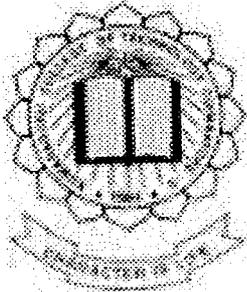


P- 891

AUTHENTICATION USING BIOMETRICS

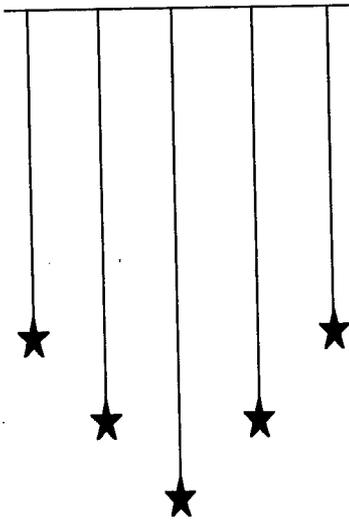


SUBMITTED BY

K.ARUNMOZHI

S.PAVITHRA

T.VARUN CHARAVARTHI



UNDER THE GUIDANCE OF,

Mrs.V.VANITHA M.E.,

Dept.of Computer Science and Engg.

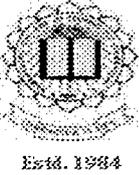


SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE AWARD OF BE DEGREE IN INFORMATION AND
TECHNOLOGY ENGINEERING UNDER BHARATHIAR UNIVERSITY

March 2003

*Department Of Computer Science Engineering
Kumaraguru College Of Technology
Coimbatore-641006.*

CERTIFICATE



DEPARTMENT OF INFORMATION AND
TECHNOLOGY ENGINEERING
KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641 006



CERTIFICATE

This is to certify that the Project Report entitled
“AUTHENTICATION USING BIOMETRICS”

has been submitted by

K.ARUNMOZHI (9927S0045)

S.PAVITHRA (9927S0065)

T.VARUNCHARAVARTHI (9927S0081)



*In partial fulfillment of the requirements for the award of the degree of Bachelor of
Engineering in Information and Technology branch of Bharathiar University,
Coimbatore during the academic year 2002 - 2003*

[Signature]
19/3/03

Guide

[Signature]
Head of Department

Was examined in project work Viva – Voce Examination on 20.3.2003.

Internal Examiner

External Examiner

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

We are thankful to our principal **Dr. K.K. Padmanaban, B.Sc(Engg)., M.Tech., Ph.D.**, for his invaluable support in the completion of this project.

We are extremely thankful to **Dr. S. Thangasamy, Ph.D.**, Head of the Department, Department of Computer Science and Engineering for his valuable advice and suggestions throughout this project.

We are indent to express our heartiest thanks to our project guide **Mrs. V. Vanitha, M.E.**, Lecturer, Department of Computer Science and Engineering, who rendered her valuable guidance and support to perform our project work extremely well.

We are indent to express our heartiest thanks to our project coordinator **Mr. K. R. Baskaran, M.S.**, Asst. professor, Department of Information Technology, and to our class advisor **Mrs. N. Chitra Devi M.E.**, Lecturer, Department of Information Technology, who has helped us to perform the project work extremely well.

We are also thankful to all faculty members of the Department of Computer Science and Engineering, Kumaraguru College of Technology, CBE for their valuable guidance, support and encouragement during the course of our project work.

We are also thankful to the company "**NEW WALK TECHNOLOGY**" for providing the project and the project guide for his valuable guidance to complete the project successfully.

We express our humble gratitude and thanks to our parents and family members who have supported and helped us to complete the project and our friends, for lending us valuable tips, support and co-operation throughout our project work.

SYNOPSIS

SYNOPSIS

The project “**Authentication Using Biometrics**” is to reduce the manual work in Forensic Department and to trace out or identify the criminals using Fingerprints. The main objective of this project is to find out the criminal using the fingerprint. The fingerprint which is gathered by Investigation Department is compared manually or by using scanner. In manual comparison the fingerprint is stored in database and compared by experts only. If the fingerprint matches with the fingerprint in the database, then the details about the fingerprint is displayed.

In our project the fingerprint comparison is done with the help of scanner, where the fingerprint is taken through the scanner and then compared. If the fingerprint matches with any of the fingerprint stored in database, the criminal can be identified. If the fingerprint does not match using scanner then the message will be displayed.

The comparison process consist of two main modules namely the enrollment and recognition. The enrollment deals with entering the convict details in to the database. Every detail is entered with a unique code number. Recognition involves comparing the fingerprint with the stored fingerprints in database. If any match is found then convict detail is displayed else proper message is thrown. It also allows verification of fingerprints. The new users can be added or deleted by administrator who mainly can access the database.

CONTENTS

CONTENTS

Chapter	Description	Page No
1.	Introduction -----	1
	1.1 Current Status of the Problem taken -----	2
	1.2 Proposed System -----	2
2.	Literature Survey -----	3
3.	Proposed Approach To a Product -----	4
	3.1 Hardware Specification-----	6
	3.2 Software Specification -----	6
	3.3 Introduction to software -----	7
4.	System Design -----	8
	4.1 Input Design -----	9
	4.2 Output Design -----	10
	4.3 Table Design -----	11
5.	Implementation Details -----	14
	5.1 Introduction -----	15
	5.2 Data Flow Diagram -----	15
	5.3 Modules Description -----	18
	5.3.1 Enrollment module -----	18
	5.3.2 Creating the Database -----	19
	5.3.3 Recognition module -----	21
6.	System Testing -----	22
7.	Conclusion and Future Scope -----	24
	7.1 Conclusion -----	25
	7.2 Future Scope of the Project -----	25
8.	References -----	26
9.	Appendix -----	28
	9.1 Coding -----	29
	9.2 Sample Screens -----	42

INTRODUCTION

1. INTRODUCTION

1.1 CURRENT STATUS OF THE PROBLEM:

In the existing system, the fingerprint comparison is done only based on the location of the ridges and furrows. It takes lots of time for the experts to compare the fingerprint with stored fingerprints. There is no extraction of the singular or minutia points. Separate records are needed to store the details about the criminal. So it is difficult to handle and retrieve the data's. There is no exact matching for the accidental fingerprint images.

1.2 PROPOSED SYSTEM:

Our project "Authentication Using Biometrics" is carried out using a scanner is an enhanced system, satisfies the requirements as needed. The fingerprint is taken through scanner and initially the singular point extraction is done and the details are stored in the database. Comparison is done based on the attributes extracted. So our proposed system is fast because there is no comparison of the actual images. Since MS Access is used as backend it provides easy storage, retrieval and updating.

2. LITERATURE SURVEY

The main objective of this project “Authentication Using Biometrics” is to find out criminal using the fingerprints. The fingerprint comparison can be done either by manually or using scanner. In existing system the comparison is done manually, which takes lots of time for the experts to compare and has difficulties in reteriving, updating and storing fingerprints in database. Our system which the enhanced system meets the requiremets as needed and provides lots advantages over the existing system. The comparison is done using the scanner.

PROPOSED APPROACH TO THE PROJECT

3. PROPOSED APPROACH TO THE PRODUCT

In the proposed system, the details of the criminal is gathered and stored in the database which is easy to access. The fingerprint can be stored in the database and can easily transfer from one place to another. The expert can easily compare the fingerprint which they get from the investigation department to the fingerprint stored in the database and give the result.

The details about the fingerprint which is matched will be obtained with in a fraction of time. So the system is very useful in forensic department to catch or identify the criminal. Using the scanner it is easy to feed the fingerprint of a person and to identify for the result. The system approximately calculates the variation in the fingerprint which is stored earlier and gives the result.

Moreover, this system uses Access as the backend to store and retrieve the data. Administrator gives all the permission to the experts in investigating a particular case. The details about the fingerprint attributes are stored in the form of the table named Data. Experts can add a new detail of a newly investigated criminal. The Convict is the table used to store the details of the crime. Experts can also update some of the data about the existing record. Experts compare the fingerprints and give the required information.

Thus the system is very useful to the forensic department of the criminals. It is sufficient and designed to satisfy the needs of the forensic department.

