ADMAS UNIVERSITY

SCHOOL OF POST GRADUATED

DEPARTMENT OF COMPUTER SCIENCE

OBJECT ORIENTED SOFTWARE DEVELOPMENT

DOCUMENTATION

FOR

CRIMINAL FACE IDENTIFICATION SYSTEM

Nam of student	Id No.
1. Andualem Biru Tadesse	PGMGC/8026/20
2. Betel Oluma Taresa	PGMGC/8068/20

Submited to Abdi Mulat (Ass.pro)

March 14, 2021

Abstract

Criminal record generally contains personal information about particular person along with photograph. To identify any Criminal, we need some identification regarding person, which are given by eyewitness. In most cases the quality and resolution of the recorded image segments is poor and hard to identify a face. To overcome this sort of problem we are doing project documentation. Identification can be done in many ways like finger print, eyes, DNA etc. One of the applications is face identification. The face is our primary focus of attention in social inters course playing a major role in conveying identify and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize face is remarkable.

Table of Contents

List Of Table

List Of Figure

Abbreviations and Definition

- CFIS: Criminal Face Identification System
- RAD: Requirement Analysis document
- FPCIB: Federal Police Commission Investigation Bureau
- UML: Unified Modeling Language UML
- DSDM: Dynamic Systems Development technique DSDM
- CPU: Central Processing Unit
- GUI: Graphical user interface
- ACL: Access control list
- CFIS: Criminal face identification system
- I/O: Input out put
- **Invariant:** constraints associated with classes
- **Precondition**: specify constraints that a caller must meet before calling an operation; are associated with a specific operation
- **Post-condition**: specify constraints that the object must ensure after the invocation of the operation.

CHAPTER ONE

REQUIREMENT ANALYSIS

1 Introduction

1.1 Background

Over the years, a lot of security approaches have been developed that help in keeping confidential data secured and limiting the chances of a security breach. Face recognition which is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness is a computer program that uses a person's face to automatically identify and verify the person from a digital image or a video frame from a video source. It compares selected facial features from the image and a face database or it can also be a hardware which used to authenticate a person [1].

This technology is a widely used biometrics system for authentication, authorization, verification and identification. In developed countries, the law enforcement creates face database to be used with their face recognition system to compare any suspect with the database. In other hand, in Ethiopia, most cases are investigated by using thumbprint identification to identify any suspect for the case. However, because of unlimited knowledge through internet usage, most criminals are aware of thumbprint identification. Therefore, they become more cautious of leaving thumbprint by wearing gloves except for non-premeditated crimes. This paper to propose a facial recognition system for a criminal database where the identification of the suspect is done by face matched rather than thumbprint matched. Face recognition is the process of identification faces from a photo or video of a person and it works by taking the photo from surveillance road cameras and compare this photo frame by frame from a set of faces in the database.

This face biometric has better comparison result comparing to other modes of biometrics techniques. This helps to detect faces from a given image/video frame through the camera and the way of mapping eyes, noses, mouth, ears, and etc. Besides this, the project used cascade classifier in the implementation of the system as a face detection approach for criminal investigations [3].

This project is aimed to identify the criminal faces. In here the technique which going to use is, manually we already store some images of the criminals in our database along with his details. By using surveillance road camera system residing at some public place or roads which automatically matches the input faces with criminal database and gives alert if the results are matched. If any image is matched up to 95% or closer to that rate then we predict that he is only the criminal.

1.2 Statement of the Problem

The problem with the present system is same as problems encountered in any manual file processing system. The existing system does not support the cropped images of criminals.

The existing system is not suitable in some cases such as if a witness can identify only a part of the criminal. Present system uses some algorithms for identifying criminal faces which are difficult to process. Finally, the existing system does not always produce better results in identifying the criminals by their images.

Take a face image of a person as an input. And compare the face image of a person with the existing face images that are already saved in the database. So, face detection is the main problem here. When it come to the real world, it might not possible to capture a full-frontal picture of a face at all the times in uncontrolled environments. Even though there are many face recognition applications available, most of these applications work in optimal condition.

Detecting a face in probe image may be relatively simple task for a human. But it's not for a computer. The computer has to decide which pixels in the image is part of the face and which are not. In a typical passport photo, where the background is clear, it's easy to do, but when the background becomes cluttered with other objects, the problem becomes extremely complex.

As mentioned in the proposal there are many challenges specially, uncontrolled background and lighting, unrestricted range of facial expression and typical variations in hair style, criminal wear specs or wig and difficult to identify facial marks, aging, sketches and other adornments. These variations have to handle by any face detection method which

hopes to operate in an uncontrolled environment. By understanding the above problems, there are several sub steps need to be performed within the steps. Even there are lot of approaches for design CFIS, the proposed solution deviates from the regular face recognition approaches.

1.3 Objective

1.3.1 General Objective

The general objective of this project is to develop criminal face identification system.

1.3.2 Specific Objective

The specific objectives of the project are:

- ✓ To gathering requirements analysis used for face detection system.
- ✓ Creating face detection system
- ✓ Test the system.
- ✓ Implement the system.
- ✓ Improve capabilities of the detecting features of the local segmentations of face.
- ✓ Through requirements being carried on face recognition techniques and available algorithm on partial face recognition and choose an appropriate method and implement face recognition system based on it. Finally, as there are many approaches to these contents, selection of the most suitable techniques has to be determined after evaluating available techniques under various factors. The methods and approach we used are as the followings:

1.3.3 Face detection approaches

Face detection is the computer vision technology and that detects or determines the location and the size of a face in image that human can do effortlessly. Anyhow when comparing with the computer vision terms, this task is not easy. A face detection system should be able to achieve the regardless of illumination, orientation or camera distance. And also, there are general problems in face detection system such as extraction and verification of faces, possibly identify only faces or face features from an uncontrolled background.

Detection will happen based on the following techniques: -

Controlled environment

It's the important and straightforward case. Normally photographers are taken under controlled light, background etc., so in that case can use simple edge detection techniques to detect faces.

Color images

To detect faces, typically skin color can be used to detect faces. And the result might be change or weak if the light conditions change. Human skin color can be change from nearly white to almost black. The major difference lies between their intensity, so chrominance is a good feature. It's not easy to establish a solid human skin color representation. However, there are attempts to build robust face detection algorithms based on skin color.

Images in motion

Detect faces in a real time videos are a challenge. There is a continuing challenge to achieve the best detecting results with the best possible performance. Another approach based on motion is eye blink detection, which has many uses aside from face detection.

Face detection difficulties

It's very hard process to build automatic robust face detection. Because, even a face detection system detect faces in different image situations and different face condition it seems really easy to do this with the human visual system. In fact, the object of "face" is hard to define by mathematical algorithms, as its large variability depending on the identity of the person, the lighting conditions and the psychological context of the person etc.

The main challenge for detecting faces to is to find a classifier which can discriminate faces from all other possible images.

- When we consider face as a global attribute, we can extract some common attributes from every face. Normally even we consider face as an ellipse object but there can be thin faces and also rounder faces. And the skin color also different from person to person. So, a system might get confuse to detect faces automatically.

- **The pose of the face:** by the position of a person where he stands in front of the camera might get change the view of the faces.
- **The facial expression:** because of face appearance the results might get change. The face feature of a smiling face can be far from the appearance of a sad face.
- **Presence of added objects:** the objects that we can find usually on a face can be affecting the result of detecting a face. If a person wears a sunglass, that will change one of the main characteristics of a faces. And also, from the natural facial features such as beards and hair style which can occult one part of a face.
- **Image condition:** the appearance of a face might get change to one another due to light conditions so illumination and the intensity have to be considered.

1.3.4 Face detection methods

Today face detection and face feature recognition has been an active research area in the computer vision field. Face detection is the basic and the first step of face recognition. A number of methods and algorithms have been proposed for face detection and these methods can group in to several categories like.

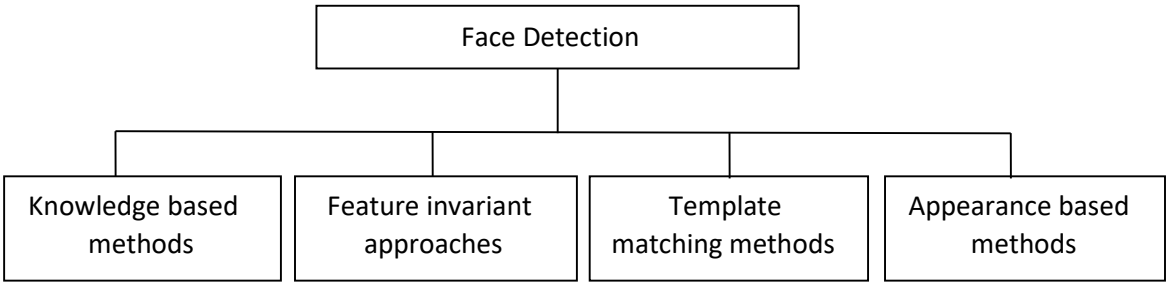


Figure 1: Face detection methods

- Knowledge based methods use predefined rules to determine a face based on human knowledge.
- Feature based approaches main target is to find the features of a face structure.
- Template matching methods use some sample face templates to judge if an image is a face.

- Appearance based methods get an idea of a face models from a set of representative training face images.

In this technical project, we had seen different survey the recent advances in face detection and the Viola-Jones face detector is first re-viewed and in feature invariant approaches usually consider about skin color, texture and facial features. Using a Viola-Jones Classifier to detect faces in a live webcam feed. In this project we are using the viola-jones face detector algorithm.

The Viola-Jones Face Detector

Viola Jones algorithm is named after two computer vision researchers who proposed the method in 2001,

Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features". Despite being an outdated framework, Viola-Jones is quite powerful, and its application has proven to be exceptionally notable in real-time face detection. This algorithm is painfully slow to train but can detect faces in real-time with impressive speed [7].

Face detection proposed by Viola and Jones based on statistic methods is most popular among the face detection approaches. This face detection is a variant of the AdaBoost algorithm which achieves rapid and robust face detection. In AdaBoost algorithm is proposed a face detection method based on AdaBoost algorithm using Haar Features that detected the face successfully with high accuracy. But still we can't satisfy with the accuracy of this method.

The basic principal of the Viola-Jones algorithm is to scan a sub window that capable of detecting faces across the given image. The standard image processing approach would be to rescale the input image to different sizes and then run the fixed size detector through these images. Rather than time consuming due to the calculation of the different size images, this approach is better. But in Viola-Jones has a method of scale invariant detector that has the same number of calculations whatever the size. This detector is constructed using a integral image that can be used to compute simple Harr-like rectangular features.

The propose method to detect face is Viola Jones Haar classifier cascade and it draws a rectangle around faces.

Advantages and disadvantages of Viola-Jones Technique

Advantages

- Viola Jones face detection algorithm is the most admired algorithms for face detection in real time.
- Another main advantage is uncompetitive detection speed.
- When comparing with other face detection algorithms Viola Jones is an especially successful method because it has a low false positive rate.

