CAPSTONE PROJECT REPORT

Report 5 – Software Test Documentation

– Hanoi, July 25, 2023 –

Table of Contents

I. Project Report

1. Status Report

#	Work Item	Status	Notes (Work Item in Details)
1	Overall Description	Completed	
2	Test Plan	Completed	
4	Test Case	Completed	
3	Test Report	Completed	

2. Team Involvements

#	Task	Member	Notes (Task Details, etc.)
1	Overall Description		
2	Test Plan		
3	Test Case		

4	Test Report		

3. Issues/Suggestions

#	Issue	Status	Notes (Solution, Suggestion, etc.)
1		Pending	
2		In Progress	
3		Completed	

II. Project Management Plan

1. Overall Description

1.1 Test Model

This project follows the V-model process to implement testing.

No	Verification Phases	Validation Phases

1	**User Requirement:** This is the first phase in the development cycle where the product requirements are understood from the customer's perspective. This phase involves detailed communication with the customer to understand his expectations and the exact requirement of the project. This is a very important activity and it needs to be managed well because most of the customers are not sure about what exactly they need. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing.	**Acceptance Testing:** User Acceptance Testing is conducted to determine whether the system is ready for release. During this phase, the tester and some alpha test user will test the system to find out whether the application meets their business needs. Acceptance testing is associated with the business requirement analysis phase and involves testing the product in the user environment. Acceptance tests uncover the compatibility issues with the other systems available in the user environment. It also discovers non-functional issues such as load and performance defects in the actual user environment.
2	**Software Requirement Specification:** Once you have your clear and detailed product requirements, it's time to complete the system design. The system designer will complete the understanding and detailing of the software and communication setup for the product under development. The system test plan was developed based on the system design. Doing this at an earlier stage leaves more time for the actual test execution afterward.	**System Testing:** System testing is the first level in which the complete application is tested as a whole. System testing is directly associated with the system design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during this system test execution.

3	**Architectural Design**: Architectural specifications are understood and designed in this phase. Usually, more than one technical approach is proposed, and based on the technical and financial feasibility the final decision is taken. The system design is broken down further into modules taking up different functionality. This is also referred to as High-Level Design (HLD). The data transfer and communication between the internal modules and with the outside world (other systems) is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.	**Integration Testing:** Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.
4	**Detailed Design:** In this phase, the detailed internal Design for all the system modules is specified, referred to as Low-Level Design(LLD). The design must be compatible with the other modules in the system architecture and the other external systems. Unit tests are an essential part of any development process and help eliminate the maximum faults and errors at a very early stage. These unit tests can be Designed at this stage based on the internal module Designs.	**Unit Testing:** The main aim of this endeavour is to determine whether the application functions as designed. In this phase, a unit can refer to a function, individual program, or even a procedure, and a White Box Testing method is usually used to get the job done. Unit tests Designed in the module Design phase are executed on the code during this validation phase. Unit testing is the testing at the code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.

1.2 Testing Levels

Regarding the testing phase, there are four main software testing phases:

1.2.1. Unit Testing

A Unit is a smallest testable portion of a system or application that can be compiled, liked, loaded, and executed. This kind of testing helps to test each module separately. The aim is to

test each part of the software by separating it. It checks that components are fulfilling functionalities or not. This kind of testing is performed by developers.

During this first round of testing, the program is submitted to assessments that focus on specific units or components of the software to determine whether each one is fully functional. The main aim of this endeavour is to determine whether the application functions as Designed. In this phase, a unit can refer to a function, individual program, or even a procedure, and a White-box Testing method is usually used to get the job done. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible. It's quite common for software developers to perform unit tests before delivering software to testers for formal testing.

You can reference at https://viblo.asia/p/unit-test-co-ban-jvElaGGoKkw

1.2.2. Integration Testing

Integration means combining. For Example, In this testing phase, different software modules are combined and tested as a group to make sure that the integrated system is ready for system testing.

Integrating testing checks the data flow from one module to other modules. This kind of testing is performed by testers.

