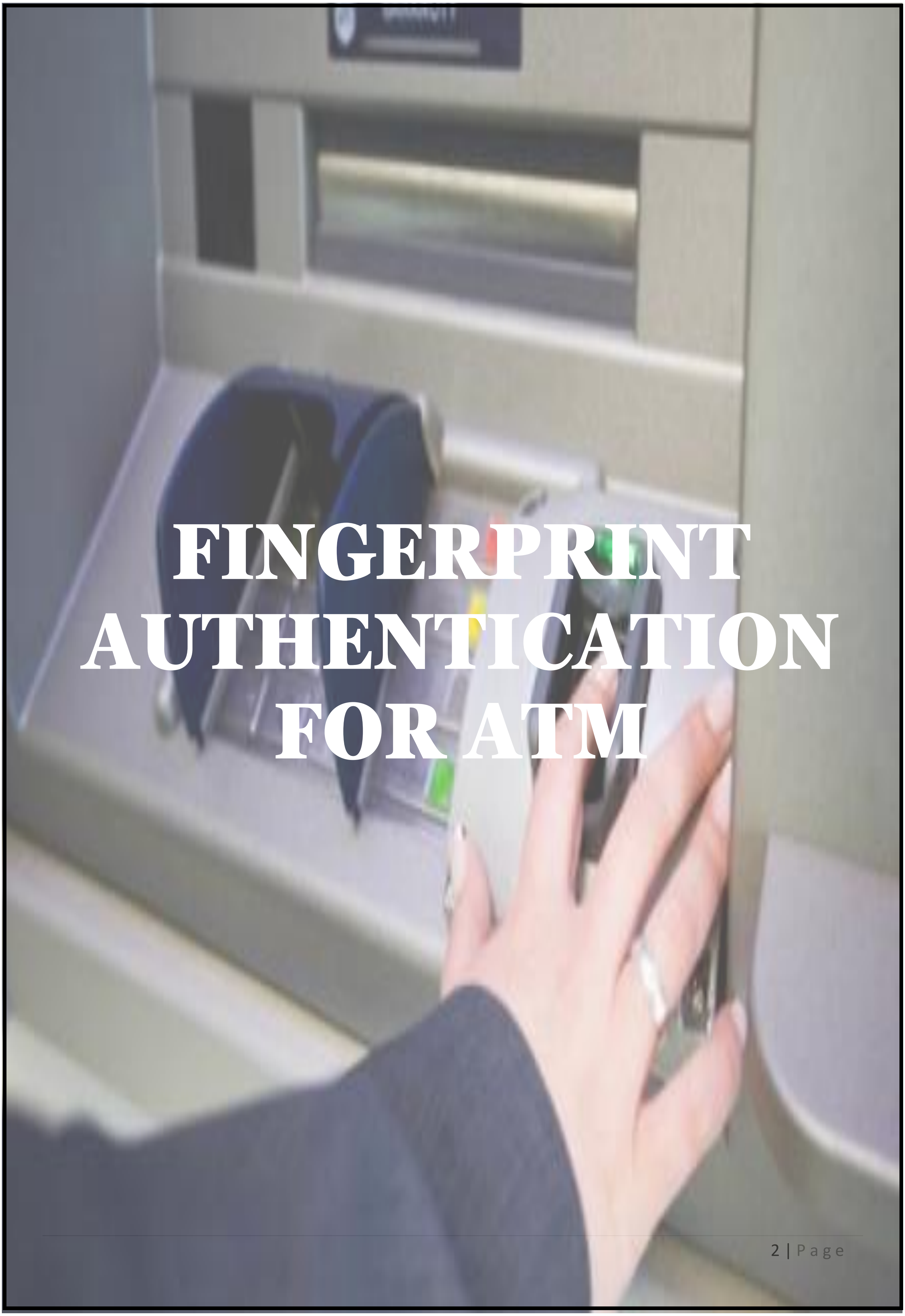
DELHI TECHNOLOGICAL UNIVERSITY

(DEPARTMENT OF APPLIED MATHEMATICS)

SOFTWARE ENGINEERING (MC-310)
INNOVATIVE PROJECT

SUBMITTED TO:
Mr. Jamkhongam Touthang

SUBMITTED BY:
Shubham Kumar (2K18/MC/111)



FINGERPRINT AUTHENTICATION FOR ATM

TABLES OF CONTENT

1. INTRODUCTION	**13**
1.1. PROBLEM STATEMENT	**13**
1.2. OBJECTIVE	**13**
1.3. DISCRIPRION OF FINGERPRINT	**13**
1.4. WHAT IS FINGERPRINT?	**14**
1.5. WHAT IS FINGERPRINT AUTHENTICATION	**15**
1.6. APPROCHES FOR FINGERPRINT RECOGINITATION	**16**
2. SOFTWARE REQUIREMENT AND SPECIFICATION	**17**
2.1. INTRODUCTION	**17**
2.2. OVERALL DESCRIPTION	**17**
2.3. FUNCTIONAL REQUIREMENT	**18**
2.4. NON-FUNCTIONAL REQUIREMENT	**19**
2.5. FRONT END DISCRIPTION	**19**
2.6. BACK END DISPRIPRTION	**20**
2.7. DATABASE DESCRIPRION	**20**
2.8. HARDWARE REQUIREMENTS	**21**
2.9. SOFTWARE REQUIREMENTS	**21**
3. DIAGRAMS	**22**
3.1. FLOW CHART	**22**
3.2. USE CASE DIAGRAM	**24**
3.3. E – R DIAGRAM	**25**
3.4. SEQUENCE DIAGRAM	**27**
3.5. ACTIVITY DIAGRAM	**28**

3.6. DATA FLOW DIAGRAM 30
3.6.1. DATA FLOW DIAGRAM LEVEL 0 30
3.6.2. DATA FLOW DIAGRAM LEVEL 1 31
3.6.3. DATA FLOW DIAGRAM LEVEL 2 31
4. SYSTEM DESIGN 32
4.1. OVERVIEW OF SYSTEM DESIGN 32
4.2. SYSTEM LEVEL DESIGN 32
4.3. ALGORITHM DESIGN 33
5. FINGERPRINT IMAGE PROCESSING 35
5.1. STEPS INVOLVED IN FINGERPRINT IMAGE PROCESSING 35
5.2. IMAGE ACQUISITION 36
5.3. RIDGE ORIENTATION 36
5.4. THINNING 36
5.5. BINARIZATION 38
5.6. MINUTIAE EXTRACTION 39
5.7. MINUTIAE TRIPLET MATCHING 40
6. SOFWARE & HARDWARE DISCRIPTION 41
6.1. MICROSOFT VISUAL STUDIO 41
6.1.1. INTRODUCTION 41
6.1.2. FEATURES 42
6.1.3. WHY MICROSOFT VISUAL STUDIO? 43
6.2. MICROSOFT SQL SERVER 2008 R2 43
6.2.1. INTRODUCTION 43
6.2.2. SERVICES 44
6.2.3. MICROSOFT SQL SERVER 2008 R2 FEATURES 45

6.2.4. WHY MICROSOFT SQL SERVER? 45

6.2.5. DATABASE DESCRIPTION 46

6.3. MATLAB R2012B 46

6.3.1. INTRODUCTION 46

6.3.2. FEATURES 47

6.3.3. WHY MATLAB? 47

6.4. SECUGEN FINGERPRINT READER 48

6.4.1. INTRODUCTION 48

6.4.2. FEATURES 49

6.4.3. WHY SECUGEN FINGERPRINT SCANNER 49

6.4.4. IMAGE DESCRIPTION 49

7. SOFTWARE SNAPSHOT 50

7.1. INTERFACESOFTWARE 50

7.2. DATABASE SOFTWARE 51

7.3. IMAGE PROCESSING SOFTWARE 53

7.4. FINGERPRINT SENSOR SDK 54

8. PROJECT MODEL 55

8.1. GRAPHICAL USER INTERFACE 55

8.2. DATABASE MODEL 57

8.3. MATLAB GUI 59

9. METHODOLOGY 62

9.1. FEASIBLE STUDY 62

9.2. TARGET 62

9.3. STEP-BY-STEP DESIGNING 62

9.4. TESTING 63

9.5. RESULT **63**

10. SUMMARY **64**

10.1. ADVANTAGES **64**

10.2. DISADVANTAGES **64**

10.3. CONCLUSION **65**

11. REFERENCES **66**

LIST OF FIGURES

S.NO	TITLE	PAGE
1.1	FINGERPRINT	14
1.2.1	FINGERPRINT	14
1.2.2	MINUTIA	15
1.3	VERIFICATION VS IDENTIFICATION	15
3.1	FLOW CHART SYMBOLS & FUNCTIONS	22
3.2	FLOW CHART DIAGRAM	23
3.3	USE CASE SYMBOLS & FUNCTIONS	24
3.4	USE CASE DIAGRAM	24
3.5	E-R DIAGRAM SYMBOLS & DESCRIPTION	25
3.6	ENTITY RELATIONSHIP DIAGRAM	26
3.7	SEQUENCE DIAGRAM	27
3.8	ACTIVITY DIAGRAM SYMBOLS	28
3.9	ACTIVITY DIAGRAM	29
3.10	DATA FLOW DIAGRAM SYMBOL	30
3.11	DFD LEVEL 0	30
3.12	DFD LEVEL 1	31
3.13	DFD LEVEL 2	31
4.1	SYSTEM DESIGN OVERVIEW	32
4.2	SIMPLIFIED FIGERPRINT RECOGNITION SYSTEM	33
4.3	MINUTIA EXTRACTOR	33
4.4	MINUTIA MATCHER	34
5.1	STEPS OF FINGERPRINT RECOGNITION	35
5.2	CLASSIFICATION OF THE THINNING ALGORITHM	36
5.3	RESULT OF THE THINNIG ALGORITHM	38
5.4	FINGERPRINT IMAGE AFTER BINARIZATION	39
5.5	RIDGE ENDING & RIDGE BIFURCATION	39

LIST OF TABLES

S.NO	TITLE	PAGE
6.1	MICROSOFT VISUAL STUDIO ADVANTAGE	43
6.2	MICROSOFT SQL SERVER ADVANTAGE	45
6.3	DATABASE DESCRIPTION	46
6.4	MATLAB ADVANTAGE	47
6.5	SECUGEN SCANNER ADVANTAGE	49
6.6	IMAGE DESCRIPTION	49

ACKNOWLEDGEMENT

I wish to express my heartfelt gratitude to my Professor, Mr. Jamkhongam Touthang for his active interest, constructive guidance, and advice during every stage of this work. His valuable guidance coupled with active and timely review of my work provided the necessary motivation for us to work on and successfully complete thesis.

We are also heartily thankful to Fleek IT Solutions, Greater Noida (U.P.), for providing me testing facilities in LAB.

It is the contribution of many persons that make a work successful. I wish to express our gratitude to individuals who have contributed their ideas, time, and energy in this work.

Words are inefficient to express our gratitude to family members for their inspirations, blessings, and support. I thank God, who has supported us at every moment.

ABSTRACT

My Project is to develop the technique for fingerprint authentication in ATM. This target can be mainly decomposed into image preprocessing, feature extraction and feature match. For each sub-task, some classical and up-to-date methods in literatures are analyzed. Based on the analysis, an integrated solution for fingerprint recognition and authentication is developed for demonstration.

My demonstration program is coded in **ASP.NET & MATLAB**. For the program, some optimization at coding level and algorithm level are proposed to improve the performance of my fingerprint authentication system. These performance enhancements are shown by experiments conducted upon a variety of fingerprint images. Also, the experiments illustrate the key issues of fingerprint recognition that are consistent with what the available literatures say

The underlying principle is the phenomenon of biometric “AUTHENTICATION”. In this project, I propose a method for fingerprint matching based on minutiae matching.

INTRODUCTION

1.1 PROBLEM STATEMENT

- In present scenario, traditional ATM system accepts only on the PIN CODE security system, enabling the other person rather than the owner to access the account very easily.
- This ensures that the Traditional ATM system is not fully secured.

1.2 OBJECTIVE

The objective of project is to provide biometric security through fingerprint authentication in ATM application. Also, the experiments illustrate the key issues of fingerprint recognition that are consistent with what the available literatures say.

The underlying principle is the phenomenon of biometrics "AUTHENTICATION", in this project, I propose a method for fingerprint matching based on minutiae matching.

1.3 DESCRIPTION OF FINGERPRINT

The fingerprint is arguably a person's most unique physical characteristic. While humans have had the innate ability to recognize and distinguish different fingerprints for millions of years, computers are just now catching up.

The twist of this software is that it can pick someone's fingerprint out of crowd, extract that fingerprint for the rest of the scene and compare it with database full of stored images.

For this software to work, it must know what a basic fingerprint looks like. Fingerprint recognition software is based on the ability to first recognize fingerprint, which is a technological feat, and then measure the various features of each fingerprint.

Figure 1.1

1.4 WHAT IS A FINGERPRINT?

A fingerprint is the feature pattern of one finger (Figure 1.2.1). It is believed with strong evidence that each fingerprint is unique. Each person has his own fingerprints with the permanent uniqueness. So, fingerprints have being used for identification and forensic investigation for a long time.

Figure 1.2.1

A fingerprint is composed of many ridges and furrows. These ridges and furrows present good similarities in each small local window, like parallelism and average width. However, shown by intensive research on fingerprint recognition, fingerprints are not distinguished by their ridges and furrows, but by Minutia, which are some abnormal points on the ridges (Figure 1.1.2). Among the variety of minutia types reported in literatures, two are mostly significant and in heavy usage one is called termination, which is the immediate ending of a ridge the other is called bifurcation, which is the point on the ridge from which two Branches derive.

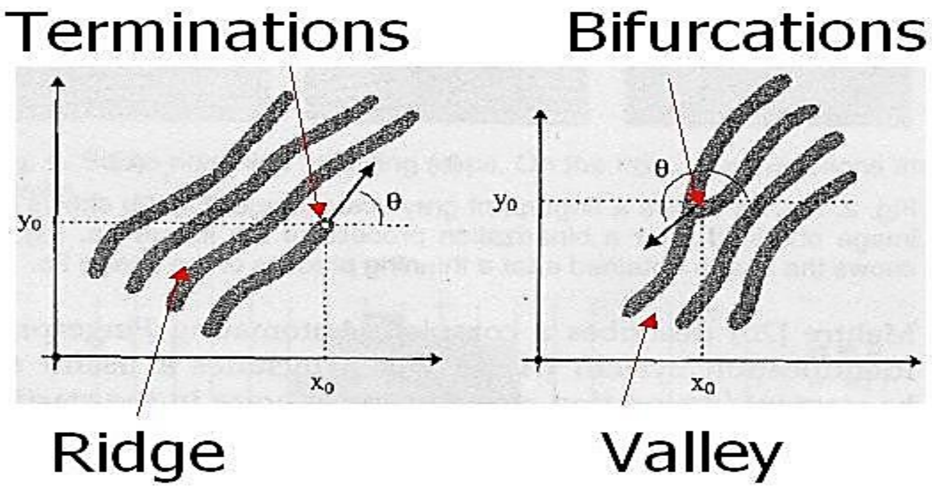


Figure 1.2.2 Minutia (Valley is also referred as Furrow, Termination is also called Ending, and Bifurcation is also called Branch)

1.5 WHAT IS FINGERPRINT AUTHENTICATION

The fingerprint authentication problem can be grouped into two sub-domains. One is fingerprint verification and the other is fingerprint identification (Figure 1.3). In addition, different from the manual approach for fingerprint authentication by experts, the fingerprint authentication here is referred as FAA (Fingerprint Authentication in ATM), which is program based.

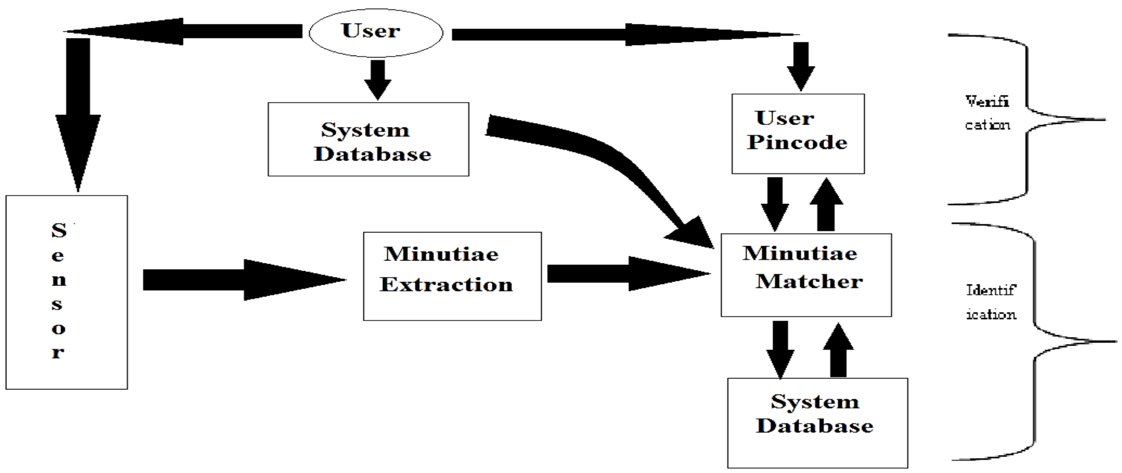


Figure 1.3 Verification vs. Identification

Fingerprint verification is to verify the authenticity of one person by his fingerprint. The user provides his fingerprint together with his identity information like his PIN-CODE. The fingerprint verification system retrieves the fingerprint template according to the PIN-CODE and matches the template with the real time acquired fingerprint from the user. Usually, it is the underlying design principle of AFAS (Automatic Fingerprint Authentication System). Fingerprint identification is to specify one person's identity by his fingerprint(s). Without knowledge of the person's identity, the fingerprint identification system tries to match his fingerprint(s) with those in the whole fingerprint database. It is especially useful for criminal investigation cases. And it is the design principle of AFIS (Automatic Fingerprint Identification System). However, all fingerprint recognition problems, either verification or identification, are ultimately based on a well-defined representation of a fingerprint.

1.6 APPROACHES FOR FINGERPRINT RECOGNITION

Two representation forms for fingerprints separate the two approaches for fingerprint recognition.

1. **Minutia - based:** The first approach, which is minutia-based, represents the fingerprint by its local features, like terminations and bifurcations. This approach has been intensively studied, also is the backbone of the current available fingerprint recognition products. I also concentrate on this approach in our project.
2. **Image-based:** The second approach, which uses image-based methods, tries to do matching based on the global features of a whole fingerprint image. It is an advanced and newly emerging method for fingerprint recognition. And it is useful to solve some intractable problems of the first approach. But our project does not aim at this method, so further study in this direction is not expanded in our thesis.

