

Software Engineering Project – Spring 2017

The objective of this project is to practice the Software Engineering concepts and techniques learn in the course. You are given the following description of a software system:

University Conference Center System (UCCS)

A University Conference Center (UCC), which has the mission to foster the organization of conferences for participating businesses, is in need of a University Conference Center System (UCCS) for the management of the UCC. The activity of the system is to automate the organization of conferences, registration, costs and record keeping of the conferences.

Here is a quick description of a participating business. A participating business is a group of people that desires to sponsor a conference and provides a list of guests who will participate in the event.

A typical event consists of several sessions during the day. There will be morning sessions (typically 2 sequential sessions), or afternoon sessions (typically 2 sequential sessions), and/or parallel sessions, either morning or afternoon. When a guest enters the conference, he or she can participate in any session, but not in any special sessions.

A special session is a session that, when available, requires an additional registration and payment. In other words, a guest who enters the conference can participate in any session but not in a special session unless he or she has filled out the registration form for that special session and the registration has been accompanied by its payment.

Each session consists of a series of presentations that requires the use of different types of equipment. For example in a session on “The Latest Discoveries in Software Analysis” a video projector, a set of speakers and a microphone must be available. Each piece of equipment is in operation at the same time. The equipment must be reserved when the session is booked. Qualified personnel are present in each session to report possible equipment problems. When an equipment problem in a session is reported, a UCC specialized technician intervenes to provide technical service to the session and an equipment replacement must be provided. When a piece of equipment is broken it must be brought to repair and it will not be available

for reservation until it has been repaired or replaced.

In addition to allocating each individual session, the UCCS prepares the schedule of the sessions for each conference; the schedule of all the conferences booked for a season; and computes the cost of each conference. The cost of each conference depends on the number of rooms occupied by the conference per session, the number of pieces of equipment used in each session, plus an additional amount of 1% of the conference registration cost for each participating guest used to cover miscellaneous services (ex. Coffee, water, technician support, etc...). The UCC has 3 sizes of rooms, named Roosevelt, Lincoln, and Washington from the smallest to the largest, whose cost per hour is \$300, \$400, and \$500, respectively.

The UCCS also provide registration of guests to a conference. To maximize the profit, more than one conference can be run at a time by the UCC as long as the resources are available.

Sample Output

Here is a sample output of the UCCS.

Conference: Software Engineering

Day 1

Date: Jan 2, 2011

Conference Registration Cost per Guest: \$500

Software Engineering Conference Costs								
Session	Room Name	Room Cost	Video Projector	Speakers	White Board	Microphone	Registered Guests	Misc Services 1%
1stMorning (8:30am-10:30am)	Roosevelt	\$300	\$100	\$50		\$30	40	\$200
2nd Morning (11am-13am)	Washington	\$500	\$100	\$50		\$30	40	\$200
1st Afternoon (2pm-4pm)	Roosevelt	\$300			\$60		40	\$200

2nd Afternoon (4pm-6pm)	Lincoln	\$400	\$100	\$50	\$60	\$30	40	\$200
Special Session (9:00am-noon)	Lincoln	\$400	\$100	\$50	\$60	\$30	10	\$50

page 1

SW & Material

List of SW and material required per group

- A 3 ring binder for the project documentation – bring to class during meeting
- A white plain soft poster board – append on the wall in class
- Git&Github for collaboration
- Any other Software of your choice for requirement collection, SW analysis & Design, and SW development.

Part 1 – Requirement Analysis - Release #1

The objective of this part is to perform *Requirements Elicitation* also called *Requirement Analysis*.

You first release is a *System Charter* for the software system and the *System Function Statement*.

A *System Charter* provides a concise description of the major responsibilities of the system. The details are left for other products of analysis and design (Part #2) and you do not need to cover them now.

- **The system charter is** a bulleted list of statements that describe in plain English everything that your system will be able to do.

Example: if you are asked to design a system that monitors sport competitions, a statement of the system charter will look like this.

- Keep track of teams and their members, including which members are participating in a competition.

A *System Function Statement (SFS)* provides a summary of the use-cases of the system. In order to prepare the SFS, you must first list which **Use-Case diagram(s)** for the specified system. They will be essential for the identification of the system functionalities. This does not require that you model all the use cases that you have identified yet, but that you identify which one must be modeled. After creating all the list of use-case diagrams, prepare the system function statement.

- The **System Function Statement** is a bulleted list of the use cases of the system. Each use case describes some piece of functionality or some action that the system must support.

Example: if you are asked to design a system that monitors sport competitions, a function statement of that system will look like this.

- Register a team in a competition.

The *System Function Statement (SFS)* serves these following purposes:

- It is used throughout the analysis and design (Part 2) to point out needed classes and operations
- It provides a source of test scenarios to validate that the system meets the requirements.
- It becomes a basis for possible “mini” systems that can be developed as prototypes.

Example: if you are asked to design a system that monitors sport competitions, a function statement of that system will look like this.

- Register a team in a competition.

PROGRESS SO FAR: After completing Part 1 you have:

- Started becoming familiar with the domain
- Learned the overall requirements of the system
- Captured the major functionalities of the system

DELIVERABLES – **Release #1**

- A system charter statement
- A system function statement
- Integrate the requirements in the SRS template provided.

You must complete this activity in one week in preparation of Part 2 which covers the Domain Analysis and will be posted next week.

Deadline: 11:59 pm, March 3, 2017.

Important: The releases must be dropped in the appropriately created dropbox under Learn by the deadline. Remember the **heavy late penalties** listed in the Syllabus that **WILL** apply if the Release is late. The penalties will apply to the whole group.

Grading scale:

- (45%) A system charter statement document
- (45%) A system function statement document
- (10%) Integration of the system charter and the system function statement information in the proper part of the SRS template attached to this project.