Disadvantages

- Actually, it takes long training time
- Limited head poses
- Not detect black faces (depend on the photo background)

1.4 Scope of the project

The scope of the project is confined to store the criminal details with image and store in the database. When a person has to be identified the images stored in the database are compared with surveillance road camera system residing at some public place or roads which automatically matches the input faces with criminal database and gives alert if the results are matched. If any image is matched up to 95% or closer to that rate then system predict that he/she is the criminal.

1.5 Current system

Currently they used manual system by store the criminal information and image by hardcopy. When they find criminals firstly checking criminals' image in the existing images, compare with them and posting in several places. Then search the criminal by traveling from one place to another place. This process is very slow to find criminals and give the result and also it takes a time.

The current system is a manual face recognition process is very slow to give the result. It is very critical to find the criminal images. Normally face recognition applications are used to recognize faces based on various factors and technologies. Although today different face recognition systems use different approaches, most of them perform key main steps, which are common to most face recognition approaches. Even the final result of recognize a face, but not all face recognition applications are same when comparing with the process of steps. Some applications might be reliable than others, depending on the quality of the image captured as well as the algorithms employed in the application. There are many techniques used for facial recognition, mainly geometric and photometric. In geometric technology especially, data relies on shapes on the face called node points (distance between the eyes, nose width, cheekbones and depth of eyes etc.). Despite improvements, there is still no system available that can report with one hundred percent accuracy.

1.6 Proposed System:

The system we proposed is a criminal face identification system, investigator now does not have the right system to find criminals in a modern way. What our system does is that it enables investigator to find criminals easily from surveillance road camera by their detecting criminal face. This system will help the investigator to know about what kind of crime. The criminal did in the previously history, ability to catch many criminals within a minute and minimize the time to catch criminals. CFIS will also help the investigator to detecting the criminal.

Addition, Split, Merge and updating of the criminal record and face. Comparing the image with the faces that are there in our database. If any new images are found then it should be entered into our database by add image module and then it should be segmented into different slices.

This project is aimed to identify the criminals in any investigation department. The technique which going to use is, manually we already store some images of the criminals in our database along with his details. By using surveillance road camera system residing at some public place or roads which automatically detected matches the input faces with criminal database and gives alert if the results are matched. If any image is matched up to 95% or closer to that rate then the system predict that he/she is only the criminal. Thus,

using this project, it provides a very friendly environment for investigator to easily design any face can identify criminals very easy.

Advantages of Proposed System:

- Very fast and accurate to detecting and identifying images.
- No need of any extra human power.
- No fever of data loss because of every data save in the system.
- Doesn't require any extra hardware device.
- Very easy to find the criminals.

1.7 Functional Requirements:

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform [9].

The functional requirements of the system are stated below:

- The system should allow investigator to fill full information's about criminal details.
- The system should allow to split the image into numbers of parts.
- The system allow investigator to keep record information about each criminal.
- The system allow investigator to identify the criminal images.
- The system enables information about each criminal.

1.8 Non-functional requirements

Non-functional requirements (also known as quality requirements), are requirements which impose constraints on the design or implementation (such as performance requirements, security, or reliability) [9].

The following are lists of non-functional requirements: -

- **Performance**
  - the system should have response within 30 to 45 seconds and should have reasonable performance during high traffic load.

- **Reliability**
  - the system should perform showing the appropriate error when the user inserts duplicated user id and verifying each form as required.
- **Security**
  - login module will be implemented for user login into the system.
  - the system should not allow unauthorized user to access it.
- **Usability**
  - the system should be easy to use and should support English language.

1.9 System Models

The Unified Modeling Language (UML) is an open method used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to write a system's blueprints, including conceptual components such as.

1.10 Use case Diagram:

Use cases are a way of describing interactions between users(actors) and a system and Use cases focus on the behavior of the system from an external point of view. Use case diagram has designed to understand functional requirements and users of the system [8].

A Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

The use case diagram shows the position or context of the use case among other use cases. As an organizing mechanism, a set of consistent, coherent use cases promotes a useful picture of system behavior, a common understanding between the Administrator and Investigator team.

- **Actors**
  - Are external entities that interact with the system.
  - We have two actors in our system ***Administrator*** and ***Investigator.***
- **Use cases**
  - describe the behavior of the system as seen from an actor's point of view.
- **System boundary**: rectangle diagram representing the boundary between the actors and the system.

Association: communication between an actor and a use case; Represented by a solid line

Include: denote common functionality

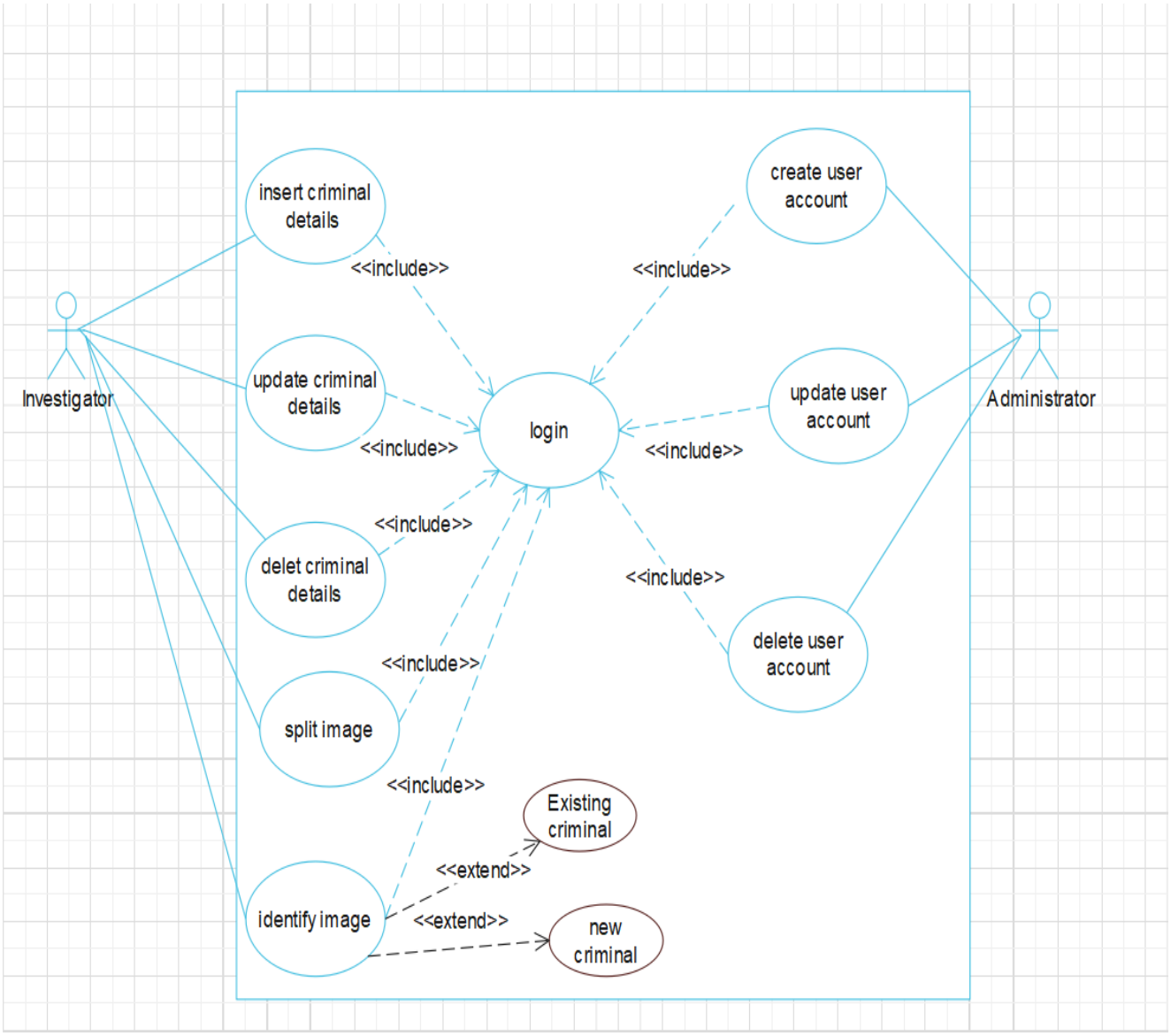


Figure 2: Use-case diagram for CFIS

1.10.1 Use case descriptions

Use case ID	UID1	**Use case Priority**	High
Use case name	inserting criminal details		
Brief description	The investigator fills full information about criminal details correctly.		
Actors	Investigator		
preconditions	login in to the system.		
Flow of Activities	Actor	System	
	1. Log into the system 2. investigator clicks Inserting criminal details 3. investigator will fill the information click	1.1 CFIS homepage will be displayed 2.1. The system will display Inserting criminal details page 3.1. The system will save and	
Exceptional Condition	1a. investigator inserts wrong credential		
post conditions	The criminal information is registered		

Table 1: Use case Description for inserting criminal details

Use case ID	UID2	**Use case Priority**	High
Use case name	updating criminal details		
Scenario	updating the criminal information's		
Brief description	The investigator updating the criminal information with image		
Actors	Investigator		
preconditions	Producer must login in.		
Flow of Activities	Actor	System	
	1. Log into the system display 2. click the button to update criminal details 3. investigator update criminal information click	1.1 CFIS homepage will be 2.1. The system will display the criminal information 3.1. The system will save and	
post conditions	The criminal information is updated		

Table 2: Use case Description for splitting and updating image

Use case ID	UID3	**Use case Priority**	High
Use case name	delete criminal details		
Brief description	The investigator deletes criminal details from criminal data base		
Actors	Investigator		
preconditions	Investigator must login in.		
Flow of Activities	Actor	System	
	1. Log into the system display 2. click the button to view image 3. investigator delete criminal details	1.1 CFIS homepage will be 2.1. The system will display delete criminal details page 3.1 delete successfully	

Table 3: Use case Description for retrieving image

Use case ID	UID4	**Use case Priority**	High
Use case name	Identifies image		
Scenario	View suggested crop through web application		
Triggering event	When investigator want to plant crop		
Brief description	This use case helps identify the criminal, the investigator knows the criminal is an existing criminals or new criminals and also to make decision on which kind of crime he/she involved previously.		
Actors	Investigator		
related use cases	Include: new criminal Extends: existing criminal		
preconditions	Investigator must login in. First fill full information about the criminal		
post conditions	Identify the criminal and she/he recorded data		
Flow of Activities	Actor	System	
	1. Logs into the system 2. Clicks the button	1.1. Displays CFIS identifying image home page 2.1. The system displays full information and image about the criminal. The system analyzes matched the image and also identify its existing or new criminal.	
Exceptional Condition	1a. Producer inserts wrong credential		

Table 4: Use case Description for Identifies image

se case ID	UID5	**Use case Priority**	High
Use case name	Create account		
Scenario	Create user accounts through web application		
Triggering event	The administrator select Create account button		
Brief description	The administrator can create new user accounts as needed.		
Actors	Administrator		
preconditions	Administrator must login in.		
Post conditions	The administrator crate user account successfully		
Flow of Activities	Actor	System	
	1. Log into the system 2. Click the “Create User Account” button 3. The admin fills the form and clicks “Save” button.	1.1. CFIS home page appears 2.1. The system displays crate user account page 3.1. The system will save the data and redirect to account managing page.	
Exceptional Condition	1a. Administrator inserts wrong credential		

