EGE UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT
501003362023 FUNDAMENTALS OF SOFTWARE ENGINEERING

Spring 2024
TERM PROJECT ASSIGNMENT

Objectives

The assignment's objective is to gain experience in software engineering practices.

Project

- The project will be chosen by each group from one of the project descriptions in the attached “Project Topics" document. The project topic that is chosen **must be** submitted under the heading **Project Topic Selection** on the **EGEDERS** page of the course by **04.03.2024** by each team. The project will be undertaken in teams consisting of **four** members. It is highly recommended that all team members belong to the same lab session.
- **Participation in the project is mandatory for all students enrolled in the course.** Neither re-enrollment in the course nor the absence of compulsory attendance alters this requirement.

Expected Work

The teams should apply waterfall process model. They are free to use any tool or report format that may be preferred for its being very suitable for the teams’ specific task. For example, teams may add a *data flow diagram* or even a *flowchart* to their analysis or design if they think they are necessary to get the job done. “*Getting the job done*” is however a strict requirement and it is expected from the team as a pay-back of the freedom that they have been granted. You will frequently find the instructors act as customers who even do not know what exactly they should expect from the project, or in general from software or computers. On the other hand, the team should be motivated to discover and demonstrate what the project may do for the end users. The teams must perform all kinds of research including moving to the field and performing interviews with the experts and potential users in the analysis phase of the project. Teams should submit either videos of the interviews with domain experts and potential users in CDs, or the telephone numbers and email addresses of the domain experts along with the requirements taken from them. Competition and *confidentiality between the teams* is encouraged as well as cooperation *and participation within the teams*.

Assessment of Project

Your work will be graded mainly based on your domain analysis and the quality of your prototype. Therefore, you are strongly advised to explore the domain deeply and come up with a competitive design that provides the best possible automation. *Getting the job done,* although crucial, is not enough for good grades. A team with a superior design will receive better grades. On the other hand, if you cannot come up with profound solutions to the key problems, you are not likely to get very good grades, no matter how beautiful (or long) your reports may be.

- Every report must include a proper introduction and conclusion, *relevant to the report content*.
- Report format and neatness must be acceptable as a *professional artifact*.

The first work expected from the teams are:

REQUIREMENTS (ANALYSIS) REPORT (25.03.2024)

1) Introduction
2) Identification of Viewpoints
   Principal Viewpoints of the System
   Description of each Viewpoint
   Viewpoint Hierarchy Diagram
3) Requirements Definition
   - Definition of requirements of each viewpoint
4) Requirements Classification (considering functionality)
   Functional Requirements
   Non-functional Requirements
   Domain Requirements
5) Requirements Classification (considering lifetime)
   Volatile Requirements
   Enduring Requirements

* Requirements identified in step 3 are to be classified in steps 4 and 5. Please utilize the provided template below for these steps, from step 3 through step 5.

ID	**Viewpoint**	**Definition of Requirement**	**Classification (considering functionality)**	**Classification (considering lifetime)**

6) [*OPTIONAL*] Requirements interview (online) with stakeholders (you should either record the interview on CD or put a section for the list of requirements taken from each expert along with his/her email and telephone number into your report)
7) Requirements Prioritization and Negotiation
8) Requirements Traceability Matrix
9) Use Cases of the Main Scenarios
   - Three primary use cases should suffice. Ensure that these use cases are substantial and integral to your system, avoiding trivial functions like "login" and "register." Focus on the core functionalities that define your system's primary purpose.

    - Use cases should only contain these sections: inputs, actors, action, outputs, and exceptions.

10) Domain Model as a UML diagram (Note that the domain model and class diagram are distinct UML diagrams. For this report, we specifically require the domain model and not the class diagram.)
11) Conclusion

PROTOTYPE PRESENTATION (13.05.2024)

The teams are required to detail the architecture of their system and present the culmination of their design efforts and prototype implementation (demo) as a short (no more than 15 min.) presentation and demonstration. This comprehensive presentation will encompass an explanation of the system's architecture, an overview of critical UI design choices, testing strategies and a demonstration of the prototype. We expect to be able to understand your system design clearly from your presentation, so you should include anything you think is necessary to achieve this.

Architecture Overview

- Describe your system's architecture. Specify if your design follows one of the established generic architectural patterns or if a new hybrid model has been developed. Highlight the main components of your architecture, the subsystems within these components, and provide concise explanations for each. Discuss how components and subsystems are interconnected.
- Illustrate your system's architecture with a block diagram, showcasing components, subsystems, and their relationships.
- Highlight the connection between your architectural design and its implementation, showing how the theoretical design becomes the practical system.

UI Design Overview

- Explain what criteria are taken into consideration in terms of *UI design principles*, *information presentation* and *interaction styles* during the design of the UI.
- Provide examples from your implementation, supplemented with corresponding screenshots, for each criterion.

Testing Overview

- Include the test cases (development or release testing) you have taken into consideration.
- Ensure that test cases are related to the specified use case(s) in the project topics document.
- Provide an overview of your test cases, detailing the elements being evaluated, the inputs used, the expected outcomes, and the resulting outputs.
- Incorporate relevant code snippets utilized during the testing process into your presentation.

Prototype Demonstration

- The prototype should cover the main use cases of your project and indicate how the user interface will function.
- Implement all aspects of the user interface.
- Implement only the use case specified in the project topics document. (It's not necessary to implement use cases like login, register, etc.)
- Provide a clear message, such as "This function has not been implemented yet," for any unaddressed use cases within the prototype.

Team Work

- The project will be undertaken in teams consisting of **four (4)** members. Students are allowed to choose one team member on their own. Subsequently, these pairs of selected individuals will be randomly combined to form full teams consisting of four members.
- Members are strongly encouraged to attend the same lab session to facilitate teamwork. However, should there be a substantial imbalance in student distribution among lab sessions, flexible arrangements will be offered. Group and/or team changes are not tolerated.
- Teams are encouraged to decide on their *division of labor* and *decision taking mechanism* as soon as possible. They may elect a team leader or may choose to employ more democratic ways of taking decisions.
- When the reports are handed in, they should be accompanied by a statement, signed by the team members, stating the percentage contribution of each team member. For example, in an equally contributing team of four, all contributions would be labeled 25%. Unequal contributions should be clearly stated and agreed so that marks can be adjusted.
- If any disagreements or challenges arise within the team regarding any aspect of the project, it is crucial to bring these to the attention of your lab instructor immediately, and certainly before the announcement of grades. Furthermore, a signed statement from all team members detailing the specific issue must be submitted.
- Group work should be the result of collaboration only within the group.

Late Submissions

Late work will receive the following penalty:

- 1 day late – 10% penalty; more than 1 day late – 10% per day penalty.