Software Requirements Specification

for

Class Rank & Sort System (CRSS)

Version 1.0 approved

Prepared by

Team 3 (Hailin Archer - deak0007, Bryan Baker - bake1358, Colin Kluegel - klue0037, Josh Spitzer-Resnick - spitz123)

CSCI 5801 - Software Engineering I, Spring 2020

University of Minnesota

Sunday, February 9, 2020

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
Archer, Baker, Kluegel, Spitzer-Resnick	2-9-2020	Initial Software Requirement Spec Document	1

Archer	2-5-2020	Sections 3.1, 3.2, 3.3, and 4.1	0.20
Spitzer-Resnick	2-5-2020	Sections 1.5, 2.5, 5.1, 5.4, and 5.5	0.19
Kluegel	2-4-2020	Sections 2.4 and 2.6	0.18
Spitzer-Resnick	2-4-2020	UC_8, UC_9, UC_10	0.172
Baker	2-3-2020	UC_4, UC_5	0.171
Baker	2-1-2020	Section 5.2	0.17
Kluegel	2-2-2020	UC_6, UC_7	0.161
Archer	2-2-2020	Section 2.2, UC_2, UC_3	0.16
Baker	2-2-2020	Sections 1.4, 2.3, 2.7, and 3.4	0.15
Archer	1-31-2020	Sections 1.1, 1.2, 1.3	0.1
Archer, Baker, Kluegel, Spitzer-Resnick	1-29-2020	UC_1	0.01

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the *Class Rank & Sort System (CRSS)*. It will explain the purpose and features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate. This document is intended for users of the software and also potential developers.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

The intended audience includes:

- Typical users, such as campers, who want to use CRSS to rank the classes they want to take, and instructors who teach the classes and determine the number of students for each class that they offer.
- Advanced/professional users, such as administrators and their assistant(s), who will run the software to place campers in the classes based on their rankings, and input the classes' information into the system.
- Programmers who are interested in working on the project by further developing it or fix existing bugs.

1.4 Product Scope

The CRSS is intended to be a software system to manage class offerings and class enrollments at Camp Voyager through the duration of the four summer sessions offered by the camp.

This system will assist with:

- course preference ranking for campers
- sorting campers into available classes
- scheduling the classes throughout each day of camp
- counselors monitoring the attendance of each class

This system will only help manage the items above. Items out of scope for this project include other camp duties like cooking, cleaning, discipline, billing, etc.

<Q - Do we need to include more or be more specific on what is out of scope for this system? Do we even need to state this?>
<Q - Document makes no mention on if instructors input class information. Does this mean that is out of scope for this design?>

1.5 References

"Homework 1 – Software Requirements Specification (SRS) Document", Watters, Jan 2020:
https://canvas.umn.edu/courses/158173/assignments/1000025?module_item_id=3541516

"HW1 - Use Cases", Archer, Baker, Kluegel, and Spitzer-Resnick, Feb 9, 2020:
https://docs.google.com/document/d/1vWCnIe-jo5KQnTxjM6sk3XusN3jAGVnW05PCVk3yOmk/edit

"IEEE Software Requirements Specification Document Template", Wiegers, 1999:
https://canvas.umn.edu/courses/158173/pages/software-requirements-specification-supporting-documents?module_item_id=3541620

