



AdventureLearn Game Application Test Cases

Version 1.1
23/04/2020

Members:

Aloysius Foo Dun Jiat
Farhan Khalifa Ibrahim
Florowski Kamil
Lee Jiho
Ng Jin Han Benedict
Nguyen Duy Khanh
Nicholas Irving Darmadji
Rickson Hu Hong Rui
Sven Tan Wei Jie
Yuen Kim Hwee

Team: NanyangGuys

CZ3003 Software System Analysis and Design, Tutorial Group SSP5

VERSION HISTORY

Version #	Implemented By	Revision Date	Approved By	Approval Date	Reason
1.0	Kamil Florowski, Nicholas Irving	22/4/2020	Nicholas Irving	24/4/2020	Initial version
1.1	Kamil Florowski, Nicholas Irving	23/4/2020	Nicholas Irving	24/4/2020	Added summary report, appendix

VERSION HISTORY	2
1. Student's Primary Test Cases	4
1.1 Student Login Test Case	4
1.2 Student Character Selection Test Case	5
1.3 Student View Profile Test Case	6
1.4 Student Change Character Test Case	7
1.5 Student View Leaderboard Test Case	9
1.6 Student View Custom Levels Test Case	10
1.7 Student View Created Levels Test Case	11
1.8 Student View Assignments Test Case	14
1.9 Student View Credits Test Case	15
1.10 Student Settings Test Case	16
2. Level Creation/Editing Test Cases	18
2.1 Student Creates/Edits Custom Level Test Case	18
2.2 Teacher Creates/Edits Assignment Test Case	22
3. Teacher's Primary Test Cases	26
3.1. Teacher Publishes Assignment Test Case	26
3.2. Teacher Checks Summary Report Test Case	28
4. Gameplay Test Cases	30
4.1. Main Gameplay Loop Test Case	30
4.1.1. Main Gameplay Loop (Successful)	30
4.1.2. Main Gameplay Loop (Unsuccessful)	32
4.1.3. Main Gameplay Loop (Successful + Use Ability)	34
4.1.4. Main Gameplay Loop (Successful + Use Ability)	36
4.2. Campaign Gameplay Test Case	37
4.3. Assignment Gameplay Test Case	40
4.4. Custom Level Gameplay Test Case	42
5. Test Cases Summary Report	44
6. Appendix A: Miscellaneous White Box Testing (Legacy)	45
[LEGACY TESTING RESULTS]	45

1. Student's Primary Test Cases

1.1 Student Login Test Case

Test Case ID: #01	Test Case Name: Student Login
Subsystem: Login	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student login with social media account.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User has a valid Google or Facebook account.

Steps	Action	Expected System Response
1	User clicks on google or facebook login button.	System displays an appropriate login pop up.
2	User selects which Google or Facebook account to log in with.	System prompts success message.

Test Value		Description	Expected Result	Actual Result
#	Social Media Account ID			
1	-	No social media account ID given, i.e. user device currently not logged-in in Google/Facebook account	Fail	Fail

2	0123	Invalid social media ID	Fail	Fail
3	105141273653383906034	Valid Google account ID	Pass	Pass
4	102292528066576	Valid Facebook account ID	Pass	Pass

Post-Conditions	
1	User logs in successfully and is directed to the character selection page.

1.2 Student Character Selection Test Case

Test Case ID: #02	Test Case Name: Student Character Selection
Subsystem: CharSelect	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student selects a character after logging in.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.

Steps	Action	Expected System Response
1	User presses one of the portraits of the four available characters.	System displays the selected character's name, sprite, special ability, and ability description.
2	User may select other icons to preview	System displays the selected

	ther character's details.	character's name, sprite, special ability, and ability description.
3	User presses "enter" to confirm their character selection.	System redirects the user to the main menu and registers the user's character into the database.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	User is directed to the main menu.
2	User's selected character is registered into the database.

1.3 Student View Profile Test Case

Test Case ID: #03	Test Case Name: Student View Profile
Subsystem: UserProfile	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student views their profile details.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.

4	User has selected a character.
5	User is in the main menu.

Test Value	Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.		

Steps	Action	Expected System Response
1	User presses on the "profile" button.	System directs user to the profile page
		System displays user's profile details: name, average score, campaign rank, currently selected character, and character's ability.

Post-Conditions	
1	User is directed to the profile page.

