Software Requirements Specification

for

Class Rank & Sort System (CRSS)

Version 0.010

Prepared by:

Sami Frank
Jake Waro
Allison Miller
Declan Buhrsmith

University of Minnesota – CSci 5801

February 3, 2020

Table of Contents

Revision History

Name	**Date**	**Reason For Changes**	**Version**
Jake	2.3.20	Initialize document	0.001
Jake	2.4.20	Product Perspective	0.002
Sami	2.4.20	External Interfaces (sec 3)	0.003
Jake	2.5.20	Complete Section 2	0.004
Jake, Sami, Declan	2.6.20	Merge Document	0.005
Sami	2.8.20	Questions for sec 3. add to appendix	0.006
Jake	2.8.20	Create consistent format, review document for logical and grammatical errors	0.007
Allison	2.8.20	Add use cases, update introduction	0.008
Jake, Allison, Declan, Sami	2.9.20	Final review, last comments on content review	0.009
Jake	2.9.20	Final formatting, ready for submission	0.01

Introduction

1.1 Purpose

This document specifies the purpose and requirements of the Class Rank & Sort System (CRSS). It describes the entire system, including system features, requirements, constraints, and interfaces with existing technology, and is intended for both users and developers of the software.

1.2 Document Conventions

The IEEE Software Requirements Specification template was used to create this document. Outstanding questions regarding unclear requirements are italicized throughout the document and prefixed with a Q and a number noting the section and order of the question.

1.3 Intended Audience and Reading Suggestions

This document is intended for anyone who might need to interact with the CRSS, including users, developers, and project managers. The document contains details on the requirements and constraints of the system as well as features, interface details, and usage information. Begin reading with section 2 - Overall Description - for an overview of the project, and continue with section 4 to learn about features of the software. Section 3 describes interface requirements and sections 5 and 6 describe non-functional and other assorted requirements. Sections 2 and 4 are most pertinent to users, while the entire document should be of interest to other readers.

1.4 Product Scope

The CRSS is a system to be used by Camp Voyager to handle registration, organization, and documentation needs for camp sessions. The software provides a way for campers to register for the camp and rank the classes they would like to attend. As the software is responsible for all camper registration, it is also capable of handling attendance, sorting, and scheduling on behalf of the instructors and camp administrators. Containing all such administrative functionality in one place helps the camp run smoothly, allowing administrators and instructors to focus more of their attention and energy on creating a better experience for the campers.

1.5 References

IEEE Software Requirements Specification Template: https://ieeexplore.ieee.org/document/278253

Overall Description

2.1 Product Perspective

This product (CRSS) is a new, self-contained product. It is being used as a stand-alone system for a summer camp to build campers' schedules. This product is not part of a larger system.

Outstanding questions:

Q 2.1.1 - Will the system need to integrate with any existing software / hardware systems the camp already employs?

2.2 Product Functions

All Actors:

- Login: user logs into the system using a username and password given to them that was generated by the system.

Campers:

- Select class: campers can select from the list of available courses that they want to rank.
- Rank: for the selected class, camper assigns a rank to the course.
- View camper schedule: displays the campers class schedule.
- Print camper schedule: prints a copy of the campers schedule.

Instructors:

- Input classes: instructors are able to input the classes they are going to teach for each summer session.
- View instructor schedule: display a schedule of the class that the instructor teaches.
- Print instructor schedule: prints the specific instructor's schedule.
- View camper schedule: displays the campers class schedule.
- Print camper schedule: prints a copy of the campers schedule.