Table 5: Use case Description for Create account

Use case ID	UID6	**Use case Priority**	Medium
Use case name	Edit account		
Scenario	Edit user accounts through web application		
Triggering event	The administrator select Edit button		
Brief description	The administrator is able to edit/update users account information.		
Actors	Administrator		
preconditions	Administrator must login in. User accounts must exist.		
Post conditions			
Flow of Activities	Actor	System	
	1. Log into the system 2. Click the button edit account 3. Admin clicks the edit button 4. The admin fills the form and clicks update button.	1.1. CFIS home page appears 2.1. The system displays edit account managing page 3.1. The system displays editing form. 4.1. The system will update the data and redirect to account managing page.	
Exceptional Condition	1a. Administrator inserts wrong credential		

Table 6 :Use case Description for Edit account

Use case ID	UID7	**Use case Priority**	Medium
Use case name	Delete account		
Scenario	Delete user accounts through web application		
Triggering event	The administrator select delete account button		
Brief description	The administrator can delete account that are not being used or that are not necessary		
Actors	Administrator		
preconditions	Administrator must login in. User account must exist		
Flow of Activities	Actor	System	
	1. Log into the system 2. Click the button manage account 3. Admin clicks the "Delete Account" button	1.1. CFIS home page appears 2.1. The system displays account managing page 3.1. The system will delete the account and redirect to account managing page form	
Exceptional Condition	1a. Administrator inserts wrong credential		

Table 7: Use case Description for Delete account

1.11 Class Diagram

Class Diagram models class structure and contents using design elements such as classes, packages and objects. It also displays relationships such as containment, inheritance,

associations and others.

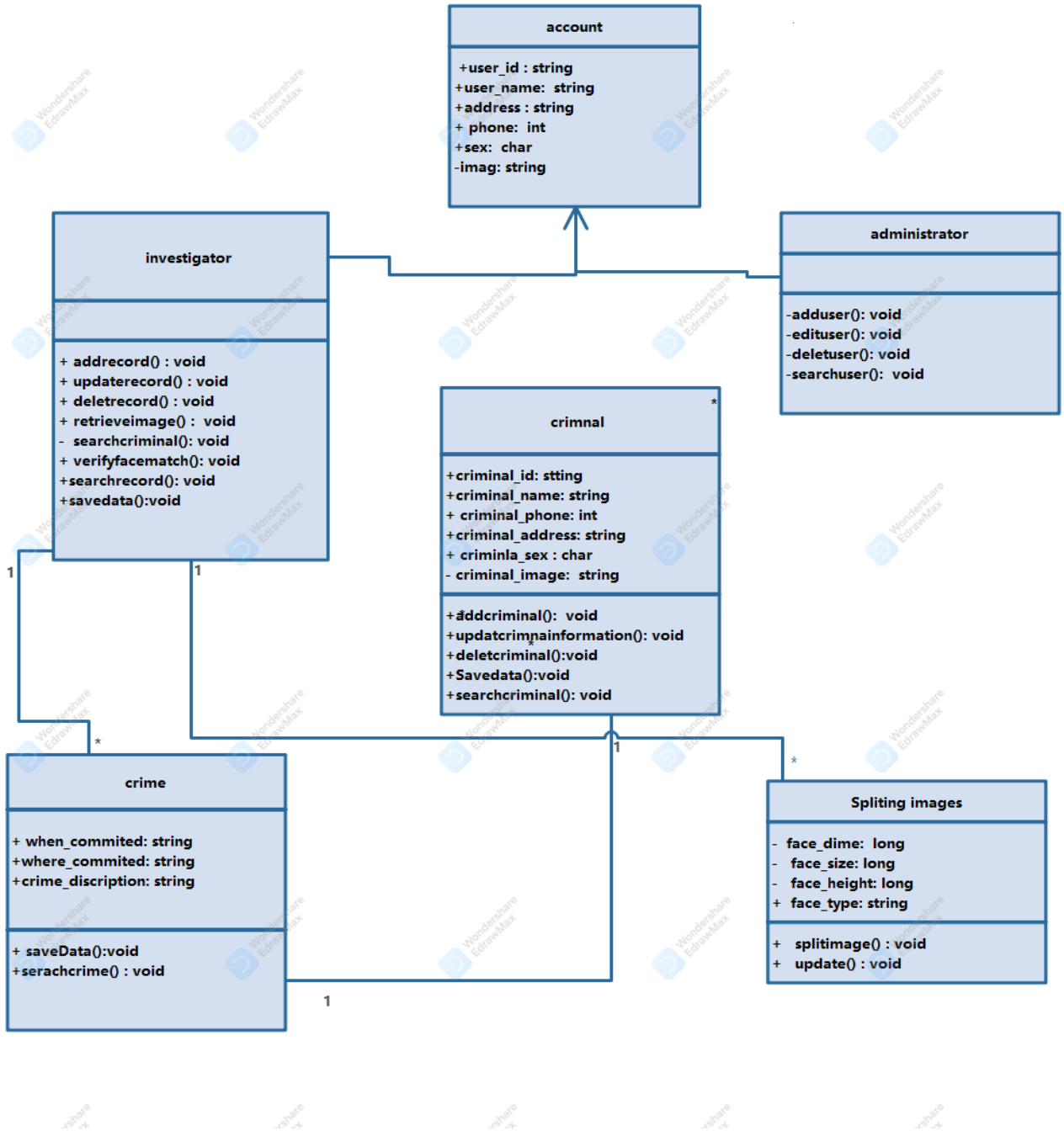


Figure 3: Class diagram for CFIS

1.12 Activity Diagram:

Activity diagrams are a loosely defined diagram technique for showing workflows of stepwise activities and actions, with support for choice, iteration and concurrency.

