CS130: Project Part B Requirements

Due: **2/16/2023, 11:59pm PST – Week 6**

Objective

The objective of Part B is to analyze the functional and non-functional requirements of your application, choose an architecture, develop a design and report on your implementation status.

Submission Instructions

Please submit the report (as a single PDF file) on **Gradescope** before the deadline. Only one person from each team should do the submission.

Report Structure

Your report should include the following sections:

- **Title Page:**
    - Team Name
    - Project Name
    - Team Members (names and UIDs)
    - GitHub Repository URL(s)

- **Analysis Description**
    - Analyze the functional requirements of your application and capture it with UML Diagrams. **Please accompany each diagram with a brief textual caption that describes its purpose.** Include **the following UML diagrams** in your report.
        - Describe the use cases of your application and the actors and external systems involved in them with a **Use Case Diagram.** Please include one use case diagram for the entire application.
        - Describe the actions of each actor and system and the control flow between them for your most important use cases with **Activity Diagrams.** Please include one or two activity diagrams showing your most important components in your application.
        - Describe the classes and data structures of your application and their attributes, relations and operations with a **Class Diagram.** Please include one class diagram for the entire application.
        - Describe some scenarios of the identified use cases involving the actors and classes of your application with **Sequence Diagrams.** Please include one or two sequence diagrams showing your most important components in your application.

        - Describe the behavior of your most complex classes (of the system as a whole) and how it responds to events with **State Machine Diagrams.** Please include one or two state machine diagrams showing your most important components in your application.
    - Explain how your diagrams correspond to the functional requirements in your application.

- **Architecture Description**
    - You learned in class several architectural patterns for structuring the landscape of your application, the structure of its components, and the user interface. Analyze and document the non-functional requirements (performance, scalability, availability etc.) of your application, describe the components of your architecture, and the architecture patterns you will organize them after and why (including how this addresses the needs of your application). This should be done at the three levels of architecture (if applicable):
        - Across components (application landscape patterns)
        - Within components (application structure patterns)
        - For the UI components (user interface patterns)
    - Create an architecture diagram for your application showing your landscape architecture (and other architectural patterns you may have used).

- **Design Description**
    - Describe the design of your (landscape level) applications, including their structure and behavior. Use design-level UML diagrams to communicate this. Highlight any GoF (or other) patterns you may have used.
    - Describe the motivations of your design choices and why they respond to the requirements that you identified and documented during analysis.

- **Implementation Status**
    - Describe where you are in implementing your application's design. Include a list of links to epics and use stories that you have already completed (broken by sprint) and the ones that are still in the backlog. Comment on the successes and challenges of using Scrum that you faced.
    - Describe the implementation status (% of completion) of your components and point out their locations in your repository.
    - Highlight any libraries, frameworks, or over-the-shelf components that you may have chosen to use when realizing your application. Describe the tradeoffs you considered when you chose them.
    - Describe any special techniques (e.g., code generation, test driven development, CI/CD, etc.) you may have used in the course of your implementation.
    - Include mock-up (or real) screenshots (if applicable) of your application's user interface to give us an idea of what it will look like. If your proposed software does not have a graphical interface, describe how a user would use the software.

For diagrams, you can use tools such as https://plantuml.com/, https://www.draw.io/, or any other tool.

Github

You are expected to use the Scrum process in the course of this project. Your Github repository should reflect that. So, we expect to see

- Your sprints defined with Github milestones with the relevant stories assigned to them.
- Your Scrum board is defined as a Github project and used to triage the issues and assign them to the sprints and to your team members.
- Your epic and stories described using the scrum templates.
- Your commits are associated with pull requests, linked to issues, and peer reviewed.

Presentation

Along with the report, create slides for a **5-minute** presentation that your team would give during the discussion section in **Week 6**. This presentation should explain your idea, show your UI mockups, and discuss the highlights of your architecture, design, and implementation progress. After each presentation, you can take feedback and questions from the **TA and other teams**.

Part B Checklist

- ❏ Project Report (submitted on Gradescope)
    - ❏ Title Page
    - ❏ Analysis description
    - ❏ Architecture description
    - ❏ Design description
    - ❏ Implementation status
- ❏ Evidence of following the process in Github
- ❏ Slides to be presented during lab
    - ❏ UI Mockups
    - ❏ Highlights of the Report contents

Report Length: The exact number of pages in the report may vary due to the size and number of diagrams, Use **10 pages as guidance**, and focus on the quality rather than the quantity.