Backend Facial Analysis

Test Case Specification

Date: 12/10/21

Test Strategy/Approach

Project Outline: User input Video File Name and the Path to save the images -> convert video file to Frames -> convert Frames to Streams -> use Microsoft Face API to detect the Emotion -> count the emotion that happened the most times and return the emotion (String)

Unit Testing (Focuses on the internal processing logic and data structures)

- Test if the input file is .mp4.
- Test if the path that the user entered is valid.
- Test error handling when the user input is invalid

Integration Testing (Focus on modules)

Use bottom-up integration testing

We start testing with each lowest-level function.

- Test function (convert the video file to Frames, save frames to specific location)
- Test function (convert Frames to Streams)
- Test function (connect to Microsoft Face API)
- Test function (detect emotion)

Validation Testing

- Testing the whole project when the developers are not present
- Perform the test at the front end

System Testing

Performance testing:

- Test run-time performance of the project in converting video to Frames.
- Test run-time performance of the project in detecting the emotion

Test Plan

Risks and Assets:

1. The person’s face is not very clear in the input video, therefore, it cannot be detected precisely

Features to be Tested:

The functions that will be tested are:

1. ConvertVideoToFrames
2. ConvertFramesToStreams

The features that are tested:

1. Detect the overall emotion of the video
2. Connecting the code to the Microsoft Face API

Features not to be Tested:

One of the function of VideoToImage class are not testes:

1. VideoToStreams - because it is implied in ConvertVideoToFrames and ConverFramesToVideo functions tests (if they work, this will function too)

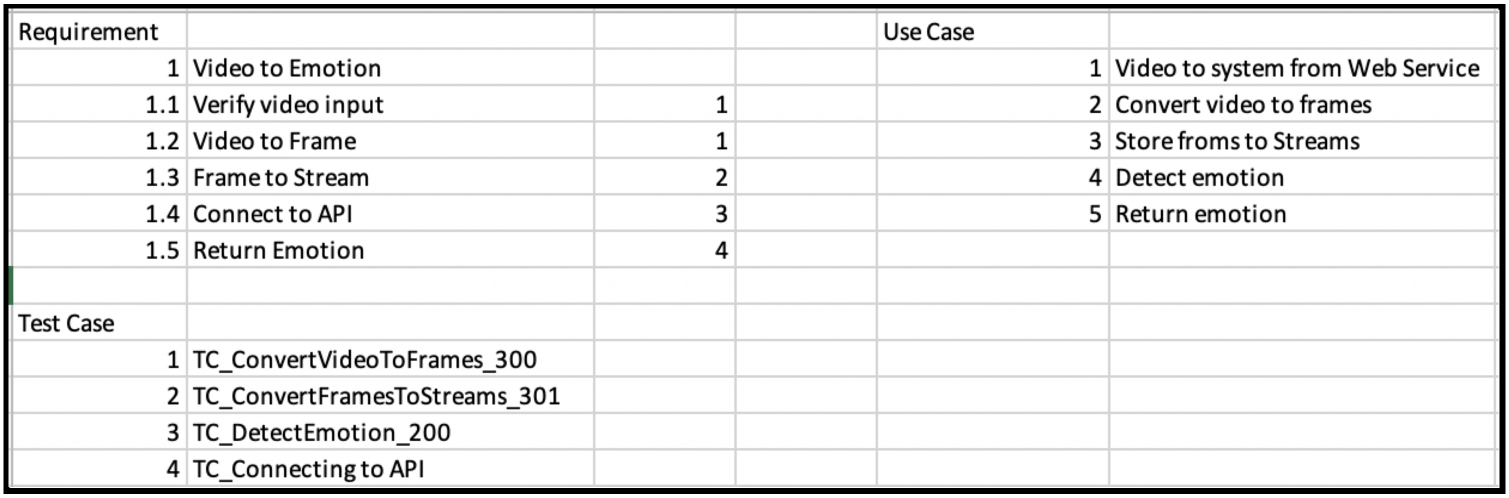
Some functions of VideoToImage class are not testes:

