**Week 13 Task List appears at end of e-mail**

**Please Reply this weekend**

We have three weeks left to work on the project, after that we will have to quickly wrap everything up and prepare for the presentation. Let’s meet on Tuesday November 13th at 2:15pm and on Thursday November 15th at 2:15pm, in the computer science lab on both days.

Let’s try to complete the class diagrams for section 5 of the design specification template by this Monday, November 12th at 11:59pm. Assuming we can meet that goal I will be proof reading the design specification template on Tuesday and checking for anything that needs to be changed or is missing. The coding convention document shouldn’t take long to complete. Let’s try to complete that by this Thursday November 15th, at 11:59pm. I would like to submit both the design specification template and the coding convention document to D2L on Friday. I will make a note to do this, but if I don’t then submit what is in the dropbox if we are running out of time to turn the documents in.

When making the object design class diagrams please analyze the motivation for the design patterns closely. The examples given for each design pattern may be nothing like what we are trying to design, but if we look at the general idea for each design pattern we should be able to find a way it can be applied to our system.

Each subsystem has been assigned to a group member for implementation and testing.

We will need to start using version control software to keep the consistency of the system code. Please research the following version control software products and be ready to decide soon which software product you would like to use: CVS, Subversion, or Microsoft Visual Source Safe (available through MSDNAA).

Please make sure you have read chapters 6 - 11. Chapter 11 contains information on testing.

**Required Reply information**

1.  Either an agreement or disagreement with the tasks that have been assigned to you.  If you disagree with the task assignments please provide suggestions for what task assignments need to be changed and how to change them.

2.   An acknowledgement that you understand the tasks that have been assigned to you.

3.  An estimate of how long you think it will take to complete the tasks that have been assigned to you.

4.  The status of what you have worked on/completed in the past week.  If something was not completed by the deadline, why was it not completed?  If a task took too long to complete I need to know so I can divide and schedule the tasks better in the future.

5. We will need to have two group meetings this week to make all the plans that need to be made for implementation, testing, and finishing the detailed design document and coding convention document.  Let’s meet in the computer science lab.  Would the following days and times next week be good for the meetings, Tuesday November 13th at 2:15pm and on Thursday November 15th at 2:15pm?

**Week 13 Task List**

**Minute Taker**

       Submit meeting minutes to D2L before Saturday November 17th 11:59pm. Use meeting minutes template provided in notes and record who attends the meeting in the meeting minutes header.  There is an example of meeting minutes in the power point titled "L8\_Project\_Communication\_ch03\_lect2" on slide 34.   Place the meeting minutes for both Tuesday’s and Thursday’s meeting in our group’s dropbox while you are working on them in addition to posting the minutes to D2L.

**Aaron Bourne**

* Use the class diagrams produced in the analysis object model to apply the following design patterns.  Copy the old class diagrams in the "Class Diagrams Analysis Model" folder of the dropbox (Dropbox\Software Engineering Project\Class Diagrams Analysis Model\) and make new class diagrams in separate documents. . Create classes and relationships between the classes which use the following design patterns:
  + Observer design pattern - very important design pattern
  + strategy design pattern - very important design pattern
  + Template design pattern

Place the class diagrams in section 5 of the design specification template.

Follow all guidelines given in the lecture notes for creating class diagrams.  Some additional information about each design pattern may be given in the textbook. There are some design patterns which are not in the textbook, and only in the lecture notes

* + Create one class diagram for each class of the Notification subsystem, event management subsystem, and event registration subsystem. Place all class diagrams in Section 5 Low Level Design of the Design Specification Template document.
* Finish completing the following sections of the Design Specification Template document: "Goals and Guidelines" (using the goals we identified during the last meeting, "System Integration", "Software Specification", "Hierarchy Chart of Components" (both you and Travis complete this section).
* Complete the following sections of the Coding Convention document: section 4 “Comments”, section 5 “Declarations”, and section 6 “Statements”.
* Start implementing and testing the notification subsystem, event management subsystem, and event registration subsystem. Please test functionality as it is implemented.
  + If a new subsystem is identified please modify subsystem diagrams to include the new subsystem.
  + Do NOT spend time on the boundary use cases, Dr. Fu has not requested that we work on them.

**Travis McMichael**

* Use the class diagrams produced in the analysis object model to apply the following design patterns.  Copy the old class diagrams in the "Class Diagrams Analysis Model" folder of the dropbox (Dropbox\Software Engineering Project\Class Diagrams Analysis Model\) and make new class diagrams in separate documents.  Create classes and relationships between the classes which use the following design patterns:
  + Composite design pattern - very important design pattern
  + Builder design pattern - very important design pattern
  + Proxy design pattern

Place the class diagrams in section 5 of the design specification template.

Follow all guidelines given in the lecture notes for creating class diagrams.  Some additional information about each design pattern may be given in the textbook. There are some design patterns which are not in the textbook, and only in the lecture notes.

* + Create one class diagram for each class of the Volunteer Interface subsystem, the Stakeholder Interface subsystem, and the User Management subsystem. Place all class diagrams in Section 5 Low Level Design of the Design Specification Template document.
* Finish completing the following sections of the Design Specification Template: "Maintainability", "Performance", "Hardware Specification", "Hierarchy Chart of Components" (you and Aaron work on this section), "User Interface Design Overview", and "User Interface Navigation Hierarchy".
* Create user interface navigation hierarchy diagram(s) and place the finished diagram(s) in the User Interface Navigation Hierarchy section of the Design Specification Template document.
* Complete the following sections of the Coding Convention document: section 1 “File names”, section 2 “File organization”, and section 3 “Indentation”.
* Start implementing and testing the volunteer interface subsystem, the stakeholder interface subsystem, and the user management subsystem. Please test functionality as it is implemented.
* Please make the changes to the use case diagram that we have discussed. Place the new use case diagram in the group’s dropbox.
* If a new subsystem is identified please modify subsystem diagrams to include the new subsystem.
* Do NOT spend time on the access control matrix, Dr. Fu has not requested that we work on this.

**Yang Zhao**

* Primary importance: Complete activity diagrams and state-chart diagrams.
* Complete the following sections of the Design Specification Template document that is in the dropbox "Assumptions", "Constraints", "Technical Environment".
* If you finish the above tasks please help with producing the object design class diagrams.

**Owen Burnett**

* **IMPORTANT: Submit the Design Specification Template document and Coding Convention document to D2L on or before Friday November 16th.**
* Submit meeting agendas for Tuesday’s and Thursday’s meeting to D2L.
* Use the class diagrams produced in the analysis object model to apply the following design patterns.  Copy the old class diagrams in the "Class Diagrams Analysis Model" folder of the dropbox (Dropbox\Software Engineering Project\Class Diagrams Analysis Model\) and make new class diagrams in separate documents.  Follow all guidelines given in the lecture notes for creating class diagrams.  Some additional information about each design pattern may be given in the textbook. There are some design patterns which are not in the textbook, and only in the lecture notes. Create classes and relationships between the classes which use the following design patterns:
  + façade design pattern - most important design pattern
  + abstract factory design pattern
  + bridge design pattern
  + command design pattern
  + Create one class diagram for each class of the database subsystem, storage subsystem, and volunteer matcher subsystem. Place all class diagrams in Section 5 Low Level Design of the Design Specification Template document.
* Complete the following sections of the Coding Convention document: section 7 “White space”, section 8 “Naming conventions”, and section 9 “Programming practices”.
* Start implementing and testing the database subsystem, storage subsystem, and volunteer matcher subsystem. Test functionality as it is implemented.