

SMART IMAGE SEARCHING



Youshra Ibrahim

Tayyaba Noor

Hasaan Sultan

Rizwan Ishaq

Supervised By

Imran Asif

*Submitted for the partial fulfillment of BS Software Engineering degree to
the Faculty of Engineering & Computer Science*

**NATIONAL UNIVERSITY OF MODERN LANGUAGES
ISLAMABAD**

JULY, 2018

ABSTRACT

Smart Image Searching (SIS) is a windows based application which helps in searching and retrieving images from a database of digital images and videos combined. The features of a face from images and videos are extracted for face recognition and saved in a database. When an image of a person is required to search, the SIS detects and matches features of the face of the person from the database. It displays all the photos and video frames containing the image of the person. The information about location, size and file name of the target image is also displayed. The SIS searches for an image on the basis of name, date and size as well.

The application has been developed using Visual Studio and SQL Server Management Studio. The Viola Jones Haar Cascade, LBPH algorithm and Emgu CV library is used for face detection, face recognition and image processing respectively.

The system is verified and validated through testing. Each functionality is tested separately using unit testing technique. The integration testing is performed to test the functioning of complete system. White box testing is also performed to check the internal structure and working of a system. All the features of the application are working satisfactorily.

CERTIFICATE

Dated: _____

Final Approval

It is certified that project report titled '**Smart Image Searching**' submitted by **Youshra Ibrahim, Tayyaba Noor, Hasaan Sultan and Rizwan Ishaq** for the partial fulfillment of the requirement of "**Bachelors Degree in Software Engineering**" is approved.

COMMITTEE

Dr. Muhammad Akbar

Dean FE & CS

Signature: _____

Col (R) Engr. Jamil Ahmad Zia

HoD Engineering

Signature: _____

Waris Ali

Head Project Committee

Signature: _____

Imran Asif

Supervisor

Signature: _____

DECLARATION

We hereby declare that our dissertation is entirely our work and genuine / original. We understand that in case of discovery of any PLAGIARISM at any stage, our group will be assigned an F (FAIL) grade and it may result in withdrawal of our Bachelor's degree.

Group Members

Signature

1. Youshra Ibrahim

2. Tayyaba Noor

3. Hasaan Sultan

4. Rizwan Ishaq

PLAGIARISM CERTIFICATE

This is to certify that the project entitled “**Smart Image Searching**”, which is being submitted here with for the award of the “**Degree of Bachelors**” in “**Software Engineering**”. This is the result of the original work by **Youshra Ibrahim, Tayyaba Noor, Hasaan Sultan** and **Rizwan Ishaq** under my supervision and guidance. The work embodied in this project has not been done earlier for the basis of award of any degree or compatible certificate or similar title of this for any other diploma/examining body or university to the best of my knowledge and belief.

Turnitin Originality Report

Processed on 12-Jun-2018 03:46PM (UTC+0500)

ID: 974981971

Word count: 12155

Similarity Index

12%

Similarity by Source

Internet Sources: 07%

Publications: 0 %

Student Papers: 11%

Date: 12/06/2018

Mr. Imran Asif

TURNITIN ORIGINALITY REPORT

Smart Image Searching by **Youshra Ibrahim, Tayyaba Noor, Hasaan Sultan** and **Rizwan Ishaq**

From **Imran Asif**

Processed on 12-Jun-2018 03:46PM (UTC+0500)

ID: 974981971

Word count: 12155

Similarity Index

12%

Similarity by Source

Internet Sources:

07%

Publications:

1%

Student Papers:

11%

SOURCES:

1. 6% match (from student papers)

Submitted to Higher Education Commission Pakistan

2. 1% match (from Internet source)

www.brainpickings.org

3. 1% match (from Internet source)

cui.unige.ch

4. 1% match (from Internet source)

www.energycity2013.eu

5. 1% match (from Internet source)

docs.microsoft.com

6. 1% match (from Internet source)

computers4you.blogspot.com

- 7.** <1% match (from student papers)
Submitted to Loughborough University
- 8.** <1% match (from student papers)
Submitted to American Public University System
- 9.** <1% match (from Internet source)
chronicle.com
- 10.** <1% match (from Internet source)
en.wikipedia.org
- 11.** <1% match (from student papers)
Submitted to International School of Management and Technology
- 12.** <1% match (from student papers)
Submitted to Asia Pacific Institute of Information Technology
- 13.** <1% match (from student papers)
Submitted to Staffordshire University
- 14.** <1% match (from Internet source)
www.hongkiat.com
- 15.** <1% match (from Internet source)
www.richardbanks.com
- 16.** <1% match (from Internet source)
www.tricksmachine.com
- 17.** <1% match (from student papers)
Submitted to Nottingham Trent University
- 18.** <1% match (from publication)
Ayyub . "System Definition and Structure", Risk Analysis in Engineering and Economics
Second Edition, 2014.
- 19.** <1% match (from Internet source)
www.ipipan.waw.pl
- 20.** <1% match (from Internet source)
www.tisqa.org
- 21.** <1% match (from publication)

ACKNOWLEDGEMENT

From the depth of our heart we express our deep sincere gratitude to the **Almighty ALLAH** for the blessings He had bestowed upon us to do this work and enabled us to complete this dissertation successfully.

We are very much thankful to our DEAN SE&CS Dr. Muhammad Akbar and HEAD ENGG. Col(R). Jamil Ahmad Zia for their constant guidance help and support.

At this moment of accomplishment, first of all we pay homage to our supervisor Imran Asif. This work would not have been possible without his guidance, support and encouragement. Under his guidance, we successfully overcame many difficulties and learnt a lot. His kind attention thought provoking ideas and scholarly supervision have contributed in enrichment and completion of this project in time and befitting manner.

TABLE OF CONTENTS

Chapter	Page
Chapter 1: Introduction	1
1.0 Introduction.....	2
1.1 Problem Statement.....	2
1.2 Smart Image Searching	3
1.3 Aims And Objectives	3
1.4 Features Of The Smart Image Searching	3
1.4.1 Input Sample Image.....	3
1.4.2 Recognize Face From Sample Image	3
1.4.3 Save Face Features In Database	4
1.4.4 Match Face Features	4
1.4.5 Display Searched Images.....	4
1.4.6 Convert Video Into Frames	4
1.4.7 Display Frames That Match With Sample Image	4
1.4.8 Search Images Through Properties	4
1.5 Resources Requirements.....	4
1.5.1 Software Requirements.....	5
1.5.1.1 Visual Studio	5
1.5.1.2 SQL Server Management Studio	5
1.5.1.3 Emgu Cv.....	5
1.5.1.4 Operating System.....	4
1.5.1.5 Edraw Max.....	5
1.5.2 Hardware Requirements	6
1.6 Report Layout.....	6
Chapter 2: Existing System.....	7
2.0 Introduction	8

2.1	Existing Systems.....	8
2.1.1	CompFight.....	8
2.1.1.1	Limitations	8
2.1.2	Retrievr	8
2.1.2.1	Limitations	9
2.1.3	Multicolor Search Lab.....	9
2.1.3.1	Limitations	9
2.2	Comparison Between Proposed System And Existing Systems Features	9
2.3	Functional Requirements	10
2.3.1	Search Images From Image Gallery	10
2.3.2	Search Images Through Properties.....	11
2.3.3	Search Images From Video	11
2.4	Non Functional Requirements Of The System	11
2.4.1	Reliability	11
2.4.2	Maintainability.....	12
2.4.3	User Friendly	12
Chapter 3: Modeling.....		13
3.0	Introduction.....	14
3.1	System Flow Diagram Of Smart Image Searching (SIS)	14
3.1.1	Flow Diagram Analysis	14
3.2	Use Case Diagram Of Smart Image Searching	16
3.2.1	Give Sample Image Use Case	18
3.2.2	Give Directory Use Case	19
3.2.3	Search Images From Directory Use Case.....	21
3.2.4	Search Images Through Properties Use Case.....	23
3.2.5	Search Images From Video Use Case	25
3.2.6	View Matched Images Use Case	27

3.2.7	View Matched Frames Use Case	29
3.2.8	Search By Date Or Time Use Case	31
3.2.9	Search By Size Use Case	33
3.2.10	Search By Name Use Case	34
3.3	Sequence Diagram	37
3.3.1	Pre-Process Sequence Diagram	37
3.3.2	Convert Video Into Frames Sequence Diagram	38
3.3.3	Save Face Features Of Directory In Database Sequence Diagram	39
3.3.4	Save Face Features Of Sample Image In RAM Sequence Diagram	40
3.3.5	Search Images Through Properties Sequence Diagram	41
3.3.6	Display Matched Images Sequence Diagram	43
3.3.7	Give Sample Image Sequence Diagram	43
3.3.8	Search Images From Video Sequence Diagram	44
3.3.9	Select Video Sequence Diagram	46
3.3.10	Provide Directory For Search Sequence Diagram	46
3.3.11	Search Images From Gallery Sequence Diagram	47
3.4	System Sequence Diagram	48
Chapter 4: Development Methodology		50
4.0	Introduction	51
4.1	Development Track	51
4.2	Application Design	52
4.3	Application Implementation	52
4.3.1	Search Images From Image Gallery	52
4.3.2	Search Images From Video	53
4.3.3	Search Images Through Properties	53
Chapter 5: TESTING, ANALYSIS AND VALIDATION		54
5.0	Introduction	55

5.1	Test Bed	55
5.2	Give Sample Image Test Case.....	56
5.3	Give Directory Test Case	57
5.4	Search Images From Image Gallery Test Case	58
5.5	Search Images Through Properties Test Case	60
5.6	Search Images From Video Test Case.....	61
5.7	View Matched Images Test Case	63
5.8	View Matched Frames Test Case	64
5.9	Search images By Date Or Time Test Case	65
5.10	Search Images By Size Test Case.....	66
5.11	Search Images By Name Test Case	68
5.12	Testing Results.....	69
Chapter 6: Conclusion And Future Work.....		70
6.0	Introduction.....	71
6.1	System Overview And Methodologies	71
6.2	Limitations	72
6.3	Challenges Involved With The Implementation	72
6.3.1	Face Detection	72
6.3.2	Face Recognition	72
6.4	Future Work.....	72
Appendices.....		74
Appendix I–Installation Guide.....		74
References.....		75

LIST OF FIGURES

Figure	Caption	Page
3.1	Flow Chart of Smart Image Searching (SIS).....	15
3.2	Use Case Diagram of Smart Image Searching (SIS).....	17
3.3	Input Sample Image Use Case.....	18
3.4	Give Directory Use Case.....	20
3.5	Search Images from Directory Use Case.....	22
3.6	Search Images through Properties Use Case.....	24
3.7	Search Images from Video Use Case.....	26
3.8	Display Matched Images Use Case.....	28
3.9	Display Matched Frames Use Case.....	30
3.10	Search Image by Date or Time Use Case.....	32
3.11	Search Image by Size Use Case.....	33
3.12	Search Image by Name Use Case.....	35
3.13	Pre-Process Sequence Diagram.....	38
3.14	Convert Video into frames Sequence Diagram.....	39
3.15	Save face features of Directory in Database Sequence Diagram.....	40
3.16	Save face features of sample image in RAM Sequence Diagram.....	41
3.17	Search Image through Properties Sequence Diagram.....	42
3.18	Display matched images Sequence Diagram.....	43
3.19	Give Sample Image Sequence Diagram.....	44
3.20	Search images from Video Sequence Diagram.....	45

3.21	Select Video Sequence Diagram.....	46
3.22	Provide directory for search Sequence Diagram	46
3.23	Search images from gallery Sequence Diagram.....	47
3.24	System Sequence Diagram	48
3.26	System Sequence Diagram.....	49

LIST OF TABLES

Table	Caption	Page
1.1	Hardware Requirements	06
2.1	Comparison between Existing Systems and the Proposed System Features. ...	10
3.1	Give Sample Image Use Case	20
3.2	Give Directory Use Case	21
3.3	Search Images from Image Gallery Use Case	22
3.4	Search Images through Properties Use Case	24
3.5	Search Images from Video Use Case	26
3.6	Display Matched images Use Case	28
3.7	Display Matched frames Use Case	30
3.8	Search Image by Date or Time Use Case	32
3.9	Search Image by Size Use Case	34
3.10	Search Image by Name Use Case	36
5.1	Give Sample Image Use Case	56
5.2	Give Directory Use Case	57
5.3	Search Images from Image Gallery Use Case	59
5.4	Search Images through Properties Use Case	60
5.5	Search Images from Video Use Case	62
5.6	View matched images Use Case	63
5.7	View matched frames Use Case	64
5.8	Search Images by Date or Time Use Case	65
5.9	Search Images by Size Use Case	67

5.10	Search Images by Name Use Case.....	68
-------------	-------------------------------------	----

CHAPTER 1

INTRODUCTION

1.0 Introduction

“Smart Image Searching “is a Window based application that will help user to quickly find particular digital images from bulk of images and also identify the presence or existence of person in a video. Photography industry needs a system through which it will find images of specific customer from majority of images and as well as from the specific video. Security agencies also need such system which help in detecting the culprit from the CCTV video.

We are proposing three methods of searching images via “search from image gallery”, “search from video” and “search image through image properties”. On the basis of these methods, system will search the images. The main requirement of two method “searching from image gallery and “searching from video” user has to provide sample image. System will search images by comparing with sample image. By using method “search from video” video is divided into frames, then the given sample image is compared with each frames. At the end matched frames will be displayed. The main purpose of this application is to provide efficient data to a customer and security agency in limited amount of time.

1.1 Problem Statement

Searching a specific image from certain directory is not an easy task. Searching from thousands of pictures user have to look at each picture which is not only time consuming task but also more effort is required to verify each picture. User can provide multiple colors in traditional images searching software, in the end system provide thousands of images that may be irrelevant for user. Similarly searching a specific person’s image from video is also tough and troublesome task and user has to watch whole video. Smart image searching is the solution of these problems. User can search specific image by providing sample image as well as by providing properties of image. User just input sample image and select directory to search and system verify each image and return those images that matched with sample image in this way user don’t have to view each picture. Similarly by searching image from video, user just input sample image and select video that video converts into frames and match each frame with sample image and return those frames that match with sample image. For searching images through properties, user can provide specific image property, system will search all images and display images matched with the given image property.

1.2 Smart Image Searching

Smart Image Searching is a window based application that helps the user to search the images from bulk of images and also allow the user to detect and recognize the person without watching whole video. The proposed application, “Smart Image Searching (SIS)” also helps the user to search image by its provided properties like time, date, name and size (MB’s). The application allow the user to efficiently utilize the time. It provides friendly user interface and allow the user to choose different categories like search images from gallery, search from video and search image by its properties. This application reduces overall searching time along with the user effort.

1.3 Aims and Objectives

The aim of Smart Image Searching is to utilize time and will reduce user effort. That user can be student, teacher or can belong to any field, not necessarily to photography agency. The objective of Smart Image Searching also helps in security. The aim is to provide user with friendly user interface that help them in quickly finding particular images in limited amount of time without consuming lot of user time.

1.4 Features of the Smart Image Searching

The features of the proposed system are Input sample image, Recognize face from sample image, save face features in database, Match face features, Display searched images, Convert video into frame, Display frames that match with sample image and Search images through properties. Detail are as follows.

1.4.1 Input Sample Image

This feature allow the user to input sample image to system. On the basis of input sample image, the system will perform searching and return the resultant images. It is necessary for a user to provide sample image while searching images from these two methods “search images from image gallery” or “search images from video”.

1.4.2 Recognize Face from Sample Image

This features allow system to recognize face features from sample image provided by user. When face features are being recognized from sample image, it is then stored in RAM, later used for searching purpose.

1.4.3 Save Face Features in Database

This feature allows the system to save all the face features of directory in database after recognizing faces from directory images. This feature helps in such a way that if user wants more searching from another sample images, directory images already saved in database. This further reduces searching time. Before saving face features, all images are first converted into grey scale level.

1.4.4 Match Face Features

This feature allows the system to match the face features of sample image that is stored in RAM with the face features of directory stored in database. After matching procedure all images are displayed to user.

1.4.5 Display Searched Images

This feature allows the system to display the resultant images to user whose face features matched with the face features of sample image. Afterwards user can use these images for editing, sharing purpose etc.

1.4.6 Convert Video into Frames

This feature allows the system to convert the video provided by user into frames. The purpose of converting video into frames is to find particular image.

1.4.7 Display Frames that Match with Sample Image

This feature allows the system to display the frames whose face features matched with the face features of sample image.

1.4.8 Search Images through Properties

This feature allows the system to search the images by properties given by the user. User can search images through date, time, size and name. By using this method of searching user doesn't need to provide sample image.

1.5 Resources Requirements

The development of "Smart Image Searching" requires some hardware and software resources. The hardware resources include laptop or PC with processor core i3 or above. The software resources include Visual Studio 2013, SQL Server Management Studio 2012, EmguCV and AForge.NET

1.5.1 Software Requirements

Software requirements for the development of the desktop application are Visual Studio 2013, SQL Server Management Studio 2012, EmguCV and AForge.NET. Following are the software requirements:

1.5.1.1 Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs for Microsoft Windows, as well as web sites, web apps, web services and mobile apps [1]. Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger.

1.5.1.2 SQL Server Management Studio

The SQL Server Management Studio (SSMS) introduces user to the integrated environment for managing SQL Server infrastructure. SQL Server Management Studio presents a graphical interface for configuring, monitoring, and administering instances of SQL Server. It also allows you to deploy, monitor, and upgrade the data-tier components used by your applications, such as databases. SQL Server Management Studio also provides Transact-SQL, MDX, DMX, and XML language editors for editing and debugging scripts [2].

1.5.1.3 Emgu Cv

Emgu CV is a cross platform .Net wrapper to the OpenCV image processing library. Allowing OpenCV functions to be called from .NET compatible languages such as C#, VB, VC++, IronPython etc. The wrapper can be compiled by Visual Studio, Xamarin Studio and Unity, it can run on Windows, Linux, Mac OS X, iOS, Android and Windows Phone [3].

1.5.1.4 Operating System

Operating system used for the development of application is windows 8 or above.

1.5.1.5 Edraw Max

Edraw Max is all in one diagram software that make it simple to create professional looking flowcharts, networks charts, UML diagrams and many more [4]. Edraw Max is now available in two versions i.e. Free Viewer Version and Professional Editable version. From Edraw Max professional chart, Gantt chart and mind map from large number of data.

1.5.2 Hardware Requirements

Hardware requirements includes the requirements for developing and testing of software. Hardware requirements for both of the development and deployment of the project are discussed below in table 1.1.

Table 1.1 Hardware Requirements

Processor	Core i3 or higher
RAM	4GB or more
OS	Windows 8 or higher

1.6 Report Layout

In the first chapter of the project report, the project overview, project scope, problem statement, main features of project and the system requirements are discussed. In the second chapter, background of existing system, limitations in the existing systems, comparison between proposed system and existing system features are discussed and it also provides the software specification requirements(hardware, software) in details. The third chapter provides the system modeling which includes flowchart, use cases, system sequence diagram. In fourth chapter, methodologies that includes application design as well as application implementation methods are discussed. In fifth chapter, the system is being develop with the help of different test cases and the test result is also discussed. Last chapter includes the system overview and methodologies, limitations, challenges involved with the implementation phase, conclusion and the future work of the system.

