Software Requirements Specification

for

Class Rank & Sort System

Version 1.0 approved

Prepared by Junren Huang, Jian Wang, Pengyin Chen, Shuai Hao

University Of Minnesota at Twin Cities

February 8, 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
CRSS	02/05/20	First time created	1.0
CRSS	02/09/20	Detail modification	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to establish a detailed description of the Class Rank & Sort System (CRSS). This document will explain the primary purpose and features of the system. It also includes the interface of the system, the primary function of the system. And the constraint of the system and how the system responds to external stimuli. This document not only intends to regular users but also for 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

- General users like campers, instructors, administrators who are involved in offering a course and selecting a course.
- Advanced/Professional Users, such as engineers or researchers, who want to use CRSS for more demanding class activities.
- Programmers who are interested in working on the project by further developing it or fix existing bugs.

1.4 Product Scope

CRSS is a tool that users can use to offer, select, and rank courses and produce schedules. Users can use it to represent classes and interact with them. They can build their schedule, edit information of courses, update user's schedules and so on so that it could improve management efficiency and satisfy different groups of people's needs.

1.5 References

IEEE Template for System Requirement Specification Documents:

https://goo.gl/nsUFwy

CRSS's GitHub page:

https://github.umn.edu/umn-csci-5801-002-s20/repo-Team5

2. Overall Description

2.1 Product Perspective

CRSS was developed for everyone who has the need to manage courses like offering courses, ranking courses, building schedules and so on so that they can satisfy their needs or use it as an efficient method of managing courses. It can provide support to different roles(campers, instructors, and administrators) and handle various situations such as updating, ranking, sorting, analyzing and so on.

It is a team project, and it has a very active developer team to support it and provide feedback to users. It was developed to run on Windows, Mac OS X and Linux.

2.2 Product Functions

Mainly this section describes the authority of different kinds of users separately. There are three different kinds of users in the system: camper, instructor and administrator. After each user login, the system will justify their identification and set up different authority for the user.

2.2.1 Camper case

Brief Description

The user has to have an account to access the Class Rank & Sort System. The functionality for camper level is rank classes and each camper can only access their own class schedule

Initial Step-By-Step Description

Before the use case can be initiated, the Camper has to access the Class Rank & Sort System.

1. System displays the classes that are available
2. Camper ranks 10 classes that he want
3. System stores the saved class ranking
4. Camper submit ranked classes
5. Camper confirm submission

2.1.2 Instructor Case

Brief Description

In this case, the user has to have an account to access the system to fully use the system function. After authorization, instructors can offer classes that they will teach and report information to administrators. After the final schedule is generated, instructors will be able to view the class list and print as they want. Also, instructors can view any camper's schedule.

Initial Step-By-Step Description

Before the use case can be initiated, the Instructor has to access the Class Rank & Sort System.

1. Instructor type in the class name that they want to offer for each session
2. Instructor submits class information to administer
3. After all classes been scheduled, instructors can view and print the class list

4. Instructors can check any camper's schedule by pressing the camper's name button

2.2.3 Administrator Case

Brief Description

The user has to have an account to access the system. Administrators have the highest privilege in the system. Most functions are made for administrators. Administrators organize all the classes by following the rule of the camp class setting in the system. After all campers finish ranking class, the administrator will rank class form high to low and show it as a list. System generates class schedules through an administrator. Administrators can also edit class information after class schedules are generated.

Initial Step-By-Step Description

Before the use case can be initiated, the Administrator has to access the Class Rank & Sort System.

1. System display all class that offer by instructors
2. Administrators organize class by following the rule of the class setting
3. After all camper ranked their class, administrator rank class form high to low
4. After the system is locked down, an administrator generate the class schedule
5. If some emergency happens, administrators have the authority to add or remove class
6. Administrators can also add campers to different classes, or remove camper information from the original class.

2.3 User Classes and Characteristics

The camper is expected to be internet literate and be able to read, register/login and choose classes.

The instructor is expected to be internet literate and be able to use buttons, menus, scroll bar and similar tools. Instructor can offer a class.

The administrator is expected to be internet literate and familiar with computer software and be able to use buttons, menus, scroll bar and similar tools.

2.4 Operating Environment

- Windows 7/8/10
- OS X
- Linux

2.5 Design and Implementation Constraints

Our system is developed by Java. It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other through well.

2.6 User Documentation

There will be a user guide.

2.7 Assumptions and Dependencies

SRS is developed in Java and therefore requires Java to be installed on the user's system. The latest stable version of Gephi requires Java version 7 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

There will be interfaces for different groups of users to login, create account, rank classes, submit, show class sections, class details, class instructors. There will also be interfaces for campers to view their assigned calendars, and print calendars option.

3.2 Hardware Interfaces

The minimum hardware requirements of Gephi are a 500 Megahertz CPU and 128 megabytes of RAM. Also, a compatible graphics card is required to make sure CRSS could have high-quality representation.

3.3 Software Interfaces

CRSS requires Java to be installed on the system, more specifically Java version 7 or 8 for its latest release. CRSS can be connected with a MySQL, SQLite or PostgreSQL database to process identification and import and store information of classes.