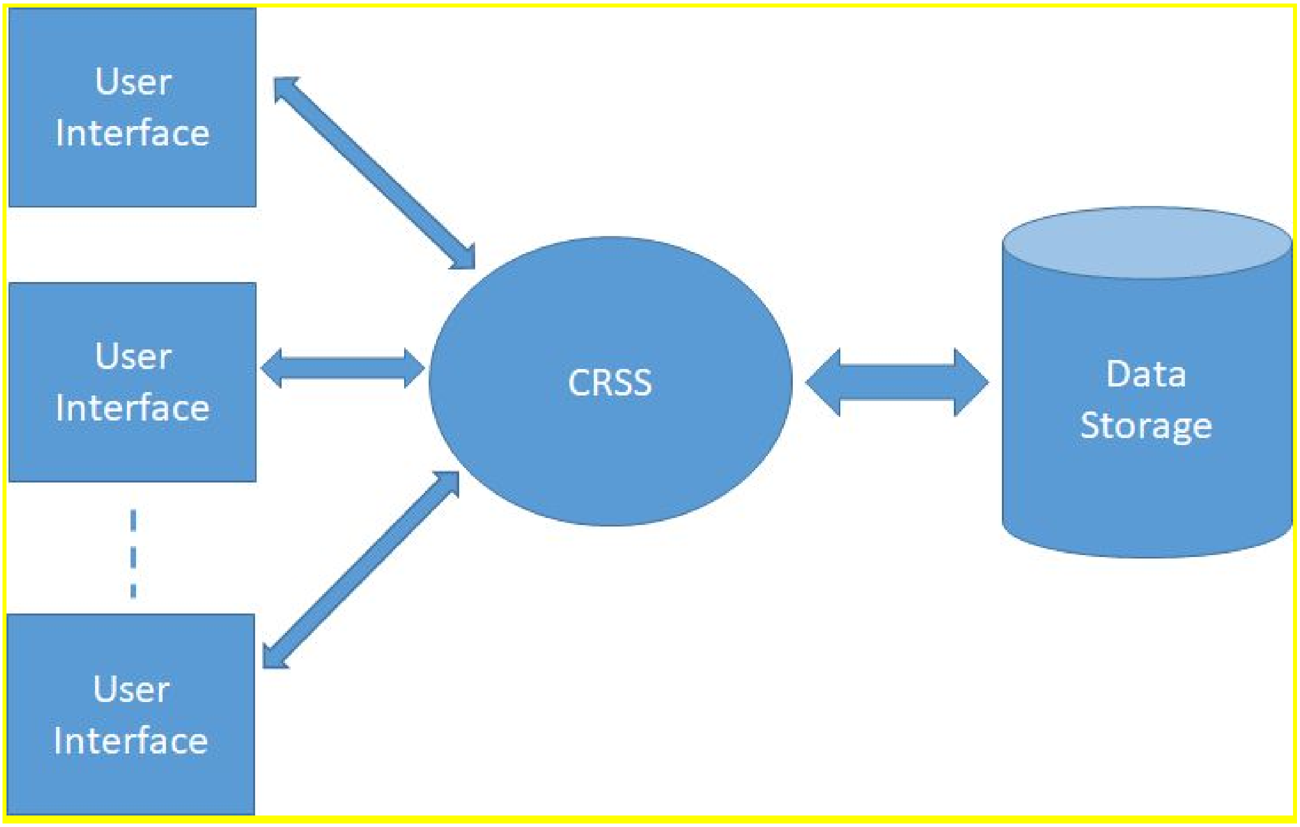
<Q - Are there UI style guides we should follow?>
<Q - Are there standards we need to adhere to re: privacy or otherwise?>

2. Overall Description

2.1 Product Perspective

Camp Voyager is requesting the CRSS to assist various stakeholders when scheduling its classes for summer camps. The system is being requested in order to give campers the ability to sign up for the classes they would like to take by giving them the ability to rank the classes they would like to take in order of preference. The system shall give instructors the ability to specify which classes they will be teaching and allow administrators the ability to oversee and make changes to classes enrollment. The system shall consist of a server/database that stores the class data and a client application that users will interact with to rank classes and create schedules.

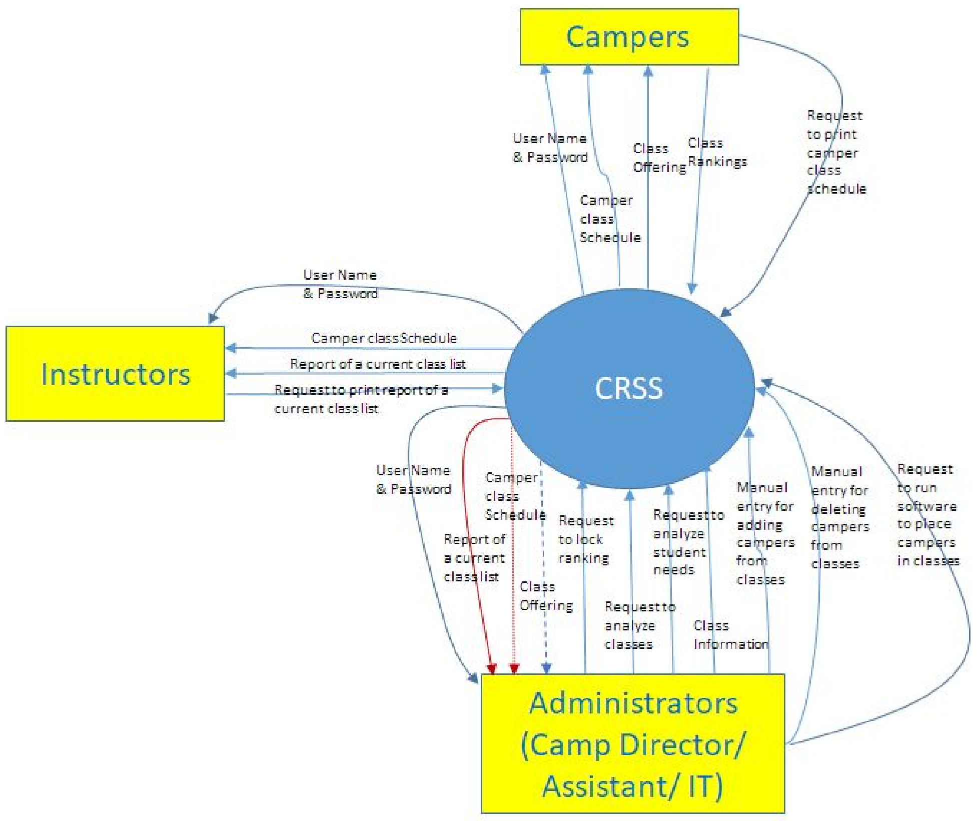
<Q - How are user accounts added to the system for logins to be generated by the system?>
<Q - What kind of data storage does the system use?>



