Mini Project Report

on

BIOMETRIC ATTENDANCE SYSTEM

Submitted by

Vikash Ranjan(12130086)

Sandeep Kumar(12130243)

Sujeet Kumar(12130081)

Prabhash Chandra(12130057)

In Partial Fulfilment for the Award of degree of

B-TECH DEGREE

in

COMPUTER SCIENCE & ENGINEERING

SCHOOL OF ENGINEERING

COCHIN UNIVERSITY OF SCIENCE &

TECHNOLOGY KOCHI-682022

MARCH 2015

Division of Computer Engineering
School of Engineering

Cochin University of Science &
Technology, Kochi-682022

CERTIFICATE

Certified that this is a bonafide record of the Mini Project work titled

Academic Progressive Report

Done By

Vikash Ranjan(12130086)
Sandeep Kumar(12130243)
Sujeet Kumar(12130081)
Prabhash Chandra(12130057)

of V1 semester Computer Science & Engineering in the year 2015 in partial fulfillment of the requirements for the award of Degree of Bachelor of Technology in Computer Science &Engineering of Cochin University of Science & Technology

Vinod Kumar P P

Head of Division
& Project Guide

Ancy Zachariah

Project Coordinator

ACKNOWLEDGEMENT

I take this occasion to thank computer science and engineering department,school of engineering,cusat for giving me opportunity to do a project . I extend my sincere and heartfelt thanks to our esteemed Guide, **Mr.Vinod Kumar P.P** for providing me with the right guidance and advice at the crucial junctures and for showing me the right way. I extend my sincere thanks to our respected Head of the division **Mr.Vinod Kumar P.P**,and our Coordinator **Mrs.Ancy Zachariah**., for allowing us to use the facilities available.

I would like to thank the other faculty members also, at this occasion. Last but not the least, I would like to thank my friends and family for the support and encouragement they have given me during the course of our work.

VIKASH RANJAN
SANDEEP KUMAR
SUJEET KUMAR
PRABHASH CHANDRA

ABSTRACT

The main objective of developing Biometeic Attendance Application For Colleges. Mini project is to provide college attendance for students in college and schools. The advantage of this application is to provide finger print based system. The students should give their finger prints before entering into class. Attendance is marked after student identification. For student identification, a fingerprint recognition based identification system is used.

Human fingerprints are rich in details called minutiae, which can be used as identification marks for fingerprint verification. The goal of this project is to develop a complete system for fingerprint verification through extracting and matching minutiae. To achieve good minutiae extraction in fingerprints with varying quality, preprocessing in form of image enhancement and binarization is first applied on fingerprints before they are evaluated. These processes are carried out by the use of IDE(Net beans), Database made using SQL and programming languages such as JAVA, MY SQL.

List of Tables

List of Figures

Contents

Chapter 1

INTRODUCTION

Fingerprint recognition has been widely used in both forensic and civilian applications. Compared with other biometrics features,fingerprint-based biometrics is the most proven technique and has the largest market shares. Although fingerprint recognition has been studied for many years and much progress has been made, the performance of even state-of-the-art matchers is still much lower than the expectations of people and theory estimation. Therefore, much effort is still needed to improve both the performance and the speed of fingerprint recognition systems. The matching algorithm plays a key role in a fingerprint recognition system.A fingerprint is the pattern of ridges and valleys on the surface of a fingertip. The endpoints and crossing points of ridges are called minutiae. It is a widely accepted assumption that the minutiae pattern of each finger is unique and does not change during one's life. When human fingerprint experts determine if two fingerprints are from the same finger, the matching degree between two minutiae pattern is one of the most important factors.The need for an effective and reliable attendance system in any organization(such as schools,banks,factories,hospitals and etc) has become most important as this can help in determining the timeliness of any staff or student.Biometric methods are much more reliable, help to save the cost,easy to use and quite affordable.

1.1 Problem stetement

Currently, attendances for college courses are recorded using papers and signatures which waste resources. Moreover, this system of tracking attendance has many holes that could easily ben cracked illegally that is when students appear only on the attendance sheet while he/she is not attending the class actually. It also wastes lecture and/or lecturer time and creates headache.

The tasks proposed are :

∗ To study the different issues associated with the problem by conducting a literature survey on fingerprint recognition algorithms and techniques.

∗ To design the attendance database schema.

1.1.1 Project Objective

The aim of this project is to design and implement a system that should allow instructors to track students' attendance for laboratory in current semester.That is if current semester is s4 then attendance will be taken for data structure and digital electronic laboratory.The tasks proposed are:

∗ To study the different issues associated with the problem by conducting a literature survey on fingerprint recognition algorithms and techniques.

∗ To design the attendance database schema.

∗ To run some experiments on fingerprint recognition.

From the gathered information:

∗ To decide how to design a database system.

∗ To decide what specific works should the system provides.

1.2 Using Biometrics

Biometric Identification Systems are widely used for unique identfication of humans mainly for verification and identification. Biometrics is used as a form of identity access management and access control. So use of biometrics in student attendance management system is a secure approach. There are many types of biometric systems like fingerprint recognition, face recognition, voice recognition, iris recognition, palm recognition etc. In this project, we used fingerprint recognition system.

1.3 Fingerprint

A fingerprint is the pattern of ridges and valleys on the surface of a fingertip. The endpoints and crossing points of ridges are called minutiae. It is a widely accepted assumption that the minutiae pattern of each finger is unique and does not change during one's life. Ridge endings are the points where the ridge curve terminates, and bifurcations are where a ridge splits from a single path to two paths at a Y-junction.The ridge endings and bifurcations points are termed as minutiae.First when minutiae patterns are calculated then it contains false minutiae so we need to remove those points to get correct matching results.Fingerprint is taken through scanner so due to present of dirt particles or noise the image taken is not of desired quality so we apply many algorithms to enhance fingerprint image.All algorithms are described briefly in next sections.Fingerprint is considered to be best way to perform identification and verification of individuals.Human fingerprints are detailed, unique, difficult to alter, and durable over the life of an individual making them suitable as long-term markers of human identity and may be employed by police or other authorities to identify individuals who wish to conceal their identity, or to identify people are incapacitated or deceased and thus unable to identify themselves, as in the aftermath of a natural disaster.There are five types of fingerprint mainly whorl,arch,left loop,right loop and tented arch.

1.4 Advantage of Fingerprint

Fingerprints are considered to be the best and fastest method for biometric identification. They are secure to use, unique for every person and does not change in one'slifetime. Besides these, implementation of fingerprint recognition system is cheap,easy and accurate up to satisfiability.Fingerprint recognition has been widely used in both forensic and civilian applications. Compared with other biometrics features , fingerprint-based biometrics is the most proven technique and has the largest market shares . Not only it is faster than other techniques but also the energy consumption by such systems is too less.Fingerprint based systems provide online updation of attendance as required.It removes tedious work done in previous system having pen and paper work.The main advantage of using fingerprint is that it is secure and unique .Fingerprint is mainly patterns of ridges and valleys present on top of fingerprint.

1.5 Using fingerprint recognition system for attendance management

Managing attendance records of students of an institute is a tedious task. It consumes time and paper both. To make all the attendance related work automatic and on-line,we have designed an attendance management system which could be implemented in CUSAT Kochi. It uses a fingerprint identification system developed in this project.This fingerprint identification system uses existing as well as new techniques in fingerprint recognition and matching. A new one to many matching algorithm for large databases has been introduced in this identifiscation system.This project removes many problems that were present in existing systems like pen-paper work,or maintenance of attendance report.When project is ready it can be installed on computers in which one system will works as server which will be kept safe.Each classroom will have one system and scanner so that administrator can start the attendance procedure and students will give their fingerprint image through scanner. fingerprint recognition system for attendance management can manage attendance of any institute. It provides automatic updation of attendance either of employees in any organization or students in any institute.Generally for large students it makes work easy and conveniently.Attendance can be managed manually or automatically but if we are using fingerprint recognition system as attendance managing procedure then we can automatically publish attendance of students as required .We can also generate attendance report monthly and it reduces manual work as all works are done automatically.

1.6 Working of Fingerprint Recognition System

Fingerprint images that are found or scanned are not of optimum quality. So we remove noises and enhance their quality. We extract features like minutiae and others for matching. If the sets of minutiae are matched with those in the database, we call it an identified fingerprint. After matching, we perform post-matching steps which may include showing details of identified candidate, marking attendance etc.

1.7 Fingerprint Enhancement

The image acquired from scanner is sometimes not of perfect quality .It gets corrupted due to irregularities and non-uniformity in the impression taken and due to variations in the skin and the presence of the scars, humidity, dirt etc. To overcome these problems , to reduce noise and enhance the definition of ridges against valleys, various techniques are applied as following.

1.8 Segmentation

Image segmentation separates the foreground regions and the background regions in the image.The foreground regions refers to the clear fingerprint area which contains the ridges and valleys. This is the area of interest. The background regions refers to the regions which is outside the borders of the main fingerprint area, which does not contain any important or valid fingerprint information. The extraction of noisy and false minutiae can be done by applying minutiae extraction algorithm to the background regions of the image. Thus, segmentation is a process by which we can discard these background regions, which results in more reliable extraction of minutiae points.

1.9 Normalization

Image normalization is the next step in fingerprint enhancement process. Normalization is a process of standardizing the intensity values in an image so that these intensity values lies within a certain desired range. It can be done by adjusting the range of grey-level values in the image.This method maintains intensity of the image.Normalization is performed on the segmented fingerprint image ridge structure so as to standardize the level of variations in the image grey-level values. By normalization, the grey-level values are made to fall within certain range that is good enough for improved image contrast and brightness.

1.10 Ridge Frequency Estimation

Another important parameter,in addition to the orientation image, that can be used in the construction of the Gabor filter is the local ridge frequency. The local frequency of the ridges in a fingerprint is represented by the frequency image. The first step is to divide the image into blocks of size W x W. In the next step we project the grey-level values of each pixels located inside each block along a direction perpendicular to the local ridge orientation. This projection results in an almost sinusoidal-shape wave with the local minimum points denoting the ridges in the fingerprint.It involves smoothing the projected waveform using a Gaussian lowpass filter of size W x W which helps in reducing the effect of noise in the projection. The ridge spacing S(i, j) is then calculated by counting the median number of pixels between the consecutive minima points in the projected waveform.In any fingerprint image, there is a local frequency of the ridges that collectively form the ridge frequency image. The ridge frequency is obtained from the extraction of the ridge map from the image.

1.11 Binarisation

The image obtained from the Gabor filtering stage is binarized and thinned to make it more suitable for feature extraction.The thinning process is described in next section. Most minutiae extraction algorithms operate on basically binary images where there are only two levels of interest: the black pixels represent ridges, and the white pixels represent valleys. Binarisation converts a greylevel image into a binary image. This helps in improving the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae.Binarisation involves examining the grey-level value of every pixel in the enhanced image, and, if the grey-level value is greater than the predefined global threshold, then the pixel value is set to value one; else, it is set to zero. The outcome of binarisatio is a binary image which contains two levels of information, the background valleys and the foreground ridges.