There is an interface which links our software to a database which stores general class information, all campers' ranks, enrollment data and class calendars. To better analyze the student's needs and enrollment status, it is necessary to store all the data in the database. There are separate tables storing the campers' ranks, enrollment status, enrollment analysis, students' needs analysis, and generated calendars.

The campers' ranking table will include students' name, student id, ranked class, class id.

The enrollment status table will include course name, course id, course time, course instructor, enrolled students id, enrolled student name.

3.4 Communications Interfaces

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

Our software will use HTTP and TCP/IP protocols to allow the communications between the back-end server and front-end interfaces. We will also use the FTP protocol to implement the email communication between our web-app and campers such as email type alerts, confirmations.

All campers' passwords will be encrypted by our software to guarantee security. Meanwhile, the back-end server will use a multi-thread mechanism to guarantee the data collected from users will not lose, interact with other user's requests and avoid the dead block. User sessions are also implemented on the server side to guarantee integrity constraints.

4. System Features

This section demonstrates the SRS most prominent features and explains how they can be used and the results they will give back to the user

4.1 Class Information Set-Up

4.1.1 Description and Priority

System allows users to provide class information. Users can add, remove, update and change course information any time they want. This is a high priority feature, many features depend on this feature.

4.1.2 Stimulus/Response Sequences

1. System prompts user to log in
2. After logging in, the user could be able to add or edit classes.
3. User adds a new class or edits a class
4. System stores the save classes
5. User submits and the system redirects the user to their saved items list to view their list of classes
6. User confirms submission

4.1.3 Functional Requirements

REQ-1: Authenticity management

4.2 Authenticity Management

4.2.1 Description and Priority

System will identify every user login to the system. This feature makes the system more secure and reliable. This is a high priority feature, and this is required for all other features.

4.2.2 Stimulus/Response Sequences

1. User tries to access CSRS
2. System prompts to process identification such as sign in or log in
3. User inputs his information
4. System verifies information and grant access to the user
5. System redirects user to the main page

4.2.3 Functional Requirements

REQ-1: A database?

4.3 Campers Rank Classes

4.3.1 Description and Priority

Campers rank courses they want, and the course will be ranked by the number from high to low. This is a medium priority feature.

4.3.2 Stimulus/Response Sequences

1. System prompts user to log in
2. After logged in, show a list of available classes
3. User ranks classes
4. System stores the saved class ranking
5. User submits classes rank, and then system redirects the user to their saved ranking list to view the full list.
6. User confirms submission

4.3.3 Functional Requirements

REQ-1: Class Information Set-Up
REQ-2: Authenticity Management

4.4 System Generates The Schedule

4.4.1 Description and Priority

After the system is locked down, an administrator runs the system to generate a schedule for each camper. This is a low priority feature.

4.4.2 Stimulus/Response Sequences

1. System prompts user to log on
2. User confirms to generate schedule
3. System stores the schedule
4. System redirects the user to their saved items list to view their list of classes with detailed information.
5. System gives access of the schedule either by person or by course to instructors
6. System gives each camper access to his own schedule

4.4.3 Functional Requirements

REQ-1: Class Information Set-Up
REQ-2: Authenticity Management
REQ-3: Campers Rank Classes

4.5 View And Print Schedule or Class Information

4.5.1 Description and Priority

After the schedule is generated, instructors and administrators could be able to view and print schedules, campers can only view and print their own schedules. This is a low priority feature.

4.4.2 Stimulus/Response Sequences

1. System prompts user to log on
2. User confirms to view and print schedules
3. System print

4.4.3 Functional Requirements

REQ-1: Class Information Set-Up
REQ-2: Authenticity Management
REQ-3: Campers Rank Classes
REQ-3: System Generates The Schedule

5. Other Nonfunctional Requirements

5.1 Performance Requirements

CRSS requires a system with at least a 500 megahertz CPU and 128 megabytes of RAM. Performance depends on the size and as a result, the system requirements for higher quality are more demanding.

5.2 Safety Requirements

To ensure the safety of all user's data of Class Rank & Sort System(due to a crash or a bug of some kind) the developer team updates Class Rank & Sort System regularly. There is a bug tracker available where users can report any bugs they have encountered so that the developers can fix it in the next release.

5.3 Security Requirements

Class Rank & Sort System does not have any security requirements and thus any type of user can use it without any additional privileges.

5.4 Software Quality Attributes

Class Rank & Sort System provides the user 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 computers before using it.

5.5 Business Rules

Class Rank & Sort System cannot be sold by their party agency or any organization without permission from founders.

6. Other Requirements

Class Rank & Sort System requires a database to store and import information of classes.

Appendix A: Glossary

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

- Rank: the relative position, value, worth, complexity, power, importance, authority, level, etc. of a person or object within a ranking.
- database: A database is an organized collection of data, generally stored and accessed electronically from a computer system. Where databases are more complex they are often developed using formal design and modeling techniques.

Appendix B: Analysis Models

The analysis model will be based on roles like campers, instructors and administrators. Our system will satisfy different roles, and based on their roles, we expand functionality that they may use in different situations.

Appendix C: To Be Determined List

- Database
- Manual page
- user guide
- User interface design
- Expected exception message