CHAPTER 2

BACKGROUND & EXISTING WORK

2.0 Introduction

The chapter gives an overview of the background of the application and information regarding existing systems. The functionalities of the existing system are also discussed in this chapter. Limitation of the existing system are also discussed in this chapter and how our proposed system “Smart Image System” fulfil those limitations. The chapter assess the existing systems in general.

2.1 Existing Systems

There are many existing systems in the market that provide information regarding images searching. Some of these existing systems are more like searching images through colours and search video through text which provide all videos related to text you enter. For example if user enter disease cancer it will provide videos related to cancer. The detail of few systems are mentioned as follows.

2.1.1 CompFight

Compfight is an image search engine tailored to efficiently locate images for blogs, computers, inspiration, and research [5]. The various search options are presented in the left-hand column: user can search either just the tags associated with photos, or throughout all the text (title, description, comments, and tags). Search results are sometimes a little wonky—for example, repeating the same search will not always bring back the same results [6]. The tool also provides you with an embed code for proper attribution to the photographer or source, while it is very efficient in locating images by using the Flickr API – and this without actually being affiliated with Flickr.

2.1.1.1 Limitations

Flickr is kind of inconvenient when it comes to photo browsing. You type a keyword and hit search, Flickr by default returns results of 24 images, one after another. CompFight is limited to tag searching only. This cannot allow you to search image from directory.

2.1.2 Retrievr

Retrievr is an experimental image search tool that uses color recognition to retrieve Flickr images that reflect the spatial color arrangement of a digital sketch you draw on canvas. Retrievr has a Flash sketch pad built into the site. Draw something anything and it will fetch Flickr images that are similar [7].

2.1.2.1 Limitations

The drawback of this application is that if you are not good in sketching then you may fail in search. Secondly it will not allow you to search image from your directory.

2.1.3 Multicolor Search Lab

Easily the most useful color-based search tool yet, and also a throwback to our deep archives, Multicolor Search Lab offers a simple yet sophisticated way of finding images based on a color or color combination you're looking for. Images are pulled from more than 10 million of Flickr's most interesting photos and you can add up to 10 colors as your search criteria, including multiple swatches of the same color to indicate ratio say user wants an image that's almost entirely yellow with a bit of blue, you may select yellow four times and blue once [7].

2.1.3.1 Limitations

The drawback of Multicolor Search Lab is that this only allows you to choose from a limited number of colors. The process works best with fewer colors. The more colors that were added, the more TinEye struggled to find useful images and the more extraneous colors appeared.

2.2 Comparison between Proposed System and Existing Systems Features

The proposed system provide functionalities that existing systems does not provide. The applications like CompFight provide limited search result, returns results of 24 images as well as it is limited to tag searching only and will not allow user to search image from directory. Multicolor Search lab allow user to choose from limited number of colors. The more colors that were added, the more difficult to find useful images.

Searching images from Retrievr, user must have good drawing skills. Retrievr will not return any result if user is not good in drawing. And in return it takes lot of time, if user is not good in drawing and draw objects again and again. The major difference between Compfight, retrievr and proposed system is that the proposed system allow you to search image from gallery while Compfight and retrievr will not allow user to search images from your gallery. Our proposed system provide the user with the opportunity not only search particular images from directory but also return search result effectively and efficiently by utilizing time. Our proposed system also provide the facility by searching image through

its properties as well as searching images within video but existing system does not provide all these facilities.

As shown in the table 2.1 most of the existing system does not search image by date, time and size as well as does not provide the facility to search image within video.

Table 2.1 Comparison between existing system and proposed system features

Features	Proposed System	Retrievr	Compfight	MultiColor Search Lab
Face Detection	✓	✓	✗	✗
Face Recognition	✓	✓	✗	✗
Search images from gallery	✓	✓	✗	✗
Search images by Date & Time	✓	✗	✗	✗
Search images by Name	✓	✗	✓	✗
Search images by Size	✓	✗	✗	✗
Search image from Video	✓	✗	✗	✗

2.3 Functional Requirements

Functional requirements of the system are those requirements which are explicitly stated during requirement gathering phase. These are the requirements which will be must for the user. The functional requirements of the system are as follow:

2.3.1 Search Images from Image Gallery

The user will give a sample image to the system, the system will perform pre-processing on sample image and then recognize faces from it and save the face features of sample

image in RAM. Then the user will select the directory from which user wants to search particular images. The system will store face features of all images in database that are in directory in the case if images are not stored in database. If the images are already stored in database then system extract one by one face feature and compare with face feature of sample image. Finally matched images are stored in RAM and displayed to user.

2.3.2 Search Images through Properties

The system will provide the facility to search images by properties like date, time, size (MB's) and name. The user will provide the properties of the image for search and then select the directory from which user wants to search. After selecting the directory, the system will start matching the provided properties with the properties of images present in the directory and will display those images whose properties matched with the properties provided by the user.

2.3.3 Search Images from Video

The user will select the video and give the sample image to extract particular image from the selected video. The system first converts selected video into frames, then recognize faces in each frame and match with the sample image face. If any face in the frame matched with the sample image face, the system will display that frame.

2.4 Non Functional Requirements of the System

Nonfunctional requirements identifies the basic criteria to judge the functions of the system. Nonfunctional requirements are different from functional requirements which identify the main functionalities of system. In easy word Nonfunctional requirements describe how system works and functional requirements describe what should system do. These requirements are compulsory for the software system as the developers consider these requirements are must for the application by using different techniques. These requirements increase the quality of the software application.

2.4.1 Reliability

Reliability is an essential attribute of the system and Smart image searching is reliable in a way that always return approximate result against each single searching. Smart image searching satisfactorily perform the task for which it is designed that is searching images

through three different method search images from gallery”, “search images from video” and “search images through properties”. For example when user input sample image and select directory from where user want to search image then smart image searching application start matching face features of sample image with all face features of images in directory and return matched images. When we again input same sample image then the application will return same result.

2.4.2 Maintainability

Smart image searching application is maintainable and flexible because coding of application is done in a way that it can be used and maintained as well as understandable easily. Comments are mentioned with the coding lines of the system to make the user understand that what action is performed during every line of code.

2.4.3 User Friendly

Interfaces of the Application is designed in a way that it provide user with a friendly interface. Simple words are used so user easily understand what is required from user for example input sample image required to upload image which user want to search from specific directory or video. User do not need to understand any technical term for this application.

CHAPTER 3

SYSTEM MODELING

3.0 Introduction

In this chapter “System Modeling”, the complete use cases, their description, flow chart and sequence diagrams of each use case are explained.

3.1 System Flow Diagram of Smart Image Searching (SIS)

A flow diagram is a visual representation of the sequence of steps and decisions needed to perform a process. Each step in the sequence is noted within a diagram shape. Steps are linked by connecting lines and directional arrows. This allows anyone to view the flowchart and logically follow the process from beginning to end [8] .

The desktop application "Smart Image Searching" provides facility to user to search images from large number of images. This application provides three different searching methods for user ease that are “search images from gallery”, “search images from video” and “search images through properties”. For searching images from gallery and video, user must provide sample image. Application will match the images with the provided sample image and return matched images. In case of video, system first convert video into frames, match each converted frames with sample image and return resultant images. The desktop application provides unique feature that user’s time will be less consumed also if user again perform searching, more time reduces because face features of directory will be already saved in database. No manual searching is required.

3.1.1 Flow Diagram Analysis

In figure 3.1 complete system that is smart image searching has been analysed for any ambiguity or defects that causes the system to fail. In windows application, the system facilitate the user to search the required images in less time. User can choose different searching methods. Each method has its own different processes that need some requirements to complete searching task. This application provides three different searching methods for user ease that are “search images from gallery”, “search images from video” and “search images through properties”. If user wants to search images from image gallery then user selects “search images from image gallery” method after selecting the method user provide sample image and directory and click on search button then the system start searching and display those images to the user that matched with sample image or if user wants to search images from video then user selects “search images from video” method. If user wants to search images of specific size, name, date or time then user must

choose “search through properties” method. The system helps user to quickly search images in limited amount of time. But it is clearly mentioned, while searching for first time, it takes some time in pre-processing of images.

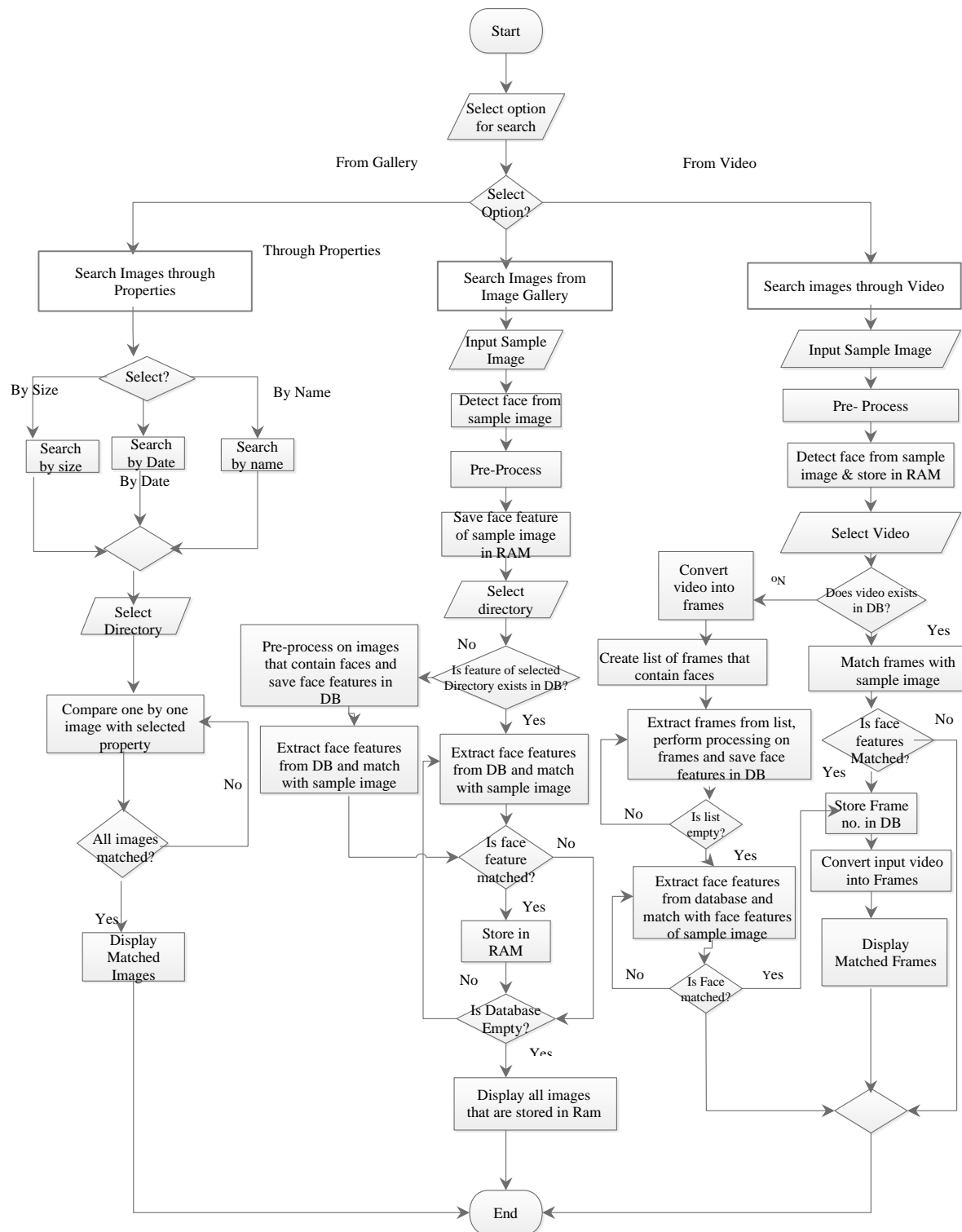


Figure 3.1 flow chart of Smart Image Searching

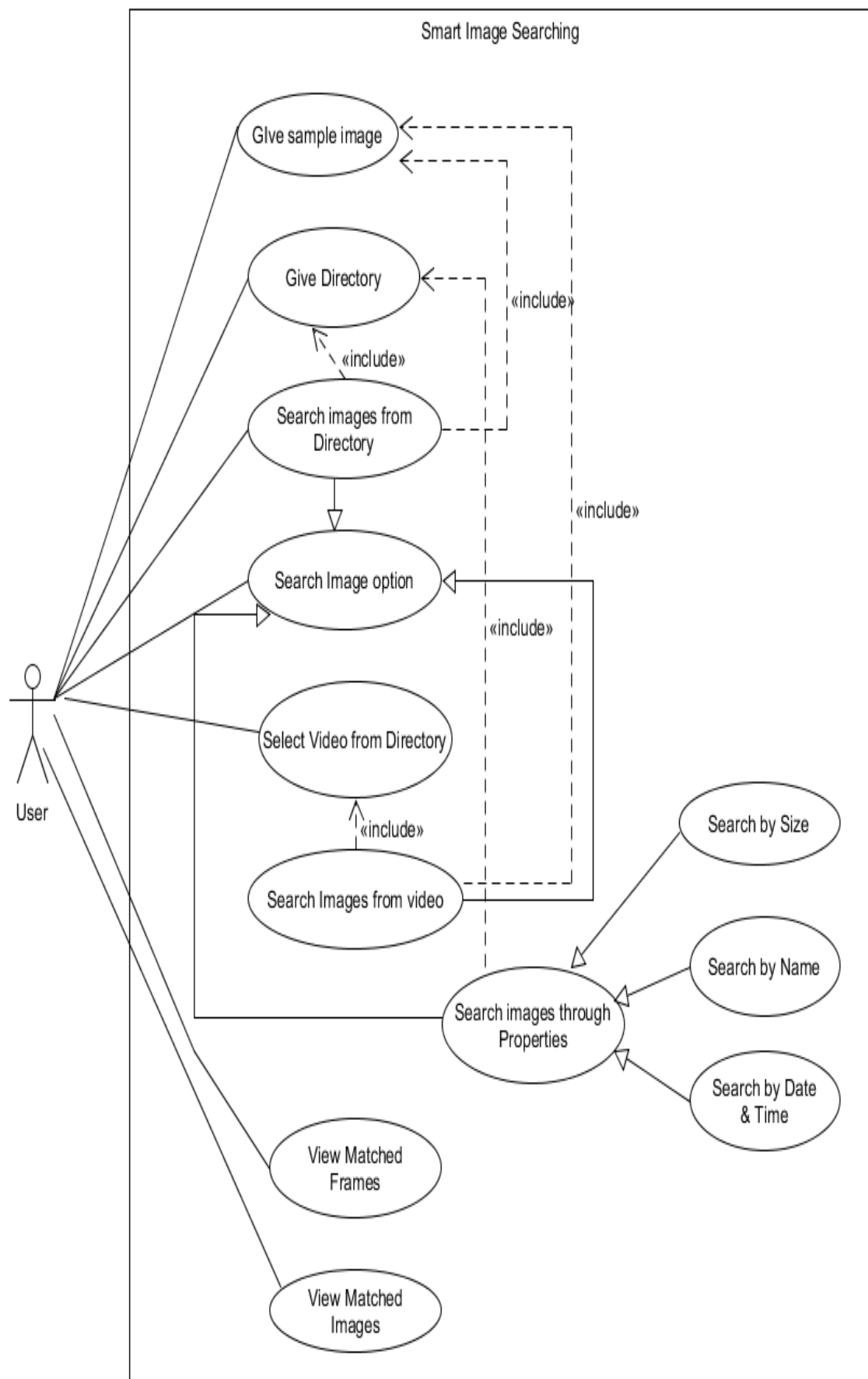
3.2 Use Case Diagram of Smart Image Searching

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements that can be used together in a way that has an effect larger than the sum of the separate elements combined [9].

The purpose of Smart Image Searching is to educate the user to search specific images quickly and efficiently. The basic functionality of system is to search images in less time.

In the use case diagram of “Smart Image Searching” actor will be any user. Anyone can use this application. User can select different option for searching purpose provided by application. This desktop application provides three different searching features to the user that are “search images from video, “search images from gallery and “search images through properties”. If the user selects “search images from image gallery” from search image option then the system opens “file open dialogue box” through which user selects sample image and also directory. After choosing sample image and directory, system will detect faces from sample image and from directory images after detection system perform pre-processing both on sample image and images in directory. In RAM face features of sample image will be store whereas face features of directory’s images will be stored in database. Now system will start comparing face feature of sample mage with each face features stored in database. During this procedure images that are matched continuously stored in RAM. After whole process complete the resultant images will be shown to user. Same process will be occur if user selects “search images from video” with an extra procedure that first video will be converted into frames and face feature of each frame will be store in database. Resultant frames will be shown to end user. User provide different properties like date, time, size and name when searching from “search images through properties”. System will match the images with input image’s property and return the result to user.

If the user run the system for the first time, the system will takes some time to perform pre-processing on images and to save face features of images that are present in directory in database. If the user has already used the application, then the face features were already saved in database, the system only extract face features from database, matches face features with the sample image and return resultant images to user as shown in figure 3.2.



3.2.1 Give Sample Image Use Case

The application allow the user to give sample image to system. On the basis of sample image the system perform searching and display images that are matched with sample image. The searching can be done through gallery or through video. The whole process is shown in figure 3.3.

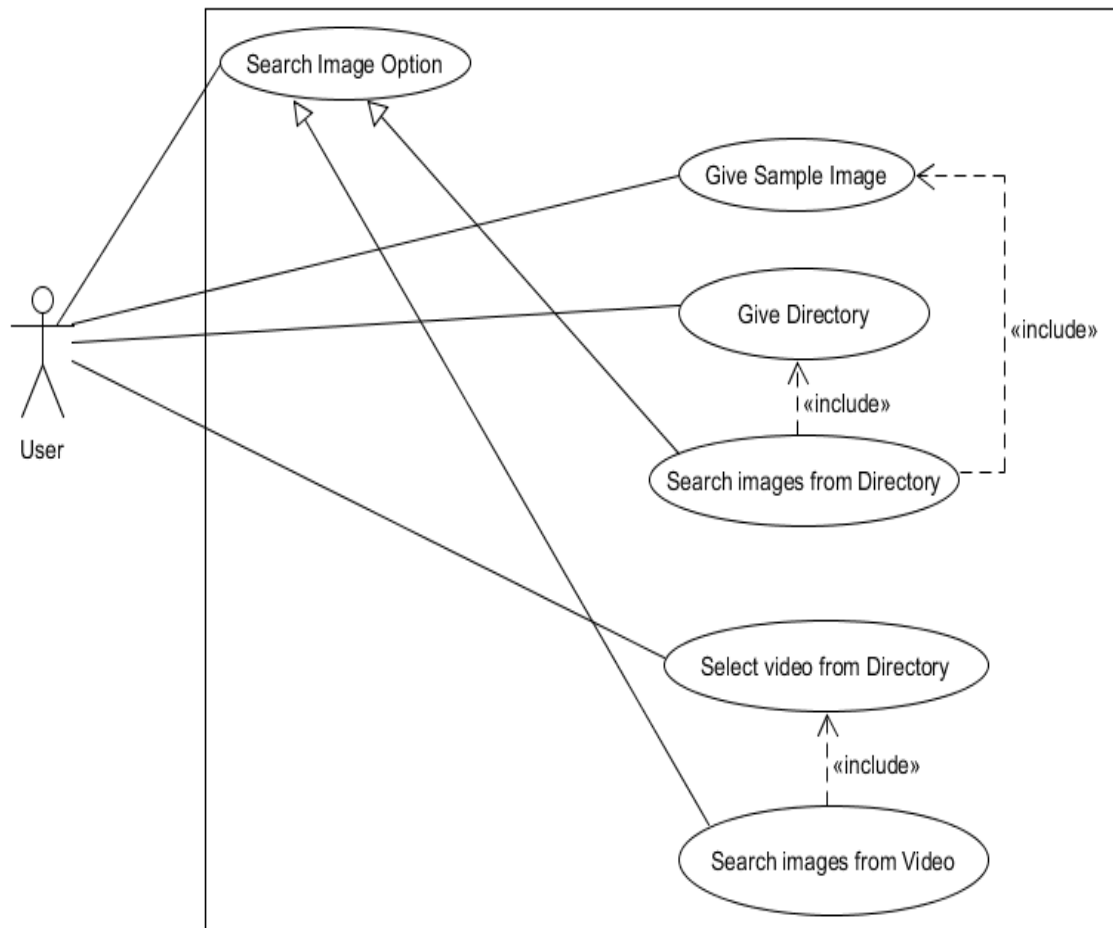


Figure 3.3 Give Sample Image Use Case

Table 3.1 Give Sample Image Use Case

Use case ID	UC-1
Use case name	Give Sample Image
Actors	User
Purpose	To give sample image for search.

