Software Requirements Specification

for

Studious

Version 1.0 approved

Prepared by <Author>

JHET Innovations

10.07.25

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 3

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 4

3.3 Software Interfaces 4

3.4 Communications Interfaces 4

4. System Features 5

5. Other Nonfunctional Requirements 6

5.1 Performance Requirements 6

5.2 Safety Requirements 6

5.3 Security Requirements 6

5.4 Software Quality Attributes 7

6. Other Requirements 7

7. User Stories 8

Appendix A: Glossary 8

Appendix B: To Be Determined List 8

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Person’s Name, not “Company X”. Multiple people are okay if they all worked on this version |  |  |  |
|  |  |  |  |

# 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.>

Studious v 1.0.0

Explain the purpose behind the creation of the document.

Briefly explain the scope of what will be covered? Not sure fluff it up a bit.

## 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.>

Times new roman 14pt bold

Arial 11pt bold for sub headings

Arial 10pt font for body

1.5 spacing

## 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.>

Refer to assignment 2 for ideas

Document intended for stockholders, dev team, project managers, QA team, aka basically key players with both technical and non technical backgrounds

After everyone has submitted go through and add a clear way that makes sense to read it

## 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.>

This section describes WHY we’re creating this software.

Ensure you establish the boundaries of your project – what is INCLUDED vs. EXCLUDED

Refer to Scope and Vision Document

## 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.>

Any external document, specifications? If not, say “None.”

Vision and scope doc, possibly ASMGNT2 doc too but idk yet, make sure you add the appropriate source location from github. Maybe just add the github link to Whicher reference

# 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.>

New self contained product

Diagram if u have time -> Google draw

## Product

<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.>

Refer to ASGMTN 2

Diagram could be useful -> google draw, etc

## 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.>

Most important user class is college studens

High school students

Grad students

Possibly teachers?

## 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.>

This should be short, and mainly include what types of environments you support. If it’s a web-based product, this is browsers and versions. (Don’t say “all browsers”. There are many web browsers with different versions… SPECIFICALLY list which ones you will support.)

If it’s a mobile application, provide the environment: Android or iOS? what version? Look at software specifications on the web, and see what they say.

Refer to previous assignments to see our supported systems and operating evnironemnts

## 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).>

You may have some or None. Think about it though, if you write “none”, but it is obvious you should have some, I will deduct points!

Socket.io comm tech?? idk

Refer to prev assignments for technologies and tools used

Programming standard

Add something about design conventions etc

## 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.>

Explain what you think should be provided. We may not be able to actually create all of these, but for a real product, what would you include?

Install manual

General tutorial

FAQ

Most likey .txt or .md files

## 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).>

You may have some or None. Think about it though, if you write “none”, but it is obvious you should have some, I will deduct points!

Refer to previous assignments

We are deploying on specific third-party platforms

Socket.io

# 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.>

Think about this in terms of standards, NOT specific features. For example,

* all buttons will have a black border
* all fonts will be Arial
* Draw a screen template showing “main area”, “menu here”, “status bar” and describe each component. If you have multiple screen layouts depending on the user’s current task/settings, describe them
* What screen resolutions will you support?
* Will you be Section 508 compliant? Are there any other standards you support?
* etc…

As you develop this section you should continually message ideas to the discord. Make a few sketches and send them and see what we all agree on. The specifics like fonts, button borders, etc can be decided after we determine the general UI layout we all want.

* We’ll try and support most screen resolutions but to be determined, look into how difficult that would be to implement
* Research Secton 508 and other standards and message group to see whether we should be compliant or not

## 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.>

If you system doesn’t include hardware, then you’ll have none. If it has hardware components, then you should describe (at a high level) how you interface with that hardware.

I don’t think we’ll have hardware, not 100% sure

## 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.>

These are internal connections to things like databases, web servers. You mainly need to explain that you have them, but (for COCS412) I don’t expect detailed information about how you actually connect to them and use them. Just explain that you **do** connect to them and use them for storage of customer information, or to process incoming web requests, etc… These are internal components of your system.

Refer to previous assignments for help

Mention our databases, how we communicate with them (high level), how the users data will be sent from front end to back end, how it will be processed, etc.

## 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.>

These are external communication mechanisms. Do you connect to a bank computer to verify credit card information? That is NOT part of your system, so it is an external communication you have. Describe it here. Do you have other systems connecting in to yours to perform some function? That would also go here.