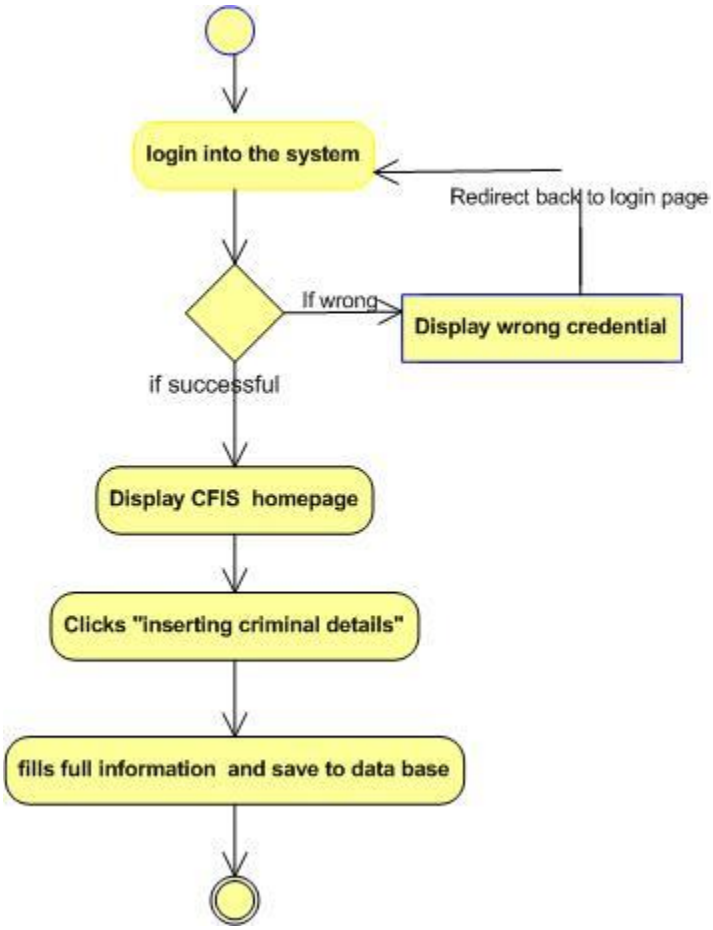


Figure 4: Activity diagram for inserting criminal details

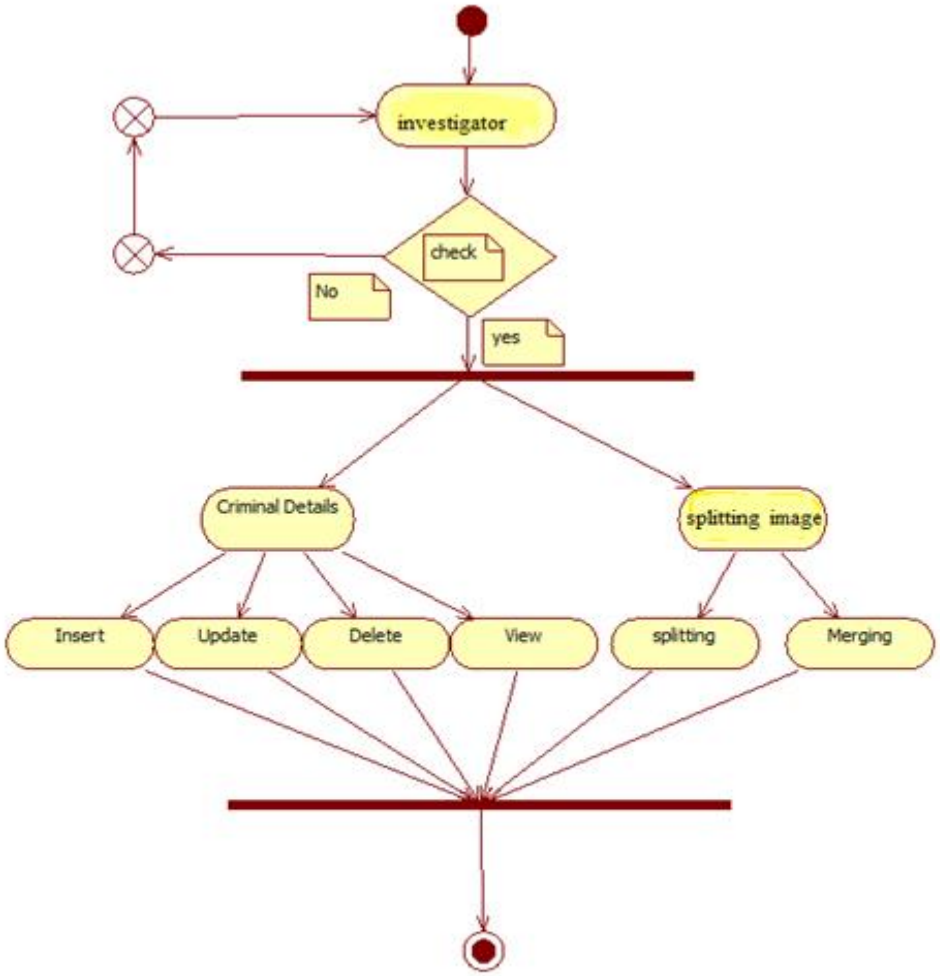


Figure 5: Activity diagram for criminal details and splitting images

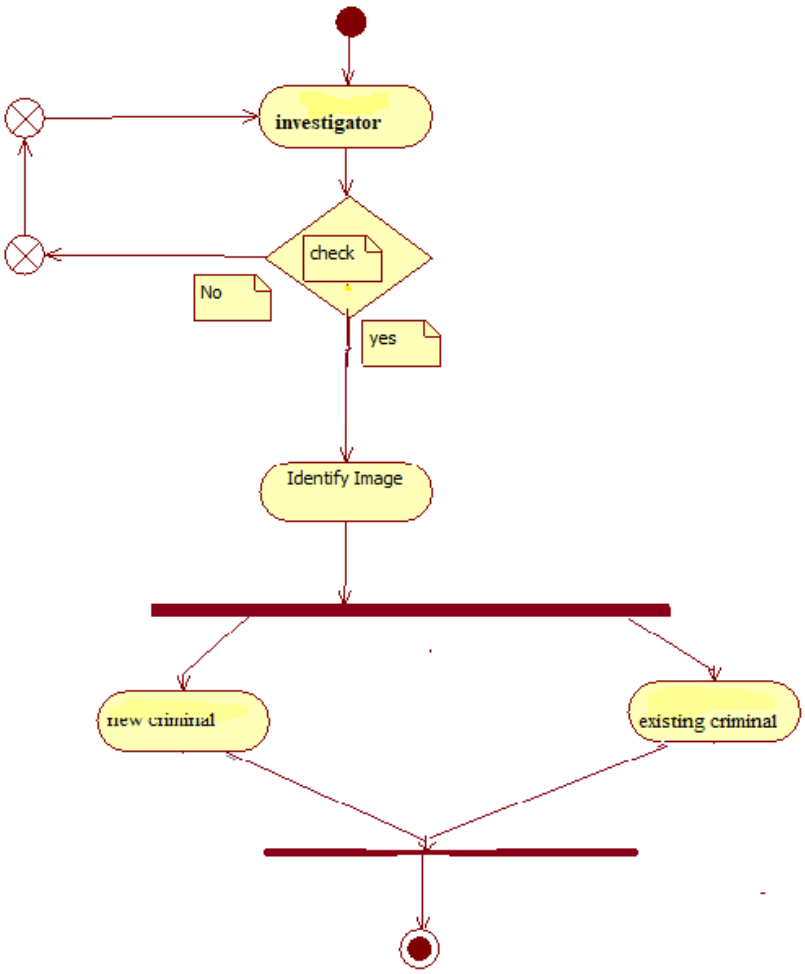


Figure 6: Activity diagram for identify image

1.13 Sequence Diagram:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called Event-trace diagrams, event scenarios, and timing diagrams. A sequence diagram shows, as parallel vertical lines ("lifelines"), different processes or objects that live simultaneously, and, as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

