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

Revision History

Name	Date	Reason For Changes	Version
Archer, Baker, Kluegel, Spitzer-Resnick	2.9.2020	Initial Software Requirement Spec Document	1.0
Kluegel	2.9.2020	Appendix C	0.31
Archer, Baker, Kluegel, Spitzer-Resnick	2.8.2020	Secondary revision	0.3
Archer	2.8.2020	Appendix B	0.25
Kluegel	2.8.2020	UC_11	0.24
Spitzer-Resnick	2.8.2020	UC_13	0.23
Archer	2.8.2020	UC_14	0.22
Baker	2.8.2020	UC_12	0.21
Archer, Baker, Kluegel, Spitzer-Resnick	2.7.2020	Preliminary revision	0.2
Archer	2.5.2020	Sections 3.1, 3.2, 3.3, and 4.1	0.11
Spitzer-Resnick	2.5.2020	Sections 1.5, 2.5, 5.1, 5.4, and 5.5	0.10
Kluegel	2.4.2020	Sections 2.4 and 2.6	0.09
Spitzer-Resnick	2.4.2020	UC_8, UC_9, UC_10	0.08
Baker	2.3.2020	UC_4, UC_5	0.07
Kluegel	2.2.2020	UC_6, UC_7	0.06
Archer	2.2.2020	Section 2.2, UC_2, UC_3	0.05
Baker	2.2.2020	Sections 1.4, 2.3, 2.7, and 3.4	0.04
Baker	2.1.2020	Section 5.2	0.03
Archer	1.31.2020	Sections 1.1, 1.2, 1.3	0.02
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 fixing 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 shall 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 shall 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?>
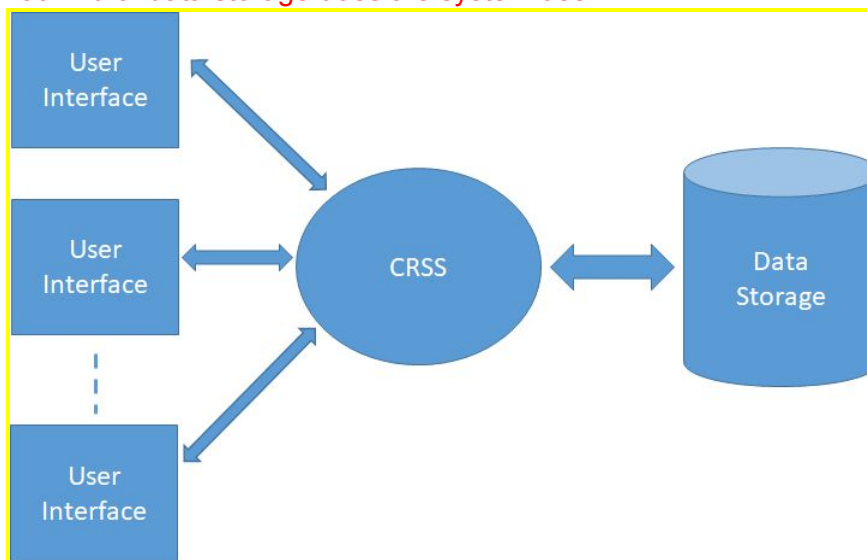
2.2 Product Functions
