# Software Requirements Specification

# for

# ILLO

Version 1.0 approved

Prepared by   
Cole Adams, Evan Mutchler, Johnny Sylvain,  
Riley Mills, Zachary Scott

Root-Digital

18 February 2023

Table of Contents

**Table of Contents** [**ii**](#__RefHeading___Toc441230970)

Revision History [ii](#__RefHeading___Toc441230971)

1. Introduction [1](#__RefHeading___Toc441230972)

1.1 Purpose [1](#__RefHeading___Toc441230973)

1.2 Document Conventions [1](#__RefHeading___Toc441230974)

1.3 Intended Audience and Reading Suggestions [1](#__RefHeading___Toc441230975)

1.4 Product Scope [1](#__RefHeading___Toc441230976)

1.5 References [1](#__RefHeading___Toc441230977)

2. Overall Description [2](#__RefHeading___Toc441230978)

2.1 Product Perspective [2](#__RefHeading___Toc441230979)

2.2 Product Functions [2](#__RefHeading___Toc441230980)

2.3 User Classes and Characteristics [2](#__RefHeading___Toc441230981)

2.4 Operating Environment [2](#__RefHeading___Toc441230982)

2.5 Design and Implementation Constraints [2](#__RefHeading___Toc441230983)

2.6 User Documentation [2](#__RefHeading___Toc441230984)

2.7 Assumptions and Dependencies [3](#__RefHeading___Toc441230985)

3. External Interface Requirements [3](#__RefHeading___Toc441230986)

3.1 User Interfaces [3](#__RefHeading___Toc441230987)

3.2 Hardware Interfaces [3](#__RefHeading___Toc441230988)

3.3 Software Interfaces [3](#__RefHeading___Toc441230989)

3.4 Communications Interfaces [3](#__RefHeading___Toc441230990)

4. System Features [4](#__RefHeading___Toc441230991)

4.x Functional Requirement x [4](#__RefHeading___Toc441230992)

5. Other Nonfunctional Requirements [4](#__RefHeading___Toc441230994)

5.x Nonfunctional Requirement x [4](#__RefHeading___Toc441230995)

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Zachary Scott | 18 Feb | Document created and fashioned for assignment’s purposes, | 1.0 |
| Zachary Scott | 18 Feb | 10 Functional requirements added. | 1.1 |

# Introduction

## Purpose

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

## 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.>

## Intended Audience and Reading Suggestions

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

## Product Scope

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

## 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.>

# Overall Description

## Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

## Product Functions

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

## User Classes and Characteristics

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

## Operating Environment

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

## Design and Implementation Constraints

<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>

## User Documentation

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

## Assumptions and Dependencies

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

# External Interface Requirements

## User Interfaces

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

# System Features (Functional Requirements)

<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.>

## Functional Requirement 1

Illo shall allow the user to toggle playback on the pomodoro timer at will.

## Functional Requirement 2

Illo shall allow the user to let the pomodoro timer repeat indefinitely in “Zen mode.”

## Functional Requirement 3

Illo shall notify the user upon the completion of a timer.

## Functional Requirement 4

Illo shall be able to produce a random exercise for the break interval of a pomodoro.

## Functional Requirement 5

Illo shall be allow the user to select a set of exercises to randomly select from for the break interval of a pomodoro.

## Functional Requirement 6

Illo shall allow the user to specify an order of appearance of up to 20 exercises for subsequent break intervals.

## Functional Requirement 7

Illo shall allow the user to skip an exercise in favor of the exercise that would have appeared in the next break interval.

## Functional Requirement 8

Illo shall allow the user to specify the length of the work interval of a pomodoro.

## Functional Requirement 9

Illo shall provide the user with a randomly selected inspirational quote during the work interval of a pomodoro.

## Functional Requirement 10

Illo shall provide visual guides to the exercises presented during the exercise interval of a pomodoro.

## Functional Requirement 11

## Functional Requirement 12

## Functional Requirement 13

## Functional Requirement 14

## Functional Requirement 15

## Functional Requirement 16

## Functional Requirement 17

## Functional Requirement 18

## Functional Requirement 19

## Functional Requirement 20

# Other Nonfunctional Requirements

## Nonfunctional Requirement 1

## Nonfunctional Requirement 2

## Nonfunctional Requirement 3

## Nonfunctional Requirement 4

## Nonfunctional Requirement 5

## Nonfunctional Requirement 6

## Nonfunctional Requirement 7

## Nonfunctional Requirement 8

## Nonfunctional Requirement 9

## Nonfunctional Requirement 10