1. FindMostReoccuringElement - because it is a private function
2. Authenticate - because it is a private function
3. GetHeaviestEmotion - because it is a private function
4. GetReplyText - because it is a private function
5. MoodType - because it is a private function

Test Case Specification:

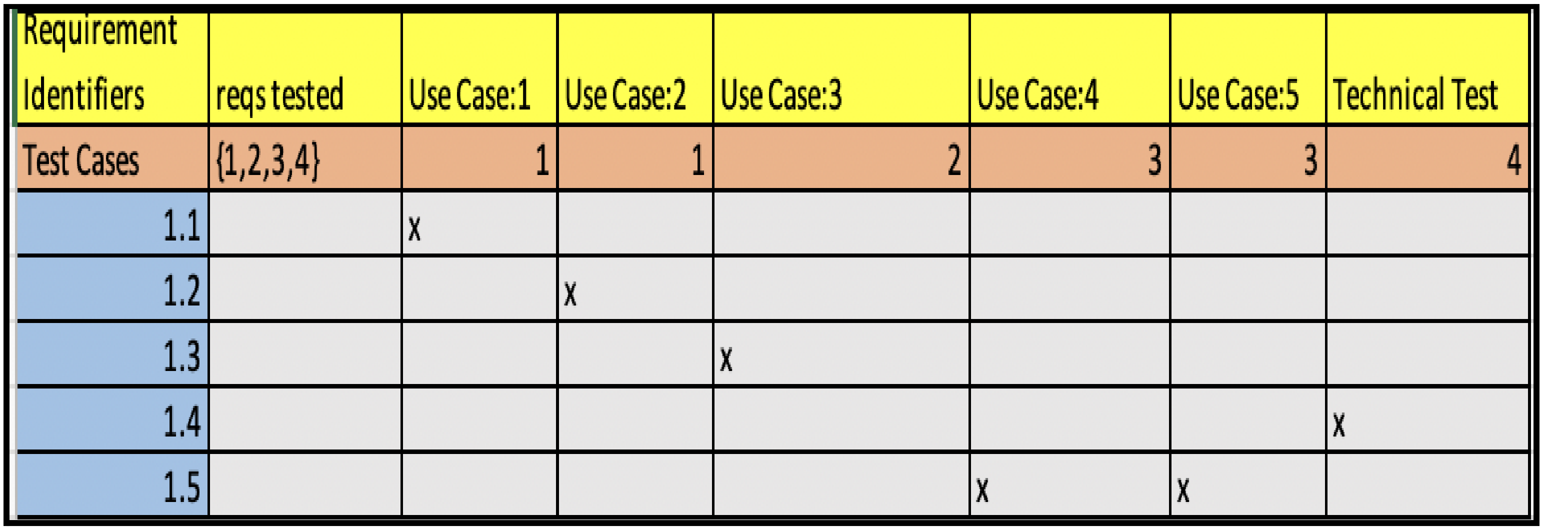
- Testing the program to see if the functions of the classes work properly
  - Test the ConvertFramesToVideo function to make sure frames are produced and saved from the video
    - # of test ~ 30 until no bugs occur
  - Test the ConvertFramesToVideo function to make sure streams are produced from the saved frames as the EmotionReader class is only able to read streams
    - Test until all frames are saved in an array of streams
  - Test EmotionReader class for the DetectFaceExtract function to make sure correct overall emotion is returned
    - # tests ~ 16 times (four for each type of emotion)
  - Connecting the code to Microsoft Face API
    - Test it until it gets connected to the API

Requirements for Traceability Matrix



Requirement				Use Case	
1	Video to Emotion			1	Video to system from Web Service
1.1	Verify video input	1		2	Convert video to frames
1.2	Video to Frame	1		3	Store froms to Streams
1.3	Frame to Stream	2		4	Detect emotion
1.4	Connect to API	3		5	Return emotion
1.5	Return Emotion	4			
Test Case					
1	TC_ConvertVideoToFrames_300				
2	TC_ConvertFramesToStreams_301				
3	TC_DetectEmotion_200				
4	TC_Connecting to API				

