CS 476: Software Development and Demonstration Requirements

Dr. Samira Sadaoui, Ali Bayeh

Through the software projects, students will gain experience in designing and implementing real-world applications. In addition to development skills, students will learn managerial skills vital for the project success. Students will develop high-quality application software that is Web-based (browser-based interfaces), user-friendly (with nice GUIs) and distributed on three tiers. Before starting the project, the development tasks must be split among the team members.

A. Timeline

1. The leader of each team should submit on UR Courses the project title, a well written problem description, including two user roles, the main functional and quality requirements, and names of the team members, no later than January 22nd (max. of two pages).
2. On March 27th, the leader of 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 28th. The presentation takes approx. 15 minutes.

B. Project Report

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

1. (2 pts) Problem definition: outline the problem requirements and include the application domain and motivations of your project (1 page).
2. (4 pts) Application benefits (2 pages): what are the benefits of your application when compared to existing systems. Choose two systems only and include their references.
3. (22 pts) Requirements elicitation and specification:

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

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

c. (3+3 pts) Describe in detail two use cases using the activity diagram. Choose the most complex use cases.

d. (6 pts) Software qualities: Correctness, Time-efficiency and Robustness. Include at least two concrete examples for each quality for each user role.

4. (23 pts) Top-level and low-level software design:

   a. (5+3 pts) Provide the MVC architecture according to the selected Web framework. Also, describe at least three benefits of using MVC for your application.

   b. (5+5 pts) Observer and 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.

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

5. (34 pts) Software construction:

   a. (3+3 pts) Submit the entire code for observer and factory patterns.

   b. (2 pts) Submit the screenshot of the entire structure of the code within the web framework (only the structure of the code).

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

   d. (2 pts) Screenshots of all table contents of the system data.

   e. (16 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. Website builders, such as WordPress, are not allowed.

   f. (5 pts) Link of your Web-based application. To be evaluated, the application should be accessible online and runnable.

6. (5 pts) Technical documentation:

   a. (1 pt) List of programming languages.

   b. (1 pt) List of reused algorithms and small programs. Include their references.

    c. (3 pts) List of software tools and environments. Provide briefly their benefits specifically for your application.

7. (10 pts) Acceptance testing: select test cases for both user roles. All the tested use cases must be different; a use case cannot be used more than once.
    a. Correctness testing using four test cases only (screenshots of both inputs and outputs).
    b. Robustness testing using four test cases only (screenshots of both inputs and outputs).
    c. Time-efficiency testing of two functions only (with screenshots). Indicate the method you used to measure the time.

C. Project Presentation

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

1. Demonstrate the software behavior 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 GoF book.
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 web application (browser-based) and NOT a mobile or desktop application.
5. Only the team leader submits the required documents (report, slides and programs). Submit the entire code on UR Courses.
6. Student attendance is mandatory for all the presentations.