Software Requirements Specification for CRSS

Version 0.01 Prepared by
Yiwen Wan wanxx149
Yao Zeng zengx352
Wing Yi (Pinki) Wong wongx596

University of Minnesota
January 2018

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the uses and the functionalities of the Class Rank & Sort System (CRSS). The functionalities and the interfaces of the software will be addressed. This document is intended for users of the software as well as the developers.

1.2 Document Conventions

This document follows the IEEE template for System Requirement Specification Documents.

1.3 Intended Audience and Reading Suggestions

- Typical users: camp director, instructors and attendees who have different authorities and want to use the CRSS to rank, sort, schedule and document attendance for the summer camp.
- Developers who are implementing the system.
- Other programmers who may further adjust the system.

1.4 Product Scope

CRSS is a system that identifies the users with different authorities and provides respective features to different user scopes. The system is visualized within a graphic window that the users can log on with their unique ids for different purposes. For attendees, they have limited access to the system where they could only rank classes and view schedules. For instructors and the camp director, they have different levels of authorizations and can perform different administrative features in the system. This document will not specify any testing of the software.

2. Overall Description

2.1 Product Perspective

CRSS is developed for attendees, instructors and camp director at Camp Voyager to rank and manage classes. The system allows attendees to submit user ranking forms for picking classes based on their preferences. It will store the ranking results as well as the submission times for each individual in the database and run the class enrollment process after the deadline once the camp director locks the ranks. The system also provides the enrollment information for the camp director to analyze the popularities of the classes, thus helps to make decisions in regard to increasing or decreasing class offerings. It gives a

structured online form for the instructors to record attendances for the attendees to verify if the students are in the correct assigned classes.

2.2 Product Functions

System Level:

- Login: Redirects users to other pages based on their unique ids and passwords
- check_Student: Checks if any assigned classes violates the class enrollment rules
- assign_Student: Assign students to classes for each timeblock while checkStudent is true
- Sort: Sorts the results based on submission time of students
- check_Instructor: Check if no assigned classes violates the teaching schedule rules
- assign_Instructor: Randomly assigns instructors to classes while checkInstructor is true
- Attendance: Generate a list of students enrolled in each class

Attendee Level:

- Rank: User ranks the classes with numeric value
- View: Generates schedule for the users
- Print: Allows user to print out the generated schedule

Instructor Level:

- Add: Assign classes to late camp attendees
- View: Generates schedule for the users
- Print: Allows user to print out the generated schedule
- Contact: Allows user to contact administrators if any issues are encountered
- Take_Attendance: Take attendance using the list of students that system generates

Camp Director Level:

- Run: Request system to perform sort and assign_Student
- Lock_Ranking: Locks the latest run as the final ranking for the term, a unique identifier is assigned to this term upon lock_ranking is performed
- Add: Assign classes to late camp attendees
- Adjust_Offering: Increase or decrease class offerings based on overall attendees' Rank

2.3 User Classes and Characteristics

- Typical users: camp director, instructors and attendees who have different authorities and want to use the CRSS to rank, sort, schedule and document attendance for the summer camp.
- Developers who are implementing the system.
- Other programmers who may further adjust the system.

2.4 Operating Environment

- Windows 7
- Windows 8
- Windows 10
- Mac OS
- Linux

2.5 Design and Implementation Constraints

CRSS is developed in Java. It is designed by well-structured APIs where each feature of the system is developed in a separate sub-program and all are dependent to each other based on the APIs.

2.6 Assumptions and Dependencies

CRSS is developed in Java and hence Java of version 7 or higher is required to be installed in the user's system.