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.>

The Context Diagram


Campers
User Name & Password
Camper class Schedule
Class Offering
Class Rankings
Request to print camper class schedule
Instructors
User Name & Password
Camper class Schedule
Report of a current class list
Request to print report of a current class list
CRSS
User Name & Password
Report of a current class list
Camper class Schedule
Class Offering
Request to lock ranking
Request to analyze classes
Request to analyze student needs
Class Information
Manual entry for adding campers from classes
Manual entry for deleting campers from classes
Request to run software to place campers in classes
Administrators (Camp Director/ Assistant/ IT)

Question:
Who provides user information to the system? (Not specified in given documents)

- Classes:
  - New: create a class object
  - Delete: delete a class object
  - Attributes: class_name, class_id, section number, instructor(s), max_students, time slots, number of students enrolled
  - Change cap: change maximum number of students allowed
  - Add instructor: add an instructor to the class
  - Remove instructor: remove an instructor to the class
  - Set section number
  - Set time slots
  - Increment student count: add a student to number of students enrolled
  - Decrement student count
- Users:
  - New: create a new user
  - Delete: delete a user from the system
  - Attributes: name, id, group, account number, password(???)
- Groups:
  - New: create a new group
  - Delete: delete a new group
  - Attributes: group_name, group_id, access_level

- Tables:
  - Class_table
  - User_table
  - Group_table
  - Section_table
  - Session_table
  - Schedule_table
  - Enrollment_table
- User interfaces:
  - Administrator shall be able to add/remove class
  - Administrator shall be able to add/edit class info
  - Instructor shall be able to view and print online report of a current class list
  - Camper shall be able to enrollment/class ranking
  - Administrator shall be able to lock ranking
  - Administrator shall be able to run software to place campers in classes (complete enrollment)
  - Administrator shall be able to manual edit (enroll/remove) camper enrollment
  - Camper shall be able to view and print class schedule
  - Administrator shall be able to run class enrollment analysis
  - Administrator shall be able to run student needs analysis (empty blocks/duplicates/etc.)
  - Instructor shall be able to view a camper's schedule

<Q - We're creating a menu for a main login screen - are they the same for every user or you only see the functions you have access to?>
<Q - What kind of data structure for storage - object database or relational database?>

2.3 User Classes and Characteristics

There are three primary classes of users for CRSS:

Camp Director / Administrator

- **Definition:** The Camp Director is in charge of all aspects of the camp. This person shall oversee both campers and instructors and manage the other day to day activities of the camp.
- **Functionality Required:** The camp director shall work with the class instructors to determine class logistics (caps, sections, etc). Administrator shall enter the course information into CRSS and lock down the camper ranking process. Administrators shall run the software tool to assign the campers to their classes and will manually update information if required. They shall review reports from the system to help plan class offerings, enrollment, and other instructor and student needs.
- **Security Required:** Due to the range of functionality the Camp Director shall require the security for this user(s) will need to be higher than that for the other users. This user(s) is in charge of managing the most aspects of this system so instructors and campers should not be able to access the functionality for manual updates to both course enrollment and class session adjustments. <Q - How many director users should we anticipate - should there be one with ultimate control?>

Instructor

- **Definition:** The class instructors are the personnel responsible for teaching the campers at Camp Voyager. These users will typically have several classes to instruct <Q - Will one instructor instruct only one subject, but multiple sessions? Or can instructors teach multiple courses for any number of sessions?> and will work closely with the administrator to provide information about classes. <Q - Are there different levels of instructors? eg: senior staff, instructor in training, etc.>
- **Functionality Required:** Instructors require access to the schedules for the sessions that they teach, camper class lists for each session that they instruct and the ability to take attendance. <Q - How do instructors send the class information to the director? Through the system or off line?> <Q - How do instructors ask for changes to schedule or student changes etc from the director? Through the system or off line?>
- **Security Required:** Instructors have the ability to manually enroll campers into classes. They can also view a campers schedule. <Q - Will instructors have access to all camper information or just for the campers in the courses they are teaching?>

Camper

- **Definition:** The campers are the attendees of Camp Voyager. They will be attending classes during camp taught by the instructors.
- **Functionality Required:** Prior to camp, campers shall select from several summer camp sessions and, within their session, they create a list of preferred classes from the courses offered during that camp session. Campers shall have access to their class schedules and instructor names. <Q - Will campers be able to see a list of other campers in their classes?> <Q - How do campers request changes to their schedule - through the system or off line?>
- **Security Required:** Campers should only be allowed access to their own information.