Integration testing allows individuals the opportunity to combine all of the units within a program and test them as a group. This testing level is Designed to find interface defects between the modules/functions. This is particularly beneficial because it determines how efficiently the units are running together. Keep in mind that no matter how efficiently each unit is running, if they aren't properly integrated, it will affect the functionality of the software program. To run these types of tests, individuals can make use of various testing methods, but the specific method that will be used to get the job done will depend greatly on how the units are defined.

You can reference at https://viblo.asia/p/tim-hieu-ve-integration-testing-QpmleJMo5rd

1.2.3. System Testing

System testing is performed on a complete, integrated system. It allows checking the system's compliance as per the requirements. It tests the overall interaction of components. It involves load, performance, reliability, and security testing. System testing most often the final test to verify that the system meets the specification. It evaluates both functional and non-functional needs for the testing.

System testing is the first level in which the complete application is tested as a whole. The goal at this level is to evaluate whether the system has complied with all of the outlined requirements and to see that it meets Quality Standards. System testing is undertaken by

independent testers who haven't played a role in developing the program. This testing is performed in an environment that closely mirrors production. System Testing is very important because it verifies that the application meets the technical, functional, and business requirements that were set by the customer.

You can reference at https://viblo.asia/p/system-testing-kiem-thu-he-thong-aWj53pOPK6m

1.2.4. Acceptance Testing

Acceptance testing is a test conducted to find if the requirements of a specification or contract are met as per its delivery. Acceptance testing is done by the user or customer. However, other stockholders can be involved in this process.

The final level, Acceptance testing (or User Acceptance Testing), is conducted to determine whether the system is ready for release. During the Software development life cycle, requirements changes can sometimes be misinterpreted in a fashion that does not meet the intended needs of the users. During this final phase, the user will test the system to find out whether the application meets their business needs. Once this process has been completed and the software has passed, the program will then be delivered to production.

You can reference at https://viblo.asia/p/user-acceptance-testing-1VgZv8R95Aw

1.3 Testing Types

1.3.1. Function Testing

- The implementation of the functional test will be passed if all functional cases in the Test case document are tested and passed.
- The test case will have to cover all logic branches that function or method could execute with different data input.
- All functional test cases have been executed to verify proper data acceptance, processing, and retrieval, and the appropriate implementation of the business rules, and passed.
- The appropriate activities will be performed when valid data is used
- The corresponding error/warning message mechanism is applied for each specific case.

1.3.2. UI/UX Testing

- UI/UX tests will be performed fully on all screens.
- UI/UX test targets to cover the verification of the overall look and feel of the system including initial position, font, text size, colour, focus, initial button, tab order, label, screen sizes, sentence width, and animation.
- Check all the UI elements for colour, size, position, width, length and

- acceptance of characters or numbers
- All GUI test cases have been executed and passed.

- Any defects related to GUI are fixed, except those accepted by the customer.

1.3.3. Acceptance Testing

- Acceptance testing is a test conducted to determine if the requirements of a specification or contract are met.
- Acceptance testing takes place at a close relation user's site and are free tests to detect bugs and strange behaviour. By that, the development team will improve the system.
- It involves alpha testing and beta testing. Alpha testing takes place at developers' sites and involves testing of the operational system by the internal members before it is released to external users. Beta testing

- takes place at user's sites and involves testing by a group of users who use the system at their locations and provide feedback, before the system is released to all users.

2. Test Plan

2.1 Test Stages

Types of Test	**Stage of Test**			
	Unit	**Integration**	**System**	**Acceptance**
Function Test	X	X	X	X

User Interface Test	X	X	X	X
System Test			X	
Acceptance Test				X

2.2 Resources

a. Human Resources

Worker/Doer	**Role**	**Specific Responsibilities/Comments**
	Test Leader/Tester	Manage test resources and assign test tasks Create Test Plan, Test Cases (IT, ST), Test Scripts (IT, ST) Review Test Data Final Inspection Test Cases Create Test Reports
	Project Manager/Developer/ Tester	Approve Test Cases (UT, IT, ST), Test Plan, Test Results, Test Reports
	Quality Assurance	Create UT Cases, UT Reports
	Developer	Create UT Cases, UT Reports

b. Environment

Purpose	Tool	Provider	Version
Unit Testing	Visual Studio Code IntelliJ IDEA	Microsoft Corporation JetBrains	
User Interface Testing	Chrome	Google	
Integration Testing	Chrome	Google	
System Testing	Chrome	Google	
Acceptance Testing	Chrome	Google	