1.4 Student Change Character Test Case

Test Case ID: #04	Test Case Name: Student Change Character
Subsystem: CharSelect	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student changes the character they would like to play as.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.
4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "profile" button.	System directs user to the profile page.
2	User presses the "change character" button.	System directs user to the character selection page.
3	User selects one of the characters.	System displays the selected character's name, sprite, special ability, and ability description.
4	User presses "enter" to confirm their character selection.	System redirects the user to the main menu and registers the user's new character into the database.

Test Value	Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.		

Post-Conditions	
1	User is directed to the main menu.
2	User's newly changed character is updated into the database.

1.5 Student View Leaderboard Test Case

Test Case ID: #05	Test Case Name: Student View Leaderboard
Subsystem: Leaderboard	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student views the current leaderboard scores.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.
4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Leaderboard" button.	System directs user to the leaderboard page.
		System displays the overall rank number, username, and score.
2 - 1	User presses the drop down selection (currently displaying "all worlds").	System displays the worlds available.
2 - 2	User selects "world 1".	System displays the leaderboard details for world 1.
3 - 1	User presses the drop down selection (currently displaying "all world 1").	System displays the worlds available.
3 - 2	User selects "world 2".	System displays the leaderboard details for world 2.

4 - 1	User presses the drop down selection (currently displaying "all worlds").	System displays the worlds available.
4 - 2	User selects "world 3".	System displays the leaderboard details for world 3.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	User is directed to the leaderboard page.

1.6 Student View Custom Levels Test Case

Test Case ID: #06	Test Case Name: Student View Custom Levels
Subsystem: CustomLevel	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student views currently available custom levels.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.

4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Custom Level" button.	System directs user to the main custom level page.
		System displays available custom levels and its details, which includes: game name, game creator.
2	User presses the right arrow button.	System directs user to the cleared custom level page.
		System displays all the custom levels that the user has cleared, which include details such as: game name, game creator, score.

Test Value	Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.		

Post-Conditions	
1	User is directed to the custom level page.

1.7 Student View Created Levels Test Case

Test Case ID: #07	Test Case Name: Student View Created Levels
--------------------------	--

Subsystem: CustomLevel	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student views custom levels of their creation.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.
4	User has selected a character.
5	User is in the custom level page.

Steps	Action	Expected System Response
1	User presses the "View Created Levels" button.	System directs user to the user-created levels page
		System displays all custom level(s) the user has created, which include details such as: game name
2	(optional) User presses the "Delete" button	System deletes the user-created custom level and updates the database

Test Value	Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.		

Post-Conditions	
1	User is directed to the created level page.
2	If user chose to delete a created level, the selected user-created level is deleted in the database

1.8 Student View Assignments Test Case

Test Case ID: #08	Test Case Name: Student View Assignments
Subsystem: StudentAssignment	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student views currently available assignments.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.
4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Assignment" button.	System directs user to the main assignments page.
		System displays all assignments given to the user, which includes details such as: assignment name, teacher, due date.
2	User presses the right arrow button	System directs user to the cleared assignments page.
		System displays all completed assignments, which include details such as: assignment name, teacher, due date, score.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	User is directed to the assignments page.

1.9 Student View Credits Test Case

Test Case ID: #09	Test Case Name: Student View Credits
Subsystem: Credits	Test Priority: Low
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Shows the game's credits.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.
4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
-------	--------	--------------------------

1	User presses the "Credits" button.	System directs user to the main credits roll.
		System displays all the credits breakdown of the application. Once the roll ends, the user is redirected to the main menu.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	User is redirected to the main menu.

1.10 Student Settings Test Case

Test Case ID: #10	Test Case Name: Student Settings
Subsystem: Settings	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Tweak basic game settings.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.

4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Settings" button.	System displays a settings pop-up.
		System allows the user to modify the game's sound effect volume and background music volume.
2	User moves the sliders of sound effect volume and background music volume.	System changes the sound effect / background music volume respectively.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	User is directed to the settings pop-up.