Overview	User can give sample image of a person, on the basis of sample image system search related images.
Type	Primary
Pre-condition	User must select “search through gallery” and “search through video” option.
Post-condition	System will store sample image and display selected image to user.
Typical course of events	
Actors Actions	System Response
1. User press “input sample image” button.	1. System will open the “open file dialogue box”
2. User select the sample image.	2. System will save and show sample image”
Alternative Flow	
2a. User didn’t select any sample image	2b. System will not save and store sample image.

3.2.2 Give Directory Use Case

This use case described different methods for searching images from directory that include “Search images from image gallery”, “Search images from video” and “Search images through properties. If user wants to search images from video then user first select search image from video option then select video from specific directory. If user want to search image from image gallery then user first select search image from image gallery option then sample image and directory will be provided to system. If user wants to search images through properties then only directory and specific property of image will be provided. System will display selected directory to user. User will select directory from which user wants to search. System give flexibility to user to search images from any folder, file and directory. The whole procedure of selecting directory is shown in figure 3.4

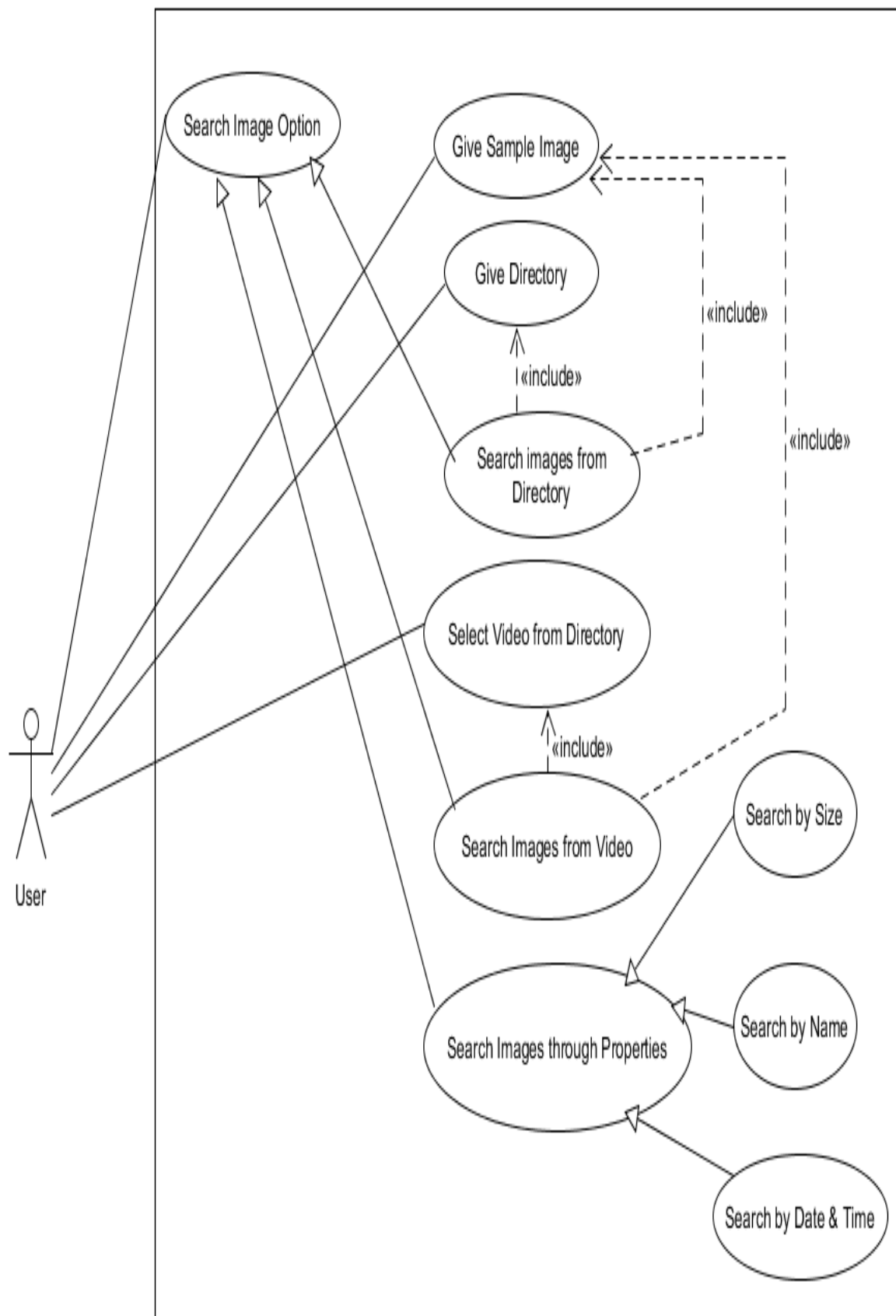


Figure 3.4 Give Directory Use Case Diagram

Table 3.2 Give Directory Use Case

Use case ID	UC-2
Use case name	Give Directory
Actors	User
Purpose	User select desired directory from which system perform searching and return matched images.
Overview	User can select directory from which system search related images.
Type	Primary and essential
Pre-condition	User must select “search through images” or “search through video” option.
Post-condition	Directory is shown to user.
Typical course of events	
Actors Actions	System Response
1. User will select “Search through Directory” button	1. System will open “open file dialogue box”.
2. User selects desired directory.	2. System will show the selected directory.
Alternative Flow	
2a. User may not select directory	2b. System will not show selected directory.

3.2.3 Search Images from Directory Use Case

Smart image searching provide the user to search images from directory. User first select searching option. Specifying directory for searching images. System pre-process directory

and detect the face from images. Now the face features of directory are being stored in Database. After this procedure face features are compared one by one with sample image provided by user earlier. Finally images are displayed to user which has been matched with sample image. Whole procedure is shown in figure 3.5.

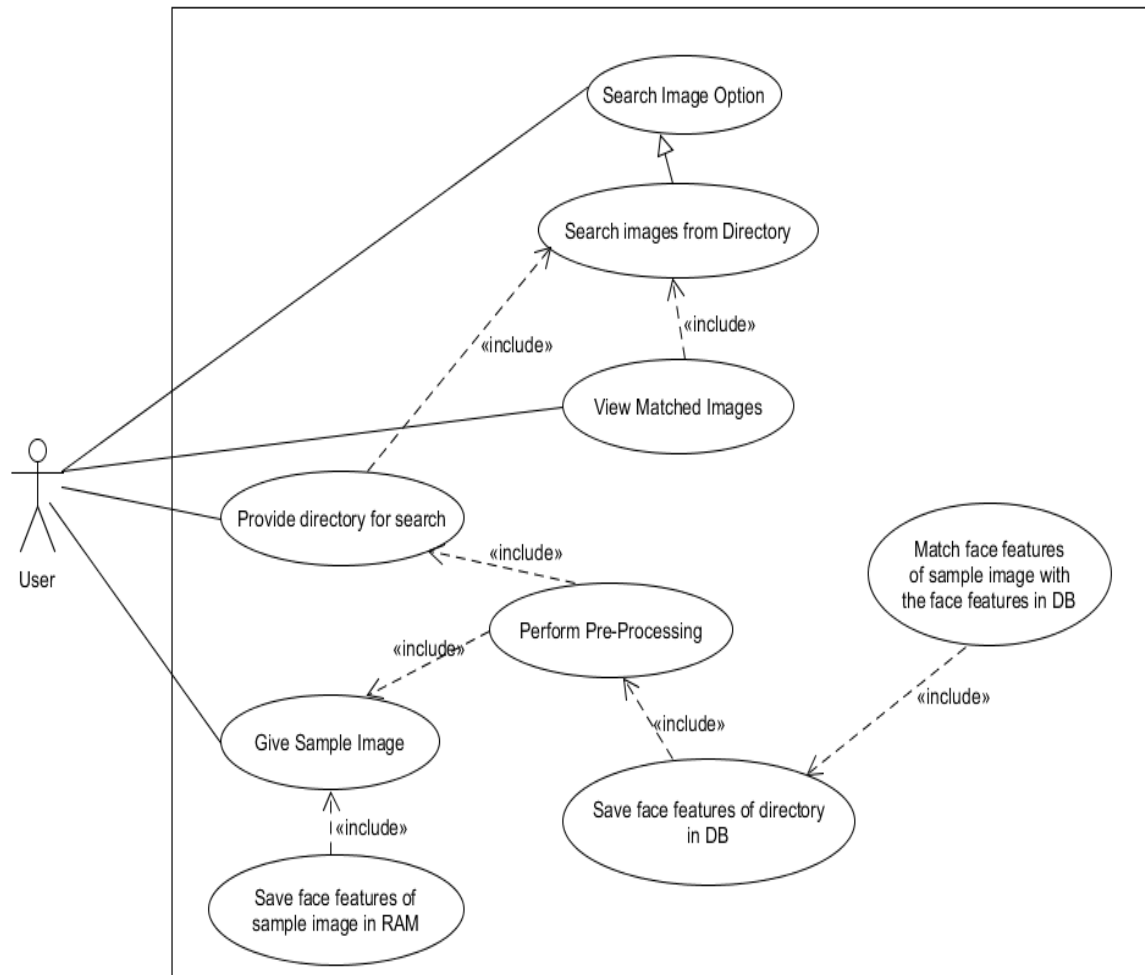


Figure 3.5 Search Image from Directory Use Case

Table 3.3 Search Image from Directory Use Case

Use case ID	UC-03
Use case name	Search Images from directory
Actors	User

Purpose	Search Image from selected directory
Overview	User will select directory and sample image then, press “search image from directory” button. System will start searching images.
Type	Primary
Pre-condition	User must give sample image and select directory for searching purpose.
Post-condition	User will display all matched images
Typical course of events	
Actors Actions	System Response
1. User give sample image and select Directory	1. System will save sample image and show selected directory to user.
2. User press “searching image from directory” button	2. System will start searching and display matched images.
Alternative Flow	
2a. User press “searching image from directory” button.	2b. Images not matched with sample image.

3.2.4 Search Images through Properties Use Case

First user will select the searching option provided by system. User can search image through image properties that are “search by date”, “search by size”, “search by name” and “search by size”. User will input image property and also select directory. System will start matching all image properties present in directory with input image property given by user. System will return all those images that matched with input image property. Procedure shown in figure 3.6.

All the other details regarding this use case is shown in table 3.4.

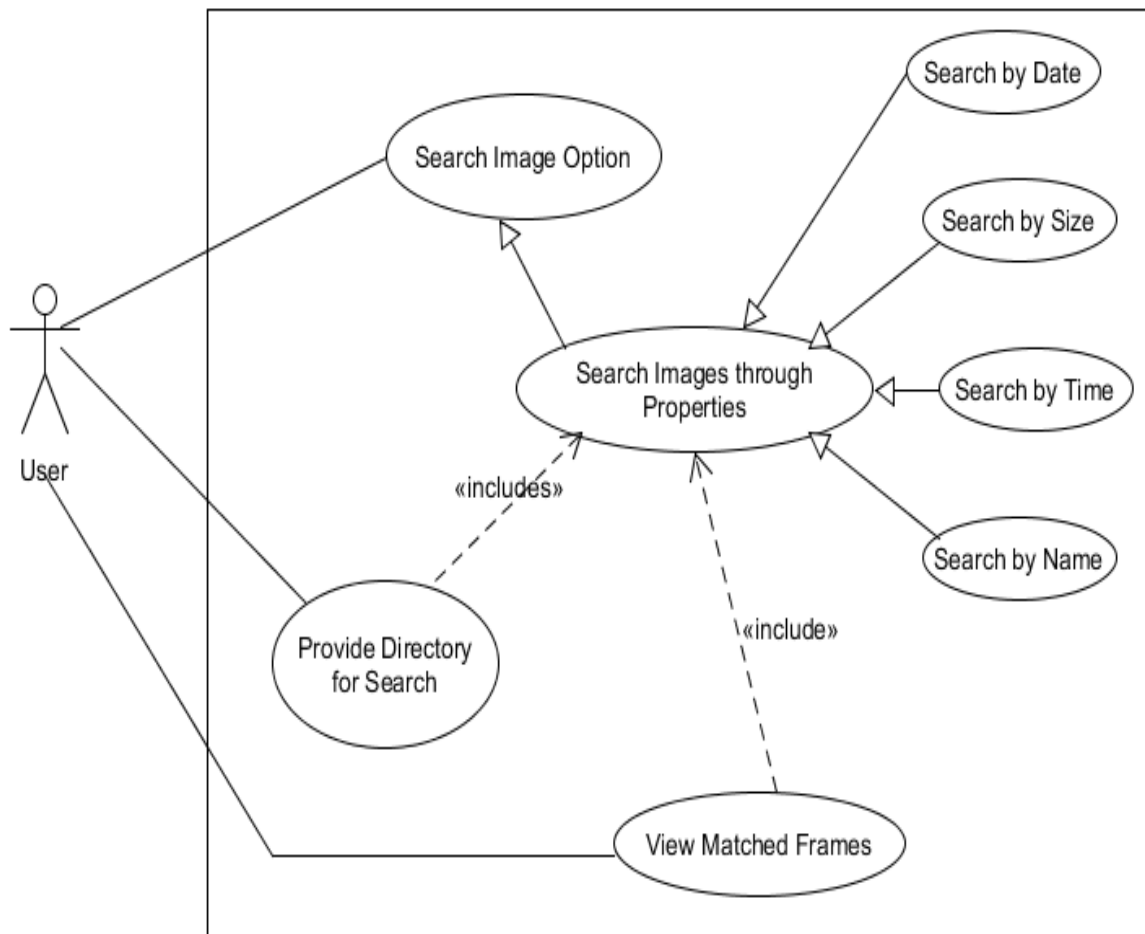


Figure 3.6 Search Images through Properties Use Case

Table 3.4 Search images through Properties Use Case

Use case ID	UC-4
Use case name	Search images through Properties
Actors	User
Purpose	To search images through properties for quick searching
Type	Primary
Pre-condition	System is in idle state

Post-condition	System will display images that matched with input image property like date, time, size and name.
Typical course of events	
Actors Actions	System Response
User select “search image through Properties” option and further provide specific image property	1. System will start searching and then display images that matched with input image property.
Alternative Flow	
1a. User provide image property	1b. System couldn’t find any image that matched with input image property.

3.2.5 Search Images from Video Use Case

User can also search the images from video. In order to Search from this method user must give sample image to system, then select specific directory. After selection of directory, user choose video from that directory. After that here actually system get involved. System will perform pre-processing on sample image detect face from sample image and save the face-features of sample image in RAM. Then system will check either video frames already present in database or not if frames are not present in database then system will convert the selected video into frames, perform pre-processing on video frames, detect faces from frames and its face features will be stored in database. Sample image’s face features were already stored in RAM. System will match the face-features of sample image that are stored in RAM with each face-features of frames that are stored in database. Store matched frame number in RAM and at the end the system match frame number in RAM with frame number in database. Finally it returns frames to user which will be matched with sample image. All the other details regarding this use case is shown in table 3.5.

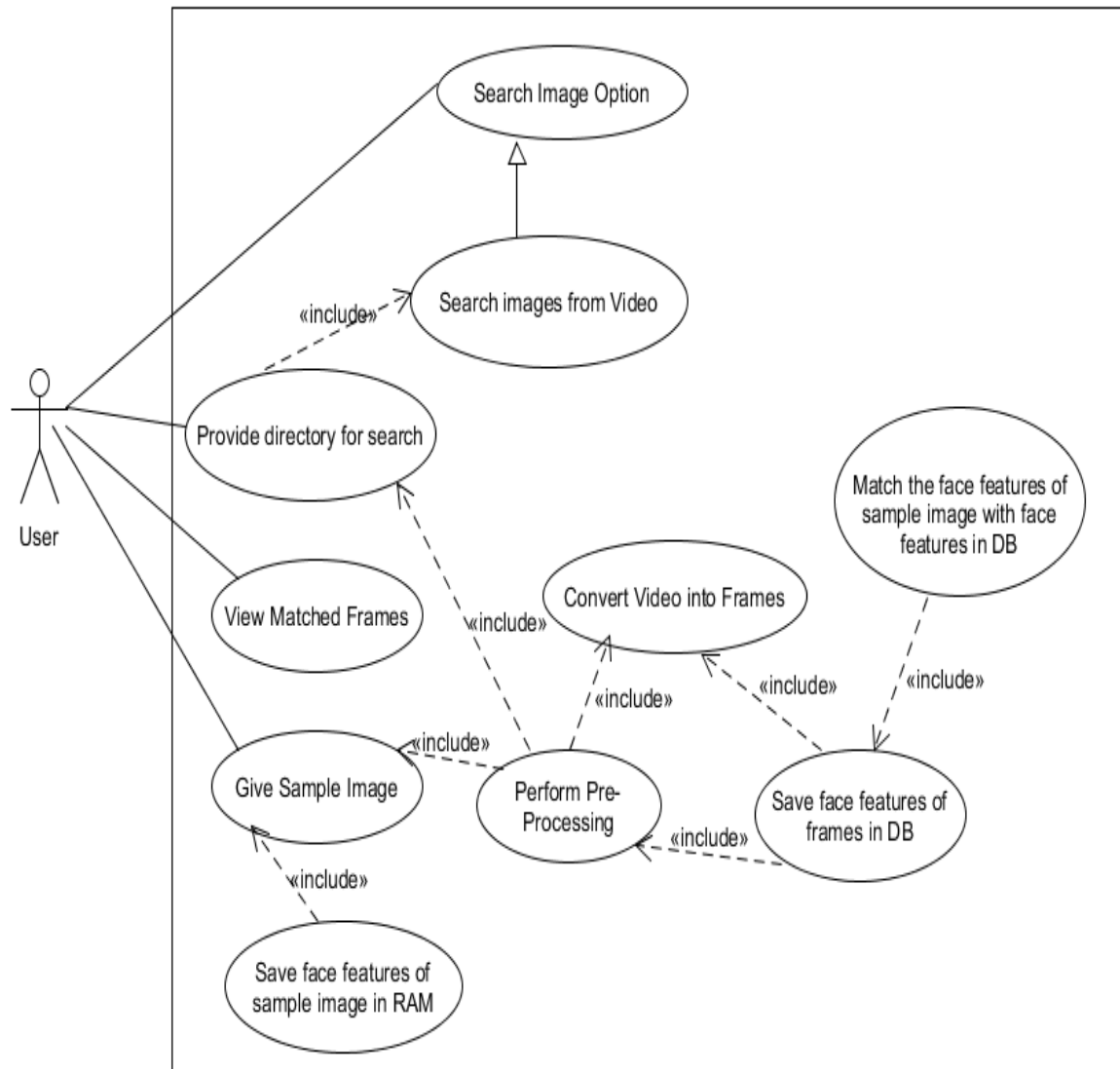


Figure 3.7 Search Images from Video Use Case

Table 3.5 Search Images from video Use Case

Use case ID	UC-5
Use case name	Search images from video
Actors	Users
Purpose	To search specific images from videos.
Overview	User give sample image, system perform pre-processing and match the image with video frames.
Type	Primary

Pre-condition	System is in idle state.
Post-condition	System return those frames that matched with sample image provided by user.
Typical course of events	
Actors Actions	System Response
1. Use give sample image and select directory for search	1. System will save sample image and show directory to user.
2. User select video from directory in order to search images from them	2. System will convert video into frames
3. User press “Search image from video” button	3. System will start searching and display matched frames
Alternative Flow	
3a. User press “Search image from video” button	3b. System doesn’t find any frames that matched with sample image

3.2.6 View Matched Images Use Case

System will display matched images when user selects one of the two searching option that is “Search image through properties” or “search image from image gallery”. After choosing any one of the two methods given above user will select directory. If user select “search image from image gallery” then system ask to input sample image. System will perform pre-processing on sample image and save face-features in RAM. Also the system will store face-features of directory’s images in database. System will start matching face-features of sample image with all face-features that are in database and return matched images. If user select “search image through properties”. System will ask user to choose directory and input image properties. System will start matching all image properties with input image properties and return those images whose properties are matched. Use case for display matched images is showing in figure 3.8. All the other details regarding this use case is shown in table 3.6.