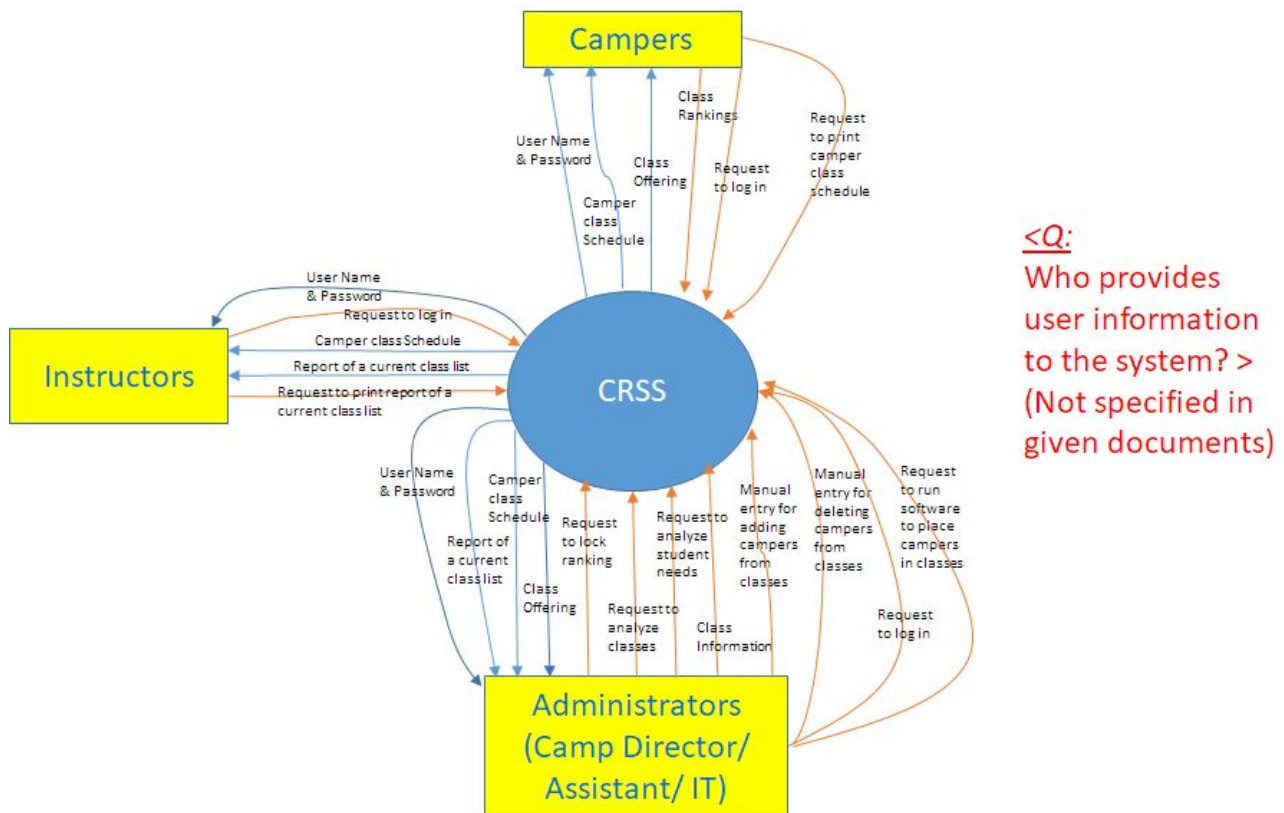
- CRSS shall perform following functions:
  - CRSS shall generate username and password for eligible users
  - Users shall be given one of the three access levels defined in section 2.3
  - All registered users shall be able to log in to CRSS
  - Administrators shall be able to add/remove class
  - Administrators shall be able to add/edit class information
  - CRSS shall ensure classes do not violate rules defined in section 5.5
  - Instructors shall be able to view and print online report of a current class list
  - Campers shall be able to rank classes for enrollment
  - CRSS shall ensure campers' class ranking following rules defined in section 5.5
  - Administrators shall be able to lock ranking
  - Administrators shall be able to run software to place campers in classes (complete enrollment)
  - Administrators shall be able to manually edit (enroll/remove) camper enrollment
  - Campers shall be able to view and print class schedule
  - Administrators shall be able to run class enrollment analysis
  - Administrators shall be able to run student needs analysis (empty blocks/duplicates/etc)
  - Instructors shall be able to view a camper's schedule
- 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