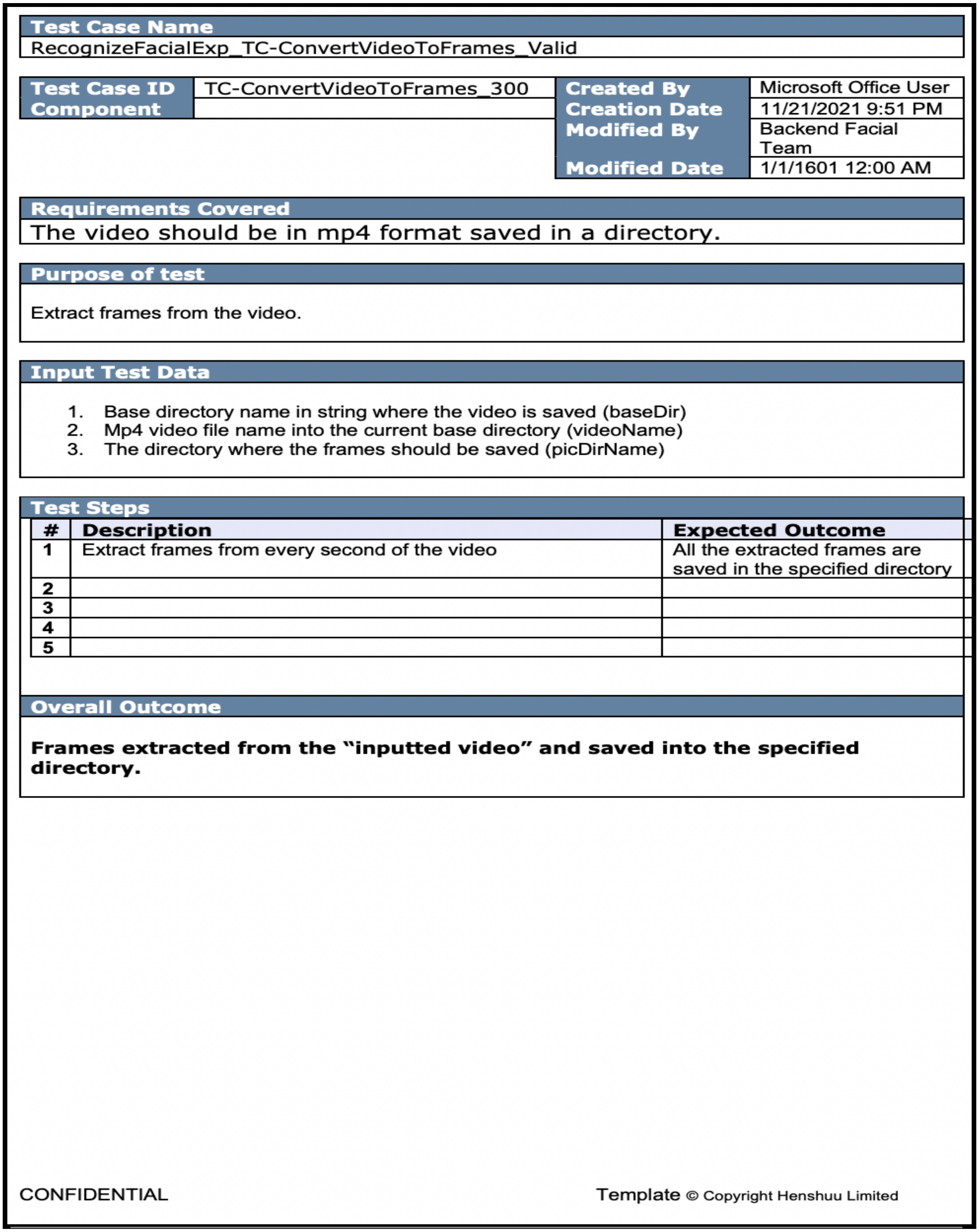
Traceability Matrix



Requirement Identifiers	reqs tested	Use Case:1	Use Case:2	Use Case:3	Use Case:4	Use Case:5	Technical Test
Test Cases	{1,2,3,4}	1	1	2	3	3	4
1.1		x					
1.2			x				
1.3				x			
1.4							x
1.5					x	x	