3.1 HARDWARE SPECIFICATION:

System	:	Pentium III @ 600 MHz.
Cache	:	128 MB.
RAM	:	128 MB.
Hard Disk	:	20 GB.
Monitor	:	14" Color Monitor.
Keyboard	:	104 Enhanced.
Mouse	:	Logitech three button mouse.

3.2 SOFTWARE REQUIREMENTS

Front End	:	Visual Basic 6.0
Back End	:	MS Access
Operating System	:	Windows 98

3.3 INTRODUCTION TO SOFTWARE USED:

Visual Basic has been chosen as it provides a GUI based environment for creating user friendly forms. Visual Basic is an ideal programming language for developing sophisticated applications in windows platform. The 'Visual' part refers to the graphical user interface. Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply add pre built objects into place on the screen. The 'Basic' part refers to the BASIC language. Visual Basic has evolved from the original BASIC languages and now contains several hundred statements, functions and keywords, many of which relate directly to the windows GUI.

General Features of Visual Basic 6.0:

- Compile a VB project to native for faster execution
- Open multiple projects in the same instance of VB.
- ActiveX documents boost the VB application to the internet and Intranet browser windows.
- Ability to do single, multiple, or Microsoft explorer style document interface application.
- The new model allows us to programmatically extend the development environment and control project, events, code visual elements.
- The application wizard is new and the setup wizard has been enhanced to enable creating a dependency file for a standard projects.
- Most control now support drag and from components specifically for employment of the web.
- The implements feature allows your classes to support multiple interfaces.
- Command line switches allow your classes to support multiple interfaces.
- Command line switches provide a way to control how VB executes.
- The resource file allows you to collect all of the versions specify text and bitmaps for an application is one place.

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 INPUT DESIGN

Input design process is to design the various needed into a machine oriented format. The main objective is to create an input layout that is easy to follow and to avoid operator errors.

The objectives of the input design is to:

- Produce output in neat form
- Get high level of accuracy

Input Data:

The goal of designing input data is to make data entry as easy, logical and free from errors. While entering the data, we should know aware of the following,

- The allocated space for each field
- Field sequence, which must match that in the source document
- The format in which data fields is entered

Approaches for entering data:

The approaches used for entering data into computer are Menus and Forms.

Menu:

A menu is a selection list that supplies computer data access or entry. Instead of remembering what to enter, the user chooses from a list of options and types the option letter associated with it. A menu limits choice of responses but reduces the chances for error in data entry.

Form:

A formatted form is a template that requests the user to enter data in appropriate locations. It is a fill in the blank type form, which also contains combo box, option box etc. The form is flushed on the screen as a unit. The cursor is usually positioned at the first blank. After the user responds by filling in the appropriated information, the cursor moves to the next line and so on until the form is completed. During this routine, the user may move the cursor up, down, right or left to various locations for making changes in the response.

4.2 OUTPUT DESIGN:

The output provides direct source of information to the user and so it is the most important in the system design phase. They also provide a permanent hardcopy of results. Proper messages is thrown on the screen based on the process.

4.3 TABLE DESIGN:

Name: Convicts

Purpose: Maintains the detail about each convict

Sno	Field Name	Type
1	Concode	Autonumber
2	Conname	Text
3	Birthdate	Text
4	Street1	Text
5	Street2	Text
6	City	Text
7	Distcode	Text
8	Country	Text
9	Pcode	Text
10	Coucode	Text

Name: Court

Purpose: Maintains court details

Sno	Field Name	Type
1	Coucode	Text
2	Name	Text

AUTHETICATION USING BIOMETRICS

Name: Data

Purpose: Maintains details about each fingerprint.

Sno	Field Name	Type
1	Id	Autonumber
2	G	Number
3	Fn	Number
4	Fx	OLE object
5	Fy	OLE object
6	Fd	OLE object
7	Fc	OLE object
8	Dirscount	Number
9	Dirs	OLE object
10	Spcount	Number
11	Spx	OLE object
12	Spy	OLE object
13	Spd	OLE object
14	Spt	OLE object

AUTHETICATION USING BIOMETRICS

Name: District

Purpose: Stores the district details.

Sno	Field Name	Type
1	Distcode	Text
2	Distname	Text

Name: Police

Purpose: Stores the details about the police.

Sno	Field Name	Type
1	Pcode	Text
2	Pname	Text

Name: Users

Purpose: Maintains details about the users.

S.no	Field Name	Type
1	User_id	Text
2	Pwd	Text

IMPLEMENTATION DETAILS

5. IMPLEMENTATION DETAILS

5.1 INTRODUCTION:

Our project “Authentication Using Biometrics” is carried out using a scanner is an enhanced system, satisfies the requirements as needed. The fingerprint is taken through scanner and initially the singular point extraction is done and the details are stored in the database. Comparison is done based on the attributes extracted. So our proposed system is fast because there is no comparison of the actual images. Since MS Access is used as backend it provides easy storage, retrieval and updating.

5.2 DATA FLOW DIAGRAM:

DFD is a representation of a system. They are excellent mechanism for communicating with customers during system analysis. A DFD , also known as a “Bubble Chart” which clarifies system requirements and identifying major transformations. It is the starting point in system design that decomposes the requirement specification down to the lowest level of detail.

A DFD represent data flow between individual statements or block of statements in a routine, data flow between sequential routines, data flow between concurrent processes or a data flow in a distributed computing system where each node represents a geographically remote processing unit.

DFD can be used at any level of detail. DFD are quite valuable for establishing naming conventions and names of system components such as subsystems, files and data links. It describes what flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structures or file organization. DFD consists of a series of bubbles joined by lines. The bubble represents data transformation and the line represents data flows in the system.

Objective Of DFD

- It provide an indication of how data are transformed as they move through the system.
- It depict the functions that transforms the data flow

Symbols In DFD

There are four symbols used,

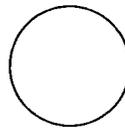
- A square, defines a source or destination of system data



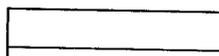
- An arrow, identifies dataflow-data in motion



* A circle or bubble, represents a process that transforms incoming data flows in to out flows



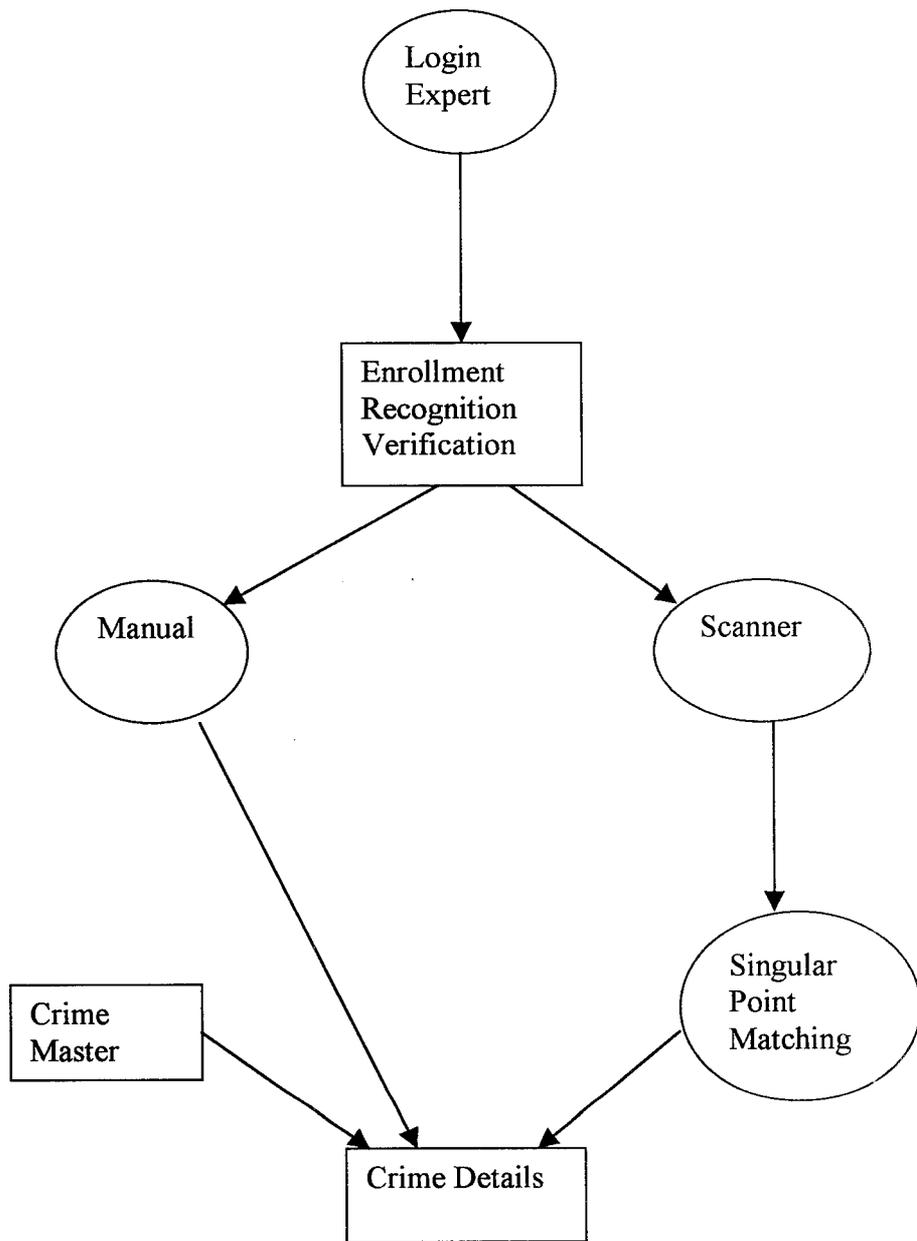
- An open rectangle , is a data store-data at rest or a temporary repository of data