1.12 Thinning

Thinning is a morphological operation which is used to remove selected foreground pixels from the binary images. A standard thinning algorithm from is used, which performs this operation using two subiterations. The algorithm can be accessed by a software MATLAB via the ‘thin’ operation of the bwmorph function. Each subiteration starts by examining the neighborhood of every pixel in the binary image, and on the basis of a particular set of pixel-deletion criteria, it decides whether the pixel can be removed or not. These subiterations goes on until no more pixels can be removed. Thinning is defined as a procedure to transform a digital binary pattern toa connected skeleton of unit width. Two basic implementations available for this approach are sequential and parallel methods.Thinning is the process of reducing the amounts of pixels in an image by removing all redundant pixels and producing simplified image containing minimum number of pixels possible.Actually we need pixels on which algorithms can be implemented easily.When we take fingerprint image of students it contains minutiae patterns which are not required for further implementations.So thinning is a process by which we remove selected minutiae patterns from the binary image.That is the fingerprint image obtained after thinning is perfect for implementing other algorithms for calculating minutiae easily.After binarisation we get image that can be defined in two dimensions but it contains some pixels which are not required further so ,we remove selected pixels from binary images.The thinning algorithm is written in matlab and imported in java codes.

Chapter 2

SYSTEM ANALYSIS

In this part we are going to discuss and analyse about the process for developing process of “Biometric Application System” including software requirement specification(SRS) and comparison between the existing and proposed system. The functional and non functional requirements are included in SRS part to provide complete description and overview of system before the developing process is carried out. Besides the proposed vs. proposed provides a view of how the proposed system will be more efficient than the existing one meet the demand of the people.

2.1 Feasibility Study

Feasibility study aim to objectively and rationally uncover the strengths and weakness of an existing business or proposed venture, opportunities and threats present in the environment, the resources required to carry through, and ultimately the prospect of success. This, project has been designed keeping in mind the resources available and the possibilities, challenges which are likely to take place.Considering the previous attendance system which were maintained on paper was not comfortable and slow process.But this will enhance the overall attendance system in any institutions,or any other workplaces where attendance is primarily required.

2.2 Precision

Precision doesn't means that the measurements are close to the target value-it means that the measurement are close to one another. They may or may not be near the target value. Using this model for attendance it is easy to take fingerprint through scanner and other works like generating attendance report online can be done easily.The total cost required depends upon which kind of computers are used.But once it is inastalled and ready to use makes whole work easy for teachhers and students with complete accuracy.

2.3 Functional Requirements

functional requirement defines a function of a system and its components. A function is described as a set of inputs , the behaviour, and outputs. "Biometric system" require two user as first administrator and second student.Administrator can start the attendance procedure for current semester and there will be time limit of 10 minutes so that no body can use scanner after that.Students can login and can verify their attendance report or can check their absenties according to day.The administrator can be teachet taking the class can update the semester ,change the students details and can easily calculate percentage as desired.An email for attendance report can be sent to students email id so that they can be updated with current attendance reports.

2.4 Existing System

Existing system allows teachers to take attendance on register and calculate percentage manually.Each teacher is responsible for their own subjects and maintaining attendance.

First start of a new semester teachers are required to make new register with names of valid students and their roll number.If new students are coming then their names shoudl be entered in register.Roll no. is provided on the basis of alphabetical orders of name so new students must be given valid roll no.

2.4.1 Disadvantages

Taking attendance on paper is tedious work.teachers have to mark attendance as per their scheduled classes.Since it is done manually there is chance of error as if marking of students attendance goes incorrectly.Suppose if large numbers of students are present then it requires more space on paper and it is time consuming as teacher should spent some time foe taking attendance per class.If teachers need to publish attendance report at last of the semester he has to calculate all their working hours and no. of hours students are present for calculating percentage of attendance.For lager institutions it becomes more difficult and takes too much time to do all this type of work on time.If a single error is encountered then it has to be done again or if a student claims that he is absent or present on any valid class hours then it is very difficult to check the previous data correctly.Overall we can say that the existing system makes the work complicated for large institutions where attendance is taken for large students group.Also it does not provide complete accuracy as it is done manually.Last bust not the least it is more time consuming.

2.5 Proposed System

The main objective of designing of biometric systemis that it is less time consuming and most of the works are done automatically if once software is installed.

Students are registered at the starting of new batch and it will be updated for each semester.The registered students database is stored on server which should be kept safe and a scanner and computer is kept in individual classes.At the time of attendance the respective teacher or administrator update the current batch students table and their registered fingerprint templates .Students give their fingerprint templates and it is matched with the server and if valid then attendance is marked.Since we have made this application for laboratory so each semester there are two sections and each section have two labs in a week.so when lab starts then administrator allows the system to be available for attendance.Students one-by-one gives their fingerprint through scanner present in the lab and after some time administrator closes the application so that no further attendance can be taken.If taken fingerprint image is matched with the existing fingerprint images then attendance is marked with date.

2.5.1 Advantages

This type of model increases accuracy,and decrease manual work.The proposed system consumes less time and it is less chance to alter any tables or databases.In case if altered then backup file is present which is used for doing all works.Students can easily get their attendance report and teachers are not bothered about calculating attendance percentage manually.This project removes all problems of existing systems.Existing system consists of pen-paper work which is very difficult to handle for teachers.So here all works are done automatically and attendance reports are generated online or can be sent to individuals as email.There is very less chance of malfunction so this system provides almost hundred percent accuracy.The overall cost of proposed system is cheap as once installed then we do not need any hardwares.So first we need to buy hardwares like scanner and system for each classroom.The proposed system can be implemented for large conferences or for attendance where large number of students are present.We can say that it is cheap,effective and makes work easily.As further modifications can be implemented so we can enhance this proposed model and can implement foron large scale.

Chapter 3

SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.For the proposed system,system design includes data flow diagrams,which includes many levels.We have also specified different modules of proposed system and their design strategies.Here we have described different tables used for fingerprint recognition system such as administrator table,student table,registration table etc and table contents are specified.Matlab module includes all algorithms used for minutiae extraction and its codes are given in appendix.

3.1 DATA FLOW DIAGRAM(DFD)

A Data Flow Diagram (DFD) is a structured analysis and design tool that can be used for flowcharting. A DFD is a network that describes the flow of data and the processes that change or transform the data throughout a system. This network is constructed by using a set of symbols that do not imply any physical implementation. It has the purpose of clarifying system requirements and identifying major transformations. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. DFD can be considered to an abstraction of the logic of an information-oriented or a process-oriented system flow-chart. For these reasons DFD's are often referred to as logical data flow diagrams.

3.1.1 Level 0

The Top-level diagram is often called a context level diagram. It is also called 0 level data flow diagram. It contains a single process, but it plays a very important role in studying the current system. The context diagram determines the boundaries. Anything that is not inside the process identified in the context diagram will not be part of the system study. The external elements, that are the sources and sink function will not be studied in detail in the context level data flow diagram.
EXTERNAL ENTITY: An external entity is a source or destination of a data flow. Only those entities which originate or receive data are represented on a data flow diagram. The symbol used is a rectangular box.
PROCESS: A process shows a transformation or manipulation of data flow within the system. The symbol used is an oval shape.
DATAFLOW : The data flow shows the flow of information from a source to its destination. Data flow is represented by a line, with arrowheads showing the direction of flow. Information always flows to or from a process and may be written, verbal or electronic. Each data flow may be referenced by the

processes or data stores at its head and tail, or by a description of its contents.

DATA STORE: A data store is a holding place for information within the system: It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.Level 0 DFD is shown in the Fig 3.1.It includes input as students fingerprint and gives output as attendance report.The process part contains students attendance management system which takes student fingerprint as input and processes it and produces attendance report as output.The student attendance system implements many algorithms for minutiae extraction.Students management system is further expressed with different attributes in level 1 DFD.Fig 3.1 shows that students gives their fingerprint image as input and the system extracts minutiae patterns and apply algorithms and produces output as attendance report.

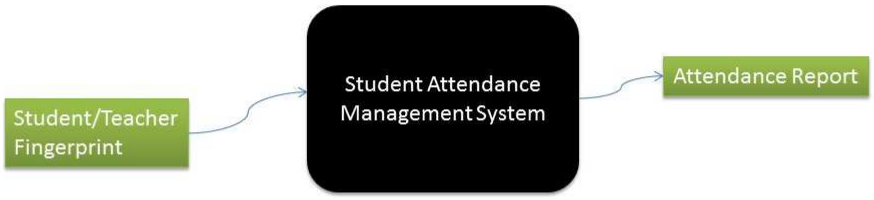
Student/Teacher Fingerprint
Student Attendance Management System
Attendance Report

Figure 3.1: level 0 DFD

3.1.2 Level 1

Fig 3.2 shown below is level 1 DFD which contains students attendance management system and its attributes.Attributes are students attendance report,students identification and marking attendance for current semester laboratory.Attendance report can be updated as per requirement.Student identification is deone by taking his/her fingerprint image through scanner and matching with existing database of current semester.If matched then attendance is marked with date and time.

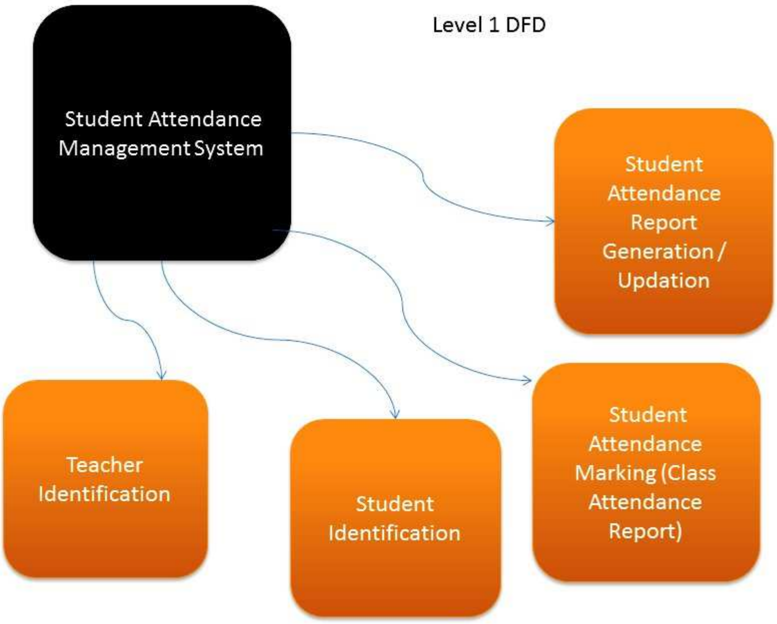
Level 1 DFD
Student Attendance Management System
Student Attendance Report Generation / Updation
Teacher Identification
Student Identification
Student Attendance Marking (Class Attendance Report)

Figure 3.2: level 1 DFD

3.1.3 Level 2

Fig 3.3 shows level 2 DFD which is elaboration of level 1 DFD.Here circles contain processes and rectangles contains inputs and outputs.Here database is also shown.Students identification contains attributes such as students fingerprint.Attendance report contains attributes such as students registration number and it is connected with semester attendance report database.The students attendance marking is connected with class attendance report database.

