Software Requirements Specification

for

Class Rank & Sort System

Version 1.0 approved

Prepared by Hanzhang Wu, Jerry Nie, Yanjun Cui,Yangjiawen Xu

University of Minnesota, Twin Cities

09/15/2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

Class Rank & Sort System	15 September 2019	Create this document and finish the main part of this document.	Alpha0.1
Class Rank & Sort System	21 September 2019	Basically finish the document and leave specific questions for further development.	Alpha0.2
Class Rank & Sort System	23 September 2019	Do some supplements and review to make some improvement.	Alpha0.3

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 system, the interfaces of the system, what the system will do, the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Camp Voyage.

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

- Campers, who want to use the Class Rank & Sort System to select the class they want to participate in.
- Instructors or directors, who want to use the Class Rank & Sort System to manage the information and specifications needed for the camp.
- Advanced/Professional Users, such as engineers and analyst, who want to use Class Rank & Sort System for further improvement and data analysis.
- Programmers who are interested in working on the project by further developing it or fix existing bugs.

1.4 Product Scope

This software system is aimed at summer camp educational institutions, and provides basic curriculum management functions for directors, instructors and students. The purpose of this system is to improve course management efficiency by providing tools to help course takers and course providers with course management functions. Otherwise, it must be implemented manually.

To be more specific, students could get enrolled in classes without time conflicts based on the timing of their submission of rankings. Instructors and directors could set/modify course settings, manage course enrollment of students, and make the course management system functional.

1.5 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.*IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

Class Rank & Sort System (CRSS) was developed for campers to select the class they want to participate in, submit class rankings and get preference on class enrollment based on the timing of their rankings. It was also developed for instructors to see enroll specific students, see campers' schedules, see the online report of a class list and see the students in their classes. Also, it was developed for camp directors to assign classes to campers, add/remove class offerings, analyze students needs, analyze classes for enrollment, set and modify class details
and enroll specific students.

2.2 Product Functions

Login page:

- Login: Log into the main page.
- Cancel: Clear user's name and password.
- Create an account: Link to an outside page and create a new account.
- Forget your name or password: Link to an outside page to set a new password.

Main page:

- My information: View user's information. (For camper, instructor and camp director)
- Log out: Log out your account. (For camper, instructor and camp director)
- My schedule: See users class schedule. (For camper and instructor)
- Course rankings: Choose courses and submit user's course rankings. (For camper)
    - View class list: View what class do the camp provides.
    - Search: Search specific class the user wants to take.
    - Choose classes: User choose what classes he/she wants to take.
    - Save: Save user's ranking.
    - Submit: Submit user's ranking.
    - Clear: Clear user's selection.
    - Quit: Quit your selection and exit this page.
- View class: View all classes. (For camp director and instructor).
    - Class list will show all classes.
    - Information column: view class description and instructor information.
    - Action column: edit or delete exist classes.
    - Enrollment column: View and edit enrollment information(Only can be accessible after run the "Lock & Run" process).
- Lock & Run: Lock student course rankings, and perform course arrangement program. After "Lock & Run", the enrollment information can be viewed by directors.

- View Campers: View all campers name, ID and information. (For camp director)
- Specific Enrollment: Enroll a specific student into a specified class.(For camp director and instructor)

2.3 User Classes and Characteristics

- Campers, who want to use the Class Rank & Sort System to select the class they want to participate in.
- Instructors or directors, who want to use the Class Rank & Sort System to manage the information and specifications needed for the camp.
- Advanced/Professional Users, such as engineers and analyst, who want to use Class Rank & Sort System for further improvement and data analysis.
- Programmers who are interested in working on the project by further developing it or fix existing bugs.

2.4 Operating Environment

- Windows 10
- Mac OS X
- Linux

Question: What operating system does the software need to be compatible with?

2.5 Design and Implementation Constraints

Class Rank & Sort System is developed in Java and built on Visual Studio Platform. The computer executing the program is recommended to have at least Intel 5-gen dual-core i3 CPU at 1.60GHz or AMD CPU that has similar performance, and with at least 1GB RAM. The recommended graphic requirement is at least DirectX 10 compatible GPU with at least 512mb RAM. We divided features into separate modules that depend on each other through well-written APIs available to make development easy. The program should be developed in English, and its databases should be developed in MySQL.

2.6 User Documentation

After the software has been developed, a video will be recorded to demonstrate how the user implement the software.