2. Level Creation/Editing Test Cases

2.1 Student Creates/Edits Custom Level Test Case

Test Case ID: #11	Test Case Name: Student Creates/Edits Custom Level
Subsystem: CustomLevel	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Student create and/or edit new custom levels.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system.
4	User has selected a character.
5	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Custom Level" button.	System directs user to the main custom levels page.
2 - 1	User presses the "Create Level" button.	System directs user to the level creation menu.
2 - 2 - 1 (AF starts)	User presses the "View Created Levels" button.	System directs user to the user-created levels page.
2 - 2 - 2 (AF ends)	User presses the "Edit" button on one of their created levels.	System directs user to the level creation menu.
3	User enters the level creation menu.	System is displaying level creation menu.

3 - 1	User enters desired level name.	System displays the given level name. If an invalid name is given, an error message pop-up will be displayed and user is prompted to try another input.
3 - 2	User selects the desired level monster.	System displays selected monster's sprite animation. Monster selection can never be invalid as the options are restricted to a select few monsters.
3 - 3	User selects a time limit.	System displays the selected time limit. Time limit can never be invalid as the options are restricted to a select few timings.
3 - 4	User presses the "Next" button.	System directs user to the question creation menu.
3 - 5	User enters the desired question title.	System displays the given question title. If an invalid title is given, an error message pop-up will be displayed and user is prompted to try another input.
3 - 6	User enters four possible answers, with one being the correct one.	System displays all the given answers. If an invalid answer is given, an error message pop-up will be displayed and user is prompted to try another input.
3 - 7	User indicates the correct answer by selecting the bullet.	System displays the correct option with a highlighted bullet.
3 - 8	Repeat 3 - 5 to 3 - 7 four more times to fill in all the five questions required to create a level.	System repeats the same updates as 3-5 to 3-7.

3 - 9	User presses the "Create" button.	System directs user to the main menu and updates the database with the user's created/edited custom level.
--------------	-----------------------------------	--

Test Value		Description	Expected Result	Actual Result
#	Level Name			
1	-	No name is given, the level name input field is left blank.	Fail	Fail
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus commodo erat arcu, placerat consequat enim viverra vitae.	Name given is too long, exceeding 100 characters.	Fail	Fail
3	Lorem Ipsum	Valid name.	Pass	Pass
4	Lorem Ipsum	Valid name, the application allows for multiple user-created levels of the same name.	Pass	Pass

Test Value		Description	Expected Result	Actual Result
#	Question Title			
1	-	No question title is given, the question input field is left blank.	Fail	Fail
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus commodo erat arcu, placerat consequat enim viverra vitae.	Question title given is too long, exceeding 100 characters.	Fail	Fail
3	Lorem Ipsum	Valid question.	Pass	Pass
4	Lorem Ipsum	Valid question, the application allows for similar question titles.	Pass	Pass

Test Value		Description	Expected Result	Actual Result
#	Answer Options			
1	-	No answer is given, the answer input field is left blank.	Fail	Fail
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus commodo erat arcu, placerat consequat enim viverra vitae.	Answer given is too long, exceeding 100 characters.	Fail	Fail
3	Lorem Ipsum	Valid answer.	Pass	Pass
4	Lorem Ipsum	Valid answer, the application allows for similar question titles.	Pass	Pass

Post-Conditions	
1	User is directed to the main menu.
2	User's created/edited level is registered/updated into the database.

2.2 Teacher Creates/Edits Assignment Test Case

Test Case ID: #12	Test Case Name: Teacher Creates/Edits Assignment
Subsystem: TeacherAssignment	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Teacher creates and/or edits created assignments.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system as a teacher
4	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Assignment" button.	System directs user to the main assignments page.
2 - 1	User presses the "Create Assignment" button.	System directs user to the level creation menu.
2 - 2 (AF Starts, Ends)	User presses the "Edit" button on one of their created assignments.	System directs user to the level creation menu.
3	User enters the level creation menu.	System is displaying level creation menu.
3 - 1	User enters desired level name.	System displays the given level name. If an invalid name is given, an error message pop-up will be

		displayed and user is prompted to try another input.
3 - 2	User selects the desired level monster.	System displays selected monster's sprite animation. Monster selection can never be invalid as the options are restricted to a select few monsters.
3 - 3	User selects a time limit.	System displays the selected time limit. Time limit can never be invalid as the options are restricted to a select few timings.
3 - 4	User presses the "Next" button.	System directs user to the question creation menu.
3 - 5	User enters the desired question title.	System displays the given question title. If an invalid title is given, an error message pop-up will be displayed and user is prompted to try another input.
3 - 6	User enters four possible answers, with one being the correct one.	System displays all the given answers. If an invalid answer is given, an error message pop-up will be displayed and user is prompted to try another input.
3 - 7	User indicates the correct answer by selecting the bullet.	System displays the correct option with a highlighted bullet.
3 - 8	Repeat 3 - 5 to 3 - 7 four more times to fill in all the five questions required to create a level.	System repeats the same updates as 3-5 to 3-7.
3 - 9	User presses the "Create" button.	System directs user to the main menu and updates the database with the user's created/edited assignmentl.