Several rules of thumb are used in drawing DFD,

- Process should be named and numbered for easy reference.
- The direction of flow is from top to bottom and from left to right. Data traditionally flow from the source to the destination, although they may flow back to a source.
- When a process is exploded into lower-level details, they are numbered.
- The names of data stores, sources and destination are written in capital letter.

DATAFLOW DIAGRAM:



5.3 MODULES DESCRIPTION:

The modules present in this project are,

- Enrollment Module
- Creating a database
- Recognition Module

5.3.1 ENROLLMENT MODULE:

Before enrolling the fingerprint image, the personal details of the convict must be entered and stored in the database. After storing the convict details in a database, the image must be enrolled. During enrollment, the attributes of the respective fingerprint is extracted and stored with unique ID in the database. The attributes extracted are ridge count, density, x & y co-ordinates, direction and actual type of the fingerprint.

The basic types of the fingerprints are,

- Whorl
- Loop
 - Left Loop
 - Right Loop
 - Double Loop
- Arch
 - Plain Arch
 - Tented Arch
- Accidental

5.3.2 CREATING THE DATABASE:

Data Base:

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The database serves as the repository of data, so a well-designed database can lead to a better program structure and reduce procedural complexity. In a database environment, common data are available and used by several users.

The design of a database is one integrated step in the whole process of system design. The process of database design is splitted into three phases:

Conceptual Design:

Take all the requirement specifications that have come and create a design from end users and create a design that is independent of any hardware or software concern. The Entity-Relationship model is the most popular technique.

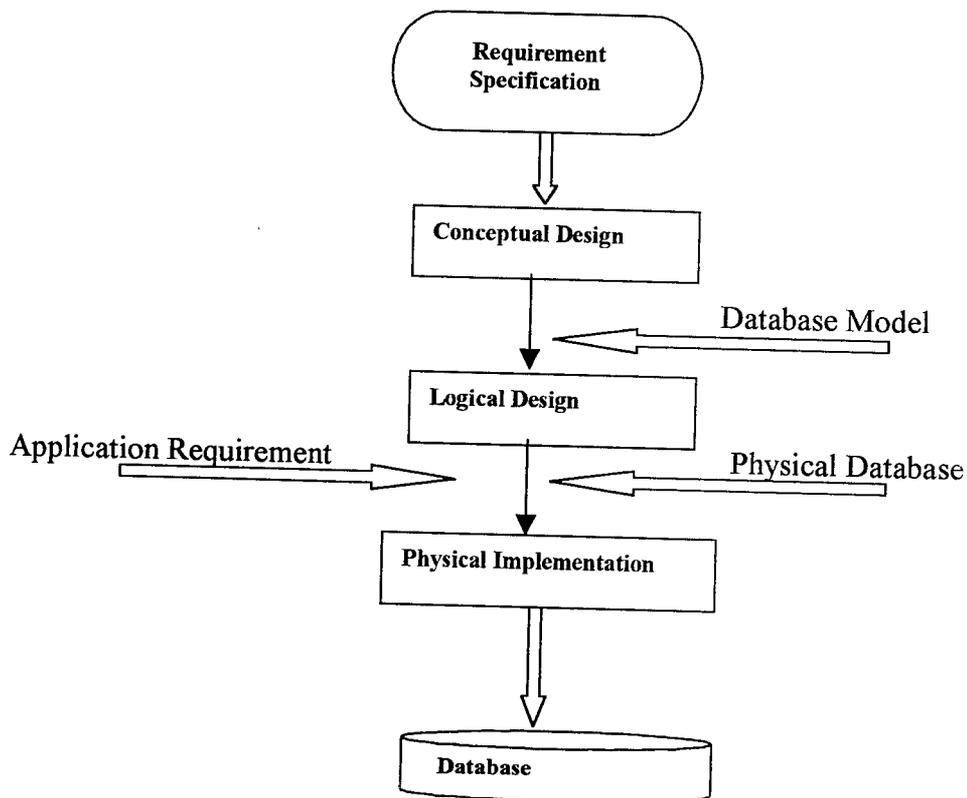
Logical Design:

Map the conceptual model into a logical model suitable for the DBMS used.

Physical Implementation:

Physical implementation is the final phase of creating a database. Implement the physical data tables and make design decisions to enhance the usefulness of the database with the proposed application.

PROCESS OF CREATING THE DATABASE:



5.3.3 RECOGNITION MODULE:

The given fingerprint image for recognition is scanned then the attributes of that fingerprint are extracted, then the extracted attributes are compared with the stored attributes in the database. Only the fingerprints which have the same value of the attributes extracted are retrieved and compared with the actual image. Due to this process of comparison, the recognition speed is increased. In the case of accidental fingerprints the singular point count and the corresponding x & y co-ordinates must be compared. This process is more enough to recognize the actual image, if available in the database.

If the match is found then the details of the respective convict is retrieved from the database and displayed. If there is no match, then the respective message will be displayed in the dialog box.

6. SYSTEM TESTING

Software testing is one in which the developer establishes test procedures for testing the software against various requirements. Many software testing techniques are available.

Preparing for unit testing:

The developer shall establish test cases (in terms of inputs, expected results, and evaluation criteria), test procedures, and test data for testing the software corresponding to each software unit. The test cases shall cover all aspects of the unit's detailed design. The developer shall record this information in the appropriate software development files.

Performing unit testing:

The developer shall test the software corresponding to each software unit. The testing shall be in accordance with the unit test cases and procedures.

Revision and retesting:

The developer shall make all necessary revisions to the software, perform all necessary retesting, and update the software development files and other software products as needed, based on the results of unit testing.

Analyzing and recording unit test results:

The developer shall analyze the results of the unit testing and shall record the test and analysis results in appropriate software development files.

CONCLUSION AND FUTURE SCOPE

7. CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

The “**Authentication using biometric**” has been developed to meet almost all the requirements of the Forensic Department. The entire system is menu driven and event driven which is useful when worked by native users.

The system has been developed in **VISUAL BASIC 6.0**. the system is more helpful and advantages over the existing system. Since fingerprints are processed much faster and reports in the required format are quite easily obtained.

Any system may have its own drawbacks and can be modified further to incorporate the required changes.

7.2 FUTURE SCOPE OF THE PROJECT:

The main objective of this project is to find out the criminal using the fingerprint. The future scope is to recognize the fingerprint even when a small part of a fingerprint is given as input. This will be helpful for Forensic Department to identify the criminal even with the small evidence.

This project can be enhanced for authentication of students in institution (for example in library, stores, etc.), employees in an organization, security access for the important records, etc.