2.7 Assumptions and Dependencies

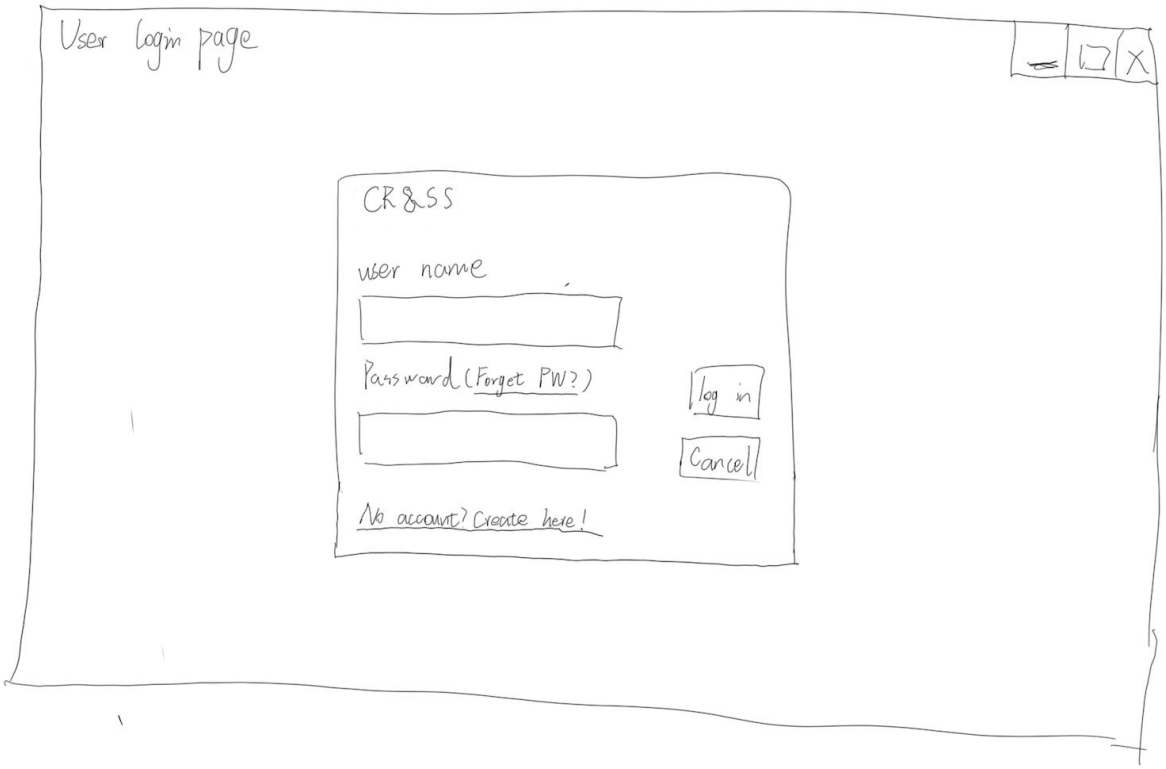
Class Rank & Sort System is developed in Java and therefore requires Java to be installed on the user's system. The latest stable version of CRSS requires Java version 8 or higher. This applies to Windows and Linux users. On Mac OS X, Java is bundled with the application.

3. External Interface Requirements

3.1 User Interfaces

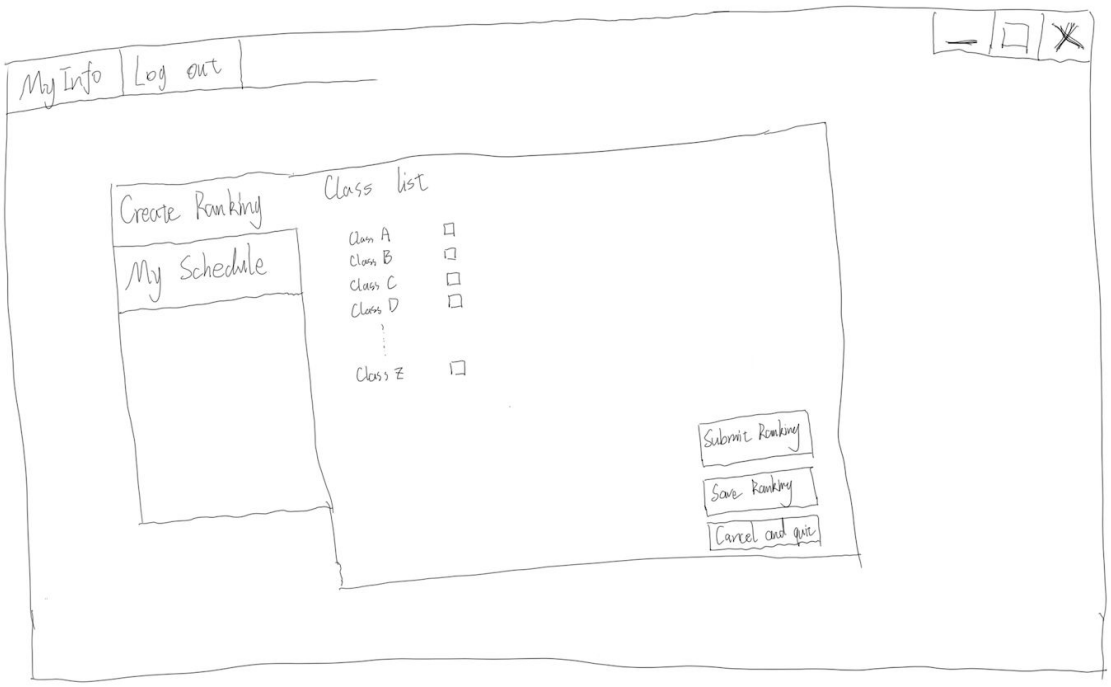
The program can work through terminal / command line.