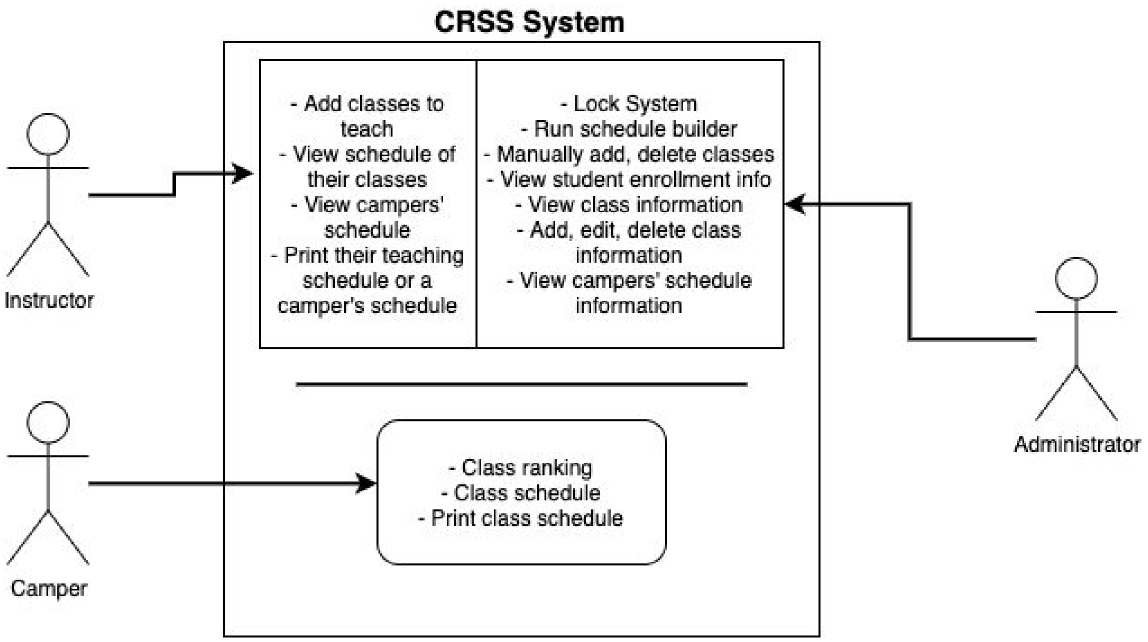
Administrators:

- Input class information: Administrators can input the camp session class information into the system. This information consists of:
    - Class titles and section numbers
    - Class times
    - Number of seats per class
    - Class information
- Save class information: system saves the information that was input for the class information.
- Assign section numbers: assigns a unique section number to each class that the instructors have added to the system
- Build schedules: takes the campers' rankings and creates schedules that incorporates both the ranking preferences and campers' relative times of registration

- Lock System: Locks the system for the purposes of assigning section numbers and building class schedules. The system is locked once per camp session and will not be unlocked afterwards.

System

- Determine user: the system verifies the credentials from the login screen. If invalid, returns a message describing the credentials as invalid. If valid, allows the user to pass to the respective next screen.


CRSS System
- Add classes to teach
- View schedule of their classes
- View campers' schedule
- Print their teaching schedule or a camper's schedule
- Lock System
- Run schedule builder
- Manually add, delete classes
- View student enrollment info
- View class information
- Add, edit, delete class information
- View campers' schedule information
- Class ranking
- Class schedule
- Print class schedule
Instructor
Camper
Administrator

Outstanding questions:

Q 2.2.1 - should the system save campers' rankings after every ranking? Or, will the camper only be able to submit their rankings once, and upon submission, the rankings will be saved?

Q 2.2.2 - should the class information input menu contain functionality to input class titles and section numbers, class times, number of seats, and other information all on one screen? Or, should this functionality be split into different menus?

Q 2.2.3 - how should the assignment of section numbers be triggered? Is this something that gets explicitly triggered by the director? Or, will this process be encapsulated within the build schedules functionality?

2.3 User Classes and Characteristics

The three main actors of this product will be **campers, instructors,** and **administrators**. The users are expected to be competent in basic single-page application navigation; that is, they are capable of logging in to a system, and navigate at the first level of the application by using different interactive tools.

- **Camper** – a camper will be a kid in their youth, likely between the ages of 6 – 17. They will be enrolled in the camp. This may or may not be their first experience at Camp Voyager. Prior to the camp session, campers will make use of the class ranking functionality. During the camp session, campers will make use of the schedule viewing functionality.

- **Instructor** – instructors are expected to be adults 18 years of age or older. They will be teaching summer classes such as swimming or climbing. The nature of their work will likely not involve much interaction with technology. Prior to the camp session, instructors will make use of the functionality to input classes they will teach. During the camp session, they will make use of the functionality to view their class schedules as well as campers' schedules.

