Awais Imran

01-134132-034

Junaid Ahmad Sheikh

01-134132-074

CS16-113

FQVRS

Facial Image Query Based Video Retrieval System

Bachelor of Science in Computer Science

Supervisor : Mr. Ghulam Ali Mirza

Department of Computer Science Bahria University, Islamabad

May 14, 2017

Certificate

We accept the work contained in the report titled "Facial Image Query Based Video Retrieval System", written by Mr. Awais Imran and Mr. Junaid Ahmad Sheikh as a confirmation to the required standard for the partial fulfillment of the degree of Bachelor of Science in Computer Science.

Approved by . . . :
Supervisor: Mr. Ghulam Ali Mirza(Assistant Professor)
. ____
Internal Examiner: Ms. Momina Moestesum(Senior Lecturer)
. ____
External Examiner: Name of the External Examiner(Title)
. ____
Project Coordinator: Dr. Arif-ur-Rahman(Assistant Professor)
. ____
Head of Department: Dr. Faisal Bashir(Associate Professor)
. ____

May 15th, 2017

“There is a tide in the affairs of men, Which taken at the flood, leads on to fortune. Omitted, all the voyage of their life is bound in shallows and in miseries. On such a full sea are we now afloat. And we must take the current when it serves, or lose our ventures.”

- William Shakespeare

Abstract

This project is made for the purpose of the facial image based video retrieval on a Microsoft Windows OS. This project is intended for Pakistan-based news channels which will assist them in logging and video retrievals for future use. The project is also applicable for intelligence agencies for surveillance purposes. Initially, the application extracts key frames from a news video. After the extraction of frames, further processing is applied, which includes face detection and logo detection. After the detection phase, the frames are indexed onto a data file and then the data file is used for Face recognition, matching the detected faces from the video, to the ones detected in the input image query. Finally, the relevant video is displayed from the instance a face matches the one from the input image and the news channel logo is displayed.

Acronyms and Abbreviations

FQVRS	Facial Image Query Based Video retrieval System
GUI	Graphical User Interface
IDE	Integrated Development Environment
OS	Operating System
TC	Test Case
UC	Use Case

Contents

List of Figures

1 Introduction

1.1 Objective

The objective of this chapter is to describe the background objectives scope and modules of our project. The chapter provides an introduction to the techniques and methodologies used in the development of the system.

1.2 About the System

Our project, Facial Image Query Based Video Retrieval System (FQVRS), is a software that will revolutionize video retrievals in the news channels of Pakistan. FQVRS is a software that will take a facial image as a query and retrieve a video from the datasets which feature the person in the query image. Using this system, the news channels in Pakistan will be able to provide background information, if any exists in their datasets, about the person that they are going to feature on their news channel. Furthermore, the system will also have a module that will identify the news channel on which the news was aired. This will be accomplished by detecting and recognizing the logo of the channel which is usually displayed at the bottom corners of every news video.

1.3 Problem Description

Our system is intended to extract key frames from a video clip and apply facial detection and recognition on the images to find relevant videos to the person (in the query image). The extraction of clear images from a video is a challenge due to the low resolution of videos and complex backgrounds and multiple extracted angles of the face, some other challenges may include recognizing a face in a frame of multiple faces. The dissimilarities between the images of the same face due to viewing direction and illumination is a challenge to overcome. A particular person can appear absolutely different when light source direction and viewpoints change as well as a rotation of face image leads to very large changes in the appearance of a face. To recognize a face we have to achieve a common ground for both the images to be matched (Input image and the extracted Image), that common ground being the frontal face image, we have to take Input of a front facing image or one should have to generate the front facing features from the non-front facing image. This goes for both, the Input image and the retrieved image from the video. The same technique should be used for logo detection to provide the name of the channel that aired the news.

1.4 Project Scope

Our system is intended to find a person provided as a query image to the system, by detecting faces from a video clip, taken from a news channel and then recognize the faces to provide the video clips (news featuring that person) as an output to the user. Our system will detect and recognize video images as well as video scene images. The system will also provide the name of the News Channel that owns the video by detecting and recognizing the logo of the news channel, that is displayed (commonly) in the corner of the video clip. The system will be able to find news video clips of the person that the user wants to research about just by providing a query image of the person. We will perform the search on the downloaded videos that we have saved in our machine. Extracting/Recording videos from websites and TV channels is NOT a part of our project scope. Recognizing multiple person through one query image is also NOT a part of our project scope.

2 Literature Review

2.1 Related Work

There are a lot of research papers available on the Internet regarding content based video retrieval. We studied five of the research papers which were most relevant to the development of our system. The work performed in those papers are summarized below.

2.1.1 Research Papers

"Content-Based Video Indexing and Retrieval using Key Frames Texture, Edge and Motion Features" [5] published by M. Ravinder and T. Venugopal, suggests a novel algorithm for content-based video indexing and retrieval using key-frames texture, edge, and motion features is presented. The algorithm extracts key frames from a video using extraction of texture, k-means clustering based method, motion, and edge features to represent a video with the feature vector. The algorithm is evaluated on a database of a large amount of videos (collected from TRECVID 2005, Google, and BBC) of four types. The performance is compared with volume local binary patterns (VLBP) method. The proposed algorithm outperforms well compare to VLBP method.

"Research and Advanced Technology for Digital Library" [8] by the authors Wolfgang Nejdl, Norbert Fuhr, Thomas Risse, and Laszlo Kovacs, suggests that for content based video retrieval the aim is to basically assign semantic tags automatically to video shots. The videos are preprocessed first (shot boundary detection and thumb generation). The objective of shot boundary detection is the segmentation of the video in shots. The following analysis algorithms are applied: similarity search, person and text recognition, concept classification. The meta-data obtained as result is then used to list the matching videos and organized according to the probability of the result which is desired.

Alex P. Pentland and Matthew A. Turk published "Face Recognition Using Eigenfaces" [4] . These writers explained the recognition of faces uses eigenfaces. Eigenfaces approach is used to this date for facial recognition. The paper suggests that face recognition is a visual problem of a very high level. They proposed a strategy for recognition based on an information theory approach, seeking to encode the information which is relevant to a group of faces which will distinguish them effectively from each other. The approach transforms facial images into a set of characteristic feature image called "Eigenfaces", which are the components of the training set of facial images. Recognition is carried out by projecting a new image into subspace populated by the eigenfaces ("face space") and then classifying the face by comparing the position of the face in face space with the position of known individuals.

”View Based and Modular Eigenspaces for Face Recognition” published by the authors Alex Pentland, Baback Moghaddam, and Thad Turner is another paper [1] which describes the significance of view based modular eigenspaces in facial recognition. In this paper, the authors explore the eigenface technique of Turk and Pentland scales. Then generalizing the strategy to modular and view based eigenspaces for the purposes of detection and recognition. The formulation permits for recognition under changing head orientations and the modular description permits the incorporation of key facial features such as mouth, nose, and eyes. These extensions rationalize variations in object pose and guide to a more robust recognition software system.

In ”Content Based Video Retrieval System” [3] published by Madhav Gitte, Harshal Bawaskar, Sourabh Sethi and Ajinkya Shinde, the publishers are presenting Content-Based Video Retrieval System (CBVR). CBVR contains Feature Extraction, Keyframe Selection, Video Segmentation, and Indexing. For video retrieval, the subsystem processes the query which was presented executes similarity matching operations which may be carried out using Euclidian Distance Algorithm, and finally, presents the result to end user.

2.2 Conclusion

Eigenfaces approach is considered fundamental in face recognition systems, however, this approach does not prove to b much effective. Percentage of successful recognition of faces by using eigenspaces is very low as compared to using Local Binary Patterns (LBP) [6] . So, we are going to use Haar scaling for face detection and LBP for face recognition. Haar scaling for the detection of faces include cascading of simple features and facial recognition using LBP includes the extraction of features from both the query image and the segments extracted from the news videos present in the datasets. The extraction of features from the images (of faces) can be used to get the measure for the similarity between the images. The logo recognition phase will be done by using speeded up robust features (SURF) [2].

3 Requirement Specifications

3.1 System Overview and Scope

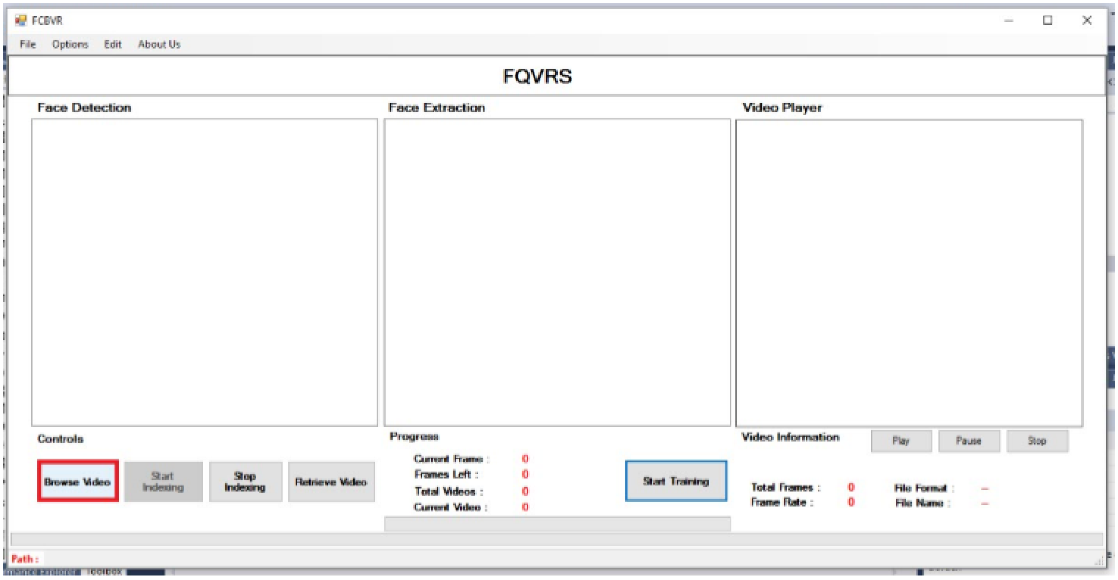
FQVRS will find news videos of a person (featured in the query image) specified by the user. FQVRS will take input a facial image as a query from the user and will display the news videos in which the person is featured in. The videos are taken from a dataset maintained by downloading news videos from news channels websites and other various sources. FQVRS will also dis-play the name of the channel on which the news video was aired. This will be achieved by detecting the logo of the news channel and then recognizing it by comparing it to the logos present in the dataset.

3.2 Existing System

We have performed our research where we found a lot of facial recognition software, but there exists none that performs these features on news channels video in Pakistan. Our system is one of its kind, just not because of its features and requirements to deal with complex backgrounds and low resolution of images and videos, but because of its application area.

3.3 Proposed System

The proposed system will be built in Microsoft Visual Studio using windows forms. Our system can be employed by any media outlet organization or private bloggers who keep interest in posting articles about the news that is trending on the TV or Web channels. Application area of our system also be used by the news channels and can also be used for surveillance purposes. Fig 3.1 provides the interface of the homepage of the proposed system.



Current Frame : 0
Frames Left : 0
Total Videos : 0
Current Video : 0

Total Frames : 0
Frame Rate : 0
File Format : –
File Name : –

Figure 3.1: Home page of FQVRS

3.4 Requirements

3.4.1 Functional Requirements

1. The system should be able to deal with complex backgrounds of the videos and the query image.
2. The system should be able to process at least three frames per second.
3. The systems should be able to put up with different orientation of faces.
4. The system should be able to recognize video images as well as video scene images.

3.4.2 Non-Functional Requirements

1. Extraction of features from image should not take more than 15 seconds.

2. Uploading a picture should not take more than 15 seconds.
3. The browsing window should open in lesser than 5 seconds.
4. In case of system crash, the reboot must not take more than 2 minutes.

3.5 Project Extends

The extends of our project include the following features:

1. Improving accuracy of face detection using classification and neural network techniques.
2. Retrieval of multiple videos from a single query image

3.6 Use Case Diagram

In Fig 3.2, the Use Case Diagram for FQVRS can be seen.

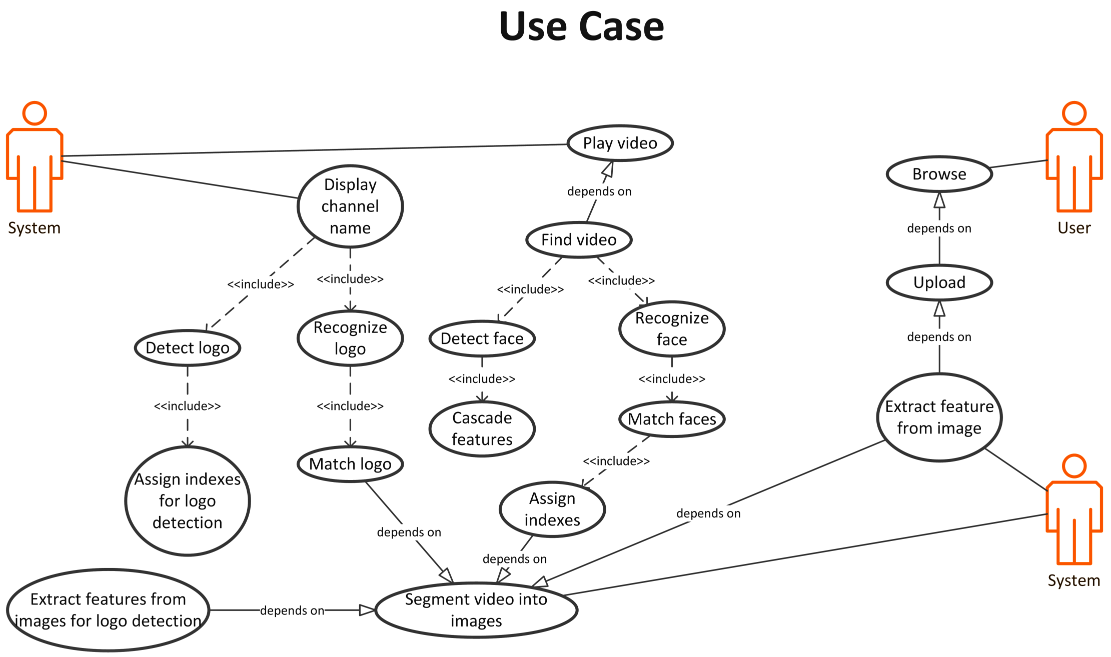
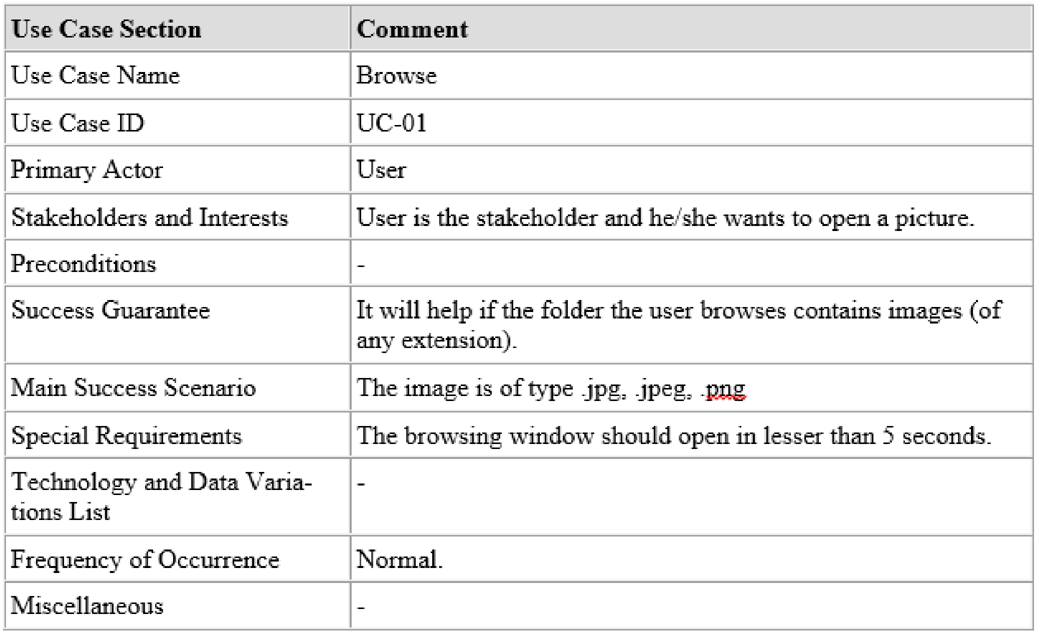


Figure 3.2: Use Case Diagram

3.7 Use Case Specification/ Fully Dressed Use Cases

In Fig 3.3, the use case for Browse Video can be seen.