REFERENCES

8. REFERENCES

1. Roger S. Pressman
“Software Engineering and Application”,
Mc Graw Hill; Fourth Edition.
2. Gary Cornell
“Visual Basic 6”,
Tata Mc Graw Hill, Thrid Edition.
3. Donald Hearn, M. Pauline Baker
“Computer Graphics”,
Prentice Hall India, Second Edition.
4. www.biolkusa.com (Dec 7th).
5. www.biometrics.com (Jan 10th).
6. www.visionics.com (Feb 12th)

APPENDIX

9. APPENDIX

9.1 CODINGS:

EXTRACTING SINGULAR POINTS:

```
' Fingerprint data structure
Public Type FPDATA
    features_number As Long 'How many minutiae are extracted from the fingerprint
image
    feature_coordinatex() As Long 'Minutiae X coordinates' array in pixels
    feature_coordinatey() As Long 'Minutiae Y coordinates' array in pixels
    feature_direction() As Long 'Minutiae directions' array in pixels
    feature_curvature() As Long 'Minutiae curvatures' array in pixels
    dirs_block_count As Long 'Size of blocked directional image
    Dirs() As Byte 'Blocked directional image
End Type
```

```
' Fingerprint preclassification data structure
Public Type FPCLASSDATA
    G As Long ' Average ridge density in fingerprint
    SpCount As Long 'Number of singular points found
    SpX() As Long 'Array of X coordinates of singular points
    SpY() As Long 'Array of Y coordinates of singular points
    SpD() As Long 'Array of directions of singular points
    SpT() As Long 'Array of types of singular points
End Type
```

```
' Fingerprint database element (stored in memory)
Public Type FPMemBaseElement
    fpd As FPDATA
    fpc As FPCLASSDATA
    ID As Long
End Type
```

```
'Function for allocating matching data if matching in multiple threads is required
Public Declare Function CreateFMO Lib "VFVBP40.DLL" () As Long
```

AUTHETICATION USING BIOMETRICS

'Function for deallocating matching data

Public Declare Sub DestroyFMO Lib "VFVBP40.DLL" (ByVal FMH As Long)

'Function for setting algorithm parameters

Public Declare Sub VFSetParameters Lib "VFVBP40.DLL" (ByRef Parameters As VF_PARAMETERS)

'Function for getting current algorithm parameters

Public Declare Sub VFGetParameters Lib "VFVBP40.DLL" (ByRef Parameters As VF_PARAMETERS)

'Function for minutiae extraction from the fingerprint bitmap

Public Declare Function FeaturesExtraction Lib "VFVBP40.DLL" (_
 ByVal Pic As Long, ByVal Ratio As Long, ByVal settings As Long, _
ByRef fpd As FPDATA, ByRef fpc As FPCLASSDATA, ByRef result As Long) As Long

'Function for the enrolled fingerprint minutiae collections generalization,
'quality validation and noise minutiae elimination

'(using the minutiae collection of three same finger fingerprints)

Public Declare Function FeaturesGeneralization Lib "VFVBP40.DLL" (_
 sfpd1 As FPDATA, sfpc1 As FPCLASSDATA, _
 sfpd2 As FPDATA, sfpc2 As FPCLASSDATA, _
 sfpd3 As FPDATA, sfpc3 As FPCLASSDATA, _
 ByVal RECOGNITION_THRESHOLD As Long, sfpd As FPDATA, sfpc As
FPCLASSDATA) As Long

'Function for two fingerprints feature collection matching

Public Declare Function FingerprintsMatching Lib "VFVBP40.DLL" (_
 test As FPDATA, sample As FPDATA, _
 testfpc As FPCLASSDATA, samplefpc As FPCLASSDATA, _
 ByVal Maximal_Rotation As Long, _
 ByVal RECOGNITION_THRESHOLD As Long, _
 ByVal mode As Long, _
 ByVal fmhandle As Long) As Long

Displays recognized features:

```

Public Function DrawFeatures(Pic As PictureBox, fpd As FPDATA, fpc As
FPCLASSDATA, Ratio As Single)
    Dim x As Integer, y As Integer
    Pic.DrawWidth = 2
    For i = 0 To fpd.features_number - 1 'Displays features
        x = -10 * Cos(0.017453292 * fpd.feature_direction(i))
        y = 10 * Sin(0.017453292 * fpd.feature_direction(i))
        Pic.Line (Ratio * fpd.feature_coordinatex(i), Ratio * fpd.feature_coordinatey(i)-
(Ratio * fpd.feature_coordinatex(i) + x, Ratio * fpd.feature_coordinatey(i) + y),
RGB(255, 0, 0)
        Pic.Circle (Ratio * fpd.feature_coordinatex(i), Ratio * fpd.feature_coordinatey(i)),
4, RGB(255, 0, 0)
    Next i
    Pic.DrawWidth = 4
    For i = 0 To fpc.SpCount - 1 'Displays singular points
        Select Case fpc.SpT(i)
            Case 1: 'Core singular point, rectangle
                x = fpc.SpX(i) - 8
                y = fpc.SpY(i) - 8
                Pic.Line (Ratio * x, Ratio * y)-(Ratio * (x + 16 - 1), Ratio * y), RGB(255, 0, 0)
                Pic.Line -(Ratio * (x + 16 - 1), Ratio * (y + 16 - 1)), RGB(255, 0, 0)
                Pic.Line -(Ratio * x, Ratio * (y + 16 - 1)), RGB(255, 0, 0)
                Pic.Line -(Ratio * x, Ratio * y), RGB(255, 0, 0)
                x = -16 * Cos(0.017453292 * fpc.SpD(i))
                y = 16 * Sin(0.017453292 * fpc.SpD(i))
                Pic.Line (Ratio * fpc.SpX(i), Ratio * fpc.SpY(i)-(Ratio * (fpc.SpX(i) + x), Ratio
* (fpc.SpY(i) + y)), RGB(255, 0, 0)
            Case 2: 'Double core singular point, circle
                Pic.Circle (Ratio * fpc.SpX(i), Ratio * fpc.SpY(i)), 8, RGB(255, 0, 0)
            Case -1: 'Delta singular point, triangle
                x = fpc.SpX(i) - 8
                y = fpc.SpY(i) - 8
                Pic.Line (Ratio * x, Ratio * (y + 16 - 1)-(Ratio * (x + 16 / 2), Ratio * y),
RGB(255, 0, 0)
                Pic.Line -(Ratio * (x + 16 - 1), Ratio * (y + 16 - 1)), RGB(255, 0, 0)
                Pic.Line -(Ratio * x, Ratio * (y + 16 - 1)), RGB(255, 0, 0)
        End Select
    Next i
End Function

```

STORING DETAILS IN DATABASE:

```
Private Sub Form_Activate()
```

```
    If Not clkform = False Then  
        If Not tempid = "" Then  
            datPrimaryRS.Recordset.Find "concode=" + CStr(val(tempid))  
            cmdAdd.Enabled = False  
            cmdUpdate.Enabled = False  
            cmdDelete.Enabled = False  
            cmdRefresh.Enabled = False
```

```
        End If  
    End If
```

```
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
```

```
    Screen.MousePointer = vbDefault  
End Sub
```

```
Private Sub datPrimaryRS_Error(ByVal ErrorNumber As Long, Description As String,  
    ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String, ByVal  
    HelpContext As Long, fCancelDisplay As Boolean)
```

```
    'This is where you would put error handling code  
    'If you want to ignore errors, comment out the next line  
    'If you want to trap them, add code here to handle them  
    MsgBox "Data error event hit err:" & Description  
End Sub
```

```
Private Sub datPrimaryRS_MoveComplete(ByVal adReason As  
    Adodb.EventReasonEnum, ByVal pError As Adodb.Error, adStatus As  
    Adodb.EventStatusEnum, ByVal pRecordset As Adodb.Recordset)
```

```
    'This will display the current record position for this recordset  
    datPrimaryRS.Caption = "Record: " & CStr(datPrimaryRS.Recordset.AbsolutePosition)  
End Sub
```

```
Private Sub datPrimaryRS_WillChangeRecord(ByVal adReason As  
    Adodb.EventReasonEnum, ByVal cRecords As Long, adStatus As  
    Adodb.EventStatusEnum, ByVal pRecordset As Adodb.Recordset)
```

```
    'This is where you put validation code  
    'This event gets called when the following actions occur
```

