Biometric Encrypted Secure Storage

Project Report submitted in partial fulfillment of the requirement
for the degree of

Bachelor of Technology.

in

Computer Science & Engineering

under the Supervision of

Mr. Amit Kumar Singh

By

Rajat Paliwal (111319)

to

Jaypee University of Information and Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that the work titled "**BIOMETRIC ENCRYPTED SECURE STORAGE"** submitted by **Rajat Paliwal** in the partial fulfillment for the award of degree of Bachelor of Technology in Information Technology from Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor:

Name of Supervisor	:	Mr. Amit Kumar Singh
Designation	:	Assistant Professor
Date	:	

Acknowledgement

I would like to express my gratitude to all those who gave us the possibility to complete this project. I want to thank the Department of CSE & IT in JUIT for giving us the permission to commence this project in the first instance, to do the necessary research work. I would also like to convey my sincere thanks to Dr. Satya Prakash Ghrera.

I am deeply indebted to my project guide Mr. Amit Kumar Singh, whose help, stimulating suggestions and encouragement helped me in all the time of research on this project. I feel motivated and encouraged every time I get his encouragement. For his coherent guidance throughout the tenure of the project, I feel fortunate to be taught by him, who gave me his unwavering support.

Date: Rajat Paliwal

ABSTRACT

In modern world, securing information is a very important task. Cryptographic systems are integral part of any system. Security requirements demand that these system be operated with very large secret keys [1]. Since it is very difficult to remember very large keys, Biometric System are used for authenticated access. In the traditional Biometric System users are identified on the basis of stored Biometric image or Biometric template of a person. Biometric authentication system have many weak points, one of them is theft of Biometric template database [2]. In this case user can’t use stolen Biometric again due to high risk of misuse. Biometric encryption is a security scheme that combines the Biometric template with a strong cryptographic scheme to provide better security by storing a key instead of Biometric template [2].

In this project a system has been designed for secure storage of data, by providing authentication by fingerprint biometric and for the security of biometric template, biometric encryption is used where a random key is generated, which is unknown to everyone. System generates another key from random key and biometric template using biometric encryption algorithm, generated key is converted into barcode and provided to user for access. System deletes biometric template form memory and stores random key as match key in database. During authentication User provides barcode and fingerprint, from them a key is generated which is matched with the stored key. If match found then authentication granted.

Database is dependent on Object Relational Mapping therefore it is being created directly from the persistence classes. Object Relational Mapping is defined as mapping the object from the object oriented environment into the table of relational database environment.

Table of Content

S. No.	Topic	Page No.
1	Biometric: An Introduction	1
1.1	Objective	1
1.2	Purpose	1
1.3	Scope	1
1.4	Description	1
1.4.1	Biometrics	1
1.4.2	Use of Biometrics	2
1.4.3	Biometric characteristics	4
1.4.4	Application of Biometrics	5
1.4.5	Advantages of Biometrics	6
1.4.6	Biometric Encryption	7
1.4.7	Verification and Identification	9
1.4.8	Barcode	10
2	Literature Review	12
3	Proposed Design	15
3.1	Software Requirements	15
3.1.1	Functional Requirements	15
3.1.2	Non-Functional Requirements	15
3.1.3	Logical Database Requirements	16
3.1.4	User Interface Requirements	16
3.2	Proposed Methods	16
3.2.1	Image Preprocessing	16
	• Image Segmentation	16
	• Binarization	17