Browse Video

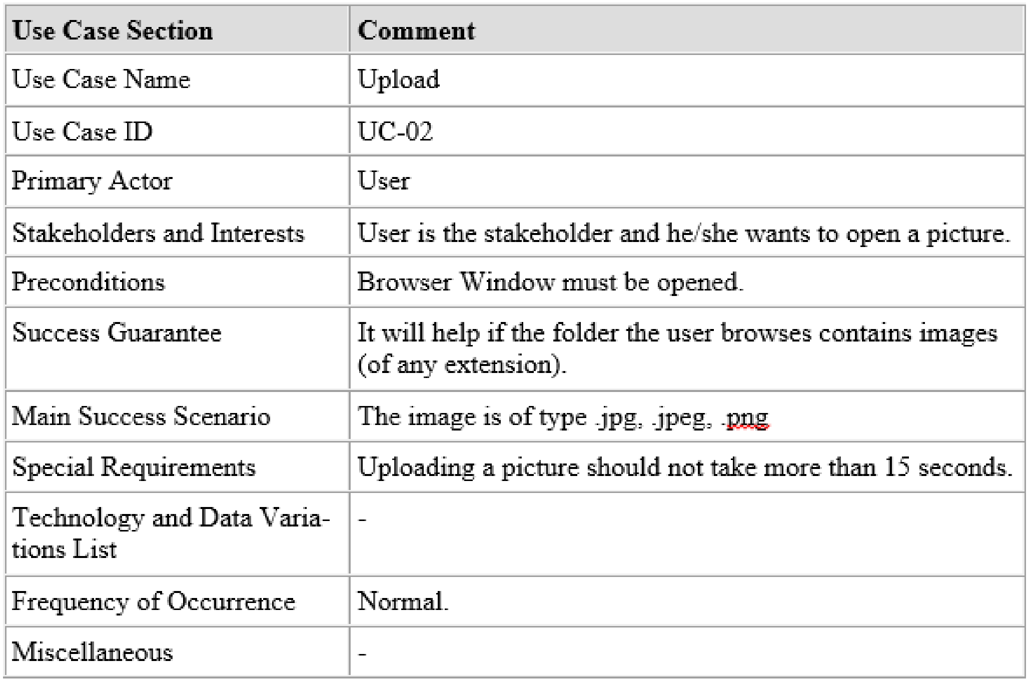


Use Case Section	Comment
Use Case Name	Browse
Use Case ID	UC-01
Primary Actor	User
Stakeholders and Interests	User is the stakeholder and he/she wants to open a picture.
Preconditions	-
Success Guarantee	It will help if the folder the user browses contains images (of any extension).
Main Success Scenario	The image is of type .jpg, .jpeg, .png
Special Requirements	The browsing window should open in lesser than 5 seconds.
Technology and Data Variations List	-
Frequency of Occurrence	Normal.
Miscellaneous	-

Figure 3.3: UC-01

In Fig 3.4, the use case for Upload Images can be seen.

Upload Images

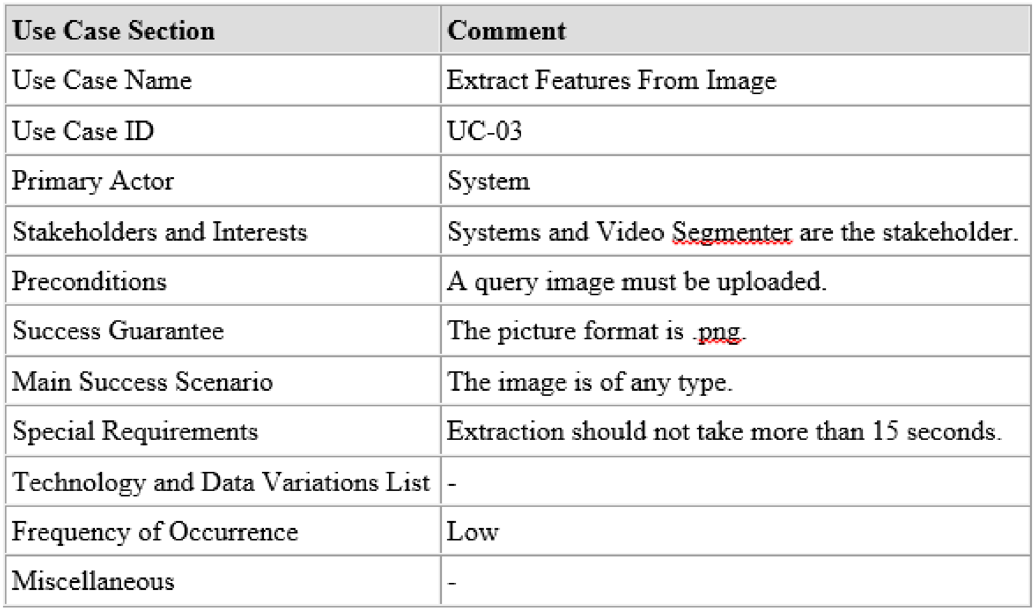


Use Case Section	Comment
Use Case Name	Upload
Use Case ID	UC-02
Primary Actor	User
Stakeholders and Interests	User is the stakeholder and he/she wants to open a picture.
Preconditions	Browser Window must be opened.
Success Guarantee	It will help if the folder the user browses contains images (of any extension).
Main Success Scenario	The image is of type .jpg, .jpeg, .png
Special Requirements	Uploading a picture should not take more than 15 seconds.
Technology and Data Variations List	-
Frequency of Occurrence	Normal.
Miscellaneous	-

Figure 3.4: UC-02

In Fig 3.5, the use case for Extract Features From Image can be seen.

Extract Features From Image

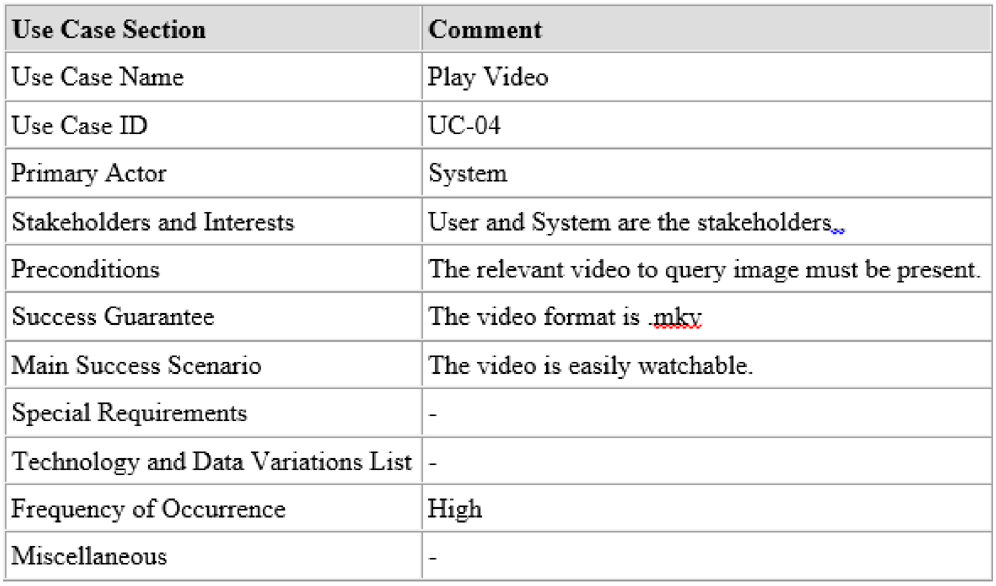


Use Case Section	Comment
Use Case Name	Extract Features From Image
Use Case ID	UC-03
Primary Actor	System
Stakeholders and Interests	Systems and Video Segmenter are the stakeholder.
Preconditions	A query image must be uploaded.
Success Guarantee	The picture format is .png.
Main Success Scenario	The image is of any type.
Special Requirements	Extraction should not take more than 15 seconds.
Technology and Data Variations List	-
Frequency of Occurrence	Low
Miscellaneous	-

Figure 3.5: UC-03

In Fig 3.6, the use case for Play Video can be seen.

Play Video

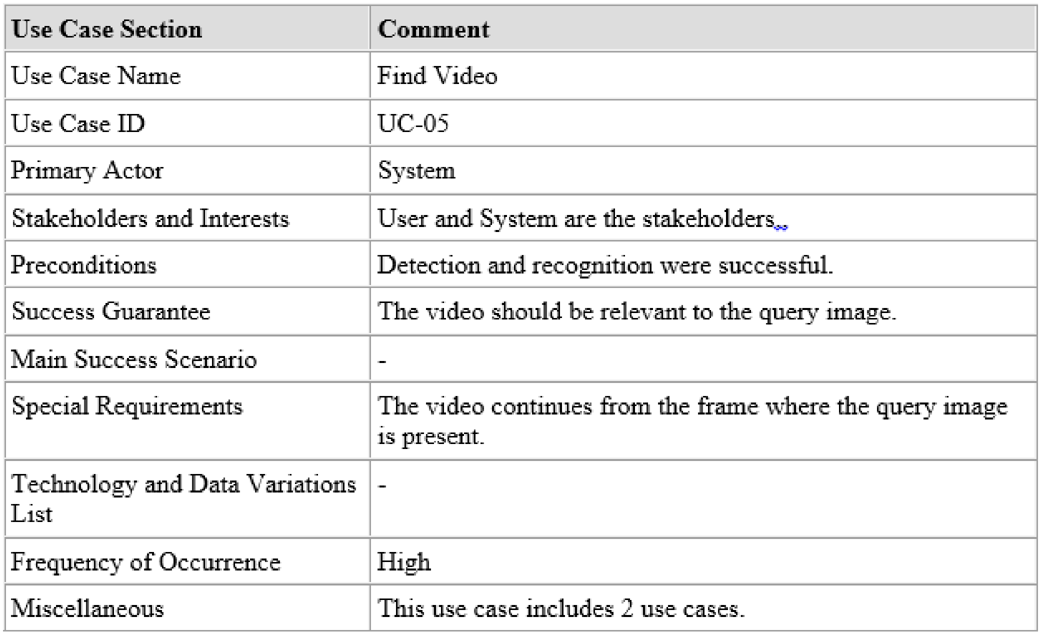


Use Case Section	Comment
Use Case Name	Play Video
Use Case ID	UC-04
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	The relevant video to query image must be present.
Success Guarantee	The video format is .mkv
Main Success Scenario	The video is easily watchable.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	High
Miscellaneous	-

Figure 3.6: UC-04

In Fig 3.7, the use case for Find Video can be seen.

Find Video

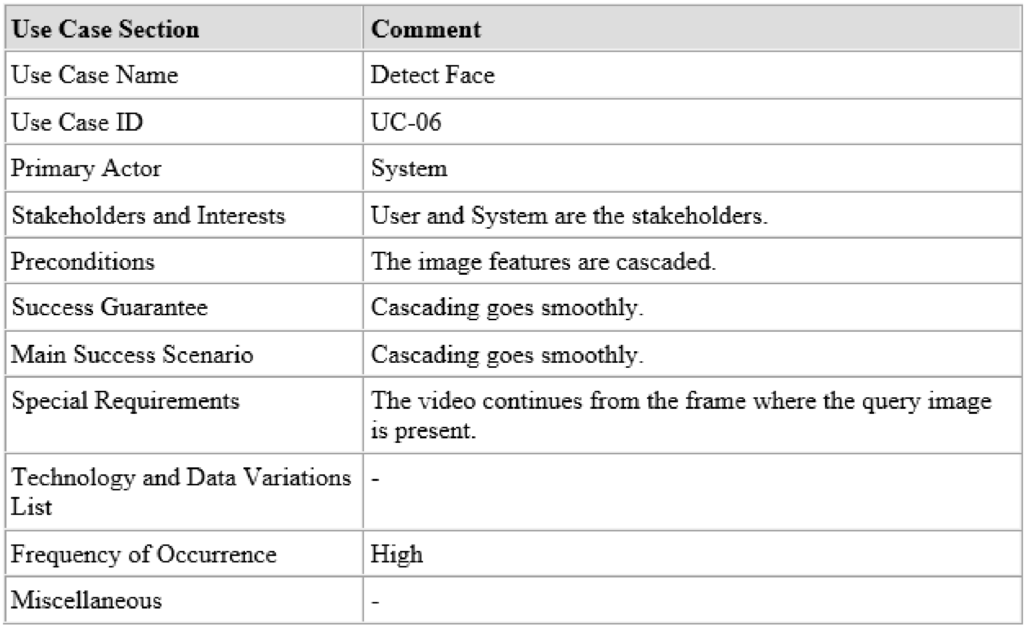


Use Case Section	Comment
Use Case Name	Find Video
Use Case ID	UC-05
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Detection and recognition were successful.
Success Guarantee	The video should be relevant to the query image.
Main Success Scenario	-
Special Requirements	The video continues from the frame where the query image is present.
Technology and Data Variations List	-
Frequency of Occurrence	High
Miscellaneous	This use case includes 2 use cases.

Figure 3.7: UC-05

In Fig 3.8, the use case for Detect Face can be seen.

Detect Face

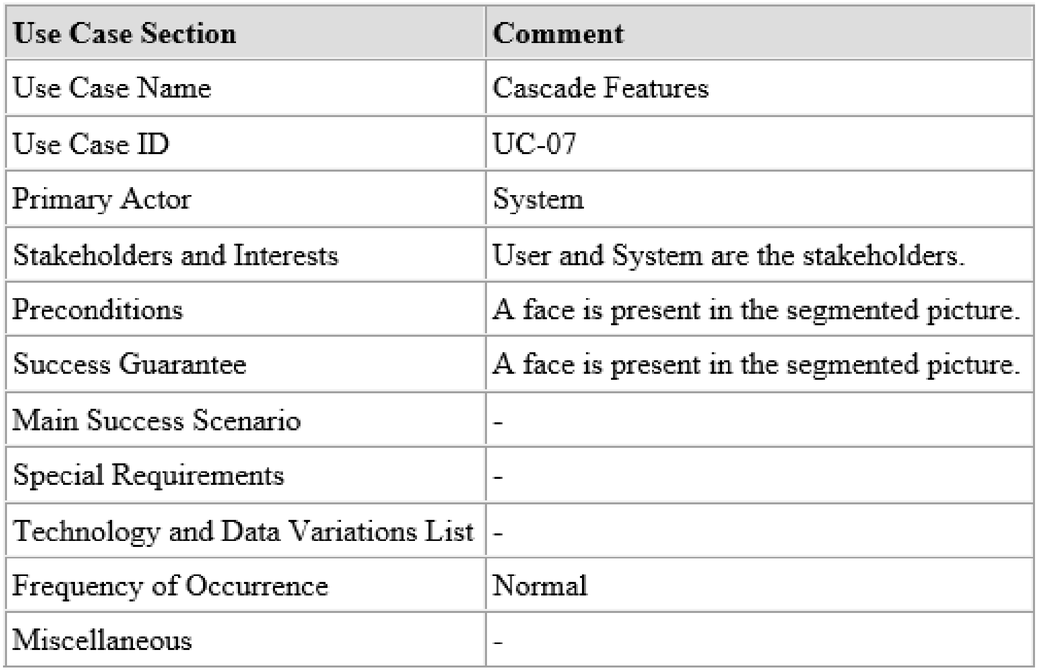


Use Case Section	Comment
Use Case Name	Detect Face
Use Case ID	UC-06
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	The image features are cascaded.
Success Guarantee	Cascading goes smoothly.
Main Success Scenario	Cascading goes smoothly.
Special Requirements	The video continues from the frame where the query image is present.
Technology and Data Variations List	-
Frequency of Occurrence	High
Miscellaneous	-

Figure 3.8: UC-06

In Fig 3.9, the use case for Cascade Features can be seen.