2.4 Operating Environment

The CRSS application will be capable of running on the following operating systems: (for hardware requirements, see section 3.2)

<Q - Should it run on specific operating systems or the most common operating systems?>

The CRSS consists of a client side application for the campers, instructors and administrators to interact with and a server application that stores the class schedules and rosters. It will work with the following browsers:

<Q - Does this have specific requirements for which web browsers are compatible?>

2.5 Design and Implementation Constraints

The CRSS shall be written in C/C++.

<Q - What databases should be used - SQL?>
<Q - What language(s) should be used for visualization?>

<Q - Should more than one user be able to log in at a time?>
<Q - Who will be responsible for maintaining the software?>
<Q - Should we adhere to a specific coding style guide or documentation framework?>

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.>

<Q - Do we have to deliver documentation to them? Do they expect to know how it works intuitively?>
<Q - How do they want us to deliver how-tos? Online user guide? PDF? Printed? Online tutorials? CDs? Different video file formats?>
<Q - What user documentation standards are there that are required?>

2.7 Assumptions and Dependencies

This system shall be developed using the CSE computer labs at the University of Minnesota.

<Q - How to store data - binary file, etc?>
<Q - How to access accounts?>
<Q - Where will the final system run and who has access to that location?>
<Q - Questions in section 2.4 relevant here>