<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.>

A CRSS user manual shall be delivered with the CRSS system. The user manual shall be in the format of <Q - What format does the customer prefer?>.

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?>
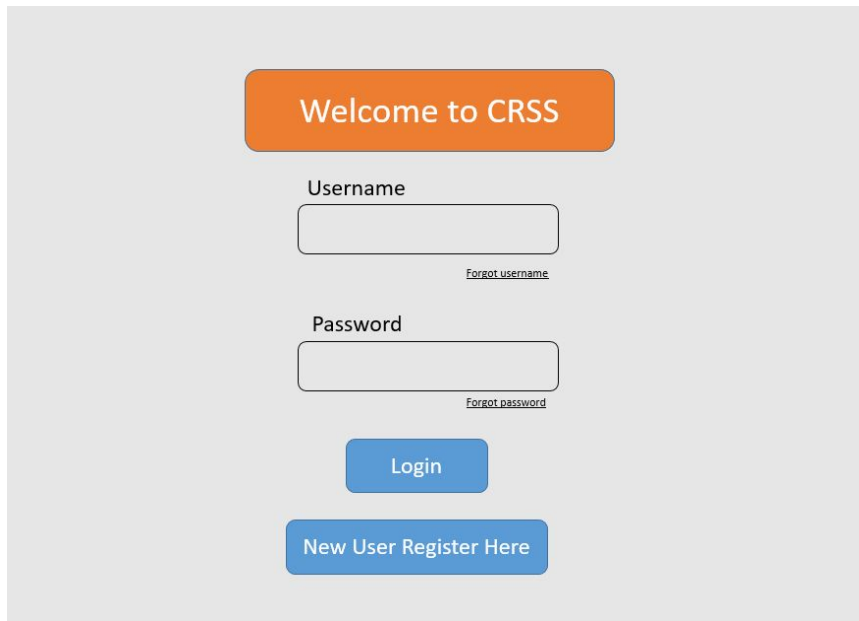<Q - Questions in section 2.4 relevant here>

3.   External Interface Requirements
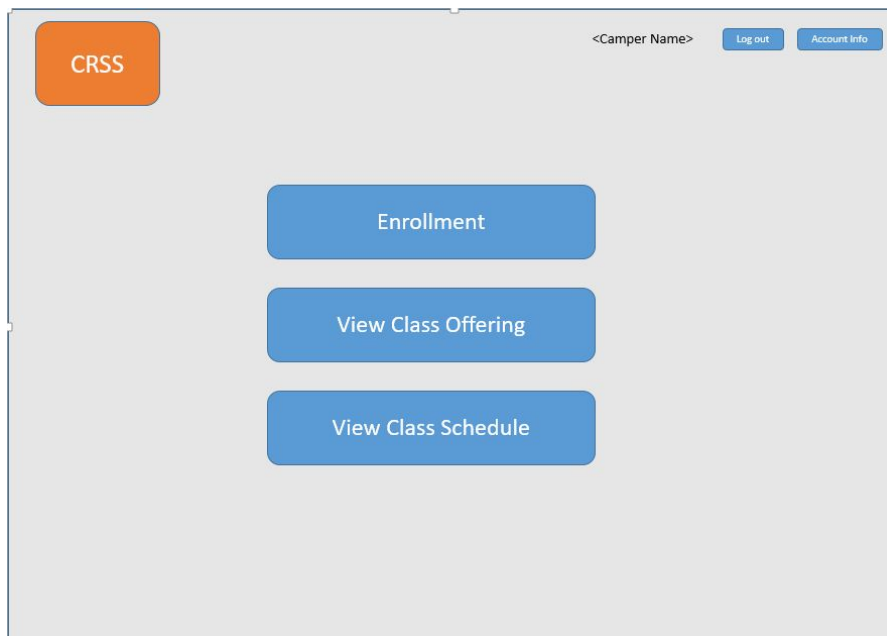
3.1   User Interfaces

CRSS shall provide following user interfaces:
1. Log in screen: This is the first screen when a user starts the system. It shall have following components:
    a. Username field: Allow a user to put in his/her username.
    b. Password field: Allow a user to put in his/her password.
    c. Login button: Start username & password authentication process
    d. New user registration button: Bring up new user registration screen. Allow a new user to register for an account in CRSS.
    e. Other functions: allow user to retrieve username and/or password when needed (provide entrances to forget username and forget password screen)

