**CSCI 5801: Software Engineering I, Spring 2020**
**Homework 1 – Software Requirements Specification (SRS) Document**
**Due Date: Sunday, February 9 at 11:55 p.m.**
**50 points total**

**Special Instructions**: You will be working in your small groups to complete this homework assignment. You should meet, skype, Google Hangout, or talk on the phone (if unable to meet in person) about the requirements for the assignment. You will only turn in one assignment per team. You must include all names on your assignment with X500 names along with your Team #. Please use the name that is listed on the class roster so we will know who you are. You will upload your work to GitHub. Only one person should upload to GitHub the final document. We encourage you to upload drafts to GitHub so you can never lose your work. We expect you to turn in your Software Requirements Specification document (must be typed) and your use cases (either incorporated in your SRS document or as a separate document (must be typed). You will be allowed to upload multiple documents. Be aware that nothing turned in after 11:55 p.m. on Monday, February 9 will be graded. We will not accept late work so be sure to push your final work to GitHub in advance of 11:55 p.m. If you are not done, turn in what you have so we can give you partial credit.

## The Problem

Summer camps can be fun and educational, and attendees often make some of the best memories of their young lives. This is especially true when the camp offers opportunities that are not available outside of the camp environment. One way to effectively engage attendees in the classes offered is to allow attendees, what we will refer to as "**campers**", to rank the classes that they can take and enroll campers in their highest ranked classes (as long as there is availability). You and your team of software engineers have been hired by Camp Voyager to implement the Class Rank & Sort System (CRSS) for assisting the various stakeholders with the ranking, sorting, scheduling and attendance documentation needs for such a camp.

### Your Mission

Your mission, should you choose not to fail the assignment, is to begin the process of developing the system by creating the first draft of a software requirements specification (SRS) document. The term "draft" does not imply poorly written but rather there may be holes that have only questions in the section of the document because you have not yet had a chance to ask questions of the users directly (e.g. elicitation, interviews.) This does not mean that you cannot start the document. You have been provided a detailed write up from the Camp director to get you started. The proposed system should be able to do the following:

- Campers need to be able to rank the classes they want to take when they attend the camp. They will attend 3 classes per day (2 hours classes). The duration of a camp session is 5 days, Monday – Friday. There will be at least 4 camp sessions throughout the summer (denoted as Session 1, Session 2, Session 3, and Session 4).
- Instructors need to offer classes. They 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. For example, an Intro to Swimming could be offered multiple times on the same day or on different days throughout a session. We would want to have the Intro to Swimming offer the same class material but possibly with different instructors. Thus we need to assign sections to the classes so we can ensure that only one section is being offered in one time slot. Instructors set the number of students for each class that they offer. They work with the camp administrator to determine this number. The administrator or their assistant can input the classes' information into the system. 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. Students will not always get the class that they want.
- Administrators will run the software to place the campers in the classes based on their rankings. The software will only be run once when the camp registration closes for a given camp session (e.g. Session 1). Any class movements after the software has been run will be completed by hand but you must provide the mechanism to enter the manual entry for adding and deleting campers from classes.

- Campers need to see their class schedules on line after being assigned their classes for the week by the administrators. Campers should be able to print the schedule also.
- Administrators should be able to:
    - Analyze classes for enrollment (low/high)
    - Add or remove class offerings
    - Analyze student needs (empty blocks/duplicates/etc.)
    - Add class information to the system (they work with the instructors to create day and times along with the number of seats.)
- Instructors need to be able to see an online report of a current class list and should be able to print it.
- Instructors need to be able to see a camper's schedule online.

## Class Rank & Sort System (CRSS)

Here you will get a set of informal "user requests" of what this system shall do. Note that this is purposely incomplete and vague so that you have a chance to develop questions. For the project where you will be developing an SRS document there will be an elicitation session. You will want to document you questions fully within the document itself if you are not able to fully describe the section.

The system should:

1. Have three levels of users, each having different sets of rights (authorization):
    a) Camp Director level
    b) Instructor level
    c) Camper (i.e. attendee) level
2. Allow campers to submit rankings only if a class/course ranking form is filled out and satisfies the following:
    a) At least 10 different of classes are ranked. The camper can rank up to 10 classes. Remember, these are the class titles (and not sections).
    b) No two classes can have the same ranking
    c) No class has more than one ranking
3. Allow the camp director to perform/run the ranking process.
4. Late camp attendees will be assigned classes based on availability. An instructor or camp director will manually enter their information to the class roster. This "override" will not require a full rerun of the assignment system. Remember, this is manual process by entering information into a specific class.
5. 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.
6. Allow the camp director to lock rankings (no changes other than manual entries will be allowed for this term of classes.) Each term of the camp will have a unique identifier (e.g. S18_Section1). There will only be one final ranking per session.
7. Attendees will not be enrolled in the same class twice.
8. Attendees will 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.
9. Some classes have different enrollment caps. The director sets these enrollment caps after consulting with the instructor.
10. Camper schedules will 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. There can be more than once instructor for a class. These schedules will be printable.
11. A class can be offered multiple times during a session by different instructors. Each class has its own unique identifier.
12. An instructor cannot teach more than one class in a single block.
13. Some class titles will be included in more than one camp. (e.g. Archery could be during all camp sessions.)
14. A camper can be reassigned a class at anytime by an administrator. This means removing the camper from one course and manually adding the camper to another class.
15. All users will have a username and password assigned to them. Only qualified users should be able to access the system. The username and passwords will be generated by the system and given to the users. Rights to different activities will be restricted based on user type.

## Your Task

Your task in this assignment is outlined above—develop specific sections of a requirements document for the SRSS and make use of use-cases to help you find the requirements. You will use the template provided on Canvas as a guideline to write up your homework assignment. You are allowed to change/add/remove from the template but remember this template highlights the information you should include when developing a specification document. Your write up must be readable, changeable, and capture all the essential information we have discussed in class. We do expect to see at **least 10** use cases.

You have a requirements document template and use case template available on Canvas for the class.

### Helpful Hints

Do not invent many unneeded requirements. Focus on the core functionality of CRSS and do not add "things that would be nice to have". "Gold plating" the requirements by adding all kinds of nice, but unneeded, functionality and checks will lead to an excessively large and complex document.

Do not try to capture all possible processing scenarios with your use cases. Focus on the main processing to clarify what required functions are needed. This is typically enough for the customer to understand the system and for you to understand the customer's needs. We will discuss use cases during class and you are expected to provide only the textual use cases for this homework. You do not need to provide the graphical use case information.

## Asking Questions

If you have questions about how CRSS should work which is normal, you should document these questions in the section that needs this information in the document. You would use these questions to elicit more information. For this first homework, you will use the information that you have to develop the first draft. I would include a "Questions Section" in the pertinent section of the SRS document as needed to highlight your understanding that you will need additional information.

## Deliverables

You are required to turn in the requirements document and your use cases. **Your documents must be saved as .pdf files.** You can put your use cases in a chapter in your requirements document, or you can submit all of them as a separate pdf document. Create a folder in your team's repo called Homework1 and place all documents in this folder. We will pull your work from GitHub.

## Due Dates

**SRSS requirements specification document and use cases: Sunday, February 9 at 11:55 p.m.**