Olde Worlde Phunne – Maze Game

Test Plan 29/07/2019

Version 1.0

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Software Application:**  Olde Worlde Phunne – Maze Game | | | **Name of Tester:**  Matthew Sawrey | | **Date:**  22/01/2018 |  |
| **Test No** | **Requirement** | **Test Case** | **Purpose/Description of Test** | **Test Data** | **Expected Result** | **Actual Result** | **Discrepancies** |
| 1 |  | On clicking the “Upload new document/s” button, the Upload New Documents form appears. | Ensuring that the application is usable and does not present any errors on use. | - | The document uploader sidebar form opens and presents to the user correctly in position and layout. |  |  |
| 2 |  | On clicking the “click to upload” form element, windows explorer correctly opens. | Ensuring that the application is usable and does not present any errors on use. | - | Windows explorer opens on click. |  |  |
| 3 |  | One or many files can be selected for upload from windows explorer. | Determining that the document uploader is capable of receiving multiple files for upload at once. | - | All files selected through windows explorer are present in the document uploader selected or invalid files list. |  |  |
| 4 |  | Dragging and dropping a file from window into the “drag and drop file” page element recognises and selects these files for upload. | Determining that the “drag and drop” area of the document uploader is working as expected. | - | All files selected through drag and drop process present in the document uploader selected or invalid files list. |  |  |
| 5 |  | When files have been selected for upload from either windows explorer selection or dragging and dropping in the relevant page element, all selected files and recognised by the uploader. | Determining that the document uploader recognises all files that have been selected for upload, and is not incorrectly filtering any files out of the selection. | - | All files selected through drag and drop process present in the document uploader selected or invalid files list. |  |  |
| 6 |  | Selected files are filtered correctly by file size. | Checking that the file size criteria check is being performed correctly. | Two files. One with a size equal to or greater than 20MB and one with a size less than 20MB. | Files of a size greater than the predefined maximum file size of 20 MB are filtered onto the invalid files list. Those that are less than 20MB are added to the selected files list. |  |  |
| 7 |  | Selected files are filtered correctly by file type. | Checking that the file type criteria check is being performed correctly. | Multiple files of every file type on the file type whitelist, and many files of file types not on the whitelist. | Files of a file type that is not permitted on the file type whitelist are filtered onto the invalid files list. Those that are on the file type whitelist are added to the selected files list. |  |  |
| 8 |  | Selected files are filtered correctly by existence within the selected files list. | Checking that the file existence check is being performed correctly. | A file that has previously been added to the selected files list. | Files that are already present on the selected files list are filtered onto the invalid files list when selected for upload a second time. |  |  |
| 9 |  | Files added to the selected files list that do not pass the filter criteria are added to the Invalid Files list. | Demonstrating that the Invalid Files list is being correctly populated with filtered-out files. | - | Files that are filtered out against the filter criteria are all added to the invalid files list. |  |  |
| 10 |  | File display names can be edited before upload. | Demonstrating that the user requirement to be able to change a file’s display name before uploading the file is satisfied. | New file name input. | File display name edit functionality works and can be edited before upload. |  |  |
| 11 |  | File category can be edited before upload. | Demonstrating that the user requirement to be able to change a file’s category before uploading the file is satisfied. | New file category input. | File category edit functionality works and can be edited before upload. |  |  |
| 12 |  | Pressing the upload button when files are ready for upload, uploads the files sequentially, returning a positive green tick against each file selected for upload. | Demonstrating that the upload process that sequentially posts all files to the appropriate API endpoint is working as expected. | - | File posts to the API work and return successful HTTP requests that then become green tick icons next to each file until the whole list of selected files has been uploaded. |  |  |
| 13 |  | When a batch of files has been uploaded, they are removed from the selected files list. | Demonstrating that the user requirement to be able to upload multiple batches of files from the document uploader without the form closing has been satisfied. | - | Once the whole selected files list has been uploaded successfully, the selected files list is reset to blank and new files can be uploaded. |  |  |
| 14 |  | A second batch of files can be selected for upload and uploaded successfully after uploading the initial batch. | Demonstrating that the user requirement to be able to upload multiple batches of files from the document uploader without the form closing has been satisfied. | Multiple files of many file types and sizes. | The document uploader works just as it initially did on the first upload when selecting files for subsequent uploads. The selected and invalid files lists are reset correctly, and the upload methods – windows explorer selection and drag and drop – both work correctly. |  |  |
| 15 |  | Pressing the “Close Form” button closes the document uploader. | Demonstrating that the form closes and the user is able to navigate away from the form without error. | - | The document uploader form closes and does not cause any problems with the originating page. |  |  |