# Testing Document

| Test Case | Test Case Description | Test Case Results |
|---|---|---|
| Opens Scenario Text File | This test checks if the authoring app has the ability to open and read the specific scenario text file for reading and further modifications | **Pass** |
| Save Scenario Text File | This test checks if the authoring app can save the specified scenario text file and overwrite changes to the edited content | N/A |
| Create New Scenarios | This test checks if the authoring app has the ability to create new scenarios, populate the fields representing options with default text which can later be edited to personalize the scenario to user choosing | N/A |
| Add New Event | This test checks if the scenario file can have a new event added in any sequence into the file, formatted correctly and neatly organized | N/A |

## Discussion – How these Test Cases were Derived and their Sufficiency

1. Opening the scenario file and reading the text file is one of the main tasks for this project as if file is mishandled while being imported, data corruption could incur causing the scenario file to misbehave and cause inconvenience to the end user. This test case was derived keeping the end-user in mind for providing user-friendly experience where everything is smooth from the get go.

2. Saving a scenario file is equally important as opening a file as the scenario file is used only after it has been saved successfully. This test is meant to check for formatting issues that might or might not occur during the saving process to provide the playing app with the correct sequence of lines which are formatted properly. This test case was derived as a case to cover up the defects of saving files that might be read-only or may not be able to be overwritten. This test case should provide the user with a scenario file being created no matter the circumstances where the end-user doesn't need to go through technical troubles of overwriting a file and losing previous work or not being able to save to a file due to read-only property of files being true.

3. Creating new scenarios is another integral part of the authoring app as it provides the user with a scenario file from the get go with a dummy scenario file to get started with. Although the process might not be appealing, this provides the user to test out each feature of the scenario file and modify to their liking as they please. This test case was derived for user-friendliness of the features to the end-users as having to learn a new software is a hard task especially if it handles working with creating scripts for external devices that cannot be easily tested.

4. Adding new events is a feature that allows the ability to add a state in the script where something happens of important. This allows the user to perhaps ask a question (an event) which can be triggered at a stage of the program, and provides the user the ability to add many important features to the scenario file. This is important because it provides the handler with different things to work with for example, each question could be an event and asked in their own space. This test was derived from the requirements of the authoring app as it was needed to provide the end-user to add events into the scenario file app.