Test Value		Description	Expected Result	Actual Result
#	Level Name			
1	-	No name is given, the level name input field is left blank.	Fail	Fail
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus commodo erat arcu, placerat consequat enim viverra vitae.	Name given is too long, exceeding 100 characters.	Fail	Fail
3	Lorem Ipsum	Valid name.	Pass	Pass
4	Lorem Ipsum	Valid name, the application allows for multiple assignments of the same name.	Pass	Pass

Test Value		Description	Expected Result	Actual Result
#	Question Title			
1	-	No question title is given, the question input field is left blank.	Fail	Fail
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus commodo erat arcu, placerat consequat enim viverra vitae.	Question title given is too long, exceeding 100 characters.	Fail	Fail
3	Lorem Ipsum	Valid question.	Pass	Pass
4	Lorem Ipsum	Valid question, the application allows for similar question titles.	Pass	Pass

Test Value		Description	Expected Result	Actual Result
------------	--	-------------	-----------------	---------------

#	Answer Options			
1	-	No answer is given, the answer input field is left blank.	Fail	Fail
2	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus commodo erat arcu, placerat consequat enim viverra vitae.	Answer given is too long, exceeding 100 characters.	Fail	Fail
3	Lorem Ipsum	Valid answer.	Pass	Pass
4	Lorem Ipsum	Valid answer, the application allows for similar question titles.	Pass	Pass

Post-Conditions	
1	User is directed to the main menu.
2	User's created/edited level is registered/updated into the database.

3. Teacher's Primary Test Cases

3.1. Teacher Publishes Assignment Test Case

Test Case ID: #13	Test Case Name: Teacher Publishes Assignment
Subsystem: TeacherAssignment	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Teacher publishes their created assignments.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system as a teacher.
4	User is in the main menu.
5	User has previously created assignments.

Steps	Action	Expected System Response
1	User presses the "Assignment" button.	System directs user to the main assignments page.
2	User presses the "Publish Assignment" button.	System directs user to the publish assignment page.
3	User selects which assignment to publish from the drop down menu.	System displays selected assignment.
4	User sets the due date by selecting the desired day, month, and year from the drop down menus.	System displays all the selected timings.

5	If the assignment has not been published, user ticks all the classes that they would like to publish the assignment to, according to the class IDs.	System displays a tick for all the class IDs of the classes the user want to publish the assignment to.
6	User presses the "Publish" button.	System directs user to the main menu, updates the database with published assignments.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	User is directed to the main menu.
2	User's assignments are assigned to the desired class groups based on class IDs.

3.2. Teacher Checks Summary Report Test Case

Test Case ID: #14	Test Case Name: Teacher Checks Summary Report
Subsystem: SummaryReport	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Nicholas Irving	Execution Date: 22/4/2020
Description: Teacher checks the performance report of their assignments and/or the game's built-in campaigns.	

Pre-Conditions	
1	Application is connected to the internet.
2	Database is appropriately filled.
3	User is successfully logged in to the system as a teacher.
4	User is in the main menu.

Steps	Action	Expected System Response
1	User presses the "Performance Report" button.	System directs user to the assignment performance report page.
2	User presses the "student" drop down menu to select the student they wish to see the report.	System displays all the currently available users.
3	User presses "View Student Assignment" button and/or "View Avg" button.	System displays the student's performance in the selected assignment and/or the student's average overall performance by displaying a graph.
4	User presses the right arrow button.	System displays the campaign performance report page.

5	User selects the desired student, world number, and section, by pressing the available drop down menus.	System displays all the available students, world number, and sections.
6	User presses the "display student" button.	System shows all the available users in the selected world and section.
7	User presses "Avg World Score" button and/or user presses "Avg Sect Score" button.	System shows the average score of the selected world and/or system shows the average score of the selected section by displaying a graph.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using drop down menus and radio buttons / tick boxes.			