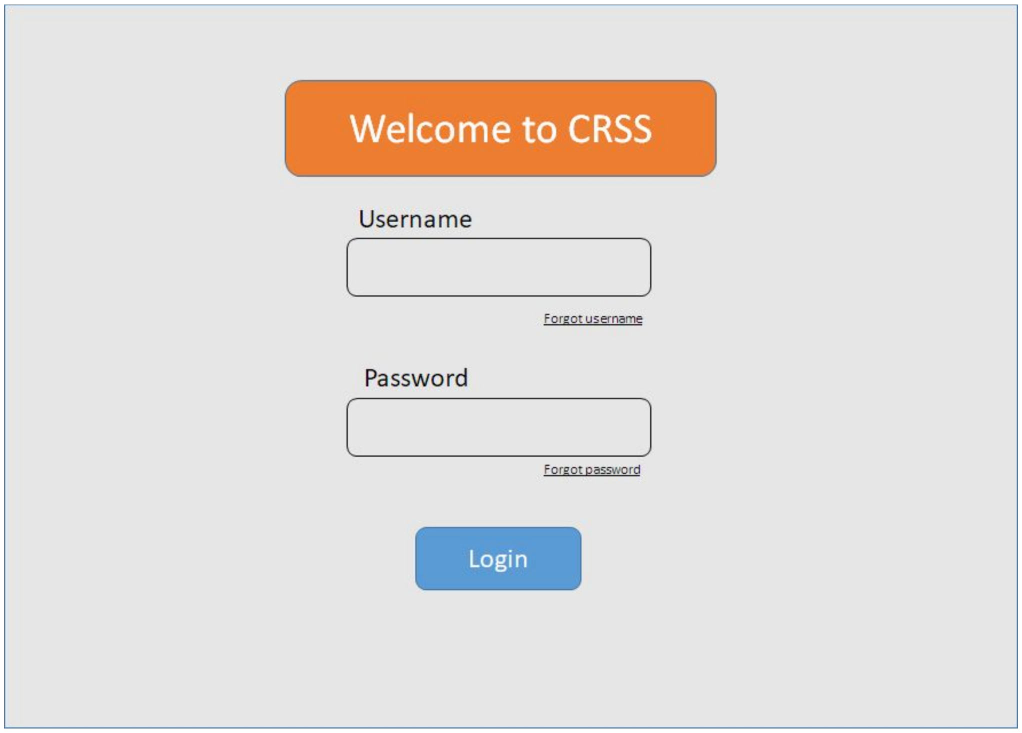
3. External Interface Requirements

3.1 User Interfaces

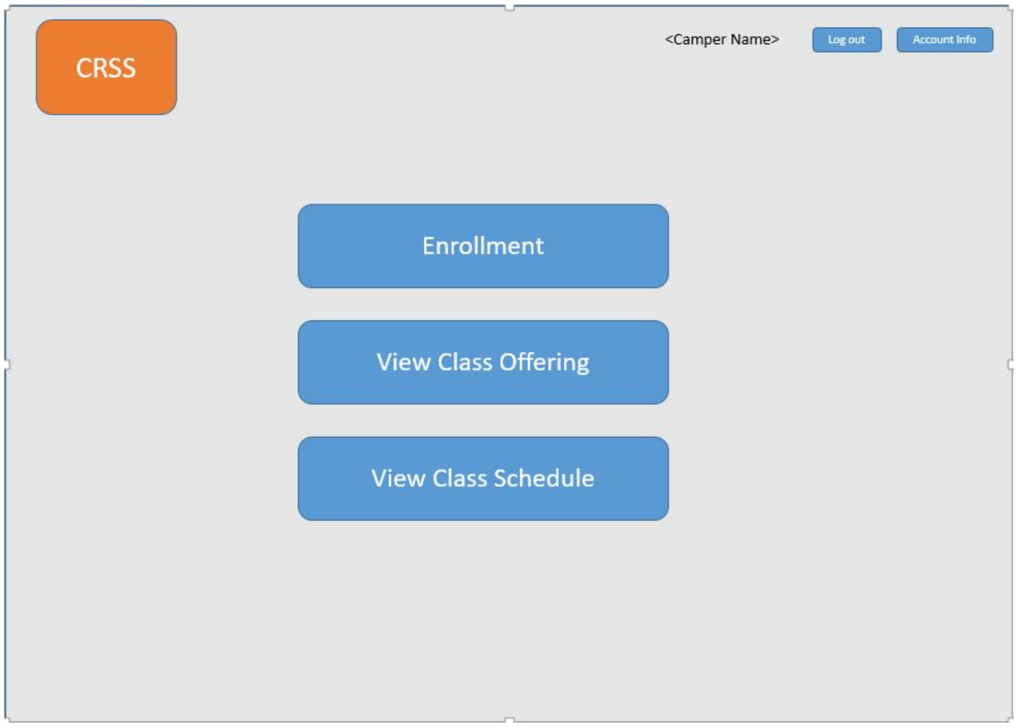
<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear 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.>

The user interfaces shall contain following windows:

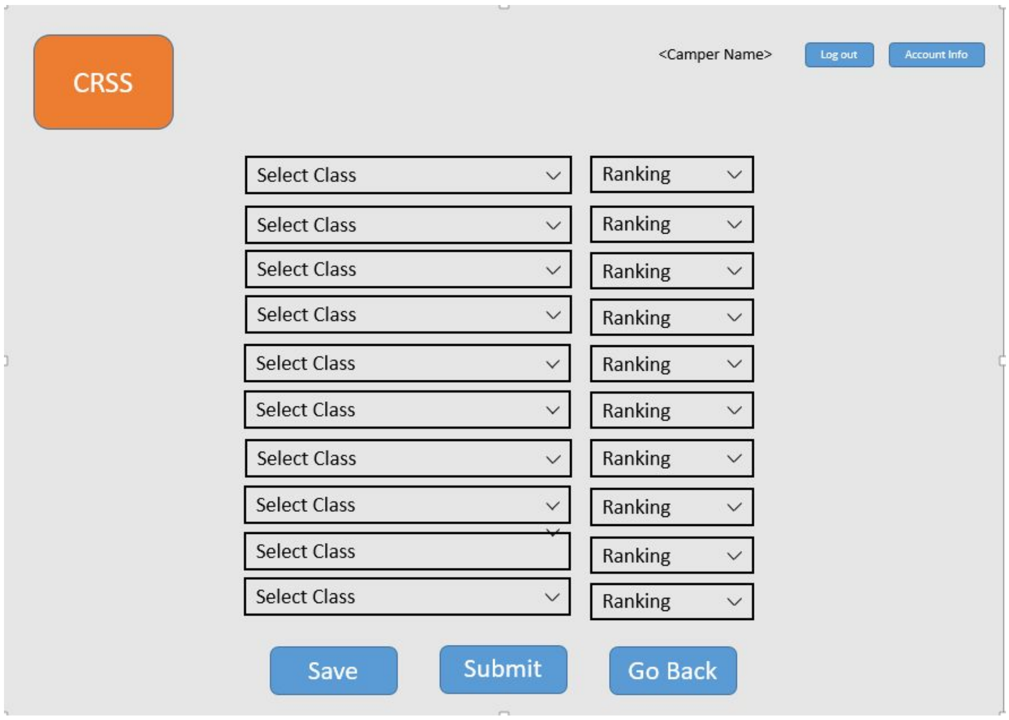
1. Log in screen



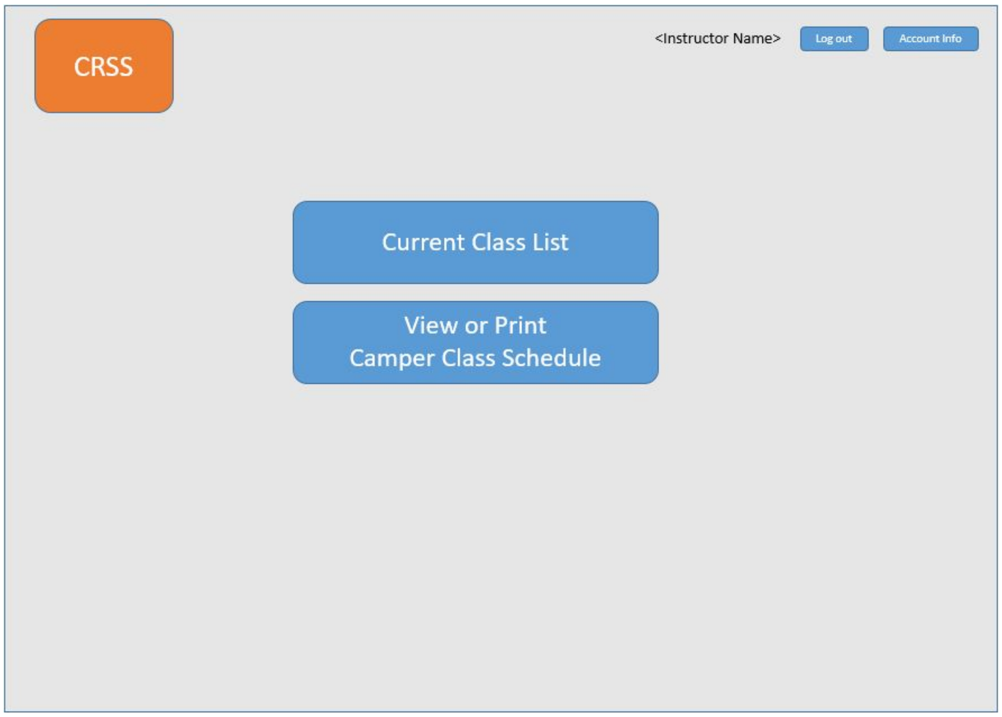
2. Camper home screen



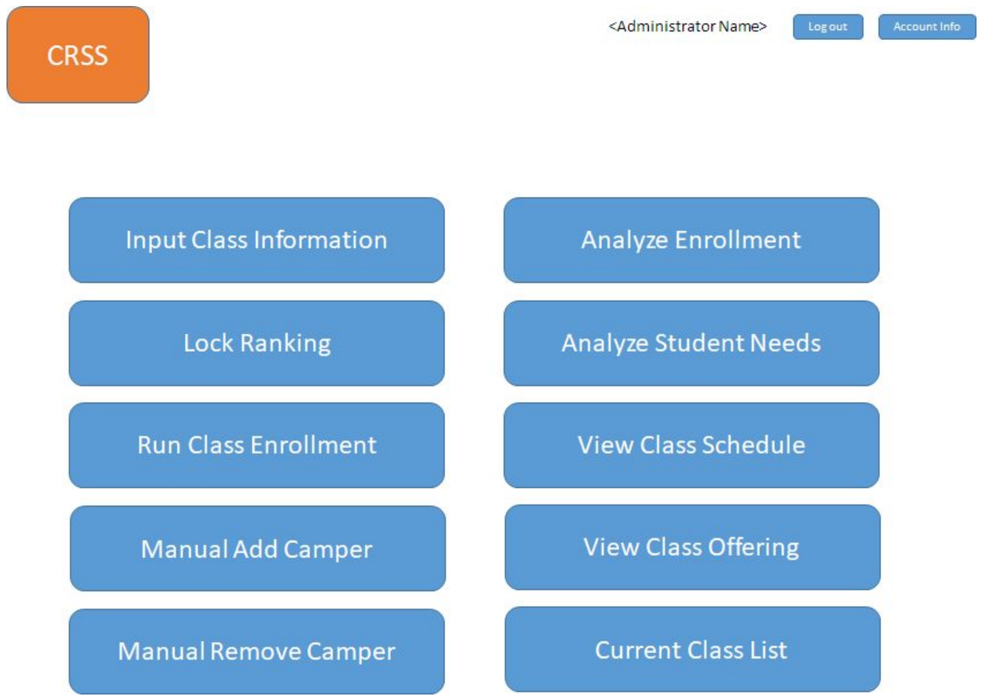
3. Class ranking screen



4. Instructor home screen



5. Administrators home screen



6. Camper schedule screen
7. Add/edit class screen
8. Manual enrollment screen
9. Enrollment statistics screen

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.>

Minimum hardware requirements for client systems:

- ___ Gigahertz CPU
- ___GB of memory
- ___GB hard drive

Minimum requirements for server:

- ___ Gigahertz CPU
- ___GB of memory
- ___TB hard drive

A system with those hardware specifications can handle network speed of <Q: What is the internet service CRSS has to work with?>

<Q - What are the minimum requirements for client systems and server?>
<Q - Are we expected to account for smart phone use?>

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.>

CRSS shall interface with the data storage device where all camp data (camper data, class information, logins, etc) is stored.