TEST CASES

Test Case 1



Test Case Name
RecognizeFacialExp_TC-ConvertVideoToFrames_Valid

Test Case ID	TC-ConvertVideoToFrames_300	Created By	Microsoft Office User
Component		Creation Date	11/21/2021 9:51 PM
		Modified By	Backend Facial Team
		Modified Date	1/1/1601 12:00 AM

Requirements Covered

The video should be in mp4 format saved in a directory.

Purpose of test

Extract frames from the video.

Input Test Data

1. Base directory name in string where the video is saved (baseDir)
2. Mp4 video file name into the current base directory (videoName)
3. The directory where the frames should be saved (picDirName)

Test Steps

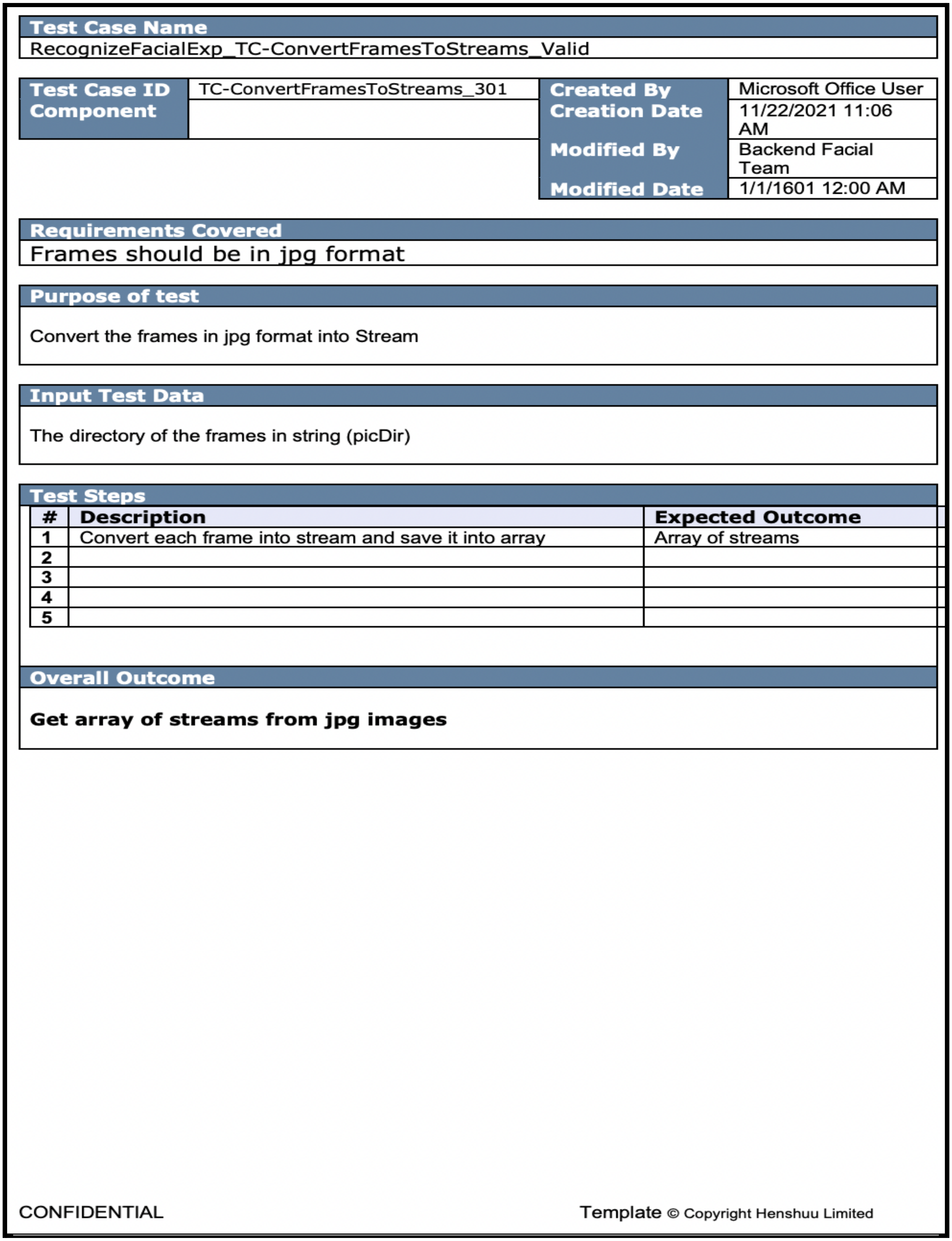
#	Description	Expected Outcome
1	Extract frames from every second of the video	All the extracted frames are saved in the specified directory
2		
3		
4		
5		

