Testing Plan -

Speech Capture, Transcription, and Analysis App

Sponsor – Dr. Zoran Djordjevic

Reed Cummings – Group Leader

Mario Jelev

Eric Rogers

James Lussier

Abstract

In this document we will go over our plan to test the product against the use cases we found in requirements.

TABLE OF CONTENTS

List of Figures 4

List of Tables 5

Introduction 6

Requirements Based Test Cases 7

Traceability of Test Cases to Use Cases 8

Techniques for Test Generation 10

Evidence the Requirements have been Places Under Config Mgmt. 11

Additional Notes 12

References 13

LIST OF FIGURES

LIST OF TABLES

Introduction

The purpose of this document is to explain how we are planning to test the software. We will re-iterate some of the requirements and look to see if we reached them with our implementation. We will also test against our use cases to see if we get desired outputs, as well as test to see what happens if the system is used improperly. Lastly, we will go over techniques we used for our testing.

Requirement Based System Level Test Cases

Functional Requirements:

The product shall capture speech from speaker-

We have accomplished this by using Amazon transcribe. We have tested different rates of speech, and different distances from the microphone.

The product shall create an .wav file from the audio captured

We have accomplished this again, by using a web based API.

The product shall have an upload button to upload mp3 file

We have met this requirement.

The product shall transcribe the .wav file into a txt file

The .wav file is transcribed, and the user has an option to download the file as a .txt file

The product shall create a use case from a template used by the interviewer

The product creates a rough template, given a very strict prompt by the interviewer.

The product shall create a brief summary of the speech

The product uses the translated text and creates a NPL based summary

The product shall detect new speakers and add a break in the page

The product detects speakers, we are working on adding break

The product shall be able to search through past transcriptions for key phrases

Have not gotten to this yet

Non-Functional:

The product shall be able to be used with less than 10 minutes of training.

We will ask random, technological illiterate people to use the software after giving them brief instructions

The product shall give analysis within 2 minutes of choosing to analyze file.

We have given the analysis function files that were created form nearly an hour of recorded speech to files that have about 30 seconds of speech and both are analyzed in under 2 minutes

Traceability of Test Cases to Use Cases

Test Case 1: Record Audio

Step1: Click on Record button, see if it starts recording.

Step2: Click on the stop button, see if site stops recording audio.

Step3A: Click button to download file OR:

Step3B: Click the button to begin transcription to see if it produces a new screen.

Alternate1 Step 1: Click on the record button.

Alternate1 Step2: Click the back button on the browser.

Alternate1 Step3: Go back to record page to see if the page resets itself

Alterante2 Step1: Click record

Alternate2 Step2: Attempt to click button to transcribe audio file. Nothing should happen

|  |  |
| --- | --- |
| **Use Case ID** | UC-1 |
| **Use Case Name** | Speech Capture |
| **Primary Actor** | User |
| **Entry Condition** | Clicked on Record File from home screen |
| **Exit Condition** | Failure/System Begins Transcription/User goes back to home screen |
| **Actor’s Goal** | To capture the speech into an audio file |
| **Priority** | Critical |

***Main Success Scenario***

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor clicks record button | System starts recording audio |
| **2** | Actor clicks stop recording button | System stops recording audio |
| **3** | Actor clicks next button to begin transcription | System begins transcription and produces a new screen. |

***Alternative Success Scenario 1: Download File/Record Again***

**Trigger Condition**: Actor wants to download file to local system and record again

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor clicks stop recording button | System stops recording audio |
| **2** | Actor clicks button to download audio file | System produces audio file in downloads folder |
| **3** | Actor clicks back button to go back to home screen | System takes you back to home screen |

***Alternative Success Scenario 2: Download File/Transcribe***

**Trigger Condition**: Actor wants to download file to local system and begin transcription

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor clicks stop recording button | System stops recording audio |
| **2** | Actor clicks button to download audio file | System produces audio file in downloads folder |
| **3** | Actor clicks next button to begin transcription | System begins transcription and produces a new screen. |

Test Case2: Upload Audio File

Step1: User clicks the select file button, a window should appear to choose file

Step2: User selects a file and confirms; the user should be taken to a new screen.

A1S1: User clicks select file button

A1S2: User Selects a file and confirms

A1S3: User clicks back button while transcribing, pops a dialog box up making sure you want to confirm.

A2S1: User clicks select file button

A2S2: User unable to select an incompatible file, system only shows MP3 and .wav files

A3S1: User clicks select file button

A3S2: User selects a file and confirms it

A3S3: System goes to sleep while transcribing, wake it up.

|  |  |
| --- | --- |
| **Use Case ID** | UC-2 |
| **Use Case Name** | Upload Audio File |
| **Primary Actor** | User |
| **Entry Condition** | Clicked on “Select File” from the upload screen |
| **Exit Condition** | Failure/System begins transcription |
| **Actor’s Goal** | User uploads an audio file for analysis |
| **Priority** | Critical |

***Main Success Scenario***

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor clicks the “Select file” button | System prompts the actor to choose a file from their file system explorer with an audio file format |
| **2** | Actor selects the audio file they want to use in their file system explorer | The user is taken to a “Transcription in Progress” page to await the transcription to be complete. |

***Alternative Success Scenario 1: Invalid or corrupt file***

**Trigger Condition**: Step 2 user input an invalid or corrupt file to the system

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor submits an invalid or corrupt file for transcription | System begins file transcription, recognizes the file as invalid or corrupt, and returns an error to the actor prompting them to try another file. |
| **2** | Actor goes back to step 1 of the Main Success Scenario |  |

Test Case 3: Upload a File for Analysis

Step1: User clicks button to analyze file

Step2: User chooses file to analyze, taken to new screen

Alternate Step 2: User chooses a file of the not correct type

Alternate Step 3: Click back during analysis

|  |  |
| --- | --- |
| **Use Case ID** | UC-3 |
| **Use Case Name** | Upload Transcription |
| **Primary Actor** | User |
| **Entry Condition** | Clicked on Upload Transcription from home screen |
| **Exit Condition** | An analysis of the uploaded transcription is completed |
| **Actor’s Goal** | Upload a pre-existing text file for analysis |
| **Priority** | High |

***Main Success Scenario***

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor clicks the “Select file” button | System prompts the actor to choose a file from their file system explorer with a text file format |
| **2** | Actor selects the text file they want to use in their file system explorer | The user is taken to an “Analysis in Progress” page to await the analysis to be complete. |

***Alternative Success Scenario 1: Invalid or corrupt file***

**Trigger Condition**: Step 2 user input an invalid or corrupt file to the system

|  |  |  |
| --- | --- | --- |
| **Step** | **Actor Action** | **System Response** |
| **1** | Actor submits an invalid or corrupt file for analysis | System begins file analysis, recognizes the file as invalid or corrupt, and returns an error to the actor prompting them to try another file. |
| **2** | Actor goes back to step 1 of the Main Success Scenario |  |

Techniques for Test Generation

* GUI performance testing
  + We will test to make sure all buttons work, and layout is mostly pleasing to the eye.
* White box testing
  + Developers are the ones testing, so we know the internal structure of the program.
* Does It Work testing
  + Our quality of testing is based on the fact if a specific function performs the way it is supposed to.

Evidence the Test Cases Have Been Placed Under Configuration Management

<https://github.com/reedpcummings/SeniorDesign-SpeechCapture>

Additional Notes:

Our testing plan is based off of things we are creating and implementing, and not pieces of software developed by other users.

For example, we test our interaction with AWS transcribe, but not on the transcription itself. We have added, and will continue to add, little things here and there to try and improve what we can of our co-existence with AWS products and other APIs.

References