2. User home screens: 3 different screens shall be designed for 3 levels of users defined in section 2.3
    a. Camper home screen: Appears after a camper successfully logged in. It shall have following components:
        i. Enrollment button: Clicking this button shall bring up class enrollment/ranking screen
        ii. View class offering button: Clicking this button shall bring up class offering screen
        iii. View class schedule button: Clicking this button shall bring up class schedule screen
        iv. Other fields: display logged in camper name, logout button, account info button
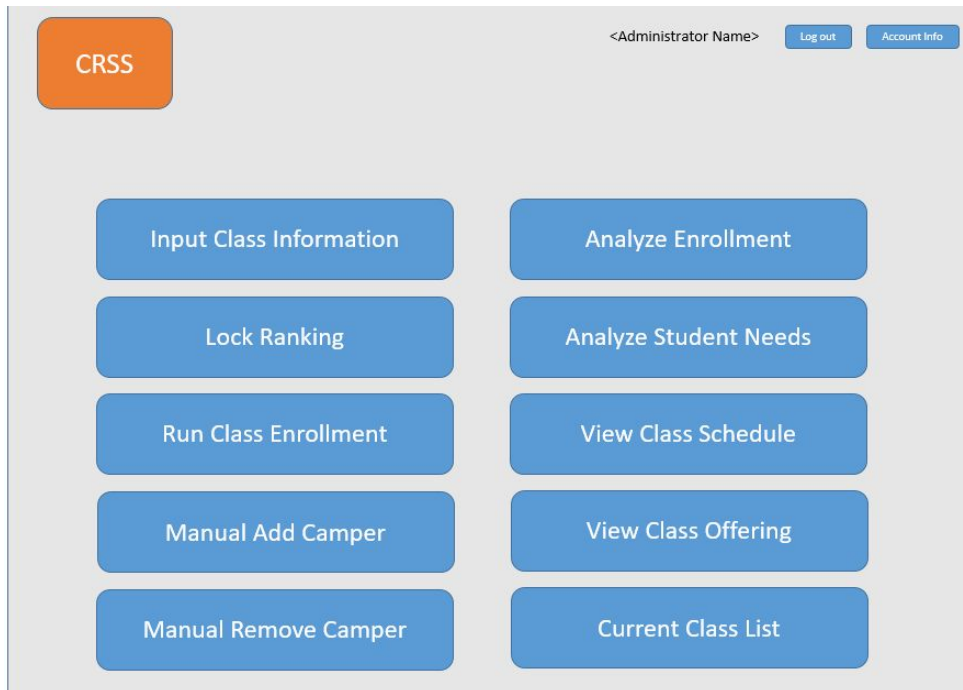
b. Instructor home screen: Appears after an instructor successfully logged in. It shall have following components:
    i. View class list button: Clicking this button shall bring up current class listing. Instructor shall be able to view or print this class listing.
    ii. View or print camper class schedule button: Clicking this button shall prompt instructor to choose a camper then go to class schedule screen
    iii. Other fields: display logged in instructor name, logout button, account info button
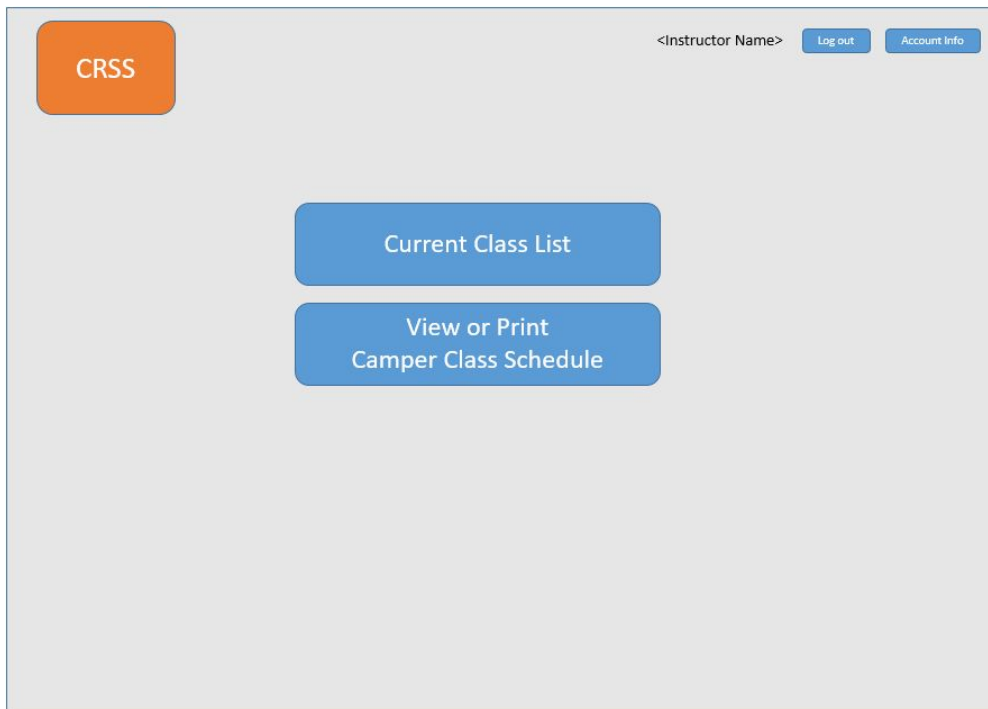
c. Administrator home screen: Appears after an administrator successfully logged in. It shall have following components:
    i. Input class information button: Clicking this button shall bring up 'Input Class Information' screen
    ii. Lock ranking button: Clicking this button shall lock ranking in the CRSS for the current enrolling session
    iii. Run class enrollment button: Clicking this button shall make CRSS place campers in classes based on rules specified in section 5.5
    iv. Manual add camper button: Clicking this button shall bring up screen to allow the administrator to manually add a camper to a class
    v. Manual remove camper button: Clicking this button shall bring up screen to allow the administrator to manually remove a camper to a class
    vi. Analyze enrollment button: Clicking this button shall show enrollment statistics of current session
    vii. Analyze student needs button: Clicking this button shall show enrollment statistics of current session
    viii. View or print camper class schedule button: Clicking this button shall prompt instructor to choose a camper then go to class schedule screen
    ix. View class offering button: Clicking this button shall display class offering information

