Software Requirements Specification

for

AssignmentTracker

Version 1.0 approved

Prepared by Narid Drake

05/09/2021

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** |
|  |  |  |  |
|  |  |  |  |

# Introduction

## Purpose

This document will outline the high-level specifications for the AssignmentTracker application (working title). This will outline the requirements for the entire system, including the both the front- and back-end components. The document shall be updated over time as requirements are modified and/or extended.

## 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

This document is primarily intended for posterity and reference by the developer, with the added benefit of serving as documentation for any parties reviewing the project (e.g., potential employers).

## Product Scope

The application is intended to assist users with managing their upcoming academic assignments and tests, as well as to allow them to log and monitor their productivity. The main functionality of the application will be to server as a combination of a year-planner and task manager, storing upcoming test dates and assignment deadlines while also allowing the user to keep track of which tasks have been completed and which are yet to be completed. Additional functionality will include the user being able to create a specific checklist for a particular assignment, or to log the amount of time they’ve invested into an assignment.

The overarching goal of this product is to assist the time- and task-management of students or self-motivated learners.

## 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

This is the specification design of the first version of this application. It is a personal project intended initially for use by the developer, with the potential for the application to be later deployed for mass-use.

## 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.>

The main features of this application will be to:

* Allow the user to view and filter their added assignments
* Discriminate assignments by factors such as whether they are completed, how soon they are due, to which paper/project they are related, etc.
* Allow the user to add/modify/remove tasks
* Assign task checklists to individual assignments
* Keep track of the amount of time the user has spent working on a particular assignment, which the user will declare.

## User Classes and Characteristics

* **Consumers:** These are the individuals who will be using the application for their own personal use, and for whom this application is intended. The expected demographic for this group is students, those studying at either the secondary or tertiary level (i.e., high-school and university undergraduate students), however it is possible the users may include self-directed learners outside of any academic institution.
* **Developers:** Those who will be maintaining and possibly extending and/or improving the application. Currently this use class only consists of myself (Narid Drake), however, if the application ever becomes mass-deployed it is possible more developers will be included. This use class will be the only group able to access/modify the application source code, and to make updates to the system.

## Operating Environment

The application will have a web-hosted server backend, developed using HTML, CSS, and JavaScript-based frameworks such as Node.js. The Front-end will consist of a Discord Bot that will use text-based commands to interact with the backend. Development of a browser application is intended after successful development of this initial interface. The user data will be stored using a MongoDB document-oriented database.

## Design and Implementation Constraints

During the development of the initial application, there will be limitations on financial support and available time for building the project. The initial application will therefore need to utilize free resources where possible and will need a large development window.

## 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.>

## Hardware Interfaces

<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>

## Software Interfaces

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

## Communications Interfaces

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

# 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.>

## View assignments

### Description and Priority

The user will be able to request to view assignments they’ve added in the system. They can either view all assignments they’ve created, or they filter their request to view assignments meeting some given parameters, such as whether it is completed/uncompleted or by which course they belong to.

Priority: 10/10

### Stimulus/Response Sequences

### Functional Requirements

REQ-1: A user should only be able to view their own assignments. They should not be able to view assignments belonging to other users.

REQ-2: Users will need to be able to filter their search query to include parameters such as completion status, course, and due date.

REQ-3: A user should also be able to search for a specific assignment by its name.

## Add/remove/modify assignments

### Description and Priority

The user will be able to add new assignment tasks and specify any details regarding the task, such as the name, due date, paper/course, etc. Only the name and due date should be mandatory, and any tasks not associated with a particular course/paper should be marked as a standalone task.

Priority: 9/10

### Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

### Functional Requirements

REQ-4: The user should be able to modify specific components of an assignment.

REQ-5: Authentication should be applied when adding/modifying/deleting an entry.

REQ-6: When an assignment is deleted, all checklists associated with the assignment should be deleted as well.

## User Authentication

### Description and Priority

### Stimulus/Response Sequences

### Functional Requirements

# Other Nonfunctional Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## 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.>

## 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.>

# 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.>