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

2.1 INTRODUCTION

- **Purpose**

  This Software Requirements Specification provides a complete description of all the functions and specifications of the ATM system of bank. The purpose is to provide extra security to the ATM systems.

- **Scope**

  The ATM system is designed to run for 24 hours and to allow bank clients to carry out transactions in a secured way. The data will be held in a bank database. The system is connected to the bank database using a modem.

- **Document Overview**

  The remainder of this document is two chapters, the first providing a full description of the project for the bank's ATM with fingerprint security. This SRS gives the details about the various requirements & about the various hardware & software interfaces.

2.2 OVERALL DESCRIPTION

The ATM system encompasses various GUI menus including the SENSOR, to provide high security. It provides secure access to the account of a customer. The ATM must be able to provide the following **services** to the customer:

- **Enter Pin:** A customer can enter the PIN Code for his/her Account.
- **Change Pin:** A customer must be able to change the pin linked to the card.
- **Enroll Finger:** A customer can enroll the fingerprint impression which has been used to provide security to the Account**.**
- **Change Fingerprint:** A customer can change the fingerprint impression.

2.3 FUNCTIONAL REQUIREMENT

The software to be designed will control a simulated automated teller machine (ATM) having:

- A sensor to enroll and detect fingerprint.
- A customer console (keyboard and display) for interaction with the customer. (for entering PIN Code)
- **Facility of Aborting transaction:** A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.
- **PIN Code format:** Pin should be of exactly 4 digits.
- **Facility of PIN Re-entering:** If the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed.
- **Denial of service, if PIN goes wrong:** If the customer is unable to successfully enter the PIN after three tries, the service would be denied for card.
- **Enroll finger:** Enroll your finger from first joint to the tip.
- **Facility of Re-enrolling the finger:** If the customer's Fingerprint is invalid, the customer will be required to re-enroll the Fingerprint before a transaction can proceed.
- **Denial of service, if fingerprint goes wrong:** If the customer is unable to successfully enroll fingerprint after three tries, the service would be denied for that card.
- **Explanation of problem:** If a transaction fails for any reason other than an invalid PIN and fingerprint, the ATM will display an explanation of the problem.

2.4NON FUNCTIONAL REQUIREMENT

There are requirements that are not functional in nature. Specifically, these are the constraints the system must follow. They are often called qualities of a system. Other terms for non-functional requirements are “constraints”,” quality attributes”,” quality goals”,” quality of service requirements” and “non-behavioral requirements”.

- **Scope:** The scope of this project is to allow the user to get access to their account through the ATM using fingerprinting functionality.

- **Functionality:** One customer at a time can process their account in the ATM machine.
- **Usability:** The desktop user interface shall be Windows XP/Vista/7 complaint.
- **Reliability:** The ATM machine must be able to scan or read the card and the fingerprint properly and identify the customer account.
- **Performance:** The ATM machine support only one customer at a time. The speed and accurate transaction decide the performance factor. The screen must be clearly visible to the user.
- **Security:** The pin number and the fingerprint in the card guarantee the security of a customer's account. The ATM system must not store any of this data in its database. The customer with a pin number and a valid card with valid fingerprint impression can do all transactions.

2.5 FRONT END DESCRIPTION

For developing the front – end interface, decided to use ASP.NET platform, with C# as the programming language, due to the following reasons:

- Easy to use and flexible interface.
- Several options for customizability.
- Proven to provide good performance and high reliability.
- Attractive and visually pleasing interface.

2.6 BACK-END DESCRIPTION

For the Image Processing (Minutiae Matching), I have decided to use MATLAB R2012b, due to the following reasons:

- Full compatibility with various databases.
- Easy to use and flexible interface.
- Word wide product for the image processing.

- Used by market-leading companies worldwide.
- Easy to code, and easy to convert the files In Dynamic Link Library (DLL) format.

For developing the database (back – end), I have decided to use MS-SQL Server 2008 R2 database, due to the following reasons:

- Native support and full compatibility with ASP.NET platform.
- Flexible, scalable, and robust database architecture.
- Used by market-leading companies worldwide.
- Strong data protection and ease of management.

2.7 DATABASE DESCRIPTION

Tools Used:

- Interface - Visual Studio 2010 Ultimate
- Database - MS SQL Server 2008 R2
- Image Processing - MATLAB R1012b

2.8 HARDWARE REQUIREMENT

- Processor - Pentium 4
- Hard Disk - 20GB
- RAM - 256 MB
- Sensor - Fingerprint Recognizer

2.9 SOFTWARE REQUIREMENT

- Operating System - Windows XP/Vista/7
- Database System - MS SQL Server 2008 R2
- Front End - Visual Studio 2010 Ultimate
- Framework - .Net Framework 4.0

DIAGRAMS

3.1 FLOW CHART

A **flowchart** is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This is a diagrammatic representation solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting, or managing a process or program in various fields.

Symbols	Name	Function
	Start/End	Indicates a start or end point
	Arrows	A connector that shows relationships between the representative shapes
	Data (Input/Output)	Represents input or output
	Process	Indicates an automated procedure
	Decision	Denotes a decision points with alternative paths

Figure 3.1 Flow chart symbols & functions

Flowcharts are used in designing and documenting complex processes or programs. Like other types of diagrams, they help visualize what is going on and thereby help the viewer to understand a process, and perhaps also find flaws, bottlenecks, and other less-obvious features within it. There are many different types of flowcharts, and each type has its own repertoire of boxes and notational conventions.

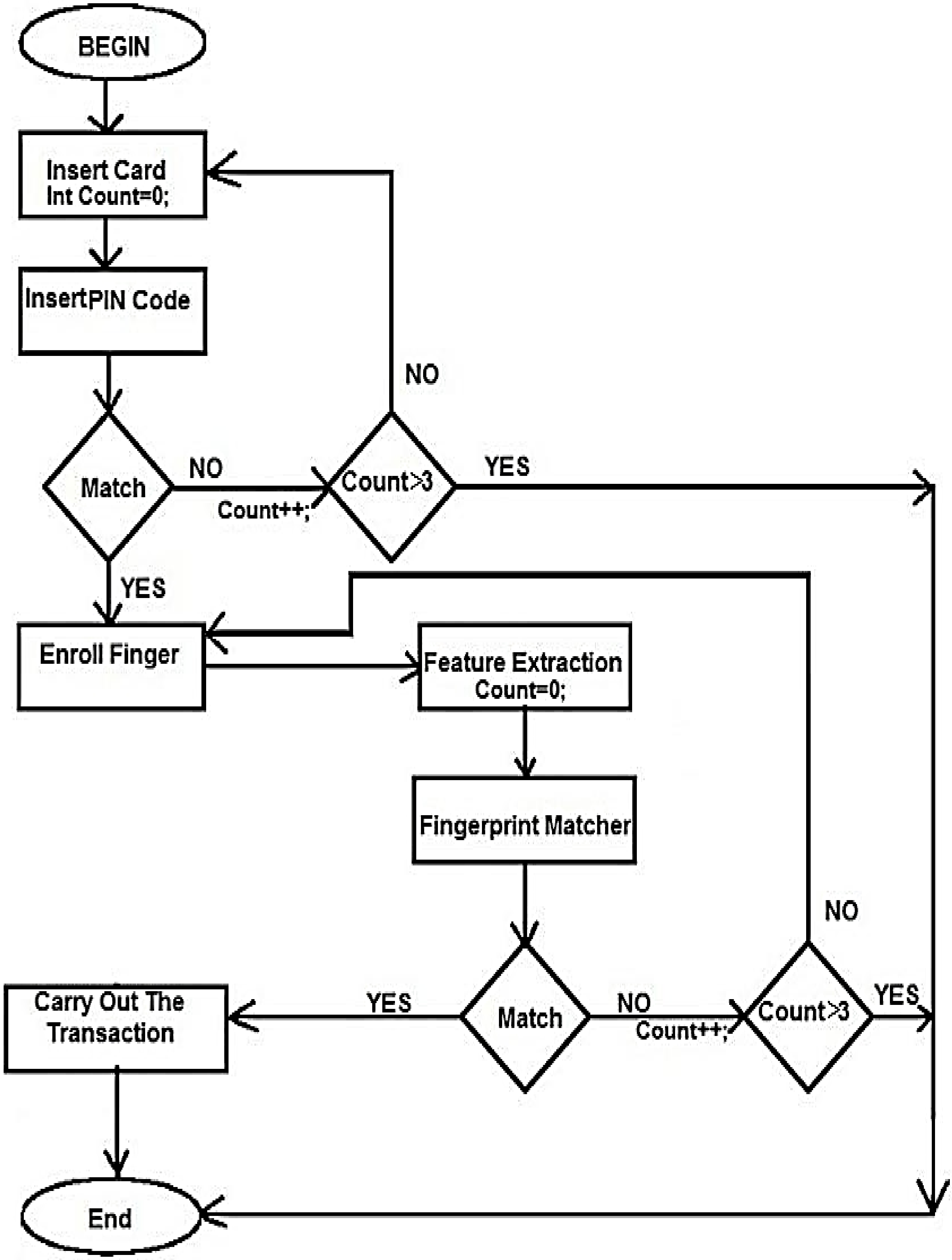


Figure 3.2 Flow chart diagram

3.2 USE CASE DIAGRAM

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

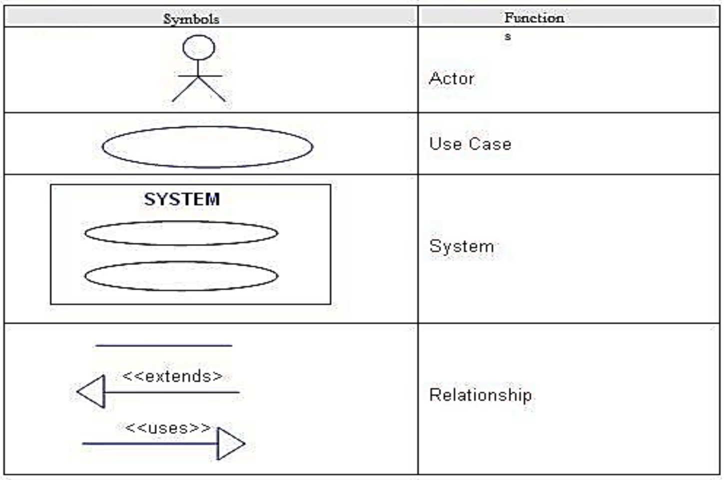


Figure 3.3 Use Case Symbols & Functions

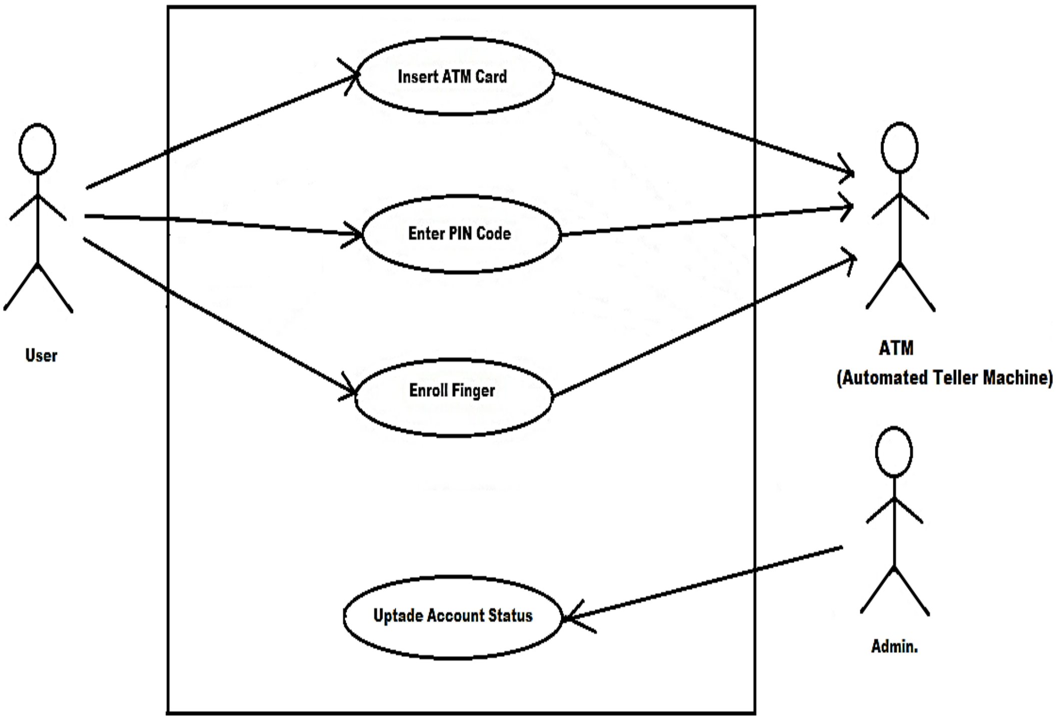


Figure 3.4 Use case diagram

3.3 E-R DIAGRAM

Entity–relationship model (**ER model**) is a data model for describing a database in an abstract way. In the case of a relational database, which stores data in tables, some of the data in these tables point to data in other tables - for instance, your entry in the database could point to several entries for each of the phone numbers that are yours. The ER model would say that you are an entity, and each phone number is an entity, and the relationship between you and the phone numbers is 'has a phone number'. Diagrams created to design these entities and relationships are called entity–relationship diagrams or ER diagrams.

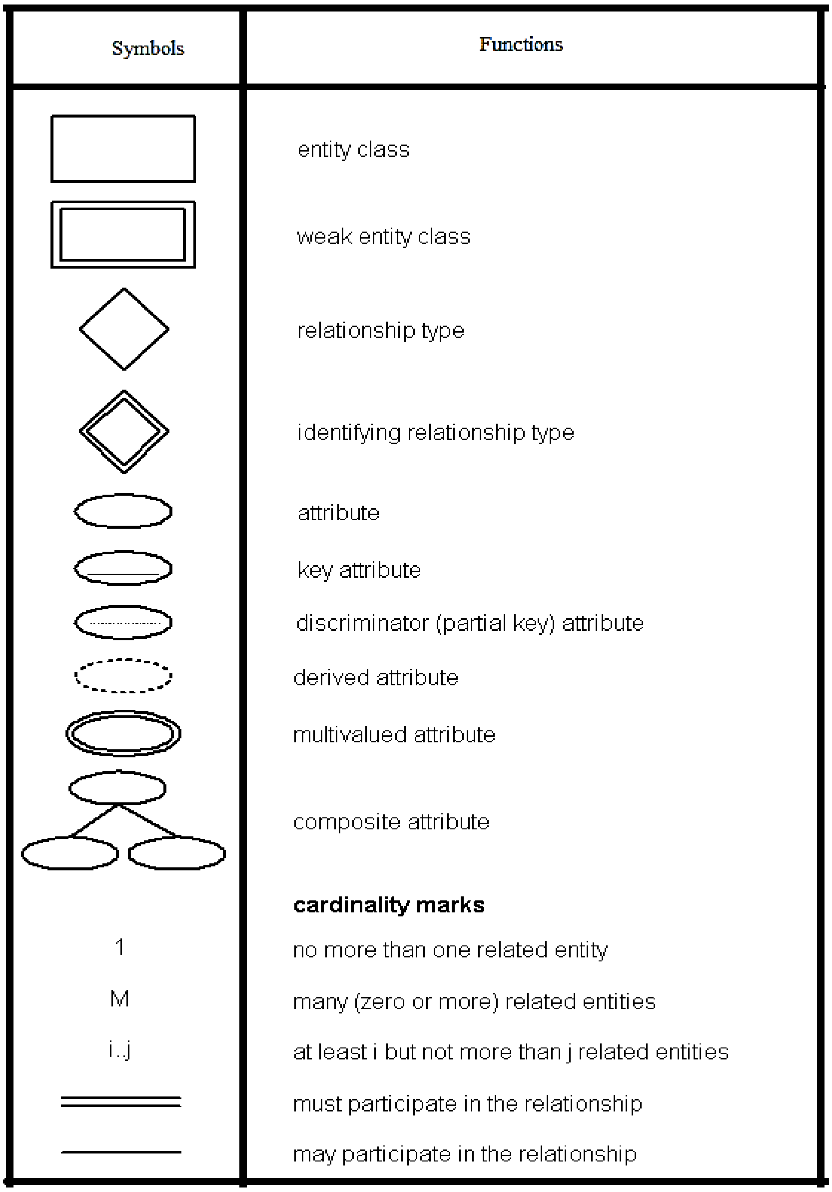


Figure 3.5 E-R diagram symbols & description

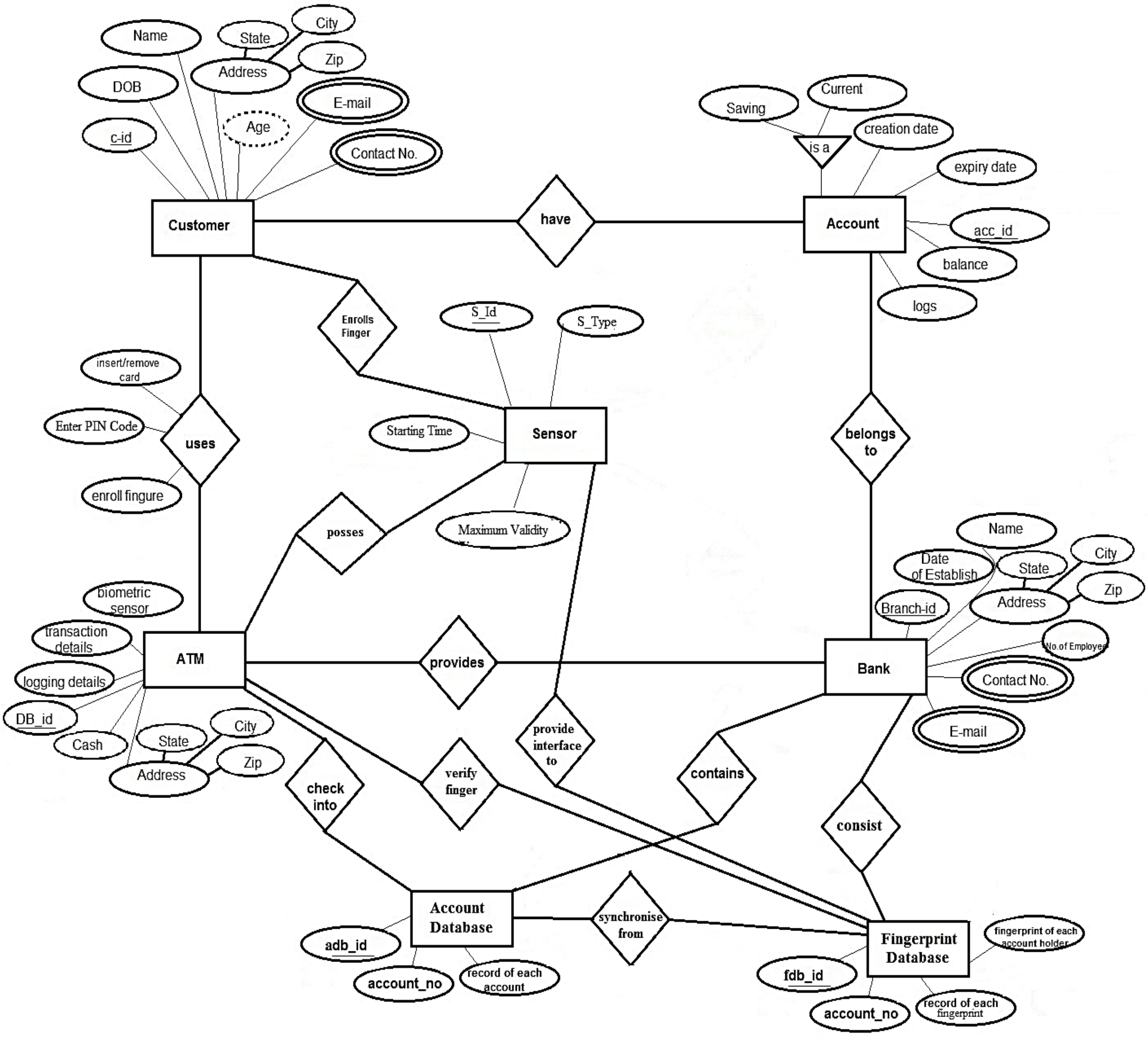


Figure 3.6 Entity Relationship Diagram

3.4 SEQUENCE DIAGRAM

A **sequence diagram** in a Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams typically are associated with use case realizations in the Logical View of the system under development.