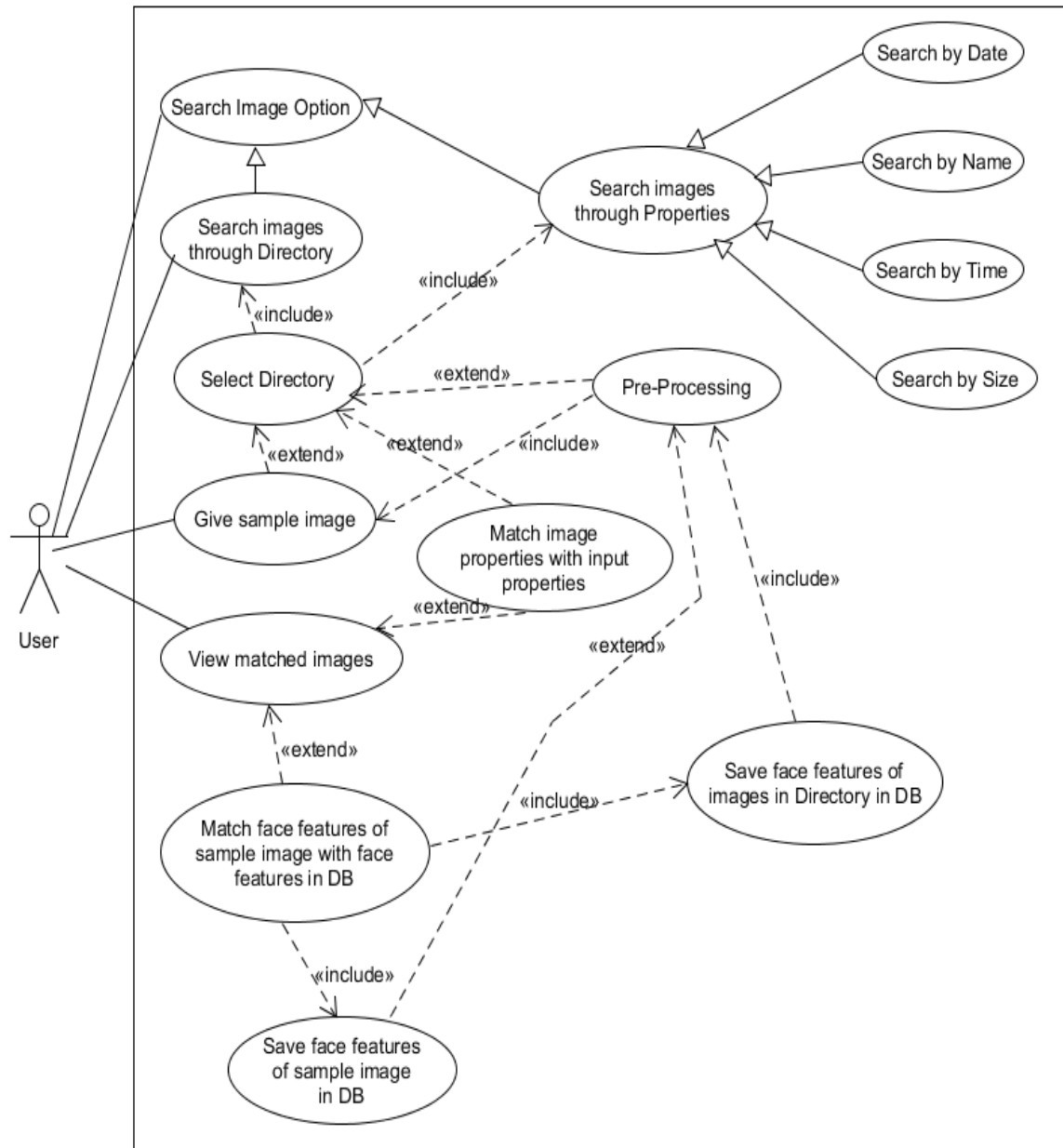


Figure 3.8 Display matched images Use Case

Table 3.6 Display matched images Use case

Use case ID	UC-6
Use Case Name	View matched images
Actors	User
Purpose	Display matched images to user.

Overview	User select searching option, system search and display matched images.
Type	
Pre-condition	User must select searching option.
Post-condition	System display matched images related to selected search option.
Typical Course of Events	
Actors Actions	System Response
1. User select search option.	1. System show searching methods to user.
2. User select specific searching method.	2. System start searching and display matched images.
Alternative Flows	
1a. User does not select search option.	1b. System does not show searching methods.
2a. User select specific searching option.	2b. system does not find images related to selected searching method.

3.2.7 View Matched Frames Use Case

User will select search option and then select search images from video option. For searching images from video, user must choose video from directory and give sample image to the system. The system will perform pre-processing on sample image and save face features of sample image in RAM then the system will start converting video into frames of equal time, that is one second and then perform pre-processing on each frame and save face features of frames in database and then match face features of sample image that are stored in RAM with face features of video frames that are stored in database store matched frame no in RAM and at the end the system match frame number in RAM with frame number in database. After completing whole procedure system will display matched frames. Use case for display matched frames is shown in figure 3.9

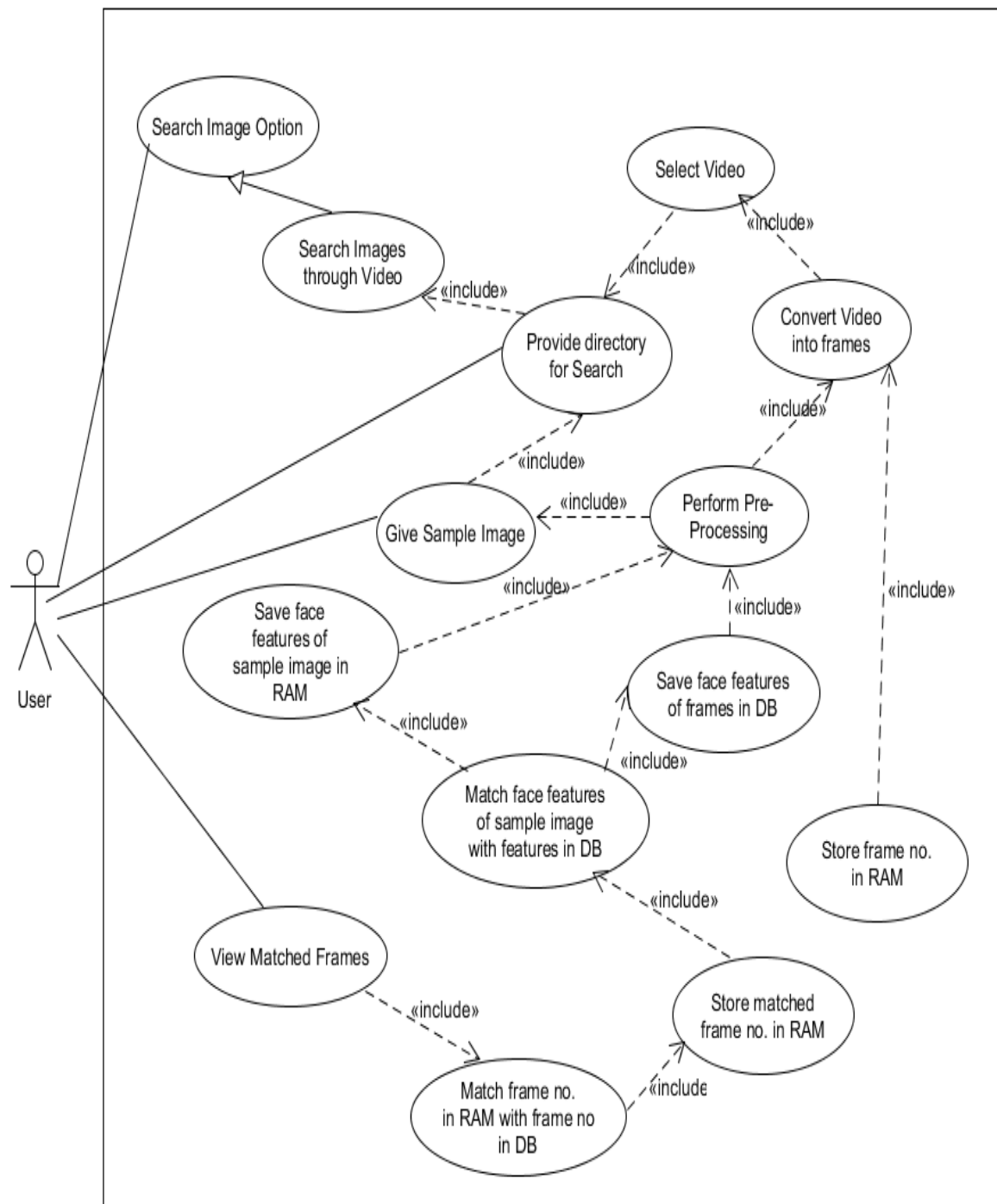


Figure 3.9 Display matched frames Use Case

Table 3.7 Display matched frames Use case

Use case ID	UC-7
Use Case Name	View matched frames

Actors	User
Purpose	Display matched frames to user
Overview	User select video and give sample image system convert video into frames then start searching and display those frames that matched with sample image.
Type	
Pre-condition	User must select searching option.
Post-condition	System display matched frames.
Typical Course of Events	
Actors Actions	System Response
1. User select search option.	1. System show searching methods to user.
2. User select video and sample image.	2. System convert video into frames, match frames with sample image and display matched frames.
Alternative Flows	
1a. User does not select search option.	1b. System will not show searching method.
2a. User select video and sample image.	2b. No frame match with sample image and system will not display any frames.

3.2.8 Search by Date or Time Use Case

User select search option and select search image through properties and input specific date or time then select directory. System match all images properties with input properties and return those images whose properties are matched. Use case search by date or time shown

in figure 3.10. Use case of search image by date or time is shown in figure and table. Use case has actor and system in the use case.

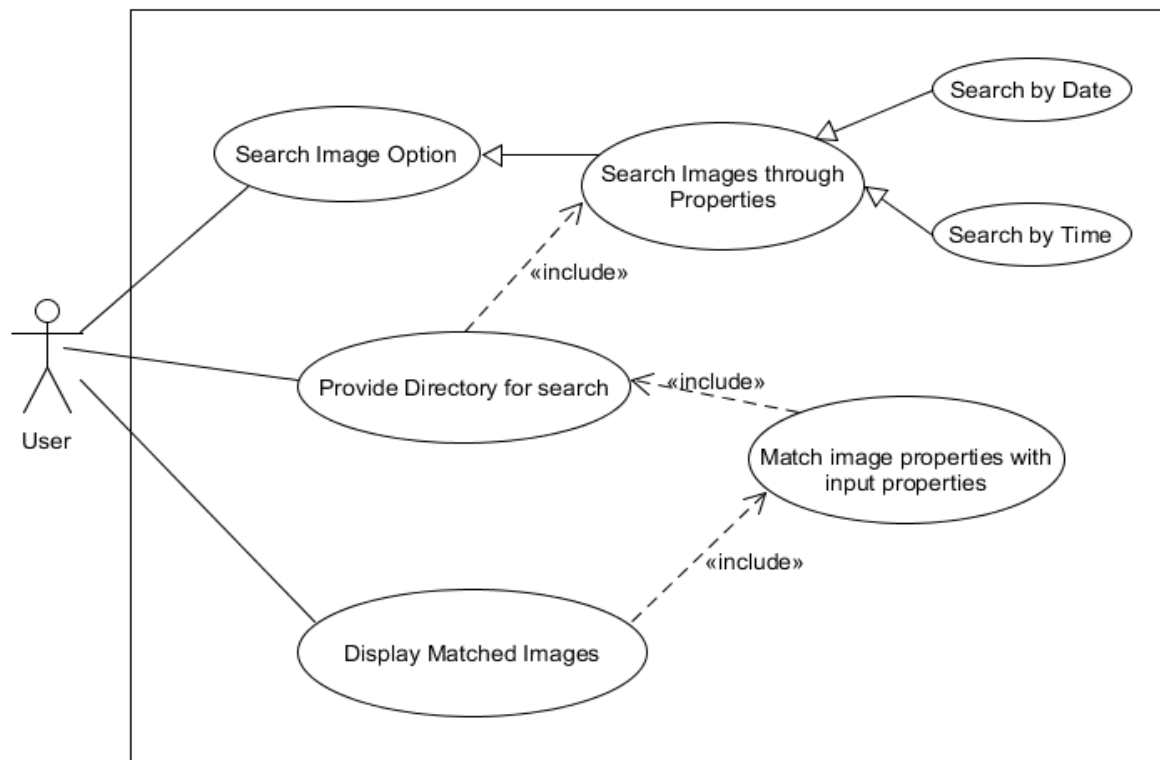


Figure 3.10 Search image by date or time Use Case

Table 3.8 Search image by date or time Use case

Use case ID	UC-8.
Use Case Name	Search image by date or time.
Actors	User.
Purpose	To search images by date or time.
Overview	User select search option then select image properties and input date or time.
Type	
Pre-condition	User must select search image through properties.

Post-condition	System display images whose date or time property match with input date or time properties.
Typical Course of Events	
Actors Actions	System Response
1. User select search through properties option.	1. System display search option like date, name or size.
2. User select search by date or time option.	2. System match images with input image properties and display images that matched with input image properties.
Alternative Flows	
2a. User select search by date or time option.	2b. System does not find images that match with input image properties.

3.2.9 Search by Size Use Case

User select search option and select search image through properties and input specific size then select directory. System match all images properties with input properties and return those images whose properties are matched. All the procedure is shown in figure 3.11.

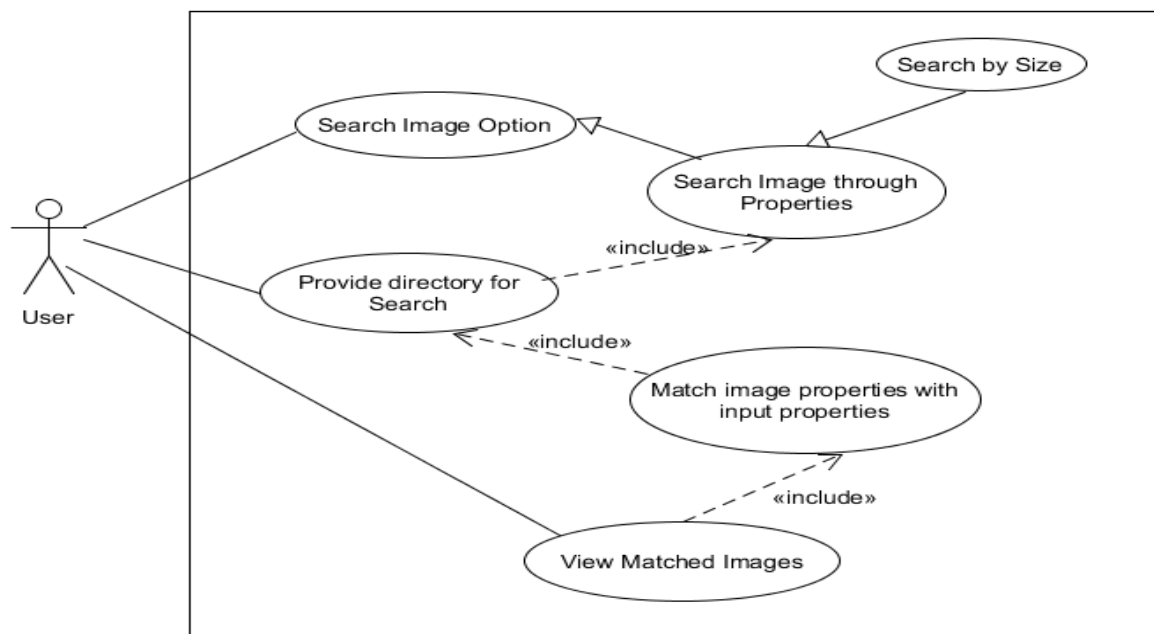


Figure 3.11 Search image by size Use Case

Table 3.9 Search image by size Use case

Use case ID	UC-9
Use Case Name	Search image by size.
Actors	User.
Purpose	To search images by size.
Overview	User select search option then select search image through properties and input size.
Type	
Pre-condition	User must select search image through properties.
Post-condition	System display images whose size property match with input size properties.
Typical Course of Events	
Actors Actions	System Response
1. User select search through properties option.	1. System display search option like date, name or size.
2. User select search by size option.	2. System match images size with input size and display matched size images.
Alternative Flows	
2a. User select search by size option.	2b. No image size matched with input size.