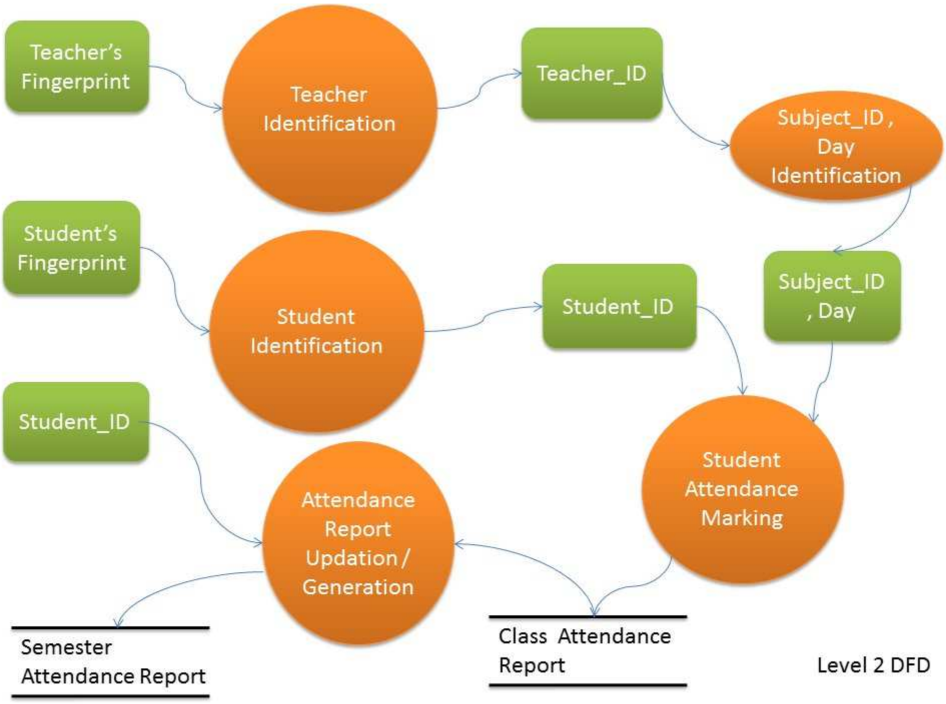

Figure 3.3: level 2 User DFD

3.2 MODULES

This project contains four module, mainly administrator module,students module,fingerprint module and matlab module.Administrator module is only responsible for allowing attendance for current semester.Students module contains information about students such as name,registration number,roll number and email id.Fingerprint module contains fingerprint template and information about gender of students.Matlab module is used for processing fingerprint image that is all required functions for finding minutiae patterns are written in matlab and the it is converted into jar file so that it can be used in java programs.Matlab functions are tested on FCV2002 database for finding minutiae patterns.We have briefly explain each module with its working and functions.The explanations are as follows:

3.2.1 Administrator module

In this module as shown in Fig 3.4(administrator form) administrator can login and star the attendance for current semester,can update semester,or verify the available semester.Administrator only can allows the start time of attendance for students present in lab.Actually we have made this biometric attendance system for laboratory .So when lab time starts first administrator allows system for attendance purpose.He checks for available semester,if not present then update the semester to current semester.If there are any new students then administrator can add to database with his fingerprint template.After the application is enabled fingerprint scanner is passed to students one by one students give their fingerprint and it is matched with present fingerprint templates and if matched then attendance is marked with date.After 10 minutes administrator close the system so that no attendance can be taken after that.Administrator module is given through user interfaces designed in java programming language in NetBeans IDE.The Fig 3.4 shown below allows administrator to start attendance and enter current semester name.When current semester is entered then students are ready to give their fingerprint image.

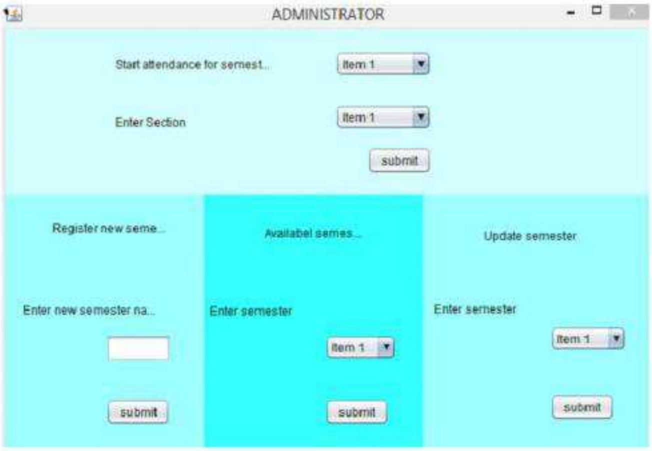

Figure 3.4: Administrator form

3.2.2 Students module

Here students registration is done as shown in Fig 3.5(Registration Form).Students also can login to check their attendance report as desired.Students unique key will be registration number which will be unique through out the course.Students information is stored in this module.Information such as students name,registration number,roll number and email id is stored in database.Email id is stored because if necessary the attendance report can be sent on their email id.Once students are registered so they cannot registered himself for second time.This module is also given through user interfaces designed in java programming language in NetBeans IDE.

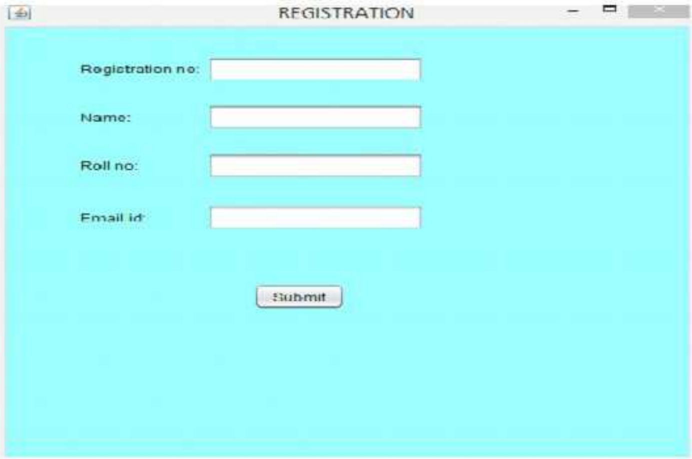

Figure 3.5: Registration form

Fig 3.6 shown below (semester form) provides us facility through which we can check attendance report.Here we can delete semester name if not required.If new student is added in current semester list then can be registered through this form.When we delete a semester if not required then it removes the corresponding table from table list.If new student is added then name of student with registration number is added to current semester list.Semester form allows administrator to add new student in current semester.

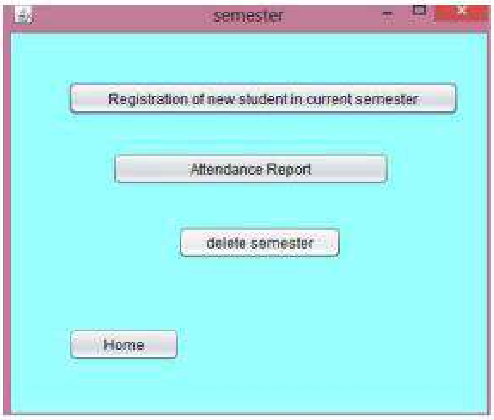

Figure 3.6: Semster form

3.2.3 Fingerprint module

This module gives the fingerprint template to be stored corresponding to students name .Here fingerprint is taken through scanner and using algorithms converted into templates to be stored in database.Generally we store minutiae patterns in database.Minutiae patterns are formed by values of x,y,theta where x is the value in x-axis,y is the value in y-direction and theta is the angle in orientation field.When fingerprint of students is taken through scanner ,if not in gray-level image then converted into gray-level image,because gray-level contains contains details of fingerprint image.Since some noise like dirt,humidity and etc decreases the quality of fingerprint image taken then we need to enhance the image for that algorithm like binarisation,normalization,ridge frequency estimation and thining is applied so that enhanced image contains necessary parts of fingerprint from where minutiae patterns can be easily obtained.All functions are written in matlab module which is explained below.

3.2.4 Matlab module

Here all algorithms to extract minutia is written and compiled in matlab.Functions are written in matlab and deployed as java package to be used in java code.Matlab is a tool for processing image.Fingerprint is taken through scanner and given to algorithms written in matlab for enhancement and finding minutiae patterns.Actually all functions written are required for our java programs.Here only false minutiae is removed and values corresponding actual minutiae patterns are taken.The functions written is tested on FCV2002 database,which contains fingerprint images taken from eight directions,because to ensure that minutiae returns are same for one fingerprint image despite of in which direction it is taken.Since we have to use these algorithms in java programs so we need to convert these functions in jar file.Jar file contains java packages which can be made by using java builder provided by matlab.Once jar file is crated we can easily import these functions in java programs.We generally import that jar file in java programs using 'import' statement.Matlab is an effective tool which provides different methods or functions for finding minutiae patterns are easily written and run successfully.It also provide tools for making packages in different programming languages such as c,c++ or java programming language.In matlab functions are written and saved with .m extension such as enhance.m function written for enhancing fingerprint image by applying different phases of algorithm discussed above.Also functions like extract-finger.m for extracting minutiae patterns.The function main-single.m is the function which is used to take image or fingerprint and find the minutiae patterns successfully.We have given all matlab functions in appendix .Matlab functions include enhance.m,extract.m etc.These all functions are used for extracting minutiae patterns also to remove false minutiae pattern we have written functions in matlab.All functions are saved as .m extensions and when executed then we convert matlab functions to jar file so that it can be used in java codes.We have written java codes in IDE called NetBeans.Al algorithms written are well executed on predefined database.Predefined database contains fingerprint images taken from different angles.

3.3 TABLE DESIGN

The table shown below contains basic information of students.Information about students like name,registration number ,gender,date of birth,email id and fingerprint template is given.Fingerprint template is taken corresponding to individual students registration number.At the time of admission admission number is given which can be taken as unique key for first 2 semester students.Once registration number is given to students and it is unique throughout the course then it is considered unique key.Basic information is stored in a table which once created cannot be altered.This table is called master table .Whenever we need any information about students we can search from this table and only we can append students information in this table.No process is done on this table.When current semester students are updated in another table can be changed or any operation is done on that tables.

3.3.1 basic_info

This table 3.1(basic info table) contains the basic information of the students.Basic information about students such as their name,admission number given at the time of admission,registration number for those who are registered in s3 semester.The fingerprint template is also present in this table.Also type of each attribute is defined as varchar(10),that is we can use either number or alphabets having length not more than 10.Fingerprint template type is defined as array[] type because it contains fingerprint template of so many students.

Attribute	Description	Type	Examples of Values
admission no.	specify students unique key	integer(6)	120291/122093
name	specify student's name	varchar(8)	sandeep
reg.no.	registration number	varchar(10)	12130243
gender	male or female	varchar(5)	male
dob	date of birth	boolean	12/06/1995
email id	email id of students	varchar(20)	kumar@gmail.com
locn	location	varchar(100)	kochi
const	constituency	varchar(100)	delhi
pass	password	varchar(20)	12345
template	fingerprint template	array[]	110001.jpg

Table 3.1: basic info table

3.3.2 other_info

This Table 3.2(other info table) contains the other information of the students.Other information like students home,district,name,city are present in this table.This table is required if some problems happen in getting information about any particular student.Also if necessary ,if teacher wants to communicate with parents of students can search information about his home and can easily communicate.If any problems happen in master table due to any system problem then we can verify students details easily and comfortably.

3.3.3 administrator

This Table 3.3(administrator table) contains the semester information.This table contains information about semester that is how many semesters are available and current semester so that attendance for current semester

Attribute	Description	Type	Examples of Values
attendance percentage	specify percentage of attendance	varchar(100)	92
street	street of the voter in which he belongs	varchar(200)	mg road
city	city of the voter	varchar(50)	samastipur
district	district of the voter	varchar(100)	patna
state	state of the voter	varchar(100)	bihar
pincode	pincode	varchar(6)	823453
landmark	nearest place for identification	varchar(100)	cusat
first_name	first name	varchar(100)	sandeep
middle_name	middle name	varchar(50)	kumar
last_name	last name	varchar(100)	suman

Table 3.2: other info table

students can be taken easily.This table also contains branch name like computer science or civil engineering so that attendance report can be published for individuals branch.We can divide students in groups because only twenty five students are allowed to enter in lab at one time.Here if there is need to delete semester then we can delete or if we need to update semester list then we can update the semester name but these all works are done by administrator.