3. External Interface Requirement

3.1 User Interfaces

1. CRSS Login Screen: There will be two text input boxes for unique id and password, and a login button.
2. Attendee's Dashboard: A main page for logged in attendees with "Class Ranking" and "My Schedule" options.
3. Instructors' Dashboard: A main page for logged in instructors with "Take Attendance", "My Schedule" and "Manually Add Attendee" options.
4. Camp director's Dashboard: A main page for logged in camp director with "Run Ranking", "Lock Ranking", "Manually Add Attendee" and "Adjust Class Offerings" options.
5. Attendee Class Ranking Screen: A list of classes and each has a drop-down menu that contains numeric value. There will be a submit button at the bottom of the list.
6. Attendee "My Schedule" Screen: Displays a class schedule with time block, class enrolled in that block, and the location of each class. There will be a print button.
7. Instructor "Take Attendance" Screen: A form with attendees' first and last names as the first and second columns, and dates and times as the title of rest of the columns with checkboxes corresponding to each attendee.
8. Instructor "My Schedule" Screen: Displays a class schedule with time block, class assigned for teaching in that block, and the location of each class.
9. Instructor and camp director "Manually Add Attendee" Screen: A form with attendee's name, attending term, classes and time schedules to be filled out.
10. Camp director "Run Ranking" Screen: A "Run Ranking" button.

11. Camp director “Lock Ranking” Screen: A “Lock Ranking” button, only active after at least one “Run Ranking” is performed.
12. Camp director “Adjust Class Offerings” Screen: A form containing the classes information including the names, the terms, the time blocks and the current number of offerings. An “Increase” and a “Decrease” button is attached for each of the classes. Once the “Increase” button is clicked, all available time blocks will be shown for adjustments. Once the “Decrease” button is clicked, all current time blocks for the chosen class will be shown for adjustments.

3.2 Software Interfaces

CRSS requires Java to be installed in the user’s system. It uses Oracle as the database management system to store and manage data.

3.3 Communication Interfaces

CRSS is integrated into the official website of Camp Voyager. Internet connection and a web browser are required for all users of the system. It can be connected with a MySQL, SQLite or PostgreSQL to store the user’s information.

4. System Features

4.1 Submit Class Ranking

4.1.1 Description and Priority

This is the attendee’s use case that allows user to submit user submission form. When the user wants to enroll in one of the classes, he needs to use this function to rank the classes so that he could be enrolled in his highest ranked classes as long as there is availability.

4.1.2 Stimulus/Response Sequences

Triggers:

User logs in as attendee before submission deadline.

Preconditions:

User is viewing a list of classes with short description of each class.

Postconditions:

The submitted ranking is visible to the admins for assigning classes.

4.1.3 Functional Requirements

Main Course:

1. System prompts users to rank classes using numeric values.
2. User ranks and submits the user ranking form (see AC1, EX1, EX2, EX3).
3. System stores the ranking of classes and submission time in database.
4. System redirects user to his ranked classes list to review the submission.

Alternate Course:

AC1 User logs off accidentally

1. System auto-saves the result.
2. Redirects user to “login” Use Cases.

Exception:

EX1 User does not rank any classes.

1. System notifies user that at least one class needs to be ranked.
2. Returns user to Main Course step 3.

EX2 User ranks two or more classes with same ranking.

1. System notifies user that no duplicate ranking is allowed.
2. Returns user to Main Course step 3.

EX3 User ranks a class with more than one ranking.

1. System notifies user that each class can only have one ranking.
2. Returns user to Main Course step 3.

4.2 View Class Schedule for Attendees

4.2.1 Description and Priority

This is the function for attendees to check the assigned schedule. After class enrollment is done, user can see and print out the time block, the class enrolled in that block, and the location of the class.

4.2.2 Stimulus/Response Sequences

Triggers:

User logs on after submission deadline and clicks on “View Schedule”.

Preconditions:

The admins have finished assigning classes.

Postconditions:

The final schedule with time block, class and location is visible to the user when he views the schedules. The final schedule is printable if the user selects print options.

4.2.3 Functional Requirements

Main Course:

1. System identifies users based on login information (see AC1).
2. User select the print option
3. System print out the schedules.