Cascade Features

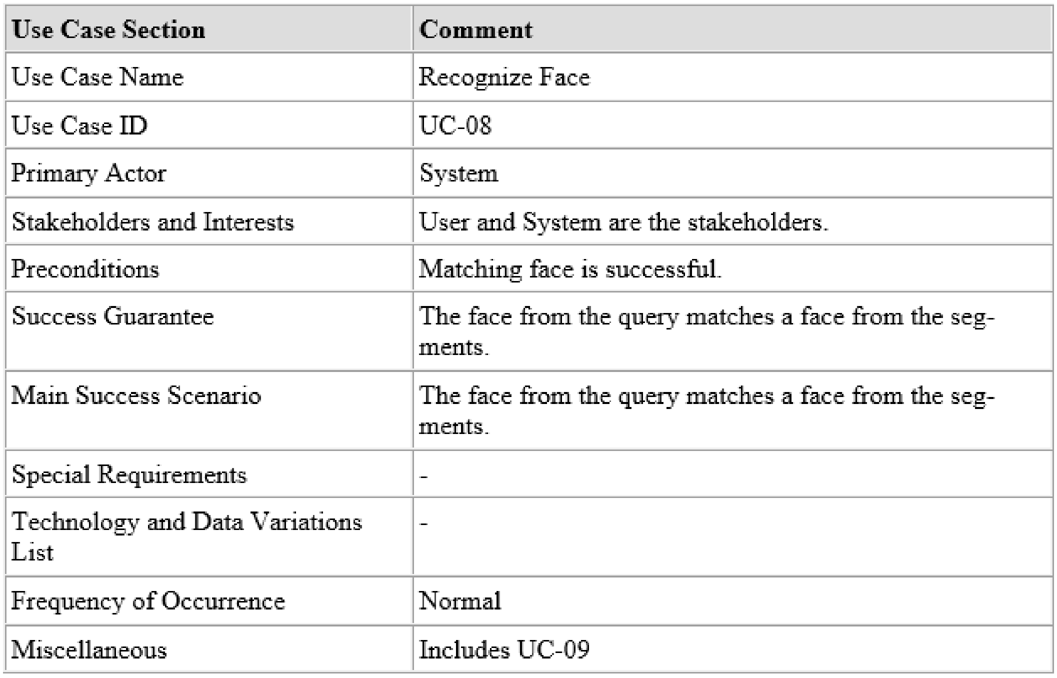


Use Case Section	Comment
Use Case Name	Cascade Features
Use Case ID	UC-07
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	A face is present in the segmented picture.
Success Guarantee	A face is present in the segmented picture.
Main Success Scenario	-
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	-

Figure 3.9: UC-07

In Fig 3.10, the use case for Recognize Face can be seen.

Recognize Face

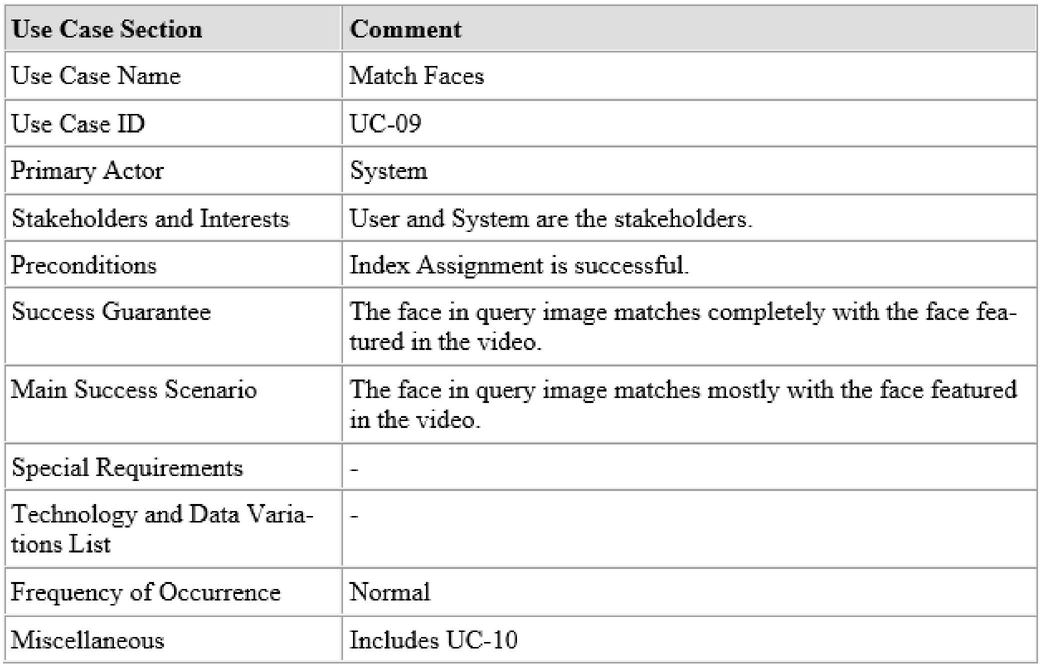


Use Case Section	**Comment**
Use Case Name	Recognize Face
Use Case ID	UC-08
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Matching face is successful.
Success Guarantee	The face from the query matches a face from the segments.
Main Success Scenario	The face from the query matches a face from the segments.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Includes UC-09

Figure 3.10: UC-08

In Fig 3.11, the use case for Match Faces can be seen.

Match Faces

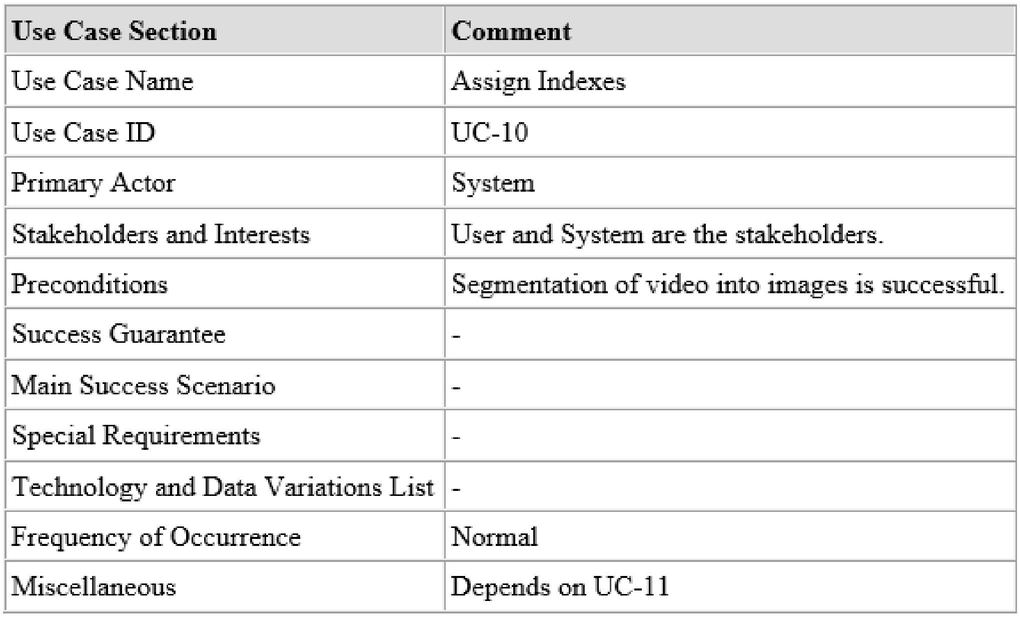


Use Case Section	Comment
Use Case Name	Match Faces
Use Case ID	UC-09
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Index Assignment is successful.
Success Guarantee	The face in query image matches completely with the face featured in the video.
Main Success Scenario	The face in query image matches mostly with the face featured in the video.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Includes UC-10

Figure 3.11: UC-09

In Fig 3.12, the use case for Assign Indexes can be seen.

Assign Indexes

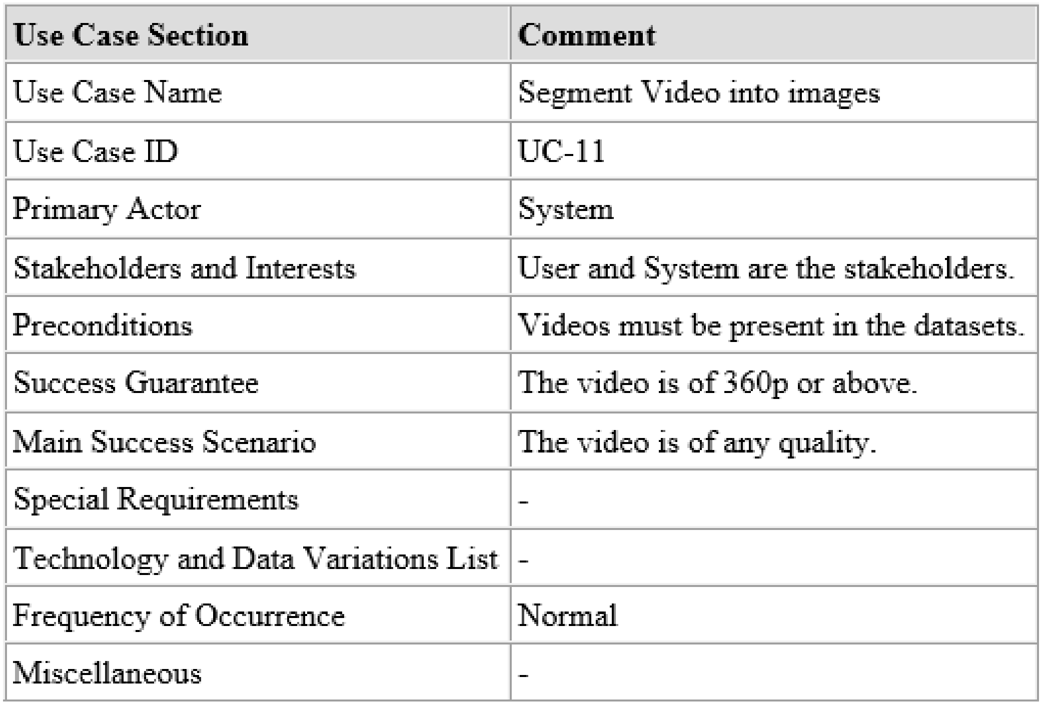


Use Case Section	Comment
Use Case Name	Assign Indexes
Use Case ID	UC-10
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Segmentation of video into images is successful.
Success Guarantee	-
Main Success Scenario	-
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Depends on UC-11

Figure 3.12: UC-10

In Fig 3.13, the use case for Segment Video Into Images can be seen.

Segment Video into Images

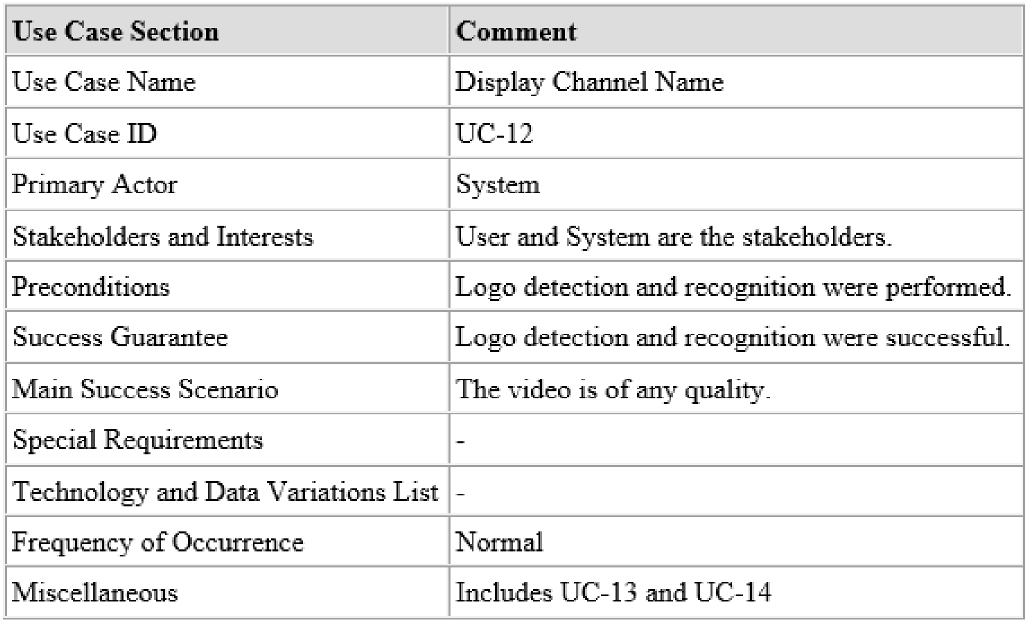


Use Case Section	Comment
Use Case Name	Segment Video into images
Use Case ID	UC-11
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Videos must be present in the datasets.
Success Guarantee	The video is of 360p or above.
Main Success Scenario	The video is of any quality.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	-

Figure 3.13: UC-11

In Fig 3.14, the use case for Display Channel Name can be seen.

Display Channel Name

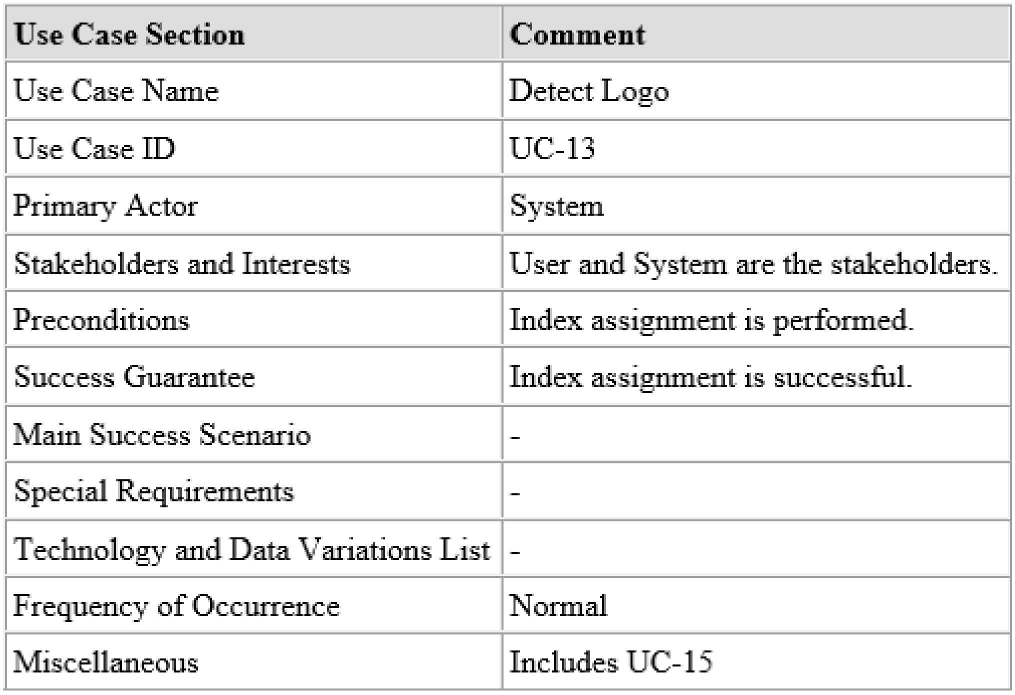


Use Case Section	Comment
Use Case Name	Display Channel Name
Use Case ID	UC-12
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Logo detection and recognition were performed.
Success Guarantee	Logo detection and recognition were successful.
Main Success Scenario	The video is of any quality.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Includes UC-13 and UC-14

Figure 3.14: UC-12

In Fig 3.15, the use case for Detect Logo can be seen.

Detect Logo

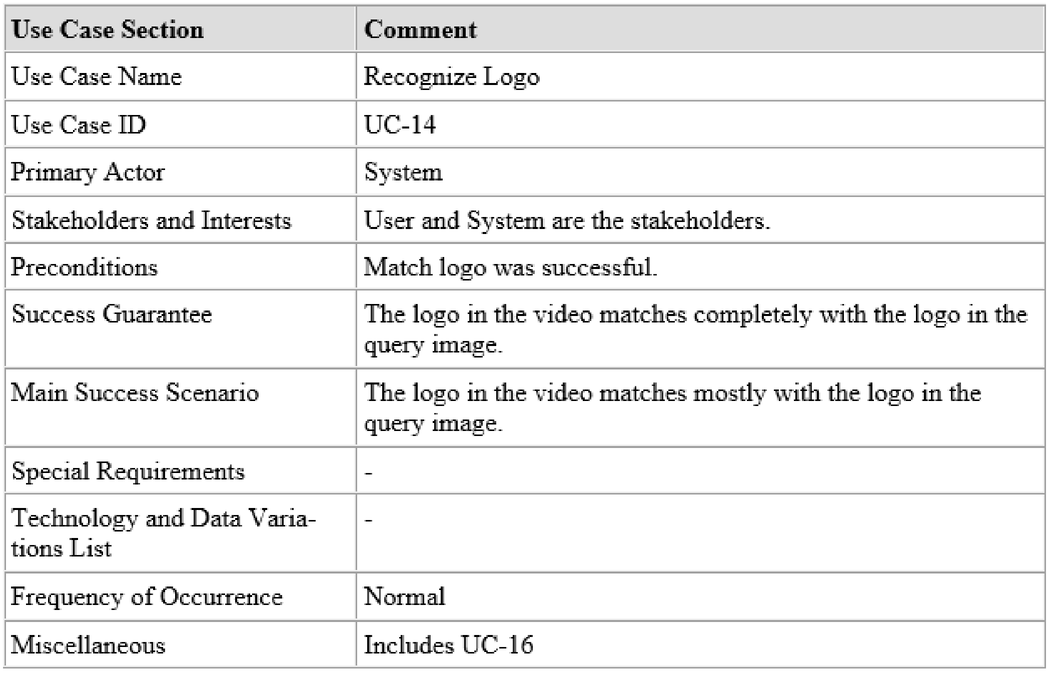


Use Case Section	Comment
Use Case Name	Detect Logo
Use Case ID	UC-13
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Index assignment is performed.
Success Guarantee	Index assignment is successful.
Main Success Scenario	-
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Includes UC-15

Figure 3.15: UC-13

In Fig 3.16, the use case for Recognize Logo can be seen.