2.3 Test Milestones

Milestone Task	Effort (MD)	Start Date	End Date
Iteration 1: System Test	**7**	**June 09, 2023**	**June 17, 2023**
Create Test Case	7	June 09, 2023	June 17, 2023
Iteration 2: Testing	**15**	**June 18, 2023**	**July 08, 2023**
Create Test Plan	6	June 18, 2023	June 25, 2023
Create Unit Test Case	6	June 18, 2023	June 25, 2023
Create Test Case Iteration 2	6	June 18, 2023	June 25, 2023
Test Iteration 2	10	June 25, 2023	July 08, 2023

Verify Test Iteration 2	10	June 25, 2023	July 08, 2023
Fix Bug Iteration 2	10	June 25, 2023	July 08, 2023
Test Document	**27**	**June 29, 2023**	**July 05, 2023**
Create Test Case	7	June 30, 2021	July 08, 2023
Execute Test Case	15	July 09, 2021	July 29, 2023
Acceptance Test	5	July 30, 2021	July 05, 2023
Iteration 3: Testing	**15**	**July 16, 2021**	July **05,** 2023
Create Test Case Iteration 3	5	July 16, 2021	July 22, 2023
Test Iteration 3	5	July 23, 2021	July 23, 2023
Verify Test Iteration 3	5	July 23, 2021	July 23, 2023
Fix Bug Iteration 3	5	July 25, 2023	July 25, 2023

2.4 Deliverables

No	Deliverables	Due Date	
		Iteration 1	**Iteration 2**
1	Test Plan	June 1, 2023	June 8, 2023
2	Test case	June 09, 2023	July 15, 2023
3	Test case review	June 09, 2023	July 15, 2023

4	Defect report	July 15, 2021	July 23, 2023
5	Final test summary report	August 23, 2021	

3. Test Cases

3.1 Unit Testing

All unit tests are located in the folder **SWP391_ONVID\onlinelearning\src\test\java\group5\swp391\onlinelearning** in source code. Here is an example of unit test cases:

3.2 Integration Testing & System Testing

The term “integration” means a process of combining into an integral whole. Similarly, integration testing of a build indicates separate testing units of a system as one.

A system test inspects every software unit to secure their proficiency as a whole or assembled build. In software engineering, a system test takes place after unit and integration testing.

Integration Test and System Test are done by testers to ensure that combined units work correctly and that the system functions as intended.

GUI testing is also done during this process to ensure that elements and functions load correctly, the text is readable, and the website interface looks good in various browser sizes.

This is the sample tests of Integration Test and System Test in SAFO system:

Detailed test cases of these types of testing are described in 2 files: SAFO_IntegrationTest.xlsx (for Integration Testing) and SAFO_SystemTest.xlsx (for System Testing).

Module	*Feedback*
Test requirement	*Write in clear, understandable language, avoiding unnecessary language, acronyms, or ambiguous phrases.*
Tester	*NguyenTK(11)*

Pass	Fail	Untested	N/A	Number of Test cases
11	0	0	0	11

ID	Test Case Description	Test Case Procedure	Expected Output	Inter-test case Dependence	Result	Test date	Tester	Note
	Submit feedback Function							
[Feedback-1]	Test submit feedback	1.Login system by student role 2.Click a course you want to feedback 3.Click feedback	1.Form feedback will display 2. The "Feedback" view form is displayed with the folowing informations: - Comment - Star rate	1.Login by Student role 2.This course was enrollled by student	Pass	22/6/202	NguyenTK	
[Feedback-2]	Test submit feedback(invallid:leave the comment field blank)	1.Login to the system with a student role. 2.Select a course you want to provide feedback for. 3.Click on the "Feedback" option. 4.Enter invalid inputs in the feedback form (e.g., leave the comment field blank). 5.Submit the feedback form.	1.Form feedback will display 2. The "Feedback" view form is displayed with the folowing informations: - Comment - Star rate 3.System display message "comment is not blank"	1.Login by student	Pass	10/7/202	NguyenTK	
[Feedback-3]	Test submit	1.Login to the system with a	1.Form feedback will display	1.Login by student	Pass	10/7/202	NguyenTK	