Alternate Course:

AC1 System identified user as student.

1. System displays the time block, the class enrolled in that block, and the location of the class.

Exception:

N/A

4.3 Check Class Session for Instructor

4.3.1 Description and Priority

This is an instructor’s use case of the CRSS for checking assigned class schedule. It lists the assigned class schedules for a logged-on instructor.

4.3.2 Stimulus/Response Sequences

Triggers:

The instructor uses log-in system to see which class it is.

Preconditions:

An instructor is going to check his/her class.
Postconditions:
The instructor gets the information about the class and decide to do something with this information
4.3.3 Functional Requirements
Main Course:

1. System determines user doesn't log in and redirect user to log in
2. User Log in (See AC2)
3. The system will show the instructor which session he is going to teach (See EX1, EX2)

Alternate Course:

AC1 System determines user is already logged on

1. Return to Main Course step 3

AC2 User logs off again

1. Return to Main Course step 1

AC3 User wants to check attendance in the instructor page

1. See Use case "Attendance taking from the instructor"

Exceptions:

EX1 A instructor doesn't assign into a class

1. See user case "Arrange Instructors into classes"
2. Use the link "Contact CD" to ask a claim for that

EX2 A instructor assigned into more than one class in a single block

1. See user case "Arrange Instructors into classes"
2. Use the link "Contact CD" to ask a claim for that

4.4 Attendance Taking from Instructor

4.4.1 Description and Priority
This function allows logged on instructors to check attendance for every attendee in his/her class. It provides a systematic way for instructors to verify if an attendee is in the right spot.
4.4.2 Stimulus/Response Sequences
Triggers:
Check attendance to the attendees by clicking the link "Take Attendance"
Preconditions:
The instructor would attend a summer camp to lead the attendees
Postconditions:
The instructor uses his ways to control the attendee activities, to record the information for his own session of summer camp
4.4.3 Functional Requirements
Main Course:

1. User click the name of the class
2. The system shows user the information of the class
3. User click the link, "Take Attendance"
4. The system shows user the list of students in the class
5. User click block near the students' name to check the students who attend the class (See AC1)
6. User put the date at the bottom of the page and click "submit"
7. The system will post the attendance information to the class page (See AC2, EX1)

Alternate Course:

AC1 Some students who have emergency (like illness) cannot attend the class

1. User would not check the block, but put comments about the emergency on the right of the students' name
2. If User give the evidence for the emergency after taking the attendance, use Alternative Course 2 (AC2)

AC2 User wants to make changes to the previous attendance information

1. User click the line "Make Changes" next to the previous attendance information in the class page
2. Return to Main Course step 4

Exceptions:

EX1 System fails to post the attendance information to the class page

1. System notifies users that an error has occurred
2. Return to Main Course step 6

4.5 Perform/Run Ranking Process

4.5.1 Description and Priority

This is a camp director's use case that allows to perform/run ranking process. It is performed on the collection of the rankings submitted by the attendees, then it determines the overall popularities of a given class/camp as well as the rankings for individual attendees.

4.5.2 Stimulus/Response Sequences

`Trigger:`

Camp director starts the process

Preconditions:

System recognizes the user as the camp director and the deadline is not passed.

Postconditions:

The rankings for each given camp are determined after each performance of the process. The preferences on class enrollments for attendees are determined.

4.5.3 Functional Requirements

Main Course:

1. System prompts user to run ranking process
2. User confirms to run
3. System determines users is not logged in and redirects user to log on (see AC2)
4. User logs on
5. System runs the ranking process
6. System saves the results and shows it to the user

Alternate Course:

AC1 System determines user is already logged on

1. Return to Main Course step 5

AC2 User logs off again

2. Return to Main Course step 3

Exception:

EX1 User has no authority in manually entering the information

1. System notifies the user that additional authority is needed.
2. Exit the system

EX2 No schedule fits the late attendee

1. System notifies the user with this information
2. Return to Main Course step 1

4.6 Lock Ranking

4.6.1 Description and Priority

This is a camp director's use case that allows to lock the rankings after a performed ranking process. After the rankings are locked, the system assigns a unique identifier to the term of the camp, and schedule classes for attendees who submitted their individual rankings before the deadline by their preferences and their submission times.