3.2.10 Search by Name Use Case

User will first select search option and then chose search image through properties. User will input specific name and also select directory. System will start matching all images properties with input properties and return those images that are matched with image

properties. Use case search by name shown in figure 3.12. Use case has actor and system in the use case

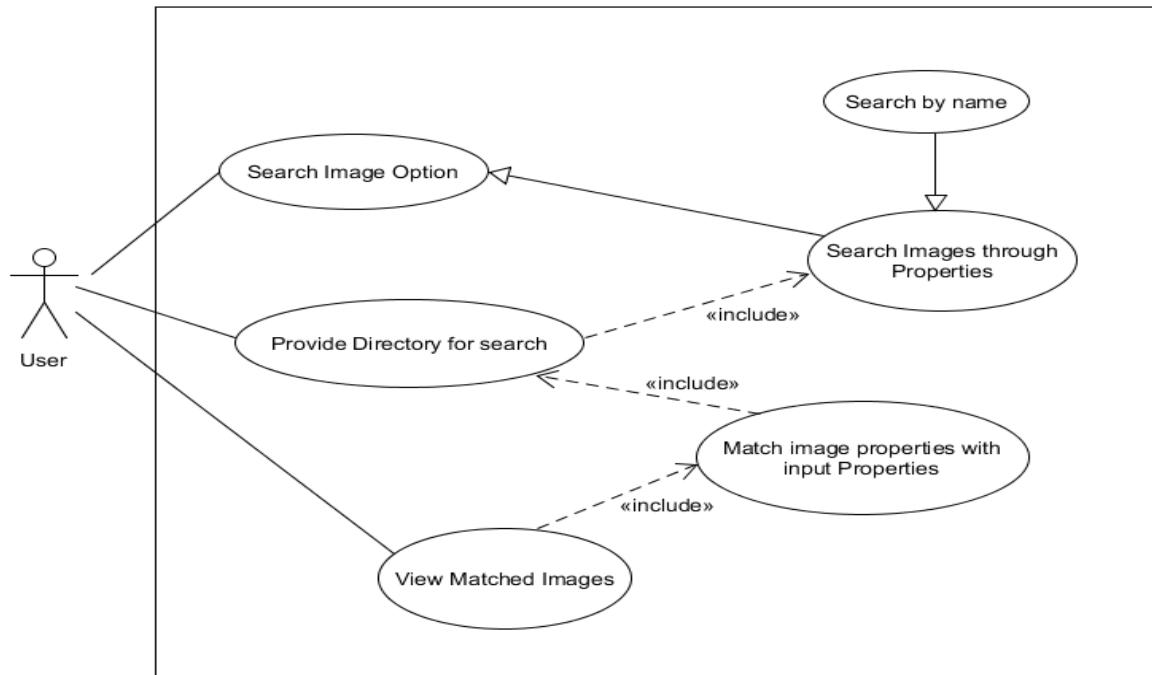


Figure 3.12 Search image by name Use Case

Table 3.10 Search image by name Use case

Use case ID	UC-10.
Use Case Name	Search image by name.
Actors	User.
Purpose	To search images by name.
Overview	User select search option then select search image through properties and input name.
Type	
Pre-condition	User must select search image through properties.
Post-condition	System display images whose name property match with input name property.

Typical Course of Events	
Actors Actions	System Response
1. User select search through properties option.	1. System display search option like date, name or size.
2. User select search by name option.	2. System match images name with input name and display matched name images.
Alternative Flows	
2a. User select search by name option.	2b. No image name matched with input name.

3.3 Sequence Diagram

The process between the user and the system is one of the most important part of any project. Sequence diagram helps with this process by displaying the interaction between user and system in a well-structured form.

A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the objects participating in the interaction by their lifelines, and the messages that they exchange arranged in time sequence. It does not show the associations among the objects. It represents an Interaction, which is a set of messages exchanged among objects within a collaboration to effect a desired operation or result. [10]

A sequence diagram has two dimensions: the vertical dimension represents time, the horizontal dimension represents different objects. Normally time proceeds down the page (although the dimensions may be reversed if desired). Usually only time sequences are important but in real-time applications, the time axis could be an actual metric. There is no significance to the horizontal ordering of the objects. [10]

In report sequence diagrams are drawn and explained against each use case. There are eleven sequence diagrams along its description. How user interact with the system? How system perform against each input?

3.3.1 Pre-Process Sequence Diagram

As shown in figure 3.6, at the start of the application, the system asks the user to select the method from which user wants to search images. User will select method. Pre-processing is a necessarily part of system. Pre-processing is done for two methods “search image from directory” and “search image from video”. When user will provide sample image to system, it will start pre-processing and recognizing the face feature, later store in RAM. Images that are in directory chosen by user will also goes through pre-processing stage which later will be store in database. On video pre-processing also performed that will be selected by user. System convert video into frames, perform pre-processing, detect and recognize the frame and will store the face features of frame in database. System will match face features of sample image with each face feature save in database and return matched images to user. The whole pre-processing process used in two methods “search images from gallery” and “search images from video” are shown in sequence diagram collectively

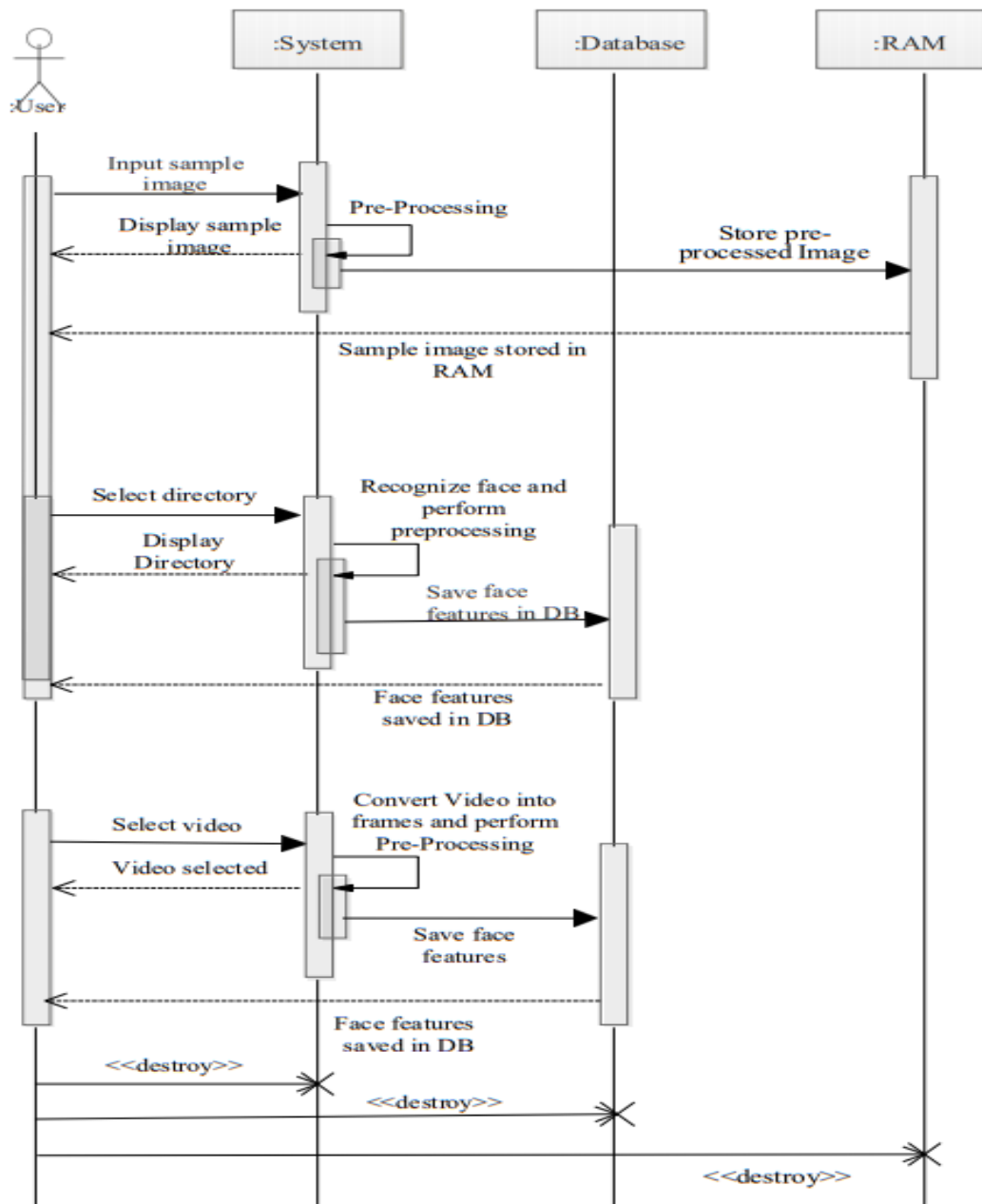


Figure 3.13 Pre-Process Sequence Diagram

3.3.2 Convert Video into Frames Sequence Diagram

This sequence diagram describes the procedure of converting video into frames. The user will select searching method “search image from video”. User will select directory and then select videos from which system search images as well as input sample image. System performs pre-processing on sample image detect face and save face-features in RAM. The selected video is then converted into frames, detect faces from frame, perform pre-processing and save the face features of frames in database. All the procedure is shown in figure 3.14.

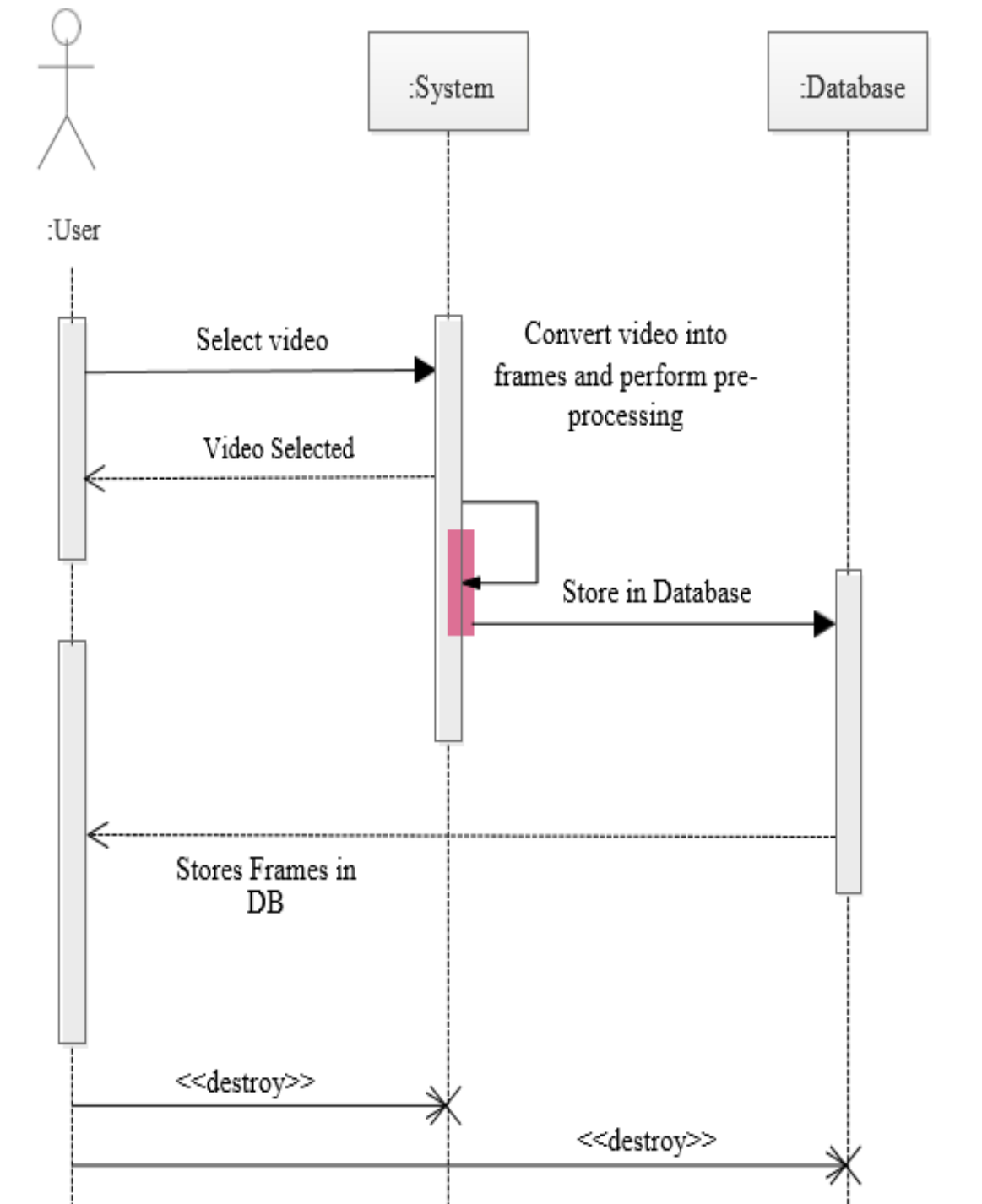


Figure 3.14 Convert Video into Frames Sequence Diagram

3.3.3 Save Face Features of Directory in Database Sequence Diagram

User will select “search images from image gallery”. System will ask user to select directory. After choosing directory, system will detect faces and create list of images that contain faces. System extract image from list perform pre-processing on image and save face-features of images in database. In the following figure 3.15 all the procedure is shown.

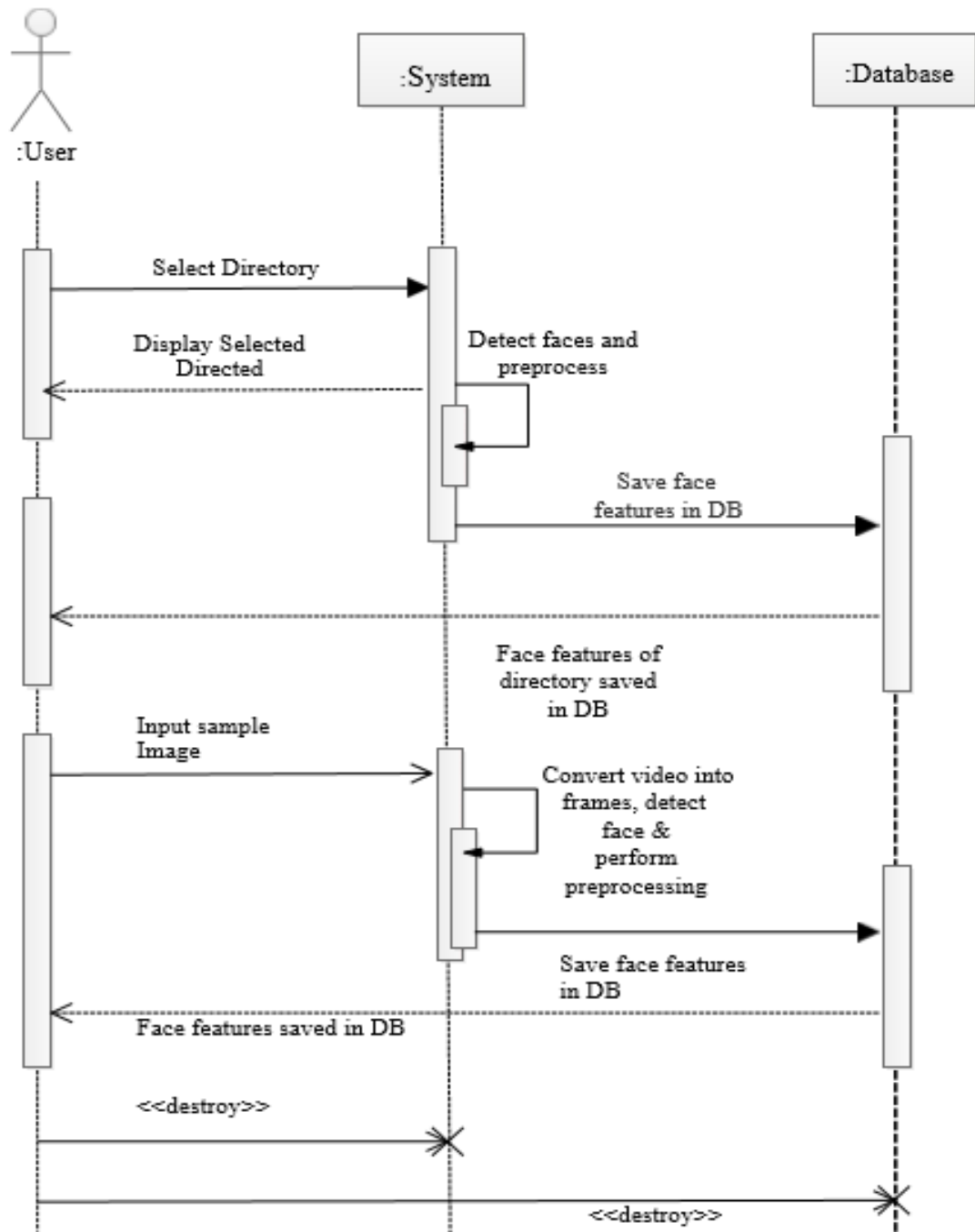


Figure 3.15. Save face features of Directory in Database Sequence Diagram

3.3.4 Save Face Features of Sample Image in RAM Sequence Diagram

Sample image is given to system. On the basis of sample image, the system perform searching. First face is detected from sample image and then it is being pre-processed. Pre-processing method convert image into grey scale image. Its features are saved in Random Access Memory. Later its face features are used to help in searching related images from gallery as well as from video. Sequence diagram is shown in figure 3.16

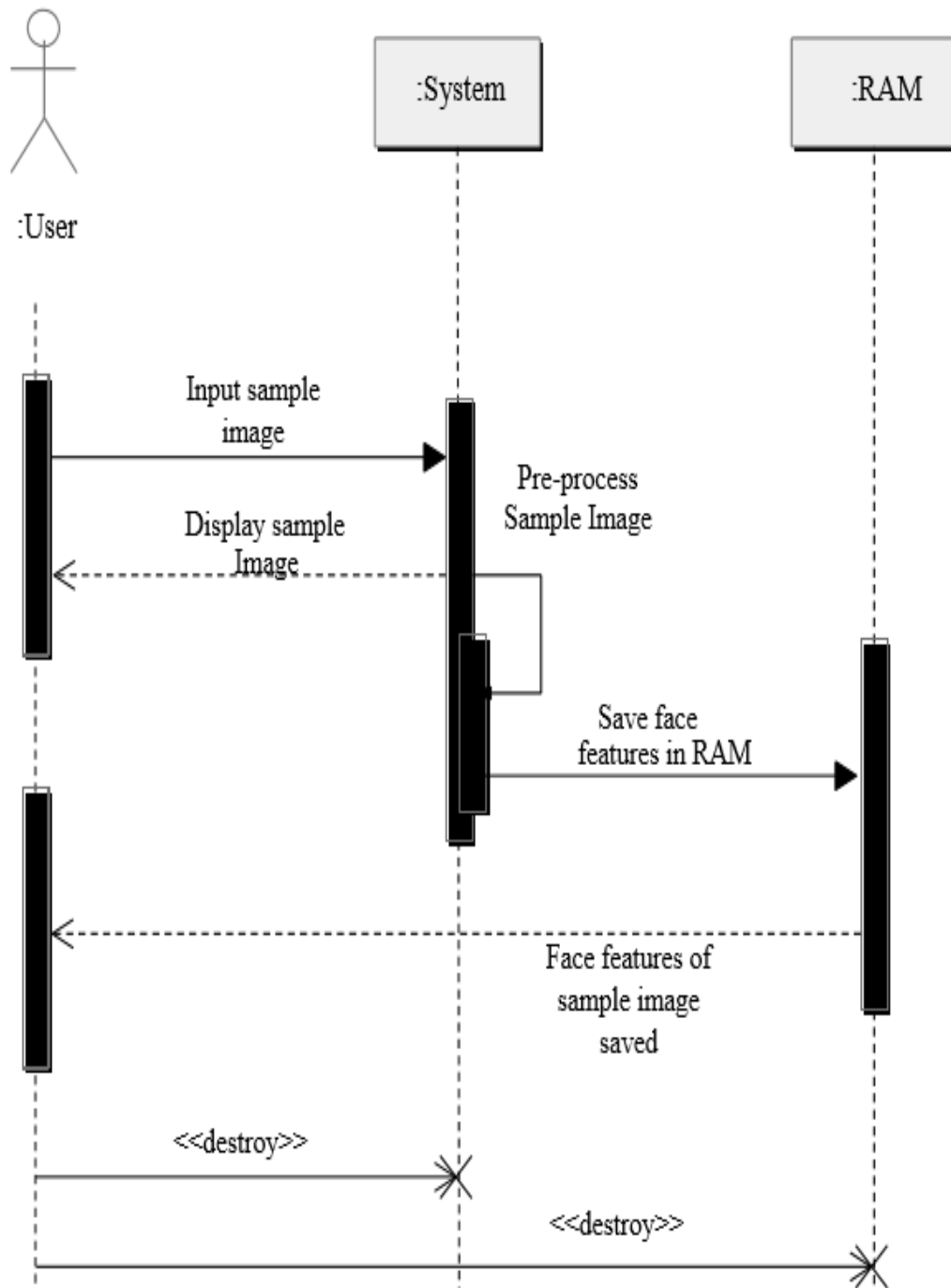


Figure 3.16 Save face features of sample image in RAM Sequence Diagram

3.3.5 Search Images through Properties Sequence Diagram

Another method of searching images is “Search images through Properties”. User input image property and also choose directory from which user want to search. Image properties include date, time, size and name. System start searching images by comparing with image properties selected by user. After searching system return images to user. All the procedure is shown in figure 3.17