Recognize Logo

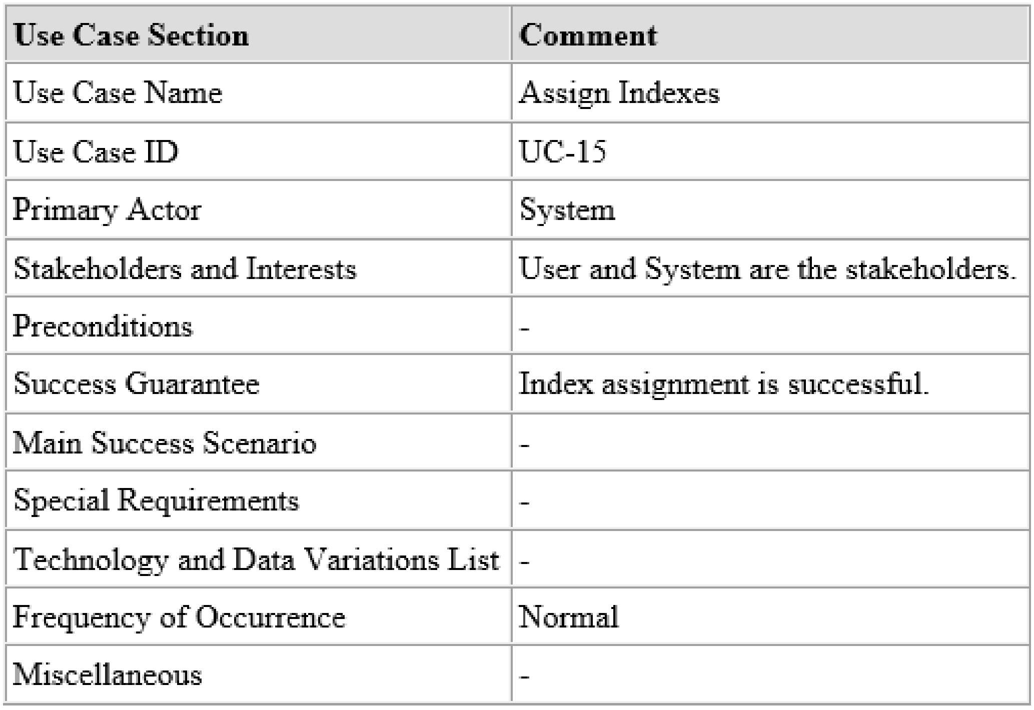


Use Case Section	Comment
Use Case Name	Recognize Logo
Use Case ID	UC-14
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Match logo was successful.
Success Guarantee	The logo in the video matches completely with the logo in the query image.
Main Success Scenario	The logo in the video matches mostly with the logo in the query image.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Includes UC-16

Figure 3.16: UC-14

In Fig 3.17, the use case for Assign Indexes to Logos can be seen.

Assign Indexes to logos

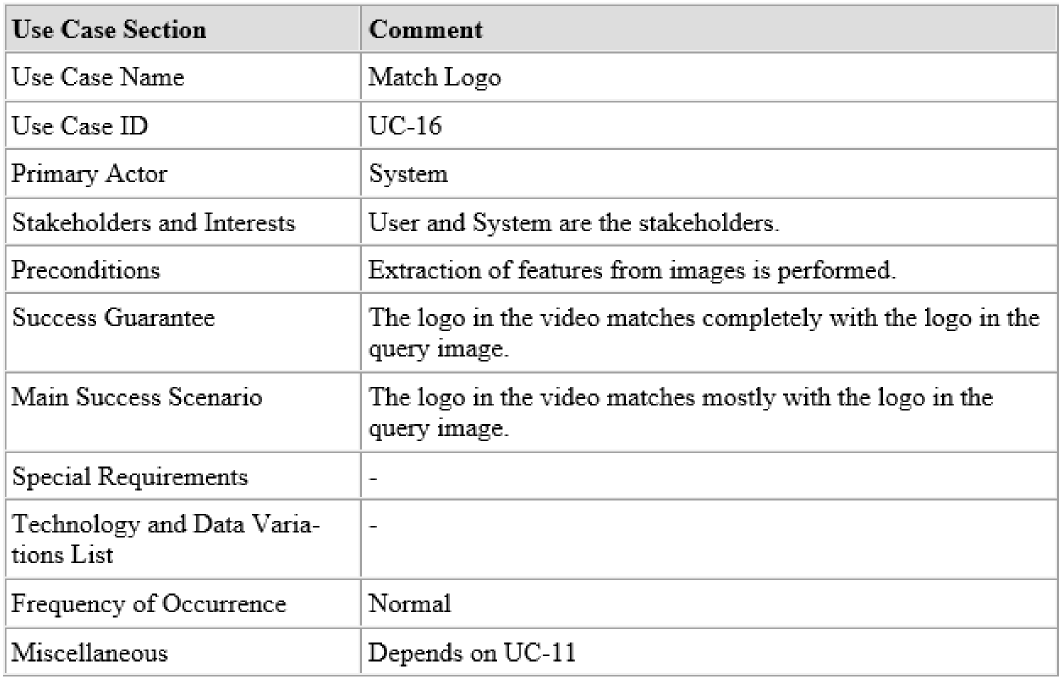


Use Case Section	Comment
Use Case Name	Assign Indexes
Use Case ID	UC-15
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	-
Success Guarantee	Index assignment is successful.
Main Success Scenario	-
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	-

Figure 3.17: UC-15

In Fig 3.18, the use case for Match Logo can be seen.

Match Logo

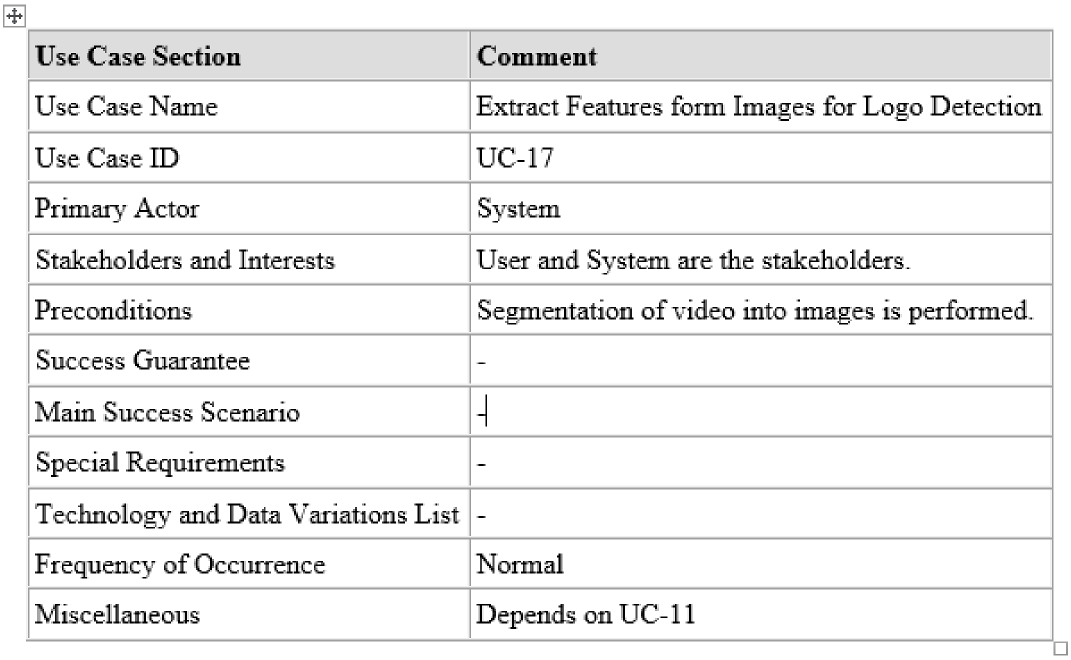


Use Case Section	Comment
Use Case Name	Match Logo
Use Case ID	UC-16
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Extraction of features from images is performed.
Success Guarantee	The logo in the video matches completely with the logo in the query image.
Main Success Scenario	The logo in the video matches mostly with the logo in the query image.
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Depends on UC-11

Figure 3.18: UC-16

In Fig 3.19, the use case for Extract Features from Image for Logo Detection can be seen.

Extract Features from Image for Logo Detection



Use Case Section	Comment
Use Case Name	Extract Features form Images for Logo Detection
Use Case ID	UC-17
Primary Actor	System
Stakeholders and Interests	User and System are the stakeholders.
Preconditions	Segmentation of video into images is performed.
Success Guarantee	-
Main Success Scenario	-
Special Requirements	-
Technology and Data Variations List	-
Frequency of Occurrence	Normal
Miscellaneous	Depends on UC-11

Figure 3.19: UC-17

4 Design

4.1 Objective

The objective of this chapter is to describe the design of FQVRS using Visual Models. The design document includes Class Diagram, Use Case diagram, Fully Dressed Use cases [7] , Package Model, Deployment Diagram, Activity Diagram, and the Domain Model.

4.2 System Architecture

The Fig 4.1 describes the architecture of our system.

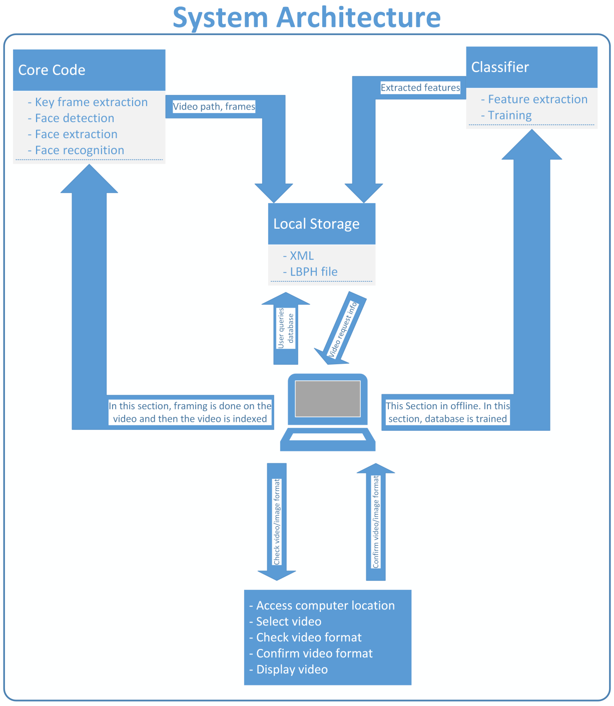


Figure 4.1: System Architecture

4.3 Class Diagram

The class diagram of FQVRS can be seen in Fig 4.2.

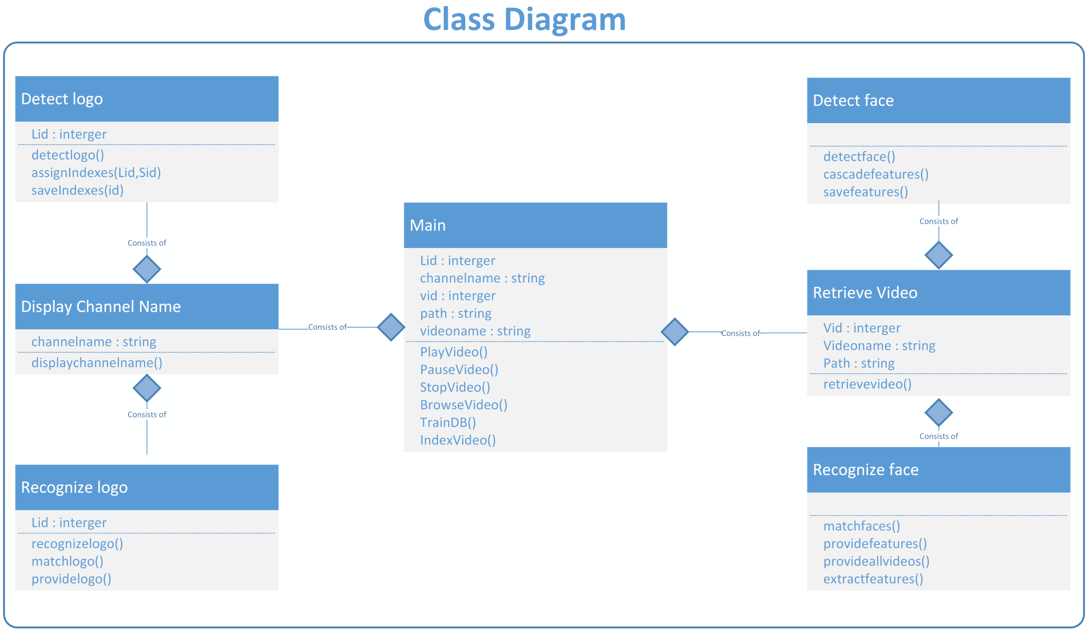


Figure 4.2: Class Diagram

4.4 Package Diagram

The package diagram of FQVRS can be seen in Fig 4.3.

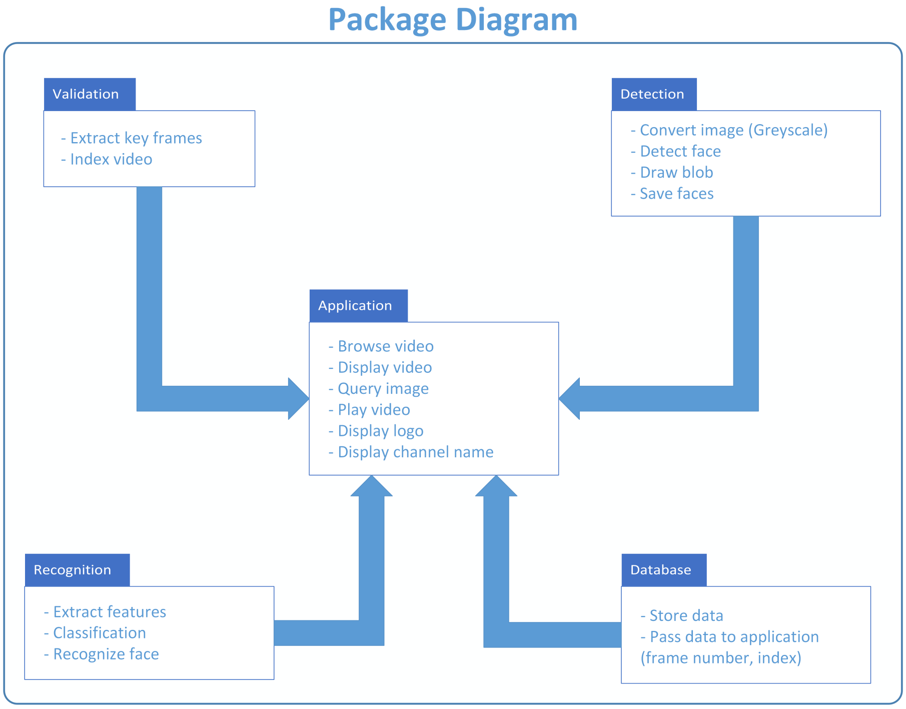


Figure 4.3: Package Diagram

4.5 Domain Model

The domain model of FQVRS can be seen in Fig 4.4.

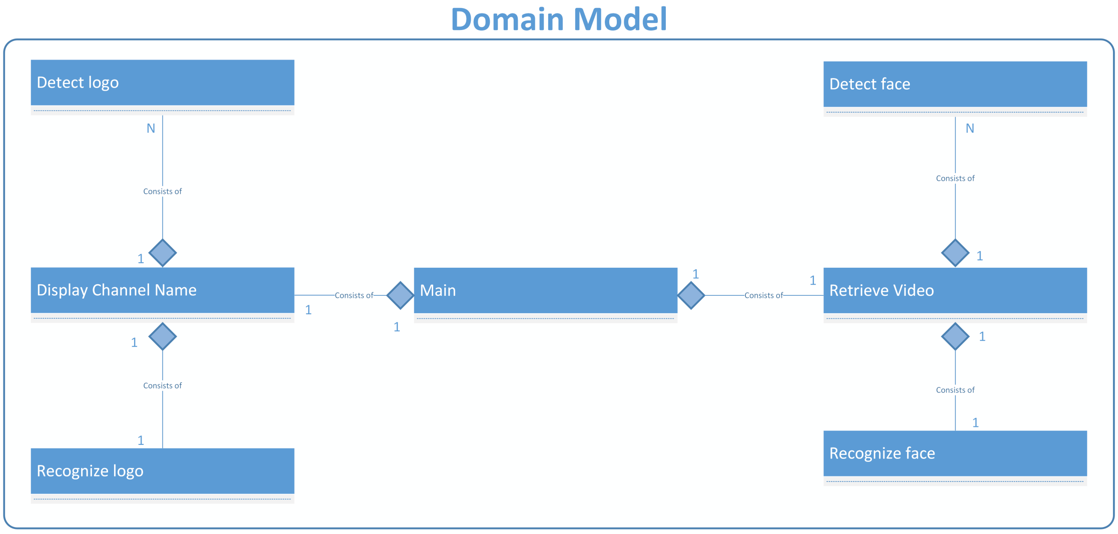


Figure 4.4: Domain Model

4.6 Deployment Diagram

The deployment diagram of FQVRS can be seen in Fig 4.5 .