	View list account							
[Account-1]	View list user success	1. Click "Users" tab in the menu. 2. Click "List Users" under "Users" tab	The "User information" view list is displayed with the folowing informations: -Id -Name -Email -Role -Status	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	
[Account-2]	View empty list users	1. Click "Users" tab in the menu. 2. Click "List Users" under "Users" tab	The system return the table with no data and message "No users exist"	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	
[Account-3]	View list students success	1. Click "Users" tab in the left menu. 2. Click "List Students" under "Users" tab	The "User information" view list is displayed with the folowing informations: -Id -Name -Email -Role -Status -Number of Courses -Number of Orders -Number of Feedbacks	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	
[Account-4]	View empty list students	1. Click "Users" tab in the left	The system return the table with no data and message "No students exist"	1. Login the system with Admin role.	Pass	11/07/2023	KhaNM	

[Account-4]	View empty list students	1. Click "Users" tab in the left menu. 2. Click "List Students" under "Users" tab	The system return the table with no data and message "No students exist"	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	
[Account-5]	View list teachers success	1. Click "Users" tab in the left menu. 2. Click "List teachers" under "Users" tab	The "User information" view list is displayed with the folowing informations: -Id -Name -Email -Role -Status -Number of Courses -Rate star -Wallet	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	
[Account-6]	View empty list teachers	1. Click "Users" tab in the left menu. 2. Click "List teachers" under "Users" tab	The system return the table with no data and message "No teachers exist"	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	
[Account-7]	View list staffs success	1. Click "Users" tab in the left menu. 2. Click "List staffs" under "Users" tab	The "User information" view list is displayed with the folowing informations: -Id -Name -Email -Role -Status	1. Login the system with Admin role. 2. Standing at admin home screen	Pass	11/07/2023	KhaNM	

Figure: Integration Test and System Test Case Example

3.3 Acceptance Testing

Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery. But our project will use the checklist as a substitute for acceptance testing. Detailed checklists are described in check list below.