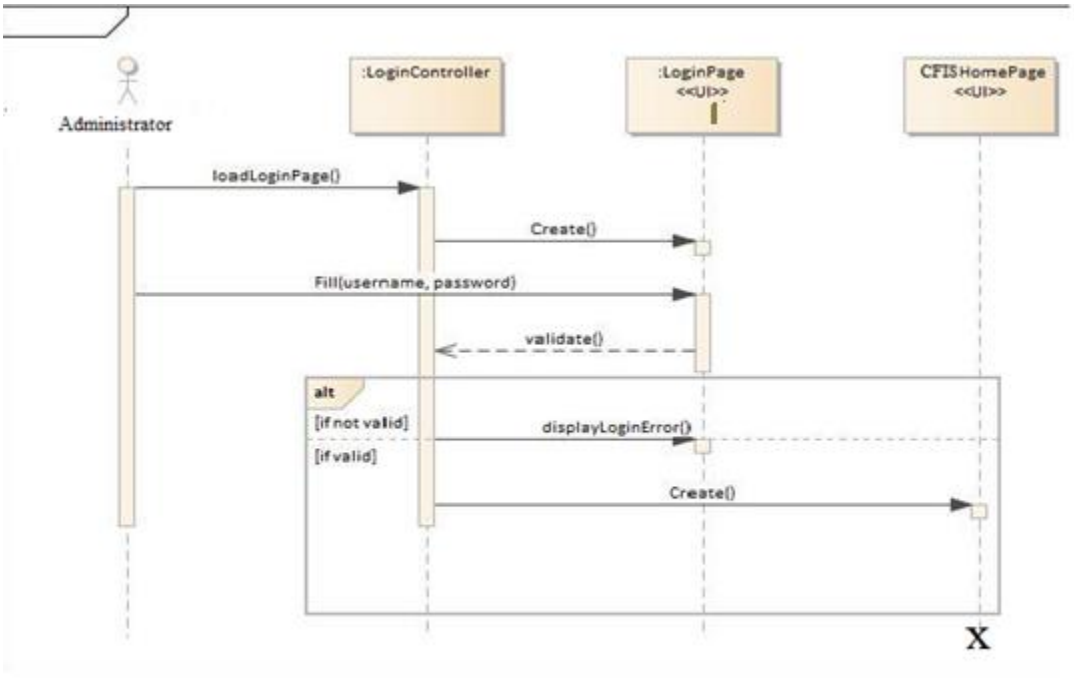


Figure 7: Sequence diagram for administrator create user account

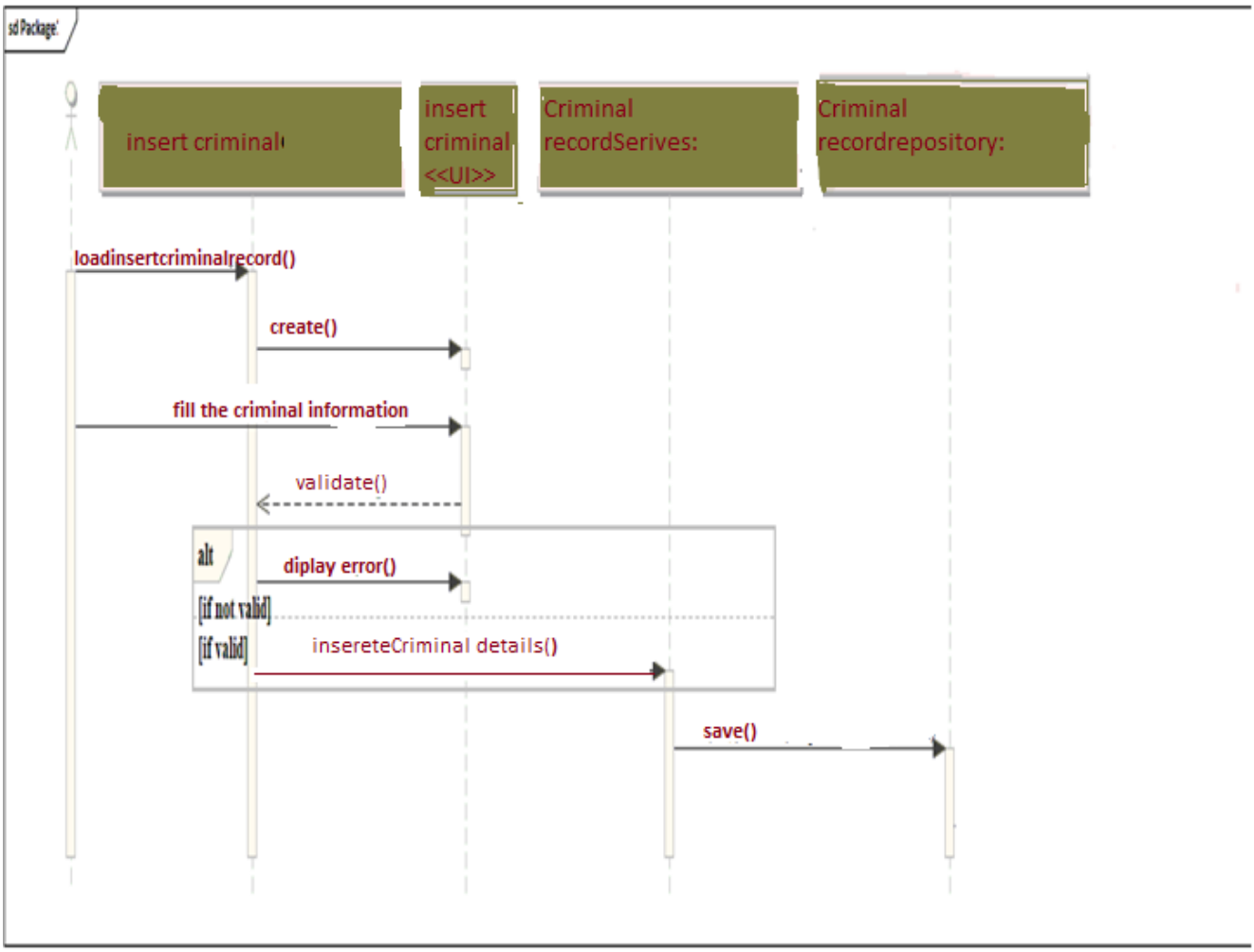


Figure 8: Sequence diagram for insert criminal datils

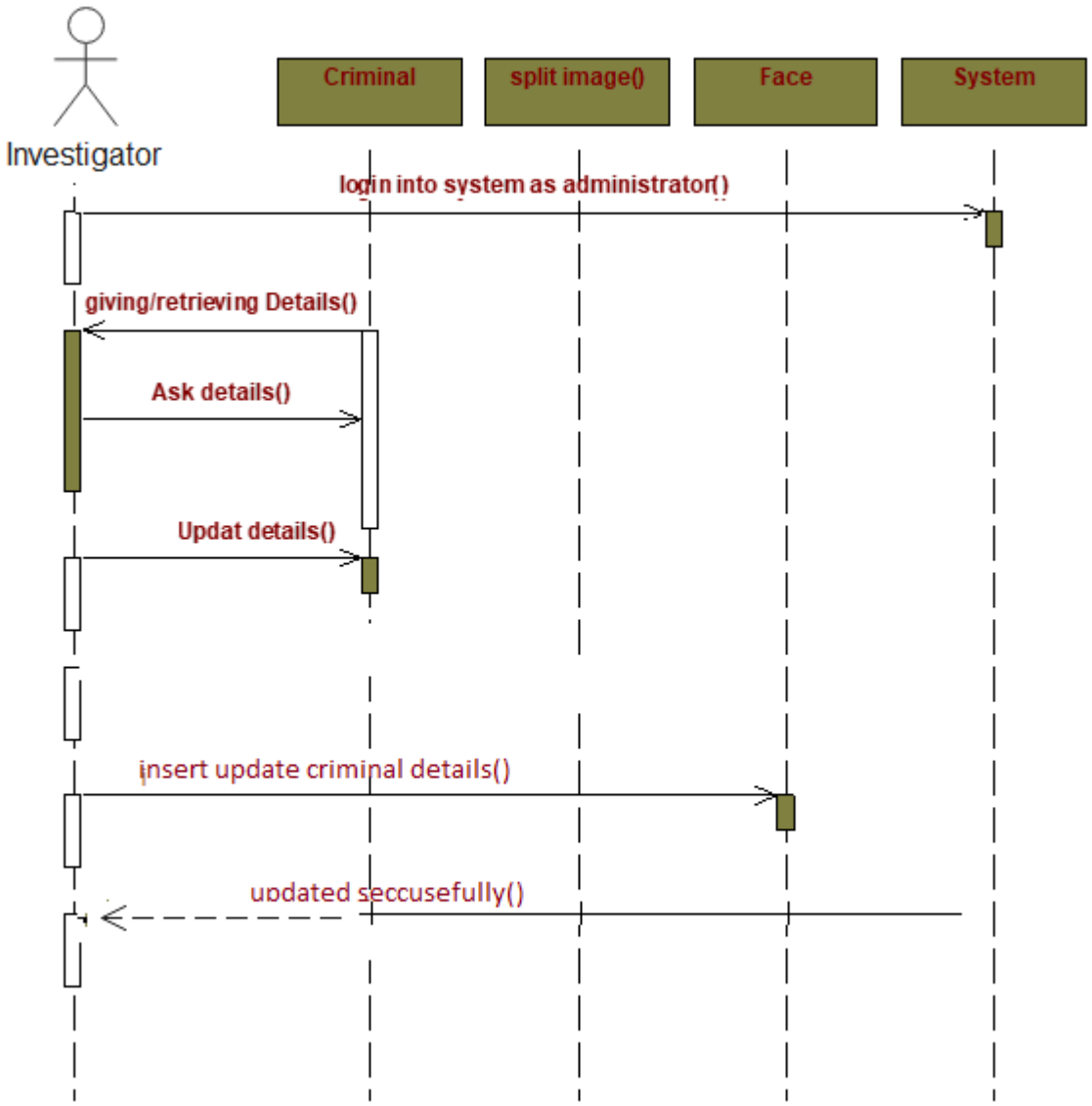


Figure 9: Sequence diagram for Investigator updating criminal details

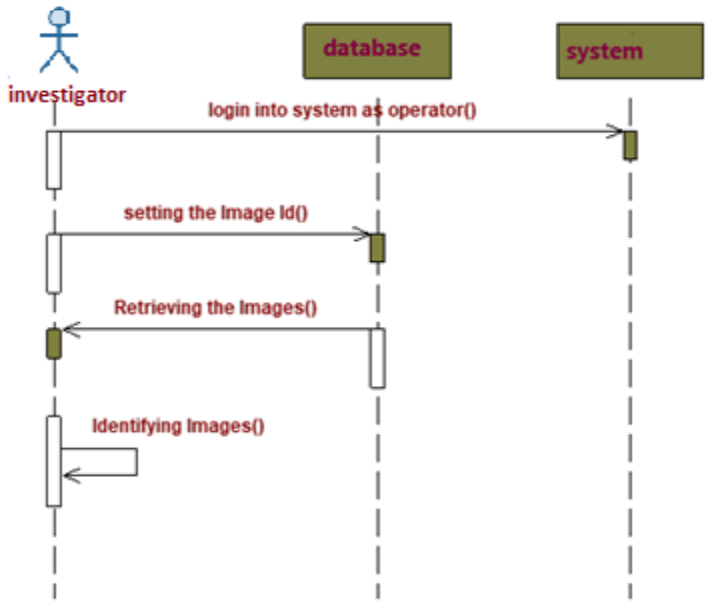


Figure 10: Sequence diagram for investigator identifies images

1.14 Collaboration Diagram

A collaboration diagram describes interactions among objects in terms of sequenced messages.

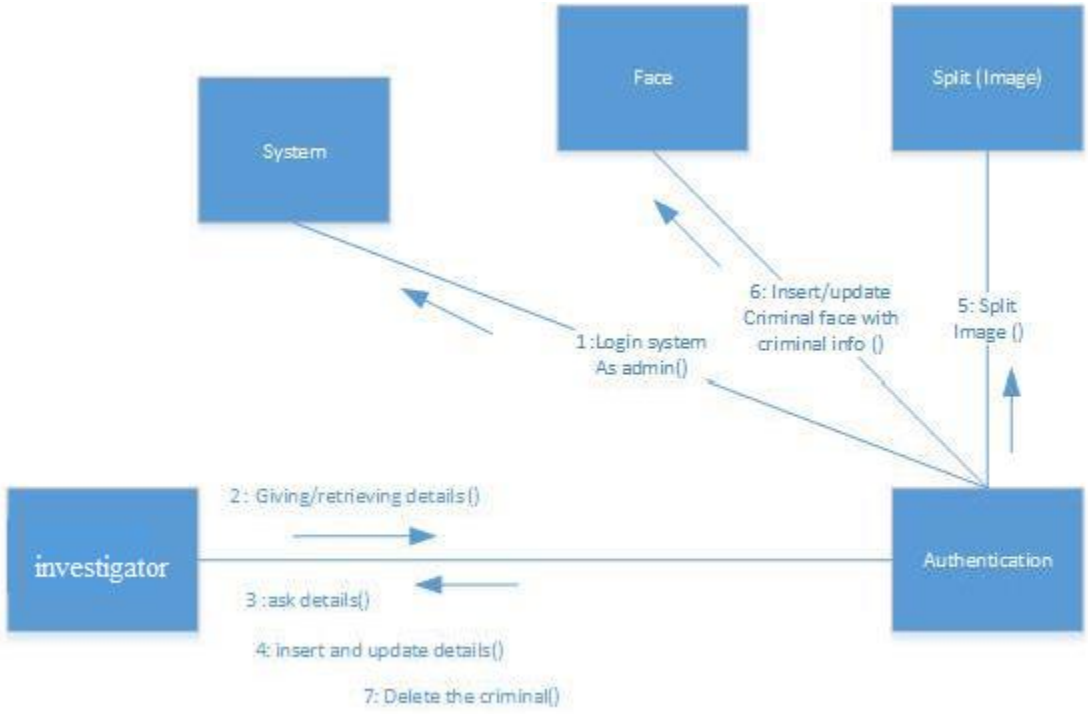


Figure 11: Collaborative diagram for investigator inserts update criminal information

