Milestone 5 Scrum Report

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

GROUP: **10**

Members Present:

1. Duong Truong Phuc Nguyen	4.Syed Abdullah
2.Huynh Huy Hoang	5. Jasmin Aro
3.Ahnaf Tahmid Khan	6.

Milestone 5 Tasks

In this milestone, you should write, implement, and execute integration tests. Integration tests test how multiple functions work together to complete a task. Depending on what is being tested, you might be able to write unit tests to do the testing and automatically compare the results. In other cases, you might need to manually check the output to check it. This will all be stated in the tests where it discusses how they should be run.

As you update the function-test matrix, you will need to add a very brief description for each integration test so the matrix will clearly show what the tests are testing. Acceptance tests will be tested against actual user requirements and will list all the tests for each requirement.

Acceptance tests are the final tests and are largely aimed at showing the customer that the correct output is produced for different inputs. This will largely require manual testing.

Deliverables due 11 days after your lab day:

- Integration tests document (for the new functions you added) stored in repository with at least 4 sets of distinct test cases (each case must have at least 4 distinct test data).
- Integration tests coded (store in repo), executed (results in Jira and in test documents) and debugged.
- Finish implementing/coding whitebox tests. Store in repo, executed, results in Jira (and on corresponding test documents, and debugged.
- One acceptance test case for each requirement added to the test cases excel sheet.
- All acceptance tests implemented and added to the testing C++ project.
- Updated requirements traceability matrix in the repository, ensuring it shows both passed (green) and failed (red) tests.
- Completed scrum report including reflection questions answered.

Rubric:

Individual	Group participation (includes GitHub commits and Jira usage)	80%
	Teamwork	20%
Group	Integration test case document (well written, complete, good test data)	10%
	Integration test code (well designed and documented)	10%
	Finish coding all functions and **main** (well-designed, written, and documented)	10%
	Finish coding blackbox and whitebox cases (well-designed, written, and documented)	5%
	Acceptance tests (well-designed, documented, and implemented)	15%
	Requirements traceability matrix updated	5%
	Test execution (performed, results recorded, issues created)	5%
	Debugging (bugs fixed, documented, Jira updated)	5%
	Git usage (used properly with good structure).	5%
	Jira usage (creates issues, tracks progress)	15%
	Scrum report & reflections	15%
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 of 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
Ahnaf Tahmid Khan	**reflection 1, reflection 2, reflection 3 , scrum**	**N/A**
Huynh Huy Hoang	**create test cases, manage github respository**	**N/A**
Syed Abdullah	**Update Jira, Did 1 integration test set (4 tests), traceability matrix**	**N/A**
Jasmin Aro	**2 integration tests (8 test)**	**N/A**
Duong Truong Phuc Nguyen	**create set of functions and add to new header file and add to GitHub, 2 test data with 20 test cases, implementing the function and store in repository, created and add C++ testing project**	**N/A**

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	
Impact on Project	
Solution or work-around	
Delayed or Blocked Task	
Reason for delay or block	
Impact on Project	
Solution or work-around	

Summary of Meeting:

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

Topic	**Discussion Summary**	**Outcome**
Divison of tasks	**Divided work up among members**	**Everyone had a task**

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**
Dividing work up	Every one had a task

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 could not be completed, the student should indicate why this was not possible.

Member	Task Attempted	Time Spent	Complete ?
Ahnaf Tahmid Khan	**reflection 1 , reflection 2, reflection 3 , scrum**	**30 mins**	**yes**
Huynh Huy Hoang	**Create test cases, manage github respository**		
Duong Truong Phuc Nguyen	**One acceptance test case for each requirement added to the test cases excel sheet.** **Finish implementing/coding whitebox tests. Store in repo, executed, results in Jira (and on corresponding test documents, and debugged.** **All acceptance tests implemented and added to the testing C++ project.**	**4 hours**	**yes**
Syed Abdullah	**Jira, 1 set of integration tests (4 test cases), traceability matrix**		
Jasmin Aro	**integration testing (canAddPackage + add package | canAddPackage + countPackages)**		

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

Major Outcomes of Meeting:

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

Outcome	Impact on Project

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

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

Reflections:

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

1. What is the difference between manual and automated testing? Why are we automating the testing process and what benefits does automation offer?

   Automated testing relies on scripts and tools to carry out tests, whereas manual testing requires testers to perform test cases without any tools. Automation enhances precision, reduces time, and effectively manages repetitive tasks, which can be challenging for manual methods. The main advantages of automation include quicker testing cycles, reduced human errors, scalability, the ability to reuse scripts, and long-term cost efficiency. It plays a crucial role in contemporary agile software development by facilitating comprehensive regression testing, enabling round-the-clock execution, and allowing testers to focus more on innovative tasks.

2. Why it is necessary to write integration tests given that the code has already passed blackbox and whitebox tests?

   Integration tests are essential as they evaluate the interactions between various system components, even after comprehensive black-box and white-box testing. While black-box and white-box testing confirm that each unit or module functions correctly, integration tests reveal issues like data inconsistencies, faulty API calls, or failures in communication between modules. By detecting problems that might be missed during standalone unit testing, integration tests help ensure that the overall behavior of the system meets user expectations and adheres to the original design.

3. List and describe one of the integration tests you created. Provide a thorough explanation of how the integration operates, detailing the flow of parameters from one function to another. Use one of your integration tests to support your answer.

   We developed an integration test for the user login process to verify the smooth operation of the login API, authentication handler, database, and session manager. The testing process commenced with a POST request containing a username and password sent to the login API. The API then passed these credentials to the authentication handler, which accessed the database to fetch the hashed password associated with the given username. The handler compared the submitted (hashed) password against the stored hash to authenticate the user. Once the credentials were validated successfully, the handler directed the session manager to create a unique session token, which was stored in the database and included in the API response. This test ensured that all components—API, database, and session manager—worked together effectively, validating the correct flow of parameters, successful session creation, and appropriate error handling for incorrect credentials.