
Software Requirements Specification

for

Comfort Studies

Version 1.3 approved

Prepared by Clay Akins, Josh King, Link Maynard, Andrew Unger

CarRamRod

02/12/2019

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 Schedule a Study Room	4
4.2 Request a Tutor	4
4.3 Order Catering to Study Room	
4.4 Order Food for Pickup from Emporium	
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
Link Maynard	1/24/19	Initial changes from template Section 1.1-1.2	1.0
Josh King	2/7/19	Initial changes from template sections: 2.4, 2.7 Added Entity-Relationship Diagram V2	1.1
Link Maynard	2/12/19	Added Section 2.3, 5.1-5.5	1.2
Clay Akins	2/12/19	Added Sections 2.1, 2.2, 2.6, 4.1, 4.2, 4.3 Added State Machine Diagram V2 Reviewed others work and approved it.	1.3

1. Introduction

1.1 Purpose

The purpose of Comfort Studies (**CS**) is to provide a reservation system to allow students to reserve study rooms, and to request a tutor if available. There will also be a food ordering system for the emporium that will allow students to pick their food up at a designated time, and to order catering from the emporium to be delivered to their study room.

1.2 Document Conventions

This document contains each section in bold and a larger font for ease of navigation throughout. It also includes throughout certain statement, words, or acronyms that are in bold text. These are higher priority words that can either be found in the data dictionary or help direct attention of specific audiences.

1.3 Intended Audience and Reading Suggestions

This Software Requirements document is intended to be read by the developers, project manager, marketing staff, users, testers, and documentation writers.

Developers and Project Managers can review this project's capabilities and more easily understand what area of the design/code needs to be focused on improving or adding features to. This also provides an initial set of guidelines for future development.

Project testers can use this document as a base for their testing strategy to find bugs easier after looking at the requirements document. Testing becomes much more organized because there is a logical method created.

The application's end users can read this SRS document to learn about what this project can do.

<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.>

1.4 Product Scope

The purpose of this software is to provide a reservation system to allow students to reserve study rooms, and to request a tutor if available. There will also be a food ordering system for the emporium that will allow students to pick their food up at a designated time, and to order catering from the emporium to be delivered to their study room. One benefit of this product is to help students who can't find a room to study in find one without wasting time physically checking each study room. This product will also allow

students who might need help with a certain subject get help from a tutor if one is available. This product will also help students save time by allowing them to order their food online and picking it up instead of waiting in line to order their food. Another benefit to this product is catering that can be delivered to a study room for a group of people who need to eat so they do not have to go to the emporium to pick it up. The goal of this product is decrease the time it takes to find a study room as well as decrease the time it takes to get food for students.

<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.>

1.5 References

<https://www.w3schools.com/>

Professor Angela Guercio- in room 422 at the main hall at days and times Tue/Thur 9:45-10:45am, 2:50-3:20pm, 4:50-5:20pm, other times by appointment, and email aguercio@kent.edu

Capstone Syllabus- Authored by Angela Guercio

Capstone Syllabus can be found on this link:

<https://drive.google.com/drive/folders/1QRkQUB7KrFW-fpwbyOZ-Mvy-2dLv71a6>

<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

<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.>

The overall concept of the application will be a brand system. One part of the system, the tutoring schedule, will be an expansion of a the current tutoring scheduling system. There will be no actual connection to the current system because the developers do not have the permissions to alter it. The current system can be located at <https://tutortrac.kent.edu/TracWeb40/Default.html>.

There is also a study room booking service, specifically for the library, but it only uses a survey service to make the reservations. It can be located at https://kent.qualtrics.com/jfe/form/SV_3mj0Awxxw7NNT0eh?Q_JFE=qdg. The Comfort Studies application will be able to automatically added reservations into a database and will have the expanded feature of allowing food to be catered to the study room.

2.2 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
 - Schedule a Study Room
 - Order Catering to Study Room
 - Request a Tutor.
 - Order Food for Pickup from Emporium
- Admin
 - Update Tutors
 - Update Emporium Menu

2.3 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.>

There are three different users that will interact with the web application: students, faculty, and tutors. Each user has a different use for the system, but will have some will be the same.

The students can use the application to view/schedule study rooms, order food from the emporium, and request a tutor if needed. They will be able to search available rooms, request a room from the search. When a room is chosen the user will be able to request a tutor to come to the study room if a tutor is available. The user will be able to order food and pick it up from the emporium to save time. The students are the most important user for the system.

The faculty can use the web application to order food. They will be able to view the menu and select the food they wish to order and pick it up at the emporium.