CRSS shall provide user interface. This user interface shall be <Q - browser or desktop application or mobile app, combination, or all?>

CRSS shall interface with database software <Q - Oracle, Hadoop, no-sql, etc>

<Q - What version of C++ are we using?>
<Q - Are we using third party libraries or tools that are open source or not?>
<Q - Are we using an Oracle database, Hadoop, no-sql?>

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.>

<Q - Do campers / instructors request changes to their schedule from an administrator via the system? Via sending email messages to other users? Forms within the system?>
<Q - What type of encryption standards are required?>
<Q - Does the system send logins over email? Is there a password recovery function that uses some kind of verification?>
<Q - Can you login with other accounts, such as Google or Facebook?>
<Q - Are there any applicable laws regarding secure communication for data protection in the case of this software? Are campers sorted by their age level group so might need birthdays to determine which classes they are eligible for?>

4. 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.>

4.1 Enroll campers in classes based on ranking, timing of submission, and class availability

<Don't really say "System Feature 1." State the feature name in just a few words.>

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).>

The system shall place campers in classes based on factors listed in 4.1.3.

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.>

- Enrollment period ends
- Camp director or his/her assistance click on complete enrollment button from their home screen

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: Campers shall get preference on class enrollment based on the timing of their submission of rankings. Campers who register first get first dibs on the ranked classes.

REQ-2: Campers shall not be enrolled in the same class twice.

REQ-3: Campers shall not be enrolled in more than one class in any given block of time. (Blocks are set by the director and are not changeable by anyone else in the camp. The blocks are 2 in the mornings and 1 in the afternoon.)

4.2 Manually remove camper from class

4.2.1 Description and Priority

Once the schedules have been created and the class rosters have been set there needs to be a way for system administrators to manually a camper from a class without having to re-run the class scheduling algorithm. This is a high priority feature as it is necessary to keep the class rosters accurate and up to date. <Question - would priority level really be dependent on how often students are reassigned? If 1 camper per session is getting reassigned that is easy to keep track of offline, if 10 are reassigned - not so much >

4.2.2 Stimulus/Response Sequence

1) Class schedules have been created
2) Admin logs in
3) Admin selects class that student is in
4) Admin selects student and clicks remove button

5. Other Nonfunctional Requirement

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.>

The CRSS requires the minimum hardware requirements specified in (2.4), internet access, and the ability to run <Q - What browsers are needed?>. Performance depends on the quantity of user campers and classes and as a result, the system requirements for larger camp sessions are more demanding.

<Q - How fast does the sorting system need to run?>
<Q - Are there requirements for real-time updating in terms of data multiple users can access or view?>

5.2 Safety Requirements

This project will abide by any and all state or federal regulations established for <Q - What safety requirements are we required to abide by?>

There are no specific OSHA regulations specific to this matter. Additional OSHA information on computer workstations can be found here:
https://www.osha.gov/SLTC/computerworkstation/index.html

Information on office ergonomics can be found from the Mayo Clinic here:
https://www.mayoclinic.org/healthy-lifestyle/adult-health/in-depth/office-ergonomics/art-20046169

<Q - Do we need to include general "don't look at screens for too long" or "get up and stretch" reminders for campers?>

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 CRSS will have three levels of users, each with a different set of rights. The three levels of users are camp director level, instructor level and camper level. Anyone who is not a member of one of these three groups should not be allowed to access the system. The members of each group should be maintained by system administrators.

Class enrollment information should only be able to be viewed by those who have the proper authorization. All users will have a username and password assigned to them to access the system.

<Q - What is each user allowed to view? Can campers view who else is enrolled in different classes?>
<Q - See communications interface questions from section 3.4>
<Q - Is privacy a concern? Do we need to satisfy any privacy or security laws?

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.>

The CRSS provides the users with an interface appropriate for their level. Due to its easy to use interface, it can be used by users with typical computer skills for simple ranking and submission tasks, as well as by slightly more advanced users for management tasks.

<Q - How adaptable/portable does this need to be to different systems?
<Q - Who maintains this system and how?>
<Q - Do the features need to be simple and intuitive or can they reference a long user guide or tutorial?>

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.>

Campers may:

- rank up to 10 classes from those provided for given session
- view, print personal schedule

Instructors may:

- propose classes
- view, print class enrollment(s) responsible for
- view all camper schedules

Administrators may:

- add camper and instructor accounts
- approve or reject class proposals
- edit class proposals or class details
- run the class rank sorting system
- analyze class enrollment levels
- analyze student schedule deficiencies
- edit class assignments
- view, print all class enrollments
- view, print all camper schedules

eg duration of a camp session is 5 days 2 morning sess, 1 afternoon, more than 1 instructor in class etc - Josh TODO

<Q - What other operating principles/requirements are there to the product? Performance requirements?>
<Q - Are there security requirements?>