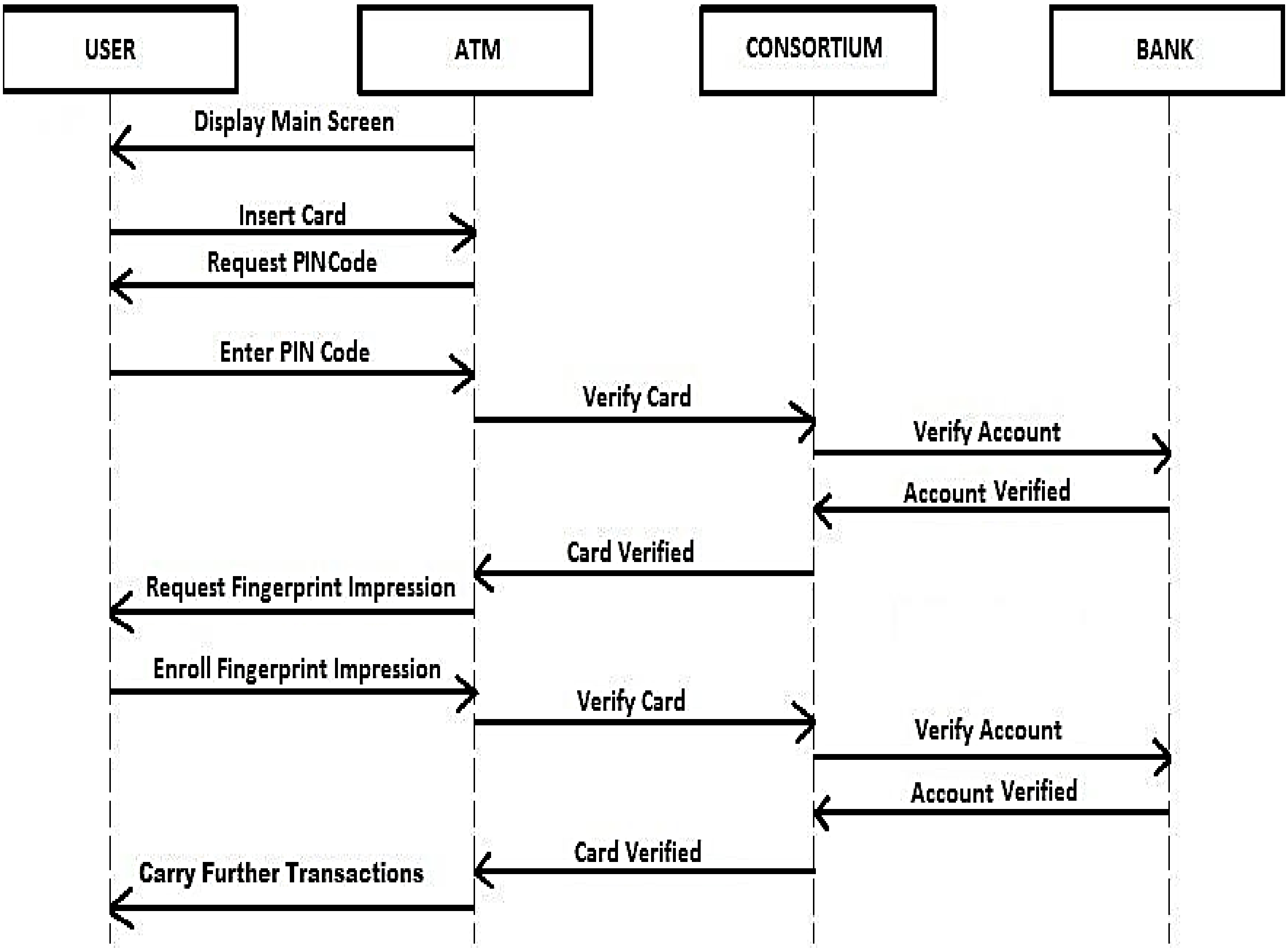


Figure 3.7 Sequence diagram

3.5 ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

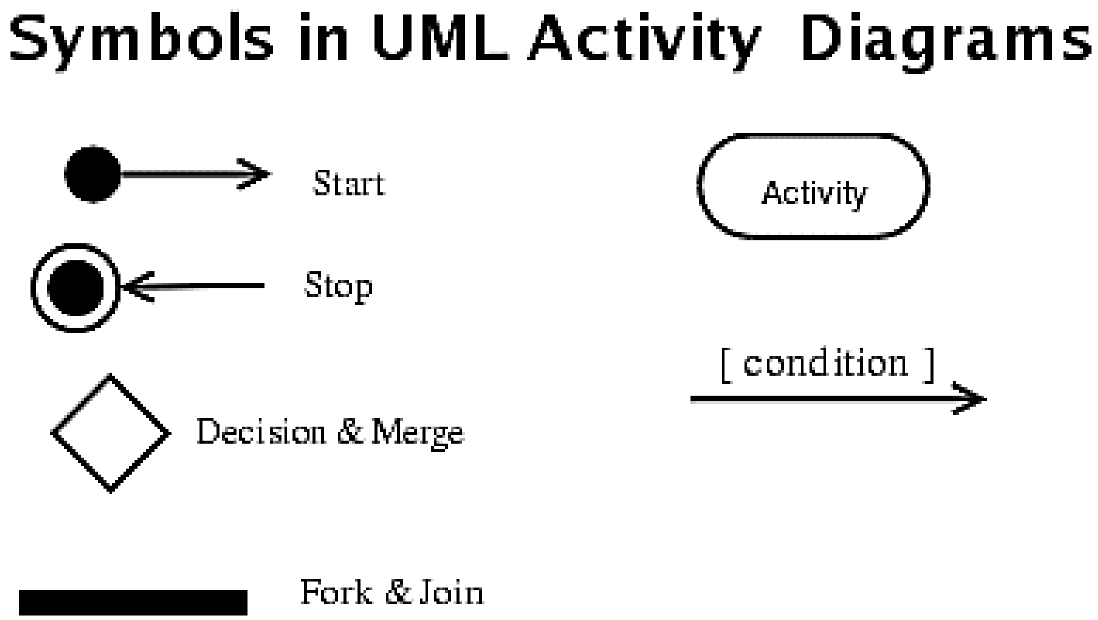


Figure 3.8 Activity diagram symbols

Activity diagrams are constructed from a limited number of shapes, connected with arrows. The most important shape types:

- *rounded rectangles* represent *activities*.
- *diamonds* represent *decisions*
- *bars* represent the start (*split*) or end (*join*) of concurrent activities.
- a *black circle* represents the start (*initial state*) of the workflow.
- an *encircled black circle* represents the end (*final state*).

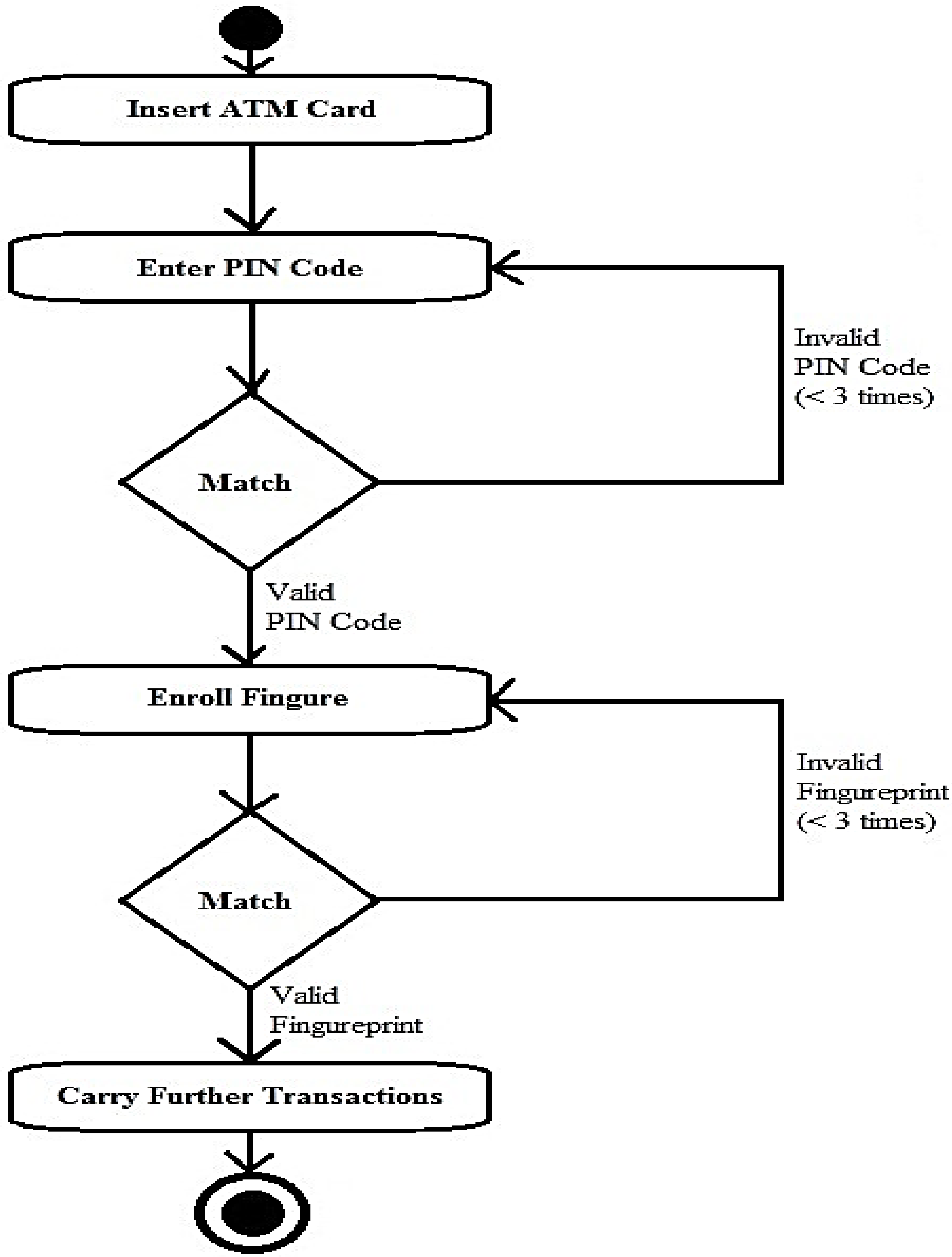


Figure 3.9 Activity Diagram

3.6 DATA FLOW DIAGRAM

A **data flow diagram** (**DFD**) is a graphical representation of the "flow" of data through an information system, modelling its *process* aspects. Often, they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).