Tutors can use the web application to view/schedule study rooms, and order food. They will be able to search for rooms and request a room if they needed one for tutoring. They also have access to order food. They will be able to view menu and select food to order. They will pick up the ordered food in the emporium.

2.4 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.>

Comfort Studies is a web application that should run on any OS with internet connection capabilities. Possible web browsers to access this application include: Google Chrome, Firefox, Safari, Internet Explorer.

2.5 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).>

The timing requirement for the CS system is the delivery date of May 2019. We have no memory requirements. The CS system will be in English. The system will run off the Kent State School server and can be accessed with any device with internet browsing capabilities. The CS system will adopt the Kent State security protocols. Car Ram Rod will be responsible for implementing the CS system, and the Kent State IT department will be maintaining the system.

2.6 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.>

A manual will be developed that will be included on the project's github repository. One of the web pages on the site will be dedicated to explaining how to work the application.

2.7 Assumptions and Dependencies

One assumption about this project is that it will always be used on a mobile phone that has the required amount of performance. If the phone does not have enough hardware resources, for example the user isn't in range of wireless internet, there may be scenarios where the application is unresponsive.

Another assumption is that when using this application on multiple devices like tablets and phones, it will work the same way. The application may just need to specifically adjust to each interface based on screen size of the device.

<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).>

3. External Interface Requirements

3.1 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.>

3.2 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.>

3.3 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.>

3.4 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.>

4. System Features

4.1 Schedule a Study Room

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

User can use this feature to choose a study room and reserve it for a specific time that is available.

This feature has a high priority.

4.1.2 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.>

1. The user must choose a date and time that they will be needing the study room
2. The user must then enter the amount of people who will be using the study room.
3. A list of study rooms will then be displayed that match the time and capacity if there is any.
4. User will choose a room and confirm it.
5. Additional options will then be displayed.

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: It will require a working database.

REQ-2:

4.2 Request a Tutor

4.2.1 Description and Priority

The user can pick a subject, the time of their study session, and the location of their study session. It will send an email to all available tutors in that subject and they can either accept or deny the request.

This feature has a medium priority.

4.2.2 Stimulus/Response Sequences

1. User selects Request a Tutor after reserving a study room.
2. User chooses the subject and clicks submit.
3. A response will tell you if there is a tutor available during the user's room reservation time.
 - a. If there is a tutor available, it will ask if the user would like to send the requests to them.
 - b. If there is not a tutor available, it will display that there is not a tutor available tutor.
4. Returns the user to the additional feature section of the Study Room Reservation feature.

4.2.3 Functional Requirements

REQ-1: The Schedule a Study Room feature must be completed

4.3 Order Catering to Study Room

4.3.1 Description and Priority

This feature allows the user to choose a minimum amount of food to be delivered to their study room. It will use a restricted catering menu.

This feature has a medium priority.

4.3.2 Stimulus/Response Sequences

1. User selects Order Catering after reserving a study room.
2. Catering Menu page is opened.
3. User can enter a quantity wanted for each item.
4. User also enters the preferred delivery time.
5. User then submits the order.
6. An order summary page is opened.
7. User can either confirm the order or not.
8. User is brought back to additional options page of the Study Room Reservation Feature.

4.3.3 Functional Requirements

REQ-1: The Schedule a Study Room feature must be completed

4.4 Order Food for Pickup from Emporium

4.3.1 Description and Priority

4.3.2 Stimulus/Response Sequences

4.3.3 Functional Requirements

REQ-1:

REQ-2:

5. Other Nonfunctional Requirements

5.1 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.>

Comfort Studies has no specific requirements on which system is to be used or needed. It is recommended to always use the most up to date operating system. The web application will perform best with the latest versions of Chrome, Internet Explorer/Edge, Firefox, Safari, and Opera. Some views and features may not be available without the latest versions of the corresponding web browsers.

5.2 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.>

The developer team will update Comfort Studies regularly to ensure users have no loss of data. All data will be stored in tables to provide data integrity and reduce data duplication. This will be achieved by separating the information into different tables and databases. The data is backed up in case of loss, damage, or breach to the server, so it can be seamlessly fixed without a significant downtime.

5.3 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.>

The web application uses object-oriented design to protect its data passed. All users must have a current and active Kent State University login to access the web application features. All users must login and have their credentials verified through this process in order to use any features of Comfort Studies.

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.>

Comfort Studies provides the users a simple graphic user interface to use. Due to its design it can be used by users with all types of experience. The user must have at least a basic knowledge of the internet before using it.