AUTHETICATION USING BIOMETRICS

Dim bCancel As Boolean

```
Select Case adReason
Case adRsnAddNew
Case adRsnClose
Case adRsnDelete
Case adRsnFirstChange
Case adRsnMove
Case adRsnRequery
Case adRsnResynch
Case adRsnUndoAddNew
Case adRsnUndoDelete
Case adRsnUndoUpdate
Case adRsnUpdate
End Select
```

```
If bCancel Then adStatus = adStatusCancel
End Sub
```

```
Private Sub cmdAdd_Click()
cmdAdd.Enabled = False

On Error GoTo AddErr
datPrimaryRS.Recordset.AddNew
```

```
Exit Sub
AddErr:
MsgBox Err.Description
End Sub
```

```
Private Sub cmdDelete_Click()
On Error GoTo DeleteErr
With datPrimaryRS.Recordset
.Delete
.MoveNext
If .EOF Then .MoveLast
End With
Exit Sub
DeleteErr:
MsgBox Err.Description
End Sub
```

```
Private Sub cmdRefresh_Click()
'This is only needed for multi user apps
On Error GoTo RefreshErr
datPrimaryRS.Refresh
```

AUTHETICATION USING BIOMETRICS

```
Exit Sub
RefreshErr:
  MsgBox Err.Description
End Sub
```

```
Private Sub cmdUpdate_Click()
  datPrimaryRS.Recordset.UpdateBatch adAffectAll
  cmdAdd.Enabled = True
  MsgBox "Saved Successfully"
```

```
  Did = txtFields(0).Text
  If clkform = False Then
    clkform = True
  Did = txtFields(0).Text
```

```
  MainForm.Enrollment.Enabled = True
  Me.Hide
  Unload Me
  Call MainForm.callenroll
End If
```

```
  On Error GoTo UpdateErr
```

```
Exit Sub
UpdateErr:
  MsgBox Err.Description
End Sub
```

```
Private Sub cmdClose_Click()
  Unload Me
```

VERIFYING FINGERPRINT PARAMETERS:

```

Public Const CntStr As String = "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security
Info=False;Data Source=data.mdb"
Public datacnt As New Adodb.Connection
Public data As New Adodb.Recordset
Public md() As FPMemBaseElement
Public mdCount As Long
Public mdAlloc As Long
' VeriFinger Parameters
' Recognition threshold
'Desired FAR : 0 = 0.01%, 1 = 0.001%, 2 = 0.0001% for every Fingerprints matching
speed
'{{{25,35,41},
' {33,40,60},
' {33,40,60},
' {33,40,60},
' {40,50,60},
' {70,100,120}}};
Public RECOGNITION_THRESHOLD_TABLE(0 To 2, 0 To 5) As Long
Public RECOGNITION_THRESHOLD As Long
' Features extraction filtering quality
Public Const EXTRACTION_QUALITY As Long = 3
' Fingerprints matching speed
Public Const FINGEPRINTS_MATCHING_SPEED As Long = 5
' Usage of orientational image
Public Const USE_BLOCKED_ORIENTATION As Long = True

Public Const Ratio As Long = 100
Public Const Maximal_Rotation As Long = 360

Private ii As Long, ID As Long, direction As Byte, RecordCount As Long

Public Function FillMemBase() As Long
    Dim C(0 To 255) As Long, G As Long
    Dim i As Long
    mdCount = 0
    If data.EOF = True Then Exit Function
    For i = 0 To 255
        C(i) = 0
    Next i

    data.MoveFirst
    Do While data.EOF = False
        mdCount = mdCount + 1
        G = data!G
    
```

AUTHETICATION USING BIOMETRICS

```

    C(G) = C(G) + 1
    data.MoveNext
Loop
mdAlloc = mdCount

For i = 1 To 255
    C(i) = C(i) + C(i - 1)
Next i
For i = 0 To 255
    C(i) = C(i) - 1
Next i

ReDim md(0 To mdAlloc - 1)

Dim counter As Long
Dim result As Long
data.MoveFirst
Do While data.EOF = False
    G = data!G
    counter = C(G)
    md(counter).ID = LoadFP(md(counter).fpd, md(counter).fpc)
    If result = -1 Then
        FillMemBase = 0
        ReDim md(0)
        Exit Function
    End If
    C(G) = C(G) - 1
    data.MoveNext
Loop
End Function
' VFGetParameters params

' params.EXTRACTION_QUALITY = EXTRACTION_QUALITY
' params.FINGERPRINTS_MATCHING_SPEED =
FINGERPRINTS_MATCHING_SPEED
' params.USE_BLOCKED_ORIENTATION = USE_BLOCKED_ORIENTATION
'
' VFSetParameters params
'
    RECOGNITION_THRESHOLD = RECOGNITION_THRESHOLD_TABLE(0,
FINGERPRINTS_MATCHING_SPEED)
End Sub

' Stores Fingerprint data in database using MS ADO
' (only present features are stored in order to reduce record size)
Public Function StoreFP(fpd As FPDATA, fpc As FPCLASSSDATA) As Long

```

AUTHETICATION USING BIOMETRICS

```
Dim dc As New Connection
Dim d As New Recordset
```

```
If fpd.features_number = 0 Then
    StoreFP = -1
    Exit Function
End If
```

```
' open server side connection for database update
d.CursorLocation = adUseServer
d.Open "data", datacnt, adOpenDynamic, adLockOptimistic, adCmdTable
' Add fingerprint features collection to the fingerprint database
Dim Barray() As Byte
d.AddNew
d!G = fpc.G
d!SpCount = fpc.SpCount
If fpc.SpCount <> 0 Then 'Checks if Singular points are present
    ToByteArray Barray, fpc.SpX, fpc.SpCount
    d!SpX = Barray
    ToByteArray Barray, fpc.SpY, fpc.SpCount
    d!SpY = Barray
    ToByteArray Barray, fpc.SpD, fpc.SpCount
    d!SpD = Barray
    ToByteArray Barray, fpc.SpT, fpc.SpCount
    d!SpT = Barray
End If
```

```
d!fn = fpd.features_number
d!dirscount = fpd.dirs_block_count
ToByteArray Barray, fpd.feature_coordinatex, fpd.features_number
d!fx = Barray
ToByteArray Barray, fpd.feature_coordinatey, fpd.features_number
d!fy = Barray
ToByteArray Barray, fpd.feature_direction, fpd.features_number
d!fd = Barray
ToByteArray Barray, fpd.feature_curvature, fpd.features_number
d!fc = Barray
d!Dirs = fpd.Dirs
d.Update
StoreFP = d!ID
```

```
End Function
```

```
' Gets the first fingerprint index with best G matching
Public Function GetFirst(fpc As FPCLASSDATA) As Long
    Dim count As Long
```

AUTHENTICATION USING BIOMETRICS

```
Dim y As Long, d1 As Long, d As Long
count = mdCount - 1
If count < 1 Then
    GetFirst = count
    ID = count
    ii = count
    direction = 2
    Exit Function
End If
d = Abs(md(0).fpc.G - fpc.G)
For y = 1 To count
    d1 = Abs(md(y).fpc.G - fpc.G)
    If d1 > d Then Exit For
    d = d1
Next y
ID = y - 1
ii = ID
GetFirst = ID
If ID = 0 Then
    direction = 3
ElseIf ID = count Then
    direction = 2
Else
    direction = 0
End If
End Function

'Gets next (nearest by G) fingerprint index
Public Function GetNext() As Long
    Select Case direction

        Case 0 ' go backward
            ID = ID - 1
            GetNext = ID
            If ID = 0 Then direction = 3
            Else: direction = 1

        Case 1 ' go forward
            ii = ii + 1
            GetNext = ii
            If ii = mdCount - 1 Then direction = 2 Else
            : direction = 0

        Case 2: ' end of the database is reached so go only backward
            ID = ID - 1
            GetNext = ID
```