We also have the following tentative interfaces:
1.User Login Page



2.Camper Main Page

Camper Main page



3.Instructor/Director Main Page

3.2 Hardware Interfaces

The minimum hardware requirements of Class Rank & Sort System are a 1.6GHz CPU and 1 Gb of RAM.

3.3 Software Interfaces

Class Rank & Sort System requires Java to be installed on the system, more specifically Java version 8 or higher. Additional information can be found on section 2.7 of this document.
Class Rank & Sort System can be connected with MySQL database to import a graph edge list.

3.4 Communications Interfaces

Class Rank & Sort System requires an internet connection to install new plugins, update already installed ones and update some of its components (APIs, modules etc.).

4. System Features

(We are not sure about what to do here.) Is this like we need to use database or what algorithm we used?

This section demonstrates Class Rank & Sort Software's most prominent features and explains how they can be used and the results they will back to the user.

4.1 User Organization Structures

4.1.1 Description and Priority

In this system, there are three types of uses: campers, instructors, and camp directors. Every person in this system has a unique account. Campers can see their own course rankings and schedules. Also, they can modify their information and course selections. Instructors can see what classes they are teaching and the students name in their courses and their courses' information. Camp directors have the most power and functionality. They can view all students' profile and their submission for course rankings and courses' information. They can also lock the system and then run it to generate rankings. After the rankings are generated, they can enter the manual entry for adding/removing classes for campers.

4.1.2 Stimulus/Response Sequences (Question: Does it means we need to describe how do people's input and output are realized?)

Users can only register accounts with the lowest privileges(camper account) at the time of registration. Instructor and director accounts are specified by the software system. After registration, users could do their work according to the simple interface.

4.1.3 Functional Requirements (Question: Does it like if the people input an invalid ID, or wrong password, how do we respond to this action?)

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1: User must have correct id and password.
REQ-2: User

4.2 Topological sorting (and so on)

4.2.1 Description and Priority

"In the field of computer science, a topological sort or topological ordering of a directed graph is a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v, u comes before v in the ordering. For instance, the vertices of the graph may represent tasks to be performed, and the edges may represent constraints that one task must be performed before another; in this application, a topological ordering is just a valid sequence for the tasks. A topological ordering is possible if and only if the graph has no directed cycles, that is, if it is a directed acyclic graph (DAG). Any DAG has at least one topological ordering, and algorithms are known for constructing a topological ordering of any DAG in linear time. "
--wikipedia

Topological ranking is used to rank students' curriculum. Camp courses have pre-courses and time. Topological ordering is used to rearrange students' curriculum schedule so that all semesters' courses can meet their pre-courses and time schedule.

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

TBD, we need further research and actual apply.

4.1.3 Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>*TBD

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

TBD, we need further research and actual apply.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The computer executing the program is recommended to have at least Intel 5-gen dual-core i3 CPU at 1.60GHz or AMD CPU that has similar performance, and with at least 1GB RAM. The recommended graphic requirement is at least DirectX 10 compatible GPU with at least 512mb RAM.

5.2 Safety Requirements

To ensure that no one of Class Rank & Sort Software's users loses any data while using Class Rank & Sort Software(due to a crash or a bug), the developer will upgrade the software regularly. Also, the software allows the user to report the bug to developer team.

5.3 Security Requirements

Only qualified users can access the Class Rank & Sort System. The qualified users contains three different kinds: Campers, instructors and camp directors. Camp directors have the highest level of security, and campers' security are the lowest. Before any action is executed, the system checks the current user's login level to avoid low-level users performing high-level operations.

Instructors can only see the course information of the courses they are responsible for, and the course title of other courses; directors can see the details of all courses.

System will automatically check current user login status. Any login that has not been operated for more than 10 minutes will be expired.

5.4 Software Quality Attributes

Class Rank & Sort System provides the users with both simple and advanced features. Due to its well designed and easy to use interface it can be used by both experts and typical users. However, users must already have a basic knowledge of graphs before using it.
Class Rank & Sort System provides the user with basic features. Users can watch a video before using the software.

5.5 Business Rules

Class Rank & Sort System is for academic uses only. If anyone uses this software for business purposes without consent, the perpetrator will be prosecuted to the fullest extent according to the law.

6. Other Requirements

We are not sure what other requirements we need to consider.

Appendix A: Glossary

References: https://en.wikipedia.org/wiki/Main_Page

- Hashing: a search method using the data as a key to map to the location within memory, and is used for rapid storage and retrieval.
- Sorting: a process of organizing data from a random permutation into an ordered arrangement, and is a common activity performed frequently in a variety of applications.

Appendix B: Analysis Models

TBD, This one is optional and we do not have any of analysis models yet, more information needs to be provided.

Appendix C: To Be Determined List

To be determined.