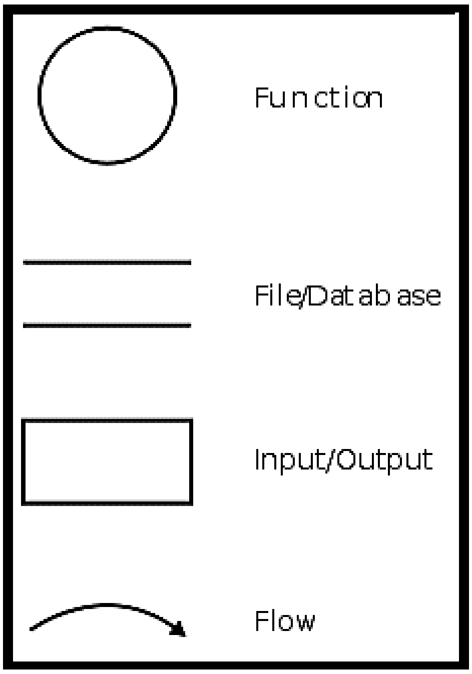


Figure 3.10 Data Flow Diagram symbol

3.6.1 DATA FLOW DIAGRAM LEVEL 0

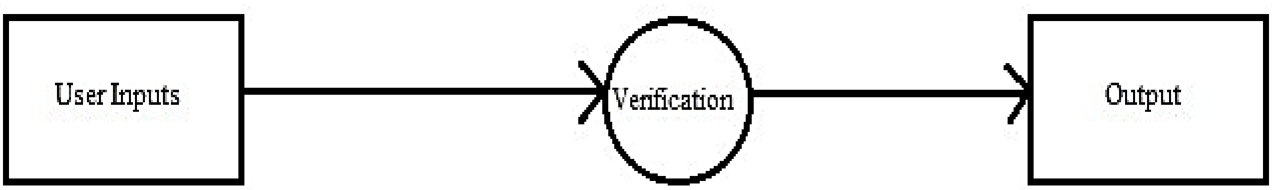


Figure 3.11 DFD Level 0

3.6.2 DATA FLOW DIAGRAM LEVEL 1

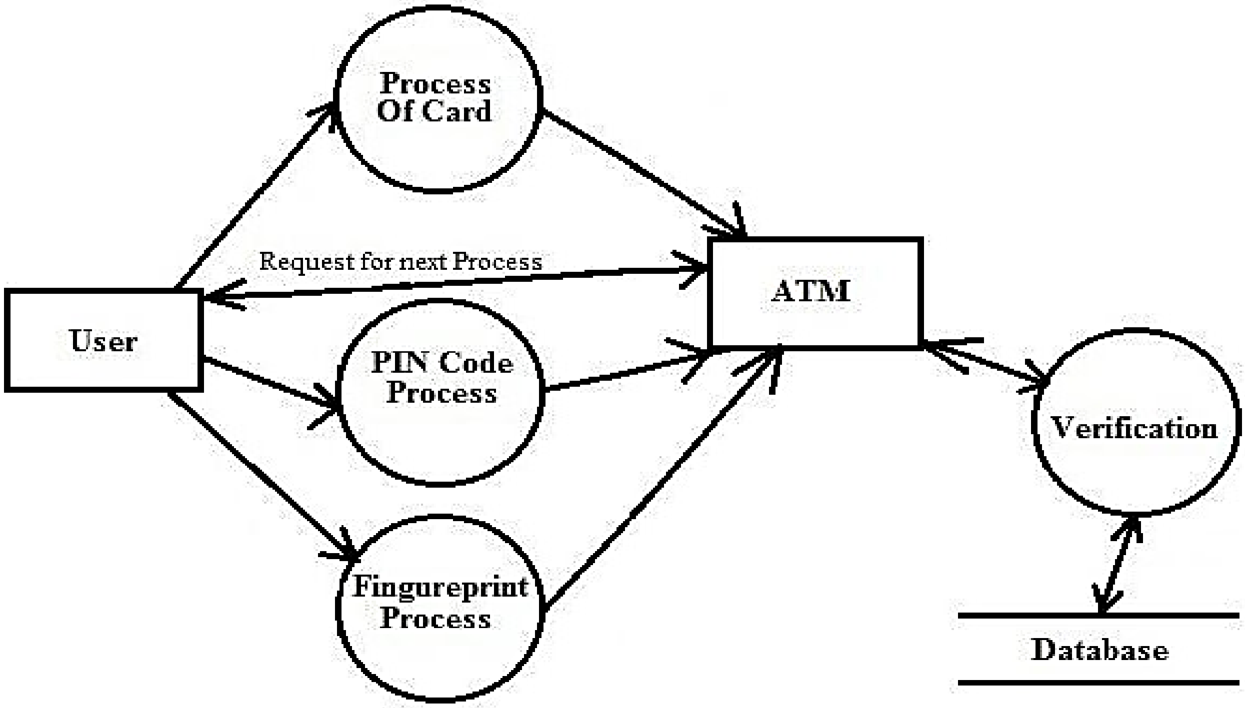


Figure 3.12 DFD Level 1

3.6.3 DATA FLOW DIAGRAM LEVEL 2

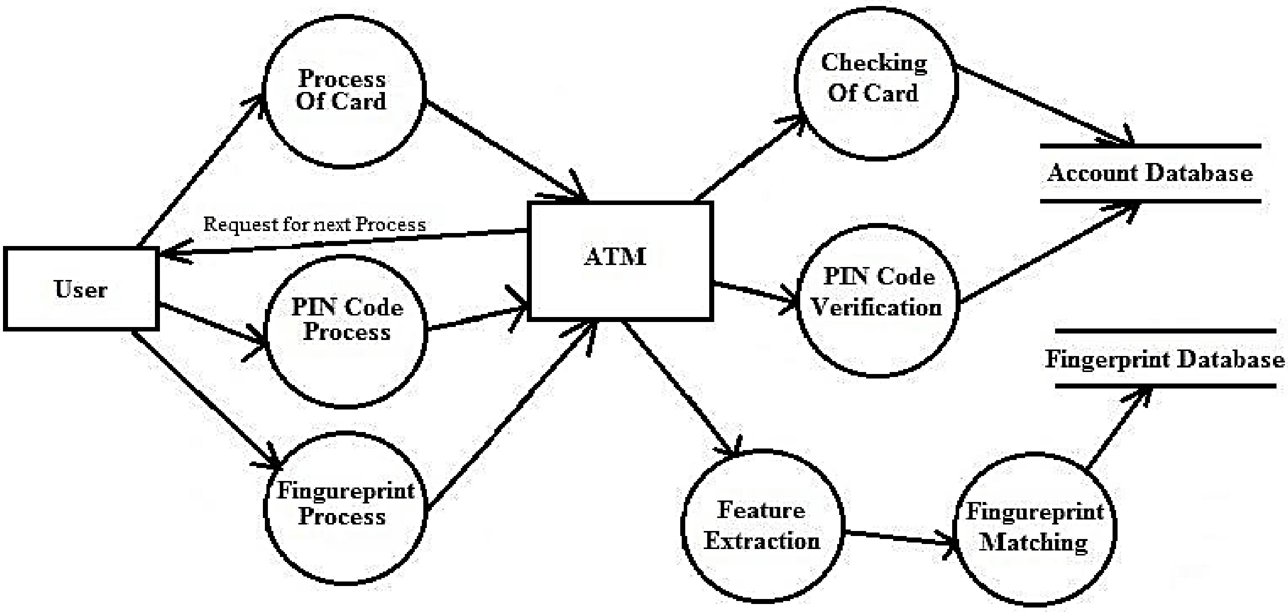


Figure 3.13 DFD Level 2

SYSTEM DESIGN

4.1 OVERVIEW OF SYSTEM DESIGN

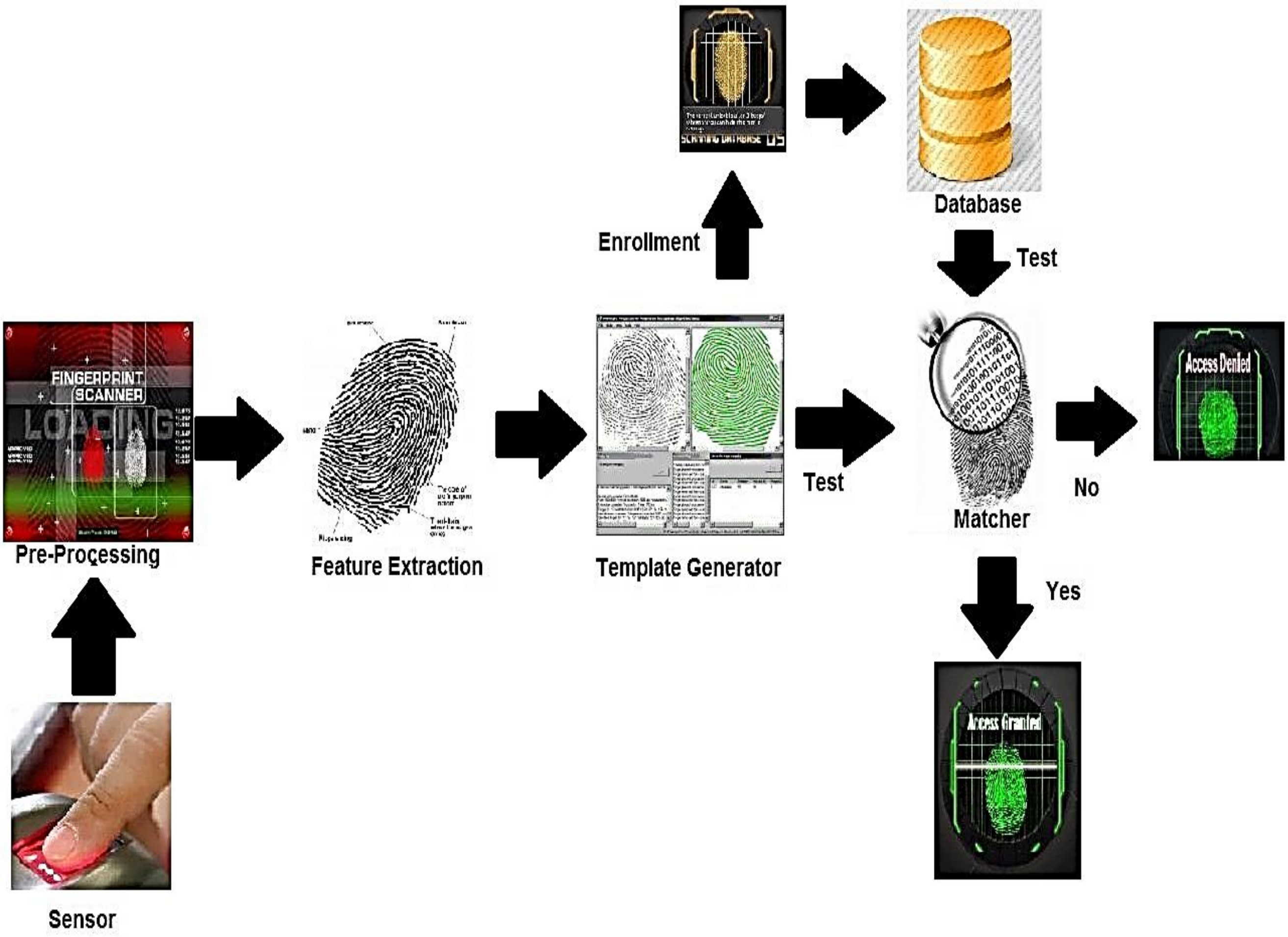


Figure 4.1 System design overview

4.2 SYSTEM LEVEL DESIGN

A fingerprint authentication system constitutes of fingerprint acquiring device, minutia extractor and minutia matcher [Figure 4.2].

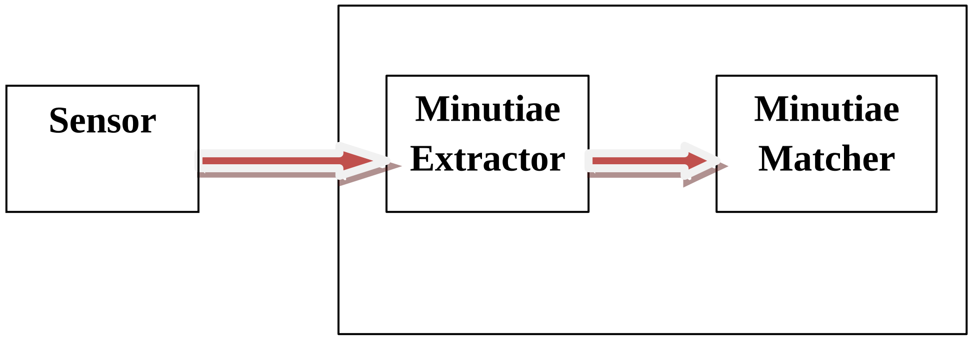


Figure 4.2 Simplified Fingerprint Recognition System

For fingerprint acquisition, optical or semi-conduct sensors are widely used. They have high efficiency and acceptable accuracy except for some cases that the user's finger is too dirty or dry. No acquisition stage is implemented.

The minutia extractor and minutia matcher modules are explained in detail in the next part for algorithm design and other subsequent sections.

4.3 ALGORITHM LEVEL DESIGN

To implement a minutia extractor, a three-stage approach is widely used by researchers. They are pre-processing, minutia extraction and post processing stage [Figure 4.3].

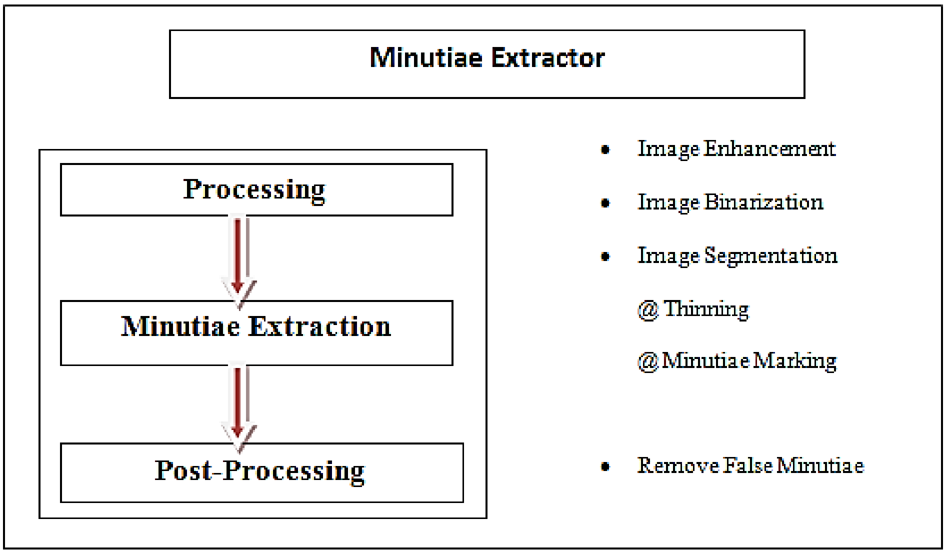


Figure 4.3 Minutia Extractor

For the fingerprint image pre-processing stage, I use Histogram Equalization and Fourier Transform to do image enhancement. And then the fingerprint image is binarized using the locally adaptive threshold method. The image segmentation task is fulfilled by a three-step approach: block direction estimation, segmentation by direction intensity and Region of Interest extraction by Morphological operations. Most methods used in the pre-processing stage are developed by other researchers, but they form a brand-new combination in our project through trial and error.

For minutia extraction stage, three thinning algorithms are tested, and the Morphological thinning operation is finally bid out with high efficiency and pretty good thinning quality. The minutia marking is a simple task as most literatures reported but one special case is found during our implementation and an additional check mechanism is enforced to avoid such kind of oversight.

For the post-processing stage, a more rigorous algorithm is developed to remove false minutia. Also, a novel representation for bifurcations is proposed to unify terminations and bifurcations.

Minutiae Matcher

- **Ridge correlation to specify reference minutiae pair**
- **Align two fingerprint images**
- **Minutiae Match**

Figure 4.4 Minutia Matcher

The minutia matcher chooses any two minutiae as a reference minutiae pair and then matches their associated ridges first. If the ridges match well, two fingerprint images are aligned, and matching is conducted for all remaining minutiae [Figure 4.4].

FINGERPRINT IMAGE PROCESSING

5.1 STEPS INVOLVED IN FINGERPRINT IMAGE PROCESSING

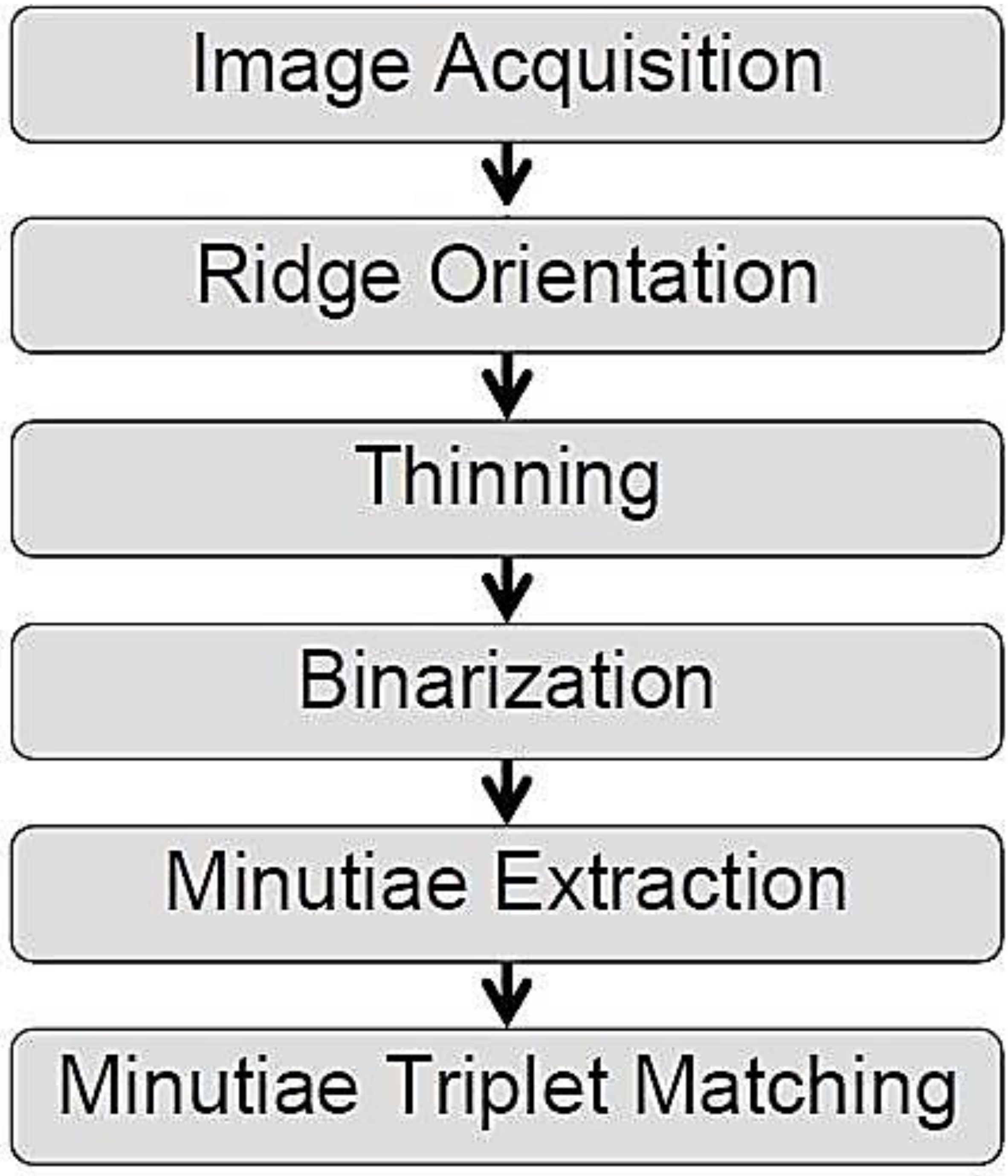


Figure 5.1 Steps of fingerprint recognition

5.2 IMAGE ACQUISITION

Image acquisition is the creation of digital images, typically from a physical scene. The term is often assumed to imply or include the processing, compression, storage, printing, and display of such images. The most usual method is by digital photography with a digital camera, digital pictures with image scanners but other methods are also employed.

Here, I am using the digital image for the image processing which will be taken by the image scanners and image sensors.

5.3 RIDGE ORIENTATION

Ridge orientation is the process of obtaining the angle of the ridges throughout the image. Ridge orientations are calculated on a block-basis for a WxW block, where W is generally equal to 16 i.e. 16x16 block.

5.4 THINNING

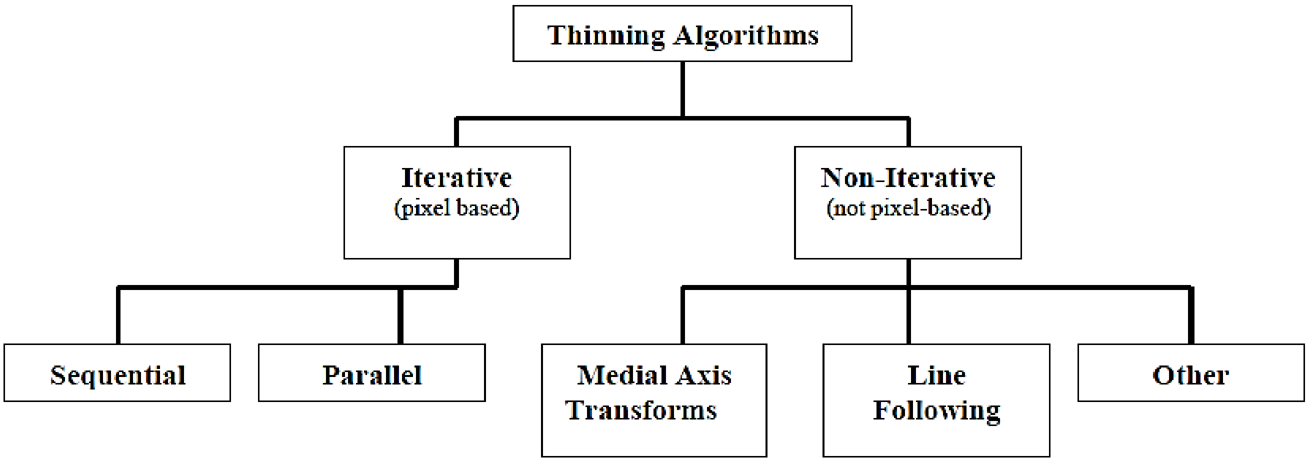


Figure 5.2 Classification of the Thinning algorithms

The table above shows a classification of thinning algorithms. The second class of sequential thinning algorithms is parallel.

In parallel thinning algorithms the decision for individual pixel deletion is based the results of the previous iteration. Like sequential algorithms, parallel thinning usually considers a 3*3neighborhood around the current pixel. A set of rules for deletion is applied based on pixels in the neighborhood. Fully parallel algorithms have trouble maintaining connectedness, so they are often broken into sub-iterations where only a subset of the pixels is considered for deletion.

Non-iterative thinning methods are not based on examining individual pixels. Some popular non-pixel-based methods include medial axis transforms, distance transforms, and determination of centerlines by line following. In line following methods, midpoints of black spaces in the image are determined and then joined to form a skeleton. This is fast to compute but tends to produce noisy skeletons. It has been conjectured that human beings naturally perform thinning in a manner like this.