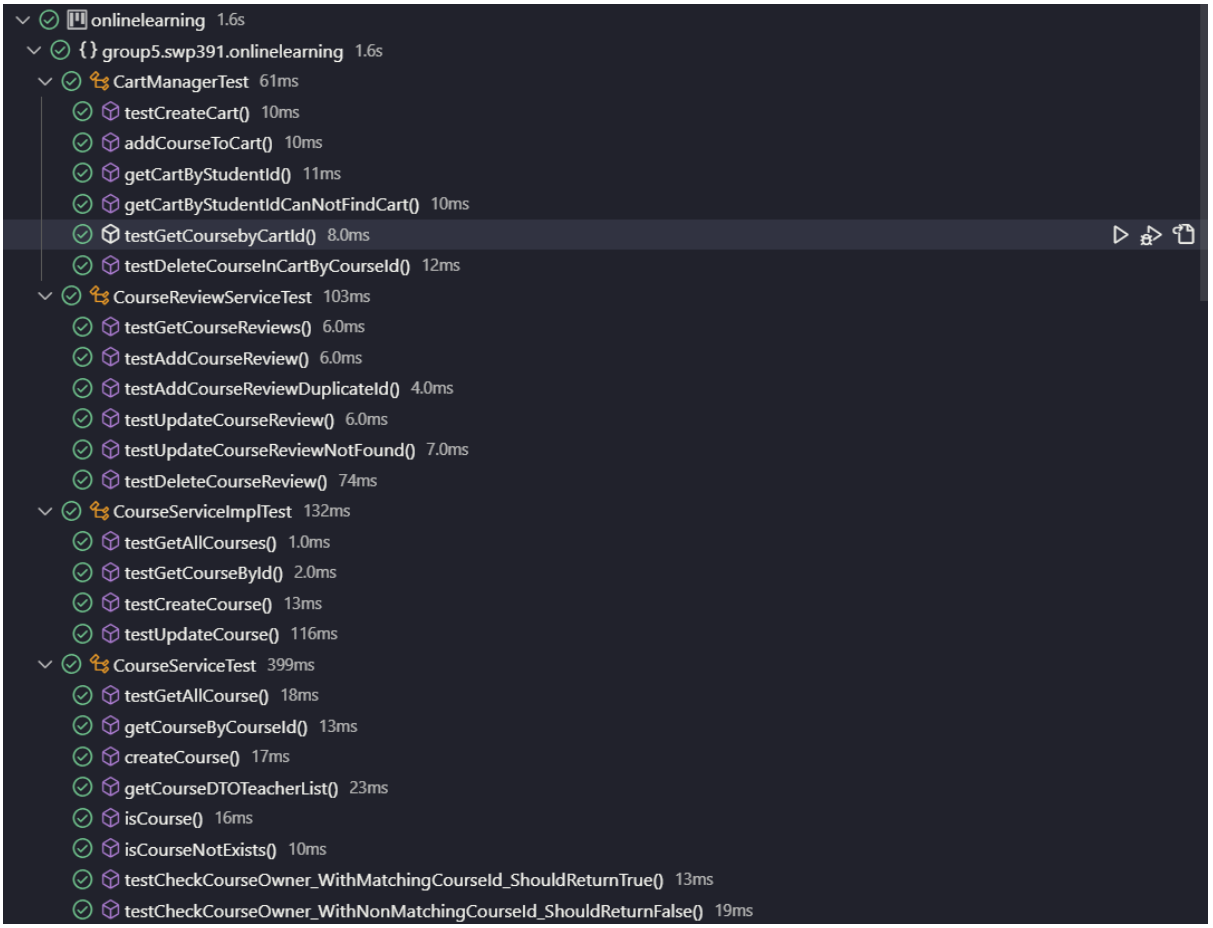
Checklist				
	ID		**Yes**	**No**
General	CL-01	Text on all pages for spelling and grammatical errors.	√	
	CL-02	Every buttons have activity and working normally	√	
	CL-03	All error messages are displayed in red.	√	
	CL-04	All mandatory fields are validated	√	
	CL-05	Delete functionality for any record on the page is asked for confirmation.	√	
	CL-06	All required field cannot be blank/space	√	
	CL-07	application crash or unavailable pages are redirected to an error page.	√	
GUI and Usability	CL-08	Screens are designed following project requirement	√	
	CL-09	All fields on page (text box, radio options, dropdown lists) should be aligned properly	√	
	CL-10	User-friendly and easy-to-use screen interface	√	