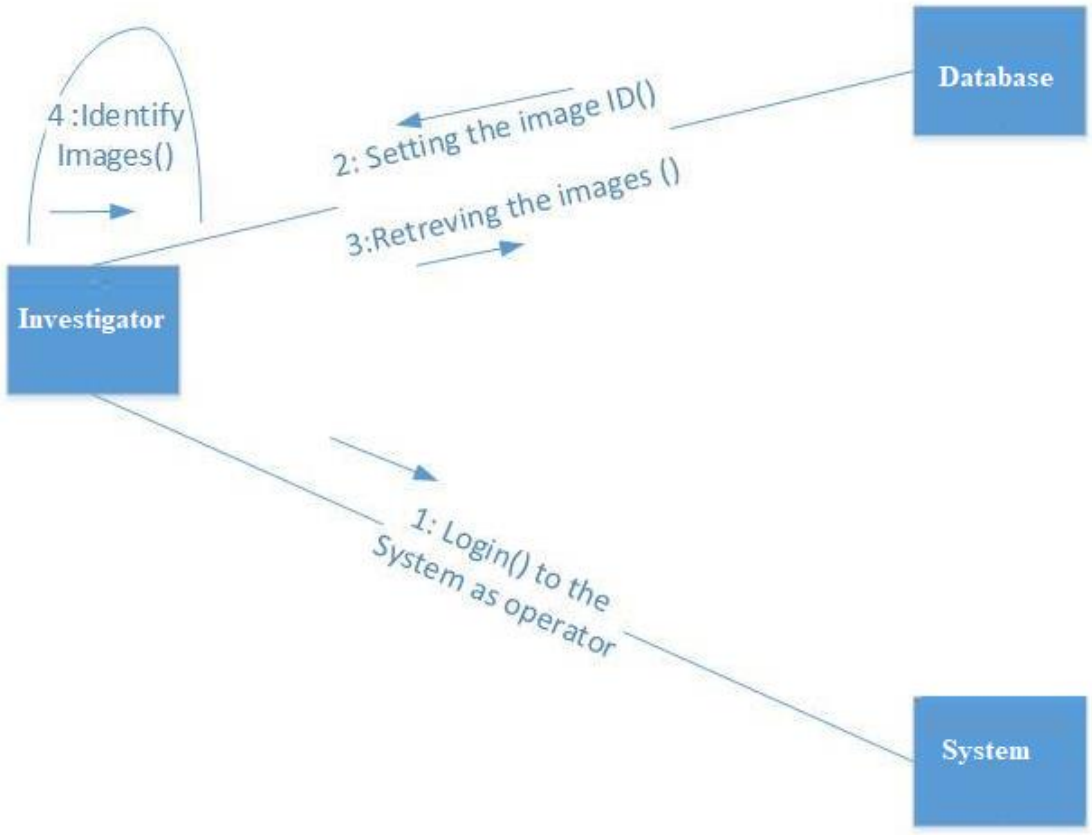


Figure 12: Collaborative diagram for investigator identifies criminal

1.15 User Interface

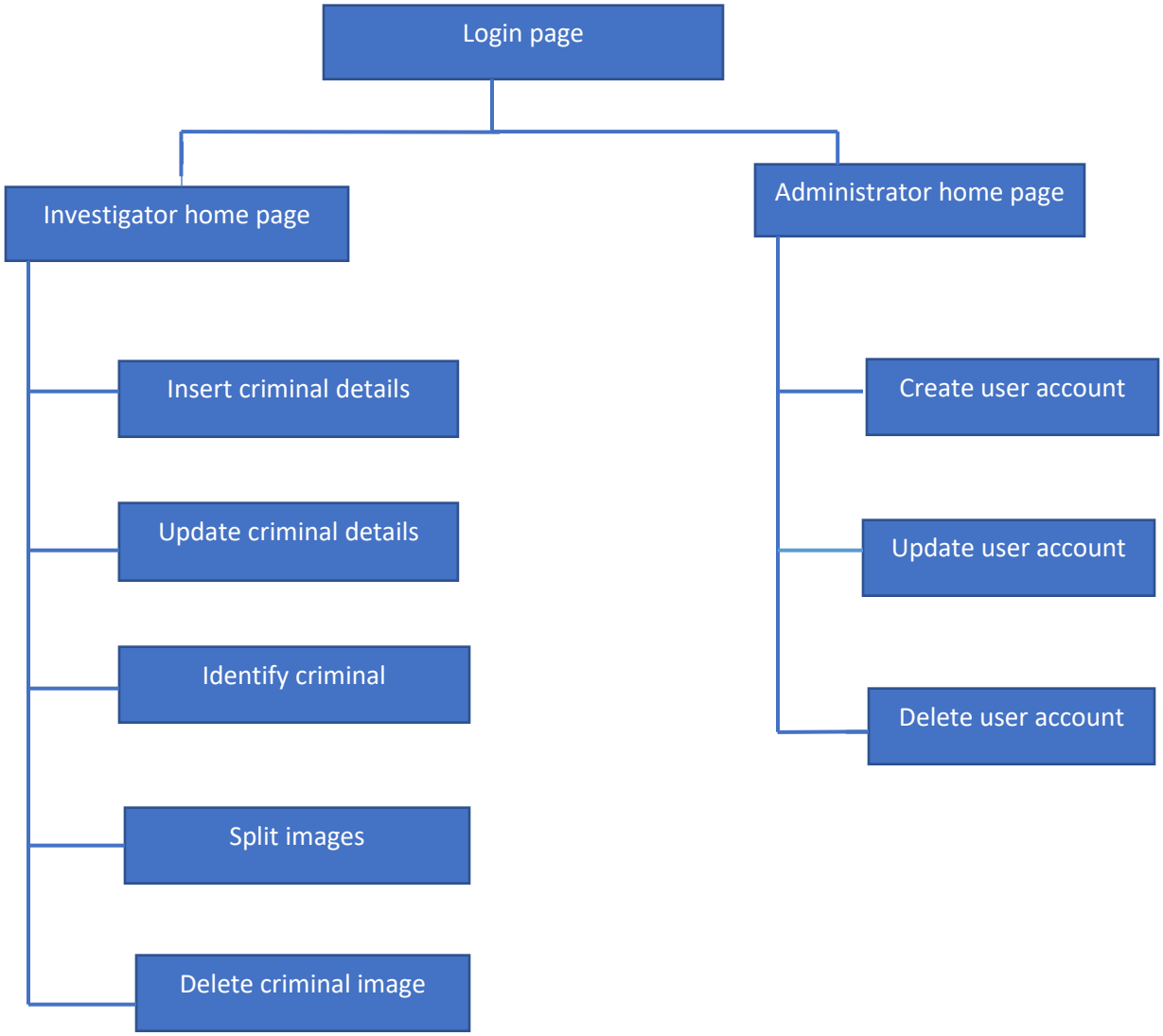


Figure 13: User interface

CHAPTER TWO

2 SYSTEM DESIGN

2.1 Overview

System design is the first design stage in which the basic approach to solving the problem is selected. During system design, the overall structure and style are decided. The system architecture is the overall organization of the system into components called subsystems. The architecture provides the context in which more detailed decisions are made in later design stages. By making high level decisions that apply to the entire system, the system designer partitions the problem into subsystems so that further work can be done by several designers working independently on different subsystems.

2.2 Purpose

The purpose of CFIS system design is to provide sufficient detailed data and information about the system and its system elements to enable the implementation consistent with architectural entities as defined in models and views of the system architecture.

The purpose of system design is to reduce the complexity of the system design (solution domain), we decompose the system into simpler parts, called subsystems, which are made of a number of solution domain classes. Each subsystem can be represented as a directory containing all the files implementing the subsystem with a set of related operations that share common purpose so as to provide service to other subsystem.

2.3 Abbreviation and Definition

GUI: Graphical user interface

ACL: Access control list

CFIS: Criminal face identification system

I/O: Input out put

Invariant: constraints associated with classes

Precondition: specify constraints that a caller must meet before calling an operation; are associated with a specific operation

Post-condition: specify constraints that the object must ensure after the invocation of the operation.

2.4 System Design Model

2.5 General Architecture of Proposed System

The system architecture is designed on the basis of the context of the system in accordance with the principles of architectural design as well as domain knowledge.

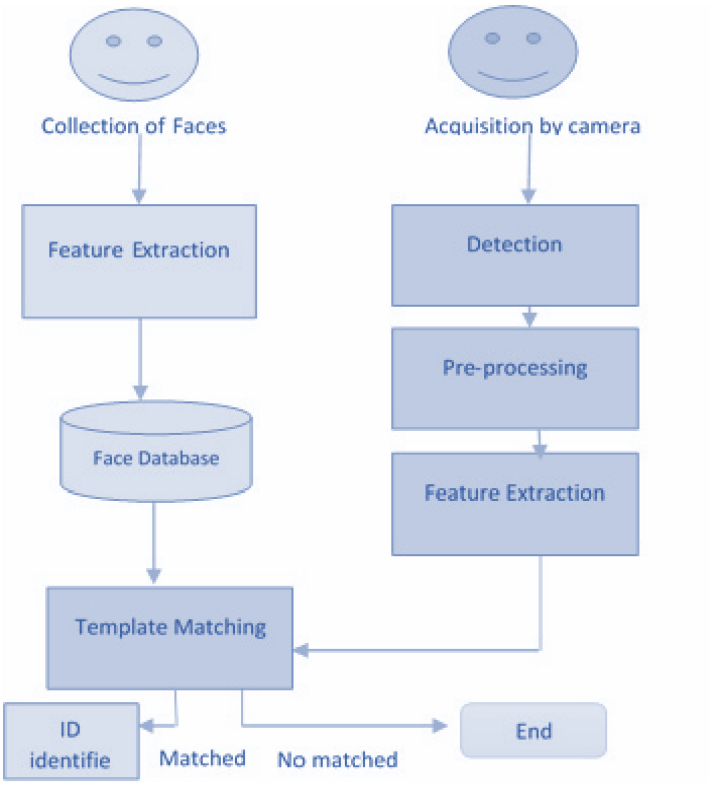


Figure 14: Architecture of CFIS

2.6 Design goals

Design goals represent the expected qualities of the system and provide a consistent set of criteria that must be considered when making design decisions. Most of the design goals of the system are inferred from the non-functional requirements and the application domain will follow the same set of criteria. The design goals can be generally grouped into performance criteria, dependability criteria, maintenance criteria and end user criteria.

Performance Criteria

Throughput: The system should be able to support a number of users at a time using the available bandwidth of the system.

Maintenance criteria

Modifiability: the system should be easily extensible to the need of the CFIS user's data formats availability and to add new functionalities to the system.

Dependability criteria

Reliability of our system is going to be designed with the concept of distributed system, due to the use of redundancy of hardware that will take on the service of the failed device

Security: the system does not allow non-authorized users using a form-based authentication**.**

User criteria utility: the system must address the possible functional requirement of the system users.

Usability: The system should be user friendly, and easy to learn and use.

Identifying Concurrency:

One important goal of system design is to identify which objects must be active concurrently and which objects have activity that is mutually exclusive. The latter objects can be folded together in a single thread of control or task. But there is no part that is concurrent in our system.

Allocating Subsystems to Processor:

In this step system designer estimates the hardware resources required and the implementation choice of either hardware or software. In our system all the subsystems will be implemented in software. The hardware requirements are general such as, 8GB of RAM, Hard Disk Drive 1TB with minimum 250 GB free disk space, CPU 4.20 GHz or faster 64-bit processor.

Management of Data Stores:

In this stage the system designer decides what format is used to store the data stores. There are DBMS systems or file systems and others. Here in our project there are no data stores except files. We then definitely prefer files to inserting and updating data.

2.7 Sub-system decomposition

The first step in system design is to divide the system into small number of components. Each major component of a system is called a sub system. Each subsystem encompasses aspects of the system that share some common property – similar functionality, the same physical location, or execution on the same kind of hardware.

A subsystem not an object nor a function but a package of classes, associations, operations, events, and constraints that are interrelated and that have a reasonably well-defined and small interface with other subsystems. A subsystem usually identified by the services it provides. A service is a group of related functions that share some common purpose such as I/O processing. A subsystem defines a coherent way of looking at one aspect of the problem.

Each subsystem has a well-defined interface to the rest of the system. The interface specifies the form of all interactions and the information flow across subsystem boundaries but does not specify how the sub system is implemented internally. Each subsystem then can be designed independently without affecting the others.

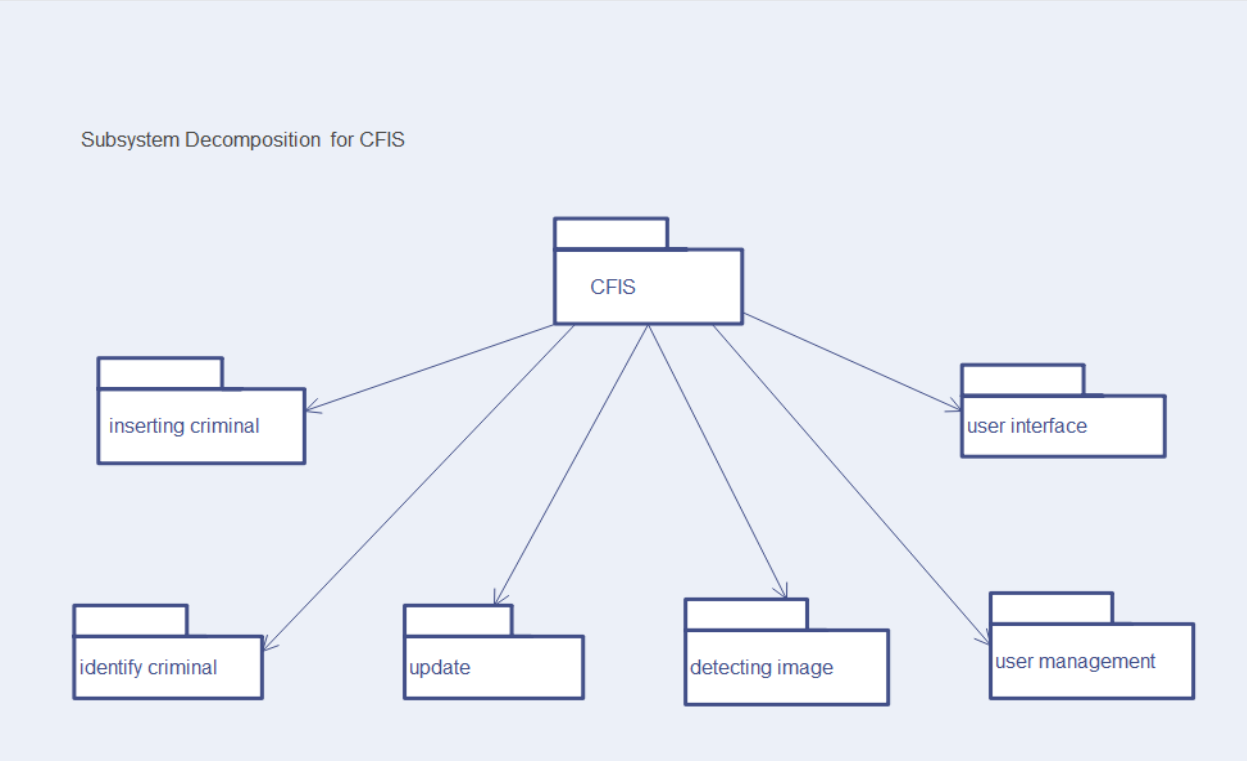


Figure 15: Subsystem decomposition for CFIS

Description subsystem decomposition for CFIS:

- The inserting criminal subsystem is responsible for registering all the criminal related data starting from the information about criminal and when crime is done and it also record and save each and every activity on the crime process
- The user management sub system enables the administrator to manage the user accounts (system user). The management includes creation of new accounts, deleting the existing account and update of accounts.
- The identify criminal subsystem identify criminal's information about the criminal details with images. It also displays the previous history.
- Update sub system enables the investigator and administrator update the information about the criminal and user.
- Detecting image subsystem is responsible for detect the criminal images.

