MILESTONE 2 -- SFT221 Scrum Report and Reflection

All students are expected to attend the SCRUM meetings and to participate. Failure to do so will result in greatly reduced grades.

GROUP:

Members Present:

1. Tzu Han Chao	4. Hao Han Kao
2. Hanbi Gong	5.
3. Afra Alam	6.

Milestone 2 Tasks

Some of the software for the project has already been written for you and is available on Blackboard. You must use this in your project and every team should add it to the source code for their repository. Anything in the main function is simply for demonstration purposes and can be replaced. The software you are being given has not been tested and you will need to test it.

You need to study the problem and the code provided for you and then:

- Add any new data structures you will require This will require a thorough analysis of the problem and the existing software. This should be done by creating a new header file in the directory where the rest of the source code has been placed. You do not want to go back and modify it later if you can avoid it as it will slow the project.
- Create a test plan for the project by replacing the text in the supplied test plan template with your test plan.

Deliverables due 4 days after your lab day:

- An analysis of the problem (no written artifacts produced).
- A series of data structures created as header files and stored in the repository.
- A test plan stored in the repository.
- Completed scrum report including reflection questions answered.

Rubric

Individual	Group participation (includes GitHub commits and Jira usage)	80%
	Teamwork	20%
Group	Data structures (complete, correct, and well-designed, & project updated)	25%
	Test plan (complete, well-written)	25%
	Git usage (used properly with good structure)	10%
	Jira usage (creates issues, tracks progress)	20%
	Scrum report & reflections	20%

Deadline	20% deduction for each day you are late	

Scrum Report

Summary of Tasks Completed or Delayed in the last week:

Here you can list all the tasks completed in the last week along with any tasks which could not be completed with a reason why they could not be completed.

Member	Tasks Completed	Tasks Delayed/Blocked
Tzu Han Chao	Git Repository Setup	
Hao Han Kao	Jira Creation	
Hanbi Gong	Scrum Report	
Afra Alam	Reflections	

For every task delayed or blocked, describe the reason for the delay or block, how it impacts the project and the proposed solution or workaround.

Delayed or Blocked Task	**n/a**
Reason for delay or block	**n/a**
Impact on Project	**n/a**
Solution or work-around	**n/a**
Delayed or Blocked Task	**n/a**
Reason for delay or block	**n/a**
Impact on Project	**n/a**
Solution or work-around	**n/a**

Summary of Meeting:

A summary of the main points discussed in the meeting and the outcomes of the discussions.

Topic	Discussion Summary	Outcome
Task Assignment	**Dividing milestone tasks to each member in group**	**Members know what to do for MS2**
Testing Software	**Some members use Mac and do not have access to Visual Studio so we discussed alternatives**	**Group 6 now uses Google Test for building tests**

Jira Layout	**Group 6's Jira layout has been cleaned up and given more tasks to reflect the current milestone**	**Jira looks much nicer and everyone is not confused with the changes**

Summary of Decisions Made:

This will include major architecture and design decisions, testing decisions, prioritization of tasks, dealing with problems encountered and other major outcomes from the meeting.

Decision	Rationale
Testing via Google Test instead of Visual Studio	Some members own a mac and do not have access to Visual Studio. For consistency and with the professor's permission, group 6 decided to use Google Test. This will affect subsequent Milestones and the test files may have slightly different syntax as a result.
Tzu-Han works on code for testing	Tzu-Han will upload all the files that need to be tested which includes three source files and one header file. She is responsible for checking through everything.
Hanbi and Hao Han works on Test Plan	Hanbi and Hao Han will work on the test plan together since the group noticed that it seemed kind of tedious. One of them should upload the final file onto Git.
Afra works on Scrum Report and Reflection	Afra will work on the Scrum Report as well as the Reflections. She decided to do both alone because they didn't seem like much work compared to the test plan.

Tasks Attempted During Meeting:

Each member is assumed to participate in the SCRUM meeting and contribute to the completion of the SCRUM report and reflections. Since the SCRUM meeting will not take more than 20-30 minutes, there is lots of time left to undertake some of the actual work tasks. In the table below, each member should list what they did to complete the SCRUM report, the reflections, and 1-4 other tasks they completed during the class period. If a task cannot be completed, the student should indicate why this was not possible.

