Software Requirements Specification

for

BlurBodyBalance

Version 1.0 approved

Prepared by the Blur 1 Team

University of Maine

Feb. 21, 2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Kenneth Seneres	2/21/2019	Initial creation	0.1

Sam Beaudoin	3/10/2019	Revisions from deliverable 1 comments	0.1

1. Introduction

1.1 Purpose

This document is created for the the fitness application "BlurBodyBalance" version 1.0. This document encompasses the complete software stack of the product.

1.2 Document Conventions

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

This document is intended mostly for developers and related staff that need to know about the software's system features, external interfaces, and nonfunctional requirements. A section that gives an overview of the product is created for both technical and non-technical staff alike. It is suggested that the document be read in order, but non-technical staff need only read up to the overview section.

1.4 Product Scope

The software is a fitness application intended for an average-shape individual of young to middle age seeking to be more active to log exercise, calories, and weight. It is a tool to help people keep track of their activity on their path to fitness with the goal of encouraging people to get on and stay on the path to fitness.

1.5 References

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

2. Overall Description

2.1 Product Perspective

The product is a new, self-contained product that seeks to carve a niche on existing generic fitness applications. The product is created with the intention of improving on already existing applications.*

2.2   Product Functions

Listed below are some of the major functions that the product shall perform:
- The application shall allow the user to track their changing weight.
- The application shall allow the user to input exercise information.
- The application shall track the users walk/run distance.
- The application shall allow the user to input diet information.
- The application shall allow the user to create fitness goals.
- The application shall create subgoals based on the user's goals.
- The application shall display all of the above information using diagrams.
- The application shall contain a personality that makes comments about the user to keep the user motivated.

2.3   User Classes and Characteristics

User classes anticipated to use this product would be able-bodied individuals of all ages looking to start getting fit with a basic knowledge of common touch interaction practices, such as swiping and tapping.

2.4   Operating Environment

The software will operate on smartphone devices running one of the two major mobile operating systems, iOS and Android, targeting the latest version of each but backwards compatible with older versions of both operating systems. There are no other software components or applications that the software needs to coexist with, other than the infrastructure provided by the operating system to perform standard application functions.

2.5   Design and Implementation Constraints

Some design and implementation constraints are listed below:
- The application must be designed under the Android framework.
- The application must run on a typical budget device's memory constraint of 1 GB RAM.
- The application will be programmed using an object-oriented programming language.
- The application shall conform to USA privacy laws.
- The application shall adhere to the Association for Computing Machinery code of ethics.

2.6   User Documentation

There are no user documentation documents available at this time.

2.7   Assumptions and Dependencies

Third-party components that might be used in the near future are listed below:
- Google authentication of user accounts

3. External Interface Requirements

3.1 User Interfaces

User interfaces have not been designed at this time.

3.2 Hardware Interfaces

The software runs on smartphone hardware that includes a capacitive touch screen, so interactions with the software will be performed through the hardware's aforementioned touch screen. There are no plans to use microphones, light sensors, biometrics, and cameras, which are also common on smartphones.

3.3 Software Interfaces

There are currently no specific software components that interface with the product at this time.

3.4 Communications Interfaces

There are no communications interfaces implemented with the product, since the product currently only requires local information.

4. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Input weight, exercise and diet information.

4.1.1 Description and Priority

The feature is a simple input, but is the primary functionality of the application, so it is under high priority.

4.1.2 Stimulus/Response Sequences

The user will navigate through the application to the input page with a few taps and tap on fields relevant to the information to put, such as weight, diet, or other, and input numbers or letters depending on the field.

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and

necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: The system shall display the weight of the user from the last 30 days on a line graph.

REQ-2: They system shall display diet information from the past 30 days on a multi-line graph.

REQ-3:

4.2   Create fitness goals.

4.2.1   Description and Priority

The feature is also simple input, but provides important functionality of the application, and it is under medium-high priority.

4.2.2   Stimulus/Response Sequences

The user will navigate through the application to the input page with a few taps and tap on a field to type in a target for a specific category, such as steps, caloric intake, or sleep amount.

4.2.3   Functional Requirements

REQ-1: The system shall display a progress bar to visualize how close the user is to their goals.

REQ-2: The system shall determine if a goal is reasonable by a function of the user's age, height, and current weight.

4.3   Keep the individual motivated.

4.3.1   Description and Priority

The feature delivers messages to the user depending on whether or not they've been keeping healthy exercise habits and hitting goal targets. The feature is optional but on by default, but provides character to the application, and is of medium priority

4.3.2   Stimulus/Response Sequences

The system delivers messages periodically as the user uses the application. The content of the messages can vary based on how well the user is doing. The user can navigate to a settings page and turn the feature off.

4.3.3   Functional Requirements

REQ-1: The system shall send a notification to the user after 5 days without use.

REQ-2: The system shall make positive comments when the user makes positive progress towards their goal.

REQ-3:   The system shall make negative/teasing comments when the user makes negative progress towards their goal 3 times in a row.

5.   Other Nonfunctional Requirements

5.1   Performance Requirements

- The system shall be deployed on Android devices.
- The system shall respond to user input within 500ms.
- The system shall take up less than 200MB of space.

5.2   Safety Requirements

- The system shall inform the user to be safe when performing fitness activities before the user registers.

5.3   Security Requirements

- The systems shall resist unauthorised, accidental or unintended usage and provide access only to registered users.

5.4   Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

5.5   Business Rules

<List any operating principles about the product, such as which individuals or roles can perform which functions under specific circumstances. These are not functional requirements in themselves, but they may imply certain functional requirements to enforce the rules.>

6.   Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>