# CS 476: Software Development and Demonstration Requirements

*Dr. Samira Sadaoui*

In this course, students will gain experience in designing and implementing real-world applications through software projects. In addition to technical skills, students will learn managerial and teamwork skills vital for the project's success. Students will develop high-quality Web-based application software (browser-based interfaces) that is user-friendly (with nice GUIs) and distributed. Before starting the project, it is recommneded to split the development tasks among the team members.

## A. Timeline

1. Each team should submit on UR Courses the project title, a problem description that includes two user roles, the main functional and quality requirements for each user role, and the names of the team members, no later than January 20th (max. of two pages).

2. On March 24th, each team should submit on UR Courses the complete and final project report, all the programs and presentation slides.

3. Project presentation and software demonstration will start on March 25th.

## B. Project Report

The final project report should include all the following documents covering the entire software development process. Also, include a cover page with the project title, complete names and e-mails of team members.

1. (2 pts) Problem definition : outline the problem requirements, the application domain and motivations of your project (1 page).

2. (3 pts) Gantt chart for the following tasks: application benefits (deliverable#3), requirements elicitation and specification (deliverables#4 and #5), software design (deliverables#6), coding (deliverables#7 and #8) and testing (deliverables#9).

3. (2+2 pts) Application benefits: highlight the benefits of your application by comparing it to two existing systems. Choose two related systems and include their references/URLs (2 pages).

4. (10 pts) Requirements elicitation:

   (a) (4 pts) List the functional requirements (**only the ones that you have implemented**) for each user role (two exactly). Name each requirement and explain it.

(b) (6 pts) Software qualities: Correctness, Time-efficiency and Robustness. For each user role, include two concrete examples for each quality.

5. (12 pts) Functional requirements specification:

   (a) (3+3 pts) For each user role, provide the use case diagram with all the use cases and actors.

   (b) (3+3 pts) Describe in detail two use cases using the activity diagram. Choose the most complex use cases (those involving multiple secondary use cases).

6. (28 pts) Software design:

   (a) (5+2 pts) MVC architecture according to the selected Web framework. Also, describe two benefits of MVC for your specific application.

   (b) (8+8 pts) Observer and Simple Factory design patterns. Explain in detail the usability of these two patterns for your specific application. Include the complete class diagram for each pattern. For each class, provide the data types of the attributes and prototypes of the methods. Also, provide the algorithms of the methods (not the classes).

   (c) (5 pts) Provide the class diagram of the whole software by incorporating the two design patterns.

7. (27 pts) Software implementation:

   (a) (2 pts) Screenshot of the entire top-level structure of the code within the web framework.

   (b) (3 pts) Deployment diagram regarding the hardware configuration of the code. Indicate the supported Web browsers, the application/Web servers and the database solution.

   (c) (17 pts) GitHub link of the entire program. All students must contribute equally to the programming part. The commit log of each student will be checked within GitHub.

   (d) (5pts) The URL of your Web-based application (deployment). The application should be accessible online and runnable.

8. (4 pts) Technical documentation:

   (a) (1 pt) List of programming languages.

   (b) (1 pt) List of reused algorithms and small pieces of code. Include their sources/URLs.

   (c) (2 pts) List of software tools and environments. Provide briefly their benefits specifically for your application.

9. (10 pts) Software testing: **The tested use cases must be different across all the test cases; each use case can only be tested once.**

(a) Correctness testing using four test cases (screenshots of both inputs and outputs).

(b) Robustness testing using four test cases (screenshots of both inputs and outputs).

(c) Time-efficiency testing of two functions (with screenshots). Indicate the method you used to measure the time.

## C. Project Presentation

For the project presentation, please focus only on the following points:

1. The two use case diagrams within a UML tool. Also, show the UML tool you have used.
2. The MVC architecture. Also, include the entire structure of the MVC code within the Web framework.
3. The two design patterns. Explain precisely their usability for your specific application. Also, include the entire code of each design pattern.
4. Demonstrate the software behaviour for both user roles.

## D. Important Notes

1. All the UML diagrams must be readable and produced with a UML tool, such as starUML, ArgoUML, Visual Paradigm or Microsoft Visio.
2. The design patterns must be designed according to the lecture notes (GoF book) and implemented from scratch.
3. The code is constructed based on the specification and design documents. It should contain the MVC architecture and the two design patterns.
4. The application must be a website (browser-based) and not an app. Website builders, such as Wordpress, are not allowed.
5. Student attendance is mandatory for all the presentations.