      x.   View class list button: Clicking this button shall bring up current class listing. Instructor shall be able to view or print this class listing.

     xi.   Other fields: display logged in administrator name, logout button, account info button

CRSS

<Administrator Name>   Log out   Account Info

Input Class Information	Analyze Enrollment
Lock Ranking	Analyze Student Needs
Run Class Enrollment	View Class Schedule
Manual Add Camper	View Class Offering
Manual Remove Camper	Current Class List

3. New user registration screen: This screen appears when a user clicks a new user registration button from CRSS login screen. This screen shall contain fields for a new user to input all required information. This screen shall also contain a button for users to submit their information for registration. The submit button shall prompt CRSS to generate a unique username and password for the new user if that user is not in the system.

4. Camper enrollment (class ranking) screen: This screen appears when a camper clicks the enrollment button from his/her home screen. It shall contain class selection fields and corresponding ranking fields to allow the logged in camper to rank classes based on rules listed in section 5.5

5. View class offering screen (camper, instructor, administrator): This screen appears when a logged in user click view class offering button from his/her home screen. Class offering screen shall allow user to view or print classes offered in a session.

6. View class schedule screen (camper, instructor, administrator): This screen appears when a logged in user click view class schedule from their home screen. This screen shall show the time block (i.e. morning or afternoon), the class enrolled in that block, the location of the class, any special instructions for the class, and the instructor(s) names. The system shall allow schedules to be printed.

7. View current class list (instructor, administrator): This screen appears when an authorized users clicks view class list from their home screen. This screen shall display the current class list.

8. Administrator input class information screen: This screen appears when a logged in administrator click input class information button from his/her home screen. This screen shall contain fields for the administrator to put all required class information in the system. This screen shall also contain a submit button for the administrator to save the class information to the system. The system shall check for conflicts between classes according to rules in section 5.5

9. Administrator manually add camper screen: This screen appears when a logged in administrator click on manually add camper button from his/her home screen. This screen shall contain fields for the administrator to select a camper and select a class. This screen shall provide an add button for the administrator to add the camper to the class.

10. Administrator manually remove camper screen: This screen appears when a logged in administrator click on manually remove camper button from his/her home screen. This screen shall contain fields for the administrator to select a camper and select a class from the camper's registered class list. This screen shall provide an remove button for the administrator to remove the camper from the class. <Q - Do we need a screen to move a camper from one class to another, or do we want to keep this a two step process?>

3.2   Hardware Interfaces

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

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

TBD (need more information to determine communication interfaces).
<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

For a complete list of features, please refer to the HW1_Team3_Use Cases document. 4.1 and 4.2 provide two sample features.

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

4.1.1   Description and Priority

<Q - Which use cases are high. medium, low priority to user?>

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

4.1.2   Stimulus/Response Sequences

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

4.1.3   Functional Requirements

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.