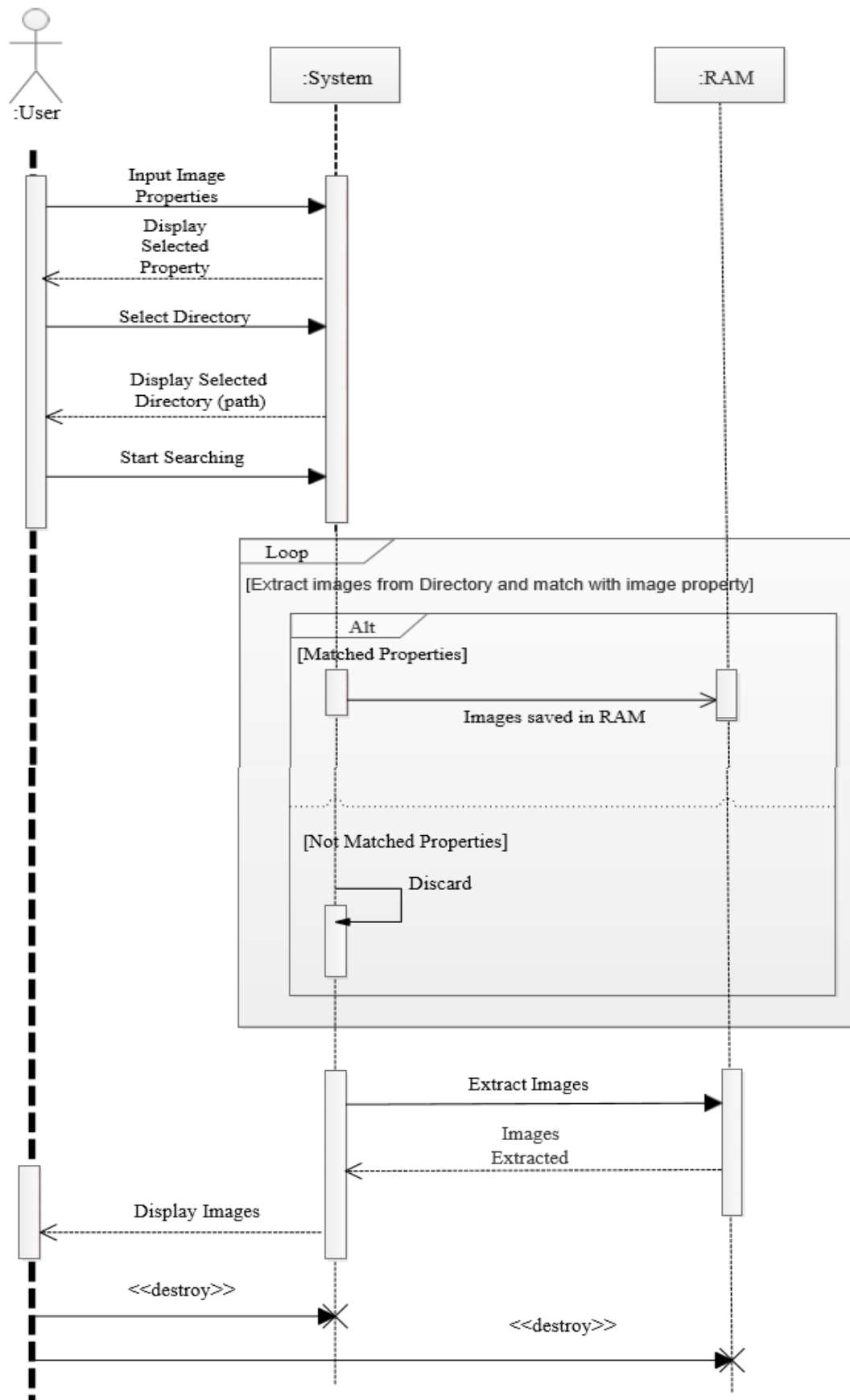


Figure 3.17 Search Image through Properties Sequence Diagram

3.3.6 Display Matched Images Sequence Diagram

User select search image through properties and search image through gallery. Searching images through properties user has to provide specific properties like date, time, name or size. On the basis of selected property system search and final result displayed to user. When searching through gallery user input sample image and select directory. System perform pre-processing on images that contain faces and save face features in database and match face features of sample image with face features in database and display matched images. All the procedure is shown in figure 3.18.

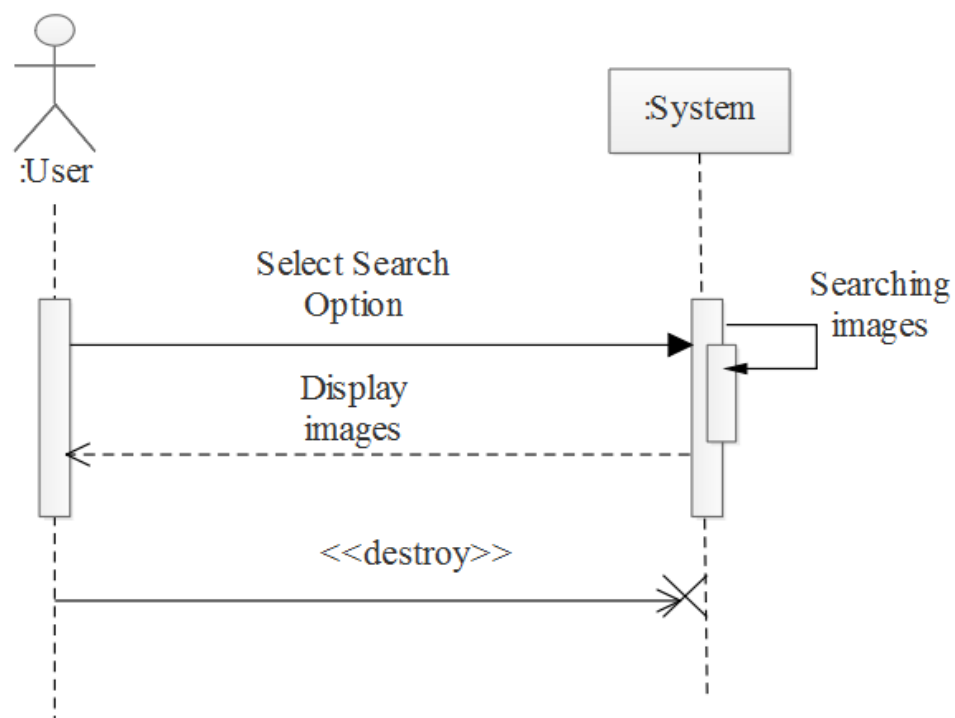


Figure 3.18 Display matched images Sequence Diagram

3.3.7 Give Sample Image Sequence Diagram

User will have to provide sample image to system if user selects “Search images from video” or “Search images from Image gallery” from searching option. System will convert input sample image provided by user into grey scale level that is called pre-processing, after providing the sample image the system detect the face from sample image and then save the face feature in Random Access Memory. These face features will be used later for searching purpose. Without providing sample image system don’t work. All the procedure is shown in figure 3.19.

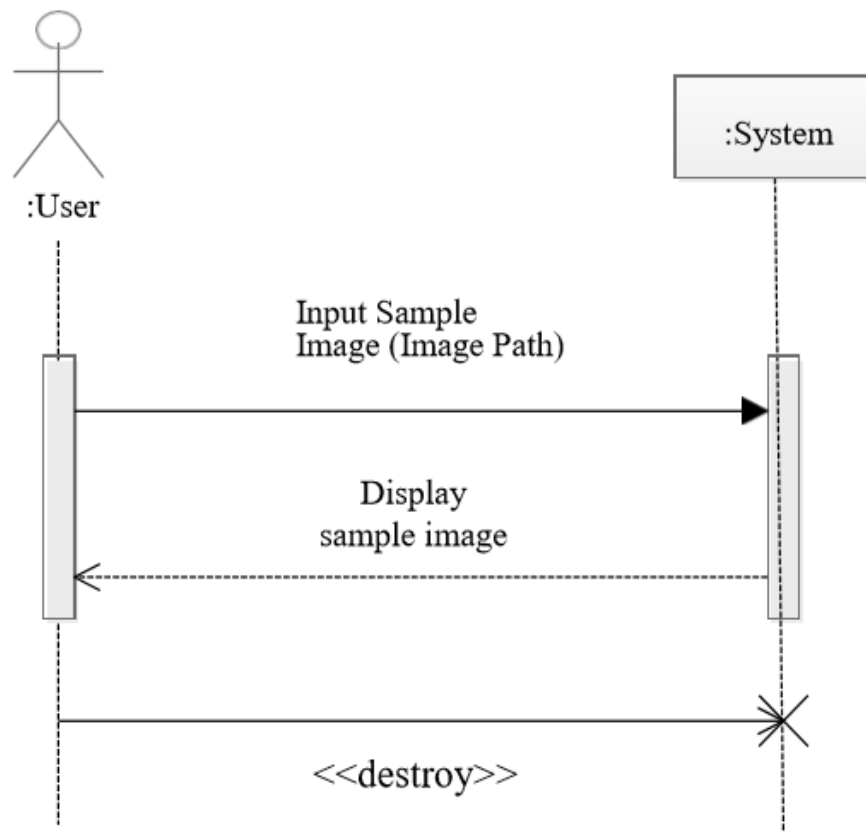


Figure 3.19 Give sample image Sequence Diagram

3.3.8 Search Images from Video Sequence Diagram

The system can also provide option “Search images from video”. User first provide directory from which video is being selected. User will also provide sample image to system. After selecting video, system start converting video into frames. There are four extraction methods to choose from, extract an image every number of frames, extract an image every number of seconds, take a total number of frames from the video or extract every single frame. After conversion system will perform pre-processing on the selected video. Pre-processing is a procedure of converting video frames into grey scale level. From these frames, face is being detected by haar cascade theorem and later these detected face will be store in database. System will then match the face feature of sample image with each face features store in database. Finally images will be saved in RAM which will match with face features in database. The images can be saved in all popular formats, such as JPG, GIF, BMP, PNG, etc. After whole procedure will done, resultant images will be displayed to user. In figure 3.20 whole procedure is shown.

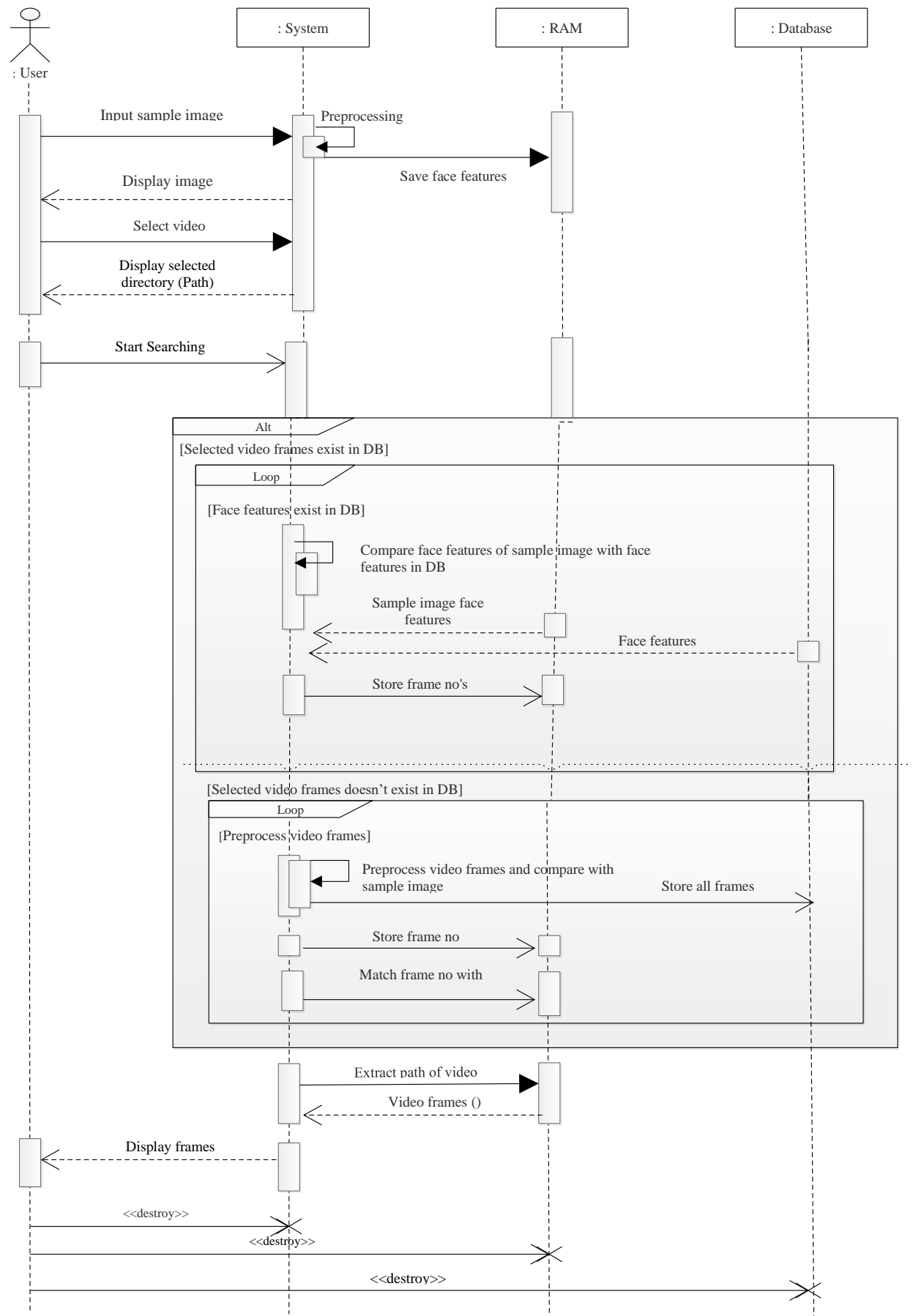


Figure 3.20 Search image from Video Sequence Diagram

3.3.9 Select Video Sequence Diagram

User select video from which user want to search specific images. In figure 3.21 whole procedure is shown.

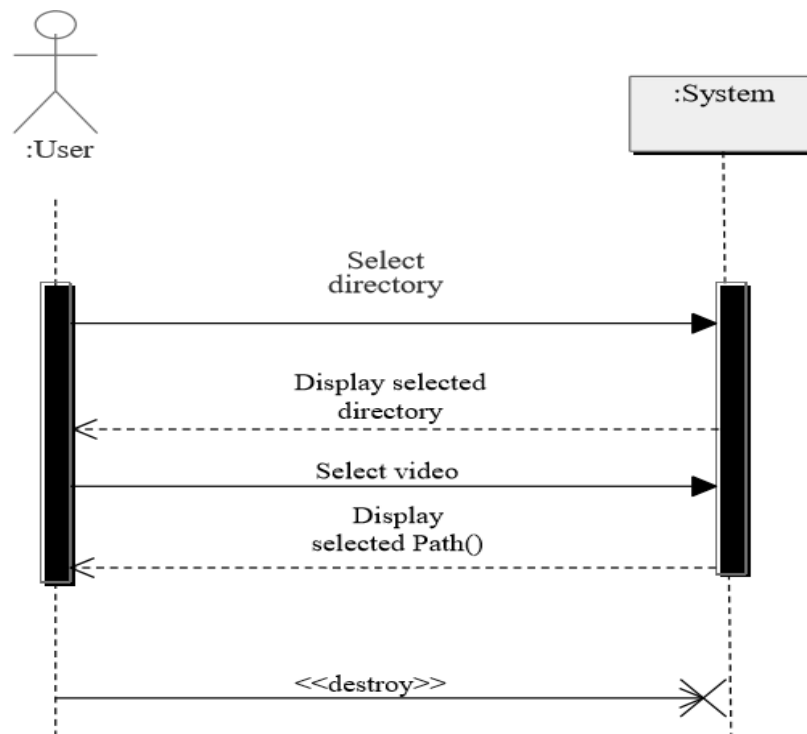


Figure 3.21 Select video Sequence Diagram

3.3.10 Provide Directory for Search Sequence Diagram

User provide directory from which user want to search specific images after selecting one of the search option. Procedure is shown in figure 3.22.

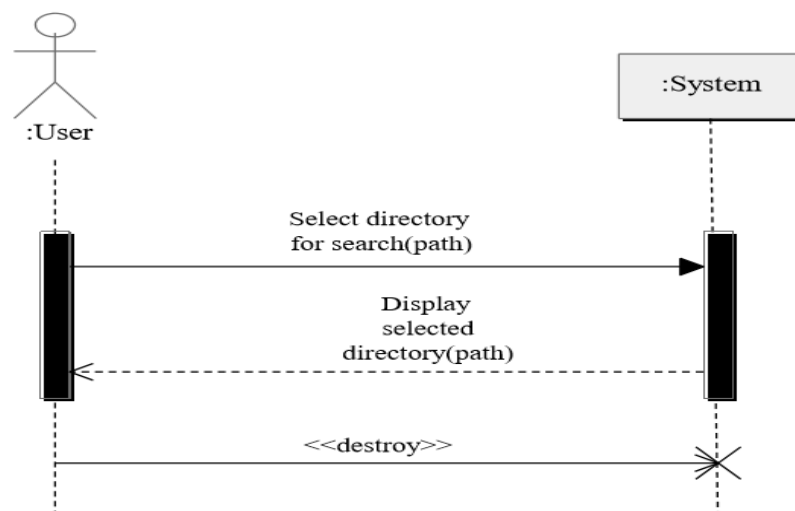


Figure 3.22 Provide directory for search Sequence Diagram

3.3.11 Search Images from Gallery Sequence Diagram

User will provide sample image to system and also select directory. System will perform pre-processing on images and save face features. After this system will match sample image with each face feature save in database and return matched images to user.

