* to better understand the purpose and how to use the application.

## Product Scope

ApolloFitness is a personal health tracking application. The primary goal of the application is to provide a single source repository for mood, hydration and nutrition. These aspects are proven to be key components to improving and maintaining personal health, much higher than exercise alone.

In order to accomplish this task, ApolloFitness takes a personalized approach to interaction. The application will push motivational and personalized notifications to engage the user. These will encourage the user to continue using the app, continue to meet their goals and ultimately, improve their personal well being. By promoting daily user interaction that translates to measurable personal results, ApolloFitness should see an increase in its user base by recommendation.

## References

The Food Data Central API developed by the US Department of Agriculture can be found at the following link: <https://fdc.nal.usda.gov/api-guide.html>. This can also be searched on the USDA’s National Agricultural Library webpage.

# Overall Description

## Product Perspective

ApolloFitness is a new self-contained mobile application designed for personal health tracking. The application will be externally supported by the Food Data Central API developed by the US Department of Agriculture.

## Product Functions

* Single user biometric sign up (Name, Age, Height/Weight, Target Goal)
* Tab Navigation interface
* Home tab: Greeting and mood input field, information at a glance
* Hydration tab: Water intake meter, quick glass button, manual entry field, and graphed hydration data
* Nutrition tab: Calorie intake meter, drop down field for food entry, and graphed nutrition data

## User Classes and Characteristics

**Mood-focused user:** This user will be drawn for the mood tracking functionality of the application. The user will be able to track their mood and identify connections to their mood combined with a healthy diet.

**Hydration-focused user:** This user will take advantage of the water reminding push notifications. The user will be able to quickly see their daily intake, input an entry for tracking and view their hydration activities over time.

**Nutrition-focused user:** This user will focus on the nutritional aspect of the application. The user will be able to monitor their calorie intake and macronutrients. The connected database will allow the user to enter specific foods without manually entering each ingredient.

**User physician or health coach:** The user will be able to share trends and results to a professional consultant. This information will be easy for the user to access allowing them to not use another tracking medium simultaneously.

## Operating Environment

Apollo fitness will be a self-contained mobile application. The application will be primarily designed for Android deployment on Android 5 or newer and will be cross-platform ready for iOS integration.

## Design and Implementation Constraints

Initial design will plan for a web based API to access the nutrition database. This is to ensure the application is lightweight on the user’s device. The timeline for the development of this application (15 weeks) will limit the level and depth of functionality that will be available upon submission. No other limitations or constraints on the developers are foreseen.

## User Documentation

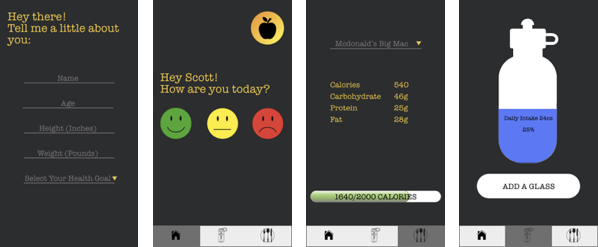
The application will walk the user through setup during initial enrollment. The use of rudimentary features such as buttons and tabs will allow the user to navigate and engage the application without need for documented tutorials. Should any problems arise for a user, an ApolloFitness email will be provided for troubleshooting.

## Assumptions and Dependencies

The FoodData Central API provides REST access to FoodData Central (FDC) developed by USDA.

# External Interface Requirements

## User Interfaces

**

## Hardware Interfaces

The application has been developed for use on mobile devices. ApolloFitness will be able to fully operate on devices running Android 5 or newer.

## Software Interfaces

ApolloFitness will primarily be developed using React-Native, a cross platform framework built with JavaScript. Data will be stored in the long term locally via a SQLite database.

## Communications Interfaces

The application will communicate with a REST API using HTTP communication standards. As a mobile device, the application will be able to utilize cellular data or wifi to communicate over the internet.

# System Features

## Nutrition Tracking

4.1.1 Description and Priority

Priority: HIGH. The nutrition tracking feature will allow the user to select and input food intake. This feature will interact with a database using a REST API. The user will be able to visually see their food items summary, macronutrient and calories totals. These totals will be calculated per day and displayed as an ‘over time’ view as well.

