

[CS304] 团队项目 - Sprint 1 (10 分)

展示时间：第 9 周实验课（不得更改时间）

第一部分：架构与 UI 设计 (2 分)

请撰写一份报告，内容包括以下信息：

1. 软件架构设计

请描述你们团队项目的软件架构。描述应包括：

- **图示：**任何类型的图示都可以，只要能够清楚地展示架构（例如：组件、各组件的角色、组件间的交互等）。
- **自然语言描述：**对架构的必要解释（例如：为什么选择该架构、图中未展示的隐藏假设等）。

2. UI 设计

在开始实现 UI 之前，最好有一个具体的 UI 设计，以指导实现并促进团队间的沟通。

请提供系统主要用户界面的 UI 设计，例如系统主要功能对应的网页界面、IDE 界面、游戏每一关的界面等。

- **不需要**提供通用界面（例如用户登录页面）的设计。
- **应当**提供项目特色功能的主要用户界面的 UI 设计。
- 请在报告中附上 UI 设计的图片。你可以使用任何工具（即使是手绘草图也可以）。不过，我们建议你使用专业 UI/UX 设计师常用的线框图工具（wireframe 工具）。
- 请注意：UI 设计不同于 UI 实现，两者将分别评分。即使你已经实现了 UI，也需要单独提交 UI 设计。**不得将 UI 实现的截图当作 UI 设计提交。**

第二部分：开发过程与协作 (2 分)

1. 第二次冲刺 (Sprint 2) 的 GitHub 项目板

根据当前进度，更新 GitHub 项目板以规划第二次冲刺（第 15/16 周）。具体来说，你需要更新：

- 用户故事
- 任务
- 优先级
- 任务分配人员
- 开始时间与持续时间

以指导下一阶段的开发。

2. Git 协作

请展示你们团队如何通过 Git 协作开发。我们将检查你们在 GitHub 上的提交分布和分支管理情况。具体操作如下：

- 进入团队 GitHub 仓库页面，点击 **Insights -> Contributors**，可以查看每位成员的提交统计；
- 点击 **Insights -> Network**，可以查看各个分支的提交历史。

请遵循 Git 最佳实践。不良实践（例如：仅有一个分支、只有一个巨大的提交、只有一个贡献者、提交信息没有意义）将被扣分。

第三部分：冲刺评审（Sprint Review）（5 分）

现在是你们团队向客户和其他利益相关者展示成果的时刻！这是你们展示已完成工作的机会。

虽然你可以简要介绍系统架构和 UI 设计，但请记住，客户最关心的是一个**可运行的产品**。请运行项目，展示已实现的功能，并重点说明这些功能如何满足用户需求。

请注意，利益相关者期望项目稳定推进。如果你的进度落后，可能会让他们产生质疑。例如，如果你计划交付 5 个重要功能，但到 Sprint 1 结束时只能展示一个功能，那么可能会被质疑是否能按时完成全部需求。

Sprint 1 评审安排：

- 展示将在**第 9 周的实验课**进行；
- 每个团队将有 **10 分钟**的展示时间；
- 可以准备幻灯片，但**不是强制**；
- **所有团队成员必须到场**，缺席者将**不予评分**；
- **踢出成员**：如果你想要移除某位团队成员，必须在评审中说明原因。

第四部分：AI 使用情况（1 分）

1. 设计阶段的 AI 使用

请填写一份调查，报告你们在**架构设计**和**UI 设计**中使用 AI 的情况。**每个团队提交一次即可**。

2. 编码阶段的 AI 使用

对于任何由 AI 生成的代码，请在代码中添加以下格式的信息：

```
makefile复制编辑AI-generated-content
tool: xxx
version: xxx
usage: xxx
```

下面是一个 Java 示例，展示了如何描述 AI 的使用：

```
java复制编辑public class FibonacciExample {
    /**
     * AI-generated-content
```

```

* tool: Copilot
* version: latest
* usage: 我选择了 fibonacci 方法，并让 Copilot 为它生成 main 方法。之后我稍作修改，更改了 n 的值。
*/
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: 我使用了提示语“生成一个 Java 的斐波那契程序”，并直接复制了它的回答。
*/
public static int fibonacci(int n) {
    if (n <= 1) {
        return n;
    } else {
        return fibonacci(n - 1) + fibonacci(n - 2);
    }
}
}

```

你可以将此类注释添加到代码、配置文件（如 XML）和文档（如 markdown）中。这些格式都支持注释。必须使用准确的字符串“AI-generated-content”开头。

如果没有使用 AI，则不必添加这些注释。但如果你未添加注释而代码又被检测为与他人高度相似，将会被视为抄袭，并扣除 100% 成绩。

提交要求

1. 设计文档提交（第一部分）

- 设计文档由团队提交至你们的 GitHub 项目仓库；
- 命名为：**design-teamID.md**（将 teamID 替换为你们的团队编号）；
- 文件格式必须是 .md；
- 其他格式（如 .docx、.txt）**将不被接受**。

2. GitHub 项目板、Insights 页面与代码（第二、三部分）

- GitHub 项目板需在你们的项目仓库中更新；
- 我们会**直接检查**你们的 GitHub 项目板、Insights 页面以及代码提交历史；
- **无需提交额外的报告**。

3. AI 使用调查（第四部分）

- 请提交 AI 使用调查表（团队只需提交一次）。