4.6.2 Stimulus/Response Sequences

Triggers:

The camp director decides to lock the ranking after his/her last run of the ranking process

Preconditions:

System recognizes the user as the camp director. At least one ranking process is performed before the lock.

Postconditions:

The ranking process cannot be run again, final rankings for each camp are determined and are being assigned as the result. A unique identifier is assigned to each term of the camp.

4.6.3 Functional Requirements

Main Course:

1. System prompts user to lock ranking
2. User confirms to lock
3. System determines users is not logged in and redirects user to log on (see AC2)
4. User logs on
5. System locks ranking, assigns unique identifiers to each term of the camp and assigns classes to attendees.
6. System disables "run ranking process" function
7. System stores the results as final rankings and shows to the user
8. Alternative Course:

Alternate Course:

AC1 System determines user is already logged on

1. Return to Main Course step 5

AC2 User logs off again

Return to Main Course step 3

Exception:

EX1 Ranking process has not been run

1. See "Perform/run ranking process" use case.

EX2 User has no authority in locking the rankings

1. System notifies the user that additional authority is needed.
2. Exit the system

4.7 Manually Enter Information

4.7.1 Description and Priority

This is a use case that allows both the camp director and a instructor to manually enter an attendee's information for class enrollment who submitted his/her rankings after the deadline and the camp director locks the rankings. Class will be assigned manually by the user based only on class availability, if there is no suitable class for such an attendee, he/she may not be enrolled into the camp.

4.7.2 Stimulus/Response Sequences

Triggers:

A late attendee wants to participate.

Preconditions:

1. The presence of at least one late attendee who wants to join the camp after the camp director locks the ranking.
2. At least one class is available.
3. System recognizes the user as the camp director or an instructor

Postconditions:

The late attendee gets to enroll into the camp

4.7.3 Functional Requirements

Main Course:

1. System directs user to a page with an information form to fill in.
2. System prompts user to fill in information
3. User fills in information
4. System determines users is not logged in and redirects user to log on (see AC2)
5. User logs on
6. System processes and stores the information of the attendee
7. System enrolls the attendee

Alternate Course:

AC1 System determines user is already logged on

1. Return to Main Course step 6

AC2 User logs off again

1. Return to Main Course step 4

Exception:

EX1 User has no authority in manually entering the information

1. System notifies the user that additional authority is needed.
2. Exit the system

EX2 No schedule fits the late attendee

1. System notifies the user with this information
2. Return to Main Course step 1

4.8 Adjust Class Offerings

4.8.1 Description and Priority

This is a camp director's use case that allows to increase or decrease the number of a class offered in a term in the future. Once the rankings are locked for the current term, the camp director investigates through the preferences based on the attendees and can make reasonable adjustments to the class offerings in the upcoming terms of the camp.

4.8.2 Stimulus/Response Sequences

Triggers:

Camp director decides the increase or decrease the number of a class offered.

Preconditions:

System recognizes the user as the camp director, the current camp ranking is locked.

Postconditions:

Class offerings are adjusted in the next term.

4.8.3 Functional Requirements

Main Course:

1. System directs user to a page with class information
2. System prompts user to increase/decrease the offerings of a particular class
3. User confirms the adjustments.
4. System determines users is not logged in and redirects user to log on (see AC2)
5. User logs on
6. System processes and stores the adjustments for the class offerings
7. System shows the user the results of the adjustments.