Post-Conditions	
1	User is directed to the main menu.
2	User's assignments are assigned to the desired class groups based on class IDs.

4. Gameplay Test Cases

4.1. Main Gameplay Loop Test Case

4.1.1. Main Gameplay Loop (Successful)

Test Case ID: #15	Test Case Name: Main Gameplay Loop
Subsystem: Gameplay	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the game with successful result..	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the system.
3	User has selected a character.
4	User is in the main menu.

Steps	Action	Expected System Response
1	Student enters the game screen.	Gameplay screen is displayed with two characters (one represents the user and the other one enemy). screen. Question and four answers are generated. Timer is initialized.
2	Student inputs the correct answer.	Student's character hits the enemy. Next question is displayed.

3	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.
4	Student answers all questions within the time limit.	Game ends with 'Stage cleared' message. Student gets reward in the form of the stars.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	Student is directed to popup window that indicates successful completion of the game.

4.1.2. Main Gameplay Loop (Unsuccessful)

Test Case ID: #16	Test Case Name: Main Gameplay Loop
Subsystem: Gameplay	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the game with unsuccessful result.	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the system.
3	User has selected a character.
4	User is in the main menu.

Steps	Action	Expected System Response
1	Student enters the game screen.	Gameplay screen is displayed with two characters (one represents the user and the other one enemy). screen. Question and four answers are generated. Timer is initialized.
2	Student inputs the correct answer.	Student's character hits the enemy. Next question is displayed.
3	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.
4-1	Student does not answers all questions	Game ends with 'Stage failed'

	within the time limit.	message. Student does not get a reward.
4-2	Student times out.	Game ends with 'Stage failed' message. Student does not get a reward.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	Student is directed to popup window that indicates unsuccessful completion of the game.

4.1.3. Main Gameplay Loop (Successful + Use Ability)

Test Case ID: #17	Test Case Name: Main Gameplay Loop - ability usage
Subsystem: Gameplay	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the game and uses special ability.	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the system.
3	User has selected a character.
4	User is in the main menu.

Steps	Action	Expected System Response
1	Student enters the game screen.	Gameplay screen is displayed with two characters (one represents the user and the other one enemy). screen. Question and four answers are generated. Timer is initialized.
2	Student inputs the correct answer.	Student's character hits the enemy. Next question is displayed.
3	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.

4-1	Students plays 'Escanor' and uses ability.	Heal ability is used. Time increased by 10 seconds.
4-2	Students plays 'Athena' and uses ability.	Shield ability is used. User blocks one damage.
4-3	Students plays 'Zeus' and uses ability.	Zap ability is used. One question option is removed randomly.
4-4	Students plays 'Mjolnir' and uses ability.	Gamble ability is used. User might skip to the next question (50% chance).
5	Student answers all questions or dealt with them using special ability within the time limit.	Game ends with 'Stage cleared' message. Student gets reward in the form of the stars.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	Student is directed to popup window that indicates successful completion of the game.

4.1.4. Main Gameplay Loop (Successful + Use Ability)

Test Case ID: #18	Test Case Name: Main Gameplay Loop - ability usage
Subsystem: Gameplay	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the game and uses special ability more than once.	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the system.
3	User has selected a character.
4	User is in the main menu.

Steps	Action	Expected System Response
1	Student enters the game screen.	Gameplay screen is displayed with two characters (one represents the user and the other one enemy). screen. Question and four answers are generated. Timer is initialized.
2	Student inputs the correct answer.	Student's character hits the enemy. Next question is displayed.
3	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.

4-1	Students plays 'Escanor' and uses ability.	Heal ability is used. Time increased by 10 seconds.
4-2	Students plays 'Athena' and uses ability.	Shield ability is used. User blocks one damage.
4-3	Students plays 'Zeus' and uses ability.	Zap ability is used. One question option is removed randomly.
4-4	Students plays 'Mjolnir' and uses ability.	Gamble ability is used. User might skip to the next question (50% chance).
5	Student uses ability more than once.	Ability is not active. Nothing happens.
6	Student answers all questions or dealt with them using special ability within the time limit.	Game ends with 'Stage cleared' message. Student gets reward in the form of the stars.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	Student is directed to popup window that indicates successful completion of the game.