Overall Outcome

Frames extracted from the "inputted video" and saved into the specified directory.

CONFIDENTIAL

Test Case 2



Test Case Name
RecognizeFacialExp_TC-ConvertFramesToStreams_Valid

Test Case ID	TC-ConvertFramesToStreams_301	Created By	Microsoft Office User
Component		Creation Date	11/22/2021 11:06 AM
		Modified By	Backend Facial Team
		Modified Date	1/1/1601 12:00 AM

Requirements Covered
Frames should be in jpg format

Purpose of test
Convert the frames in jpg format into Stream

Input Test Data
The directory of the frames in string (picDir)

Test Steps

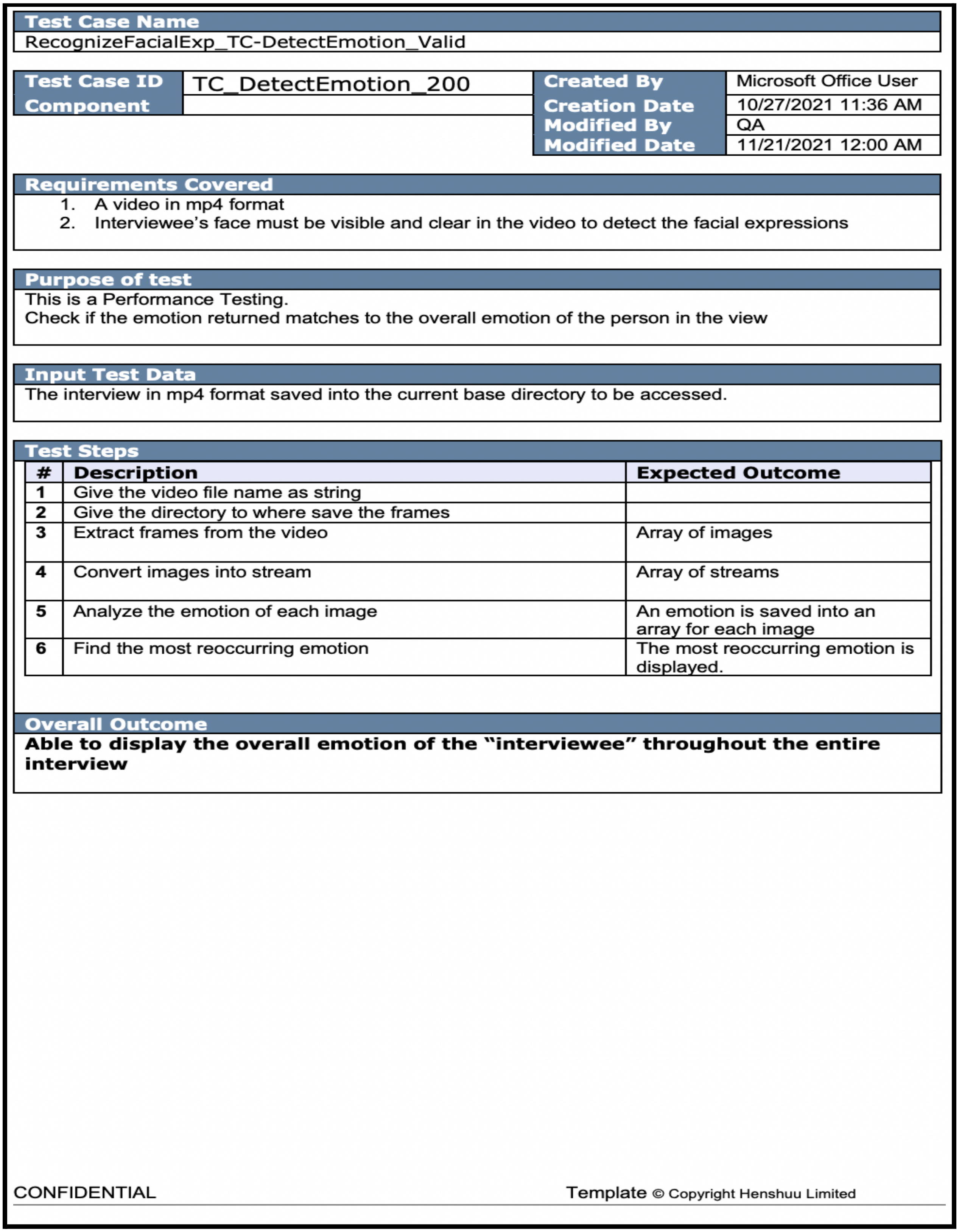
#	Description	Expected Outcome
1	Convert each frame into stream and save it into array	Array of streams
2		
3		
4		
5		