- **Administrator / Director** – administrators will be camp employees either in higher positions such as the camp director or system administrator. Their use of the system will be more of a management focus, and they will likely use the system more than instructors. Administrator users will have privileges to all of the functionality listed in section 2.2. Prior to camp, administrators will make use of the functionality to input class information, build schedules, and manually add, edit, or delete class schedule information. During the camp session, administrators will make use of the functionality to view instructors' and campers' schedules as well as the functionality to manually add, edit, or delete class schedule information.

2.4 Operating Environment

Using the newest JDK and SDK released by Oracle, Java SE 12, the following platforms and operating systems will have the capabilities to utilize the product:

- Windows:
  - Windows Server 2019, (64-bit)
  - Windows Server 2016, (64-bit)
  - Windows Server 2012 R2, (64-bit)
  - Windows Server 2012, (64-bit)
  - Windows 10, (64-bit)
  - Windows 8.x, (64-bit)
  - Windows 7 version SP1, (64-bit)

- Linux:
  - Oracle Linux 7.x, (64-bit)
  - Oracle Linux 6.x, (64-bit)
  - Red Hat Enterprise Linux 7.x, (64-bit)
  - Red Hat Enterprise Linux 6.x, (64-bit)
  - Unbuntu Linux 18.10, (64-bit)
  - Unbuntu Linux 18.04 LTS, (64-bit)

- macOS
  - 10.12+ excluding version 10.15, x64

- Java virtual machines (JVMs):
  - Must be 64-bit

Outstanding questions:

Q 2.4.1 - Are there any specific requirements for a programming language to use for this product?

2.5 Design and Implementation Constraints

There are no foreseeable corporate, regulatory, or hardware limitations of notice at this time.

Outstanding questions:

Q 2.5.1 - Will CRSS need to interface with any other applications?

Q 2.5.2 -How will the data be stored? Should the system use cloud technologies to store data, or have a local database?

Q 2.5.3 - Are there any language requirements? Are there any other forms of communication accessibility to be considered?

Q 2.5.4 - How should the product handle user security to sign into the system? If using an API to handle security, will the functionality be placed on the API or within the product's code to verify a user?

Q 2.5.5 - Who will be responsible for maintaining the software after it is delivered?

2.6 User Documentation

Before building a plan on the schedule of delivering CRSS, we will need to understand all deliverables in order to build a timeline.

Outstanding questions:

Q 2.6.1 - Are there any manual / reference materials requested? These materials would require time apart from building the program to prepare. Knowing if any tutorials or additional documentation is needed will help determine product timelines.

2.7 Assumptions and Dependencies

Assumptions

1. This document has been prepared based on an initial statement from the client about the product requested. We have not had a chance to communicate with the client, so we have listed questions in locations throughout the document that require additional information as to not make certain assertions. Appendix B can be referenced to locate sections with outstanding questions.
2. We are building an entirely new system; however, we have made the assumption that we will make use of APIs to aid in our efforts. Types of APIs we may use would consist of tools such as graphics libraries, or a specific database technology.
3. CRSS will use Java SE 12. This is the most recent Java SE release that includes both a published JDK and SDK. However, there are other versions of Java as well as other languages that could be used to build CRSS.

4. CRSS is built on the assumption that the camp and all actors have internet access.

Dependencies

1. With the assumption stated that we will be using APIs as part of our product, there may be specific dependencies for using those tools, such as needing to use a specific version, or simply noting that the module needs to be installed. We do not have explicit dependencies to list for APIs, but rather, the assumption that we will have dependencies according to APIs the product would use.

External Interface Requirements

3.1 User Interfaces

While our team has not yet conducted elicitation interviews with Camp Voyager to discuss the user interfaces of the Class Rank & Sort System (CRSS), there are some basic functionalities that can be discussed.

There will be security features such as a login screen to ensure each user has access to the correct user interface (I.e. a camper doesn't see action items that only an administrator can execute).

Some standard buttons and functions for all users will include logging out, saving and canceling certain actions.

There will be detailed error messages to constrain a user from not following guidelines set in Camp Voyagers requirement specification.