4.2. Campaign Gameplay Test Case

Test Case ID: #19	Test Case Name: Campaign Gameplay
Subsystem: Gameplay	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020

Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the game in campaign mode.	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the system.
3	User has selected a character.
4	User is in the main menu.

Steps	Action	Expected System Response
1	Student clicks on 'Campaign' option.	World selection screen is displayed.
2	User chooses the world and clicks on that picture.	User is redirected to level selection screen.
3	User clicks on chosen level.	Gameplay is loaded with two characters (one represents the user and the other one enemy). screen. Question and four answers are generated. Timer is initialized.
4	Student inputs the correct. answer.	Student's character hits the enemy. Next question is displayed.
5	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.
6	Time is up.	Game ends with 'Stage Fail' message.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	System will display a popup window that indicates the failure of the game.

4.3. Assignment Gameplay Test Case

Test Case ID: #20	Test Case Name: Assignment Gameplay
Subsystem: Gameplay	Test Priority: High
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the game as an assignment..	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the game.
3	User has selected a character.
4	User is in the 'Assignments' section

Steps	Action	Expected System Response
1	Student chooses an assignment from the list and clicks on it.	Gameplay is loaded with two characters (one represents the user and the other one enemy). screen. Assignment question and four answers are generated. Timer is initialized.
2	Student inputs the correct. answer.	Student's character hits the enemy. Next question is displayed.
3	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.
4	Time is up.	Game ends with 'Stage Fail' message.

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	System will display a popup window that indicates the failure of the game.

4.4. Custom Level Gameplay Test Case

Test Case ID: #21	Test Case Name: Custom Level Gameplay
Subsystem: Gameplay	Test Priority: Medium
Test Designed By: Team NanyangGuys	Design Date: 21/4/2020
Test Executed By: Kamil Florowski	Execution Date: 22/4/2020
Description: Student plays the custom levels..	

Pre-Conditions	
1	Application is connected to the internet.
2	User is successfully logged in to the game.
3	User has selected a character.
4	User is in the 'Custom Levels' section

Steps	Action	Expected System Response
1	Student chooses a custom level from the list and clicks on 'Play' button.	Gameplay is loaded with two characters (one represents the user and the other one enemy). screen. Custom question and four answers are generated. Custom timer is initialized.
2	Student inputs the correct. answer.	Student's character hits the enemy. Next question is displayed.
3	Student inputs the wrong answer.	Student's character is hit by the enemy. Timer additionally goes down by 10s. The question keeps showing.

4	Time is up.	Game ends with 'Stage Fail' message.
---	-------------	--------------------------------------

Test Value		Description	Expected Result	Actual Result
#	No test values are given as all the input fields are restricted to only the valid options, using buttons, drop down menus, and radio buttons / tick boxes.			

Post-Conditions	
1	System will display a popup window that indicates the failure of the game.

5. Test Cases Summary Report

Work Pack	AdventureLearn Black Box Testing
Planned Date for Testing	22/4/2020
Actual Date for Testing	23/4/2020
Total Test Case Executed	21
Number of Test Case Passed	46
Number of Test Case Failed*	14
Bugs in Progress	32
Bugs Waiting Approval	0
Number of Closed Bugs	0
Coverage	100%
Remarks	

* The number of test cases failed was of those cases that were meant to fail.