Attribute	Description	Type	Examples of Values
semester name	semester name	varchar(50)	s6
available semester	available semesters	varchar	s3/s4
branch-name	name of branch	varchar	eee/cse

Table 3.3: administrator table

3.3.4 registration table

This is temporary table used for students registration.Here this Table 3.4(registration table) contains attributes like name, registration number,roll number and gender.When we need to store fingerprint template corresponding to individual students name then we need to register students.For registration we require name ,registration number etc.Registration number is unique so it can be considered as unique key.If new students are present in current semester the we need to register that student for that we require all these information present in this table.

Attribute	Description	Type	Examples of Values
reg.no.	registration number of student	varchar(10)	12130243
name	specify soldier's service no.	char(10)	vikash
roll no.	roll number of student	varchar(5)	83
gender	male or female	varchar(5)	male

Table 3.4: registration table

3.3.5 fingerprint template

This Table 3.5(template table) is used to store fingerprint template(minutia values).Actually fingerprint image taken from scanner is not stored in database but we store minutiae patterns.Minutiae patterns are found by using algorithms ,here which is written in matlab and combined as java packages.This table contains fingerprint template and gender because fingerprint image for male or female differs in ridge density.As we are able to find minutiae patterns according to gender(male/female)then we can enhance searching techniques.Also we can improve search time.Minutiae patterns are stored as array file because it contains different values.

Attribute	Description	Type	Examples of Values
template	fingerprint template	array[]	1.000.000
gender	gender of students	varchar(10)	male/female

Table 3.5: template table

3.3.6 Attendance

This Table 3.6(attendance table) is used to show attendance of students.This table contains attributes like name,registration number,date and attendance percentage.This table is used to store information about the students attendance according to date so that we can easily evaluate attendance percentage and can produce attendance report easily whenever required.This table can be updated whenever class is taken and updating is done according to date.If students is present then we mark attendance as 'yes' and if absent the mark it as 'no'.This table contains attendance percentage which is calculated by considering number of hours the students is present out of total number of hours lab is taken.Here it is calculated automatically so it reduces manual work or we can say that pen-paper work is eliminated.

3.4 Input and Output Design

for any system we require inputs for processing which gives output.If input is correct and system programs are also correct then we can get cor-

Attribute	Description	Type	Examples of Values
name	name of students	varchar(10)	sujeet
date	attendance taken on which date	boolean	12/03/2013
attendance percentage	attendance in percentage	varchar(100)	95
reg.no.	registration number	varchar(10)	12130057

Table 3.6: attendance table

rect outputs.Input and output are main parts of work for any designed system.Here for biometric attendance system input is the fingerprint image and output is the attendance that present or absent.Input is taken through scanner and it given to programs where it is processed and minutiae patterns are calculated and matched to minutiae patterns stored in database.If matched then attendance is marked as 'yes' otherwise 'no'.When input is taken then we have used user interfaces so that for current semester students fingerprint images are taken.We have explained input and output design briefly as given below.

3.4.1 Input Design

Input design is the link that ties the information system into the world of its users. The input design involves determining the inputs, validating the data, minimizing the data entry and provides a multi-user facility. Inaccurate inputs are the most common cause of errors in data processing. Errors entered by the data entry operators can be controlled by input design. The user-originated inputs are converted to a computer based format in the input design. Input data are collected and organized into groups of similar data. Once identified, the appropriate input media are selected for processing. All the input data are validated and if any data violates any conditions, the user is warned by a message. If the data satisfies all the conditions, it is transferred to the appropriate tables in the database. In this project the student details are to be entered at the time of registration. A page is designed for this purpose which is user friendly and easy to use. The design is done such that users get appropriate messages when exceptions occur.

3.4.2 Output design

Computer output is the most important and direct source of information to the user. Output design is a very important phase since the output needs to be in an efficient manner. Efficient and intelligible output design improves the system relationship with the user and helps in decision making. Allowing the user to view the sample screen is important because the user is the ultimate judge of the quality of output. The output module of this system is the selected notifications.Outputs are the only way through which we can say that the system or application is working properly or not.Since user only gives the input that's why he is the perfect person to validate the output.In biometric attendance system when fingerprint template is matched then attendance is marked which shows that the given fingerprint image as input is valid and system is working properly.We have used user interfaces through which inputs for specific student is taken and his attendance is marked.Output design for this application generally refers to the results and information that are generated by the system for many end-users;output is the main reason for developing the system and the basis on which they evaluate the usefulness of the application.

Chapter 4

SOFTWARE TOOLS

A programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs, that can be combined together to accomplish a task, much as one might use multiple hand tools to fix a physical object. The ability to use a variety of tools productively is one hallmark of a skilled software engineer.The most basic tools are a source code editor and a compiler or interpreter, which are used ubiquitously and continuously. Other tools are used more or less depending on the language, development methodology, and individual engineer, and are often used for a discrete task, like a debugger or profiler. Tools may be discrete programs, executed separately – often from the command line – or may be parts of a single large program, called an integrated development environment (IDE). In many cases, particularly for simpler use, simple ad hoc techniques are used instead of a tool, such as print debugging instead of using a debugger, manual timing (of overall program or section of code) instead of profiler, or tracking bugs in a text file or spreadsheet instead of a bug tracking system.Here we have used IDE as NetBeans and programming language java.For processing fingerprint image we have used matlab tool.The brief descriptions are given below.

4.1 JAVA

Java is one of the most job seeking programming language in the recent I.T industry . Object Oriented Programming (OOP) is an approach to program organization and development, which attempts to eliminate some of the pitfalls of conventional programming methods by incorporating the best of structured programming features with sev eral new concepts. It is a new way of organizing and dev eloping programs and has nothing to do with any particular language.

4.2 JAVA FEATURES

Compiled and Interpredted : Mostly a computer programming language is either compiled or interpreted. Java combines both these approaches thus making Java a two-stage system. First, Java compiler translates source code into what is known as bytecode instructions. Bytecodes are not machine instruction and therefore in the second stage, Java interpreter generates machine code that can be directly executed by the machine that is running the java program.

Platform -Independent and Portable : Java programs can be easily moved from one computer system to another, anywhere and anytime. changes and upgrades in operating systems, processors and system resources will not force any changes in java programs.

Java ensures portability in two ways. First, Java compiler generates bytecode instruction that can be implemented on any machine. Secondly , the size of the primitive data types are machine independent.

Object-Oriented: Java is a true object-oriented language. Almost everything in java is an object. All program code and data reside within objects and classes. Java comes with an extensive set of classes arranged in packages.

Robust and secure: Java is a robust language. It provides many safeguards to ensure reliable code. It has strict compile time and run time checking for data types. It is designed as a garbage-collected language relieving the programmers virtually all memory management problems. Java also incorporates the concepts of exception handling which captures series errors and eliminates any risk of crashing the system.

Java systems not only verify all memory access but also ensure that no viruses are communicated with an applet. The absence of pointers in java ensures that programs cannot gain access to memory locations without proper authorization.

Distributed: Java is designed as a distributed language for creating applications on networks. It has the ability to share both data and programs. Java applications can open and access remote objects on Internet as easily as they can do in local system.

This enables multiple programmers at multiple remote locations to collaborate and work together on a single project.

Multithreaded : Multithreaded means handling multiple tasks simultaneously . Java supports multithreaded programming. This means that we not wait for the application to finish one task before beginning another.

4.3 NetBEANS IDE

What is integrated development environment (IDE)?

An integrated development environment (IDE) is a programming environment that has been packaged as an application program, typically consisting of a code editor, a compiler, a debugger, and a graphical user interface (GUI) builder. The IDE may be a standalone application or may be included as part of one or more existing and compatible applications. The BASIC programming language, for example, can be used within Microsoft Office applications, which makes it possible to write a WordBasic program within the Microsoft Word application. IDEs provide a user-friendly framework for many modern programming languages, such as Visual Basic, and Java.IDEs for developing NetBeans and HTML applications are among the most commonly used.

NetBeans IDE :

The NetBeans IDE is written in Java and can run on Windows, OS X, Linux, Solaris and other platforms supporting a compatible JVM. NetBeans is an integrated development environment (IDE) for developing primarily with Java, but also with other languages, in particular PHP, C/C++, and HTML. It is also an application platform framework for Java desktop applications and others.

NetBeans Platform :

Framework for simplifying the development of Java Swing desktop applications. The NetBeans IDE bundle for Java SE contains what is needed to start developing NetBeans plugins and NetBeans Platform based applications; no additional SDK is required.

Applications can install modules dynamically. Any application can include the Update Center module to allow users of the application to download digitally signed upgrades and new features directly into the running application. Reinstalling an upgrade or a new release does not force users to download the entire application again.

The platform offers reusable services common to desktop applications, allowing developers to focus on the logic specific to their application. Among the features of the platform are:
∗User interface management (e.g. menus and toolbars)
∗ User settings management
∗Storage management (saving and loading any kind of data)
∗ Window management
∗Wizard framework (supports step-by-step dialogs)
∗ NetBeans Visual Library
∗Integrated development tools

NetBeans IDE – Base IDE Features

NetBeans IDE is a modular developer tool for a wide range of application development technologies. The base IDE includes an advanced multi-language editor, Debugger and Profiler, as well as tools for versioning control and developer

Templates and Samples Applications

NetBeans IDE gives you skeleton applications in the form of project templates for all the technologies it supports. In addition, it provides a set of sample applications, some of which can be recreated step by step by following a related tutorial available on NetBeans.org.

The IDE provides project templates and sample projects that help you create Java SE applications, Java EE applications, Java ME applications, HTML5 applications, NetBeans Platform applications, PHP application, and C/C++ applications.

Databases and Services

The Services window gives you access to many ancillary resources, such as databases, servers, web services, and issue trackers.

You can start and stop databases and servers directly in the IDE. When working with databases, you can add, remove, and modify your data in the IDE. When you have deployed an application to a server, you can manage your deployed resources because they are displayed in the Servers node.

PluginManager

While using the IDE, you can always go to the Plugin Manager from the Tools menu to add, remove, or update the installed set of features.

A wide variety of plugins are available for all types of development, from Java SE, Java EE, Java ME, HTML5, Groovy, and PHP to C/C++ development. Community contributed plugins are also available in the NetBeans Plugin Portal.

4.4 MySQL

MySQL is the world's second most widely used open-source relational database management system (RDBMS). The SQL phrase stands for Structured Query Language. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for- profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation. MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP (Linux, Apache, MySQL, Perl/PHP/Python) open source web application software stack (and other 'AMP' stacks). Free-software-open source projects that require a full- featured database management system often use MySQL. For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, MODx, Joomla, WordPress, phpBB, Drupal and other software. MySQL is also used in many high-profile, large-scale websites, including Wikipedia, Google(though not for searches), Facebook, Twitter, Flickr, and YouTube.

4.5 MATLAB

Matlab is basically a high level language which has many specialized toolboxes for making things easier for us.Here we have used Matlab for processing fingerprint image.Matlab provides an efficient tools for processing fingerprint images and finding minutiae.All functions used for extracting minutiae is written in Matlab and it is tested on FCV2002 database to find minutiae patterns.Functions like extraxt-finger.m,enhance.m is written in Matlab.These functions implement algorithms to enhance fingerprint image(generally grey-level image) using techniques such as segmentation , normalizations,Binarisation and thining.After tested successfully on FCV2002 database we have created jar file which can be imported in java programs to use all these features for finding minutiae patterns.Jar file is an executable file which contains java package.MATLAB is a high performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include: Math and computation.MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar noninteractive language such as C or Fortran.