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

    4.2.3   Functional Requirements

        REQ-1:   Administrators have the ability to remove a camper from a class

5.   Other Nonfunctional Requirement

5.1   Performance Requirements

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

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

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

Class ranking rules:
- 7) Campers will not always get the class that they want
- 12) Campers must rank at least 10 different classes titles (not sections), and can rank up to 10 classes
- 13) No two classes can have the same ranking
- 14) No class has more than one ranking
- 17) There will only be one final ranking per session

Class assignment rules:
- 8) There will always be enough classes and seats in all of the classes on any given day to put the students in a class for each day
- 9) The software will only be run once when the camp registration closes for a given camp session
- 10) Any class movements after the software has been run will be completed manually
- 15) Late camp attendees will be assigned classes manually based on availability, without requiring a full rerun of the assignment system
- 16) Attendees get preference on class enrollment based on the timing of their submission of rankings. People who register first get first dibs on the ranked classes.
- 19) Attendees will not be enrolled in the same class twice
- 20) Attendees will not be enrolled in more than one class in any given block of time
- 21) Some classes have different enrollment caps
- 28) A camper can be reassigned a class at any time by an administrator

Class scheduling constraints:
- 1) Campers will attend 3 classes per day (2 hours classes), with 2 blocks in the morning and 1 in the afternoon
- 2) The duration of a camp session is 5 days, Monday - Friday
- 3) There will be at least 4 camp sessions throughout the summer
- 4) Instructors can teach the same class multiple times during a given camp session, so there should be sections of the classes spread through the session with different times that can be on the same or different days
- 5) The same class material can be taught by different instructors
- 6) Only one section of a class is offered in one time slot

- 23) There can be more than one instructor for a class
- 24) A class can be offered multiple times during a session by different instructors
- 26) An instructor cannot teach more than one class in a single block

System requirements:
- 11) Have three levels of users (camp director, instructor, and camper) each with different authorizations
- 18) Each term of the camp will have a unique identifier (denoted as Session 1, Session 2, Session 3, and Session 4)
- 22) Camper schedules will show the time block (morning or afternoon), the class enrolled in that block, the location of the class, any special instructions for the class, and the instructor(s) names
- 25) Each class has its own unique identifier
- 27) Some class titles will be included in more than one camp session
- 29) All users will have a username and password assigned to them
- 30) Only qualified users should be able to access the system
- 31) The username and passwords will be generated by the system and given to the users

The above requirements are taken directly from and numbered in order of appearance in the project description.[1]

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
- set enrollment caps by working with an administrator

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
- lock rankings
- set blocks
- set enrollment caps

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

---

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

6.   Other Requirements

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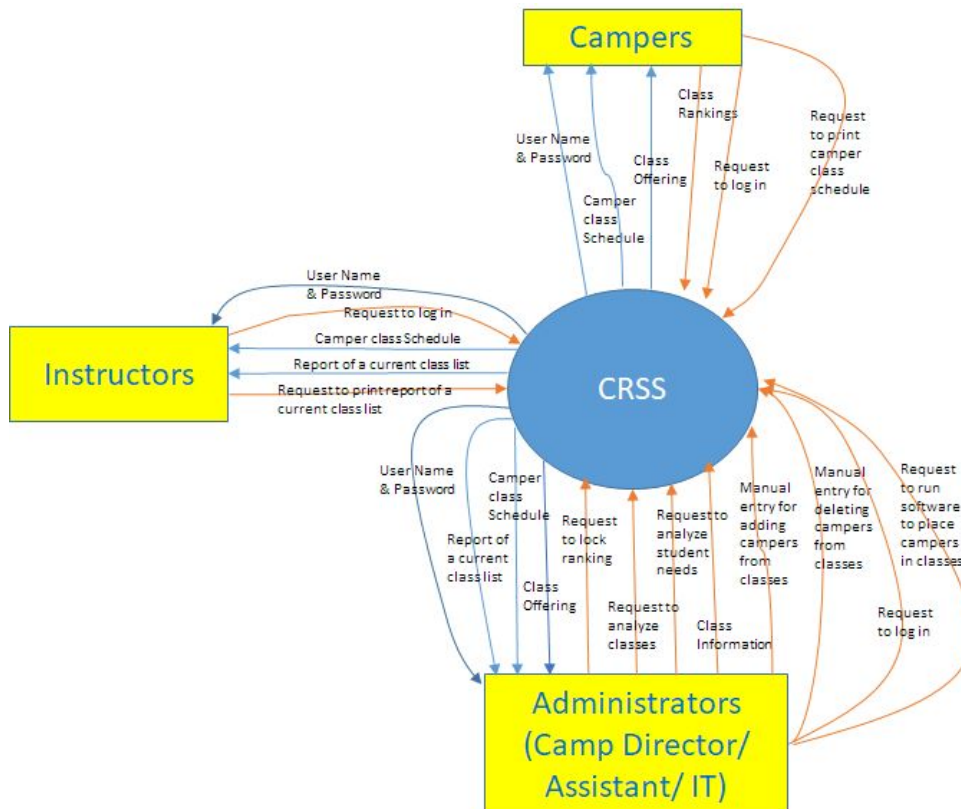<Q - Upon completion of the project who owns the source code? Can we re-use this source code for another camp or project?>
<Q - Do we need to get a patent? Who's going to own this code?>