6. Appendix A: Miscellaneous White Box Testing (Legacy)

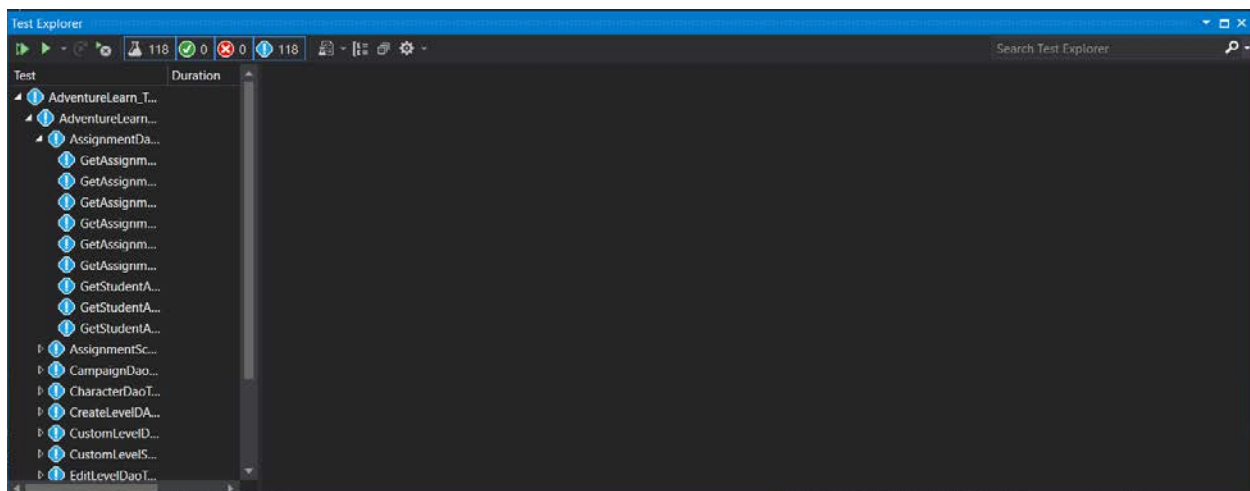
White box testing was primarily done to test the data access objects in the data access layer. Most of the tests encompass the valid and invalid inputs for the DAO methods.

The drivers use a popular open-source automated unit testing framework for Microsoft .NET programming languages called **NUnit**. This was done by installing several additional plugins to the VS development environment (**NUnit**, **NUnit adapter**, **NUnit console runner**).

[LEGACY TESTING RESULTS]

Please check the most recent .sln project source code for unit testing of all updated methods. Same testing principles apply and the type of tests are specified in the comments.

Current testing:



Legacy testing:

Methods (Scripts)	Parameters	Expected Results	Actual Results	Additional Remarks
StudentDao: CheckStudentExist_IsSuccessful()	StudentName = "sample" StudentPassword = "123"	Student found	Student found	Test passed (username valid, password valid)
	StudentName = "invalid" StudentPassword = "123"	No student found	No student found	Test passed (username invalid, password valid)

	StudentName = "Sample"	No student found	No student found	Test passed (username valid, password invalid)
	StudentPassword = "invalid"	No student found	No student found	Test passed (username invalid, password invalid)
StudentDao: UpdateStudentCharacter_IsSuccessful()	StudentNewChar = 1 StudentId = 1	Update successful	Update successful	Test passed (charID valid, studentID valid)
	StudentNewChar = -1 StudentId = 1	Update fails	Update fails	Test passed (charID invalid, studentID valid)
	StudentNewChar = 1 StudentId = -1	Update fails	Update fails	Test passed (charID valid, studentID invalid)
	StudentNewChar = -1 StudentId = -1	Update fails	Update fails	Test passed (charID invalid, studentID invalid)
AssignmentScoreDAO: GetAvgAssignmentScores_IsSuccessful()	AssignmentID = 1	Scores obtained	Scores obtained	Test passed (assignmentID valid)
	AssignmentID = 0	No results	No results	Test passed (assignmentID invalid)
	AssignmentID = -1	No results	No results	Test passed (assignmentID invalid)
AssignmentScoreDAO: GetMaxAssignmentScores_IsSuccessful()	AssignmentID = 1	Scores obtained	Scores obtained	Test passed (assignmentID valid)
	AssignmentID = 0	No results	No results	Test passed (assignmentID invalid)
	AssignmentID	No results	No results	Test passed

	= -1			(assignmentID invalid)
AssignmentScoreDAO: GetMinAssignmentScores_IsSuccessful()	AssignmentID = 1	Scores obtained	Scores obtained	Test passed (assignmentID valid)
	AssignmentID = 0	No results	No results	Test passed (assignmentID invalid)
	AssignmentID = -1	No results	No results	Test passed (assignmentID invalid)
AssignmentScoreDAO: GetStudentAssignmentScore_IsSuccessful()	StudentID = 1	Scores obtained	Scores obtained	Test passed (studentID valid)
	StudentID = 0	No results	No results	Test passed (studentID invalid)
	StudentID = -1	No results	No results	Test passed (studentID invalid)
CustomLevelDao: GetStudentCustomLevel_IsSuccessful()	StudentID = 1	Level obtained	Level obtained	Test passed (studentID valid)
	AssignmentID = 0	No results	No results	Test passed (studentID invalid)
	StudentID = -1	No results	No results	Test passed (studentID invalid)
CustomLevelDao: GetCustomLevel()	CustomLevelID = 1	Level obtained	Level obtained	Test passed (level ID valid)
	CustomLevelID = 0	No results	No results	Test passed (level ID invalid)
	CustomLevelID = -1	No results	No results	Test passed (level ID invalid)
CampaignDao: GetLevel()	WorldID = 1 SectionID = 1 LevelID = 1	Campaign level obtained	Campaign level obtained	Test passed (worldID valid, sectionID valid, levelID valid)
	WorldID = 1	No level obtained	No level obtained	Test passed (worldID valid,

