AdventureLearn Game Application
Test Plan

Version 1.0
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	23/4/2020	Nicholas Irving	24/4/2020	Initial version

1. Introduction

This document serves to outline multiple stages of testing that will be performed on AdventureLearn before its release/submission. There will be two main stages for the tests:

1. White Box Test Stage
2. Black Box Test Stage

The white box test stage can be performed while the system is not yet fully completed. This phase primarily focuses on unit testing key data access methods in the data access layer in order to check whether all interactions with the database are functioning as per intended. Due to some limitations in the Godot engine, a separate “tester” project needs to be created as the main AdventureLearn project is classified by Visual Studio as a “class library” project. This severely limits the extent of unit testing that can be done.

The black box test stage is to be done after the application is close to the release version. This phase will be significantly more comprehensive than the previous. Our team will be testing all the features of the application to detect bugs, user-friendliness, and possible dips in performance.

2. Features to be tested

2.1 Student's primary

Feature	Importance	Reason
Login	High	Without login student cannot access AdventureLearn and therefore competition online is impossible.
Character Selection	High	Without character selection user cannot play.
View Profile	High	This option allows user to see statistics and also to change character.
Change Character	Medium	There is possibility that user might want to change character as different characters have different special ability.
View Leaderboard	High	This is one of the core features.
View Custom Levels	Medium	User might want to view levels that others created.
View Created Levels	Medium	User might want to view his own levels.
View Assignments	High	This is one of the core features.
View Credits	Low	User might want to see credits.
Settings	Medium	Student might want to change settings such as sound volume.

2.2 Level Creation/Editing

Feature	Importance	Reason

Student Create/Edit level	Medium	Student might want to create levels in order to challenge friends with them.
Teacher Create/Edit assignments	High	This is the core feature of AdventureLearn.

2.3 Teacher's primary

Feature	**Importance**	**Reason**
Publish assignments	High	This is the core feature of AdventureLearn.
Check Summary Report	High	This is the core feature of AdventureLearn.

2.4 Gameplay

Feature	**Importance**	**Reason**
Main Gameplay	High	Gameplay is the centre section of the game.
Campaign Gameplay	Medium	User might want to play a campaign. This feature does not include online competition.
Assignment Gameplay	High	This is the core feature of AdventureLearn. User must be able to play and accomplish assignments sent by teacher.
Custom Gameplay	Medium	It is not the most important feature in terms of education on the teacher-student basis. However this option enables competition between students.

3. Features not to be Tested

As observed in the previous section, all features will be tested.

4. Testing Approach

Our application will go through two levels of testing: Unit, and Acceptance, being performed during the white box testing stage and black box testing stage respectively.

4.1. Unit Testing

As mentioned in the introduction, unit testing will be done as a part of the white box testing stage to ensure that key data access methods are functioning as intended. A separate Visual Studio "tester" project will be created.

NUnit, an open source unit testing framework for .NET Framework/Core and Mono applications, will be installed in the tester project. For each of the key data access objects in AdventureLearn's data access layer, a corresponding data access **test object** will be created in the tester project. These test objects subsequently contain **tester methods** that will call the original data access object's methods and check whether they are properly functioning.

For better illustration, see table below.

AdventureLearn DA Method	Tester Project DA Test Method	Test Outcome
`CustomLevelDao.GetCustomLevel()`	`CustomLevelDaoTest.GetCustomLevel_IsSuccessful()`	`valid`
	`CustomLevelDaoTest.GetCustomLevel_IsNotSuccessful()`	`invalid`

Do note that not all data access methods will be tested. Some methods do not require any parameters as input and some others have relatively low priorities. Due to time constraints, only key data access methods will be unit tested. The unit testing will be done by some members of our team.

4.2. Acceptance Testing

This level of testing is done to check whether all features of the application, which corresponds to the various project requirements, are working as per intended. Once the application is nearing its final release, an **internal acceptance testing** will be conducted, where members of the group cross-check all the available features with the project requirements and test their functionalities using black-box methods to be outlined in the next section.

Since the program lacks real-world client/shareholders, customer acceptance testing will not be conducted.

5. Item Pass/Fail Criteria

Test results:

- 21 - tests executed
- 46 - number of test case passed
- 14 - number of test case failure

Feature	**Pass criteria**	**Fail criteria**
Login	User is able to log in and access main menu screen.	User fails to log in.
Character Selection	Student is able to select character.	Student fails to choose any character.
View Profile	Student is able to view his own profile along with statistics.	Student cannot access view profile option or account data fails to download.
Change Character	Student is able to change character.	Student is not able to change character
View Leaderboard	Leaderboard can be accessed from student account.	Leaderboard cannot be accessed or data fails to download.
View Custom Levels	Student is able to see custom levels that others created.	Student fails to see levels that others created
View Created Levels	Student is able to see his own previously created levels.	Student cannot access his own levels.
View Assignments	Student is able to see assignments sent by the teacher.	Students fails to see the list consists of assignments.
View Credits	Student is able to view credits.	Student is not able to view credits.
Settings	Student is able to enter settings and change things such a sound volume.	Student is not able to enter settings or features such as sound volume cannot be

		changed.
Student Create/Edit Level	Student is able to successfully create their own custom level(s), with valid questions and answers.	Student's custom level(s) fails to be created even with all valid input fields.
Teacher Create/Edit Assignment	Teacher is able to successfully create assignment(s), with valid questions and answers.	Teacher's assignment(s) fails to be created even with all valid input fields.
Publish assignments	Teacher is able to create and publish assignments.	Teacher fails to publish assignments.
View summary report	Teacher is able to view the summary report.	Student is not able to generate the report or data are not properly downloaded.
Main Gameplay	Student is able to play the game.	Student is not able to play the game.
Campaign Gameplay	Student is able to successfully enter the campaign mode, select the world, choose level and play the game.	Student is not able to enter the campaign mode or gameplay screen is not properly displayed.
Assignment Gameplay	Student is able to firstly view assignments and then play the chosen one.	Students fails to enter assignments or assignment gameplay is not properly displayed.
Custom Gameplay	Student is able to enter custom levels mode and then play the chosen one.	Students either fails to enter the custom level section or custom gameplay is not properly displayed.

6. Environmental Needs

6.1. White Box Testing Stage

Members of the development team will use their personal device to conduct the white box testing phase. The following application and NuGet plugins are required:

1. Visual Studio
2. NUnit version 3.10 and above
3. NUnit3TestAdapter version 3.10 and above

The tester project needs to have access to all functionalities of the main AdventureLearn project. Hence, **object references** need to be established between the two projects before any test fixtures can be run. Additionally, the tester project needs to have the **GodotSharp.dll** and **GodotSharpEditor.dll** referenced as well.

All the test fixtures are to be found within Visual Studio's **test explorer**.

6.2. Black Box Testing Stage

In this phase, members of our group will install the near-finalised version of the application's .apk file. This can be done using each member's android device, or an android emulator on their personal computers. Members will explore all the application's functionalities and test whether each feature is in line with the project requirements.

Feedbacks will then be delivered to the lead developer and/or project manager for further improvements to the application.

7. Training Needs

The testing team should be familiar with the application, its features and objectives. Team is required to get familiar with environmental needs from the paragraph above.

8. Schedule

As testing is the last stage of project development, test plan will be executed in the week commencing on 20th of April.