**ITCS 6112**

**Software System Design and Implementation**

Project Description

The goal of this course is the direct experience with a large-scale project in which the student plays the role of the customer and soft-ware engineer. In this course, students will gain experience in developing and implementing real-word applications. Students will also develop graphical user interfaces for their software systems.

The project is not limited to the coding part. Indeed, students will analyze, design

and implement a software following the phases of the software life cycle: feasibility study, requirements analysis and specification, high-level design, detailed design, coding and module testing, integration and system testing. The products produced in the software process are the following: feasibility study document, requirements specification document, design specification document, programs, system test plan. The evaluation of the project depends on the quality of these products.

Once you determine your group members and project topic, one student from each group will submit the group members and project description.

1. Timeline (All reports are to be submitted as a team and will be assigned a team grade. )

a. One student from each groups should submit to the instructor the project title and the general requirements by Jan 31, 2014.

b. The project report, programs and presentation slides should be submitted to Moodle.

On April 14th, you should also submit to the instructor a hard copy of the whole report and presentation slides.

c. Orally present and demonstrate the software. Presentations will start on April 14. The presentation should take take 15-20 minutes.

2. Project Components

The final project report should include all the following products covering the whole

software development process.

1. Problem definition.
2. Feasibility study document . Clearly state if you are developing a new system, extending or improving an existing one.
3. Software requirements specification document. This document should include:
4. Consistent and unambiguous functional requirements. Give the complete use-case diagram with all the use cases and actors. Define in detail at least five use cases (the most complex ones).
5. System constraints (programming languages, operating system, hardware, databases, Web services, software tools, etc).

4. Design specification document, including:

a. Software architecture.

b. Sequence diagrams.

c. Complete class diagram with all the classes and their relationships.

d. Two examples of object diagrams.

e. Component diagram including the algorithm(s) behind your implementation.

f. Deployment diagram.

5. Technical documentation, including:

a. Programming languages.

b. Reused algorithms and programs.

c. Tools and environments.

d. Database system. List all your tables.

e. Text files. Indicate the formats of the text files used in your code.

6. User documentation including several screen shots of the GUI.

7. Code by including all the necessary libraries. The code is constructed based on the

specification and design documents.

Important Notes:

All the UML diagrams must be legible and produced using a UML tool. You can use Microsoft Visio.