Possibly socket.io? research if that is an external communication

# 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.>

<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 – FR#1, OR REQ-1, ETC.>

<Be certain to cover all aspects of functionality, including but not limited to:

* Search
* View
* Add
* Update
* Delete
* Integration
* Data/Information Statements for your information landscape, such as:

FR#XX - The system will store and manage information for Customers, including:

-Customer Name

-Customer Address

-etc

-etc

Below is an EXAMPLE set of Functional Requirements for format, form, completeness, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **Req #** | **Features/**  **Function** | **Functional Requirement Statement** | **Priority**  **(L, M, H)** |
| FR#1 | Access  Authentication | The system will allow users to create an account profile using a provided link. | H |
| FR#2 | Access  Authentication | The system must allow users to delete their account profile. | H |
| FR#3 | Access  Authentication | The system must allow users to retrieve their username. | H |
| FR#4 | Access  Authentication | The system must allow users to reset their password. | H |
| FR#5 | Access  Authentication | The system must allow users to log out of the system. | H |
| FR#6 | Profile Management | The system must allow users to modify their account information. | H |
| FR#7 | Profile Management | The system must allow users to access their account profile information while offline. | M |
| FR#8 | Appointment Management | The system must allow users to create an appointment. | H |
| FR#9 | Appointment Management | The system must allow users to modify appointment information. | H |
| FR#10 | Appointment Management | The system must allow users to cancel appointments. | H |

# 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.>

Only work when connected to wifi

Think of more, communicate w team for help, possibly use chatgpt for ideas (make sure they are realistic and align with the project/other members work)

|  |  |  |  |
| --- | --- | --- | --- |
| **Req #** | **Technical**  **Category** | **Non-Functional Requirement Statement** | **Priority**  **(L, M, H)** |
| PR#1 | Performance | The system will …….. | H |

## 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.>

|  |  |  |  |
| --- | --- | --- | --- |
| **Req #** | **Technical**  **Category** | **Non-Functional Requirement Statement** | **Priority**  **(L, M, H)** |
| SR#1 | Safety | The system will …….. | H |

Research any policies or regulations that would apply to our project

Research if well have any of these

## 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.>

JSON Web tokens for authentiaiton

Researfh external security policies

We want secure authentication

We want users to trust the system

Find stuff that satifies above thigns

|  |  |  |  |
| --- | --- | --- | --- |
| **Req #** | **Technical**  **Category** | **Non-Functional Requirement Statement** | **Priority**  **(L, M, H)** |
| SC#1 | Security | The system will …….. | H |

## 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.>

Key ones are gonna be maintainability, testability, useability, scaleability, correctness, etc etc etc

|  |  |  |  |
| --- | --- | --- | --- |
| **Req #** | **Technical**  **Category** | **Non-Functional Requirement Statement** | **Priority**  **(L, M, H)** |
| SQ#1 |  | The system will …….. | H |

Add any and all of the non-functional requirements defined below, to ensure a complete set of technical and system requirements. Each may include a Section and table of appropriate requirements:

***Technical Requirements Statements***

* *Infrastructure/Connectivity*
* *Hardware*
* *Usability*
* *Technology*

***System Requirements Statements – covering all areas defined below***

* *Data Retention*
* *Backup & Recovery*
* *System Audit / Logging*
* *Service Level Agreements*
* *Disaster Recovery*
* *Contingency Plans*

# 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.>

Resarch something that may be added here. Possibly work with persons 3 and 4 to figure out what requirements they DON’T have listed in their section

# User Stories

<Define a series of user stories representative of the core functionality from the perspective of your various users. The user stories will define what the system will do and WHY the system will do it for the specific users. Follow the format below, and ensure your user stories are aligned to the functionality you defined above in your functional requirements>

<Below are some examples using the format required>

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **USER STORY ID** |  | **ROLE** |  | **FUNCTION PERFORMED** |  | **BUSINESS REASON** |
| US#1 | As a | Property Manager | I need the system to | provide a map-based UI containing all relevant property management data layers that can be interacted with to obtain property details | so that I may | have a centralized application for obtaining property information in a geographic context, to use as the launch point for various property management workflows. |
| US#2 | As a |  | I need the system to |  | so that I may |  |
| US#3 | As a |  | I need the system to |  | so that I may |  |
| US#4 | As a |  | I need the system to |  | so that I may |  |

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.>

After everyone has submitted their work go through and fill out the Appendix sections

Appendix B: 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.>

List here any open questions or things you know still need to be done to the SRS, but haven’t been addressed yet.

Not sure what tbd means if that means documents or features or idk. Researfh the correct answer.