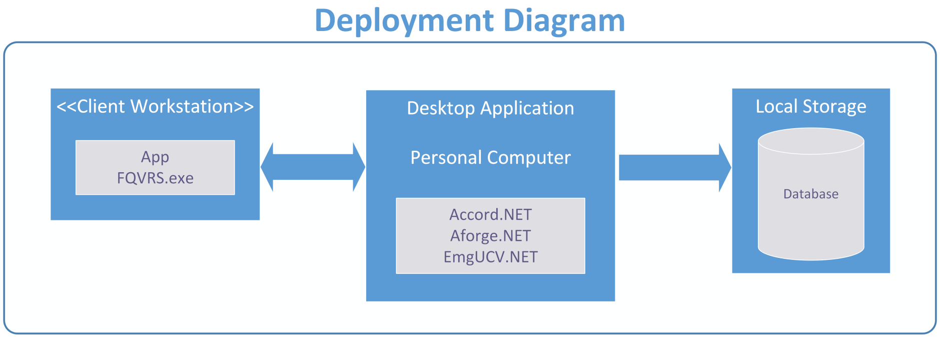


Figure 4.5: Deployment Diagram

4.7 Activity Diagram

The activity diagram of FQVRS can be seen in Fig 4.6.

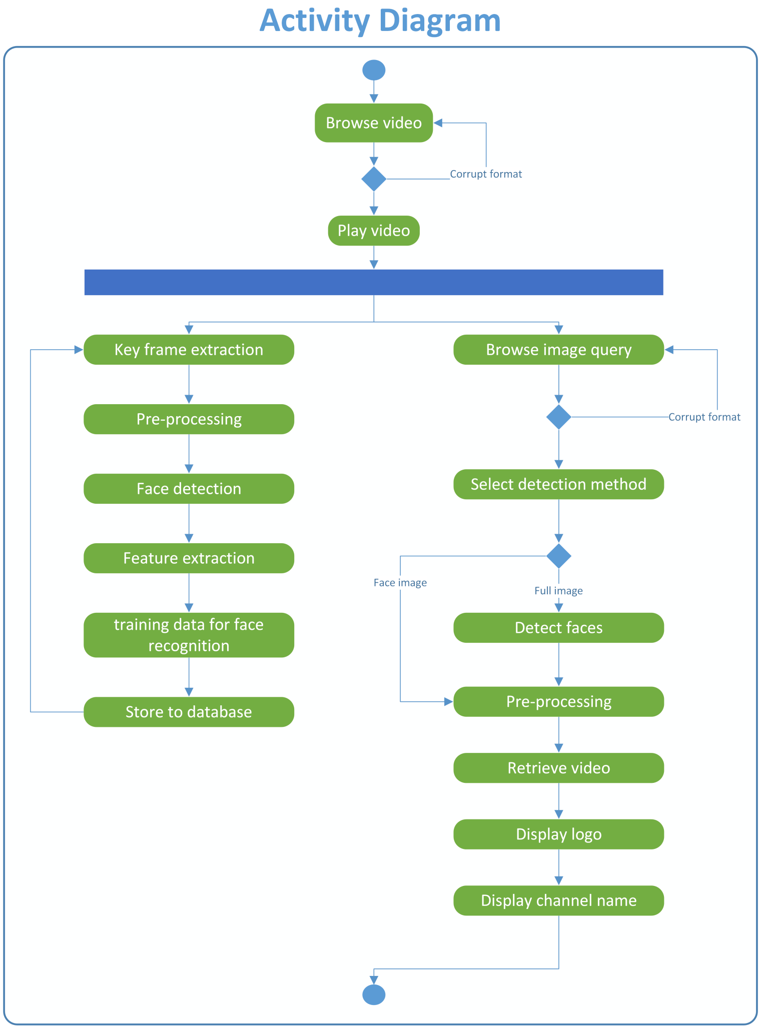


Figure 4.6: Activity Diagram

4.8 System Sequence Diagrams

System sequence diagram for Use case 1 can be seen in Fig 4.7.

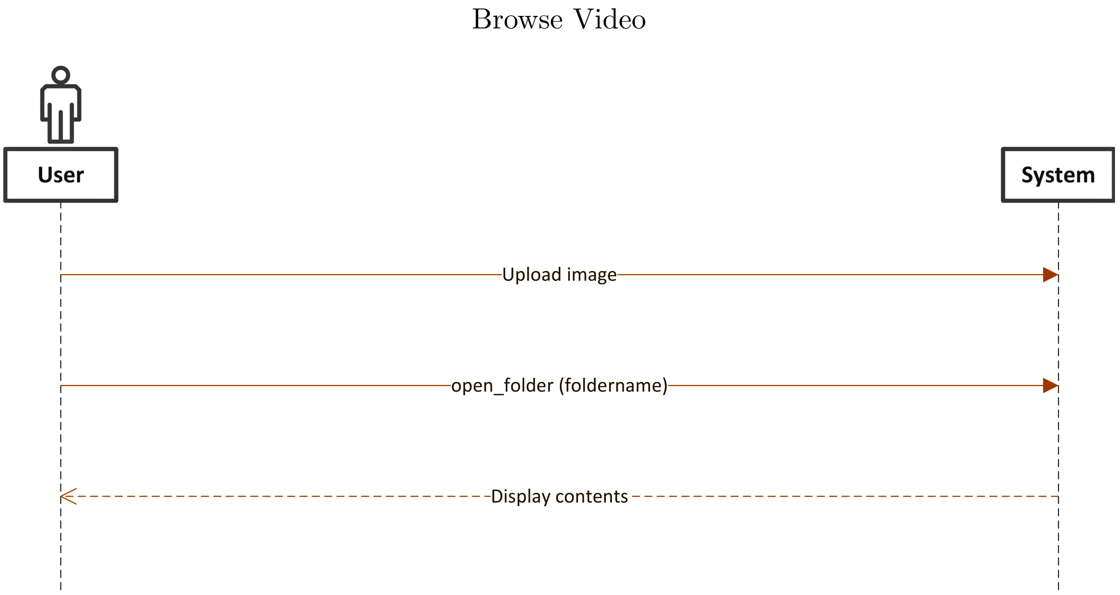


Figure 4.7: System Sequence Diagram of Use case 1

System sequence diagram for Use case 2 can be seen in Fig 4.8.

Upload Video

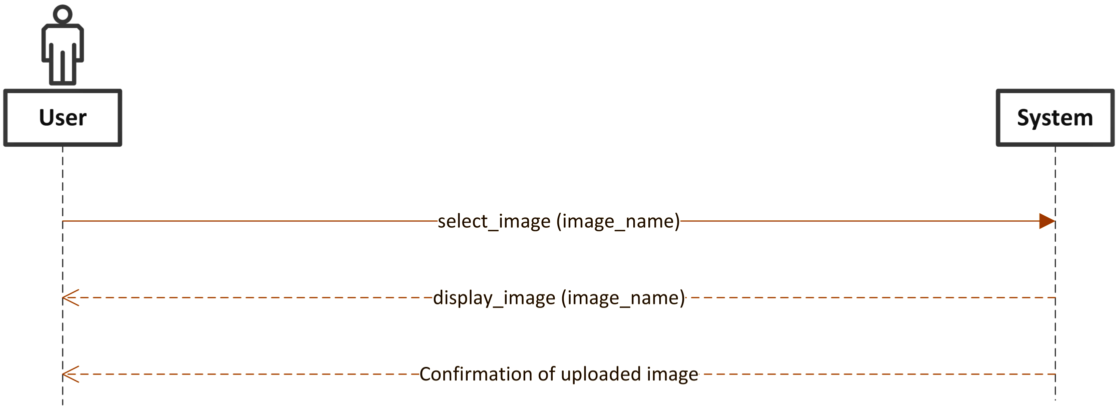

User
System
select_image (image_name)
display_image (image_name)
Confirmation of uploaded image

Figure 4.8: System Sequence Diagram of Use case 2

System sequence diagram for Use case 3 can be seen in Fig 4.9.

Extract Features From Images

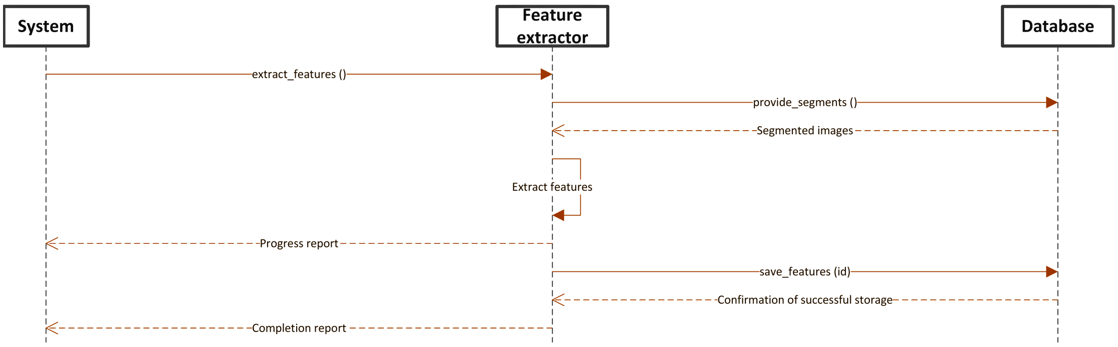


Figure 4.9: System Sequence Diagram of Use case 3

System sequence diagram for Use case 4 can be seen in Fig 4.10.

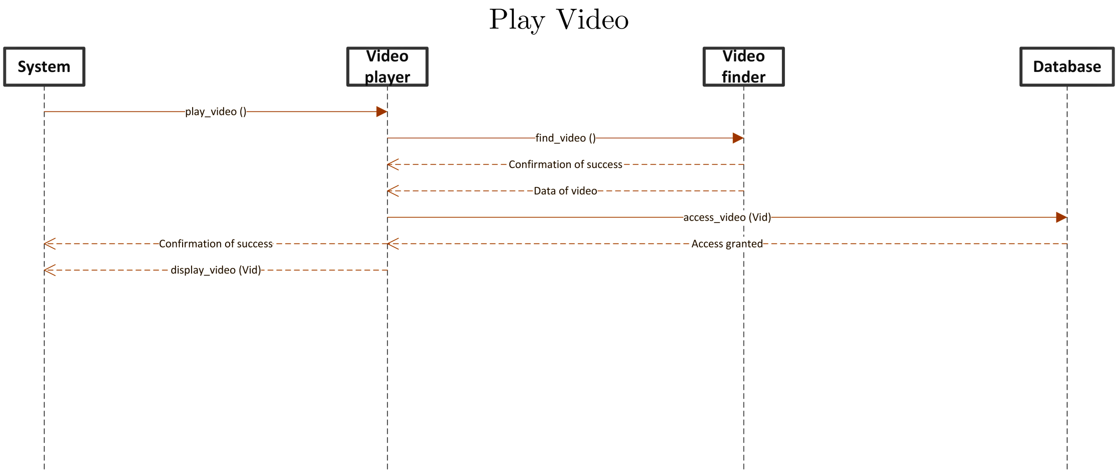


Figure 4.10: System Sequence Diagram of Use case 4

System sequence diagram for Use case 5 can be seen in Fig 4.11.

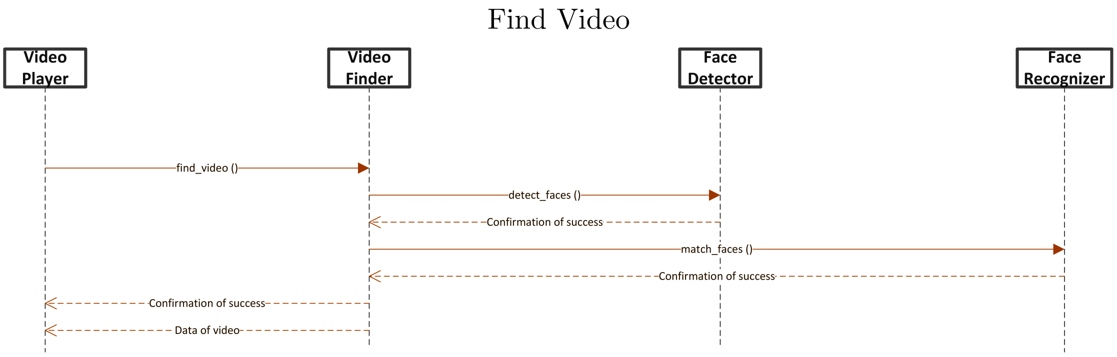


Figure 4.11: System Sequence Diagram of Use case 5

System sequence diagram for Use case 6 can be seen in Fig 4.12.

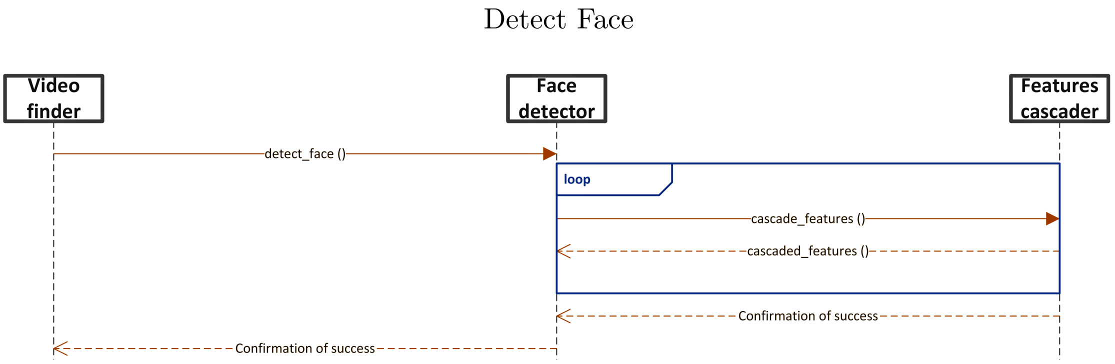


Figure 4.12: System Sequence Diagram of Use case 6

System sequence diagram for Use case 7 can be seen in Fig 4.13.

Cascade Features

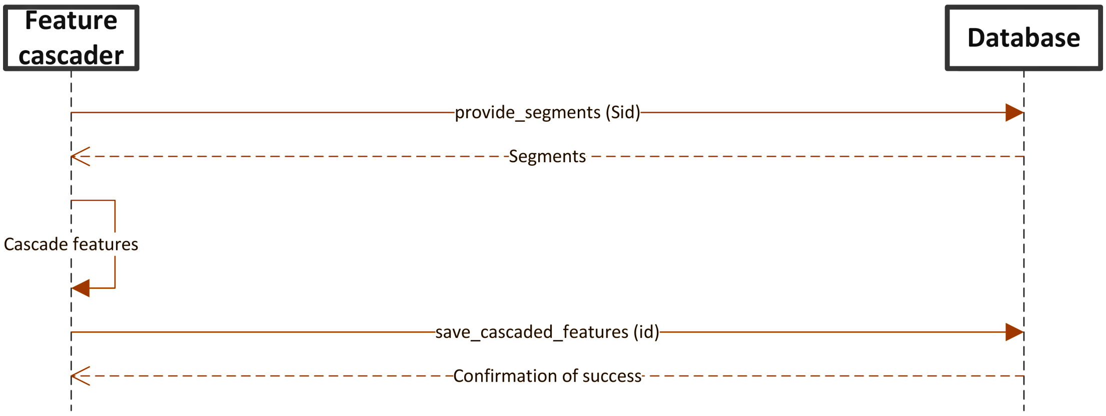


Figure 4.13: System Sequence Diagram of Use case 7

System sequence diagram for Use case 8 can be seen in Fig 4.14.