2.8 Deployment Diagram

Deployment diagrams show "the allocation of Artifacts to Nodes according to the Deployments defined between them." Deployment of an artifact to a node is indicated by placing the artifact inside the node. Instances of nodes (and devices and execution environments) are used in deployment diagrams to indicate multiplicity of these nodes. For example, multiple instances of an application server execution environment may be deployed inside a single device node to represent application server clustering.

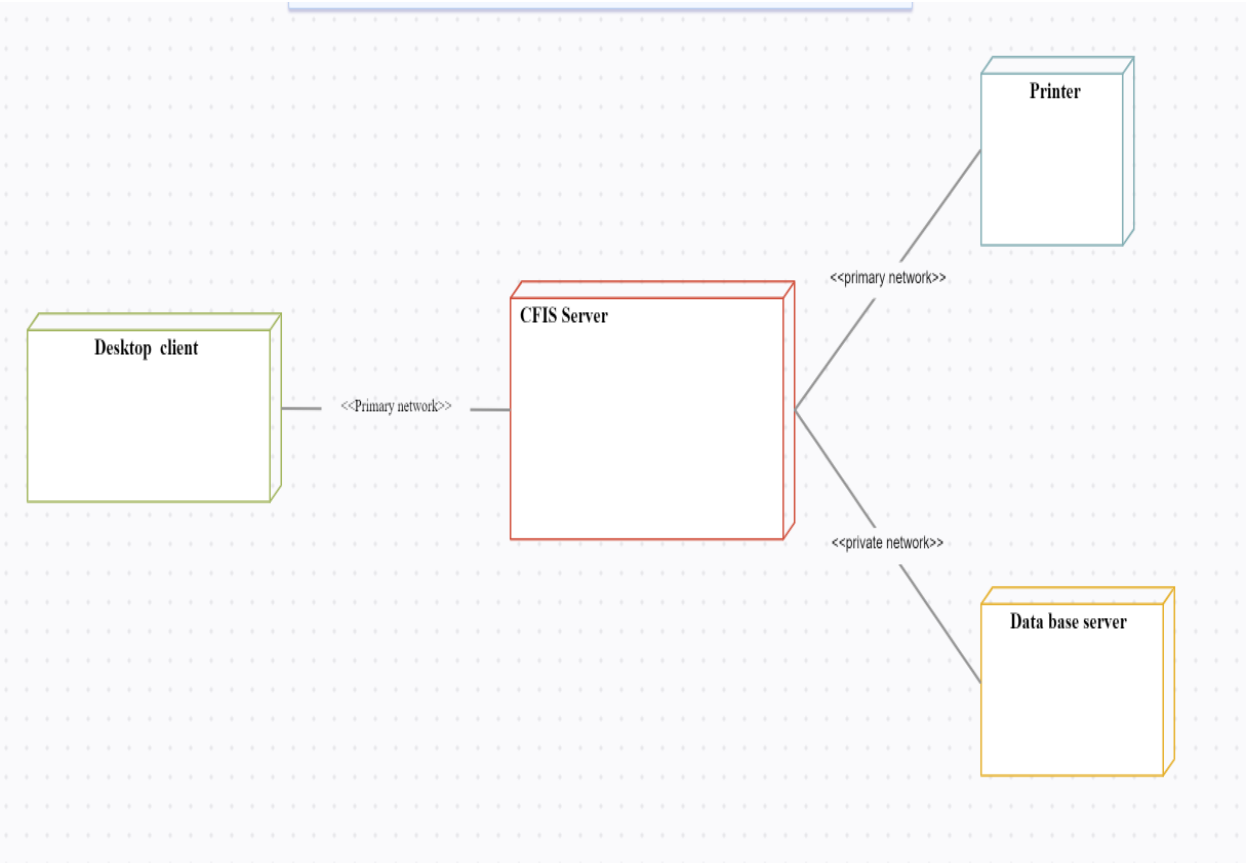


Figure 16: Deployment diagram for CFIS

2.9 Database Design

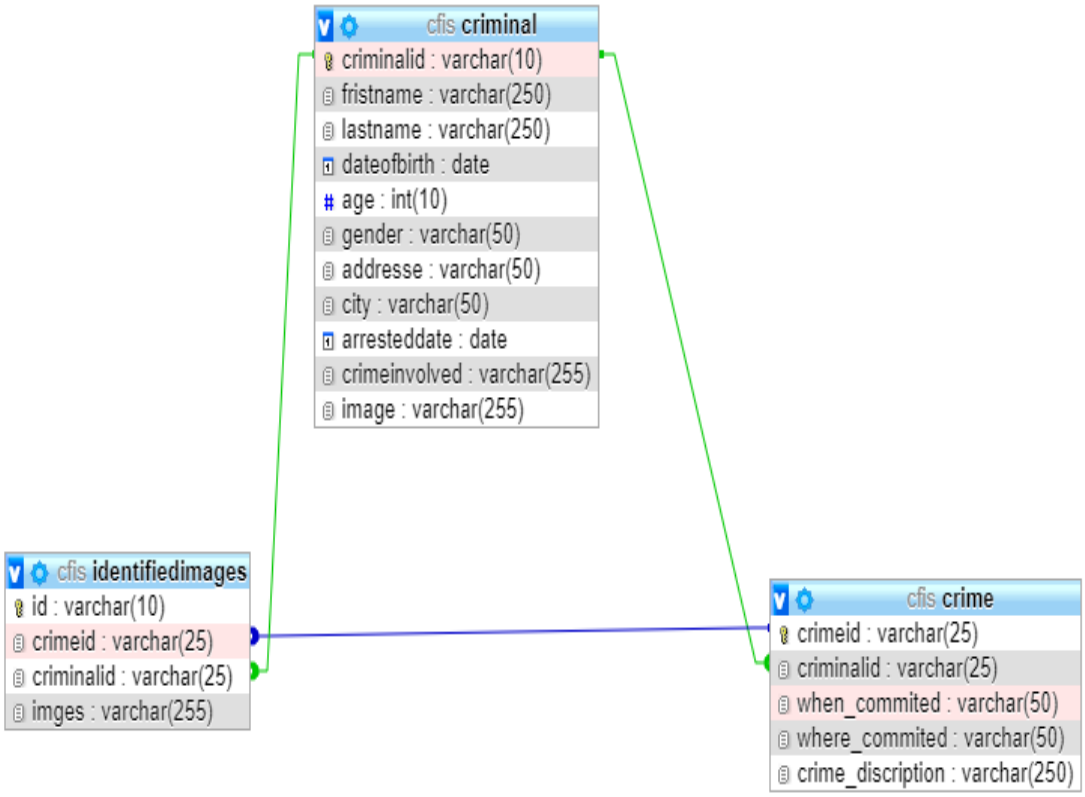
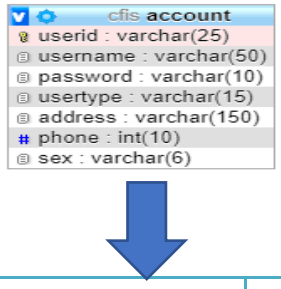


Figure 17: Database design



Attribute	Data Type	Description
userid	Varchar	Primary key
username	Varchar	Not null, maximum of 50-character sets.
password	varchar	Not null, the user password maximum of 10-character sets
usertype	Varchar	Not null, maximum of 15-character the user maybe Administrator or investigator
address	Varchar	Not null, maximum of 150-character sets the address is city (like Addis Ababa, Jima, Gonder)
Phone	Integer	Not null, maximum of 10-character
Sex	Varchar	Not null, maximum of 6-character

Table 8: Description for class account

Operation

addrecord() : is used to adding the details of the criminals along with the criminal photo.

updaterecord() : is used to update criminal record

deletrecord() : is used to delete the criminal details along with the photo.

retrieveimage() : is used to retrieve images

searchcriminal(): is used to search criminal information

verifyfacematch(): is used to identify the criminal images

Constraints

invariants:

All attributes should not be null

Pre-condition

user_id: user id must exist

Post-condition

verifyfacematch(): the criminal images identified

2.10 Interface Design

Actually, every application has one user interface for accessing the entire application. In this application also, we are providing one user interface for accessing this application. The user interface designed completely based on the end users. It is providing friendly accessing to the users. This user interface has attractive look and feel. Technically we are using the swings in core java for preparing this user interface.