Another method of centerline determination is by following contours of objects. By simultaneously following contours on either side of the object a continual centerline can be computed. The skeleton of the image is formed from these connected centerlines.
Medial axis transforms often use gray-level images where pixel intensity represents distance to the boundary of the object. The pixel intensities are calculated using distance transforms. In Figure below the maximum pixel intensity would increase toward the dark lines at the centres of the circles. Note that there are other methods of computing medial axis transforms.
The following is the result of the Thinning algorithm when applied to a binary image:

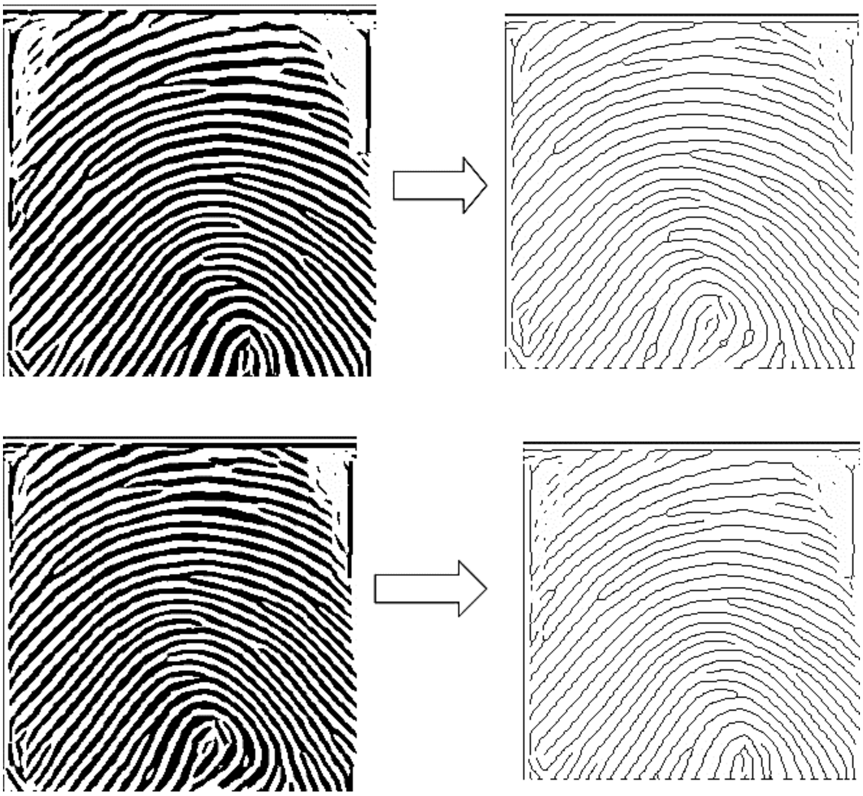


Figure 5.3 showing the result of thinning algorithm

5.5 BINARIZATION

Fingerprint binarization is to transform the 8-bit Gray fingerprint image to a 1-bit image with 0-value for ridges and 1-value for furrows. After the operation, ridges in the fingerprint are highlighted with black color while furrows are white.

A locally adaptive binarization method is performed to binarize the fingerprint image. Such a named method comes from the mechanism of transforming a pixel value to 1 if the value is larger than the mean intensity value of the current block (16x16) to which the pixel belongs (Figure 5.4).

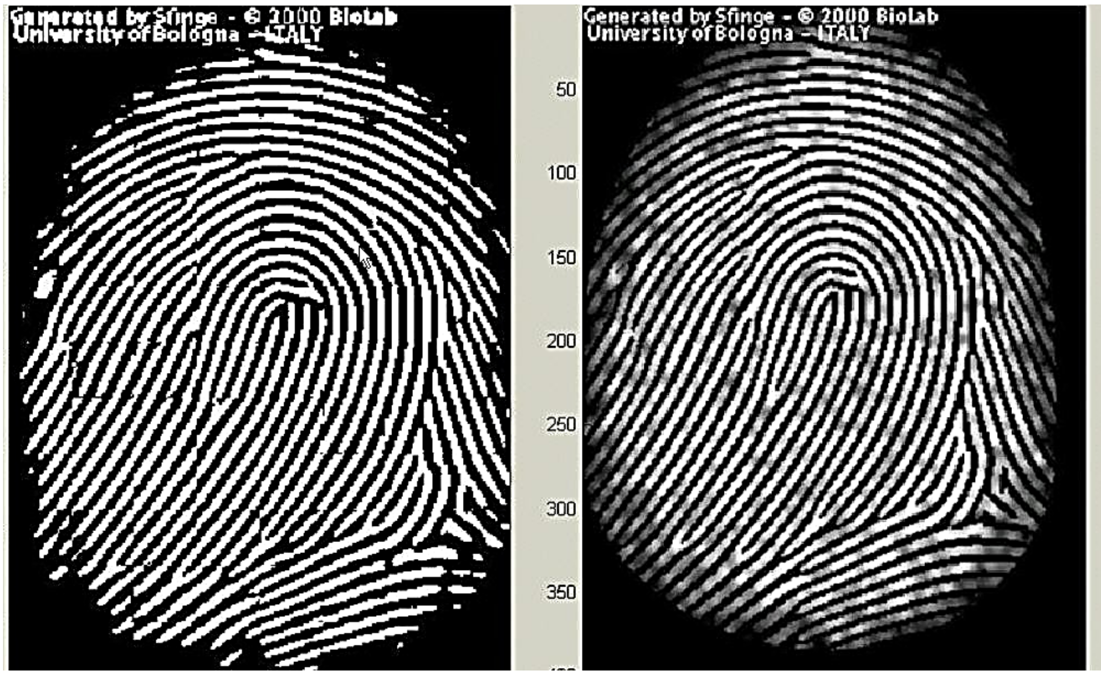


Figure 5.4 the fingerprint image after adaptive binarization binarized image (left), enhanced gray image (right)

5.6 MINUTIAE EXTRACTION

Our implementation of fingerprint identification and verification is based the topological structural matching of minutiae points. I only consider two kinds of minutiae: ridge endings and bifurcations as shown in the following figure:

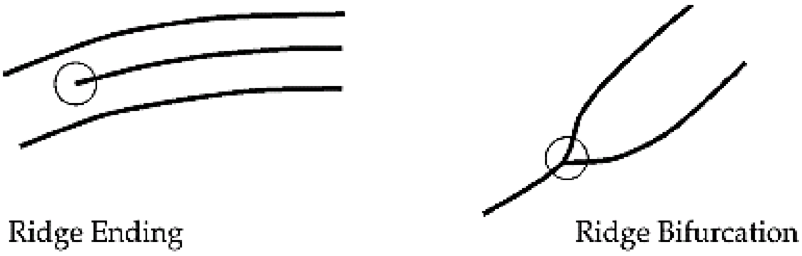


Figure 5.5 Ridge ending (left) & ridge bifurcation (right)

Minutiae extraction from a perfectly thinned ridge-map of a fingerprint image is a trivial task. All I need to do is to count the number of ridge pixels, every ridge pixel on the thinned image is surrounded by it.

However, due to noise, limitation on image acquisition, skin deformations etc. The fingerprint image obtained is never ideal. As a result, there are a lot of spurious minutiae that crop up if I simply follow the above approach to minutiae detection. To solve the problem, various heuristics have been proposed and I have implemented the following rules to remove most of the spurious minutiae, resulting from noise in the thinned image:

- If several minutiae form a cluster in a small region, then remove all of them except for the one nearest to the cluster center.
- If two minutiae are located close enough, facing each other, but no ridges lie between them, then remove both in addition to the noise in the fingerprint image, the thinned image may not be ideal. If such is the case, minutiae extraction may not yield the correct results.

In addition to the noise in the fingerprint image, the thinned image may not be ideal. If such is the case, minutiae extraction may not yield the correct results.

5.7 MINUTIAE TRIPLET MATCHING

Minutiae triplet matching are the step which comes after minutiae extraction and it is here that we match the minutiae obtained from two sample fingerprint images and test whether they are from the same fingerprint or not.

However, a crucial step that needs to be carried out before we can use brute force and match minutiae on two images is alignment of the images. Alignment is necessary so that we correctly match the images. We also need to take care of difference in positioning of minutiae due to plastic deformations in the finger. The algorithms prevalent for minutiae-matching either include the use of details of ridges on which minutiae are present or use the Hough transform. Both these methods and most other methods are difficult to implement, and several complicated functions need to be implemented.

Hence, we decided to implement a minutiae matching algorithm which was inspired by the techniques involving computation of local and global minutiae features.

As the name suggest, it check and compare the three main component of the image during the image processing. It simultaneously checks for the ridges-ending, bifurcation, and ridges-pixels. So, the accurate matching could be achieved by eliminating the risk of unnecessary matching.

SOFTWARE & HARDWARE DESCRIPTION

6.1 MICROSOFT VISUAL STUDIO

6.1.1 INTRODUCTION

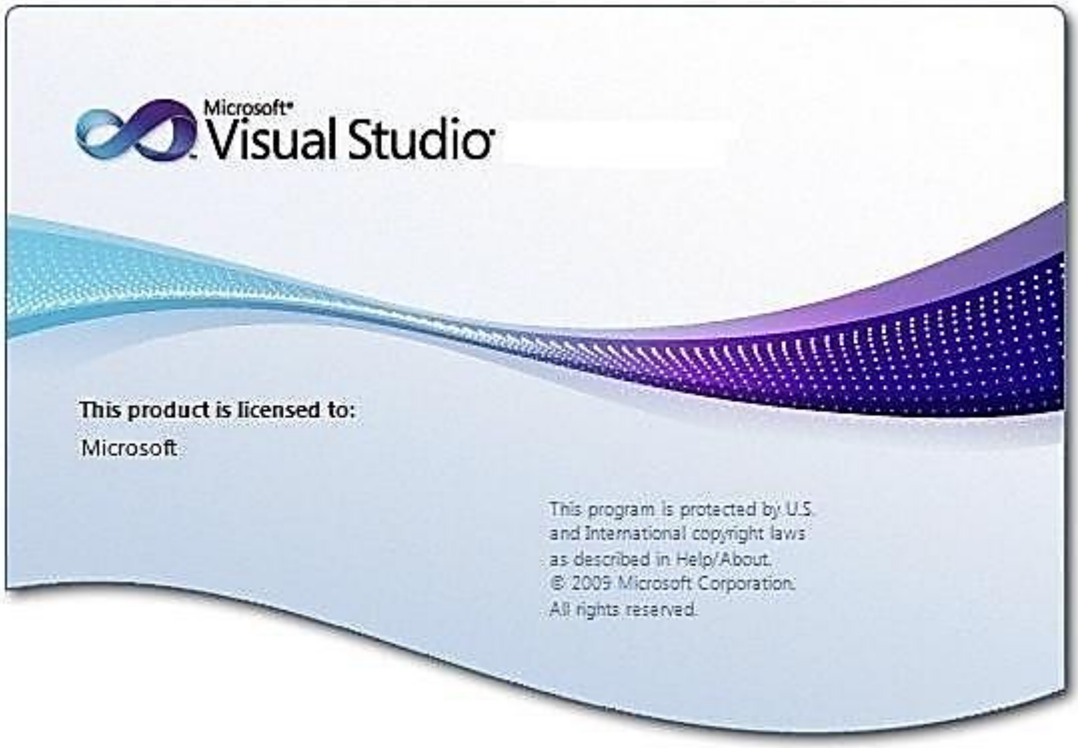
Microsoft Visual Studio is an Integrated Development Environment (IDE) from Microsoft. It is used to develop console and graphical user interface applications along with Windows Forms or WPF applications, web sites, web applications, and web services in both native codes together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that enhance the functionality at almost every level—including adding support for source-control systems (like Subversion and Visual SourceSafe) and adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports different programming languages by means of language services, which allow the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C/C++ (via Visua C++), VB.NET (via Visual Basic .NET), C# (via Visual C#), and F# (as of Visual Studio 2010). Support for other languages such as M, Python, and Ruby among others is available via language services installed separately. It also supports XML/XSLT, HTML/XHTML, JavaScript and CSS. Individual language-specific versions of Visual Studio also exist which provide more limited language services to the user: Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

Microsoft provides "Express" editions of its Visual Studio 2010 components Visual

Basic, Visual C#, Visual C++, and Visual Web Developer at no cost. Visual Studio 2012, 2010, 2008 and 2005 Professional Editions, along with language-specific versions (Visual Basic, C++, C#, J#) of Visual Studio Express 2010 are available for free to students as downloads via Microsoft's DreamSpark program.



Snapshot 6.1 Microsoft Visual Studio

6.1.2 FEATURES

The feature set is divided into eight areas:

1. Debugging and Diagnostics.
2. Testing Tools
3. Integrated Development Environment
4. Database Development
5. Development Platform Support
6. Architecture and Modelling
7. Lab Management
8. Team Foundation Server

6.1.3 WHY MICROSOFT VISUAL STUDIO?

Microsoft Visual Studio	Other Development Technology
Reduced clutter & complexity of UI	Much Complex UI
Support multiple document windows	Support only single document window
Supports multiple databases	Supports only single database
It includes IntelliTrace technology	It doesn’t include such technology
Supports more than 42 languages	Supports only single languages
Generate DLL file	Doesn’t generate DLL file
Easily to add plug-in into UI	Can’t allow plug-in to UI

Table 6.1 Microsoft advantage

6.2 MICROSOFT SQL SERVER

6.2.1 INTRODUCTION

Microsoft SQL Server is a Relational Database Management System developed by Microsoft. As a database, it is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet). There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time). Its primary query languages are T-SQL and ANSI SQL.

SQL Server Management Studio (SSMS) is a software application first launched with the Microsoft SQL Server 2005 that is used for configuring, managing, and administering all components within Microsoft SQL Server. The tool includes both script editors and graphical tools which work with objects and features of the server. A central feature of SQL Server Management Studio is the Object Explorer, which allows the user to browse, select, and act upon any of the objects within the server. It also has an "express" version that can be freely downloaded.

Starting from version 11, the application has been rewritten in WPF that is like Visual Studio 2010.

Figure 6.2 Snapshot SQL Server Studio

6.2.2 SERVICES

SQL Server also includes an assortment of add-on services. While these are not essential for the operation of the database system, they provide value added services on top of the core database management system. These services either run as a part of some SQL Server component or out-of-process as Windows Service and presents their own API to control and interact with them.

1. Service Broker
2. Replication Service
3. Analysis Service
4. Reporting Service
5. Notification Service
6. Integration Service
7. Full Text Search Service
8. SQLCMD
9. Visual Studio
10. SQL Server Management Studio

11. Business Intelligence Development Studio

6.2.3 Microsoft SQL Server 2008 R2 Features

1. Power Pivot for Excel and SharePoint
2. Master Data Services
3. StreamInsight
4. Reporting Services Add-in for SharePoint
5. Data-tier function in Visual Studio

6.2.4 WHY MICROSOFT SQL SERVER?

Microsoft SQL Server	Other Database
Less expensive	**Much expensive**
User friendly interface	**Complex interface**
Several servers may connect to single database	**Very few severs may connect to single database**
Vertical scalability	**Vertical & Horizontal scalability**
Best performance with Microsoft windows	**Less compatible with Microsoft Windows**

Table 6.2 Microsoft SQL Server advantage

6.3.4 DATABASE DESCRIPTION

The image scanned by the sensor is of following description:-

Database Type	Database Name	Details	Image Size	Size of Set
RDBMS	Microsoft SQL Server 2008 R2	User personal information	260 x 300 pixels (77.5 KB)	10 users x 1 fingerprint

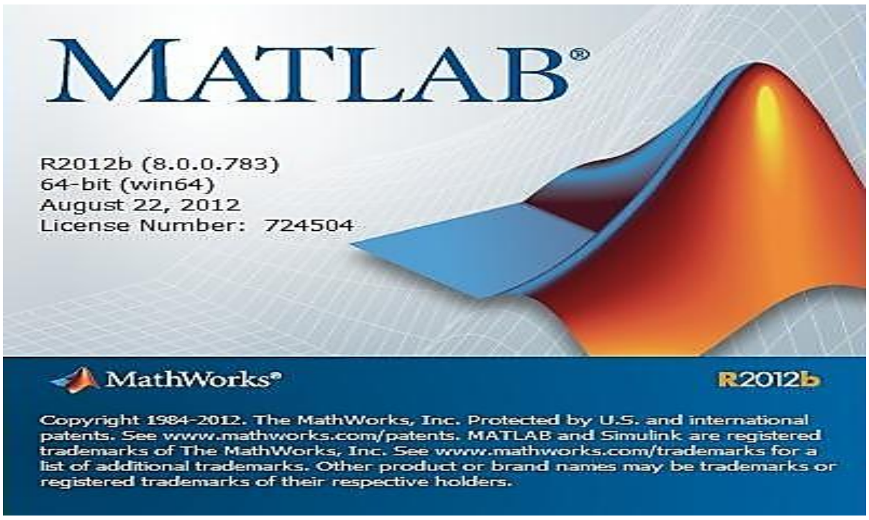
Table 6.3 Database description

6.3 MATLAB

6.3.1 INTRODUCTION

MATLAB (**mat**rix **lab**oratory) is a numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, and FORTRAN.

Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Image toolbox, adds extra feature of image processing with easily and compatible mode.



Snapshot 6.3 Matlab

6.3.2 FEATURES

1. High-level language for numerical computation, visualization, and application development.
2. Interactive environment for iterative exploration, design, and problem solving.
3. Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations
4. Built-in graphics for visualizing data and tools for creating custom plots.
5. Development tools for improving code quality and maintainability and maximizing performance.
6. Tools for building applications with custom graphical interfaces.

7. Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel®

6.3.3 WHY MATLAB?

Matlab	Other Image Processing Tool
Much user-friendly & flexible	**Much complex design**
Support all basic computer languages (such as C, Java, .Net, etc)	**Only support single computer language.**
Easy to connect with various databases	**Does not connect with database**
Doesn't require additional hardware	**Does requires additional hardware**
Easily connect to peripheral devices	**Doesn't connect to any peripheral device**

Table 6.4 Matlab advantage

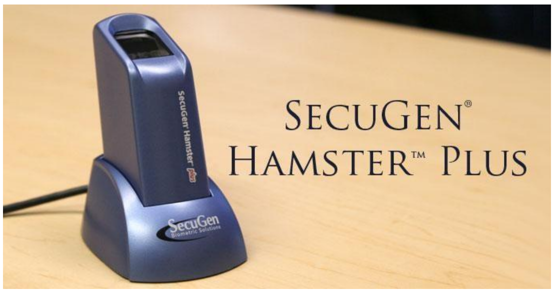
6.4 SECUGEN FINGERPRINT SCANNER

6.4.1 INTRODUCTION

SecuGen Hamster Plus is the improved version of SecuGen's popular and versatile fingerprint reader product line, with Auto-On and Smart Capture. Packaged in a comfortable, ergonomic design, the Hamster Plus features the industries most rugged and advanced optical sensor using patented SEIR fingerprint biometric technology.

Auto-On is an Automatic Finger Placement Detection technology that automatically checks for the presence of a finger. When used with Auto-On-compatible software, the Hamster Plus will turn on and scan your finger as soon as you touch the sensor - all without having to prompt the system.

Smart Capture ensures quality fingerprint scanning of difficult fingers. By automatically adjusting the brightness of the sensor, Smart Capture allows the Hamster Plus to capture high quality fingerprints from a wide range of traditionally difficult fingers, including those from dry, wet, scarred, or aged skin, and even in bright ambient conditions such as under direct sunlight.



