[CS304] Team Project - Sprint 1 (15 points)

> Presentation: week 9 lab session (no rearrangement)
>
> Report submission: Before presentation. 40% penalty for late submission.

Part I. Architecture & UI Design (5 points)

Please write a report that includes the following information.

1. Architectural Design

Please describe the software architecture of your team project. The description should include:

- Diagrams: any type of diagram is okay, as long as it clearly illustrates the architecture (e.g., components, role of each component, the interactions between components, etc.)
- Natural language description: necessary explanations of the architecture (e.g., why you use this architecture, descriptions of the diagrams, etc.).

2. UI Design

Before you start implementing UI, it's better to have a concrete UI design to guide implementation and facilitate communication.

Please provide the UI design for **primary user interfaces** for your system, e.g., the webpages for primary features of the system, the UI for IDE, the UI for each level of the game, etc.

You DON'T have to provide UI design for common interfaces like user login page. You should provide UI design for the primary user interfaces with respect to the notable features of your own team project.

You should include images as the UI design output in the report. You could use any tools for this purpose (you could even draw sketches by hand). Yet, we recommend you to explore **wireframe tools** that are widely used by professional UI/UX designers.

Part II. Process & Collaborations (5 points)

1. Github Project board for the 2nd sprint

Based on the current progress, update the **GitHub Projects** board for the second sprint (week 15/16). Specifically, you may update the user stories, tasks, priority, assignees, start time and duration to guide the next sprint.

2. Git collaboration

Please demonstrate how your team collaborate using `git`. We'll check the commit distribution and branches on GitHub. In particular, on your GitHub team repo page, click `Insights -> Contributors`, you could see the commit stats for all team members; click `Insights -> Network`, you could see the commit history in branches.

Please follow git best practices. Bad practices (e.g., only 1 branch, only 1 giant commit, only 1 contributor, meaningless commit messages) will incur score penalties.

Part III. Demo & Presentation (5 points)

Basic requirements (up to 50% points will be deducted if failing these requirements):

- Each team will give a 10-minute presentation **in the lab session of week 9**.
- Every team member needs to show up during the presentation.

Please prepare a slide that includes the following content:

- **Retrospective**: Review the events and activities of the first sprint. Review on what went well, what didn't, recurring issues, etc. Decide specific actions or improvements to implement in the next sprint.
- **Demo & Deliverables**: a demo of the current implementation (i.e., what your team will deliver to the customers at the end of this sprint). We will also evaluate your progress based on the demo.
- **Contribution for each team member**: if not specified, we assume that members have equal contribution. Please report contribution as it is, and **it cannot be adjusted once the score is released**.

Submissions

For this milestone, you should have made some coding progress, and **we'll check your team repo**. Other submissions should all be made on GitHub Classroom.

1. Submitting the design report (Part I) and the presentation slides (Part III)

The design document and slides should be submitted by each team to your team repo.

- Upload two documents, with the name `design-teamID.md` and `sprint1-teamID.pdf`, to the team repo.
- The file format of design document should be `.md`, and `pdf` for slides. Other file format such as `.docx`, `.txt` will NOT be accepted.

2. GitHub Projects & Insights (Part II)

The **GitHub Projects** board should be updated in your team project repo. We'll directly check the `GitHub Projects` board and the `Insights` page of your team repo. You don't have to submit additional report for this part.

AI Usage Annotations

As AI is reshaping the software engineering practice, you are allowed to leverage AI through the development process of the team projects.

However, for any code, script, configurations, documentation, ideas, and artifacts generated or influenced by AI, you must include clear and concise references alongside the AI-generated entity, stating clearly the AI tools, versions, models, prompt, means of usage, and any adaptation you did for this entity.

Failures to do so will be considered as violations of academic integrity and could incur 100% score penalty for this milestone.

Before any content generated by AI, you should add the following information with the exact format.

```
AI-generated-content
tool: xxx
version: xxx
usage: xxx
```

Below is an example of how you should describe your AI usage in Java.

```java
public class FibonacciExample {
    /**
     * AI-generated-content
     * tool: Copilot
     * version: latest
     * usage: I selected the fibonacci method and ask Copilot to generate a main method for it.
     * I slightly adapt the generated code by modifying the value of n.
     */
    public static void main(String[] args) {
        int n = 10;

        for (int i = 0; i < n; i++) {
            System.out.print(fibonacci(i) + " ");
        }
    }

    /**
     * AI-generated-content
     * tool: ChatGPT
     * version: 3.5
     * usage: I used the prompt "generate a fibonacci Java program", and
     * directly copy the code from its response
     */
    public static int fibonacci(int n) {
        if (n <= 1) {
            return n;
        } else {
            return fibonacci(n - 1) + fibonacci(n - 2);
        }
    }
}
```

You could use **comment** to add such description to any code, configurations (e.g., in `xml` format), and documentation (e.g., in `markdown` format), which all support comment. You must use **the exact string "AI-generated-content"** to start the description.

You don't have to add the comments if you didn't use AI in the code. However, any code without such comments, if detected to be similar to other code during duplication detection, will be treated as plagiarism.