Overall Outcome
Get array of streams from jpg images

CONFIDENTIAL

Template © Copyright Henshuu Limited

Test case 3



Test Case Name
RecognizeFacialExp_TC-DetectEmotion_Valid

Test Case ID	TC_DetectEmotion_200	**Created By**	Microsoft Office User
Component		**Creation Date**	10/27/2021 11:36 AM
		Modified By	QA
		Modified Date	11/21/2021 12:00 AM

Requirements Covered

1. A video in mp4 format
2. Interviewee's face must be visible and clear in the video to detect the facial expressions

Purpose of test

This is a Performance Testing.
Check if the emotion returned matches to the overall emotion of the person in the view

Input Test Data

The interview in mp4 format saved into the current base directory to be accessed.

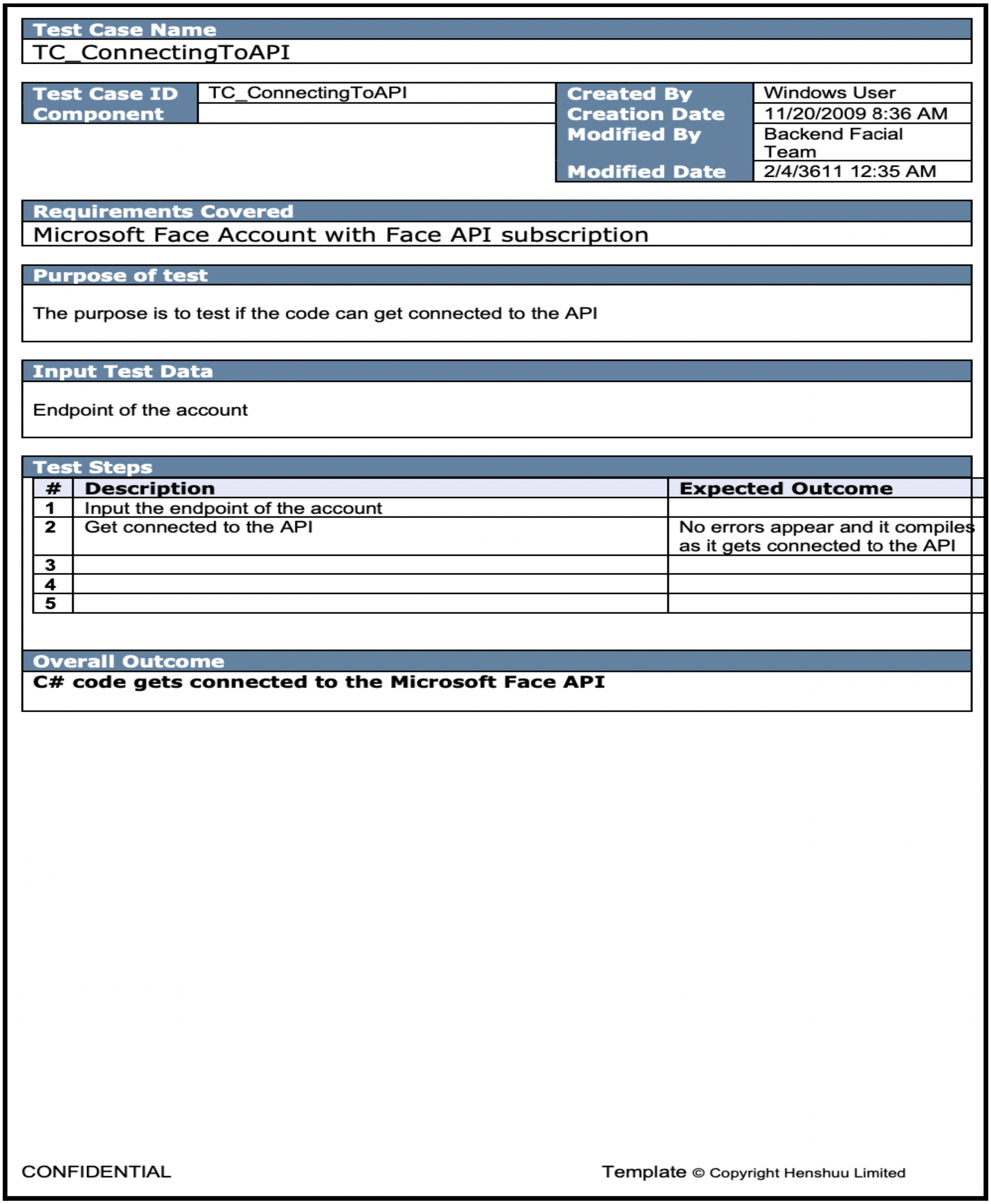
Test Steps

#	Description	Expected Outcome
1	Give the video file name as string	
2	Give the directory to where save the frames	
3	Extract frames from the video	Array of images
4	Convert images into stream	Array of streams
5	Analyze the emotion of each image	An emotion is saved into an array for each image
6	Find the most reoccurring emotion	The most reoccurring emotion is displayed.

Overall Outcome

Able to display the overall emotion of the "interviewee" throughout the entire interview

CONFIDENTIAL



Test Case Name

TC_ConnectingToAPI

Test Case ID	TC_ConnectingToAPI	**Created By**	Windows User
Component		**Creation Date**	11/20/2009 8:36 AM
		Modified By	Backend Facial Team
		Modified Date	2/4/3611 12:35 AM

Requirements Covered

Microsoft Face Account with Face API subscription

Purpose of test

The purpose is to test if the code can get connected to the API

Input Test Data

Endpoint of the account

Test Steps

#	Description	Expected Outcome
1	Input the endpoint of the account	
2	Get connected to the API	No errors appear and it compiles as it gets connected to the API
3		
4		
5		

Overall Outcome

C# code gets connected to the Microsoft Face API

CONFIDENTIAL

Test Results Summary

Test Case 1:

- Converting video to frames – checks if frames are produced from the input video [successful]

Test Case 2:

- Converting frames to streams – checks if all the frames are converted into stream as the EmotionReader can only read streams [successful]

Test Case 3:

- Detect emotion – checks if an overall emotion is returned at the end of the process [successful]

Test case 4:

- Connecting to Microsoft Face API – checks if the code correctly gets connected to the Microsoft Face API account [successful]