- Error messages while ranking:
    - Message that constrains a camper from only ranking 10 classes.
    - Message notifying a camper that they can't rank the same class title.

- Error message while logging in:
    - Message notifying the user that credentials failed.

- General responsive messages:
    - Message confirming that action has been saved and/or updated.
    - Confirming logout of system.

* This is not a comprehensive list as our team has not yet completed elicitation interviews.

3.2 Hardware Interfaces

The main hardware interface that will be taken into consideration is the connection between the CRSS software and a printer.

- The CRSS software will print a schedule when a user executes a print action.

Outstanding questions:

Q 3.2.1 - Are there any requirement specifications for the printer?

3.3 Software Interfaces

The CRSS software will need a database of all users and their authorization levels. This information will allow the CRSS software to handle which user interface to show a user upon login.

It has not been clarified how this information will be transferred to the CRSS, whether it be an integration between an existing Camp Voyager software, a manual import of needed data, or something else. It has not been clarified how user information will be transferred to the CRSS.

Outstanding questions:

Q 3.3.1 - How will the CRSS obtain pertinent user information in order to create a basic profile for each user, with access to everything in their authorization level?

Q 3.3.2 - Will the CRSS have to integrate with an existing Camp Voyager system that obtains rostering for who's enrolled/staffed at the camp?

Q 3.3.3 - If the CRSS is not integrating with an existing system, how will / should the CRSS obtain this pertinent user information?

3.4 Communications Interfaces

Clarification is required in regards to how a user's username and password is relayed to them. Specifically, our team needs to know if the CRSS will integrate with existing Camp Voyager software to handle communication or if the CRSS will need more user information like an email address or something else to handle the communication of login details.

<u>If the CRSS is integrating with existing Camp Voyager software:</u>

Our team will need more information on what functionalities the existing software has for third parties and where in the existing software the CRSS should route this information.

<u>If the CRSS is handling the communication of login information to the user:</u>

There will be sensitive information communicated, and our team will need more information from Camp Voyager on the level of security the CRSS should ensure.

Our team will also need to know how to format the message, and if there should be more information besides the username and password.

Outstanding questions:

Q 3.4.1 - If the CRSS is handling login information, do we need security features with those credentials?

Q 3.4.2 - Should it be required that the user changes their password upon first login to the system?

Q 3.4.3 - Should there be a time limit from when communication of credentials has been made to the first use of those credentials?

4. System Features

4.1 Add Classes

Name	Add Classes
ID	inst_001
Description	Instructors can add the classes they will be teaching, to the system
Actors	Instructors
Organizational Benefits	Allows for instructors to easily communicate the classes they intend to teach for a given camp session.
Frequency of Use	Once per session
Triggers	Instructor logs into the system.
Preconditions	System has to exist. The classes they will teach have to be valid options for the summer camp. Instructor is logged into the system. System is unlocked.
Postconditions	The system has the class options that will be available for campers to rank.
Main Course	1. Instructor is on landing page where they have options to add new classes 2. Instructors selects to add a new class 3. Display shows fields to populate: 1. Class name 2. Number of students 4. System saves the new class 5. System redirects instructor to be able to add additional classes

Alternate Courses	AC1 Instructor logs out 1. Instructor returns to login portal (See use case "Login to System") AC2 Instructor inputs a class they have already added 1. System displays an error stating the class already exists 2. Returns user to Main Course step 1.
Exceptions	EX1 System fails to save newly added class 1. System notifies instructor error has occurred in saving the newly added class. 2. Return to Main Course step 1.

4.2 Assign section numbers

Name	Assign Section Numbers
ID	admin_001
Description	The system assigns section numbers to classes that have been added to the system by instructors.
Actors	Administrators
Organizational Benefits	Allows for clarity in the schedule by uniquely identifying classes, because multiple instructors may be teaching the same class, just at different times.
Frequency of Use	Once per session.
Triggers	Administrator selects to assign section numbers.
Preconditions	System has to exist. Administrator is logged into the system. The system must be locked.
Postconditions	Each class offered is now uniquely identified.