Member	Task Attempted	Time Spent	Complete?
Afra Alam	**Jira Tasks Setup**	**15 min**	**Yes**
Tzu Han Chao	**Git Repository Setup for MS2**	**20 min**	**Yes**
Hanbi Gong	**Test Case Sheet**	**15 min**	**No**

Hao Han Kao	**Test Case Sheet**	**15 min**	**No**

SCRUM Tasks Selected for Next Week:

The tasks each member has selected to pursue for this class or the next week.

Group Member	Task Description
Afra Alam	Reflection Questions and Scrum Report
Tzu-Han Chao	Git management
Hanbi Gong	Test plan
Hao Han Kao	Test plan

Major Outcomes of Meeting:

This is where you should highlight the major accomplishments of the class.

Outcome	Impact on Project
Some of the group members cannot access Visual Studio on their macbooks, so we decided on Google Test instead	**The project should still be able to be completed, however some of the syntax on the test project files will differ due to platform differences.**
Milestone 2 tasks created on Jira	**Milestone 2 progress can be tracked easily**
m2 branch created on git repository	**Work for m2 can be submitted on that branch before being merged with main for security's sake**

Things That Went Well in This Meeting:

Here you can highlight things which worked well. This indicates that the way you worked on these items is working and should be continued.

Topic/Work Item	Reason for Success
Everything	**Group was able to discuss what they wanted and stay on track with the project. Everyone is on the same page.**

Things That Did NOT Go Well in This Meeting:

This is where you can list things which did not go well in the class. You should analyze why this happened and suggest how you can improve it next time. This will lead to the goal of *continuous process improvement*.

Topic/Work Item	Reason for Problem and How to do Better
n/a	**n/a**

Reflection Questions:

Answer the following questions using your own words. Make sure that each answer comprises a minimum of 100 words.

1. In this milestone you have been asked to analyze a problem and design software (functions) to complete the solution without writing the software.
    a. Is this process more difficult than just writing the software to complete the project? If so, why is it more difficult? If not, why is it easier than just writing the software?

        This process is kind of more difficult than just writing the software straight on because we had to read the whole program and understand what was missing to complete the program. If we were to write the functions from scratch, we would be able to design the software however we want based on what was required from the question. But it was also kind of easier than writing the program from scratch because we just had to write the function definitions. Overall, we thought it was kind of difficult because we had to really understand the problem and what the code is supposed to do in order to create a software properly.

    b. Describe two advantages of developing software in this manner rather than just moving on to writing the functions without writing specifications first.

        Two advantages of designing software without actually writing functions is ease of accessibility and productivity. If we were to write the code while developing the software, it would take us at least twice as long to get everything done on top of making sure that everything worked. Building functions from scratch without having any specifications on hand means that we would have to think of what the client would need out of a software. Designing a software by analyzing the problem is good because the developers can quickly figure out a solution without worrying about the right syntax as they would if they just went on to build the functions. This, in turn, means that the

software would theoretically be produced faster.

2. Why is it a good idea to create a test plan? Describe at least 3 advantages of test plans.

   The benefits of writing a test plan are that it describes the scope of a test, explains what resources the team has and provides transparency. Creating a test plan analyzes the software. This means that the developers can determine what kind of trials and tribulations the software can go through in the testing phase. Test plans can also describe what sort of tools and resources the team can use, so that the testing phase can be as efficient as possible and the people who are testing the software won't be in the dark. Test plans also provide transparency because it easily documents and describes what the developers would be going through for the client or company.

3. Describe the process you used to analyze and understand the existing software.

   We were able to understand the existing software by looking at the code and the problem. First, we read through the document that was given to us to know what kind of software we were supposed to be building. In this case, we're supposed to build a pathfinder for some trucks. Then, we looked at the bits of code that were given to us. The professor gave us a hint that there was a header function missing as well as the function definition for some of the functions, so we read through the code to see what was missing. Understanding the logic of what was happening inside the code on a line-by-line basis is important for this. After that, we built our test plan based on what code we came up with and the requirements stated by the problem.