If an item is not available via the database. The user will be able to manually enter data into the fields to update their stats.

4.1.2 Stimulus/Response Sequences

Stimulus: User selects food items from the drop down menu.

Response: System searches for food items using API. System updates calorie and macronutrient values based on returned values.

Stimulus: User presses on calories meter display.

Response: System returns higher detail view including macronutrient values and graphical history.

Stimulus: User selects ‘other’ from drop down menu.

Response: System provides text fields for manual entry of food item and values.

Stimulus: User manually enters values into name and calories fields (minimum).

Response: System updates associated values and displays new values.

4.1.3 Functional Requirements

REQ-1: The app nutrition screen shall display a drop down field that the user may scroll or type.

REQ-2: The app nutrition screen shall display a meter showing the calories entered total for the day.

REQ-3: The app nutrition screen shall clear the calories entered total each day.

REQ-4: The app nutrition screen shall record the entered values and update the associated fields.

## Hydration Tracking

4.2.1 Description and Priority

Priority: HIGH. The hydration tracking feature is provided to allow a user to input and view their daily water intake. A meter will be displayed reflecting their total intake based on their goal. The goal will be calculated using a defined weight based calculation.

4.2.2 Stimulus/Response Sequences

Stimulus: User clicks the add a glass button.

Response: System updates and displays the water intake value by 8 ounces.

Stimulus: User manually enters water volume consumed.

Response: System updates and displayed the water intake based on the input value.

4.2.3 Functional Requirements

REQ-1: The hydration tracking screen shall display the water goal based on the recommended water intake for the users weight [total ounces = (pounds/2)]

REQ-2: The hydration tracking screen shall display a button to add a single glass of water to the total intake value.

REQ-3: The hydration tracking screen shall update and display the total intake value in response to button press or manual entry.

## Profile/User Settings

4.3.1 Description and Priority

Priority: MEDIUM. This feature will provide an initial on-boarding view that requires a user to enter their name, gender, age, height, weight and fitness goal. These metrics will all the application to calculate and set the recommended goals for the user. The user will also be able to interact with a settings menu to update these values since user biometrics are dynamic.

4.3.2 Stimulus/Response Sequences

Stimulus: User launches app for first time

Response: System displays a welcome window requesting the user to input biometric data.

Stimulus: User changes age, height, weight or fitness goal values.

Response: System recalculates nutrition and hydration goals to reflect the updated values.

4.3.3 Functional Requirements

REQ-1: The application shall request on-boarding details upon initial launch.

REQ-2: The application shall save user biometrics and not re-display on-boarding menu during subsequent launches.

REQ-3: The application shall recalculate target values and retain history when user changes biometric fields.

# Other Nonfunctional Requirements

## Performance Requirements

The nutrition database will initially be designed for over-the-air connection to limit the space demands of the application. Usage history and data will be stored locally on the user’s device to improve response time to inquiries.

## Safety Requirements

Disclaimer will be provided within the application notifying the user that the application should not be a replacement to guidance from a licensed healthcare professional.

## Security Requirements

All data will be stored locally on the user’s device. The application will not need to provide any encryption or additional security measures to conceal user data. Users will be able to share their information over secure channels of their own choosing.

## Software Quality Attributes

The software stack used in application development provides the development team with cross platform capability. Source code updates will provide updates to all platforms served by the application.

The components developed within the application will be capable of reuse in similar applications (or additional features) at the discretion of the development team.

## Business Rules

No applicable business rules need be applied in the development of this application.

# Other Requirements

No additional requirements or considerations required.

**Appendix A: Glossary**

**API** - Application programming interface. A set of procedures and functions enabling the creation of applications which can access an operating system, application, or other device.

**HTTP** - Hypertext transfer protocol. An application protocol defining how data should be formatted and transmitted and how web servers should respond to certain commands.

**REST** - Representational state transfer. An architectural style in computer software that defines constraints to be used in the creation of web services.

**SQlite** - An end program embedded relational database management system.

**iOS** - Mobile operating system designed by Apple Inc.

**React-Native** - A Facebook created open-source mobile application framework used to develop applications for Android and iOS.