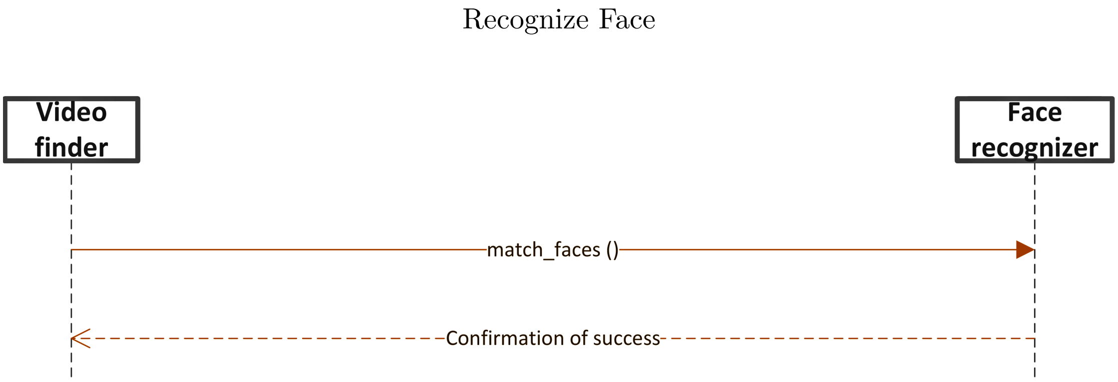


Figure 4.14: System Sequence Diagram of Use case 8

System sequence diagram for Use case 9 can be seen in Fig 4.15.

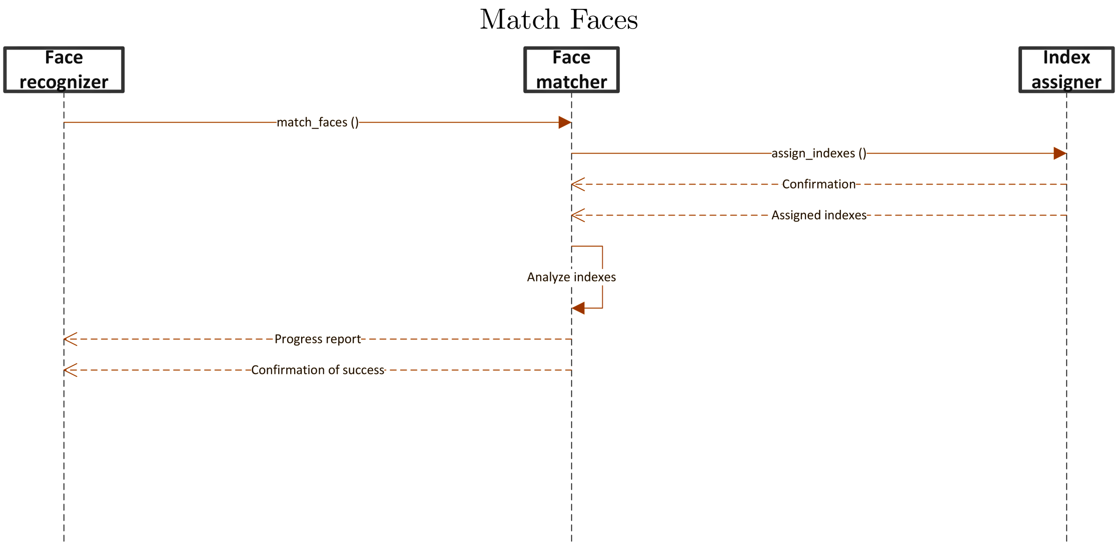


Figure 4.15: System Sequence Diagram of Use case 9

System sequence diagram for Use case 10 can be seen in Fig 4.16.

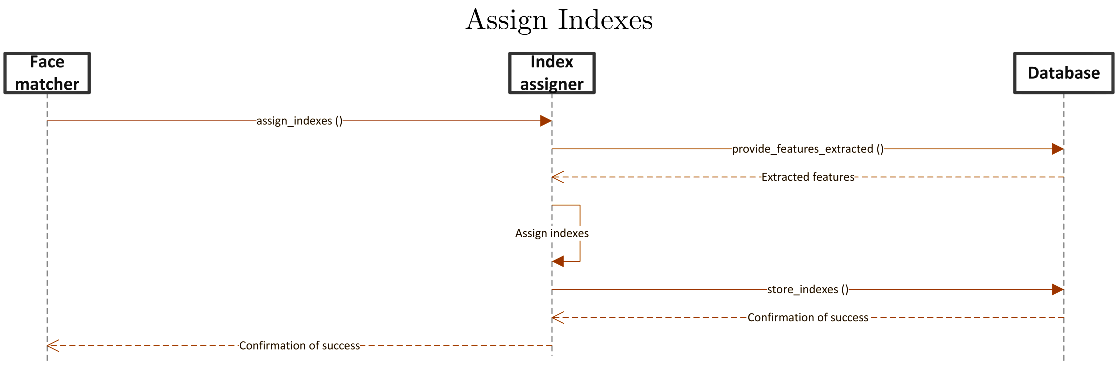


Figure 4.16: System Sequence Diagram of Use case 10

System sequence diagram for Use case 11 can be seen in Fig 4.17.

Segment Videos Into Images

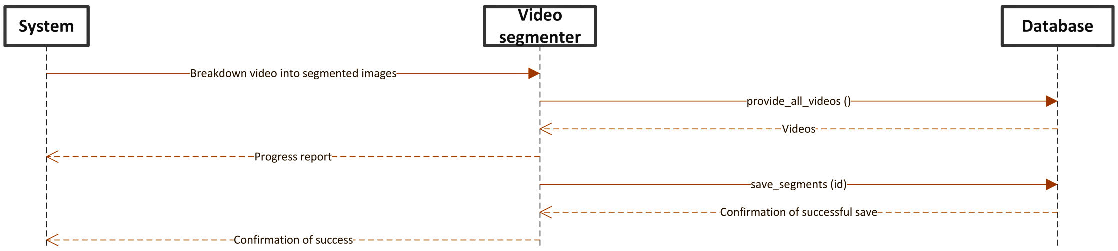


Figure 4.17: System Sequence Diagram of Use case 11

System sequence diagram for Use case 12 can be seen in Fig 4.18.

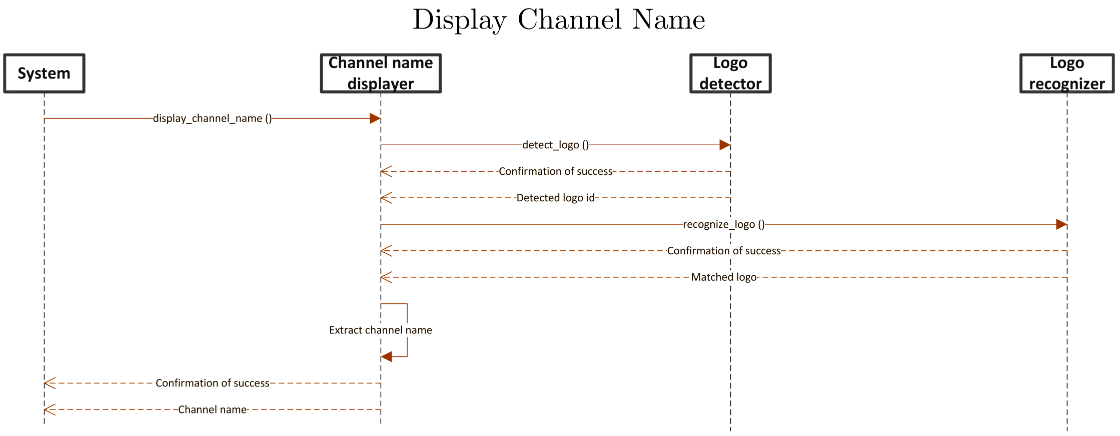


Figure 4.18: System Sequence Diagram of Use case 12

System sequence diagram for Use case 13 can be seen in Fig 4.19.

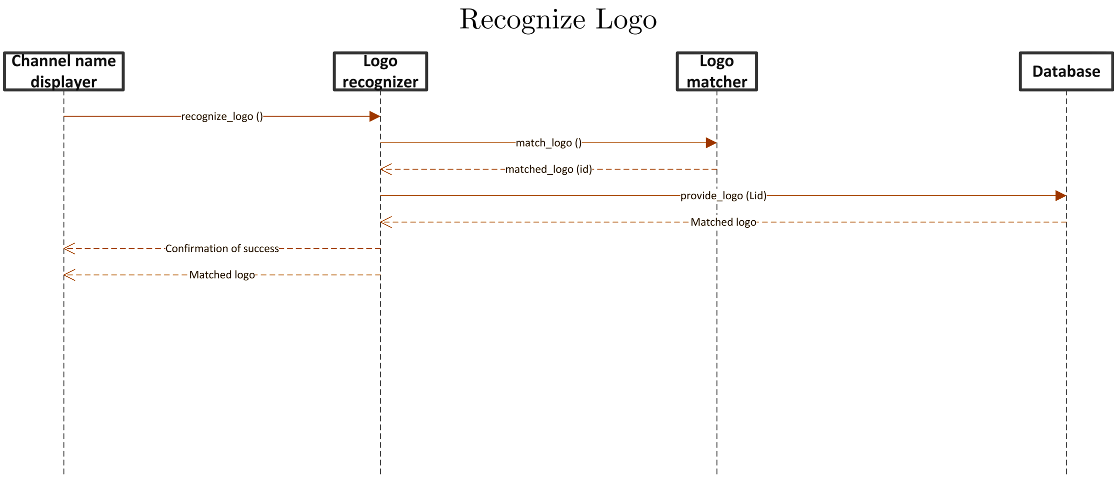


Figure 4.19: System Sequence Diagram of Use case 13

System sequence diagram for Use case 14 can be seen in Fig 4.20.

Detect Logo

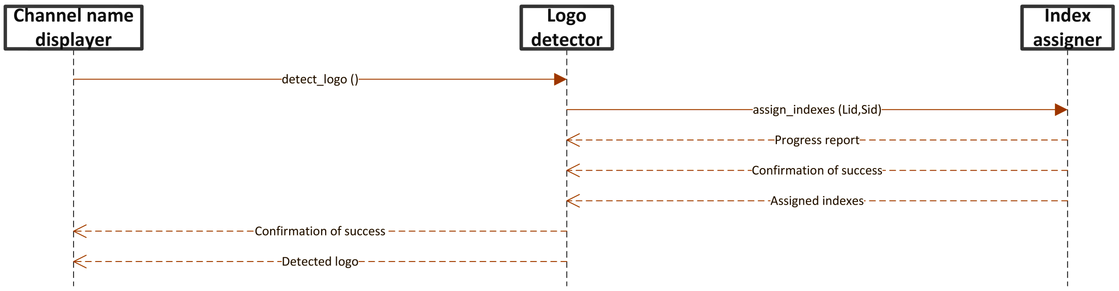


Figure 4.20: System Sequence Diagram of Use case 14

System sequence diagram for Use case 15 can be seen in Fig 4.21.

Assign Indexes to Logos

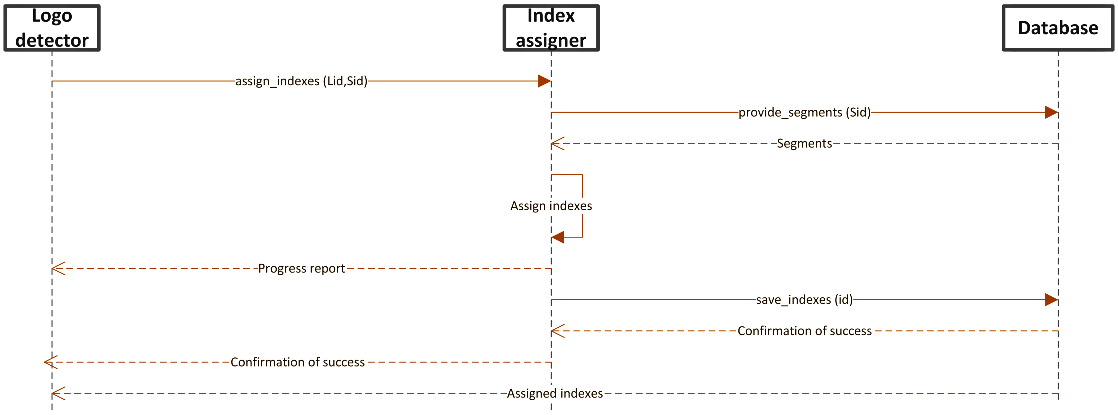


Figure 4.21: System Sequence Diagram of Use case 15

System sequence diagram for Use case 16 can be seen in Fig 4.22.

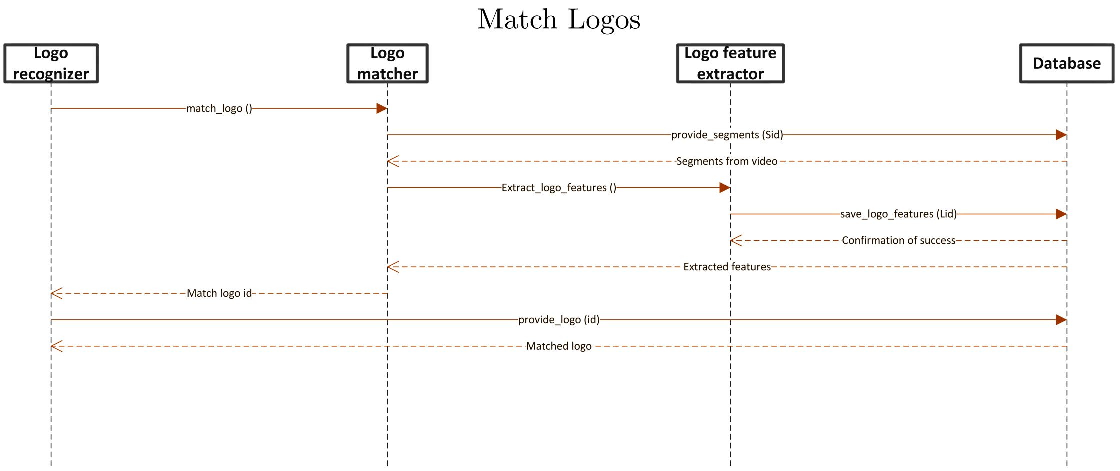


Figure 4.22: System Sequence Diagram of Use case 16

System sequence diagram for Use case 17 can be seen in Fig 4.23.

Extract Features From logo Images

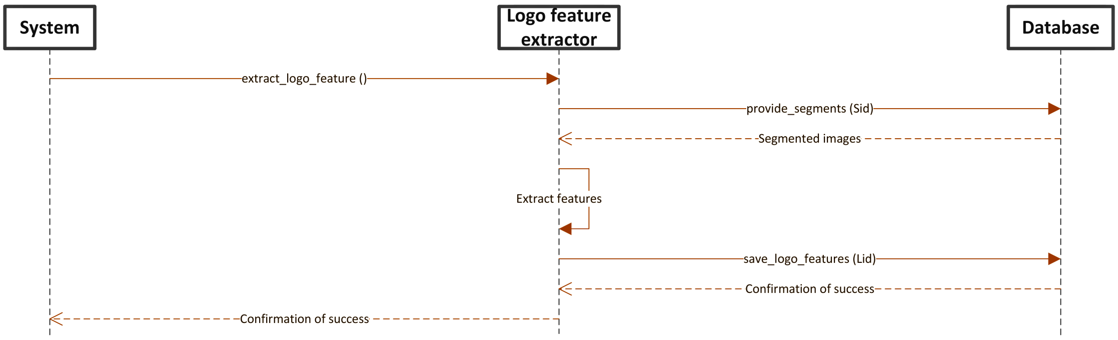


Figure 4.23: System Sequence Diagram of Use case 17

5 System Implementation

5.1 Objective

The objective of this chapter is to provide an overview of the tools and technologies and the libraries used to develop this system. This chapter also provides the system architecture of the FQVRS.

5.2 Tools and Technologies Used

The following are the tools and technologies used in the development of FQVRS.

5.2.1 Environment

Microsoft Visual Studio 2013 (an IDE from Microsoft Corporations) was used as a coding environment in the development phase of the system. The software provides a very interactive coding environment as is widely used in the development of such systems.

5.2.2 Language

We have used C-sharp programming language for the development of our system. The basic idea was to take advantage of the Windows Forms that the C-sharp provides us to be more interactive with the user, for browsing a file and opening image queries, windows forms provide a very practical user interface.

5.2.3 Libraries

The aforge.net framework is used for the image processing in the system. Aforge.Net, operating under the name Accord.Net, provides APIs for Computer Vision, Image Processing, and Video Processing. The libraries from aforge.net were used for the retrieval of frames from videos, facial detection, and facial recognition.

5.2.4 Metro Framework

Metro Framework is a Windows Modern UI for.NET WinForms Applications. It provides a Windows 8 like the theme for a very elegant user interface. The Framework was used by adding additional libraries to the system.

5.2.5 Methodology

We are going to use Haar scaling for face detection and LBP for face recognition. Haar scaling for the detection of faces include cascading of simple features and facial recognition using LBP includes the extraction of features from both the query image and the segments extracted from the news videos present in the datasets. The extraction of features from the images (of faces) can be used to get the measure for the similarity between the images. The logo recognition phase will be done by using speeded up robust features (SURF) [2].