Main Course	1. Administrator starts on landing page where they have options to lock the system. 2. Administrator selects to lock the system. 3. The administrator then selects to assign section numbers. 4. System determines section numbers and saves these to the system. 5. The system displays a list of classes and their respective class information.
Alternate Courses	AC1 Administrator logs out 1. Administrator returns to login portal (See use case "Login to System"). AC2 System is already locked 1. Return to Main Course step 3. AC3 Section numbers have already been assigned and need to be reassigned 1. Return to Administrator's class editing page (See use case "Manually add, cancel, update schedules").
Exceptions	EX2 System fails to save section numbers 1. System notifies administrator error has occurred in saving the section numbers. 2. Return to Main Course step 3. EX3 System fails to lock 1. System notifies administrator error has occurred in locking the system. 2. Return to Main Course step 2.

4.3 Lock camper rankings

Name	Lock Camper Rankings
ID	admin_002
Description	This action will disable the ability for campers to rank classes, and instructors to add classes.

Actors	Administrators
Organizational Benefits	Will allow Camp Voyager to have a static set of rankings to be able to execute scheduling software.
Frequency of Use	Once per session, before the session begins.
Triggers	Administrator presses a lock button to execute action.
Preconditions	System has to exist. .Administrator is logged in. Administrator must allow enough time for campers to rank so there's data to use.
Postconditions	Campers can no longer rank classes, instructors can no longer add classes..
Main Course	1. Administrator is on the landing page where they have the option to lock the system. 2. select option to lock the systems. 3. A confirmation popup appears to confirm the system is locked.
Alternate Courses	AC1 System is already locked 1. Lock option will be unclickable.
Exceptions	EX1 System fails to lock rankings 1. System notifies administrator error has occurred in locking the rankings. 2. Return to Main Course step 2.

4.4 Login to system

Name	Login to System
ID	user_001
Description	This will be the first action by any user using the CRSS. Allows the user to access the system and perform actions.
Actors	Administrators, Instructors, Campers
Organizational Benefits	Ensures access to only those who have been granted it. Allows for actors to be routed to the correct user interface given their level of authorization.

Frequency of Use	Campers: multiple times before camp. *Q1: Does the camper have access to the system while attending camp?* Instructors: multiple times before and during the session. Administrators: multiple times before and during the session.
Triggers	The user enters their credentials onto the login screen.
Preconditions	System has to exist. The user must have valid credentials.
Postconditions	The user will have access to actions within their authorization level.
Main Course	1. The user navigates to the system. 2. The user is prompted to login. 3. The user has access to actions within their authorization level.
Alternate Courses	AC1 User was already logged in. 1. Return to Main Course step 3. AC2 User entered incorrect information. 1. Return to Main Course 2.
Exceptions	EX1 System fails to interpret users login credentials. 1. Return to Main Course step 2. Outstanding questions: *Q 4.4.1* - Would Camp Voyager want a security implementation to limit the amount of times the EX1 flow can happen? If so, how would they like to handle being locked out of the system?

4.5 Create camper schedule

Name	Create Camper Schedule
ID	admin_003
Description	Once ranking is locked, software is run to create campers' schedules.
Actors	Administrator
Organizational Benefits	Automates the schedule building process.
Frequency of Use	Once per session.
Triggers	Build process action is initiated

Preconditions	System has to exist. System must be locked. Must have camper data to run. Administrator is logged in.
Postconditions	Schedules have been created and are available for viewing / printing.
Main Course	1. Admin chooses to run the software. 2. System confirms they want to build the schedule. 3. Schedules are created.
Alternate Courses	AC1 Admin clicks cancel on confirmation 1. System returns to Main Course step 1.
Exceptions	EX1 System fails to retrieve rankings 1. System returns an error message detailing the error. 2. System returns to Main Course step 1. EX2 System fails to generate schedules 1. System returns an error message detailing the error. 2. Return to Main Course step 1

4.6 Rank camper selections

Name	Rank Classes
ID	camper_001
Description	Ability to rank classes according to camper preferences.
Actors	Camper
Organizational Benefits	Assists campers in getting schedules they enjoy.
Frequency of Use	Outstanding questions: *Q 4.6.1* - should a camper be able to rank their classes only once, or should they be allowed to change rankings while the system is unlocked?
Triggers	Camper logs into system while it is unlocked.
Preconditions	System has to exist. Camper is logged into the system. System is not locked.