4.6 Operation Environment

In computer software, an operating environment or integrated applications environment is the environment in which users run application software. The environment consists of a user interface provided by an applications manager and usually an application programming interface (API) to the applications manager.The environment in which users run programs. For example, the DOS environment consists of all the DOS commands available to users. The Macintosh environment, on the other hand, is a graphical user interface that uses icons and menus instead of commands.Here to run biometric attendance application system we require operating system like Linux,Windows Vista/7/8 and processor intel core processor with memory minimum 1 GB and hardware disk space upto 20 GB for database and use for future.We have used MySql for creating database.The operation environment is shown below in the Table 4.1:

PROCESSOR	INTEL CORE PROCESSOR OR BETTER PERFOMANCE
OPERATING SYSTEM	WINDOWS VISTA/7/8 LINUX
MEMORY	1 GB OR MORE
HARDWARE DISK SPACE	MINIMUM 20 GB FOR DATABASE USAGE FOR FUTURE
DATABASE	MYSQL

Table 4.1: operation environment

Chapter 5

SYSTEM IMPLEMENTATION

Implementation is the stage in the project where the theoretical design is turned into a working system and is giving confidence on the new system for the users that it will work efficiently and effectively. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the change over, an evaluation of change over methods. Apart from planning major task of preparing the implementation are education and training of users. The implementation process begins with preparing a plan for the implementation of the system. According to this plan, the activities are to be carried out, discussions made regarding the equipment and resources and the additional equipment has to be acquired to implement the new system.

Chapter 6

RESULT

The outcomes obtained from this project are very useful for our future as we have learnt how to process images in matlab and making of user interfaces in NetBeans IDE successfully.A scientific approach was developed during this project which helps us in developing skills and self confidence in coding and working in softwares like Matlab.We have tried to build an application system for laboratory so that easily application can be taken and attendance report can be published as required.Various new thing like minutiae is studied successfully and algorithms for finding minutiae patterns are considered.Algorithms like gender estimation was invented in this project.This project is only applied for laboratory but we extend this project for whole university.We have made user interfaces using java programming language so we learn many new things to be implemented in java programs.We have used XAMP software for server side and database problems.A better skill is developed to work in NetBeans and Matlab.This project was very useful because we tried to interface fingerprint scanner with our java programs.We have also learnt how to work with Linux operating system.

Chapter 7

Future Work

Regarding our fingerprint identification system, we will planning to introduce more indexing techniques like ridge density tolerance etc. for making the search more faster. Also the key used will be more efficient when complex key will be used. We will trying to reduce matching error rates. Student attendance system is designed using LAN in this project. We have thought of using wireless LAN. The problem of using wireless LAN is that wireless devices are costly and implementation is hard because the wireless devices work in small area. Our institute CUSAT is spread over large area so we may not use wireless network now. As an alternate, we may use mobile network which would be suciently suitable because now-a-days 3G network provides much speed. It would meet necessary throughput and data fetching speed requirements.We have only tried this application for laboratory but if required we can extend for whole subjects and for whole departments.If necessary we can connect this application system to main server of university so that attendance report can be updated on university website as per requirement.

Chapter 8

CONCLUSION

This project mainly comprised of development of attendance management system and fingerprint identification system. Attendance management is very helpful in saving valuable time of students and teachers, paper and generating report at required time.This project presented a framework using which attendance management can be made automated and on-line. A general implementable approach to attendance management was proposed using LAN. Further, an idea for using portable devices alongwith wireless LAN or mobile 3G network was suggested.Fingerprint Identification System used for student identification is faster in imple mentation than any other fingerprint identi cation systems. For fingerprint recognition, prevalent enhancement techniques like minutiae extraction using Crossing Number concept followed by spurious and boundary minutiae removal,fingerprint classification, reference point detection, etc. are employed. Also, var- ious new concepts are invented in this fingerprint identification system like gender estimation and key based one to many matching. Fingerprint classification and gender estimation are employed to partition the database.

References

[1] Samuel King Opoku, March 2013,“An automated biometric attendance management system,” International Journal of Computer Science and Mobile Computing., Vol. 2, Issue. 3, pp.18 – 25

[2] By L. Hong, Y.Wan , and Anil K. Jain. Image Enhancement: Algorithm and performance algorithm". IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8):777 789, May 1998

[3] By L. Hong. Personal Identification Using Fingerprints". PhD thesis, Michigan State University, 1998.

[4] Lynn Beighley, Michael Morrison ’Head First PHP MySQL’ O’Reilly publications.

[5] Rajib Mall, ’Fundamentals of software engineering’, Third edition, PHI Learning private ltd.

[6] Raymond Thai. Image Enhancement and Minutiae Extraction".Technical report, The University of Western Australia.

[7] N.K. Ratha, K. Karu, S. Chen, and A. K. Jain. Real-Time Matching System for Large Fingerprint Database". IEEE Trans. PAMI, 18(8):799-813, 1996.