5.3 System Architecture

The Fig 5.1 describes the architecture of our system.

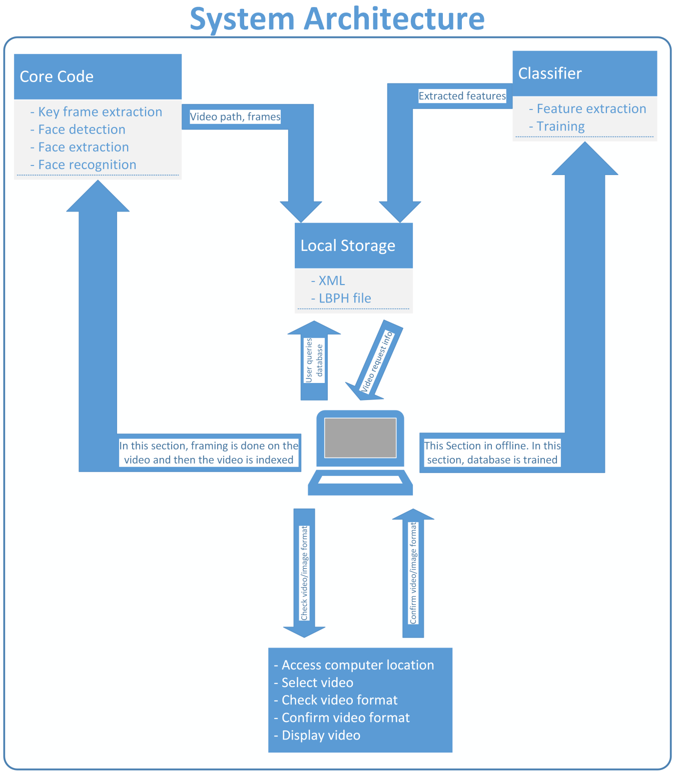


Figure 5.1: System Architecture

6 System Evaluation and Testing

6.1 Objective

The objective of this project is to test and evaluate the functionalities of the system. We use different techniques to find the restrictions of the software.

6.2 Usability Testing

Usability testing delivers the specifies the time a software system requires to execute an explicit task of the system. Usability testing is assessed by the objective audience of the software application.These tests express if the system is executing the tasks of its intentions or not. The criteria for usability testing is whether the system is easy to use, easy to learn to use, safe to use, and consistent. All these criterion have been tested in the following subtopics.

6.2.1 Usability

The system is very easy to use as the User Interface is not very complex and the text on the buttons are self explained.

6.2.2 Learn-ability

The UI of the system is very easy to comprehend, even for a new user.

6.2.3 Safety

The system is safe to use.

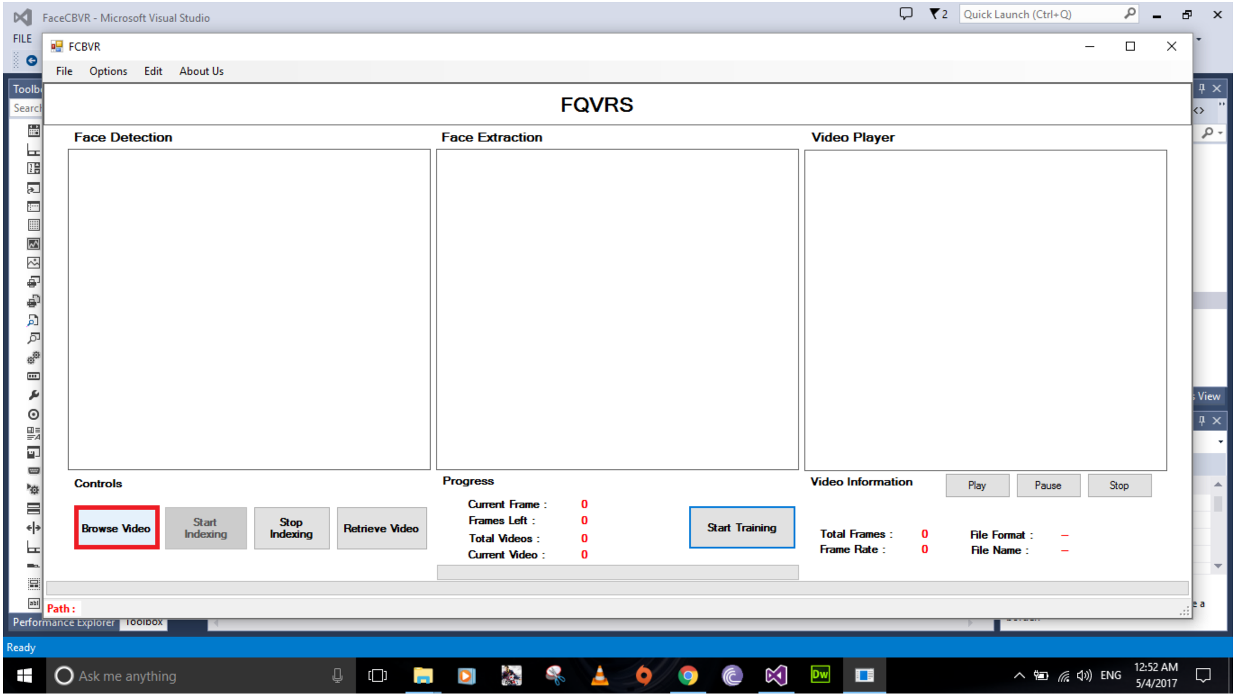
6.2.4 Consistency

The UI of the system is consistent so it will provide no difficulty for user to perform the functionalities provided by the system.

6.3 GUI Testing

The Graphical User Interface of our system is self-explanatory. Our system's GUI contains the buttons,status bar, picture box, and media-player. All the buttons are performing their functionalists efficiently. Status bar provides the information to the system about the current running operation. The GUI has a consistent color scheme so the user can easily understand the system. The Figures, 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6 shows the graphical user interface of FQVRS.

Home Page



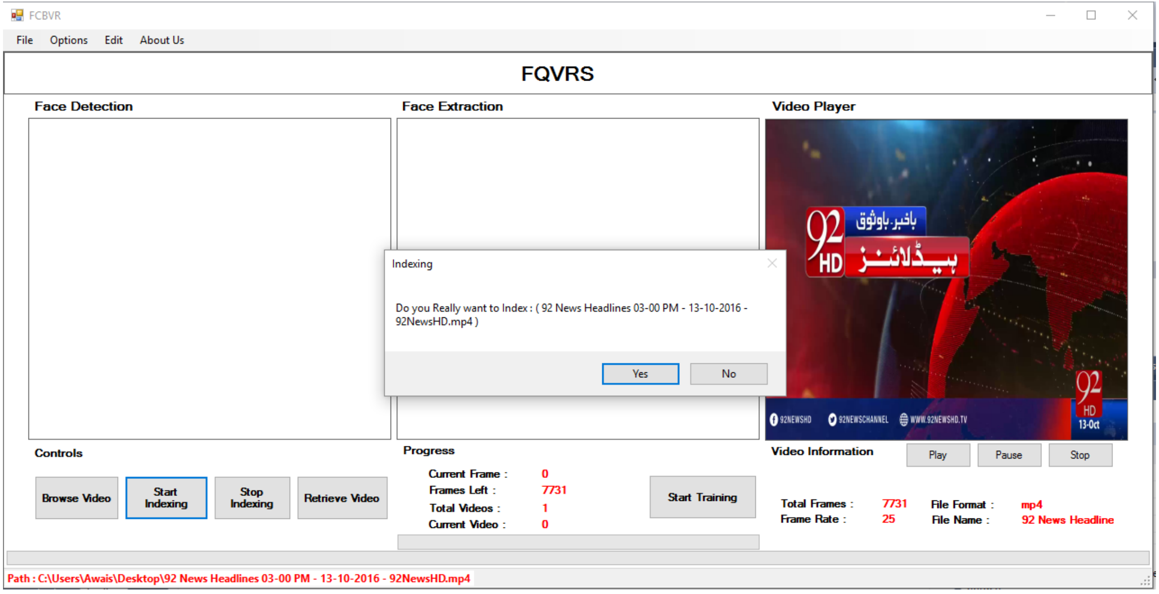
Current Frame : 0
Frames Left : 0
Total Videos : 0
Current Video : 0

Total Frames : 0
Frame Rate : 0
File Format : –
File Name : –

Path :

Figure 6.1: GUI-01

Indexing Video



Indexing

Do you Really want to Index : (92 News Headlines 03-00 PM - 13-10-2016 - 92NewsHD.mp4)

Yes No

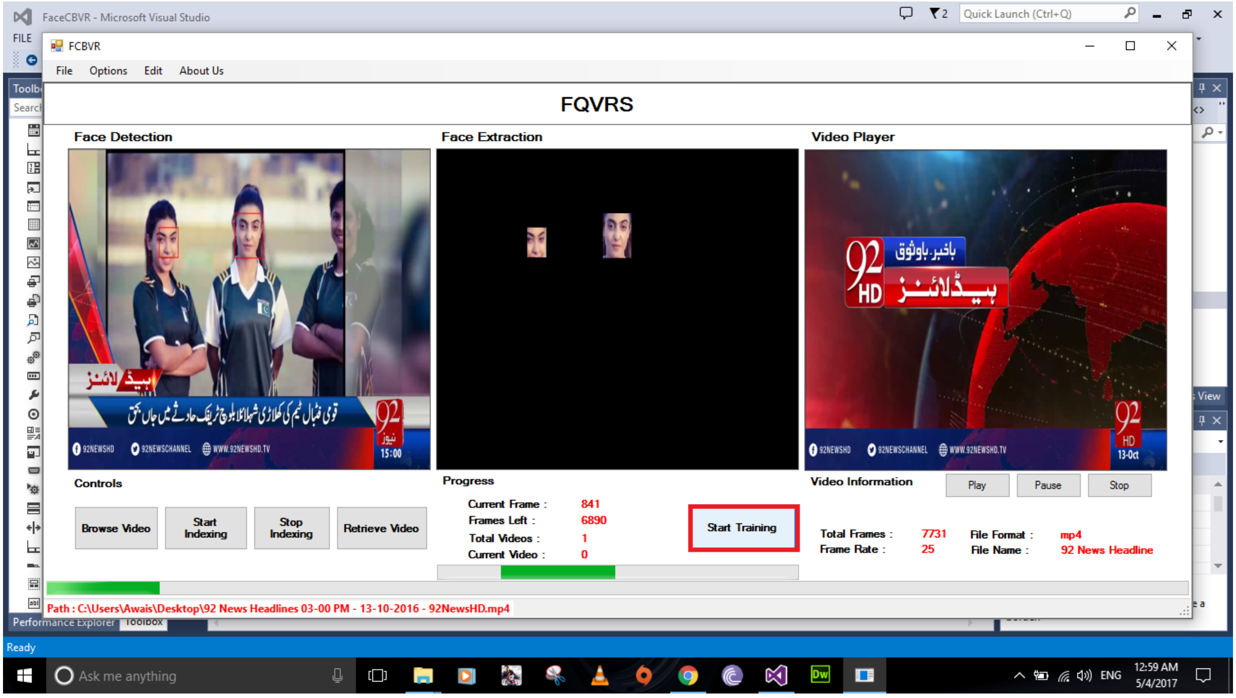
Current Frame : 0
Frames Left : 7731
Total Videos : 1
Current Video : 0

Total Frames : 7731
Frame Rate : 25
File Format : mp4
File Name : 92 News Headline

Path : C:\Users\Awais\Desktop\92 News Headlines 03-00 PM - 13-10-2016 - 92NewsHD.mp4

Figure 6.2: GUI-02

Training Database



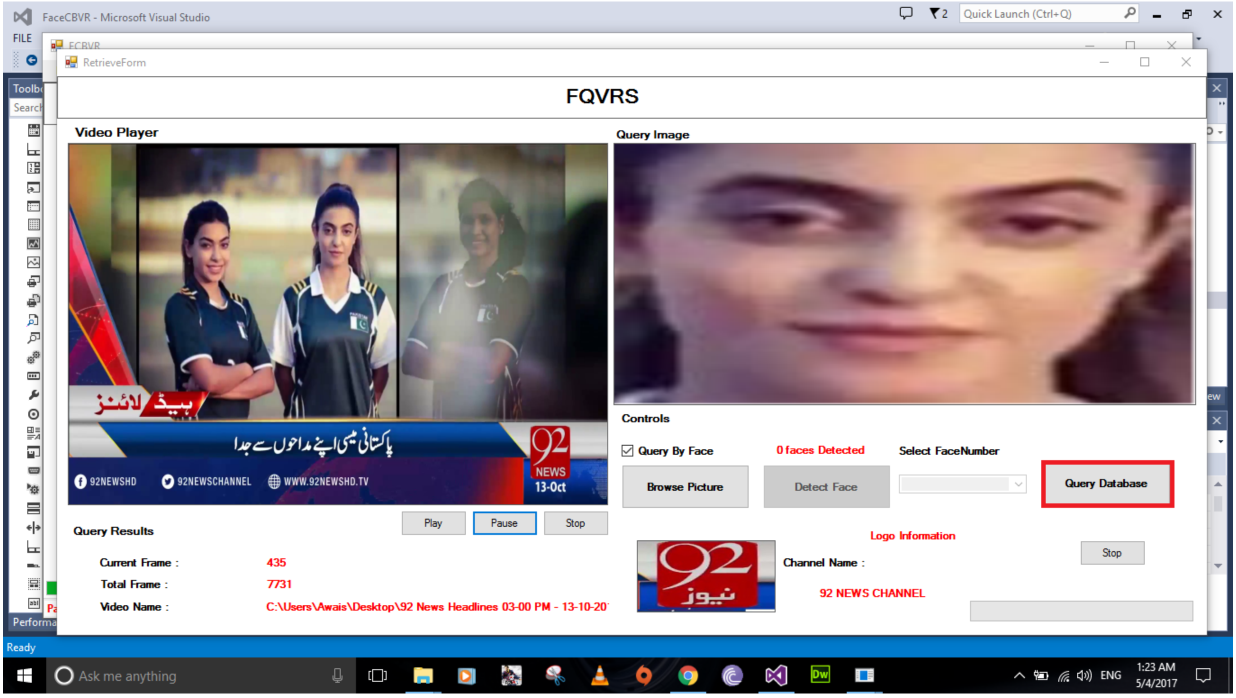
Current Frame : 841
Frames Left : 6890
Total Videos : 1
Current Video : 0

Total Frames : 7731
Frame Rate : 25
File Format : mp4
File Name : 92 News Headline

Path : C:\Users\Awais\Desktop\92 News Headlines 03-00 PM - 13-10-2016 - 92NewsHD.mp4

Figure 6.3: GUI-03

Querying Database



☑ Query By Face

0 faces Detected

Query Results

Current Frame : 435

Total Frame : 7731

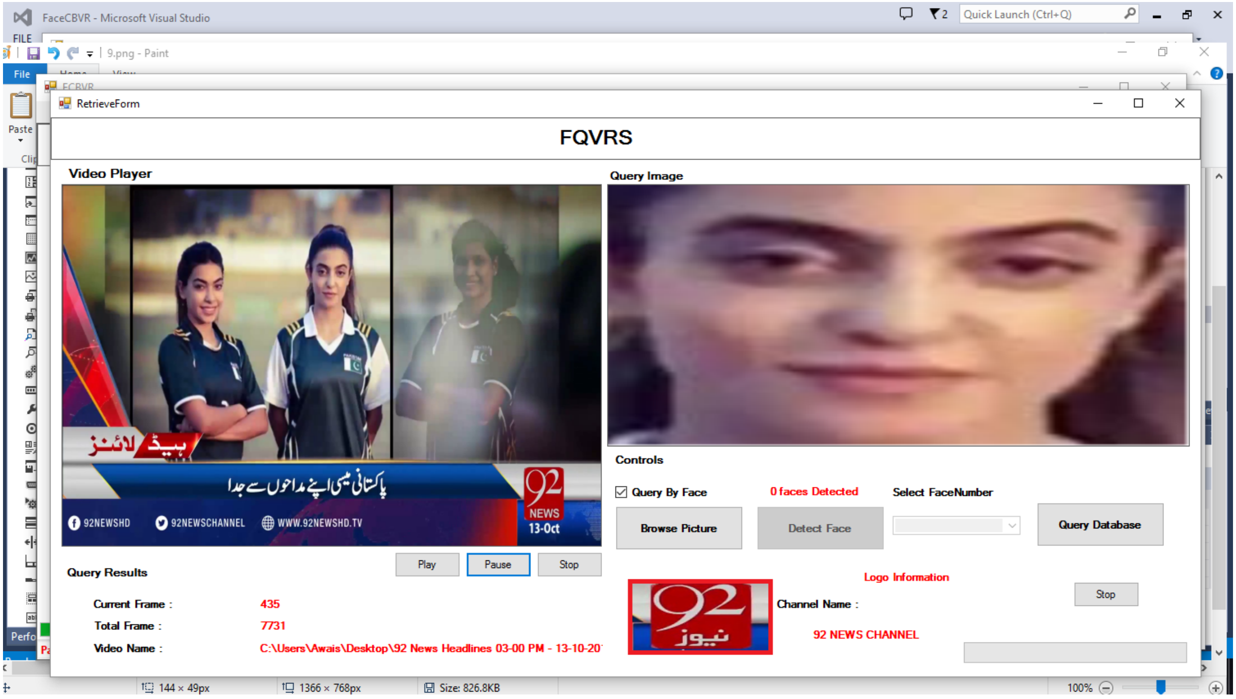
Video Name : C:\Users\Awais\Desktop\92 News Headlines 03-00 PM - 13-10-20

