1. **Gather requirements and initial information**

This project requires that the lead and the designer to make a driver for the rest of the team and work on the first two steps of the software design document. Additionally, the designer must create the class diagrams for the project. The SWE will code the even number problems while the tester will code the odd numbered problems. The SWE should write AAA test in English for the problems the Testers codes. While the Tester writes AAA tests in English for the problems the SWE coded. Since only Java’s collections classes can be used when coding these problems, the SWE and the Tester must code in Java. All team members should commit and push their work to the repo, through issuing pull requests that the lead is responsible for verifying. The lead is Kylan, the Designer is Taylor, the SWE is DeAnna, and the Tester is Brandon.

1. **Outline the system architecture**

The system will be built in Java with each problem implemented as its own class file, and a single Driver class will serve as the entry point to run all problem solutions. The Software Engineer (DeAnna) will code the even-numbered problems, while the Tester (Brandon) will code the odd-numbered problems, and both will follow a consistent structure (such as including a run() method) so the Driver can easily call their solutions. The Designer (Taylor) will create a Visio class diagram showing the Driver and problem classes, while also working with the Lead (Kylan) to ensure the Driver matches the design. To support testing, the SWE will write AAA-style (Arrange-Act-Assert) test cases in English for the Tester’s problems, and the Tester will write AAA tests for the SWE’s problems. All code, diagrams, and documents will be pushed to the shared GitHub repository through pull requests, which the Lead will review and approve to maintain project stability.

A diagram of a process

AI-generated content may be incorrect.