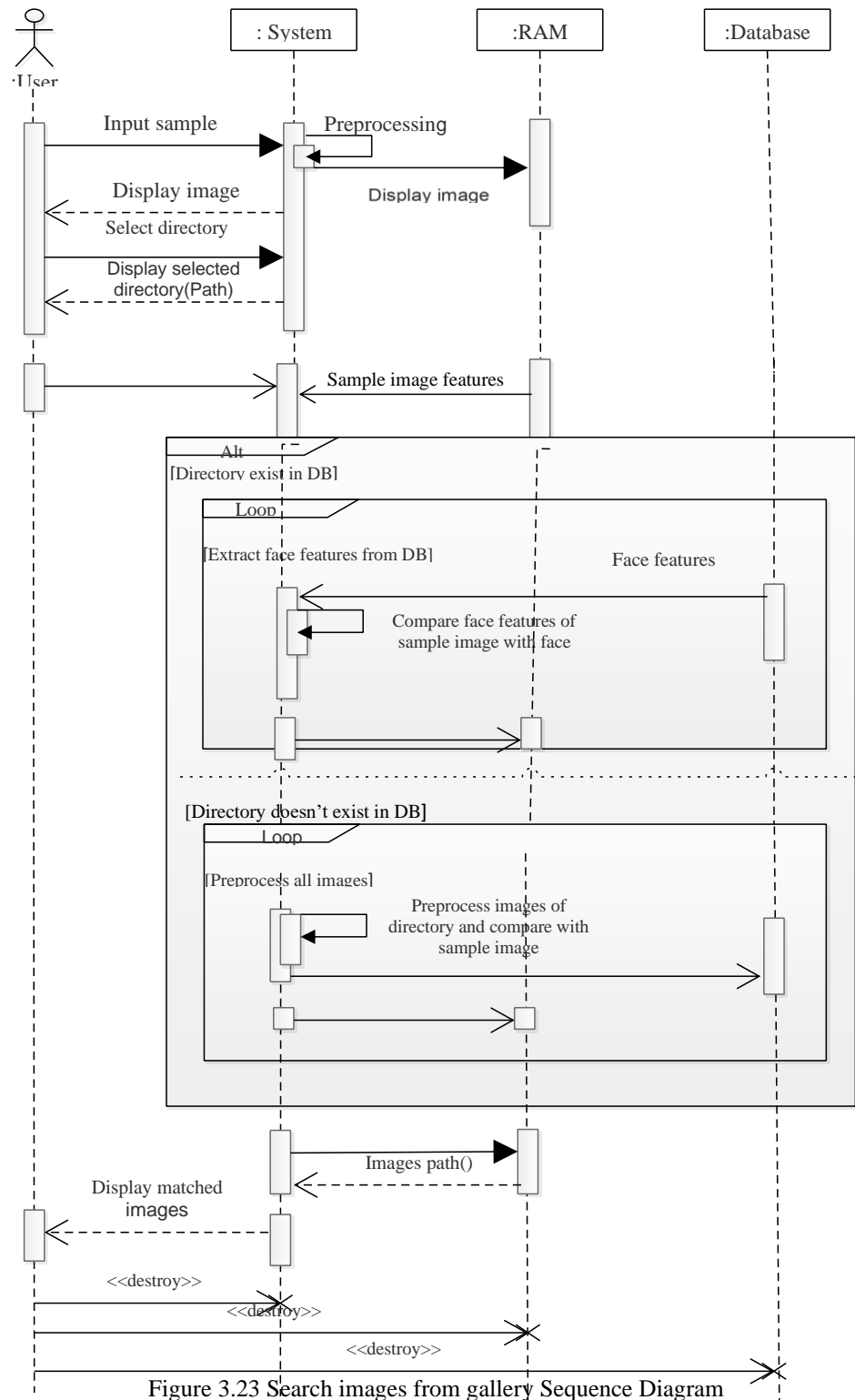


Figure 3.23 Search images from gallery Sequence Diagram

3.4 System Sequence Diagram

The UML system sequence diagram (SSD) illustrates events sequentially input from an external source to the system. The SSD will define the system events and operations. System sequence diagrams are a timeline drawing of an expanded use case. Events are related by time with the top events occurring first. System events are the important items. These are events that cause a system to respond [11].

All systems are treated as a black box; the diagram places emphasis on events that cross the system boundary from actors to systems. A system sequence diagram should be done for the main scenario of the use case, and frequent or complex alternative scenarios [12]. A whole procedure is shown in figure 3.24 and 3.25.

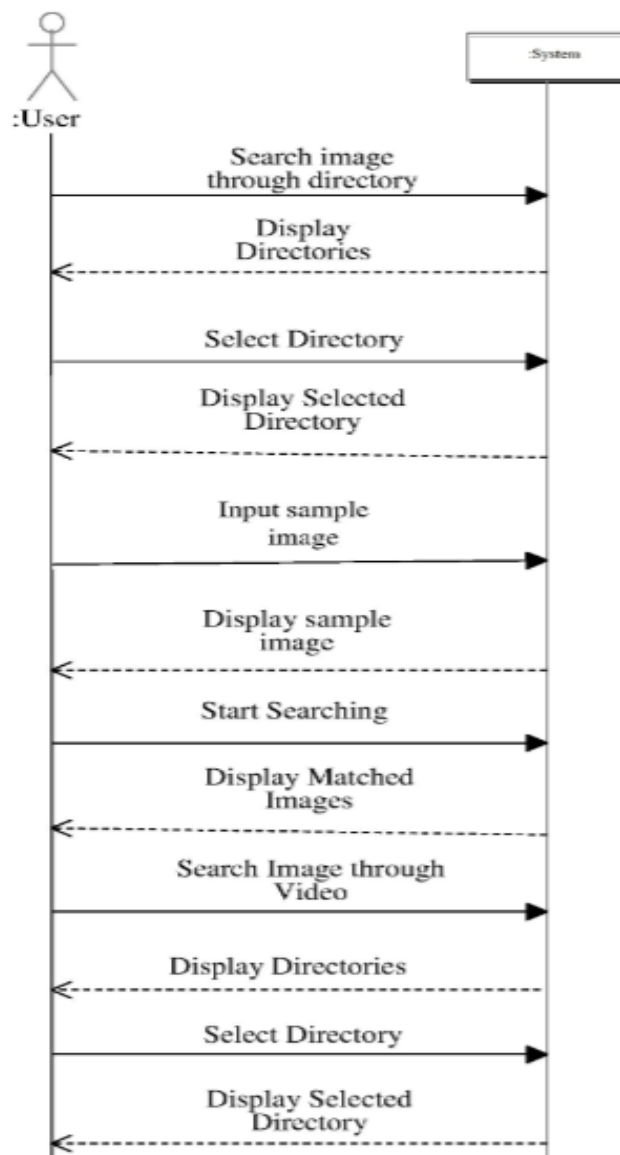


Figure 3.24 System Sequence Diagram

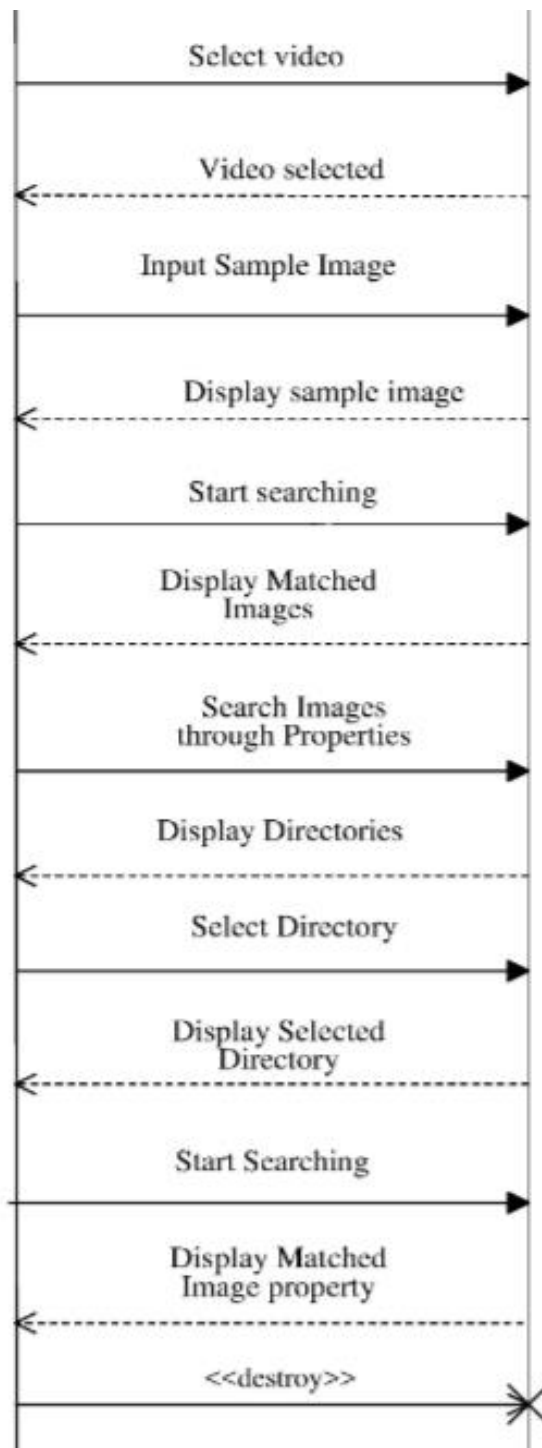


Figure 3.25 System Sequence Diagram

CHAPTER 4

DEVELOPMENT METHODOLOGY

4.0 Introduction

The chapter four, development methodology provides the brief documentation on the development and implementation of the application, Smart Image Searching. The application allow the users to search specific images from video, image gallery and through properties. The chapter also describe the complete working and the whole process of the Smart Image Searching which is a Window based application which could help any user to search image easily.

4.1 Development Track

All the modules of the project are created efficiently and integrated with utmost care. The requirements of all the project areas are summarily evaluated. By keeping in mind the analysis of the requirement gathered from all project areas, modules are created with the help of user interaction with the system. For the development of these modules, Visual studio and C# language are used. After the completion of the development phase, the integration of all the modules including search images from image gallery, search images from video, search image through properties are done. At the end testing is performed so that all the functionalities of the application can be used with ease and without any errors. The application is implemented with the help of programming languages such as C#. For the front-end development TelerikUI design language is used. Due to its user friendliness nature, the implementation of the application become easy. All the front-end components such as buttons, picture box, image box etc. are developed using the TelerikUI design language. All the back-end programming is handled by C# programming language. C# is a very efficient programming language for event and database handling. The development of the application can be achieved with the help of different tools. Two of these tools which we considered for our project are Visual Studio 2013 and SQL Server Management Studio 2012. Visual Studio provides rich development tools to all developers globally on any platform. Development teams will be able to develop software for Web, mobile, server and desktop with Visual Studio.

In the first phase of development all the modules are developed separately. Each module is given a complete focus individually in Visual Studio. The testing is also performed on all these modules separately. The modules like Search image from Image gallery, Search image from video and Search image through properties all these modules are developed

separately. But in order to make the complete application, all the modules are integrated together so that the system can be accessed from the home screen.

4.2 Application Design

The main engineering frameworks are kept in mind to design an efficient application. User needs are considered to make more effective design. The development of application design is done while considering the basic needs of application itself. For example one of the basic need of application is that it is developed in C# so when the design is developed this need should be kept in mind. Different interfaces of the application change or move among themselves by getting input from the user. In other words to move between the activities, there are components that receive the input from user. So we decided to design application in such a way that it uses tabs, which allow the user to keep going on to the next activities. The application is designed in traditional desktop application structure. Which means user get clear instruction to use the application by choosing help button. The application is supported by windows OS. The application can be easily installed on any windows OS device.

4.3 Application Implementation

When the use cases of all the functionalities are designed, the implementation of the application has been started. It is important to implement all the modules of the application separately. The modules of the application are Search image from image gallery which allow any user to search image by matching input image selected by user with all images present in image gallery, Search image from video allow users to extract specific frames and also save those frames in the form of image, Search image through properties which help users to search image by input specific property of image i.e. Search by date or time, Search by name and search by size. Different modules are used to integrate the system and moves between the activities.

4.3.1 Search Images from Image Gallery

When user start the application, the user is shown the three options i.e. Search images from image gallery, Search images from video and Search image through properties. In order to search images from gallery user must select search image from image gallery option. After selecting the option user select sample image and also select directory from which user

want to search specific images. System perform pre-processing on sample image and check either the face-features of selected directory are present in database or not. If the face-features of selected directory present in Database then system just extract images from database one by one and match it with the sample image and return those images that are matched. If the face-features of selected directory are not present in database then the system first perform pre-processing on images and save face-features in database then the system extract image from database and match with sample image and return those images that are matched.

4.3.2 Search Images from Video

Smart Image Searching application also provide another searching method that is “Search images from video”. User first provide sample image and then select video. After that system will start converting video into frames. If the face features of that selected video are already present in database then it simply match the face features of frames with provided sample image face features. And finally display frames to user. If the video frames are not stored in database, it convert video into frames, perform pre-processing on that frames and will store all face features of frames in database. After it, match all face features one by one with sample image face feature. Frame will be display to user in the form of images.

4.3.3 Search Images through Properties

User can also search images by providing specific properties of images. When user select search image through properties option then the system will display three more options i.e. Search by date and time, Search by name and search by size. User will select any one of these options and provide directory then the system start matching the input property with all images properties present in directory and display those images whose properties are matched.

CHAPTER 5

TESTING, ANALYSIS AND VALIDATION

5.0 Introduction

The process of testing is a way to find and remove bugs from the system and then execute the system in a real world environment. After the implementation of the system there is a greater chance that errors might occur, with testing we can minimize these errors. It is essential that a software application meets its requirements and testing ensure that all the requirements are fulfilled. Testing can be done in various ways, there is not just one way to perform testing several methods are available for example integration testing, white box, black box, unit testing and system testing. Among all these testing types two are most important unit testing and integration testing. When all the modules of the system are in the separate form then the unit testing is performed on each module separately. At the later part of the testing all these modules are integrated so after that integration testing is performed. In order to test whether all the requirements are satisfied or not system testing is performed. Each and every functionality of software program is tested in system testing. The application's response to the user's action is evaluated. The output is analyzed in system testing. The output is generated as a result of the user's input. Each functionality of the system is tested in real world environment. Functionalities are evaluated on the basis of the output generated by the user's input.

In the windows application, smart image searching the internal structure of the system is tested by using white box testing. The external structure does not require any coding skills so black box testing is performed on the external structure. Different stages of the development cycle required different testing methods so in simple word all the testing types like unit, integration, system, white box and black box testing are performed during the development according to their need.

5.1 Test Bed

A test bed is a platform for conducting rigorous, transparent, and replicable testing of scientific theories, computational tools, and new technologies. In order to test an application certain number of software and hardware tools are required. The application's test bed includes all of these hardware and software tools. For the application Smart Image Searching, the test bed which we have used involves laptop or Pc installed windows 7 or higher and more than 4GB extra memory is required. Every functionality of the system is tested for every possible input, in this way the application will generate different responses for different inputs.

5.2 Give Sample Image Test Case

In this test case when user gives the sample image the system convert the image in grey scale image and save face features in RAM. Test results are shown in table 5.1.

Table 5.1 Give sample image Test Case

Test Case ID: TC-1	Use Case Reference UC-01	QA Test Engineer	Name of Personnel
Test Case Version 1.0	Give sample image	Developer	Youshra Ibrahim
Test Date	29 th March 2018		
Objective	Save face features of sample image in RAM for searching.		
Environment	Application must be installed on system and must be in running environment.		
Considerations	User must select search option.		
Pre-requisites	User wants to search images from video or from image gallery.		
Step #	Execution Description	Procedure Result	
1.	System display searching option to user.	System will provide user with searching options. After selecting the searching option user provide sample image and the system perform pre-processing on sample image and save face features in RAM on the	
2.	User select searching option that is Search image through video or Search image through image gallery.		
3.	User provide sample image to the system then the system perform pre-processing and save face features in RAM.		
4.	If user select search image through image gallery then the user provide directory from which user want to search relevant images and if user select search image through		

	video then the user select specific video from which user want to search sample image.	basis of sample image the system perform searching.
Comments: The system provides searching option after selecting the specific searching option user provide sample image to the system.		
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed	

Conclusion

When the user select the searching option then the system ask user to provide sample image on the basis of sample image the system start searching through image gallery or through video.

5.3 Give Directory Test Case

In this test case when user provide directory to the system, then it start searching the specific images from directory. Test results are shown in table 5.1.

Table 5.2 Give Directory Test Case

Test Case ID: TC-02	Use Case Reference UC-02	QA Test Engineer	Name of Personnel
Test Case Version 1.0	Give Directory	Developer	Tayyaba Noor
Test Date	31 st March 2018		
Objective	Search Images from provided directory		
Environment	Application must be installed on system and must be in running environment. User must select any search option.		
Considerations	User must select search option along with directory.		

Pre-requisites	User must provide sample image to system	
Step #	Execution Description	Procedure Result
1.	Users first select the Search option from main menu.	System will return those images that are searched from directory provided by the user.
2.	Then the user provide sample image if the selected options are “Search images from gallery” or “Search images from video”.	
3.	After that user provide directory from which images are being searched.	
Comments:		
The application provide searched images that are matched with sample image to user.		
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed	

Conclusion:

After successful running of application, system let the user to select sample image and then directory. On the basis of sample image, system search image from directory and return matched images to user.

5.4 Search Images from Image Gallery Test Case

In the test case of Search images from image gallery, the users can search images from any directory. User provide directory as well as sample image to system. This test case checks that on the basis of sample image that user provide, either all matched images are successfully return to the user or not. Test results are shown in table 5.1.

Table 5.3 Search images from image gallery Test Case

Test Case ID: TC-03	Use Case Reference UC-03	QA Test Engineer	Name of Personnel
--------------------------------------	---	-------------------------	--------------------------

Test Case Version 1.0	Search images from directory	Developer	Tayyaba Noor
Test Date	31 st March 2018		
Objective	Provide matched images to user.		
Environment	Application must be installed on system and must be in running environment.		
Considerations	User must select the option of “Search images from image gallery”.		
Pre-requisites	User must select searching option i.e. “search images from image gallery”.		
Step #	Execution Description	Procedure Result	
1.	Users first select the Search image option from main menu.	System will matched images from directory by comparing with sample image and displayed to user.	
2.	Then the users have to choose a searching method that is search images from image gallery or search images from video.		
3.	It is required from user to select sample image along with directory.		
4.	After that, system start pre-processing and then save the face features of images to database.		
5.	Finally system start search images by matching the face features of sample image with face feature that are in directory and return matched images.		
Comments:			

The application display matched images to user.	
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed

Conclusion

On the basis of selected directory and sample image, system will display images to user after matching procedure.

5.5 Search Images through Properties Test Case

In this test case of search images through properties, user don't need to provide sample image. The search is based upon date, time, name and size. User can search any images from any directory. Application return images to user based upon selected property. So the test case checks either the system return images successfully to user or not.