Alternate Course:

AC1 System determines user is already logged on

1. Return to Main Course step 6

AC2 User logs off again

1. Return to Main Course step 4

Exception:

EX1 User has no authority in manually entering the information

1. System notifies the user that additional authority is needed.
2. Exit the system

EX2 The offerings of a class that is being adjusted has reached to 0 or has passed the maximum number that a camp can offer

1. System notifies the user that an error has occurred.
2. Return to Main Course step 1

4.9 Identify User in Login Process

4.9.1 Description and Priority

This is the system use case that system identifies user as campus director, instructor, or student based on the unique id for user. It will redirect user to different use cases for further actions.

4.9.2 Stimulus/Response Sequences

Triggers:

The user clicks on "Log in".

Preconditions:

The user knows his own id and password.

Postconditions:

The system will redirect users to other use cases based on what the user needs.

4.9.3 Functional Requirements

Main Course:

1. System determines user is not logged in and redirects user to log on.
2. User logs on (see AC1, EX1).
3. System identifies user based on the unique id and password (see AC2, AC3, AC4, AC5, AC6, AC7).

Alternate Course:

AC1 User logs off again.

1. Return to Main Course step 1

AC2 System identifies user as attendee before ranking submission deadline.

1. See "Submit Class Ranking" Use Case

AC3 System identifies user as attendee after ranking submission deadline.

1. See "View Class Schedule for Attendees" Use Case

AC4 System identifies user as instructor before ranking submission deadline.

1. System notifies user that class ranking is currently in progress and final schedule is not completed yet.

AC5 System identifies user as instructor after ranking submission deadline.

1. See use case "Check Class Session for Instructor"

AC6 System identifies user as camp director before ranking submission deadline.

1. See use case "Perform/Run Ranking Process"

AC7 System identifies user as camp director after ranking submission deadline.

1. See use case "Manually Enter Information"

Exception:

EX1 User's login information does not match any records in the database.

1. System notifies user that login/password is wrong.
2. Returns user to Main Course step 1.

4.10 Assign Students to Classes

4.10.1 Description and Priority

This is a system use case that assigns students to classes based on preference, submission time, and class availability.

4.10.2 Stimulus/Response Sequences

Triggers:

After submission deadline, and the camp director locks the rankings.

Preconditions:

Class ranking and students' submission time is stored in database.

Postconditions:

Schedule is available for view by students.

4.10.3 Functional Requirements

Main Course:

1. System sort students according to their submission time (AC1).
2. System starts at the first-time block.
3. Assign students to classes based on their ranking (AC2, AC3) and store the result in database.

4. System move to the next time block and repeat step 3 until no more time block can be found.

Alternate Course:

AC1 Students have the same submission time.

1. System sorts students by their names in alphabetical order.
2. Return to Main Course Step 3

AC2 Class is full.

1. System assign students to their next ranked class and store the result in database.
2. Return to Main Course step 4.

AC3 Duplicate classes are assigned.

1. System assign students to their next ranked class and store the result in database.
2. Return to Main Course step 4.

Exception:

N/A

4.11 Arrange Instructors into Classes

4.11.1 Description and Priority

This is the system user case that arrange the instructors into different classes to let every class have an instructor. It will arrange instructor to the classes to let instructor and attendees know who is going to take this class.

4.11.2 Stimulus/Response Sequences

Triggers:

Pick instructor in the instructor list and put them into random classes

Preconditions:

TBA will be shown at the instructor name in the class information page

Postconditions:

An Instructor will be shown in each class information page

4.11.3 Functional Requirements

Main Course:

1. System notices that there will be no instructor in a class
2. System randomly pick a name in the instructor list (See EX1, EX2)
3. System put the name into the class
4. System will post the name of the instructor into class information page

Exceptions:

EX1 System fails to arrange an instructor into a class

1. System notifies that an error has occurred
2. Return to Main Course step 2

EX2 The number of instructors is less than the number of classes

1. System sends this situation to camp director
2. System waits for CD's command
3. Return to Main Course step 1