AUTHETICATION USING BIOMETRICS

```
Case 3: ' begining of the database is reached so go only forward
    ii = ii + 1
    If ii = mdCount Then GetNext = -1 _
    Else: GetNext = ii
End Select
End Function

' Reads Fingerprint data from database
Public Function GetFP(indx As Long, tfpd As FPDATA, tfpc As FPCLASSDATA) As
Long
    CopyFPD tfpd, md(indx).fpd, 0
    CopyFPC tfpc, md(indx).fpc, 0
    GetFP = md(indx).ID
End Function

' Loads fingerprint from database to memory
Public Function LoadFP(tfpd As FPDATA, tfpc As FPCLASSDATA) As Long
    Dim Barray() As Byte
    tfpc.G = data!G

'Singular points, if any exist
    tfpc.SpCount = data!SpCount
    If tfpc.SpCount > 0 Then
        ReDim tfpc.SpD(0 To tfpc.SpCount - 1)
        ReDim tfpc.SpX(0 To tfpc.SpCount - 1)
        ReDim tfpc.SpY(0 To tfpc.SpCount - 1)
        ReDim tfpc.SpT(0 To tfpc.SpCount - 1)
        Barray = data!SpX
        ToLongArray tfpc.SpX, Barray, tfpc.SpCount
        Barray = data!SpY
        ToLongArray tfpc.SpY, Barray, tfpc.SpCount
        Barray = data!SpD
        ToLongArray tfpc.SpD, Barray, tfpc.SpCount
        Barray = data!SpT
        ToLongArray tfpc.SpT, Barray, tfpc.SpCount
    End If

'Features
    tfpd.features_number = data!fn
    tfpd.dirs_block_count = data!dirscount
    If tfpd.features_number > 0 Then
        ReDim tfpd.feature_coordinatex(0 To tfpd.features_number)
        ReDim tfpd.feature_coordinatey(0 To tfpd.features_number)
        ReDim tfpd.feature_direction(0 To tfpd.features_number)
        ReDim tfpd.feature_curvature(0 To tfpd.features_number)
```

AUTHETICATION USING BIOMETRICS

```
ReDim tfpd.Dirs(0 To 4095)
Barray = data!fx
ToLongArray tfpd.feature_coordinatex, Barray, tfpd.features_number
Barray = data!fy
ToLongArray tfpd.feature_coordinatey, Barray, tfpd.features_number
Barray = data!fd
ToLongArray tfpd.feature_direction, Barray, tfpd.features_number
Barray = data!fc
ToLongArray tfpd.feature_curvature, Barray, tfpd.features_number
tfpd.Dirs = data!Dirs
LoadFP = data!ID
Else: LoadFP = -1
End If
End Function

'Verifies fingerprint with database
Public Function Verify(ID As Long, RecThreshold As Long, tfpd As FPDATA, tfpc As
FPCLASSDATA) As Long
Dim Barray() As Byte

If data.EOF = True And data.BOF = True Then
Verify = -1
Exit Function
End If

data.MoveFirst
Do While data.EOF = False
If ID = data!ID Then Exit Do
data.MoveNext
Loop
If data.EOF = True Then
Verify = -1
Exit Function
End If

Dim sfpd As FPDATA
Dim sfpc As FPCLASSDATA
ReDim sfpd.feature_coordinatex(0 To 999)
ReDim sfpd.feature_coordinatey(0 To 999)
ReDim sfpd.feature_direction(0 To 999)
ReDim sfpd.feature_curvature(0 To 999)
ReDim sfpd.Dirs(0 To 4095)
ReDim sfpc.SpX(0 To 99)
ReDim sfpc.SpY(0 To 99)
ReDim sfpc.SpD(0 To 99)
ReDim sfpc.SpT(0 To 99)
```

AUTHETICATION USING BIOMETRICS

```
sfdc.G = data!G  
sfdc.SpCount = data!SpCount
```

```
If sfdc.SpCount <> 0 Then  
  Barray = data!SpX  
  ToLongArray sfdc.SpX, Barray, sfdc.SpCount  
  Barray = data!SpY  
  ToLongArray sfdc.SpY, Barray, sfdc.SpCount  
  Barray = data!SpD  
  ToLongArray sfdc.SpD, Barray, sfdc.SpCount  
  Barray = data!SpT  
  ToLongArray sfdc.SpT, Barray, sfdc.SpCount  
End If
```

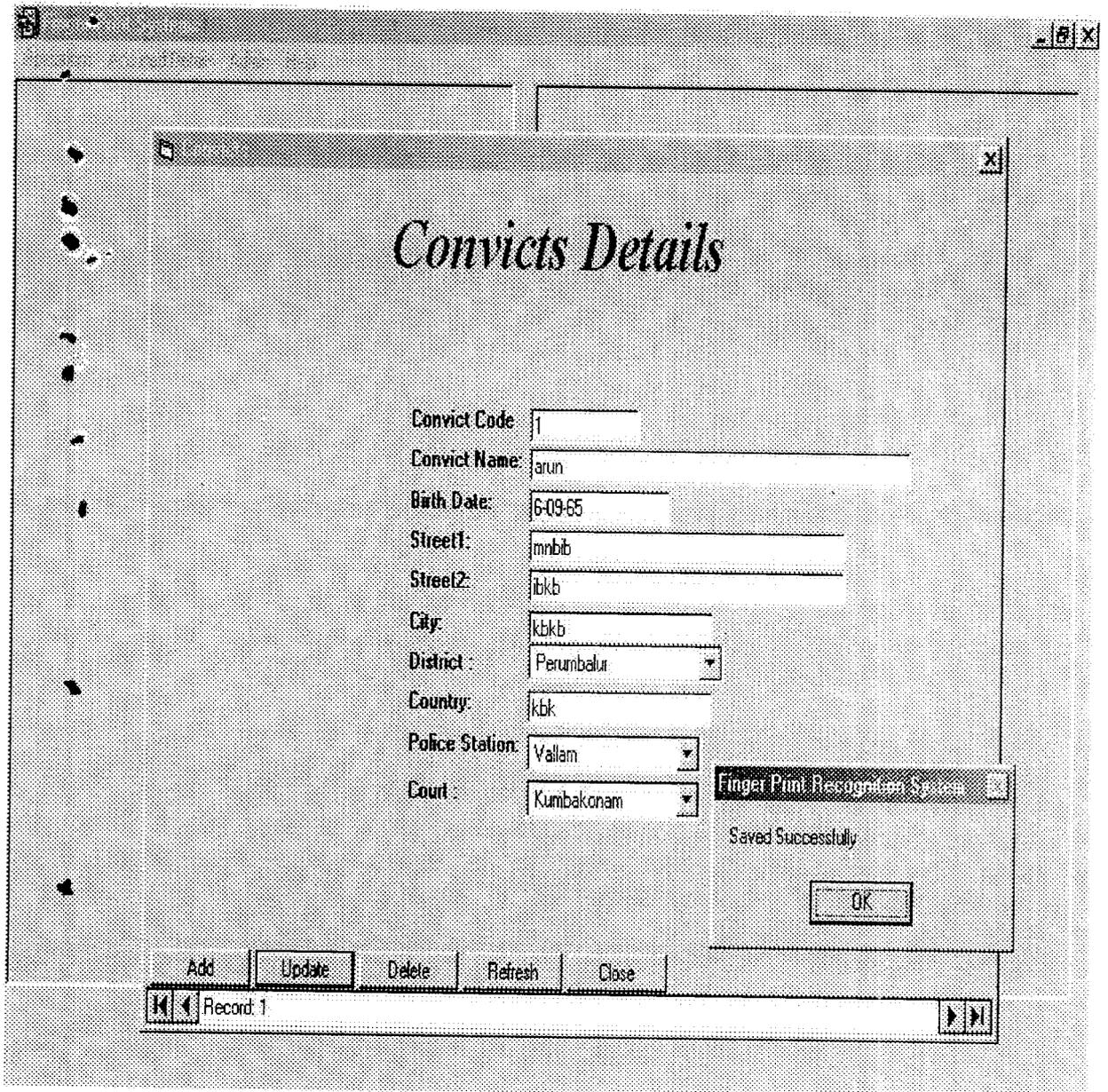
```
sfpd.features_number = data!fn  
sfpd.dirs_block_count = data!dirscount  
Barray = data!fx  
ToLongArray sfpd.feature_coordinatex, Barray, sfpd.features_number  
Barray = data!fy  
ToLongArray sfpd.feature_coordinatey, Barray, sfpd.features_number  
Barray = data!fd  
ToLongArray sfpd.feature_direction, Barray, sfpd.features_number  
Barray = data!fc  
ToLongArray sfpd.feature_curvature, Barray, sfpd.features_number  
sfpd.Dirs = data!Dirs  
Dim mode  
mode = 2
```

```
Dim oldspeed  
Dim params As VF_PARAMETERS  
' VFGetParameters params  
oldspeed = params.FINGERPRINTS_MATCHING_SPEED  
params.FINGERPRINTS_MATCHING_SPEED = 0  
' VFSetParameters params  
  
' Verify = FingerprintsMatching(  
    tfpd, sfpd, tfpc, sfdc, Maximal_Rotation, _  
    RecThreshold, mode, 0)  
  
params.FINGERPRINTS_MATCHING_SPEED = oldspeed  
' VFSetParameters params
```