Postconditions	Camper has ranked class data on file.
Main Course	Outstanding questions: *Q 4.6.2* - If camper hasn't ranked courses, would the first menu guide them to rank classes? *Q 4.6.3* - Would camper have to always navigate to certain menu page to rank classes? *Q 4.6.4* - Does the system save temporary rankings? Or should it only save on submission? 1. Camper selects course and selects a respective rank 2. System saves course/ranking combination 3. System redirects camper to additional courses to rank 4. Camper submits class rankings
Alternate Courses	AC1 Camper is indifferent to courses ● See question 4.6.5 Outstanding questions: *Q 4.6.5* - Would the camper have to select rankings? If they have no preferences, should they still fill out the rankings? *Q 4.6.7* - Can the system save a 'null' ranking list given no data?
Exceptions	EX1 System fails to save course ranking 1. System notifies Camper error has occurred in saving the course ranking. 2. Return to Main Course step 1.

4.7 Manually add, cancel, update schedules

Name	Manually add, cancel, update schedules
ID	admin_004
Description	Administrator is given the ability to make manual changes to schedules of any student.

Actors	Administrator
Organizational Benefits	Serves as a precaution in case any conflicts arise.
Frequency of Use	Multiple times per session
Triggers	A student, for whatever reason, is no longer allowed to take a certain class.
Preconditions	System must exist. A student must also exist and this student must be taking at least one class. The system must be secure. Administrator is logged into the system.
Postconditions	The camper will be assigned a different class to the one they were previously enrolled in and or the class no longer exists.
Main Course	1. Admin is on landing page where they have options to make a class change 2. Admin selects camper whose schedule needs change 3. Display shows: 1. Old class information (Time, Class ID, Schedule) 2. New class information (Time, Class ID, Schedule) 4. System saves the new schedule 5. System notifies all affected parties, the student, the instructor which had its student removed, and the instructor which had the student added
Alternate Courses	AC1 administrator logs out 1. Administrator returns to login portal (See use case "Login to System") AC2 Admin enters the same class, one with the same Class ID that the camper is being removed from 1. System displays an error stating these are the same classes 2. Return user to Main Course step 2.

Exceptions	EX1 System fails to save revised schedule 1. System notifies Admin error has occurred in saving the revised schedule. 2. Return to Main Course step 2. EX2 Administrator loses connection to portal while editing schedule 1. System displays internet connection not stable warning upon completion of a form and requires the administrator to submit the form upon stable connection

4.8 Add class details

Name	Add class details
ID	admin_005
Description	The Administrator assigns the metadata that includes the enrollment size, number of seats per class, and the time at which the classes are going to happen.
Actors	Administrators
Organizational Benefits	Organizes the class registration process and gives all parties involved important information about the classes offered at camp.
Frequency of Use	Once per class
Triggers	Option to add class metadata
Preconditions	System must exist. The system must be secure. The administrator must be logged into the system. Classes must exist in the system.
Postconditions	Each class offered now has enrollment size, number of seats, days class is offered, and the time of day it is offered.

Main Course	1. Administrator is on landing page where they have options to add class metadata. 2. Administrator sets the metadata and saves their changes to the system. 3. Students and Instructors will be able to see this information upon registering and viewing class schedules.
Alternate Courses	AC1 Admin logs out 1. Admin returns to Login Screen (see "Login to System" use case) AC2 Admin adjusts metadata for a class that does not exist 1. System displays an error stating this class does not exist in the system 2. Returns user to Main Course step 2.
Exceptions	EX1 System fails to save revised metadata 1. System notifies Admin that an error has occurred in saving the metadata. 2. Return to Main Course step 2. EX2 Administrator loses connection to portal while editing schedule 1. System displays internet connection not stable warning upon completion of a form and requires the administrator to submit the form upon stable connection

4.9 See/print finished schedules online

Name	See and print finished camper schedules
ID	camper_002
Description	Campers and instructors have the ability to view and print the campers' schedules
Actors	Campers, Instructors
Organizational Benefits	Allows campers to print their schedule for the session to help with day-to-day organization; allows instructors to see their schedule to help campers with any problems that might arise