APPENDIX-A

MATLAB FUNCTIONS

```
% Usage:  [ ret ] = ext_finger( img, display_flag );
%
% Argument:   img - FingerPrint Image
%             display_flag
%
% Returns:    ret - Minutiae

function [ ret ] = ext_finger( img, display_flag )
if nargin==1; display_flag=0; end
   block_size_c = 24; YA=0; YB=0; XA=0; XB=0;
% Enhancement ----------------------------------------------------------
if display_flag==1; fprintf(' >>> enhancement '); end
yt=1; xl=1; yb=size(img,2); xr=size(img,1);
[ binim, mask, cimg, cimg2, orient_img, orient_img_m ] = f_enhance(img);
% Making Mask ----------------------------------------------------------
if display_flag==1; fprintf('done.\n >>> making mask '); end
   mask_t=mask;
for y=19:size(mask,1)-block_size_c*2
for x=block_size_c:size(mask,2)-block_size_c*2
       n_mask = 0;
for yy=-1:1
for xx=-1:1
             y_t = y + yy *block_size_c;
             x_t = x + xx *block_size_c;
if y_t > 0 && x_t > 0 && (y_t ~= y || x_t ~= x) && mask(y_t,x_t) == 0
               n_mask = n_mask + 1;
end
end
end
if n_mask == 0
continue
end
if mask(y,x) == 0 || y > size(mask,1) - 20  ||  y < yt || y > yb || x < xl || x > xr
cimg2(ceil(y/(block_size_c)), ceil(x/(block_size_c))) = 255;
           mask_t(y,x) = 0;
continue;
end
for i = y:y+1
for j = x-9:x+9
if i > 0 && j > 0 && i < size(mask,1) && j < size(mask,2) && mask(i,j) > 0
else
cimg2(ceil(y/(block_size_c)), ceil(x/(block_size_c))) = 255;
            mask_t(y,x)=0;
break
```

```
end
end
end
end
end
mask=mask_t;
    inv_binim = (binim == 0);
thinned =  bwmorph(inv_binim, 'thin',Inf);
    mask_t=mask;
if numel(find(mask(125:150,150:250)>0)) > 0 && numel(find(mask(250:275,150:250)>0)) > 0
mask(150:250,150:250)=1;
end
method=-1; core_y = 0; core_x = 0; core_val=0; lc=0;
    o_img=sin(orient_img); o_img(mask == 0) = 1;

    lower_t=0.1;
    [v,y]=min(cimg);
    [dt1,x]=min(v);
    delta1_y=y(x)*block_size_c/2; delta1_x=x*block_size_c/2;

v(x)=255; v(x+1)=255;
    [dt2,x]=min(v);
    delta2_y=y(x)*block_size_c/2; delta2_x=x*block_size_c/2;

v(x)=255; v(x+1)=255;
    [dt3,x]=min(v);
    delta3_y=y(x)*block_size_c/2; delta3_x=x*block_size_c/2;

db=60;
if dt1 < 1 && delta1_y+db < core_y && delta1_y > 15 || dt2 < 1 && delta2_y+db < core_y &&
delta2_y > 15 || dt3 < 1 && delta3_y+db < core_y && delta3_y > 15
        core_val=255;
end
for y=10:size(o_img,1)-10
for x=10:size(o_img,2)-10
            s1=0; t=10; %few of bad cores here
if y < 50 && x > 250
               t=11;
end
if y > 38
yt=20;
else
yt=5;
end
if lc > 0.41 && (core_y + 60 < y)
break;
```

```
end
if mask(y,x)==0 || mask(max(y-t,1),x)==0 || mask(y,min(x+t, size(o_img,2)))==0 || mask(y,max(x-t,1))==0 || mask(max(y-t,1),min(x+t,size(o_img,2)))==0 || mask(max(y-t,1),max(x-t,1))==0 || o_img(y,x) < lc || o_img(y,x) < 0.1
continue
end
if dt1 < 1 && delta1_y+db < y && delta1_y > 15 || dt2 < 1 && delta2_y+db < y && delta2_y > 15 || dt3 < 1 && delta3_y+db < y && delta3_y > 15
continue
end
        test_m=min(o_img(1:y-yt,max((x-10),1):min(x+10,size(o_img,2)) ));
if numel(test_m)>0 && min(test_m) >= 0.17
continue
end
for a=y:y+2
for b=x:x+1
             s1=s1+o_img(a,b);
end
end
        s1=s1/6; s2=[]; i=1;
for a=y-3:y-1
for b=x:x+1
s2(i)=o_img(a,b);
             i=i+1;
end
end
if min(s2) < lower_t
          s2=sum(s2)/6;
else
          s2=s1;
end
        s3=[]; i=1;
for a=y:y+2
for b=x+2:x+3
s3(i)=o_img(a,b);
             i=i+1;
end
end
if min(s3) < lower_t
          s3=sum(s3)/6;
else
          s3=s1;
end
        s4=[]; i=1;
for a=y:y+2
for b=x-2:x-1
```

```
s4(i)=o_img(a,b);
             i=i+1;
end
end
if min(s4) < lower_t
          s4=sum(s4)/6;
else
          s4=s1;
end
        s5=[];
        i=1;
for a=y-3:y-1
for b=x-2:x-1
s5(i)=o_img(a,b);
             i=i+1;
end
end
if min(s5) < lower_t
          s5=sum(s5)/6;
else
         s5=s1;
end
        s6=[]; i=1;
for a=y-3:y-1
for b=x+2:x+3
s6(i)=o_img(a,b);
             i=i+1;
end
end
if min(s6) < lower_t
         s6=sum(s6)/6;
else
         s6=s1;
end
if s1-s2 > core_val
          core_val=s1-s2;
          core_x=x;
          core_y=y;
lc=o_img(y,x);
method=1;
end
if s1-s3 > core_val
          core_val=s1-s3;
          core_x=x;
          core_y=y;
lc=o_img(y,x);
```

```
method=2;
end
if x < 300 && s1-s4 > core_val
            core_val=s1-s4;
            core_x=x;
            core_y=y;
lc=o_img(y,x);
method=3;
end
if x < 300 && s1-s5 > core_val
            core_val=s1-s5;
            core_x=x;
            core_y=y;
lc=o_img(y,x);
method=4;
end
if s1-s6 > core_val
            core_val=s1-s6;
            core_x=x;
            core_y=y;
lc=o_img(y,x);
method=5;
end
end
end
if core_y > 37
yt=20;
else
yt=5;
end
   test_smooth = 100;
if core_y > 0
      test_smooth= sum(sum(o_img(core_y-yt-5:core_y-yt+5,core_x-5:core_x+5)));
end
if lc > 0.41 && (test_smooth < 109.5 && method~=2 || test_smooth < 100) %&& min(min(o_img(1:core_y-yt,core_x-10:core_x+10))) < 0.17
      start_t=0;
      core_val=1/(core_val+1);
else
      core_x=0;
      core_y=0;
      core_val = 255;
end
mask=mask_t; path_len = 45;

% Finding Minutiae -------------------------------------------------------
```

```
if display_flag==1; fprintf('done.\n >>> finding minutiae '); end
    minu_count = 1;
minutiae(minu_count, :) = [0,0,0,0,0,1];
    min_path_index = [];
% loop through image and find minutiae, ignore certain pixels for border
for y=20:size(img,1)-14
for x=21:size(img,2)-21
if (thinned(y, x) == 1) % only continue if pixel is white
% calculate CN from Raymond Thai
                CN = 0; sx=0; sy=0;
for i = 1:8
                  t1 = p(thinned, x, y, i);
                  t2 = p(thinned, x, y, i+1);
                  CN = CN + abs (t1-t2);
end
                CN = CN / 2;
if ((CN == 1) || (CN == 3)) %&& mask(y,x) > 0
skip=0;
for i = y-5:y+5
for j = x-5:x+5
if i>0 && j>0 && mask(i,j) == 0
skip=1;
end
end
end
if skip == 1
continue;
end
                   t_a=[];
                   c = 0;
for e=y-1:y+1
for f=x-1:x+1
                           c = c + 1;
                           t_a(c) = orient_img_m(e,f);
end
end
                   m_o = median(t_a); m_f = 0;
if CN == 3
                      [CN, prog, sx, sy,ang]=test_bifurcation(thinned, x,y, m_o, core_x, core_y);
if prog < 3
continue
end
if ang < pi
                         m_o = mod(m_o+pi,2*pi);
end
else
```

```
progress=0;
xx=x; yy=y; pao=-1; pos=0;
while progress < 15 && xx > 1 && yy > 1 && yy<size(img,1) && xx<size(img,2) && pos > -1
pos=-1;
for g = 1:8
                    [ta, xa, ya] = p(thinned, xx, yy, g);
                    [tb, xb, yb] = p(thinned, xx, yy, g+1);
if (ta > tb)  && pos==-1 && g ~= pao
pos=ta;
if g < 5
pao = 4 + g;
else
pao = mod(4 + g, 9) + 1;
end
xx=xa; yy=ya;
end
end
progress=progress+1;
end
if progress < 10
continue
end
if mod(atan2(y-yy,xx-x), 2*pi) > pi
                m_o=m_o+pi;
end
end
minutiae(minu_count, :) = [ x, y, CN, m_o, m_f, 1];
            min_path_index(minu_count, :) = [sx sy];
            minu_count = minu_count + 1;
end
end% if pixel white
end% for y
end% for x

% Filtering False Minutiae -----------------------------------------------
if display_flag==1; fprintf('done.\n >>> filtering false minutiae '); end
   minu_count = minu_count -1;
   t_minutiae = [];
   t_minu_count = 1;
   t_mpi = [];
for i=1:minu_count
      X = minutiae(i,1); Y = minutiae(i,2);
rc=0;
for y=max(Y-2,1):min(Y+2, size(binim,1))
if rc > 0
break
```

```
end
for x=max(X-2,1):min(X+2, size(binim,2))
if mask(y,x) == 0
rc = rc + 1;
break
end
end
end
if rc > 0
continue;
else
            t_minutiae(t_minu_count, :) = minutiae(i, :);
            t_mpi(t_minu_count, :) = min_path_index(i, :);
            t_minu_count = t_minu_count + 1;
end
end
minutiae = t_minutiae;
    min_path_index = t_mpi;
    minu_count = size(minutiae,1);
    t_minu_count = 1; t_minutiae = [];
    dist_m = dist2(minutiae(:,1:2), minutiae(:,1:2));
    dist_test=49;
for i=1:minu_count
      reject_flag = 0;
      P_x = minutiae(i,1); P_y = minutiae(i,2);
for j = i + 1 : minu_count
if dist_m(i,j) <= dist_test
           reject_flag = 1;
end
end
if reject_flag == 0 && mask(P_y, P_x) > 0
         reverse_p = 0;
if min_path_index(i,1) == 0
           x = P_x;
           y = P_y;
else
           x =  min_path_index(i,1);
           y =  min_path_index(i,2);
end
         p1x=P_x; p1y=P_y;
x1=x; y1=y;
iter = 0;
for m=1:path_len
iter = iter + 1;
cn = 0;
for ii = 1:8
```

```
            t1 = p(thinned, x1, y1, ii);
            t2 = p(thinned, x1, y1, ii+1);
cn = cn + abs (t1-t2);
end
cn = cn / 2;
if cn ~= 3 && cn ~= 4 || m == 1
for n=1:8
if reverse_p == 0 || iter > 1
                [ta, xa, ya] = p(thinned, x1, y1, n);
else
                [ta, xa, ya] = p(thinned, x1, y1, 9-n);
end
if ta == 1 && (xa ~= p1x || ya ~= p1y) && (xa ~= x || ya ~= y)
                p1x = x1; p1y = y1;
x1 = xa; y1 = ya;
break;
end
end
end
end
      t_minutiae(t_minu_count, :) = minutiae(i, :);
      t_minu_count = t_minu_count + 1;
end
end
minutiae = t_minutiae;
   minu_count = t_minu_count-1;
   tmpvec1 = size(img,1).*ones(minu_count,1);
   tmpvec2 = ones(minu_count,1);