Appendix A: Glossary

CRSS: Class Rank & Sort System

Appendix B: Analysis Models

Appendix C: To Be Determined List

1.4 Product Scope
<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 of if instructors input class information. Does this mean that is out of scope for this design?>

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

2.1 Product Perspective
<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
<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
<Q - How many director users should we anticipate - should there be one with ultimate control?>
<Q - Will one instructor instruct only one subject, but multiple sessions? Or can instructors teach multiple courses for any number of sessions?>
<Q - Are there different levels of instructors? eg: senior staff, instructor in training, etc.>
<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?>
<Q - Will instructors have access to all camper information or just for the campers in the courses they are teaching?>
<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?>

2.4 Operating Environment
<Q - Should it run on specific operating systems or the most common operating systems?>
<Q - Does this have specific requirements for which web browsers are compatible?>

2.5 Design and Implementation Constraints
<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
<Q - What format does the customer prefer?>

2.7 Assumptions and Dependencies
<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>

3.1 User Interfaces
<Q - Do we need a screen to move a camper from one class to another, or do we want to keep this a two step process?>

3.2 Hardware Interfaces
<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
<Q - What is the user interface? Browser or desktop application or mobile app, combination, or all?>
<Q - What is the database? 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
<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?>

4.1 System Features
<Q - Which use cases are high. medium, low priority to user?>

5.1 Performance Requirements
<Q - What browsers are needed?>
<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
<Q - What safety requirements are we required to abide by?>
<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
<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
<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
<Q - What other operating principles/requirements are there to the product? Performance requirements?>
<Q - Are there security requirements?>

6.0 Other Requirements
<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?>

Use Case Questions

UC_1: Rank classes
<Q - Can campers change ranking before lock down?>
<Q - Does the system allow camper to re-rank classes with new priority>
<Q - Are the schedules already created when the campers are ranking their classes?>

UC_3: Analyze class enrollment
<Q - Need more information on when classes in next session are scheduled>

UC_4: Create class offering schedule
<Q - Will classes need to be set up for each summer session independently or will all session schedules be set up at once?>

UC_5: Enroll campers in classes
<Q - Will this be run once per summer session?>
<Q - How will administrators know? Set up deadline for rank placements? Random otherwise?>

UC_7: User logs in
<Q - Will we be required to implement a password recovery feature that doesn't require contacting the administrator? >

UC_8: Adjust class offerings and class info
<Q - How do instructors communicate class proposals to administrators?>

UC_11: Print camper class schedule
<Q - Do administrators or instructors also need the ability to print schedules?>

UC_12: Lock camper rankings
<Q - Does the administrator tell the system to automatically stop accepting responses by a deadline or manually decide responses are no longer accepted?>

UC_13: Print report of current class list
<Q - Should the print option be offered from the main menu or only from the view class enrollment screen? UC_10>

UC_14: System generates usernames and passwords
<Q- Should instructors and administrators register in different ways?>
<Q - Do administrators manually create user accounts for registered campers or staff?>
<Q - Should the system provide a temporary password that users can change?>
<Q - Should all users be enrolled in the camp in some way or are other accounts possible like parent accounts, etc?>