CS2450 A03 – Project Proposal (Team assignment)

Learning Objective:

- Practice teamwork and problem solving as you put together two projects proposals
- Get to know your team
- Break down the work into tasks of manageable size
- Create a preliminary project schedule

Description:

This assignment needs to be completed as a team.

Put together two project proposals.
I will select one proposal to be developed, reserving the right to require changes if needed.
If you have a preferred project, list it first. Explain why you consider it the better choice for your team.

Brief project overview:
The assignments of the following weeks will ask you to develop a prototype, gather requirements, write a specification, create black-box test cases, UML diagrams etc. Then there will be a 4 week period where you implement and unit-test your project.
Since we have not discussed most of these steps yet, your preliminary schedule should only include the 4 week implementation period.
Before you decide which projects might be right for your team, consider the unique combination of skills and experiences that your group members bring to this project.
Things to consider:

- **Programming skills**
  Exchange your A01 experience to gauge how comfortable / experienced the team members are in this area
- **Additional Software Development Experience**
  See whether you have team members with experience in areas like creating / accessing databases, using web services, networking, programming for mobile devices, etc.
- **Management skills**
  Does anyone on your team have experience in project management (maybe in areas unrelated to software development?)
- **Organizational Skills**
  Who is good at writing schedules, reports, diagrams, … and who is reliable and can make sure things are turned in on time?
- **Programming Language**
  Keep in mind that this is not a programming class and that I won't teach any language specific programming skills. Because of that I strongly recommend that you choose a programming language that is familiar to most of the team members.

- **Complexity**
  In order to get the most out of this class your project should have multiple 'components' that interact with one another (5+ classes, maybe the integration of a web service or a database, etc.)
  The project should be complex enough so that the work can be divided among the members appropriate to their skills. At the same time it should be small enough so that it can be completed within the given time.

For each of the two project proposals you need to do the following:

1. **Describe the project** in one or two paragraphs
   If possible include a drawing or diagram
2. Include a **UML use case diagram** that describes all actors and all major interactions.
   The use case diagram should list at least 3 interactions
3. Divide up the implementation work (4 weeks) into tasks.
   If a task takes longer than a week, divide it up into smaller tasks.
   Create a table of all the **tasks** T1, T2, T3, … .
   Each task should have a clear description that is verifiable
   Each task should including any dependencies, and each task should be verifiable
4. **Create a preliminary 4 week schedule**
   Find a software tool to create a Gantt chart (similar to Figure 23.6 in the book)
   There are a number of free project management tools (more toole) available online. However, for our purpose a diagramming program like for example Visio works just fine.
   Keep in mind that a schedule only has value if it is used on a regular basis. Prefer clarity over excessive detail
5. **Create a staff allocation chart** similar to Figure 23.7 in the book

Turning in:

Select one student who turns in the team assignments for your team.

Create a **text document** (docx or pdf) that includes the following **5 items for each of the proposals**:

1. project description
2. UML use case diagram
3. table of tasks
4. schedule
5. staff allocation chart.

In addition I want you to **lists all team members and a brief description of their contribution** to this assignment.

Turn in this assignment via Canvas