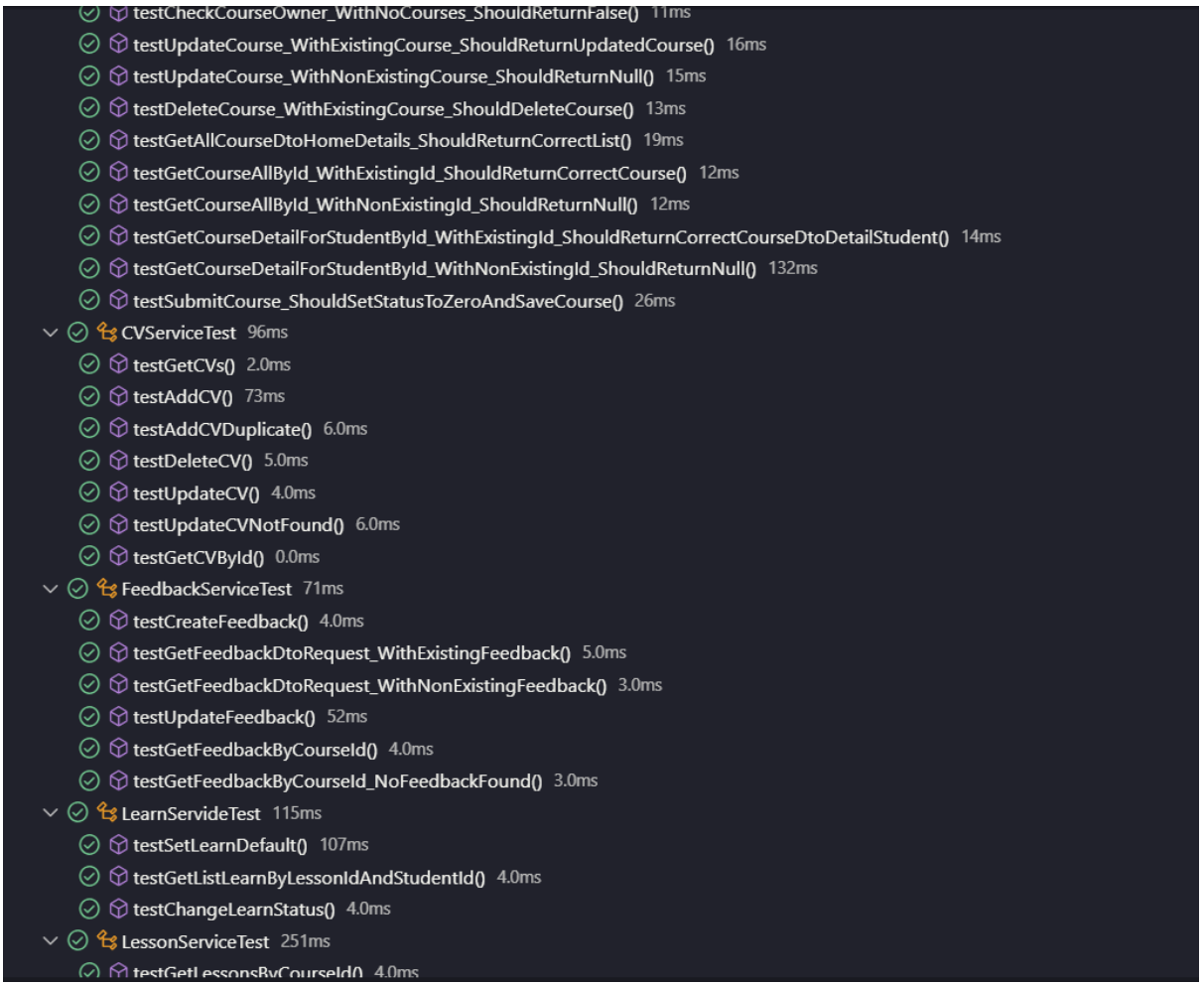
	CL-11	Information is arranged symmetrically with adequate spacing between components.	√	
	CL-12	The most important fields are located where they are easy to see.	√	
	CL-13	Information is presented in the order that the user needs it.	√	
	CL-14	The important fields and buttons are located where they are easy to see.	√	
	CL-15	The text is easy to read. Do not use slang, acronyms, and abbreviations	√	
	CL-16	The text is clear, concise, and meaningful	√	
	CL-17	Pop-up menus are provided for the user to access information about an object's properties or perform specific tasks on the object.	√	
Database	CL-18	Correct data is saved in the database.	√	
	CL-19	Not null column has values	√	
	CL-20	Fields are designed with correct data type and data length.	√	
	CL-21	Input field leading and trailing spaces are truncated before committing data to the database	√	
	CL-22	Input numeric fields with minimum, maximum, and float values	√	
Performance	CL-23	Quick response display	√	