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.>

Comfort Studies requires a active Kent State login for the user to use the application. Depending on the user they will be limited on they can do. Refer to Section 2.3 for the roles that can be performed.

6. Other Requirements

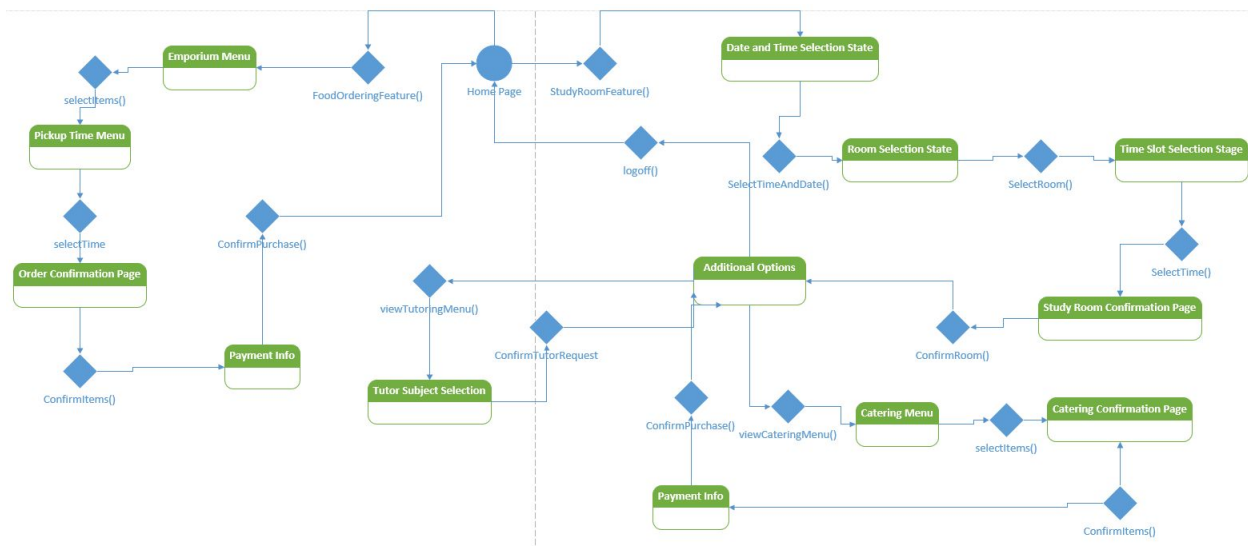
<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

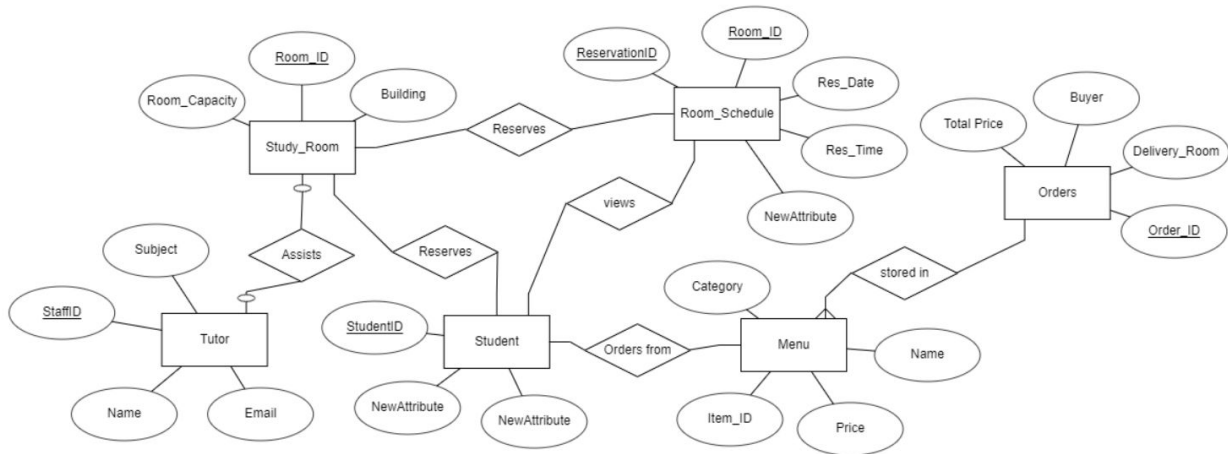
<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

Appendix B: Analysis Models

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



State Machine Diagram Version 2.



ER diagram Version 2.

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.>