Logo Information

Channel Name : 92 NEWS CHANNEL

Figure 6.4: GUI-04

Detecting Logo



☑ Query By Face

0 faces Detected

Current Frame : 435

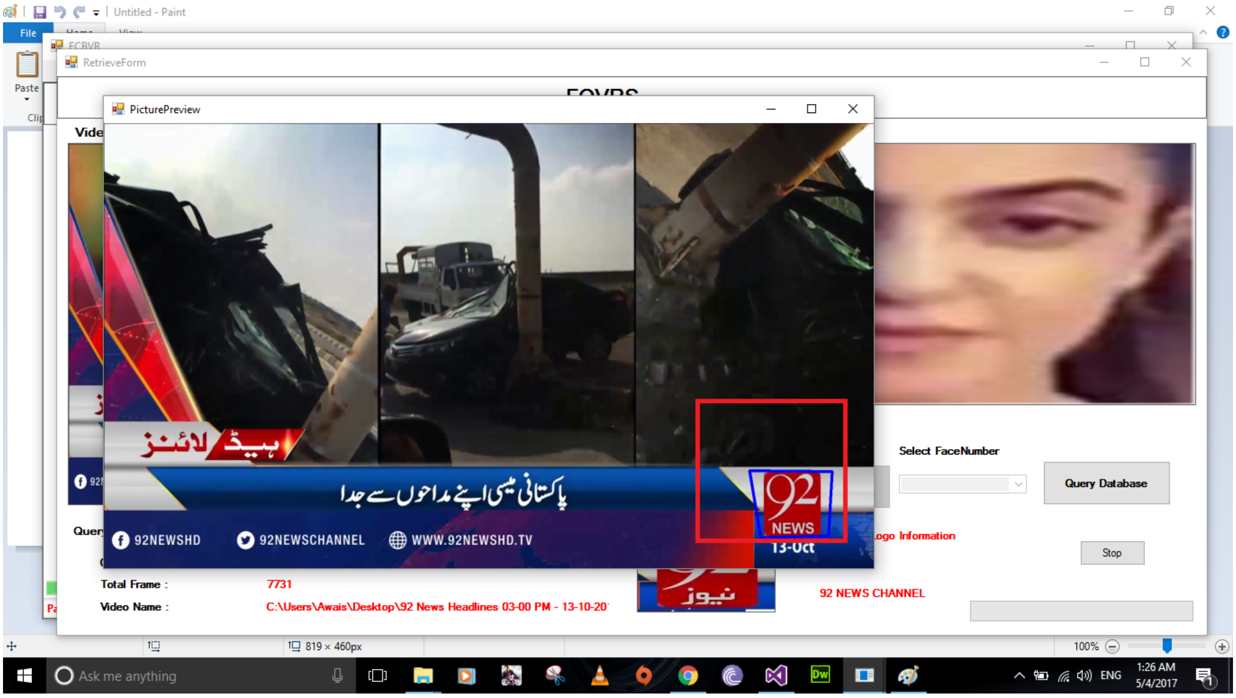
Total Frame : 7731

Video Name : C:\Users\Awais\Desktop\92 News Headlines 03-00 PM - 13-10-20

Channel Name : 92 NEWS CHANNEL

Figure 6.5: GUI-05

Displaying Detected Logo



Total Frame : 7731

Video Name : C:\Users\Awais\Desktop\92 News Headlines 03-00 PM - 13-10-20

Logo Information: 92 NEWS CHANNEL

Figure 6.6: GUI-06

6.4 Performance Testing

The software performance testing is castoff to evaluate the performance of the system. This technique yields the outcomes that how much reliable the system is. System performance testing carried out the following points.

1. The system will take some in face detection as, in face detection, the system will process individual pixel of the frame.

2. The system will take an adequate amount of time (3-4 minutes) for the training of the system.

6.5 Compatibility Testing

Compatibility testing is a form of non-functional testing which is used to evaluate if the system is compatible with the device it is going to be used on or not. The requirements to run FQVRS efficiently are mentioned below.

6.5.1 Device

1. Intel Core i3, 2.8 GHz or equivalent.

2. 4GB Ram

3. 1 GB free storage space on disk.

6.5.2 Operating System

1. Windows 7 or greater version.

2. Microsoft Visual Studio 2010 or greater version.

3. .NET framework 4.5 or greater.

6.6 Installation Testing

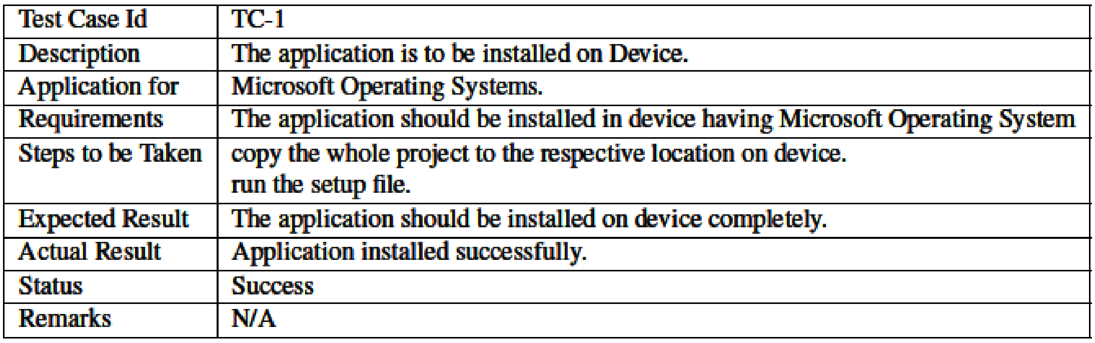
To run this system Microsoft OS must be installed on the device. This system is built on .NET framework 4.5, so.NET framework 4.5 or a greater version must be installed.

6.7 Test Cases

Test cases for the evaluation of FQVRS are displayed under.

6.7.1 Test Case 1

The first test case states that the application should be installed on the device. For the application to be successfully installed, the device should be running on a Microsoft OS. To carry out this task, the user must copy the whole project into the desired location and run the setup file.

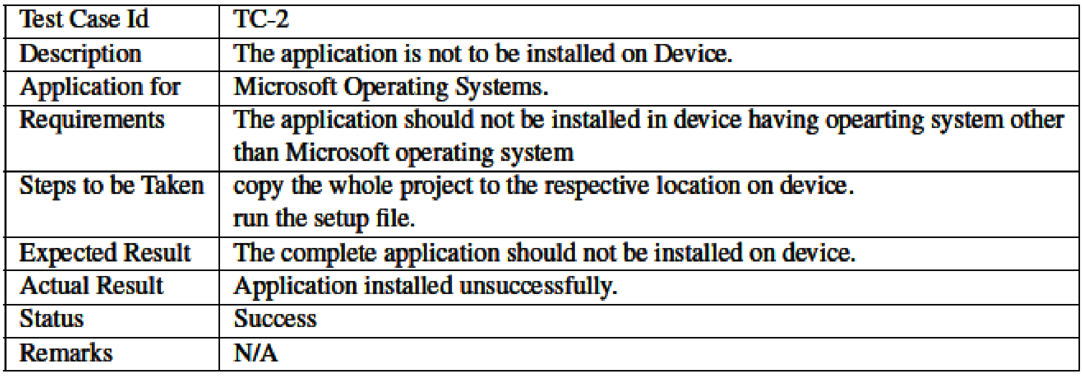


Test Case Id	TC-1
Description	The application is to be installed on Device.
Application for	Microsoft Operating Systems.
Requirements	The application should be installed in device having Microsoft Operating System
Steps to be Taken	copy the whole project to the respective location on device. run the setup file.
Expected Result	The application should be installed on device completely.
Actual Result	Application installed successfully.
Status	Success
Remarks	N/A

Figure 6.7: Test Case 1.

6.7.2 Test Case 2

The second test case states that the application should not be installed on the device. For the application to be successfully installed, the device should not be running on an Os except for the Microsoft OS.

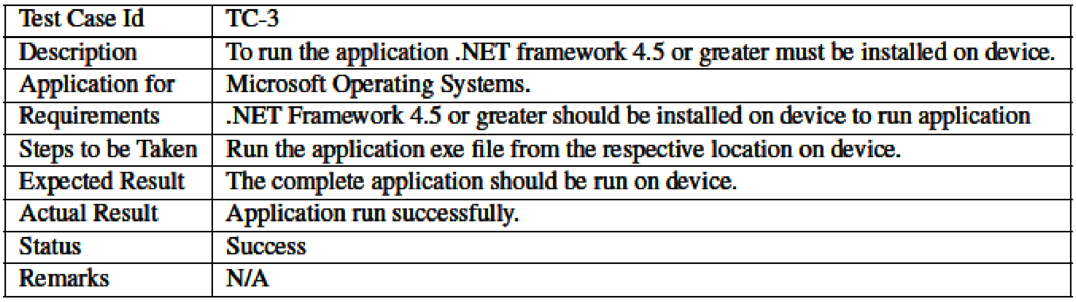


Test Case Id	TC-2
Description	The application is not to be installed on Device.
Application for	Microsoft Operating Systems.
Requirements	The application should not be installed in device having opearting system other than Microsoft operating system
Steps to be Taken	copy the whole project to the respective location on device. run the setup file.
Expected Result	The complete application should not be installed on device.
Actual Result	Application installed unsuccessfully.
Status	Success
Remarks	N/A

Figure 6.8: Test Case 2.

6.7.3 Test Case 3

This test case states that in order to run the application, the user must have .NET Framework 4.5 or higher version running on the device.

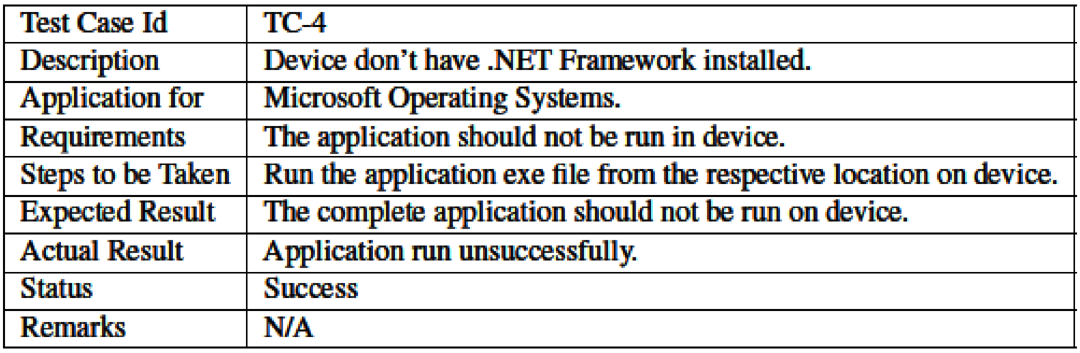


Test Case Id	TC-3
Description	To run the application .NET framework 4.5 or greater must be installed on device.
Application for	Microsoft Operating Systems.
Requirements	.NET Framework 4.5 or greater should be installed on device to run application
Steps to be Taken	Run the application exe file from the respective location on device.
Expected Result	The complete application should be run on device.
Actual Result	Application run successfully.
Status	Success
Remarks	N/A

Figure 6.9: Test Case 3.

6.7.4 Test Case 4

This test case provides the information if the device is not running on .NET Framework 4.5 or a greater version.

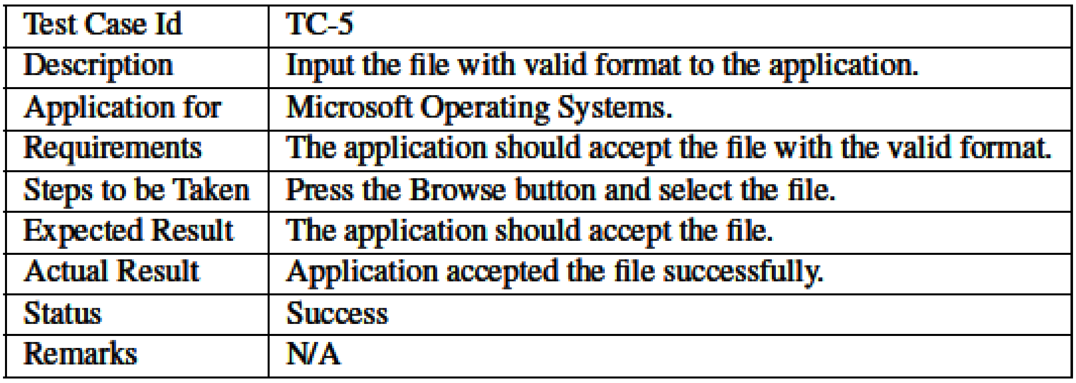


Test Case Id	TC-4
Description	Device don't have .NET Framework installed.
Application for	Microsoft Operating Systems.
Requirements	The application should not be run in device.
Steps to be Taken	Run the application exe file from the respective location on device.
Expected Result	The complete application should not be run on device.
Actual Result	Application run unsuccessfully.
Status	Success
Remarks	N/A

Figure 6.10: Test Case 4.

6.7.5 Test Case 5

This test case provides the solution to a valid format of the input file.



Test Case Id	TC-5
Description	Input the file with valid format to the application.
Application for	Microsoft Operating Systems.
Requirements	The application should accept the file with the valid format.
Steps to be Taken	Press the Browse button and select the file.
Expected Result	The application should accept the file.
Actual Result	Application accepted the file successfully.
Status	Success
Remarks	N/A

Figure 6.11: Test Case 5.

6.8 Accuracy Testing

This section describes the accuracy of FQVRS in the following departments.

1. Face Detection in Indexing Video.
2. Video Retrieval from only face images.
3. Video Retrieval from full images.

6.8.1 Face Detection in Indexing Video

The system is 70 percent accurate in detection of faces during the indexing of video. This means that the number of False Positives and True Negatives add up to 30 percent of the total number of faces.

6.8.2 Video Retrieval from Only Face Images

The system is 90 percent accurate in recognition of faces during the retrieval of video.This means that the number of False Positives are 10 percent of the total number of face images in our dataset.

6.8.3 Video Retrieval from full images.

The system is 50 percent accurate in detection of faces during the indexing of video. This means that the number of False Positives and True Negatives add up to 50 percent of the total number of query images.

7 Conclusion

The project, Facial Image Query Base Video Retrieval System, is intended for Pakistan-based news channels which will assist them in logging and video retrievals for future use. The project is also applicable for intelligence agencies for surveillance purposes.The main objective of the project is to provide the facilities of indexing and retrieving the video based on facial images and retrieval of the video is based on the user query. It has been a prodigious and fruitful effort. After developing the project, we recognize the basic image processing and its application. We also gain a perspective on how the image processing is implemented on a desktop application.We have now a great idea on how the system testing is performed on a project and have also learned how to solve the problems generated by integrating the units of the system.

Bibliography

[1] Baback Moghaddam Alex Pentland and Thad Turner. View based and modular eigenspaces for face recognition, November 1994.

[2] Tinne Tuytelaars Herbert Bay and Luc Van Gool. Surf: Speeded up robust features, 2008.

[3] Sourabh Sethi Madhav Gitte, Harshal Bawaskar and Ajinkya Shinde. Content based video retrieval system, August 2014.

[4] Alex P. Pentland and Matthew A. Turk. Face recognition using eigenfaces, June 1991.

[5] M. Ravinder and T. Venugopal. Content-based video indexing and retrieval using key frames texture, edge and motion features, April 2016.

[6] Abdenour Hadid Timo Ahonen and Matti Pietikainen. Face recognition with local binary patterns, January 2017.

[7] Emina Torlak. Uml sequence diagrams, February 2015.

[8] Thomas Risse Wolfgang Nejdl, Norbert Fuhr and Laszlo Kovacs. Research and advanced technology for digital library, September 2016.