Snapshot 6.4 SecuGen Fingerprint Scanner

6.4.2 FEATURES

1. High-performance, maintenance-free optical fingerprint sensor.
2. Sensor resistant to scratches, impact, vibration and electrostatic shock.
3. Auto-On™ (Automatic Finger Placement Detection)*.
4. Smart Capture™ (Self-Adjusting Fingerprint Brightness).
5. USB connection.
6. Removable weighted stand.
7. Compact, lightweight and portable.
8. Integrated finger guide.
9. Readily accessible for any finger.

6.3.3 WHY SECUGEN FINGERPRINT SCANNER?

SecuGen Fingerprint Scanner	Other Image Scanning Device
Easily connect to any device	Hard connection algorithms needed
Save images in bit format	Images need to convert in bit format.
Doesn't require external hardware for utilization	Requires external board & hardware for utilization
High quality of scanning sensation	Low quality of scanning sensation
Much effective over dirty fingers	Not effective over a dirty fingure

Table 6.5 SecuGen scanner advantage

6.3.4 IMAGE DESCRIPTION

The image scanned by the sensor is of following description:

Sensor Type	Image Size	Pattern Size	Resolution
Optical Fingerprint Sensor	260 x 300 pixels	16.1 mm x 18.2 mm	500 DPI (Dot Per Inch)

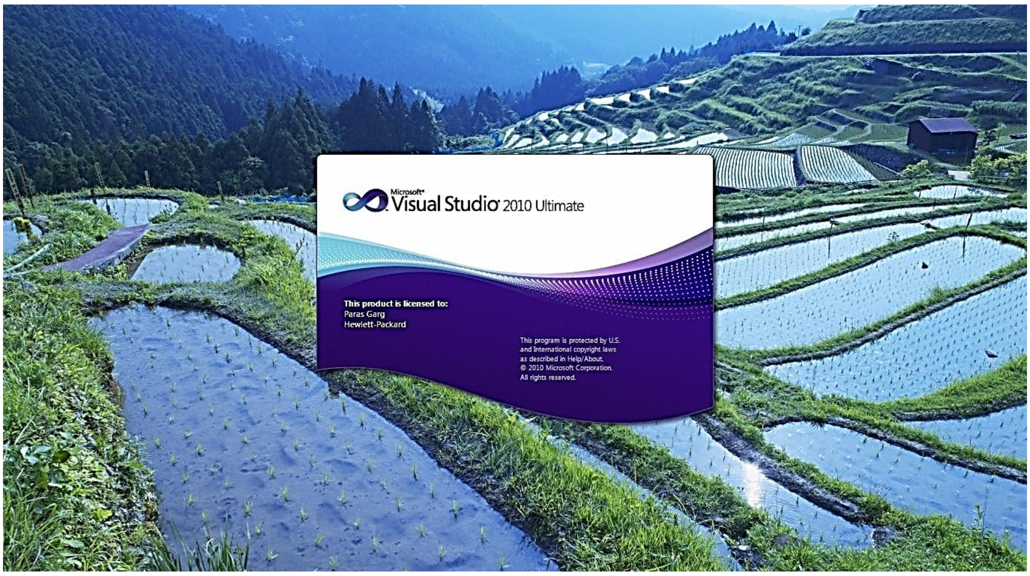
Table 6.6 Image description

SNAPSHOTS

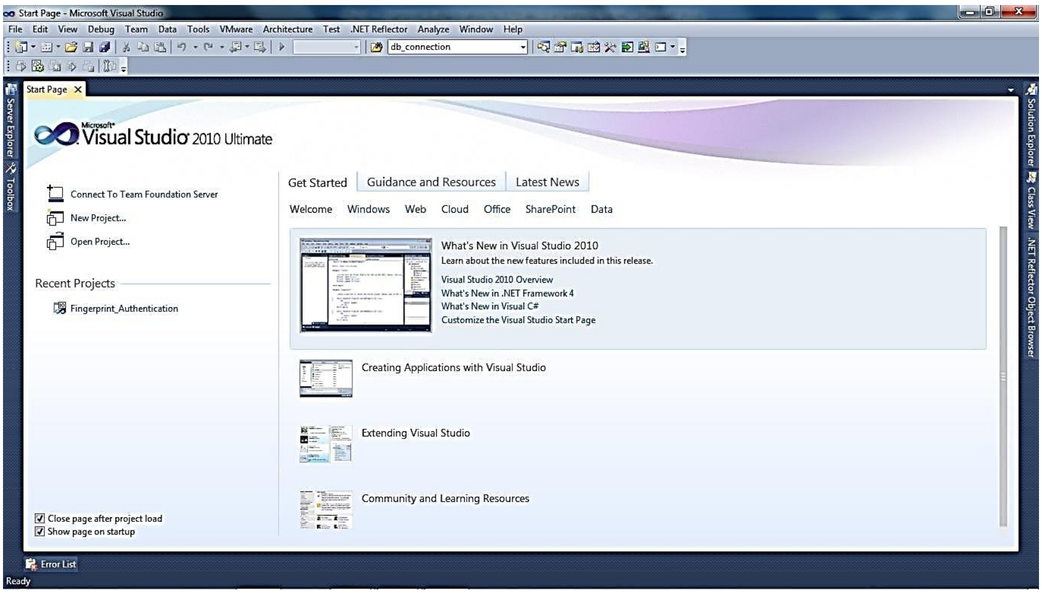
7.1 INTERFACE SOFTWARE

The front end or the interface of our project is created on the Microsoft Visual Studio 2010 Ultimate. The following snapshots are used to aware from the software environment, how to use it and how to modify the application interface and utility of project by the software functionality.

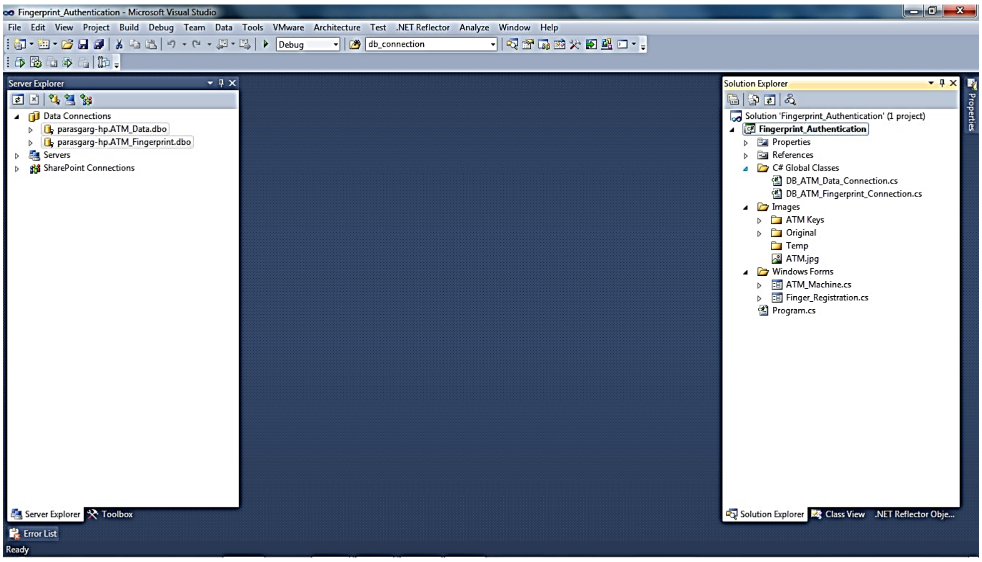
STEP 1 – Start Microsoft Visual Studio 2010 Ultimate.



STEP 2 – Choosing the project from the Recent Project List.



STEP 3 – Checking the database connectivity from Server Explorer in the left and checking the Solution Explorer in the right to modify the project interface.



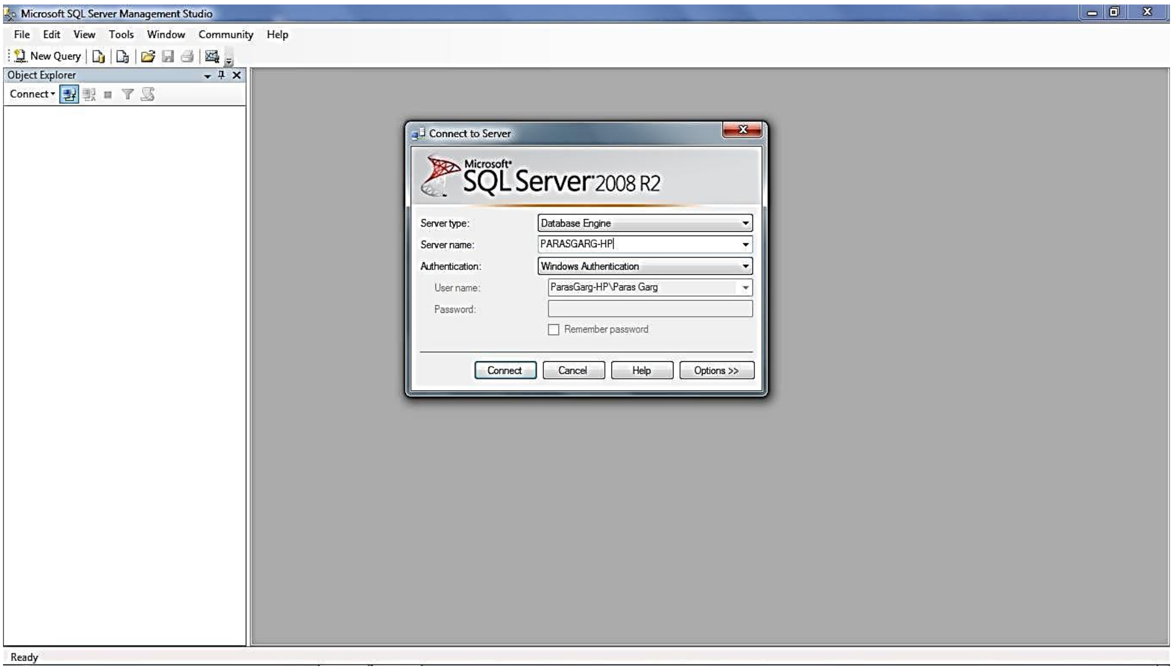
7.2 DATABASE SOFTWARE

The back end of project is created on the Microsoft SQL Server 2008 R2. The following snapshots are used to aware from the software environment, how to use it and how to modify the back end of project for the database connectivity and other database related queries by the software functionality.

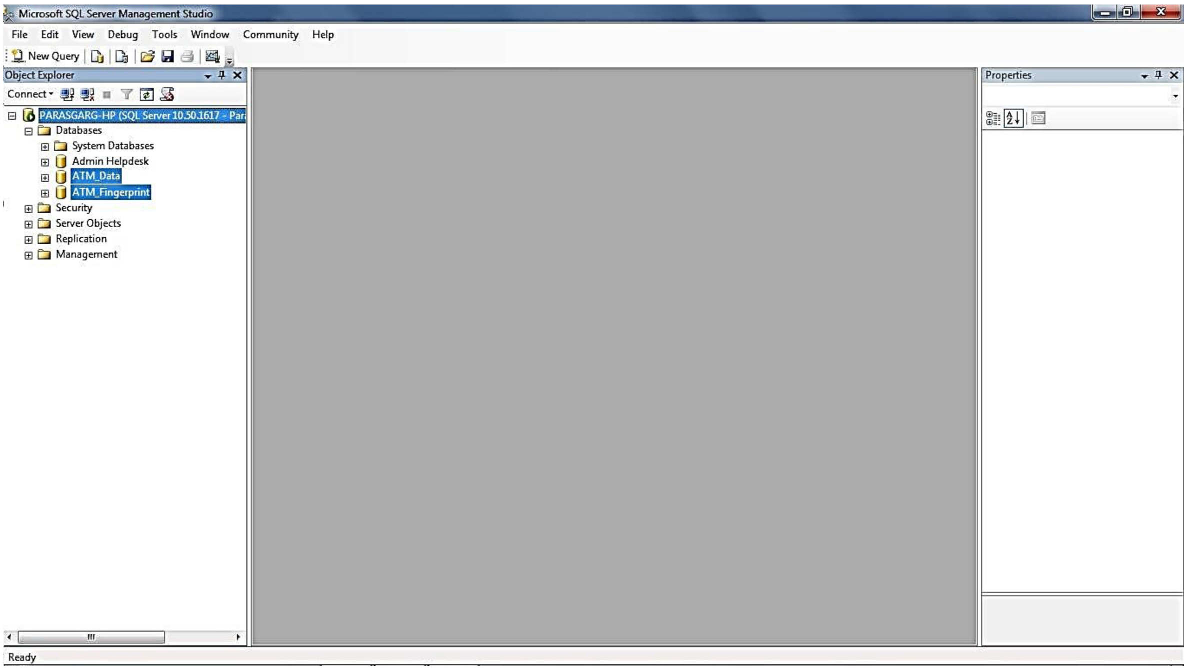
STEP 1 - Start Microsoft SQL Server 2008 R2

.

STEP 2 – Connecting the RDBMS to the Local/Server host depends upon the Authentication mode whether a Windows/SQL Server Authentication.



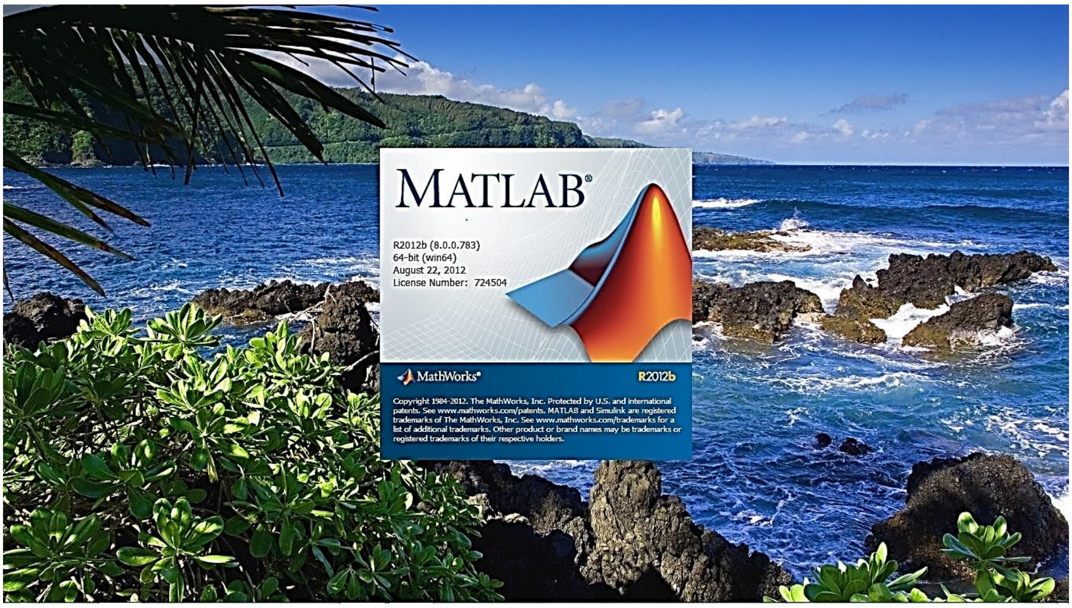
STEP 3 – Check and modify the database simply by scrolling and expanding the navigation panel in the left of the RDBMS.



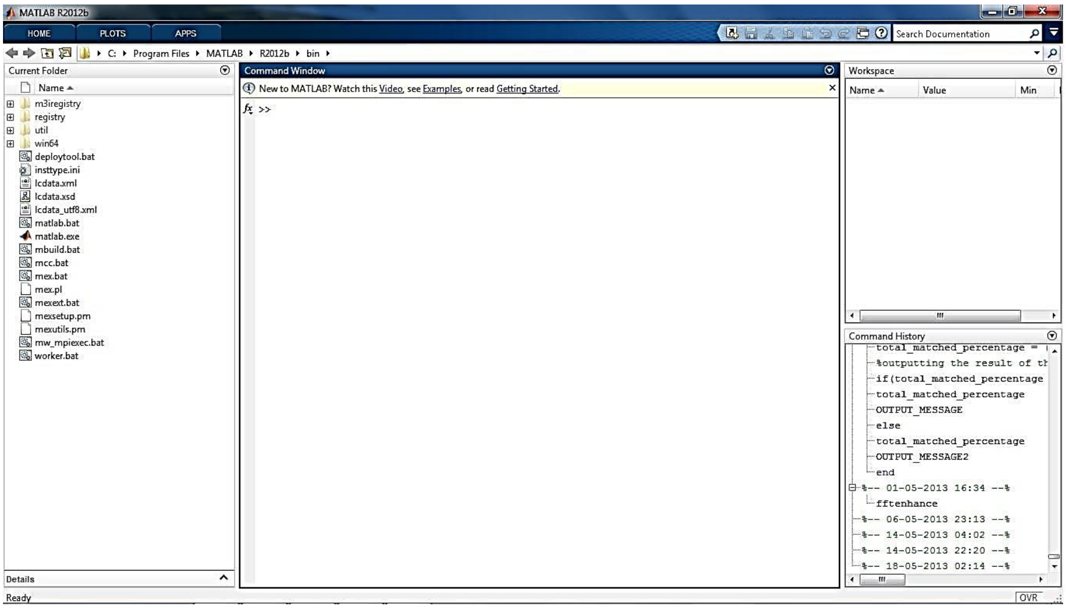
7.3 IMAGE PROCESSING SOFTWARE

The back end functionality of our project is created on the Mathworks Matlab R2012b. The following snapshots are used to aware from the software environment, how to use it and how to modify the back end functionality of project for the image processing and the database connectivity related to information sharing by the software functionality.

STEP 1 - Start Mathworks Matlab R2012b.



STEP 2 – The interface of the back end functionality is use for image processing and we have to write the code in the white black space known as command window.

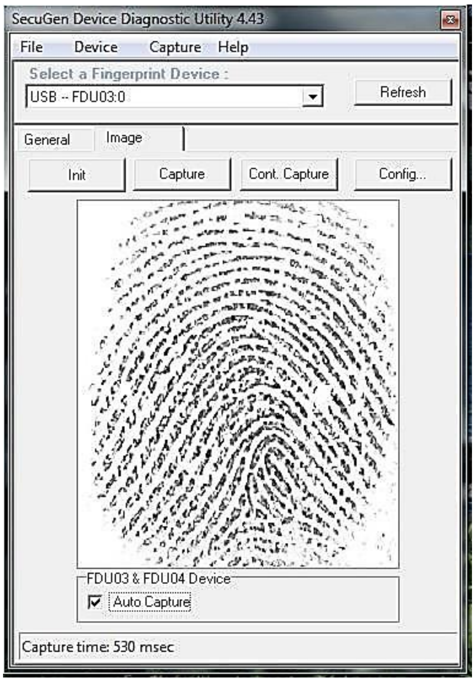


7.4 FINGERPRINT SENSOR SDK

The SDK software I am using is used to connect the sensor to the computer device using the Universal Serial Bus (USB) cable. Which is easy to connect and process and use for faster transfer of image scanned to the device.