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.>

<Q - How will the data be stored?>
<Q - Upon completion of the project who owns the source code? Can we re-use this source code for another camp or project?>
<Q - Do we need to get a patent? Who's going to own this code?>

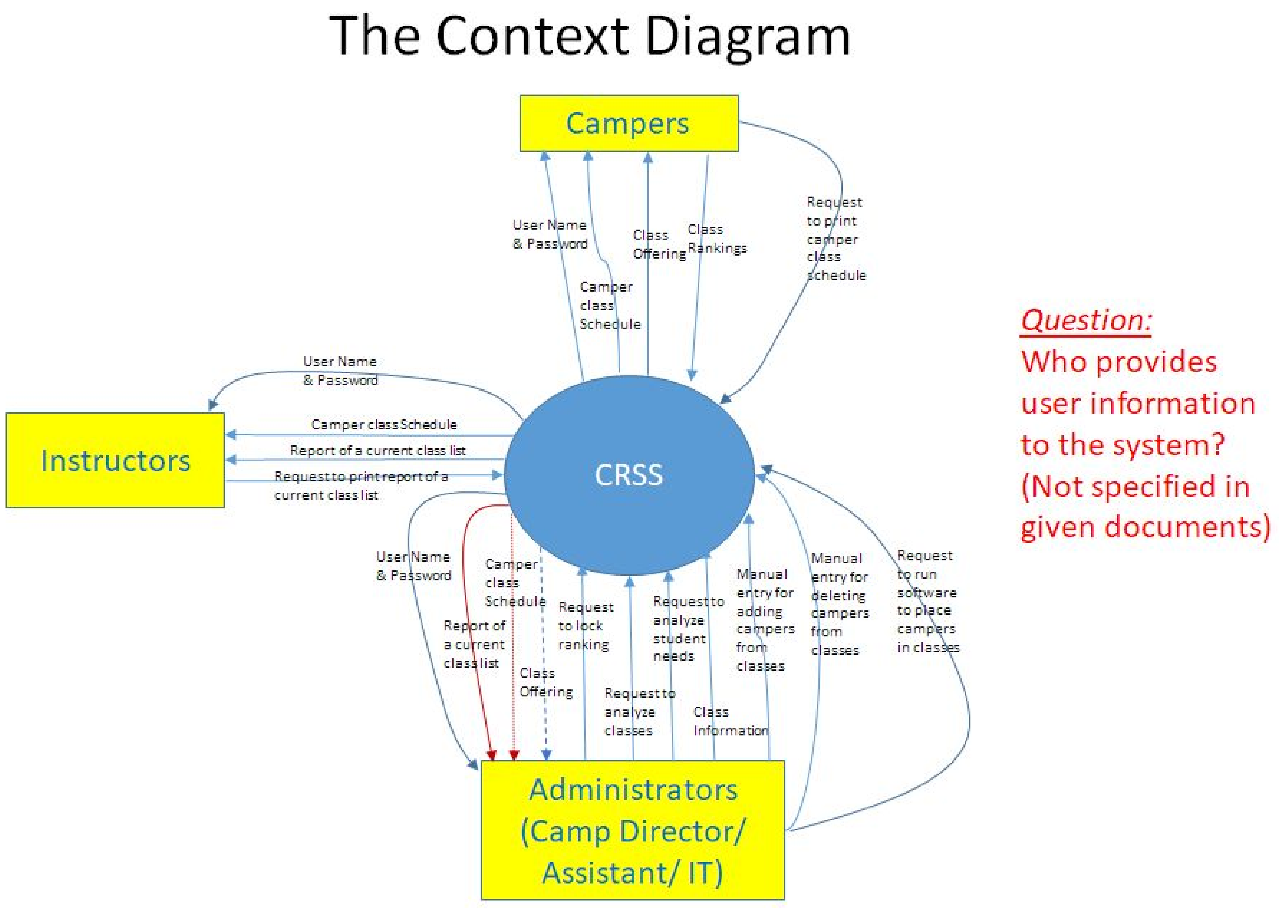
Appendix A: Glossary

<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.>

CRSS: Class Rank & Sort System

Appendix B: Analysis Models

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


The Context Diagram
Campers
User Name & Password
Camper class Schedule
Class Offering
Class Rankings
Request to print camper class schedule
Instructors
User Name & Password
Camper class Schedule
Report of a current class list
Request to print report of a current class list
CRSS
User Name & Password
Camper class Schedule
Report of a current class list
Class Offering
Request to lock ranking
Request to analyze classes
Request to analyze student needs
Class Information
Manual entry for adding campers from classes
Manual entry for deleting campers from classes
Request to run software to place campers in classes
Administrators (Camp Director/ Assistant/ IT)
Question:
Who provides user information to the system? (Not specified in given documents)

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.>

document all questions - pg#s
put the SECTIONS that are completely/mostly TBD?