Quark App

System Requirements Specification

Version 1.0 approved

Prepared by Jazer Barclay

University of Brighton

22nd March 2022

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Jazer Barclay | 22/03/22 | Initial Version | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose for this document is to explain the process by which the preliminary requirements from our project planning were analysed and refined.

## Intended Audience and Reading Suggestions

This product is primarily targeted at knowledge workers in the academic field such as students in both foundational and further education and researchers.

This app, however, can be used for simple time tracking over extended periods of time allowing a more general usage.

## Product Scope

This product is a software application which is accessed from a desktop application and is used to track work units in increments of 30 minutes with a split of 25 minutes of work and 5 minutes of rest.

The application extends to a website that can be accessed for historic tracking data for an individual to view and to compare against other users of the application in a ranked leader board.

## References

* IEEE 830
* Business Analysis, Requirements, and Project Management (Chapter 4) [K. Cox (2022)]
* CI536 Requirements Documentation (Week 3) by Karl Cox (2022)

## Overview

This document is laid out as per the IEEE-830 standard.

# Overall Description

This system is designed to be a facility for tracking a user’s time during extended work sessions and has many applications such as recording study sessions, research on topics that require great depth and complex writing or development tasks.

## Product Perspective

Diagram

Description automatically generatedThe time tracking application is broken down into a back-end, website and desktop application. The back-end serves as the data storage system where custom user implementations can access the infrastructure for tracked session storage. The website is where the user may view their tracked information in a standard format and view the stats of other users of the system. The desktop application serves as a simple, minimal method to track their work sessions and upload the data to the back-end system.

## Product Functions

The functionality is split between the website where insight is generated and the desktop app which serves as a method of time tracking via a fixed timer. The functions are labelled for where they will need implementing to meet the user requirements.

* Signup (Website & Application)
* Login (Website & Application)
* Start timer (Application)
* Stop timer (Application)
* Log 30 minute session (Application)
* View personal historic stats (Website)
* View other user’s historic stats (Website)
* View leader board of all users (Website)

## User Classes and Characteristics

There are three types of users in this system. The first two are authorised users and the third is non-authorised users. The first is the administrator. They have direct access to the server and thus the data, infrastructure and source code. They will be responsible for hosting and managing the services exposed to the second type of user.

The second type of user is the end user. They are the authorised users, customers, who have signed up for the service and operate the desktop application. Their main interaction is via the use of the timer and accessing their profile via the website or API.

The third type of user is the unauthorised user. They have not signed up for the service and as a result have no profile data of their own on the system. They may wish to access other end user’s data on the system which they are not allowed access to until they have signed up for the service, becoming an end user.

All of these users must have basic computer skills which include working with a web browser such as Google Chrome or Mozilla Firefox. Since all interactions with the UI of the desktop and website, the system cannot be used without access and knowledge of how desktop applications and specifically web browser functionality as part of their operating system.

## Operating Environment

The desktop application will be supported on any desktop or laptop device running an up-to-date version of the Windows, MacOS or Linux operating systems.

The website will be accessible by any modern web browser from any device with a minimum screen resolution of 280x360 and JavaScript support.

## Design and Implementation Constraints

* Data must be stored in a relational database for fast queries and optimal storage
* Passwords must be stored encrypted via a salt and hash function
* Unauthorised users must not be allowed to interact with the data via the website or associated API
* Secure communication between server and client via HTTPS connections
* Robust enough to handle server error responses and notify the end user

## User Documentation

Along side the source code for the application and website, a PDF manual for the server-side API will be available for the developer audience.

# External Interface Requirements

## User Interfaces

On the website, there will be a navigation panel that allows easy access to traverse between any pages. This will have a wildcard link that, depending on login status and active screen, will show different links.

The pages must conform to a standard layout and scheme being mobile responsive and mutable for desktop use.

The desktop application must remain small in size and be unobtrusive to the user’s view. To ensure its focus is available, the window will remain top most compared to other apps on the screen.

## Hardware Interfaces

The websites will be accessible via mobile and desktop devices with varying screen sizes. This must be taken into consideration for layout and design factors.

## Software Interfaces

The website and application will both share a connection to a server-side database via an API. This API will be public facing allowing direct interaction with it. All data stored in the database will only be visible via the API reducing the attack surface.

These communications must be over an encrypted connection via HTTPS.

## Communications Interfaces

Communications via the API and website must be over an encrypted connection via HTTPS.

# 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.>

## Website Signup

4.1.1 Description and Priority

A new or existing user can create a new account using their given email address, username and password. This is a high priority feature that is required for any user to become an active member.

4.1.2 Stimulus/Response Sequences

The user will first access the website via search engine or url. They will click on the signup button on the home page and fill in the required form. Upon completion and successful account creation, the user will be forwarded to the login screen. Here they can use their new account details to perform a login. Should the validation of user details fail upon signup, the form will display an appropriate error.

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:

REQ-2:

## System Feature 2 (and so on)