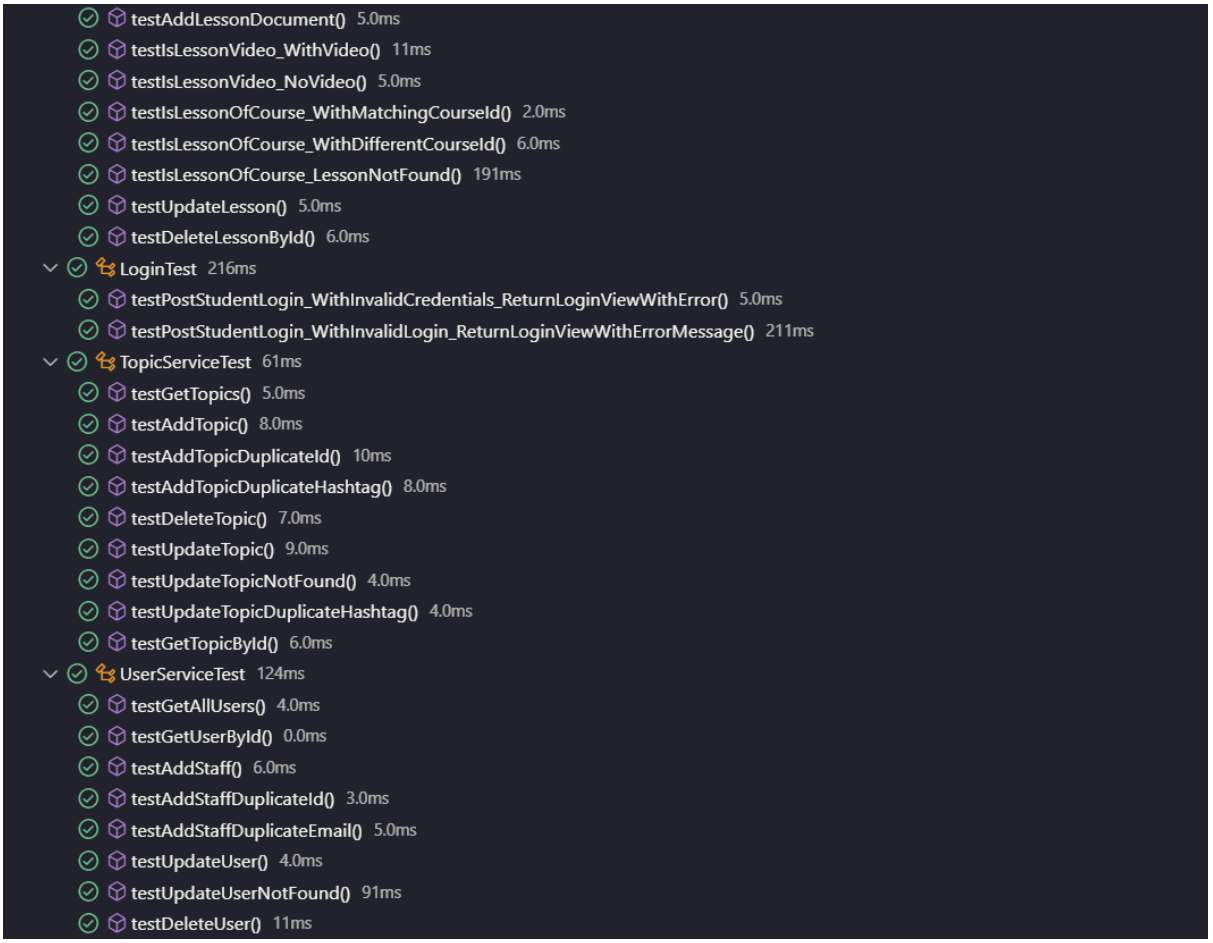
	CL-24	Receive uninterrupted notifications	√	
	CL-25	Multiple users can access at the same time	√	
Security	CL-26	Check password security and password policy enforcement.	√	
	CL-27	Verify old password When change password	√	
	CL-28	Check application logout functionality.	√	