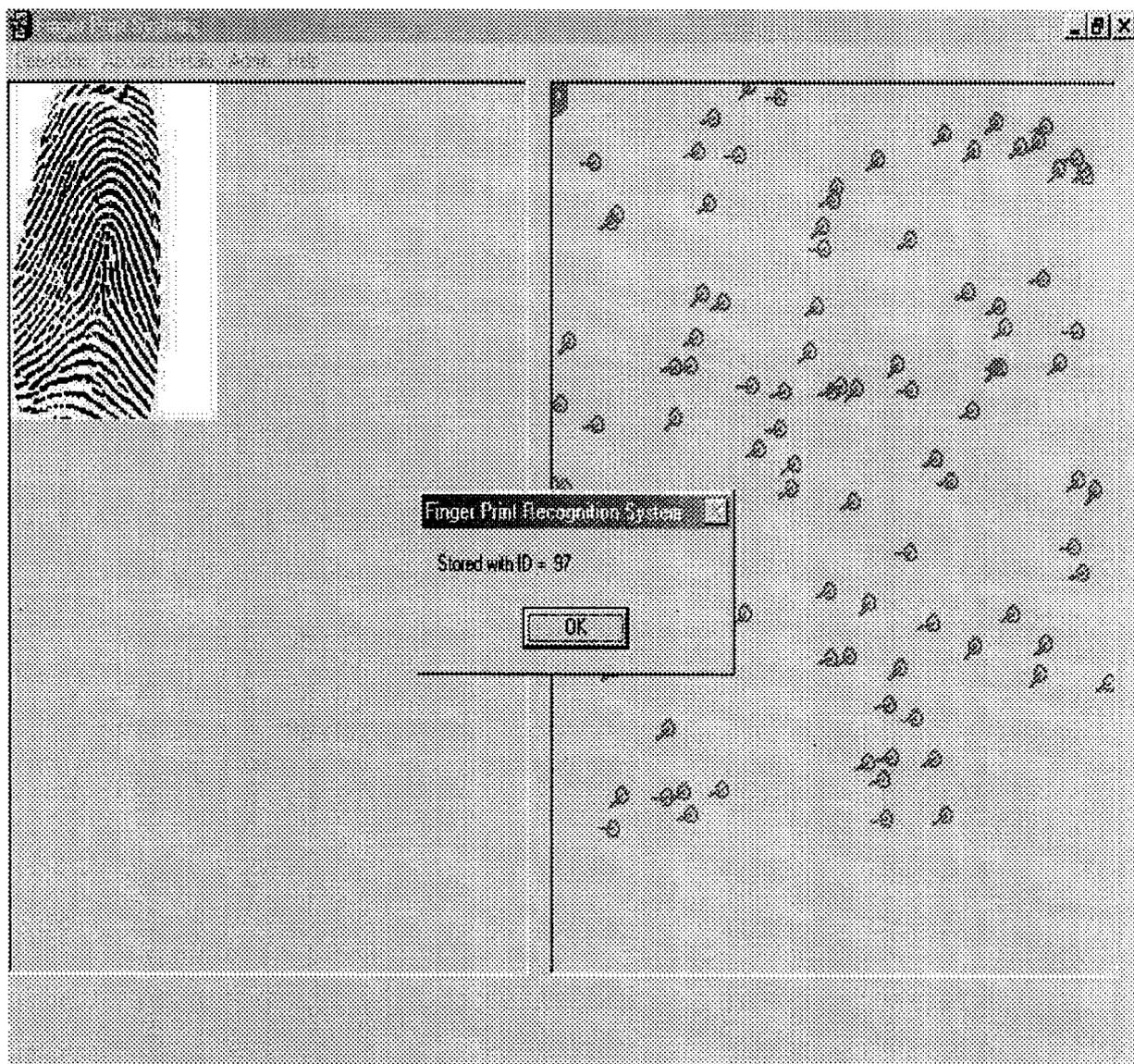
```
End Function
```

9.2 SAMPLE SCREENS:

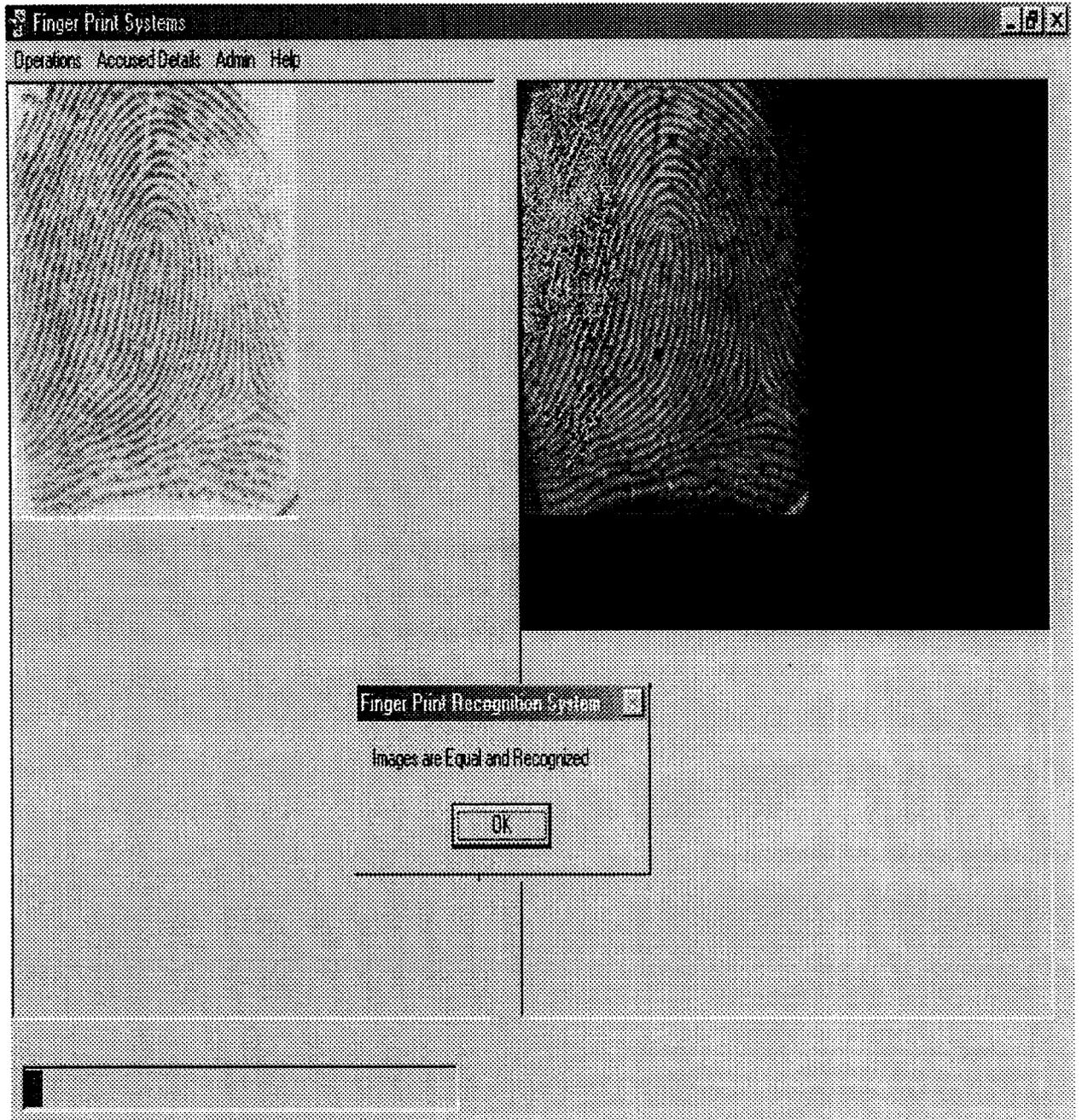
CONVICT DETAILS SCREEN



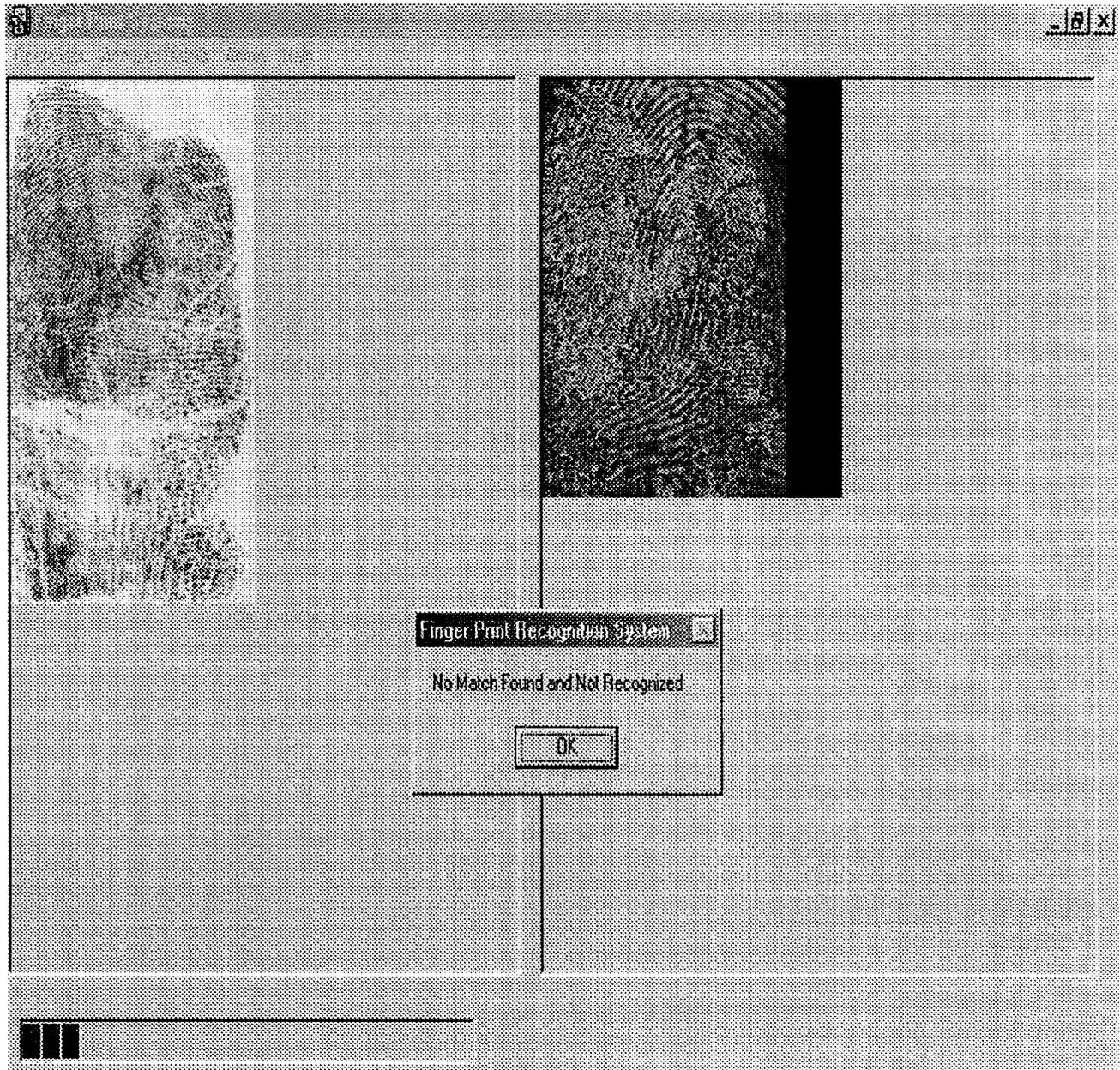