Frequency of Use	Anytime before or during a session
Triggers	Camper or instructor chooses option to view or print schedule
Preconditions	Camper or instructor must be registered and logged in to the system. Camper must have a schedule to view. To print, system must be connected to a printer.
Postconditions	Camper or instructor now has visual access to a camper's schedule as well as a printed copy if requested.
Main Course	1. Camper or instructor navigates from landing page to the specific camper's profile. 2. Camper or instructor selects an option to view the camper's schedule. 3. If desired, camper or instructor selects option to print the camper's schedule.
Alternate Courses	AC1: Camper or instructor chooses to cancel printing 1. Return to Main Course step 2.
Exceptions	EX1: Issue with printer (e.g. out of paper) 1. System notifies user that there is an issue with the printer and waits for further instructions from the user, which might include AC1

4.10 Online Report of Current Class List

Name	Online report of current class list
ID	inst_002
Description	Instructors have the ability to see and print a report of their current class list
Actors	Instructors
Organizational Benefits	Keeps instructors organized - helps with camp organization
Frequency of Use	Anytime during a session
Triggers	Instructor selects option to view a report for their current list of classes

Preconditions	Instructor is logged in to the system. Classes must exist. Instructor must be registered to teach classes. To print, the system must be connected to a printer.
Postconditions	Instructor now has visual access to a report of a class list as well as a printed copy if requested.
Main Course	1. Instructor navigates from the landing page to their profile 2. Instructor selects an option to view a report of their class list 3. If desired, instructor selects an option to print the class list
Alternate Courses	AC1: Instructor chooses to cancel printing of class list report 1. Return to Main Course step 2.
Exceptions	EX1: Issue with printer (e.g. out of paper) 1. System notifies user that there is an issue with the printer and waits for further instructions from the user, which might include AC1.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

An application is needed to make registering and looking at classes as easy as possible. This can be done via a desktop application, a web application, or a mobile application.

One performance requirement is a database large enough and fast enough to store user account information (Usernames and Passwords), the user roles (Camper, Camp Director, or Instructor), and the schedules for both the instructors and the campers.

Outstanding questions:

Q 5.1.1 - should the CRSS be a desktop application, web application, a mobile application, or a combination of them?

5.2 Security Requirements

Base level encryption is a requirement especially given the different account users using this application. Passwords should be considered strong (at least 6 characters, resulting in a combination of upper- and lower-case letters, symbols, and numbers, that do not contain any words found in a dictionary). The password should be mutable given system administrator permission.

5.3 Software Quality Attributes

The application should be robust and be maintained if any bugs or exploits arise after public release. It should also be easy to use and work without any unexpected behavior, such as, and not limited to: freezing and crashing, the ability to reuse the same account for multiple camp sessions, as well as the ability to back up account information.

5.4 Business Rules

Campers are given the ability to rank classes and to view and print their class schedules.

Instructors are given the ability to manually add a class that they will teach and to edit existing classes. They are also able to view / print both their own teaching schedules and campers' schedules

Camp directors are given the ability to run the ranking process, manually enter information to the class roster, set campers' schedules, and set an enrollment cap for specific classes after consulting with the respective instructors. Directors are also given the ability to reassign a Camper to a different class at any time. They can lock the system as well as remove and add course offerings.

6. Other Requirements

Appendix A: Glossary

CRSS: Class Rank and Sorting System

Types of users:

1. *Administrator: The camp director.*
2. *Instructor: The camp counselors or the system users who run classes.*
3. *Camper: The users who will fill out class rankings. They will have the lowest level of authorization.*

Types of use case ID's:

1. *User_#: applies to all users of the CRSS.*
2. *Admin_#: applies to admin level users of the CRSS.*
3. *Inst_#: applies to instructor level users of the CRSS.*
4. *Camper_# applies to camper level users of the CRSS.*

Appendix B: To Be Determined List

Functionalities to be determined can be found in sections:

- 2.1
- 2.2
- 2.4

- 2.5
- 2.6
- 3.2
- 3.3
- 3.4
- 4.4
- 4.6
- 5.1