LAB#1: Requirements Elicitation

1. OBJECTIVES

The following objectives are to be completed by the beginning of Lab #2.

1.1 Meet your team members, discuss your individual strengths and interests and select the team leader (to be done by the end of Lab #1. This nomination can be changed in Lab #2 if the team wishes)
1.2 Elicit and document software requirements
1.3 Develop initial Use Case model
1.4 Develop a mockup prototype of the User Interface

2. INTRODUCTION

2.1 In this and the next four lab sessions, you will work as part of a team. Each team will comprise 4-6 members.

2.2 All work products generated in lab sessions must be updated on the team repository regularly. The team leader should ask the lab technician to create a SVN folder for the members so that the lab supervisor can access that folder after the Lab#1.

2.3 During the requirements elicitation, the software requirements are determined and documented in appropriate technical format.

2.4 Use Case diagrams enable the proposed system to be easily visualised and helps in refinement of the requirements.

2.5 UI mockups help in elicitation and refinement of software requirements.

3. PROCEDURE

3.1 Meet your team

3.1.1 Your first task is to meet your software development team. Teams have been previously determined.

3.1.2 Choose a name for your team. Appoint a team leader among your team members. It is the responsibility of the team leader to ensure that there is a fair distribution of work among the team members and that each member **(including the team leader)** is pulling his/her own weight. Note that the team leader can be changed once the team has determined the strengths of each team member. If you choose to change your Team Leader after the Lab 1, please let your lab supervisor and lab technician know.

3.1.3 Follow the instructions provided by the lab technician to fill in the Team Registration Form by the end of the lab session.

3.2 Elicit and document requirements

3.2.1 Determine the target users of your application. Elicit functional and non-functional requirements of the application. Note that in doing this you may wish to have some of the team members act as customers and the rest of the team be the project development team. This is an artificial way of defining the requirements but within the constraints of this course project it is a practical way to proceed.

3.2.2 Document the requirements in appropriate technical format. The requirements should clearly state who performs what system functionality, taking what input and producing what output.

3.2.3 Atomise the requirements such that they are verifiable and traceable. See page 126~130 of Fox for requirements specification heuristics. You will be writing test cases in Lab #4 to verify the requirements. You will also be asked to demonstrate traceability from requirements to the final product.

3.2.4 Document important terms of your application (e.g., user, device, input, output) in a data dictionary. Explain each term with a brief description. Identify attributes of each term and relationships between terms.

3.3 Visualize and refine requirements with Use Case models

3.3.1 From the set of functional requirements, identify the preliminary Use Cases. Depict them on a Use Case diagram using the UML modeling tool.

3.3.2 For each Use Case, start writing the use case description about how the user interacts with the system to carry out the system functionality. As a rule-of-thumb, each Use Case should have a maximum of 6~7 steps in its flow of events. A small Use Case indicates that the functionality has been sliced too finely; a large Use Case can be further broken down.

3.3.3 Iterate over your Use Case model to identify included Use Cases, extended Use Cases, and generalization relationships, if any.

3.4 Develop User Interface Mockups

3.4.1 Develop a UI prototype by sketching a series of screens either using pen and paper or some suitable software tool. The UI design should incorporate good human-computer interaction (HCI) principles. The UI mockups do not need to be working software. They can be hand-drawn.

3.4.2 Use the UI mockups to elicit and refine the requirements.

4. DELIVERABLES

- Documentation of functional and non-functional requirements
- Data dictionary
- Initial Use Case Model, consisting of Use Case diagram and Use Case descriptions
- UI Mockups

Please submit the deliverables to your SVN repository (under the folder “lab1”) before Lab#2 starts. Your Lab Supervisor will want to see and discuss the deliverables with you during Lab #2.

The initial Use Case model will be refined and elaborated in Lab#2.

5. REFERENCES

- Introduction to Software Engineering Design: Processes, Principles and Patterns with UML2 – C. Fox