   minutiae_for_sc = [minutiae(:,1)/size(img,2) (tmpvec1 - minutiae(:,2) +
tmpvec2)/size(img,1)];
   dist_m = sqrt(dist2(minutiae_for_sc(:,1:2), minutiae_for_sc(:,1:2)));
for i=1:minu_count
    [d,ind] = sort(dist_m(i,:));
for j = 1 : minu_count
if  dist_m(i,ind(j)) == 0
continue
end
     theta_t = mod(atan2(minutiae(i,2) - minutiae(ind(j),2), minutiae(i,1) - minutiae(ind(j),1)),
2*pi);
     ridge_count = 0;
     p_y = minutiae(i,2);  p_x = minutiae(i,1);
     t_x = 0; t_y = 0;
current=1; radius = 1;
while p_y ~= minutiae(ind(j),2)
if thinned(p_y, p_x) > 0  && current == 0 && (t_x ~= p_x || t_y ~= p_y)
current = 1;
```

```
            ridge_count = ridge_count + 1;
else
if thinned(p_y, p_x) == 0
current = 0;
end
end
         t_x = p_x; t_y = p_y;
         p_x = round(minutiae(i,1) - radius*cos(theta_t));
         p_y = round(minutiae(i,2) - radius*sin(theta_t));
radius = radius + 1;
end
end
end
if core_val < 1
minutiae(minu_count+1, :) = [core_x, core_y, 5, start_t, 0,1];
    minu_count = minu_count + 1;
end
if dt1 < 1
minutiae(minu_count+1, :) = [delta1_x, delta1_y, 7, 0, 1,1];
     minu_count = minu_count + 1;
end
if dt2 < 1
minutiae(minu_count+1, :) = [delta2_x, delta2_y, 7, 0, 1,1];
     minu_count = minu_count + 1;
end
if dt3 < 1
minutiae(minu_count+1, :) = [delta3_x, delta3_y, 7, 0, 1,1];
     minu_count = minu_count + 1;
end

% Return Minutiae -------------------------------------------------------
if display_flag == 1
fprintf('done.\n');
      minutiae_img = uint8(zeros(size(img, 1),size(img, 2), 3));
for i=1:minu_count
x1 = minutiae(i, 1); y1 = minutiae(i, 2);
if minutiae(i, 3) == 1                    %Termination
if minutiae(i, 4) > pi
for k = y1-2: y1 + 2
for l = x1-2: x1 + 2
            minutiae_img(k, l,:) = [255, 0, 0];
end
end
else
for k = y1-2: y1 + 2
for l = x1-2: x1 + 2
```

```
        minutiae_img(k, l,:) = [205, 100, 100];
end
end
end
elseif minutiae(i, 3) == 2
for k = y1-2: y1 + 2
for l = x1-2: x1 + 2
        minutiae_img(k, l,:) = [255, 0, 255];
end
end
elseif minutiae(i, 3) == 3          %Bifurcation
if minutiae(i, 4) > pi
for k = y1-2: y1 + 2
for l = x1-2: x1 + 2
        minutiae_img(k, l,:) = [0, 0, 255];
end
end
else
for k = y1-2: y1 + 2
for l = x1-2: x1 + 2
        minutiae_img(k, l,:) = [255, 0, 255];
end
end
end
elseif minutiae(i, 3) == 5
for k = y1-4: y1 + 4
for l = x1-4: x1 + 4
        minutiae_img(k, l,:) = [0, 255, 0];
end
end
elseif minutiae(i, 3) > 5
for k = y1-2: y1 + 2
for l = x1-2: x1 + 2
         minutiae_img(k, l,:) = [128, 128, 0]; % gold for delta
end
end
end
end
combined = uint8(minutiae_img);
for x=1:size(binim,2)
for y=1:size(binim,1)
if mask(y,x) == 0
combined(y,x,:) = [0,0,0];
continue
end
if (thinned(y,x)) % binim(y,x))
```

```
combined(y,x,:) = [255,255,255];
else
combined(y,x,:) = [0,0,0];
end% end if
if ((minutiae_img(y,x,3) ~= 0) || (minutiae_img(y,x,1) ~= 0) ) || (minutiae_img(y,x,2) ~= 0)
combined(y,x,:) = minutiae_img(y,x,:);
end
end% end for y
end% end for x
if core_val < 1 && YA > 0
for k = YA-2: YA + 2
for l = XA-2: XA + 2
combined(k,l,:) = [20, 255, 250];
end
end
for k = YB-2: YB + 2
for l = XB-2: XB + 2
combined(k,l,:) = [20, 255, 250];
end
end
end
end
ret=minutiae;
end
```

```
% Argument:   img - FingerPrint Image
%
% Returns:    binim   - binary image
%             mask    - binary mask
%             cimg1,2 - coherence image
%             oimg1,2 - The orientation image in radians.
function [ binim, mask, cimg1, cimg2, oimg1, oimg2 ] = f_enhance( img )
enhimg =  fft_enhance_cubs(img,6);             % Enhance with Blocks 6x6
enhimg =  fft_enhance_cubs(enhimg,12);
   [enhimg,cimg2] =  fft_enhance_cubs(enhimg,24);
blksze = 5;   thresh = 0.085;
normim = ridgesegment(enhimg, blksze, thresh);
   oimg1 = ridgeorient(normim, 1, 3, 3);

   [enhimg,cimg1] =  fft_enhance_cubs(img, -1);
   [normim, mask] = ridgesegment(enhimg, blksze, thresh);
   oimg2 = ridgeorient(normim, 1, 3, 3);
   [freq, medfreq] = ridgefreq(normim, mask, oimg2, 32, 5, 5, 15);
binim = ridgefilter(normim, oimg2, medfreq.*mask, 0.5, 0.5, 1) > 0;
```

```
end
%oimg -  [OUT] block orientation image(can be viewed using
%        view_orientation_image.m)
%fimg  - [OUT] block frequency image(indicates ridge spacing)
%bwimg - [OUT] shows angular bandwidth image(filter bandwidth adapts near the
%        singular points)
%eimg  - [OUT] energy image. Indicates the 'ridgeness' of a block (can be
%        used for fingerprint segmentation)
%enhimg- [OUT] enhanced image
%img   - [IN]  input fingerprint image (HAS to be of DOUBLE type)

function [enhimg, cimg, oimg,fimg,bwimg,eimg] =  fft_enhance_cubs(img, BLKSZ)
global NFFT;

if BLKSZ > 0
    NFFT       =   32;     %size of FFT
    OVRLP      =   2;      %size of overlap
    ALPHA      =   0.5;    %root filtering
    RMIN       =   4;%%3;      %min allowable ridge spacing
    RMAX       =   40;     %maximum allowable ridge spacing
    ESTRETCH   =   20;     %for contrast enhancement
    ETHRESH    =   19;      %threshold for the energy
else
    NFFT       =   32;     %size of FFT
    BLKSZ      =   12;     %size of the block
    OVRLP      =   6;      %size of overlap
    ALPHA      =   0.5;    %root filtering
    RMIN       =   3;      %min allowable ridge spacing
    RMAX       =   18;     %maximum allowable ridge spacing
    ESTRETCH   =   20;     %for contrast enhancement
    ETHRESH    =   6;      %threshold for the energy
end

  [nHt,nWt]   =   size(img);
img        =   double(img);    %convert to DOUBLE
nBlkHt     =   floor((nHt-2*OVRLP)/BLKSZ);
nBlkWt     =   floor((nWt-2*OVRLP)/BLKSZ);
fftSrc     =   zeros(nBlkHt*nBlkWt,NFFT*NFFT); %stores FFT
nWndSz     =   BLKSZ+2*OVRLP; %size of analysis window.
%-------------------------
%allocate outputs
%-------------------------
oimg       =   zeros(nBlkHt,nBlkWt);
fimg       =   zeros(nBlkHt,nBlkWt);
bwimg      =   zeros(nBlkHt,nBlkWt);
eimg       =   zeros(nBlkHt,nBlkWt);
```

```
enhimg     =   zeros(nHt,nWt);

%-------------------------
%precomputations
%-------------------------
    [x,y]      =   meshgrid(0:nWndSz-1,0:nWndSz-1);
dMult      =   (-1).^(x+y); %used to center the FFT
    [x,y]      =   meshgrid(-NFFT/2:NFFT/2-1,-NFFT/2:NFFT/2-1);
    r          =   sqrt(x.^2+y.^2)+eps;
th         =   atan2(y,x);
th(th<0)    =   th(th<0)+pi;
    w          =   raised_cosine_window(BLKSZ,OVRLP); %spectral window

%-------------------------
%Load filters
%-------------------------
loadangular_filters_pi_4;   %now angf_pi_4 has filter coefficients
    angf_pi_4 = angf;
loadangular_filters_pi_2;   %now angf_pi_2 has filter coefficients
    angf_pi_2 = angf;
%-------------------------
%Bandpass filter
%-------------------------
    FLOW       =   NFFT/RMAX;
    FHIGH      =   NFFT/RMIN;

dRLow      =   1./(1+(r/FHIGH).^4);    %low pass butterworth filter
dRHigh     =   1./(1+(FLOW./r).^4);    %high pass butterworth filter
dBPass     =   dRLow.*dRHigh;          %bandpass

%-------------------------
%FFT Analysis
%-------------------------
for i = 0:nBlkHt-1
nRow = i*BLKSZ+OVRLP+1;
for j = 0:nBlkWt-1
nCol = j*BLKSZ+OVRLP+1;
%extract local block
blk     =   img(nRow-OVRLP:nRow+BLKSZ+OVRLP-1,nCol-OVRLP:nCol+BLKSZ+OVRLP-
1);
%remove dc
dAvg    =   sum(sum(blk))/(nWndSz*nWndSz);
blk     =   blk-dAvg;   %remove DC content
blk     =   blk.*w;     %multiply by spectral window
%--------------------------
%do pre filtering
```

```
%--------------------------
blkfft  =   fft2(blk.*dMult,NFFT,NFFT);
blkfft  =   blkfft.*dBPass;             %band pass filtering
dEnergy =   abs(blkfft).^2;
blkfft  =   blkfft.*sqrt(dEnergy);      %root filtering(for diffusion)
fftSrc(nBlkWt*i+j+1,:) = transpose(blkfft(:));
dEnergy =   abs(blkfft).^2;             %----REDUCE THIS COMPUTATION----
%--------------------------
%compute statistics
%--------------------------
dTotal          =   sum(sum(dEnergy))/(NFFT*NFFT);
fimg(i+1,j+1)   =   NFFT/(compute_mean_frequency(dEnergy,r)+eps); %ridge separation
oimg(i+1,j+1)   =   compute_mean_angle(dEnergy,th);         %ridge angle
eimg(i+1,j+1)   =   log(dTotal+eps);                        %used for segmentation
end;%for j
end;%for i

%-------------------------
%precomputations
%-------------------------
   [x,y]       =   meshgrid(-NFFT/2:NFFT/2-1,-NFFT/2:NFFT/2-1);
dMult       =   (-1).^(x+y); %used to center the FFT

%-------------------------
%process the resulting maps
%-------------------------
for i = 1:3
oimg = smoothen_orientation_image(oimg);            %smoothen orientation image
end;
fimg    =   smoothen_frequency_image(fimg,RMIN,RMAX,5); %diffuse frequency image
cimg    =   compute_coherence(oimg);                    %coherence image for bandwidth
bwimg   =   get_angular_bw_image(cimg);                 %QUANTIZED bandwidth image
%-------------------------
%FFT reconstruction
%-------------------------
for i = 0:nBlkHt-1
for j = 0:nBlkWt-1
nRow = i*BLKSZ+OVRLP+1;
nCol = j*BLKSZ+OVRLP+1;
%--------------------------
%apply the filters
%--------------------------
blkfft  =   reshape(transpose(fftSrc(nBlkWt*i+j+1,:)),NFFT,NFFT);
%--------------------------
%reconstruction
%--------------------------
```