STEP 1 – Start the SDK and we can have the general information about the sensor.

STEP 2 – We can view the fingerprint which is scanned by the sensor device which increase the Graphical User Interface (GUI) and become more users friendly.

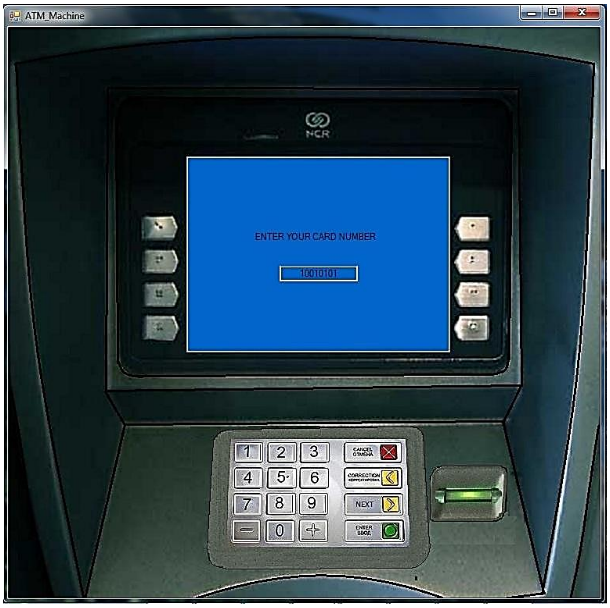


PROJECT MODEL

8.1 GRAPHICAL USER INTERFACE (GUI)

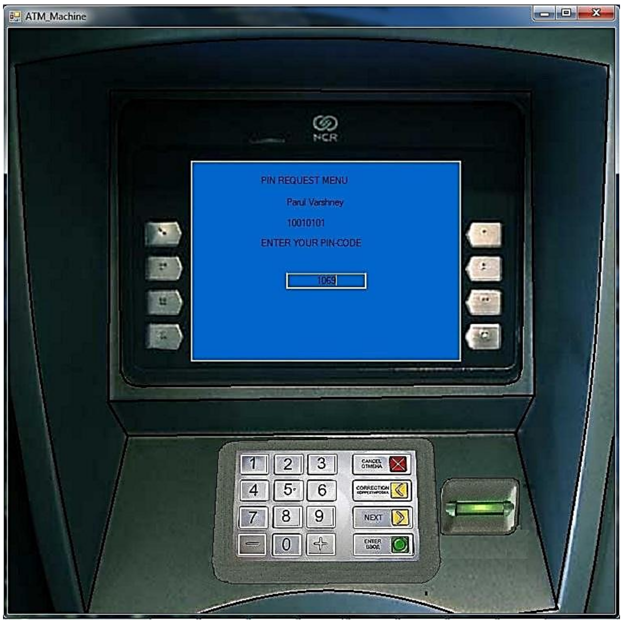
STEP 1

Snapshot is showing the welcome page



STEP 2

Snapshot is showing the card request page

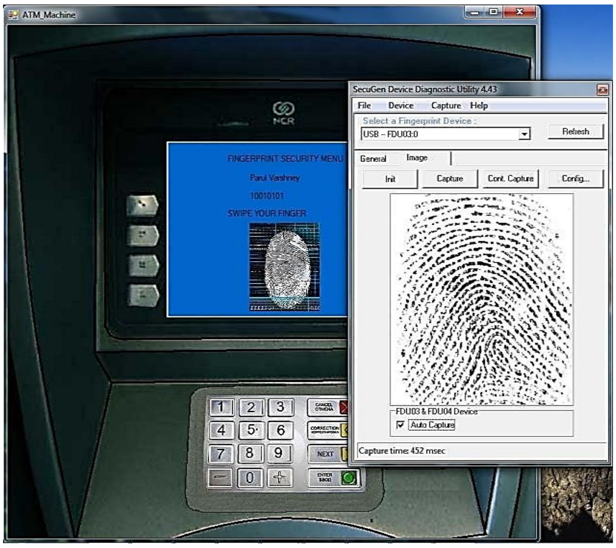


STEP 3

Snapshot is showing the pin request page

STEP 4

Snapshot is of request for enroll finger

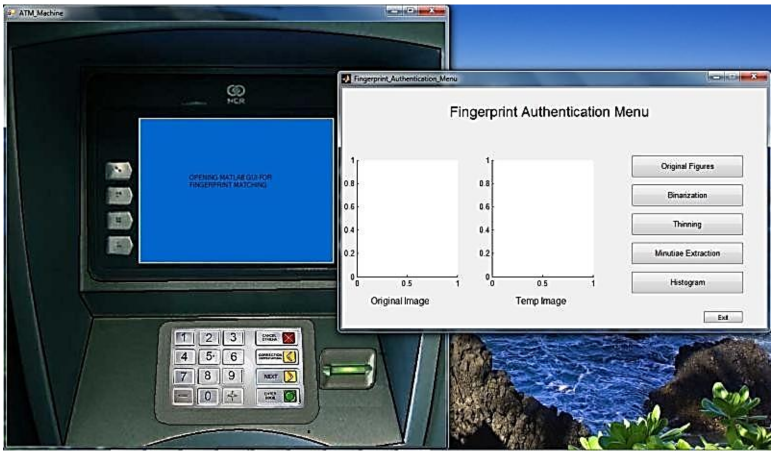


STEP 5

Snapshot is of request for fingerprint

STEP 6

Snapshot is of updating current entries in db



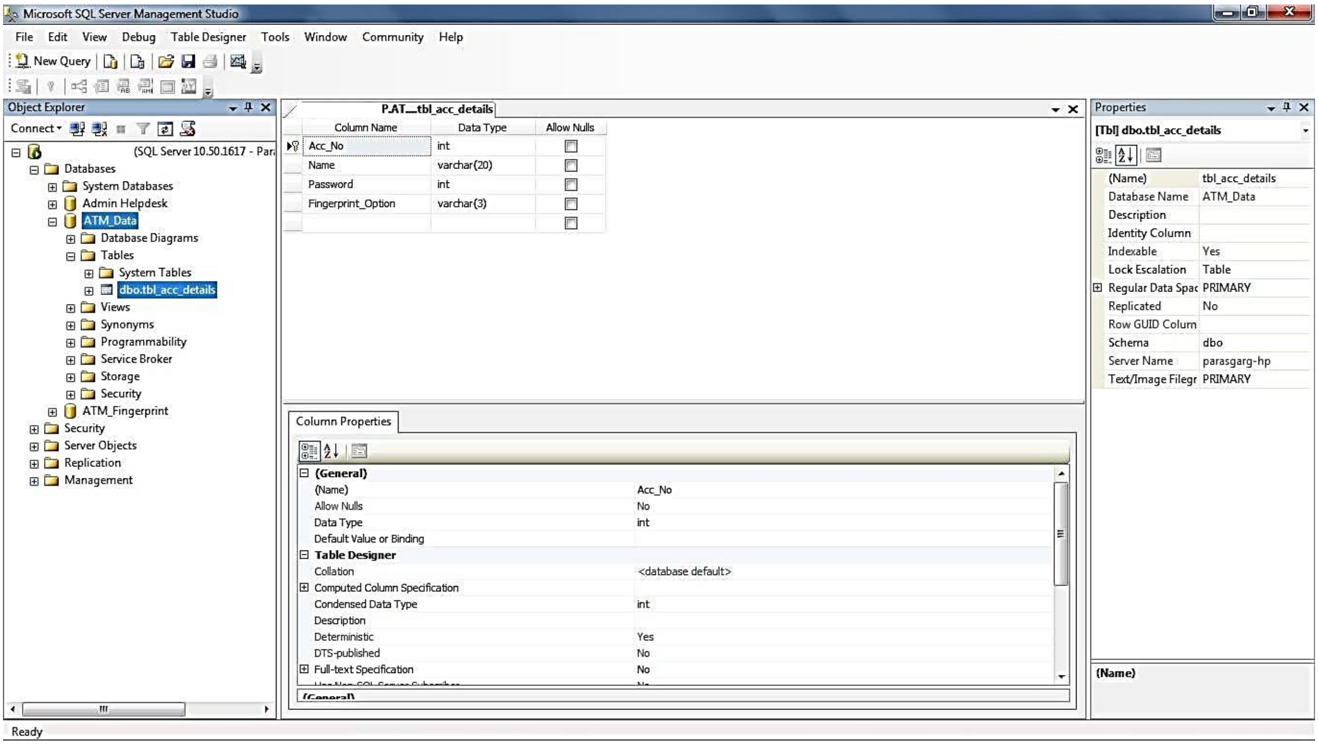
STEP 7

Snapshot is of request for MATLAB GUI

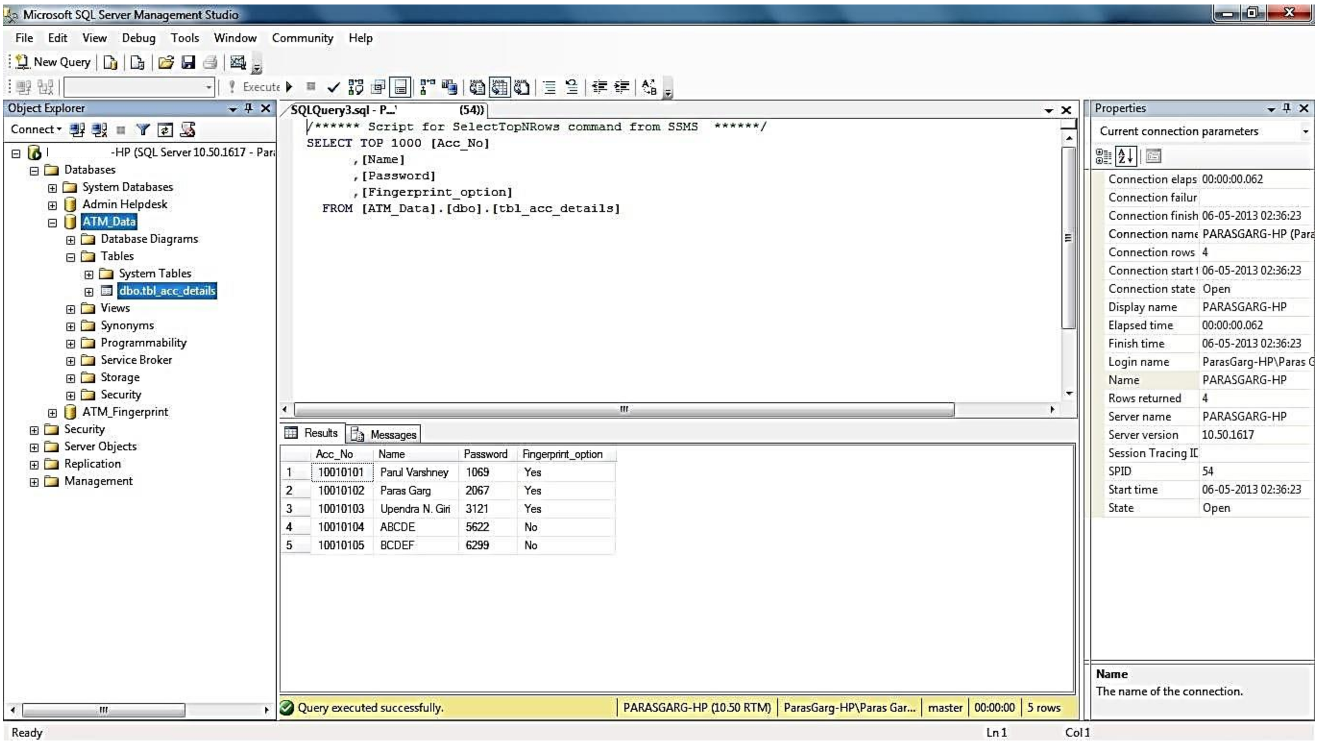
STEP 8

Snapshot is showing the end of process

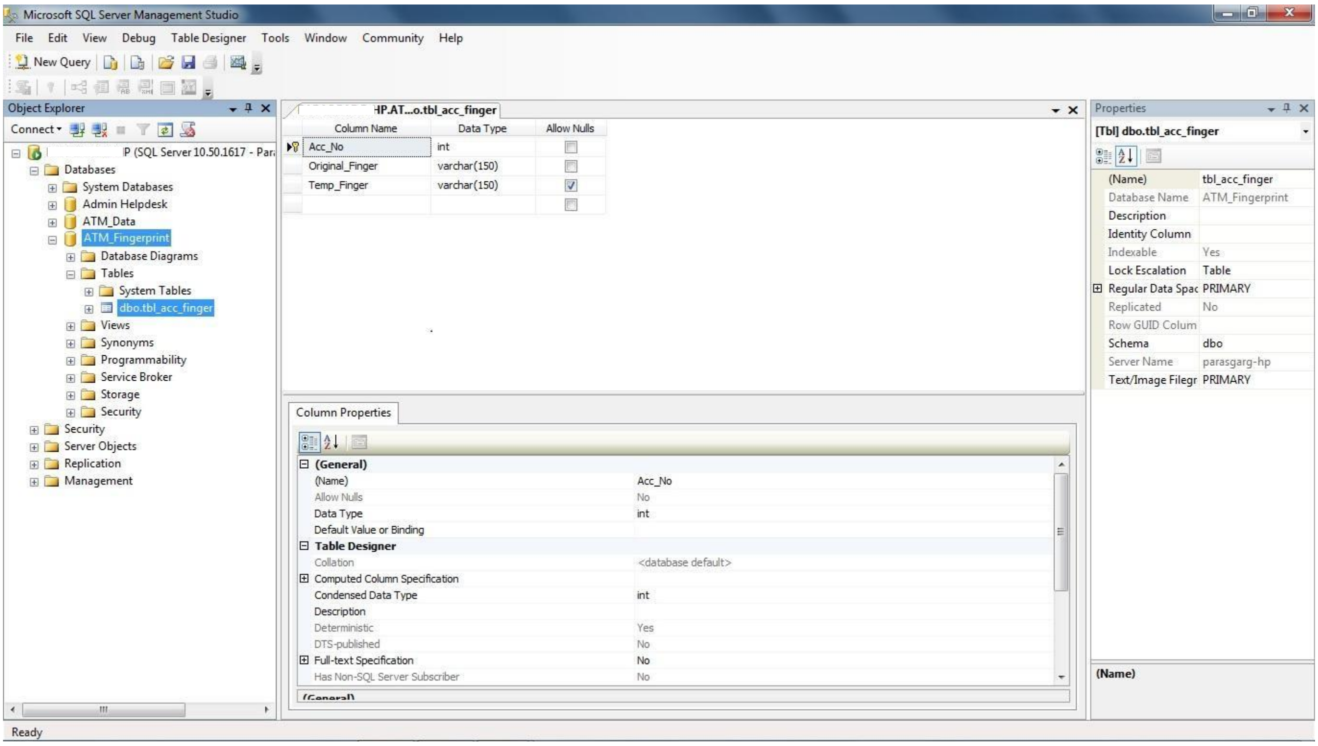
8.2 DATABASE MODEL



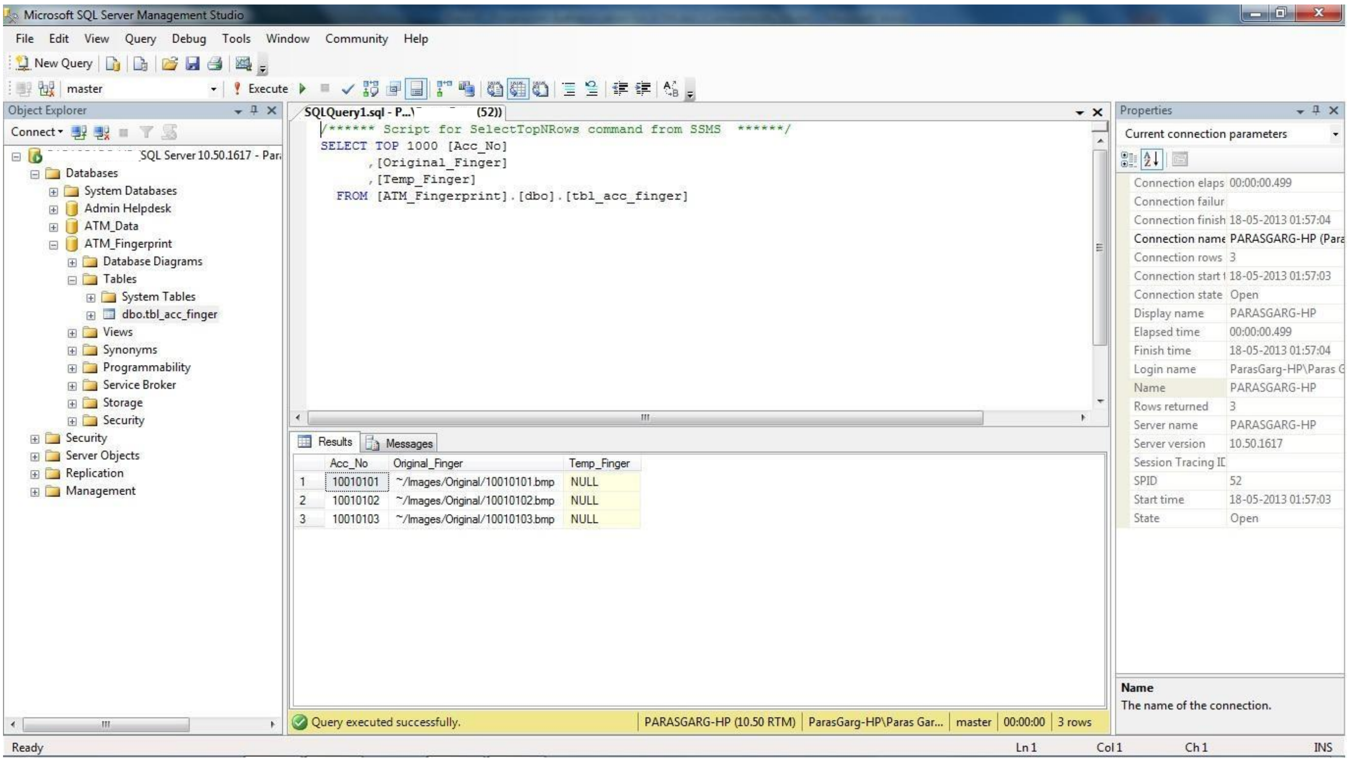
Snapshot is showing the structure of table of details