ENROLLMENT SCREEN



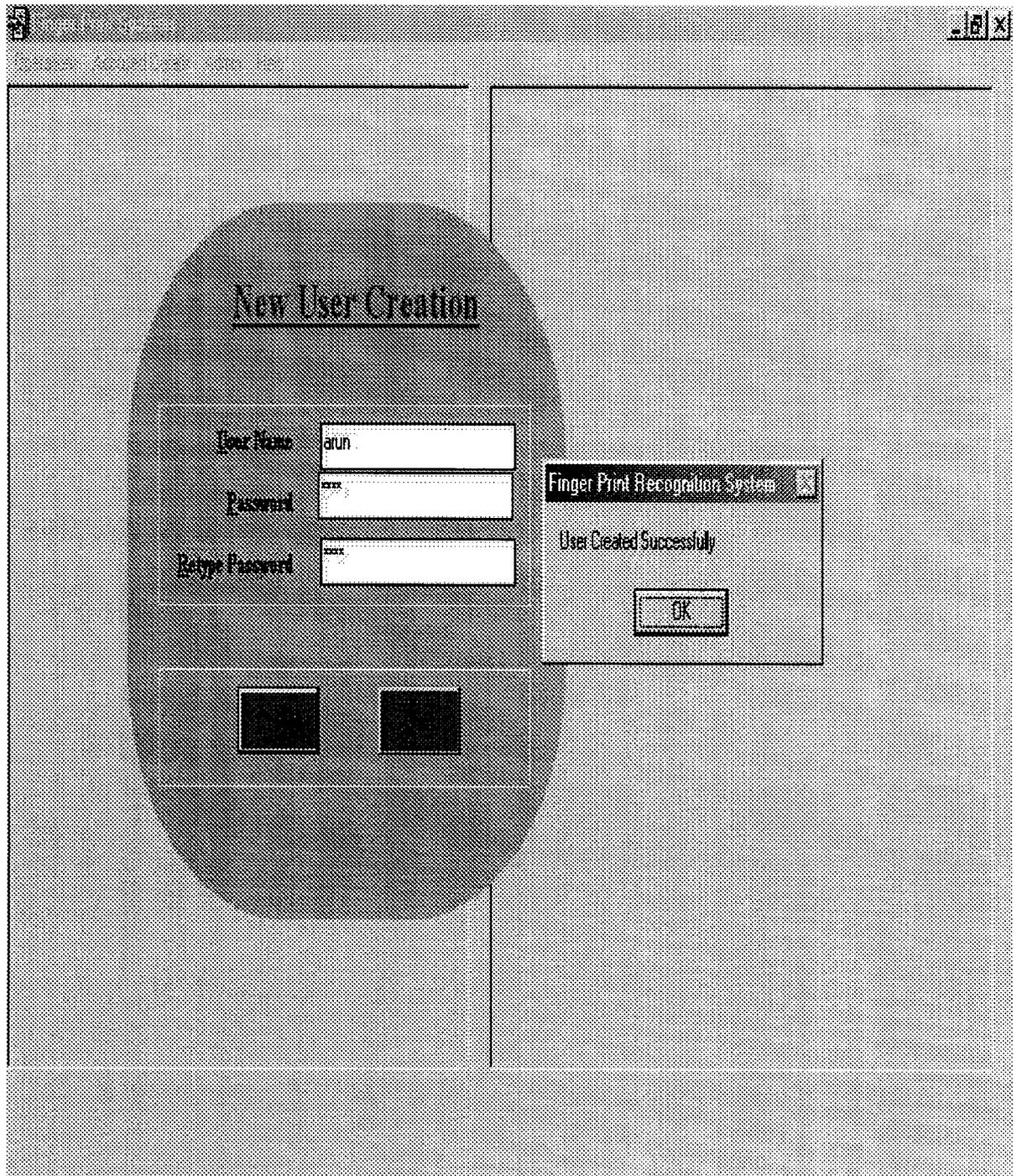
**RECOGNITION SCREEN
(MATCH FOUND)**



**RECOGNITION SCREEN
(NO MATCH FOUND)**



SCREEN TO ADD USERS



SCREEN TO LIST USERS