APPENDIX-B

NetBeans Java Code

```
package fingerprints;



import java.awt.Color;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import javax.swing.JOptionPane;

    public addministrator() {

        initComponents();

        try{

Class.forName("com.mysql.jdbc.Driver");

Connection con;con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fingerprint","root","vikash");

Statement s=con.createStatement();

 ResultSet rs=s.executeQuery("select * from semester_list");

 jComboBox3.removeAllItems();

 jComboBox4.removeAllItems();

 jComboBox5.removeAllItems();

 jComboBox6.removeAllItems();

 jComboBox6.addItem('A');
 jComboBox6.addItem('B');

 jComboBox6.addItem('C');

 jComboBox6.addItem('D');

while(rs.next())

{



        jComboBox3.addItem(rs.getString("sem"));

        jComboBox4.addItem(rs.getString("sem"));

        jComboBox5.addItem(rs.getString("sem"));

}

}

catch(ClassNotFoundException | SQLException e){ System.out.println(e);}

    }

    @SuppressWarnings("unchecked")

    // <editor-fold defaultstate="collapsed" desc="Generated Code">

    private void initComponents() {



        jPanel1 = new javax.swing.JPanel();

        jPanel5 = new javax.swing.JPanel();

        jPanel2 = new javax.swing.JPanel();

        jLabel2 = new javax.swing.JLabel();

        jComboBox3 = new javax.swing.JComboBox();

        jLabel7 = new javax.swing.JLabel();

        jButton3 = new javax.swing.JButton();

        jLabel1 = new javax.swing.JLabel();
```

```
jLabel3 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();
jButton2 = new javax.swing.JButton();
jComboBox4 = new javax.swing.JComboBox();
jLabel8 = new javax.swing.JLabel();
jButton4 = new javax.swing.JButton();
jLabel9 = new javax.swing.JLabel();
jTextField2 = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
jComboBox5 = new javax.swing.JComboBox();
jLabel5 = new javax.swing.JLabel();
jComboBox6 = new javax.swing.JComboBox();
jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("ADMINISTRATOR");

jPanel1.setBackground(new java.awt.Color(204, 255, 255));

jPanel5.setBackground(new java.awt.Color(153, 255, 255));

jPanel2.setBackground(new java.awt.Color(51, 255, 255));

jLabel2.setText("Availabel semester");
```

```
        jComboBox3.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1", "Item 2",
"Item 3", "Item 4" }));

        jComboBox3.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                jComboBox3ActionPerformed(evt);

            }

        });


        jLabel7.setText("Enter semester ");


        jButton3.setText("submit");

        jButton3.addActionListener(new java.awt.event.ActionListener() {

            public void actionPerformed(java.awt.event.ActionEvent evt) {

                jButton3ActionPerformed(evt);

            }

        });


        javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);

        jPanel2.setLayout(jPanel2Layout);

        jPanel2Layout.setHorizontalGroup(

            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(jPanel2Layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED_SIZE, 102,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                .addContainerGap(114, Short.MAX_VALUE))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addGap(0, 0, Short.MAX_VALUE)
                .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jButton3)
                    .addComponent(jComboBox3, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(27, 27, 27))
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel2Layout.createSequentialGroup()
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 161,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addContainerGap())
        );
        jPanel2Layout.setVerticalGroup(
            jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(31, 31, 31)
                .addComponent(jLabel2)
                .addGap(65, 65, 65)
                .addComponent(jLabel7)
                .addGap(17, 17, 17)
                .addComponent(jComboBox3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(40, 40, 40)
                .addComponent(jButton3)
                .addContainerGap(126, Short.MAX_VALUE))
        );

        jLabel1.setText("Register new semester");

        jLabel3.setText("Update semester");

        jLabel6.setText("Enter new semester name");

        jTextField1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jTextField1ActionPerformed(evt);
            }
        });

        jButton2.setText("submit");
        jButton2.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton2ActionPerformed(evt);
            }
        });

        jComboBox4.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1", "Item 2",
"Item 3", "Item 4" }));
        jComboBox4.addActionListener(new java.awt.event.ActionListener() {
```

```
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jComboBox4ActionPerformed(evt);
    }
});

jLabel8.setText("Enter semester");

jButton4.setText("submit");
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

jLabel9.setText("New semester name");

jTextField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField2ActionPerformed(evt);
    }
});

javax.swing.GroupLayout jPanel5Layout = new javax.swing.GroupLayout(jPanel5);
jPanel5.setLayout(jPanel5Layout);
jPanel5Layout.setHorizontalGroup(
      jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel5Layout.createSequentialGroup()
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jPanel5Layout.createSequentialGroup()
            .addGap(19, 19, 19)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
              .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED_SIZE, 183,
javax.swing.GroupLayout.PREFERRED_SIZE)
              .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 153,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()
            .addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()
                .addComponent(jButton2)
                .addGap(40, 40, 40))
              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()
                .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE, 105,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))))
        .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
          .addGroup(jPanel5Layout.createSequentialGroup()
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addGroup(jPanel5Layout.createSequentialGroup()
                .addGap(63, 63, 63)
                .addComponent(jLabel3))
              .addGroup(jPanel5Layout.createSequentialGroup()
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                  .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 97,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(jLabel9, javax.swing.GroupLayout.PREFERRED_SIZE, 127,
javax.swing.GroupLayout.PREFERRED_SIZE))))
            .addContainerGap(78, Short.MAX_VALUE))
          .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel5Layout.createSequentialGroup()
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jComboBox4, 0, 79, Short.MAX_VALUE)
              .addComponent(jButton4)
              .addComponent(jTextField2))
            .addGap(26, 26, 26))))
    );
    jPanel5Layout.setVerticalGroup(
      jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

      .addGroup(jPanel5Layout.createSequentialGroup()

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

          .addGroup(jPanel5Layout.createSequentialGroup()

            .addGap(26, 26, 26)

            .addComponent(jLabel1))

          .addGroup(jPanel5Layout.createSequentialGroup()

            .addGap(34, 34, 34)

            .addComponent(jLabel3)))

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

          .addGroup(jPanel5Layout.createSequentialGroup()

            .addGap(61, 61, 61)

            .addComponent(jLabel6)

            .addGap(18, 18, 18)

            .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(38, 38, 38)

            .addComponent(jButton2))

          .addGroup(jPanel5Layout.createSequentialGroup()

            .addGap(60, 60, 60)

            .addComponent(jLabel8)

            .addGap(10, 10, 10)

            .addComponent(jComboBox4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGap(18, 18, 18)

            .addComponent(jLabel9)
```

```
                .addGap(18, 18, 18)
                .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGap(31, 31, 31)
            .addComponent(jButton4)
            .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );

        jLabel4.setText("Start attendance for semester");

        jComboBox5.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
        jComboBox5.addActionListener(new java.awt.event.ActionListener() {
          public void actionPerformed(java.awt.event.ActionEvent evt) {
            jComboBox5ActionPerformed(evt);
          }
        });

        jLabel5.setText("Enter Section");

        jComboBox6.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Item 1", "Item 2", "Item 3", "Item 4" }));
        jComboBox6.addActionListener(new java.awt.event.ActionListener() {
          public void actionPerformed(java.awt.event.ActionEvent evt) {
            jComboBox6ActionPerformed(evt);
          }
        });
```

```
    jButton1.setText("submit");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
      }
    });

    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addComponent(jPanel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(114, 114, 114)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jButton1)
          .addGroup(jPanel1Layout.createSequentialGroup()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
              .addComponent(jLabel5)
              .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED_SIZE, 224,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jComboBox5, 0, 99, Short.MAX_VALUE)
              .addComponent(jComboBox6, 0, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
        .addGap(23, 23, 23)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
          .addComponent(jLabel4)
          .addComponent(jComboBox5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(29, 29, 29)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
          .addComponent(jLabel5)
          .addComponent(jComboBox6, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jButton1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 22, Short.MAX_VALUE)
        .addComponent(jPanel5, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
```

```
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>


    private void jComboBox5ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    try{
```

```
Class.forName("com.mysql.jdbc.Driver");
Connection con;con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fingerprint","root","vikash");
Statement s=con.createStatement();
if (jTextField1.getText().toString().equals(""))
{
  JOptionPane.showMessageDialog(null,"Semester field cannot be left blank");
  jTextField1.requestFocus();
}
else
    {
      String su;
      su = "insert into semester_list (sem) Values('"+jTextField1.getText()+"')";
      s.executeUpdate(su);
      ResultSet rs=s.executeQuery("select * from semester_list");
      jComboBox3.removeAllItems();
      jComboBox4.removeAllItems();
      jComboBox5.removeAllItems();

      while(rs.next())
      {

        jComboBox3.addItem(rs.getString("sem"));
        jComboBox4.addItem(rs.getString("sem"));
        jComboBox5.addItem(rs.getString("sem"));
      }
```

```
          String su1;

    su1 = "create table "+jTextField1.getText()+" (name varchar(30),roll varchar(4),regno
varchar(10),addmmission_no varchar(10));";

s.executeUpdate(su1);

          this.dispose();

          String g=jTextField1.getText();

          registration s1 = new registration(g);

          s1.setVisible(true);

       }

    }

catch(ClassNotFoundException | SQLException e){ System.out.println(e);

JOptionPane.showMessageDialog(null,"Semester is already available");

jTextField1.requestFocus();}

    }


    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {

       // TODO add your handling code here:

    }


    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {

      try{

Class.forName("com.mysql.jdbc.Driver");

Connection con;con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fingerprint","root","vikash");

Statement s=con.createStatement();

if (jTextField2.getText().toString().equals(""))
```

```
{
   JOptionPane.showMessageDialog(null,"New Semester field cannot be left blank");
   jTextField2.requestFocus();
}
else
{
String su;
   su = "rename table "+jComboBox4.getSelectedItem().toString()+" to "+jTextField2.getText()+"";
s.executeUpdate(su);
String b="update semester_list set sem='"+jTextField2.getText()+"' where
sem='"+jComboBox4.getSelectedItem().toString()+"'";
 s.executeUpdate(b);
 ResultSet rs=s.executeQuery("select * from semester_list");
         jComboBox3.removeAllItems();
         jComboBox4.removeAllItems();
         jComboBox5.removeAllItems();

         while(rs.next())
         {

            jComboBox3.addItem(rs.getString("sem"));
            jComboBox4.addItem(rs.getString("sem"));
            jComboBox5.addItem(rs.getString("sem"));
         }
 JOptionPane.showMessageDialog(null,"Semester has updated");
 jTextField2.setText("");
}

}

catch(ClassNotFoundException | SQLException e){ System.out.println(e);

JOptionPane.showMessageDialog(null,"Semester is already available");

jTextField1.requestFocus();}// TODO add your handling code here:

    }


    private void jComboBox6ActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

    }


    private void jComboBox3ActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

    }


    private void jComboBox4ActionPerformed(java.awt.event.ActionEvent evt) {

        // TODO add your handling code here:

    }


    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

     this.dispose();

     String b=jComboBox3.getSelectedItem().toString();

            sem s1 = new sem(b);

            s1.setVisible(true);   // TODO add your handling code here:

    }
    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(addministrator.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(addministrator.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(addministrator.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(addministrator.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        //</editor-fold>

        //</editor-fold>


        /* Create and display the form */

        java.awt.EventQueue.invokeLater(new Runnable() {

            public void run() {

                new addministrator().setVisible(true);

            }

        });

    }


    // Variables declaration - do not modify

    private javax.swing.JButton jButton1;

    private javax.swing.JButton jButton2;

    private javax.swing.JButton jButton3;

    private javax.swing.JButton jButton4;

    private javax.swing.JComboBox jComboBox3;

    private javax.swing.JComboBox jComboBox4;

    private javax.swing.JComboBox jComboBox5;

    private javax.swing.JComboBox jComboBox6;

    private javax.swing.JLabel jLabel1;

    private javax.swing.JLabel jLabel2;

    private javax.swing.JLabel jLabel3;

    private javax.swing.JLabel jLabel4;

    private javax.swing.JLabel jLabel5;
```

```
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JPanel jPanel2;
    private javax.swing.JPanel jPanel5;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}


package fingerprints;


import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import javax.swing.JOptionPane;
    public registration() {
        initComponents();
    }
public registration(String a) {
    sem=a;
        initComponents();
```

```
}
 @SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jDesktopPane1 = new javax.swing.JDesktopPane();
    jPanel1 = new javax.swing.JPanel();
    jLabel1 = new javax.swing.JLabel();
    jTextField1 = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jTextField3 = new javax.swing.JTextField();
    jTextField4 = new javax.swing.JTextField();
    jTextField2 = new javax.swing.JTextField();
    jButton1 = new javax.swing.JButton();
    jButton2 = new javax.swing.JButton();
    jLabel5 = new javax.swing.JLabel();

    javax.swing.GroupLayout jDesktopPane1Layout = new javax.swing.GroupLayout(jDesktopPane1);
    jDesktopPane1.setLayout(jDesktopPane1Layout);
    jDesktopPane1Layout.setHorizontalGroup(
        jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGap(0, 100, Short.MAX_VALUE)
    );
    jDesktopPane1Layout.setVerticalGroup(
    jDesktopPane1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGap(0, 100, Short.MAX_VALUE)

);


setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);

setTitle("REGISTRATION\n");

setBackground(new java.awt.Color(51, 255, 255));

setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));

setForeground(new java.awt.Color(0, 0, 0));


jPanel1.setBackground(new java.awt.Color(153, 255, 255));


jLabel1.setText("Registration no:");


jTextField1.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        jTextField1ActionPerformed(evt);

    }

});


jLabel2.setText("Name:");


jLabel3.setText("Roll no:");


jTextField3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField3ActionPerformed(evt);
    }
});

jTextField4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField4ActionPerformed(evt);
    }
});

jTextField2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jTextField2ActionPerformed(evt);
    }
});

jButton1.setText("Submit");
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setText("home");
```

```
    jButton2.addActionListener(new java.awt.event.ActionListener() {

      public void actionPerformed(java.awt.event.ActionEvent evt) {

        jButton2ActionPerformed(evt);

      }

    });


    jLabel5.setText("Admission No:");


    javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

    jPanel1.setLayout(jPanel1Layout);

    jPanel1Layout.setHorizontalGroup(

      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

      .addGroup(jPanel1Layout.createSequentialGroup()

        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

          .addGroup(jPanel1Layout.createSequentialGroup()

            .addGap(30, 30, 30)


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()

                .addComponent(jButton2)

                .addGap(38, 38, 38))

              .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel1Layout.createSequentialGroup()


.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                  .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT_SIZE, 87,
Short.MAX_VALUE)
                  .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
                  .addComponent(jLabel5, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)))

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
              .addComponent(jTextField1)
              .addComponent(jTextField3)
              .addComponent(jTextField4)
              .addComponent(jTextField2, javax.swing.GroupLayout.DEFAULT_SIZE, 157,
Short.MAX_VALUE)))
          .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(172, 172, 172)
            .addComponent(jButton1)))
        .addContainerGap(211, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(
      jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
      .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(29, 29, 29)
        .addComponent(jButton2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 74,
Short.MAX_VALUE)
```

```
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel5))
                .addGap(26, 26, 26)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel1))
                .addGap(18, 18, 18)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jTextField4, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel2))
                .addGap(27, 27, 27)
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jLabel3))
                .addGap(43, 43, 43)
                .addComponent(jButton1)
                .addGap(93, 93, 93))
        );

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        );

        pack();
    }// </editor-fold>

    private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jTextField4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
    this.dispose();

addministrator s1;

      s1 = new addministrator();

s1.setVisible(true);

  // TODO add your handling code here:

   }



   private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

   try{

Class.forName("com.mysql.jdbc.Driver");

Connection con;con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/fingerprint","root","vikash");

Statement s=con.createStatement();

String su;

if(jTextField1.getText().toString().equals(""))

{

JOptionPane.showMessageDialog(null,"admission no can not be left blank ");

jTextField1.requestFocus();

}

else if(jTextField3.getText().toString().equals(""))

{

   JOptionPane.showMessageDialog(null,"Registration no can not be left blank ");

jTextField3.requestFocus();

}

else if(jTextField4.getText().toString().equals(""))

{
   JOptionPane.showMessageDialog(null,"Name can not be left blank ");
jTextField4.requestFocus();
}
else if(jTextField2.getText().toString().equals(""))
{
   JOptionPane.showMessageDialog(null,"Roll no can not be left blank ");
jTextField2.requestFocus();
}
else
{
   su = "insert into "+sem+" (name,roll,regno,addmmission_no) Values
('"+jTextField4.getText()+"','"+jTextField2.getText()+"','"+jTextField3.getText()+"','"+jTextField1.getText()
+"')";
s.executeUpdate(su);
JOptionPane.showMessageDialog(null,"Registration of "+jTextField4.getText()+" is complited go for next
registration");
jTextField1.setText("");
jTextField2.setText("");
jTextField3.setText("");
jTextField4.setText("");
jTextField1.requestFocus();
}


   }


catch(ClassNotFoundException | SQLException e){ System.out.println(e);}   // TODO add your handling
code here:
    }

    private void jTextField3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }
    public static void main(String args[]) {
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
              try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(registration.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(registration.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(registration.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(registration.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
```

```
        }
        //</editor-fold>

        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new registration().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JDesktopPane jDesktopPane1;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    // End of variables declaration
```