Snapshot is showing the view of table of details

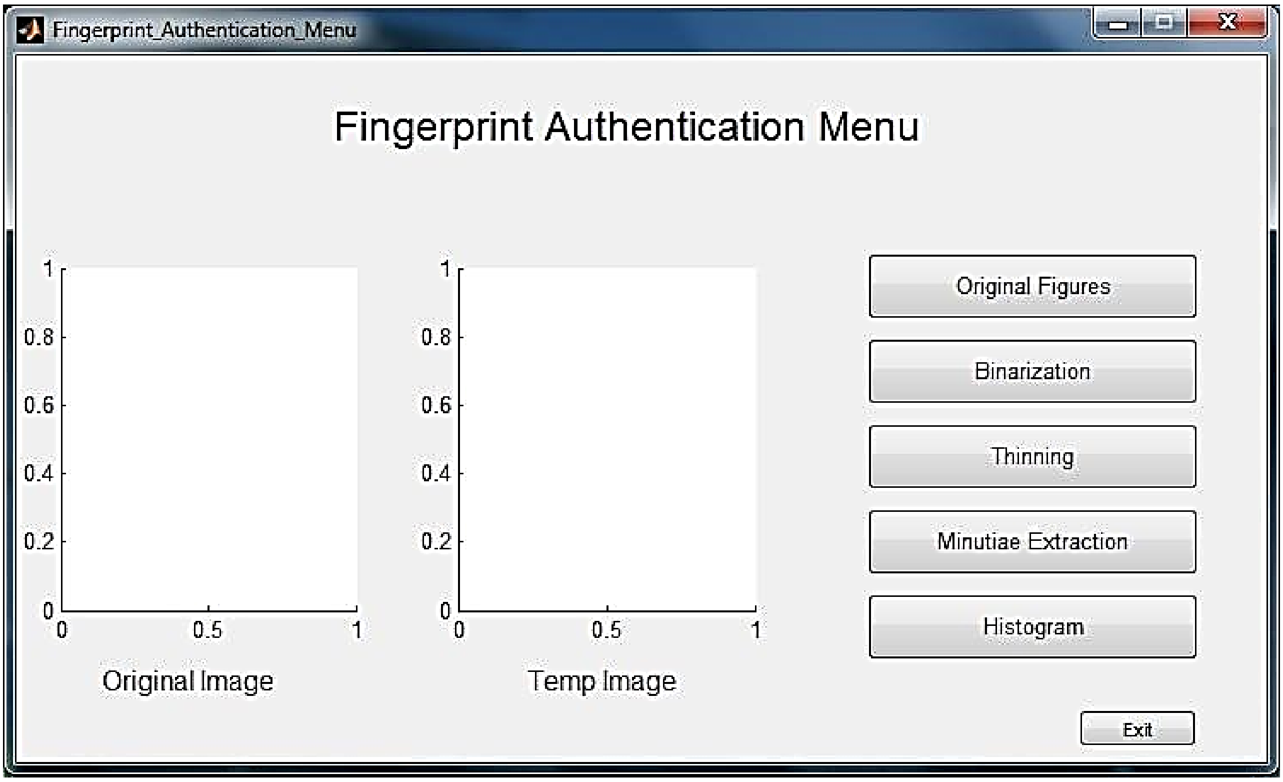


Snapshot is showing the structure of table of fingerprint



Snapshot is showing the view of table of fingerprint

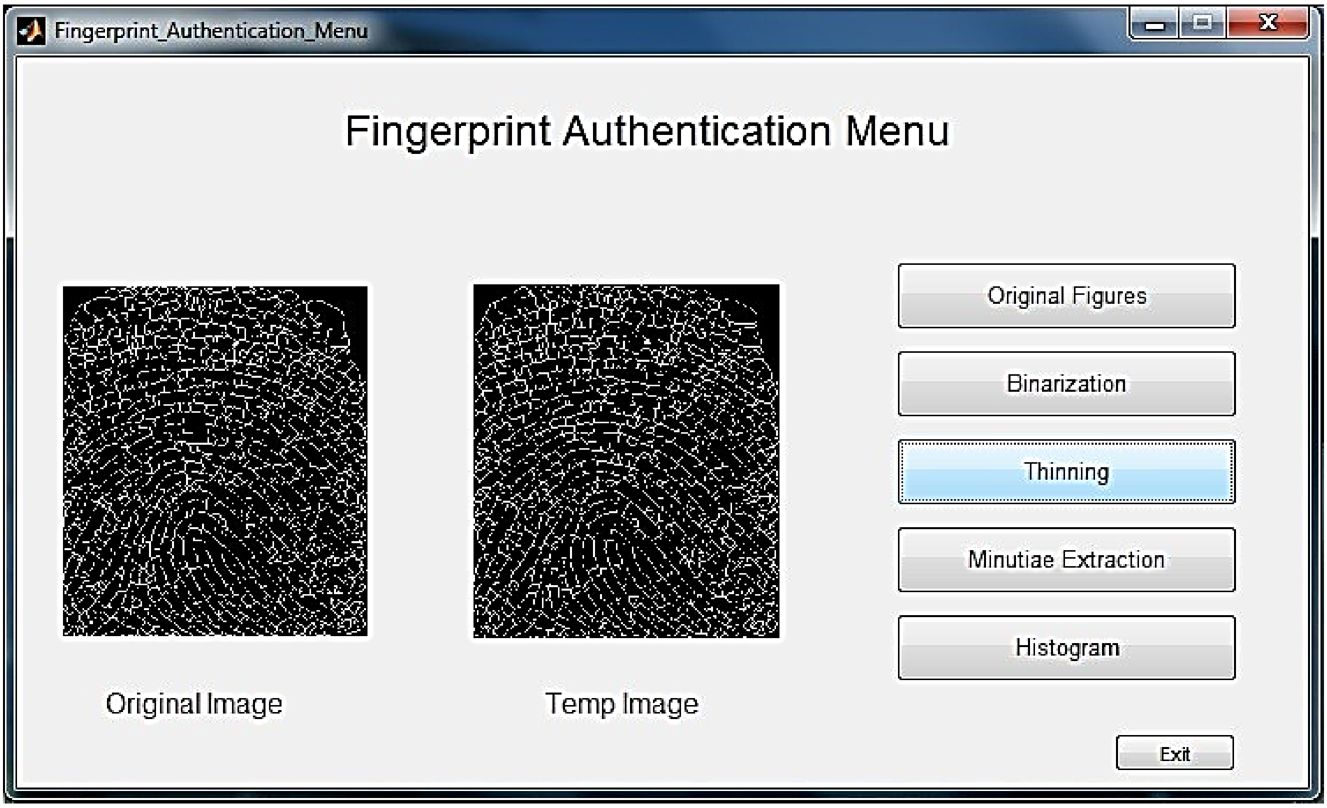
8.3 MATLAB GUI



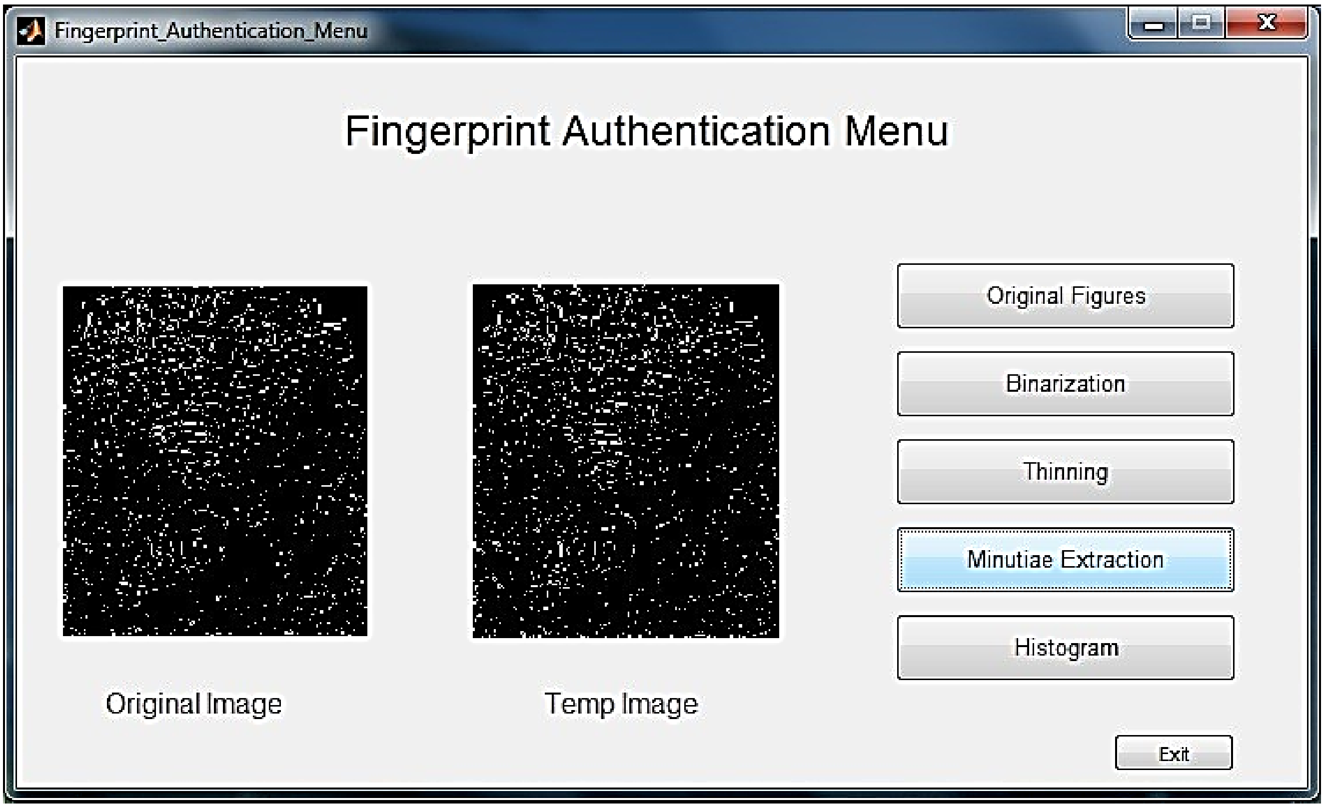
Snapshot is showing starting of MATLAB GUI

Snapshot is showing different figures

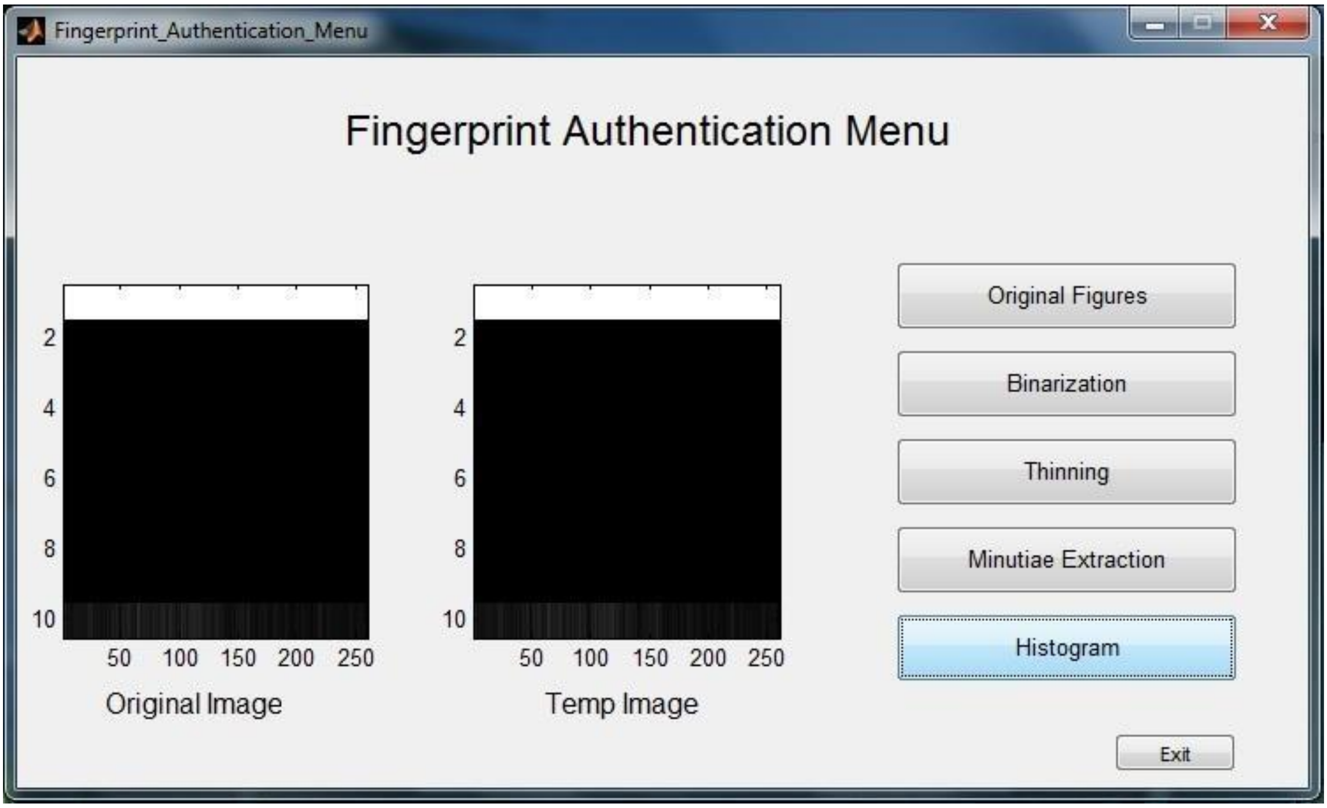
Snapshot is showing converted .bmp images to binary image



Snapshot is showing converted binary image to thinning



Snapshot is showing converted thinned image to minutiae of image



Snapshot is showing histograms of minutiae images

METHODOLOGY

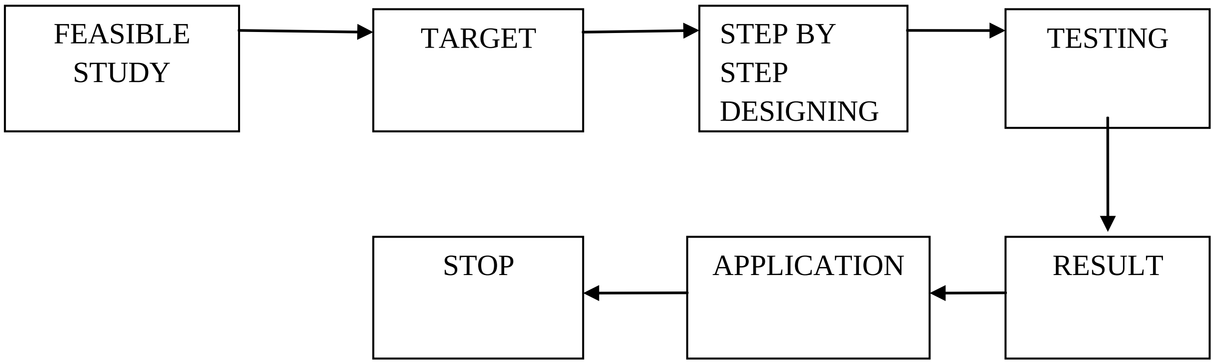


Figure 9.1: Block diagram of Methodology

9.1 FEASIBLE STUDY

The essence of this project is taken from various books and journals based on various integrity and security check systems. Various PDF were downloaded via Internet and references.

Subject related to the interface study has been done by understanding the .Net, Matlab and SQL server environment and studying the language use to code the logic from the reference guides and taken help from the project guide.

9.2 TARGET

Target of project is to provide an addition of sophisticated level of security and integrity to the user performing transactions through traditional ATM Systems.

9.3 STEP-BY-STEP DESIGNING

In this project following steps are taken for the designing of the system:

a. A biometric sensor performs scanning of the fingerprint of intended user.
b. Minutiae algorithm performs matching of that image with the images stored in database.

c. Generating MATLAB data with the help of MATLAB.
d. Creating database for the storage of data.
e. Developing interface as a connecting medium of the user & machine.
f. Fetching required information needed for proper security check from user side.

9.4 TESTING

The following steps are performed during testing the whole system:

a. Performing unit testing.
b. Creating test cases for integrity testing and whole system testing.

9.5 RESULT

The result is being measured by comparing the security components of the ATM system or by matching the PIN-Code and the fingerprint pattern matching which decide the end result of the system.

SUMMARY

10.1 ADVANTAGES

1. Very high accuracy and security
    - Identification (Do I know who you are?)
    - Verification (Are you who you claim to be?)
2. Is the most economical biometric PC user authentication technique.
3. It is one of the most developed biometrics.
4. Easy to use.
5. Small storage space required for the biometric template, reducing the size of the database memory required.
6. It is standardized.
7. Replace traditional methods (PINs, Passwords).

10.2 DISADVANTAGES

- **General Limitations**

1. Misidentification
    – False Acceptance
    – False Rejection
2. Privacy
3. Image captured at 500 dots per inch(dpi). Resolution: 8 bits per pixel. A 500-dpi fingerprint image at 8 bits per pixel demands a large memory space, 240 KB approximately → Compression required (a factor of 10 approximately).

- **Limitations for individual**

1. Dry, wet, or dirty hands.
2. For some people it is very intrusive because it is still related to criminal identification.

10.3 CONCLUSION

A smartcard based ATM fingerprint authentication scheme has been proposed. The possession (smartcard) together with the claimed user's Biometrics (fingerprint) is required in a transaction. The smartcard is used for the first layer of mutual authentication when a user requests transaction. Biometric authentication is the second layer. The fingerprint image is encrypted via 3D map as soon as it is captured, and then is transmitted to the central server via symmetric algorithm. The encryption keys are extracted from the random pixels distribution in a raw image of fingerprint.

The stable features of the fingerprint image need not to be transmitted; it can be extracted from the templates at the central server directly.

After this, the minutia matching is performed at the central server. The successful minutia matching at last verifies the claimed user. Future work will focus on the study of stable features (as part of encryption key) of fingerprint image, which may help to set up a fingerprint matching dictionary so that to narrow down the workload of fingerprint matching in a large database.

REFERENCES

1. Bhanu Bir, Tan Xuejun, Computational Algorithms for Fingerprint Recognition
2. Kluwer Academic Publishers, 20 Das, K. *Design and Implementation of an Efficient Thinning Algorithm*
3. Bachelor of Technology thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur
4. Hong, L., Wan, Y. and Jain, A. Fingerprint Image Enhancement: Algorithm and Performance Evaluation
5. IEEE Transactions on Pattern Analysis and Machine Intelligence,1998