4. Test Reports

4.1 Unit Test Report




testCheckCourseOwner_WithNoCourses_ShouldReturnFalse() 11ms
testUpdateCourse_WithExistingCourse_ShouldReturnUpdatedCourse() 16ms
testUpdateCourse_WithNonExistingCourse_ShouldReturnNull() 15ms
testDeleteCourse_WithExistingCourse_ShouldDeleteCourse() 13ms
testGetAllCourseDtoHomeDetails_ShouldReturnCorrectList() 19ms
testGetCourseAllById_WithExistingId_ShouldReturnCorrectCourse() 12ms
testGetCourseAllById_WithNonExistingId_ShouldReturnNull() 12ms
testGetCourseDetailForStudentById_WithExistingId_ShouldReturnCorrectCourseDtoDetailStudent() 14ms
testGetCourseDetailForStudentById_WithNonExistingId_ShouldReturnNull() 132ms
testSubmitCourse_ShouldSetStatusToZeroAndSaveCourse() 26ms
CVServiceTest 96ms
testGetCVs() 2.0ms
testAddCV() 73ms
testAddCVDuplicate() 6.0ms
testDeleteCV() 5.0ms
testUpdateCV() 4.0ms
testUpdateCVNotFound() 6.0ms
testGetCVById() 0.0ms
FeedbackServiceTest 71ms
testCreateFeedback() 4.0ms
testGetFeedbackDtoRequest_WithExistingFeedback() 5.0ms
testGetFeedbackDtoRequest_WithNonExistingFeedback() 3.0ms
testUpdateFeedback() 52ms
testGetFeedbackByCourseId() 4.0ms
testGetFeedbackByCourseId_NoFeedbackFound() 3.0ms
LearnServideTest 115ms
testSetLearnDefault() 107ms
testGetListLearnByLessonIdAndStudentId() 4.0ms
testChangeLearnStatus() 4.0ms
LessonServiceTest 251ms
testGetLessonsByCourseId() 4.0ms



4.2 Integration Test and System Test Report

No	Module code	Pass	Fail	Untested	N/A	Number of test cases
1	Learn	12	0	1	0	13
2	Feedback	11	0	0	0	11
3	Account	18	0	0	0	18
4	Course	27	0	0	0	27
5	Admin	11	0	0	0	11
6	Guest	8	0	0	0	8
7	Profile	6	0	0	0	6
8	Subject	8	1	0	0	9
9	Cart	7	0	1	0	8
10	Search	9	0	0	0	9
	Sub total	117	0	1	0	120

Test coverage **97,50** %
Test successful coverage **97,50** %

4.3 Acceptance Test Report

Checklist				
	ID		**Yes**	**No**
General	CL-01	Text on all pages for spelling and grammatical errors.	√	
	CL-02	Every buttons have activity and working normally	√	
	CL-03	All error messages are displayed in red.	√	
	CL-04	All mandatory fields are validated	√	
	CL-05	Delete functionality for any record on the page is asked for confirmation.	√	
	CL-06	All required field cannot be blank/space	√	
	CL-07	application crash or unavailable pages are redirected to an error page.	√	
GUI and Usability	CL-08	Screens are designed following project requirement	√	
	CL-09	All fields on page (text box, radio options, dropdown lists) should be aligned properly	√	
	CL-10	User-friendly and easy-to-use screen interface	√	
	CL-11	Information is arranged symmetrically with adequate spacing between components.	√	

	CL-12	The most important fields are located where they are easy to see.	√	
	CL-13	Information is presented in the order that the user needs it.	√	
	CL-14	The important fields and buttons are located where they are easy to see.	√	
	CL-15	The text is easy to read. Do not use slang, acronyms, and abbreviations	√	
	CL-16	The text is clear, concise, and meaningful	√	
	CL-17	Pop-up menus are provided for the user to access information about an object's properties or perform specific tasks on the object.	√	
Database	CL-18	Correct data is saved in the database.	√	
	CL-19	Not null column has values	√	
	CL-20	Fields are designed with correct data type and data length.	√	
	CL-21	Input field leading and trailing spaces are truncated before committing data to the database	√	
	CL-22	Input numeric fields with minimum, maximum, and float values	√	
Performance	CL-23	Quick response display	√	

Security	CL-24	Check password security and password policy enforcement.	√	
	CL-25	Send otp code to email when changing password	√	
	CL-26	Check application logout functionality.	√	