**Test Cases**

**Test Case ID**: 1

**Product Module**: UploadCallBack

**Purpose**:

Create window for the Uploader to select a file and give feedback.

**Assumptions**:

Tkinter is installed.

**Preconditions**:

Uploader button on home page is selected.

**Steps**:

After Uploader button is pressed, a new window is created, the old window is not deleted. The new window will have 3 buttons. Press any of them to test further modules.

**Expected Outcomes**:

Old window will not be deleted. New window is created with 3 buttons: Browse, Create Audio and Feedback.

**Actual Outcomes**:

Old window is deleted. New window is created with 2 buttons: Browse and Feedback.

**Post Conditions**:

UploadCallBack is used to transition between selecting to be an Uploader and actually uploading a file.

**Result**:

SUCCESS

**Test Case ID**: 2

**Product Module**: BrowseCallBack

**Purpose**:

Allow user to select file, extract text from image, and display the text on the window generated in UploadCallBack

**Assumptions**:

Tkinter, pytesseract are installed.

**Preconditions**:

Browse button on upload window is selected.

**Steps**:

Open a file dialog box. Allow the file to be selected. Open the image and extract text from it. Display this extracted text on the window.

**Expected Outcomes**:

Dialog box is opened. Extracted text is displayed on upload window. Error message if incompatible file format selected.

**Actual Outcomes**:

Dialog box is opened. Extracted text is displayed on upload window. Error message if incompatible file format selected.

**Post Conditions**:

BrowseCallBack allows us to select any file and displays the output after performing the necessary OCR on the image.

**Result**:

SUCCESS

**Test Case ID**: 3

**Product Module**: FeedbackCallBack

**Purpose**:

Collect data on accuracy of optical character recognition.

**Assumptions**:

Tkinter is installed.

**Preconditions**:

Feedback button on upload page is selected.

**Steps**:

After Feedback button is pressed, a new window is created, the old window is deleted. The new window will have a text entry box in which the user can enter the actual text in the image.

**Expected Outcomes**:

The average accuracy of the algorithm is stored.

**Actual Outcomes**:

The average accuracy of the algorithm is stored.

**Post Conditions**:

FeedbackCallBack is used to determine the usefulness of the OCR algorithm by calculating the accuracy of the algorithm over test data.

**Result**:

SUCCESS

**Test Case ID**: 4

**Product Module**: SubmitCallBack

**Purpose**:

Submit button for feedback window.

**Assumptions**:

Tkinter is installed.

**Preconditions**:

Enter button on feedback window is selected.

**Steps**:

After the enter button is pressed, the feedback is stored. It is compared with the extracted text and accuracy is stored.

**Expected Outcomes**:

Accuracy of algorithm is stored.

**Actual Outcomes**:

Accuracy of algorithm is stored.

**Post Conditions**:

SubmitCallBack is used to store accuracy of the algorithm

**Result**:

SUCCESS

**Test Case ID**: 5

**Product Module**: AudioCallBack

**Purpose**:

Allow extracted text to be converted to audio. Create an mp3 file, play audio generated

**Assumptions**:

gtts and pyttsx are installed.

**Preconditions**:

Create Audio button on Uploader window is selected.

**Steps**:

After the writer button is pressed, a canvas opens. Write a letter clearly in the space and press done. Then press RunOCR.

After the Create Audio button is pressed, current window is closed and Audio window opens. There are two buttons in this window. ‘Save mp3 file’ and ‘Play mp3 file’. These two buttons will either save the audio generated from extracted text or play it respectively.

**Expected Outcomes**:

Button pressed:

Text to Speech Conversion done.

Mp3 file generated.

Mp3 file saved.

Mp3 file played.

Quit: Exit the application.

**Actual Outcomes**:

Button pressed:

Text to Speech conversion function called.

Mp3 file generated.

Mp3 file saved.

Mp3 file played.

Quit: Exit the application.

**Post Conditions**:

AudioCallBack is used to convert extracted text to speech/audio.   
 CreateAudio saves the mp3 file in the same folder as the application.   
 PlayAudio plays the mp3 file generated from the extracted text.

**Result**:

SUCCESS

**BLACK BOX TESTING**

**Purpose**:

Extract printed text from an uploaded image.

**Assumptions**:

Necessary modules and dependencies are installed. Python 2.7 or a later version is used on a Linux platform.

**Preconditions**:

The python script is run from command line.

**Steps**:

Follow the instructions on each window i.e. upload the image.

**Expected Outcomes**:

Entire text extracted from image.

**Actual Outcomes**:

If the size of the writing is moderately sized, and clearly written, 100% accuracy, else slightly lower accuracy

**Post Conditions**:

Feedback (accuracy) of the extracted text is given to the user

**Result**:

SUCCESS

**Justification**:

Lower accuracy can be expected at times when the uploaded image has a lot of noise other than text.