Table 5.4 Search images through properties Test Case

Test Case ID: TC-04	Use Case Reference UC-04	QA Test Engineer	Name of Personnel
Test Case Version 1.0	Search images through properties	Developer	Tayyaba Noor
Test Date	31 st March 2018		
Objective	To return images to user based upon selected property.		
Environment	Application must be installed on system and must be in running environment.		
Considerations	User must select an option of Search images through properties.		
Pre-requisites	User must select directory for searching images by specific property.		
Step #	Execution Description	Procedure Result	

1.	Users first select the option of Search images through properties.	
2.	Then the users have to choice a Date, Time, Name or Size (Mb's) along with directory from which user wants to search.	System will return matched images to user in less time.
3	System search images from directory based upon selected property.	
Comments: The application will return matched images to user after choosing any property.		
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed	

Conclusion

The system will return matched images to user on the basis of selected property. System provide flexibility to search images from any image property like Date, Time, Name and Size.

5.6 Search Images from Video Test Case

In this test case of search images from video, user will provide sample image & video from directory. System will convert video into frames and perform pre-processing on video and save the face features of frames into database. And then match the frames with sample image and return to user. The test case checks either the application return the image from video to user successfully. Test results are shown in table 5.1.

Table 5.5 Search images from video Test Case

Test Case ID: TC-05	Use Case Reference UC-05	QA Test Engineer	Name of Personnel
--------------------------------------	---	-------------------------	--------------------------

Test Case Version 1.0	Search images from video	Developer	Tayyaba Noor
Test Date	31 st March 2018		
Objective	To return the matched images from video to user.		
Environment	Application must be installed on system and must be in running environment.		
Considerations	User must select the option search images from video along and then select sample image.		
Pre-requisites	User must provide video along with sample image to system for searching purpose.		
Step #	Execution Description	Procedure Result	
1.	User selects option “search images from video” and then provide sample image and video to system.	System will return matched frames to user.	
2.	System will start converting video into frames and perform pre-processing on them, save face features in Database.		
3.	After that system will compare face features of sample image with each face features of frames in Database, return matched frames to user,		
Comments:			
The application return set of matched frames in a limited amount of time with accuracy.			
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed		

Conclusion

As the user provide video, system will start searching images from video after converting into frames and displayed matched frames to user. This helps in security.

5.7 View Matched Images Test Case

In the test case of view matched images, user can select any option that are provided. On the basis of selected method, system will start searching from directory and return images to user. This test case checks that either all images will be viewed to user that are matched with sample image. Test results are shown in table 5.1.

Table 5.6 View matched images Test Case

Test Case ID: TC-06	Use Case Reference UC-06	QA Test Engineer	Name of Personnel
Test Case Version 1.0	View matched images	Developer	Tayyaba Noor
Test Date	31 st March 2018		
Objective	To view all matched images to user.		
Environment	Application must be installed on system. User must have selected the search method.		
Considerations	User must select searching method and also provide sample image along with directory.		
Pre-requisites	User must select searching method in order to view matched images.		
Step #	Execution Description	Procedure Result	
1.	User should install an application and chose searching method. System will perform searching and return matched images which will be viewable to user.	System will return images to user that can be viewed to user.	

Comments:	
The application return images after searching by using specific method.	
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed

Conclusion

When user will select searching method, system start searching images from directory and displayed to user.

5.8 View Matched Frames Test Case

In the test case of view matched frames, user can select Search image from video option. After selecting the searching method user provide sample image. System convert video into frames and start matching each frame with sample image and return those frames that are matched. At the end user can view all those matched frame.

Table 5.7 View matched frames Test Case

Test Case ID: TC-07	Use Case Reference UC-07	QA Test Engineer	Name of Personnel
Test Case Version 1.0	View matched frames.	Developer	Youshra Ibrahim
Test Date	1 st April 2018		
Objective	To view all matched frames to user.		
Environment	Application must be installed on system. User must have selected the search image from video option.		
Considerations	User must select search image from video option. Provide sample image and also select specific video.		
Pre-requisites	User must select video and also provide sample image for searching.		

Step #	Execution Description	Procedure Result
1.	User should install an application and chose searching method “Search image from video”. System will perform searching and return matched frames which will be viewable to user.	System will return frames to user that can be viewed to user.
Comments: The application return frames that are matched with sample image.		
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed	

Conclusion

When user will selects search image from video, system start searching sample image from video and displayed to user.

5.9 Search images by Date or Time Test Case

In this test case user select search image through properties option. User provide specific date or time and also provide directory from which user want to search images. Test results are shown in table 5.1.

Table 5.8 Search images by date or time Test Case

Test Case ID : TC-08	Use Case Reference UC-08	QA Test Engineer	Name of Personnel
Test Case Version 1.0	Search image by date or time	Developer	Youshra Ibrahim
Test Date	1 st April 2018		
Objective	To return those images that matched with input date or time.		

Environment	Application must be installed on system and must be in running environment.	
Considerations	User must select an option of Search images through properties.	
Pre-requisites	User must provide image property like date, time, size and name to system.	
Step #	Execution Description	Procedure Result
1.	Users first select the option of Search images through properties.	System will return images that matched with input date or time to user.
2.	Then the users have to provide a date or time along with directory from which user wants to search.	
3	System search images from directory based upon specific date or time.	
Comments: The application will return matched images to user after providing specific date or time.		
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed	

Conclusion

The system will return matched images to user on the basis of input date or time.

5.10 Search Images by Size Test Case

In this test case of search images by size user first select searching option that is search image through properties. After selecting the option user input specific size of image and provide directory from which user want to search image of input size. Test results are shown in table 5.1.

Table 5.9 Search images by size Test Case

Test Case ID : TC-09	Use Case Reference UC-09	QA Test Engineer	Name of Personnel
Test Case Version 1.0	Search image by size.	Developer	Youshra Ibrahim
Test Date	1 st April 2018		
Objective	To return images to user that matched with input size.		
Environment	Application must be installed on system and must be in running environment.		
Considerations	User must select an option of Search images through properties.		
Pre-requisites	User wants to search images by size so must select directory along with size property.		
Step #	Execution Description	Procedure Result	
1.	Users first select the option of Search images through properties.	System will return matched images to user.	
2.	Then the user input Size (Mb's) along with directory from which user wants to search.		
3	System search images from directory based upon input size.		
Comments: The application will return matched images to user after providing specific size of image.			
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed		

Conclusion

The system will return those images from directory that matched with input size.

5.11 Search Images by Name Test Case

In this test case of search images by name user first select searching option that is search image through properties. After selecting the option user input specific name of image and provide directory from which user want to search image of input name.

Table 5.10 Search images by name Test Case

Test Case ID: TC-10	Use Case Reference UC-10	QA Test Engineer	Name of Personnel
Test Case Version 1.0	Search image by name.	Developer	Youshra Ibrahim
Test Date	1 st April 2018		
Objective	To return those images from directory that matched with input name.		
Environment	Application must be installed on system and must be in running environment.		
Considerations	User must select an option of Search images through properties.		
Pre-requisites	User must select directory along with name property.		
Step #	Execution Description	Procedure Result	
1.	Users first select the option of Search images through properties.	System will return those images to user that are matched with input name.	
2.	Then the user provide specific name and directory from which user want to search image.		

3.	System search images from directory based upon input size.	
Comments: The application will return matched images to user after providing specific name.		
Status	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail <input type="checkbox"/> Not Executed	

Conclusion

The system will return matched images to user on the basis of input name.

5.12 Testing Results

Software testing helps in determining the quality of the system. The quality of the system can be analyzed by observing the results produced by the system in the real environment objectives and situations. After performing testing on our system we have observed that all the modules are working fine. So the testing phase has covered all the functionalities of the system.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.0 Introduction

The complete overview of the system is outline in this chapter. The future work and the development methodologies are also discussed in this chapter. During the development of the application certain aims and objectives are also achieved this chapter also describe those goals and objectives. Like every system this application also has few limitations as well. The limitation of the application are also discussed in this chapter. We have also faced certain challenges during the development of the application so these challenges are also summarized.

6.1 System Overview and Methodologies

Agile development methodology is used throughout the development of our application, Smart Image Searching. In the agile development methodology, a software system is strongly clarified and delivered through the repeated feedback and iterative structure of the development. In the agile method the project is continuously tested, integrated and planned. Due to this continuous iteration and feedback the system becomes refined and error free. Smart Image Searching provides facility to users to search images through three different ways that save time and increase efficiency.

The application allows the user to search images from desktop, Laptops etc. It provides three different searching methods “Search images from image gallery”, Search images from video” and Search images through properties. All methods have different way to search. User first open an application, select “search image” option. Application then provides three different searching method. User will select one option and after choosing option input sample image and directory will be provided for efficient searching. System will start converting image into grey scale and store the face feature in RAM. And then it will check either the directory images are already stored in RAM. If images were stored in database, then system will match images from those images in database. If no then covert directory’s images into grey scale and will store face features in Database. Finally display the matched images to user. By choosing option “Search images through properties” user will provide video clip. System then convert video into frames. Again like in searching images from gallery system check either frames are stored in Database. If the frames are stored then system will return matched frame otherwise convert video into frames, perform pre-processing and then store the frames in Database. Finally images will be displayed to user.

Searching images through properties, user only have to provide image properties (Date, Time, Size and Name). On the basis of selected option, system will search images from selected directory and then display to user.

6.2 Limitations

The first limitation of the system is that it only run on windows devices like Computers, laptops. Operating systems such as Android OS, Mac and Linux and also windows less than 7 cannot support this application. Another limitation of this system is that to run this extra 4GB Memory is required.

6.3 Challenges Involved with the Implementation

The implementation of our application, Smart Image Searching involves certain amount of challenges. The challenges that we face during the implementation are Face Detection, Face Recognition, Database connectivity, Deployment etc.

6.3.1 Face Detection

The application can only run in windows environment. This means that the application can only be developed in the visual Studio. The first challenge that our team faced is face detection. If we provide image having two faces, it will be difficult for system to detect face separately. It give same name to second person. However we have overcome this challenge.

6.3.2 Face Recognition

For searching, user provide sample image. On the basis of sample image, system will provide matched images. So recognition is involved that is challenged faced by our team members.

6.4 Future Work

Smart Image Searching is developed efficiently. The effective ways of development are used for the development of the application. During the development, all the functional requirements has been fulfilled. In the future, the next versions of this applications can facilitate the users with new features and functionalities. For example our existing application provides “search images from image gallery”, “Search images from video” and

“Search images through properties” but the future version also involves search images from live CCTV camera, it will be a web based application and also facilitate the sharing of images to all social media.

APPENDICES

Appendix I–Installation Guide

To install this application user need to have laptop or Pc along with windows 7 or higher and more than 4GB extra memory.

User can install application through CD.

The installation process took few seconds to install. After installation process is completed, user can use application in appropriate way.



Figure A-1.1 Search Images by Date

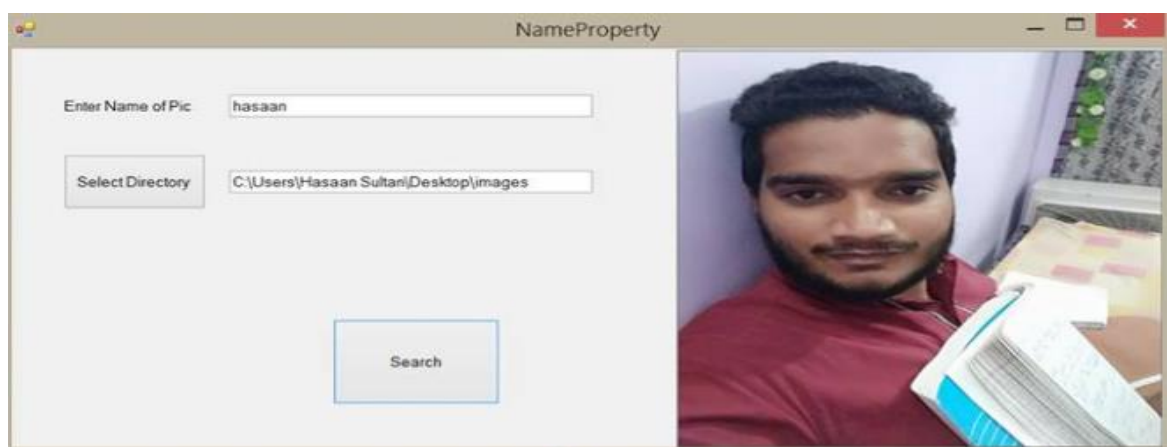


Figure A-1.2 Search Images by Name

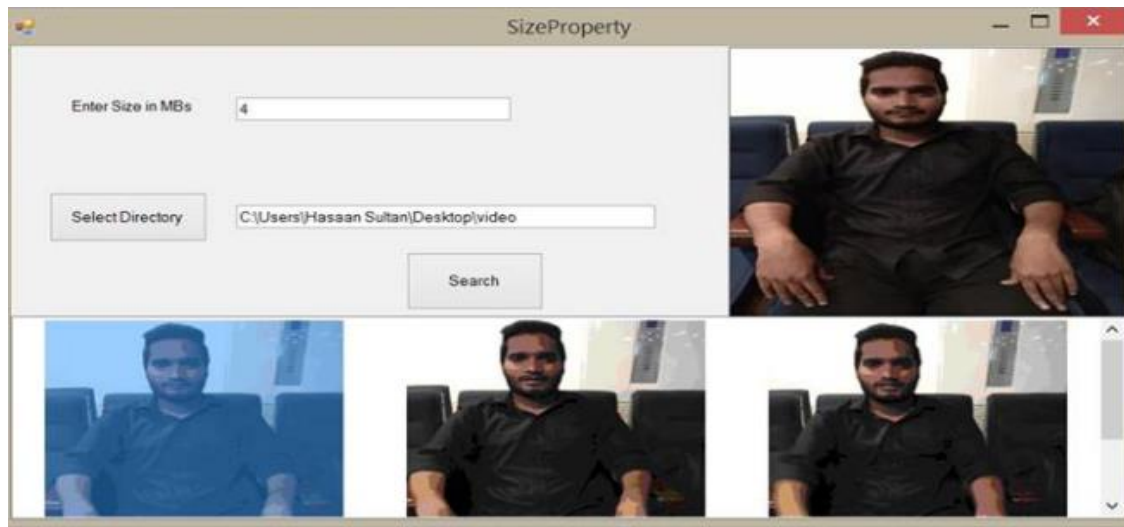


Figure A-1.3 Search Images by Size

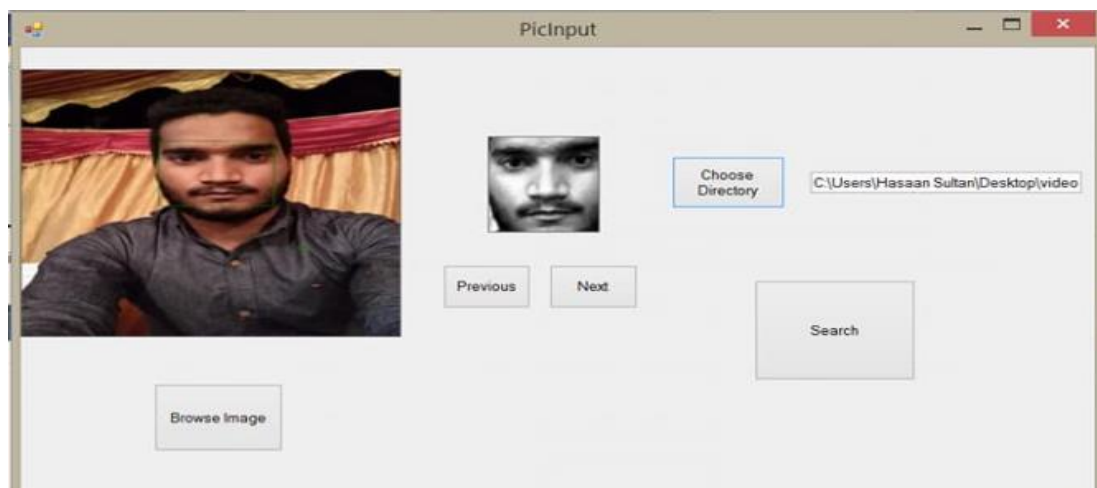


Figure A-1.4 Search Image by Name

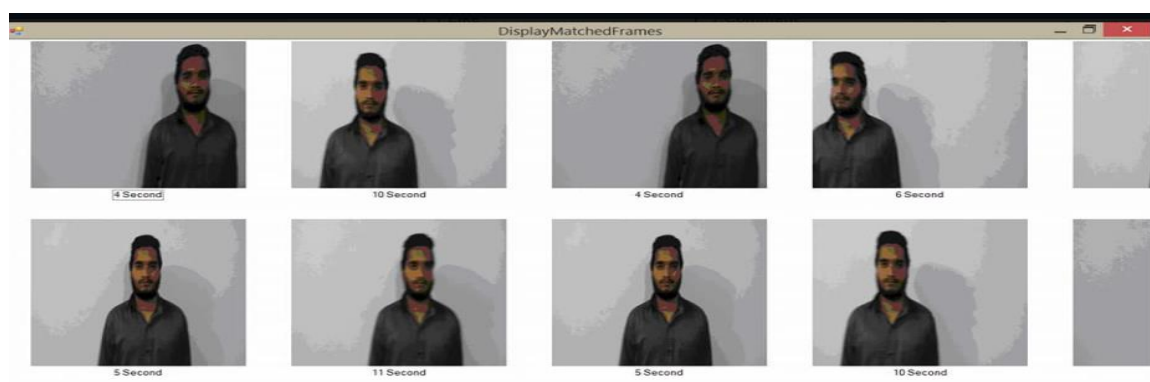


Figure A-1.5 Search Images From Video

REFERENCES

1. “Microsoft Visual Studio,” [Online]. Available: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio. [Accessed 10 february 2018].
2. “Tutorial: SQL Server Management Studio (SSMS),” [Online]. Available: <https://docs.microsoft.com/en-us/sql/ssms/tutorials/tutorial-sql-server-management-studio>. [Accessed 10 February 2018].
3. “Main Page,” [Online]. Available: http://www.emgu.com/wiki/index.php/Main_Page. [Accessed 10 february 2018].
4. [Online]. Available: <https://www.edrawsoft.com/flowchart.php>. [Accessed 10 february 2018].
5. R. Teucher, “Compfight / A Flickr Search Tool,” [Online]. [Accessed february 2018].
6. J. Jones, “Find Free Images Quickly With Compfight – ProfHacker - Blogs - The Chronicle of Higher Education,” [Online]. Available: <https://www.chronicle.com/blogs/profhacker/find-free-images-quickly-with-compfight/41025>. [Accessed 9 february 2018].
7. “Brain Picking,” [Online]. Available: <https://www.brainpickings.org/2010/10/01/best-image-search-tools/>. [Accessed 25 September 2017].
8. “Flowchart,” [Online]. Available: <https://www.smartdraw.com/flowchart/>. [Accessed 10 February 2018].
9. [Online]. Available: <http://searchsoftwarequality.techtarget.com/definition/use-case>. [Accessed 10 February 2018].
10. [Online]. Available: <https://www1.in.tum.de/paid.globalse.org/PAID/Misc-Documents/TogetherJ/UserGuide/Chapter4.html>. [Accessed 11 February 2018].

11. "UML System Sequence Diagram," [Online]. Available:
<http://www.comptechdoc.org/independent/uml/begin/umlsyssequence.html>.
[Accessed 19 June 2018].
12. "System Sequence Diagram," 18 May 2018. [Online]. Available:
https://en.wikipedia.org/wiki/System_sequence_diagram. [Accessed 19 May 2018].