	SectionID = 1 LevelID = -1			sectionID valid, levelID invalid)
	WorldID = 1 SectionID = -1 LevelID = 1	No level obtained	No level obtained	Test passed (worldID valid, sectionID invalid, levelID valid)
	WorldID = 1 SectionID = -1 LevelID = -1	No level obtained	No level obtained	Test passed (worldID valid, sectionID invalid, levelID invalid)
	WorldID = -1 SectionID = 1 LevelID = 1	No level obtained	No level obtained	Test passed (worldID invalid, sectionID valid, levelID valid)
	WorldID = -1 SectionID = 1 LevelID = -1	No level obtained	No level obtained	Test passed (worldID invalid, sectionID valid, levelID invalid)
	WorldID = -1 SectionID = -1 LevelID = 1	No level obtained	No level obtained	Test passed (worldID invalid, sectionID invalid, levelID valid)
	WorldID = -1 SectionID = -1 LevelID = -1	No level obtained	No level obtained	Test passed (worldID invalid, sectionID invalid, levelID invalid)

CharacterDao: GetCharacter_IsSuccessful()	StudentID = 1	Character obtained	Character obtained	Test passed (studentID valid)
	StudentID = 0	No student found	No student found	Test passed (studentID invalid)
	StudentID = -1	No student found	No student found	Test passed (studentID invalid)

InsertAssignment_IsSuccessful()							
TeacherID	Assignment name	DueDate	MonsterID	TimeLimit	Questions	Expected Results	Actual Results
1	"Lorem ipsum"	10/5/2020	1	60	List<Questions>	Assignment inserted	Assignment inserted
-1	"Lorem ipsum"	10/5/2020	1	60	List<Questions>	Insertion failed (invalid teacherID)	Insertion failed (invalid teacherID)
1	-1	10/5/2020	1	60	List<Questions>	Insertion failed (invalid assignment name)	Insertion failed (invalid assignment name)
1	"Lorem ipsum"	"Lorem ipsum"	1	60	List<Questions>	Insertion failed (invalid due date)	Insertion failed (invalid due date)
1	"Lorem ipsum"	10/5/2020	-1	60	List<Questions>	Insertion failed (invalid monsterID)	Insertion failed (invalid monsterID)
1	"Lorem ipsum"	10/5/2020	1	-1	List<Questions>	Insertion failed (invalid time limit)	Insertion failed (invalid time limit)
1	"Lorem ipsum"	10/5/2020	1	60	Question	Insertion failed (expected a list, given a single question)	Insertion failed (expected a list, given a single question)

InsertCustomLevel_IsSuccessful()

StudentID	CustomLevelName	MonsterID	TimeLimit	PublicLevel	Questions	Expected Results	Actual Results
1	"Lorem ipsum"	1	60	1	List<Questions>	Assignment inserted	Assignment inserted
-1	"Lorem ipsum"	1	60	1	List<Questions>	Insertion failed (invalid teacherID)	Insertion failed (invalid teacher ID)
1	-1	1	60	1	List<Questions>	Insertion failed (invalid assignment name)	Insertion failed (invalid custom level name)
1	"Lorem ipsum"	1	60	1	List<Questions>	Insertion failed (invalid due date)	Insertion failed (invalid monster ID)
1	"Lorem ipsum"	-1	-1	1	List<Questions>	Insertion failed (invalid monsterID)	Insertion failed (invalid time limit)
1	"Lorem ipsum"	1	60	-1	List<Questions>	Insertion failed (invalid time limit)	Insertion failed (invalid public level)
1	"Lorem ipsum"	1	60	1	Question	Insertion failed (expected a list, given a single question)	Insertion failed (expected a list, given a single question)