Figure 18: CFIS login page

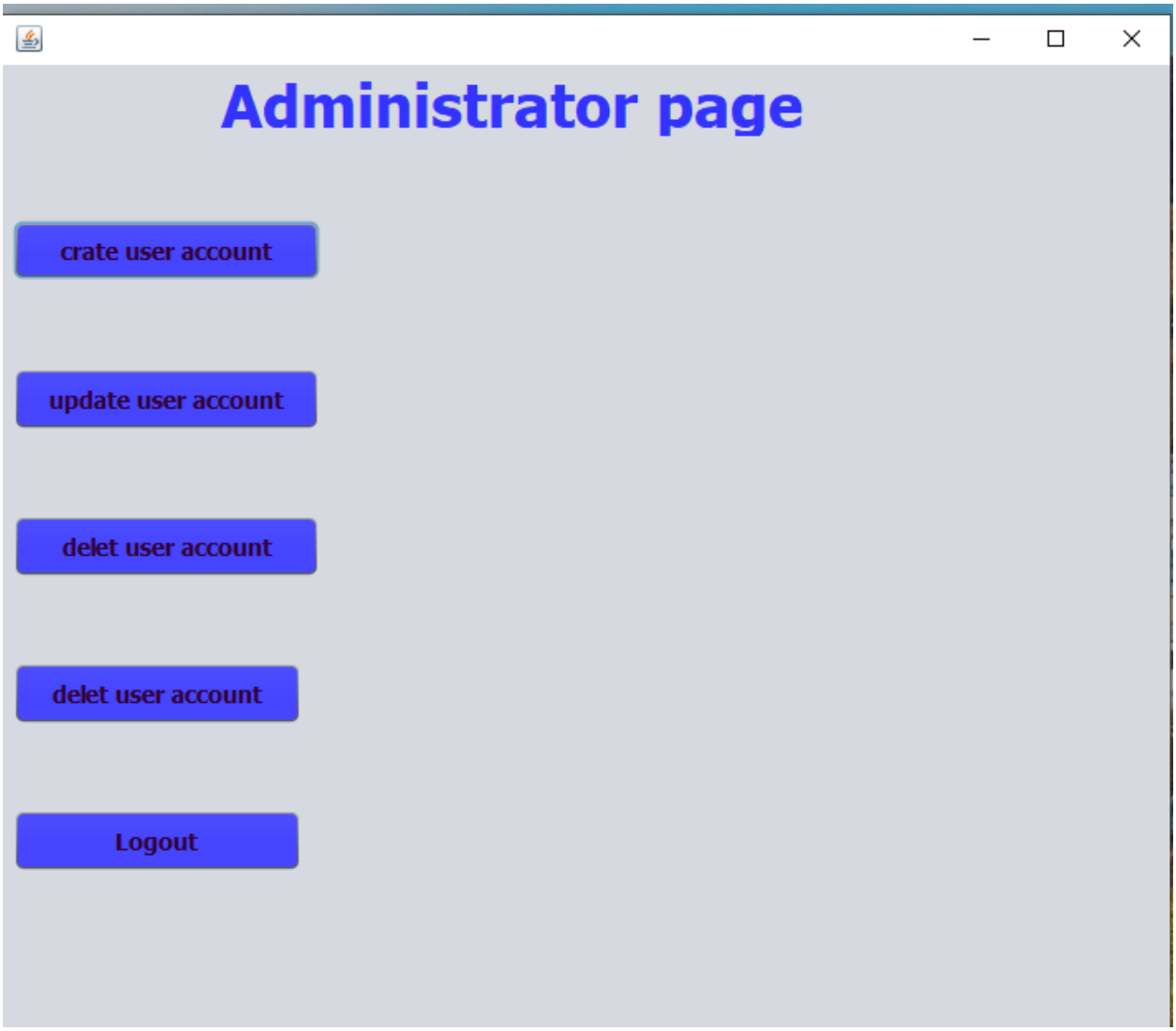


Figure 19: Admin page

2.11 Access control list

An access control lists a table that state an actor access right on a privilege. An actor access rights can be of the type Access, full control and no access. Each column of the access control matrix is called an Access Control List (ACL) while each row is called a capability list. An ACL is connected to the privileges and outline actions each actor can perform on that privilege. In our access control list, we take the sub system as actor's privileges.

	Inserting criminal subsystem	User account management subsystem	Identify criminal subsystem	Update sub system	Detecting subsystem
Investigator	✓		✓	✓	✓
Administrator		✓		✓	

CHAPTER THREE

TESTING

3 Introduction

Testing is the process of finding differences between the expected behavior specified by system models and the observed behavior of the system. It's the process of executing a program with the intention of finding errors.

The system testing is performed after the implementation to ensure that the system is functioning according to the specification without errors and performance of the system is satisfactory. Then quality of the system can improve.

3.1 Objective of system testing

- System testing should allow measuring system, evaluating the system and find verify an error.
- All tests should be traceable to customer requirements.
- Tests should be planned long before testing begins.
- Testing should begin in the small and progress to the large.
- To be most effective, testing should be conducted by an independent third party.

3.2 Features to be tested

3.2.1 Suggestions for Future Enhancements

The following future enhancement are proposed in order to overall limitations of functionality identified above and improved the application in general.

The Future enhancements of this project include the following:

- Detecting and identify any criminals face by using road surveillance camera from anywhere, any time.
- Use an algorithm to measure facial expressions and angle. A robust algorithm like Neural Network should be used to improve the system's accuracy in above conditions.
- Improve as a security application

As the current system is a generic face recognition application, it can be improved as security application which we can be applied to a wide variety of application areas including access control for PCs, road surveillance, private surveillance, criminal identification and for security in ATM transactions. In addition, face recognition system is moving towards the next-generation smart environment where computers are designed to interact more like humans.

3.3 Unit testing

Unit testing is carried out at each increment to determine that each module is functioning as required. Therefor errors could be identified and rectified prior to the integration of modules within the system.

Following core unit has identify to be tested

- Face detection module
- Verification module

3.4 Integration testing

Integration testing is the phase of software testing in which individual software modules are combined and tested as a group. It is focused on interactions between components.

Bottom-up testing

- First each component at the lowest level of the system hierarchy is tested. Then test the next higher component calling the previous tested component
- Requires component driver- is a partial implementation of a component that depends on the tested component
- Advantage: interface faults can be more easily found
- Disadvantage:
  - The program as an entity does not exist until the last module is added
  - Faults found in the top layer may often lead to changes in the lower layers, invalidating previous tests

Top-down testing

- Is a reverse of bottom-up.
- In this case, the component being tested may call the untested component. This way testing to the next level will continue

- Requires component stubs- is a partial implementation of components on which the tested component depends
- Advantage: it starts with user interface components
- Disadvantage:
  - the development of test stubs is time consuming
  - if processing at low levels in the hierarchy is required to adequately test upper levels, no significant data can flow upward in the program structure

Due to the advantage and disadvantage of top down and bottom up testing we are preferred bottom up testing because of its easily found and bottom down testing is time consuming.

- It will check following module for integration testing
  - Face detection
  - Matching module
  - Verification module

3.5 testing strategy

A test strategy is an outline that describe the testing approach of the software development cycle.

3.5.1 Testing Strategy: Integration Testing

Integration testing was carried out at the end of each increment to determine that the integrated modules were functioning according to the specification without errors are required.

Data can be lost across an interface, one module can on another; sub-functions when combined may not produce the desired major function. Global data structures can present problems. Integration testing is a systematic technique for the program structure while at the same time concluding tests to uncover errors associated with interface. All modules are combined in this testing step. Then the entire program is tested as a whole. Each of the module is integrated and tested separately and later all modules are tested together for some time to ensure the system as a whole works well without any errors.

Entrance Criteria: All the units developed in an increment are integrated.

Exit Criteria: The integrated units perform in the expected behavior without errors.

Test type	Face detection
Test Unit/functionality	Detect the face region
Description	In this case system check whether its image is a face or not. If it is a face the region of the face will be detected.
Test plan	Click on *detection face* button to detect the faces
Input	An image that the format of JPEG, PNG
Output (Unsuccessful)	Display an error message

Table 9: Test Case: 01 Face detection

3.5.2 Testing Strategy: Overall System Testing

Overall system testing was done after integration testing ensure that the modules integrated within system increments were functioning as specified as a final product.

Test Type	System Integrated Testing
Rational	This is ensuring that overall system is functioning according to the specification and without errors but have some limitations.
Test plan	Adding a new record
Description	This checks whether the input image is an image or not. If it is an image it detects the image features and save it in the database along with the criminal ID
Test plan	Click on start recognition button in main window button Check relevant out put Check database values to verify
Success criteria	Display an error
Output (Unsuccessful)	Display an error message

Table 10: Test case: 02 Overall System Testing

3.6 Test Case Design

A test case is a set of input data and expected results that exercises a component with the purpose of causing failures and detecting faults.

There are two type of test case deign

Black-Box Testing

The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.

White-Box Testing

White-box testing is the detailed investigation of internal logic and structure of the code. White-box testing is also called **glass testing** or **open-box testing**. In order to perform **white-box** testing on an application, a tester needs to know the internal workings of the code.

The tester needs to have a look inside the source code and find out which unit/chunk of the code is behaving inappropriately. we are using white-box testing in the system.

Input	Expectation result	Action outcome	Pass/fail
Valid username and password	The system allows to login	Yes	Pass
Invalid username and password	The system displays incorrect username and password	Yes	Pass
Valid username and invalid password	The system displays incorrect username and password	Yes	Pass
Invalid username and valid password	The system displays incorrect username and password	Yes	Pass

Table 11: Test Case Design

3.7 Test case specification

Test Case Specification document described detailed summary of what scenarios will be tested, how they will be tested, how often they will be tested, and so on and so forth, for a given feature. It specifies the purpose of a specific test, identifies the required inputs and expected results, provides step-by-step procedures for executing the test, and outlines the pass/fail criteria for determining acceptance.

Test Case Specification has to be done separately for each unit. Based on the approach specified in the test plan, the feature to be tested for each unit must be determined. The overall approach stated in the plan is refined into specific test techniques that should be followed and into the criteria to be used for evaluation. Based on these the test cases are specified for the testing unit.

Conclusion

On conclusion of this project it can be mentioned that the project has effectively achieved the objectives defined at the initiation of the project. The scope of the project is confined to store the criminal details with image and store in the database. When a person has to be identified the images stored in the database are compared with surveillance road camera system residing at some public place or roads which automatically matches the input faces with criminal database and gives alert if the results are matched. If any image is matched up to 95% or closer to that rate then system predict that he/she is the criminal.

Glossary

Administrator: the person who own the right to manage the system.

Investigator, who belongs to the investigating department, is responsible for entering the criminal details and maintains them. He adds, deletes and updates the criminal details. And also, he identifies criminals.

FPCIB which is named "**Ethiopian Federal Forensic Department Bureau**" which is located near "Black Lion Hospital" in Addis Ababa, Ethiopia. Furthermore, the accuracy of checking each and every one of the applicants from the criminals' record is depending on the ability and experience of the crime investigation officers and they have the responsibility in checking and identifying individuals as a criminal.

References

[1]. Face Recognition Methods & Applications, [Online], Available: [Accessed 23rd October 2014]

http://www.ijetae.com/files/Volume4Issue7/IJETAE_0714_19.pdf.

[2]. Face Recognition: "Eigenfaces and Fisherfaces Performance across Pose, by Alan Brooks [Online]", [Accessed 23rd October 2014] Available:

http://dailyburrito.com/projects/facerecog/facerecreport.html.

[3]. "Ethiopian Police Clearance Certification System Using by Mulatu Gebeyaw Astarkie Debre Tabor University and Kamil Dimililer Near East University"

*[**Conference Paper** · November 2019].*

Available: (https://www.researchgate.net/publication/337439350)

[4]. Vishnu G. and et al, B. R. (2017). "Face Recognition Techniques: A Survey. International Journal of Innovative Research in Computer and Communication Engineering."

[5]. Riddhi P. and Shruti B., R. P. (2013). "A Literature Survey on Face Recognition Techniques". International Journal of Computer Trends and Technology (IJCTT).

[6]. Fundamentals: Software Engineering Dr. Rami Bahsoon School of Computer Science The University Of Birmingham r.bahsoon@cs.bham.ac.uk www.cs.bham.ac.uk/~rzb.

[7]. Face Detection using Viola Jones Algorithm
https://www.mygreatlearning.com/blog/viola-jones-algorithm/.

[8]. "Samas Yeach Yourelf UML in 24 Hours", third edition ©2004 by Joseph Schmuller.

[9]. "Object-Oriented Software Engineering Conquering Complex and Changing Systems", © 1999 Bernd Bruegge and Allen H. Duto Carnegie Mellon University School of Computer Science Pittsburgh, USA

[10]. "Samas Teach Yourelf UML in 24 Hours", third edition ©2004 by Joseph Schmuller.

[11]. "Object-Oriented Software Engineering Conquering Complex and Changing Systems",

© 1999 Bernd Bruegge and Allen H. Duto Carnegie Mellon University School of Computer Science Pittsburgh, USA.

[13]. "tutorials point on System Analysis & Design - System Design" website https://www.tutorialspoint.com/system_analysis_and_design/system_design.htm.