- Thinning 17
- Feature Extraction 17

3.2.2 Biometric Encryption 17

- Random key Generation 17
- Key Generation/ Retrieval 17

3.2.3 Database Access (Hashing) 18

3.3 Software Design 18

3.3.1 Data Flow Diagram 19

- Context Level DFD 19
- DFD Fragments 20

3.3.2 UML Diagrams 24

- Use Case Diagram 24
- State Machine Diagram 26
- Activity Diagram 27

3.3.3 Database Design 28

- ER Diagram 28

3.4 Technology Used 29

3.4.1 Java 29

3.4.2 Hibernate ORM 29

3.4.3 Hibernate Query Language 29

3.4.4 J2EE 29

3.4.5 UML 30

3.5 Softwares Used 30

3.5.1 Netbeans IDE 30

3.5.2 Visual Paradigm 30

3.6 Database Used 30

3.6.1 MySql 30

3.7 Performance Metric 31

4 Implementation 32

4.1 Algorithms 32

4.1.1 Image Threshold 32

4.1.2 Image Binarization 33

4.1.3 Image Thinning or skeletonization 34

4.1.4 Cross number method 36

4.1.5 Binder module 37

4.1.6 Extractor module 38

4.1.7 KeyGenerator module 39

4.1.8 Database module 39

- HelperData 39
- Create Database 40

4.1.9 Enrollment 41

4.1.10 Verfication 42

4.1.11 Biometric Encryption 44

4.1.12 Biometric Decryption 44

4.1.13 Bacode Encoder 44

4.1.14 Barcode Decoder 45

4.2 Screenshots 45

4.2.1 Admin Module 45

4.2.2 Fingerprint module 47

4.2.3 User Module 51

4.3 Memory Usage 55

5 Experimental Analysis 56

5.1 Security of Biometric 56

6 Conclusion & Future work 57

7 Bibliography 58

List of Figures

S.No.		Title	Page No.
1		Introduction	
	1.1	Biometric Types	2
	1.2	Key Retrieval	8
	1.3	Key Binding	9
	1.4	Verification	9
	1.5	Identification	10
	1.6	1D and 2D Barcode	11
2		Literature Review	
3		Proposed Design	
	3.1	Context Level DFD	19
	3.2	Create Database	20
	3.3	Delete User	20
	3.4	Add/Delete Information	21
	3.5	Enroll User	22
	3.6	Login User	23
	3.7	Fingerprint Template Preprocessing	24
	3.8	Overall Use case Diagram	25
	3.9	Create Database	26
	3.10	Validation State	26
	3.11	Create Database	27
	3.12	Login User	27
	3.13	Overall ER Diagram	28
4		Implementation	
	4.1	Homepage	45
	4.2	Create Database	46
	4.3	Create Database Message	46

4.4	Fingerprint Preprocessing	47
4.5	Choose image from database	48
4.6	After Binarization	48
4.7	After Thinning	49
4.8	On Extraction	49
4.9	After Extraction	50
4.10	Enroll User	51
4.11	Barcode Encode/Decode	51
4.12	Encoding Key	52
4.13	Decoding Barcode	52
4.14	After Decoding	53
4.15	Person Verifier	53
4.16	Verification of User	54
4.17	Memory Requirement	55
4.14	Live Memory Snapshot	55

List of Tables

S.No.	Title	Page No.
1.1	Authentication Solutions and employing Biometrics	6
2.1	Literature Review	14

CHAPTER 1

Biometric: An Introduction

1.1 Objective

To implement a positive sum technology that achieves strong authentication, security[1] and privacy using biometrics with double check, with special focus on creating a system for preventing biometric theft using biometric encryption [2-3].

1.2 Purpose

To create a positive sum biometric authentication system which removes the threat of biometric template theft and additional security required for the security of biometric template and provide an easy and secure way to access data.

1.3 Scope

- Authentication of user using Fingerprint biometric.
- Biometric encryption for ensuring biometric template security.
- Implementation of positive sum and double check methodology.

1.3 Description

In this segment basic terms will be defined which are going to be used in this project.

1.4.1 Biometrics

Nearly 40 years ago, IBM suggested that a computer user should be recognized at a computer terminal “By something he knows or memorizes…By something he carries….By a personal characteristics”. Biometrics is the science of establishing the identity of a person on the physical, chemical or behavioral attributes.

Biometrics is a characteristics which uniquely defines a person. E.g. Fingerprint, face, ear, signature, voice, keystroke pattern, gait etc.

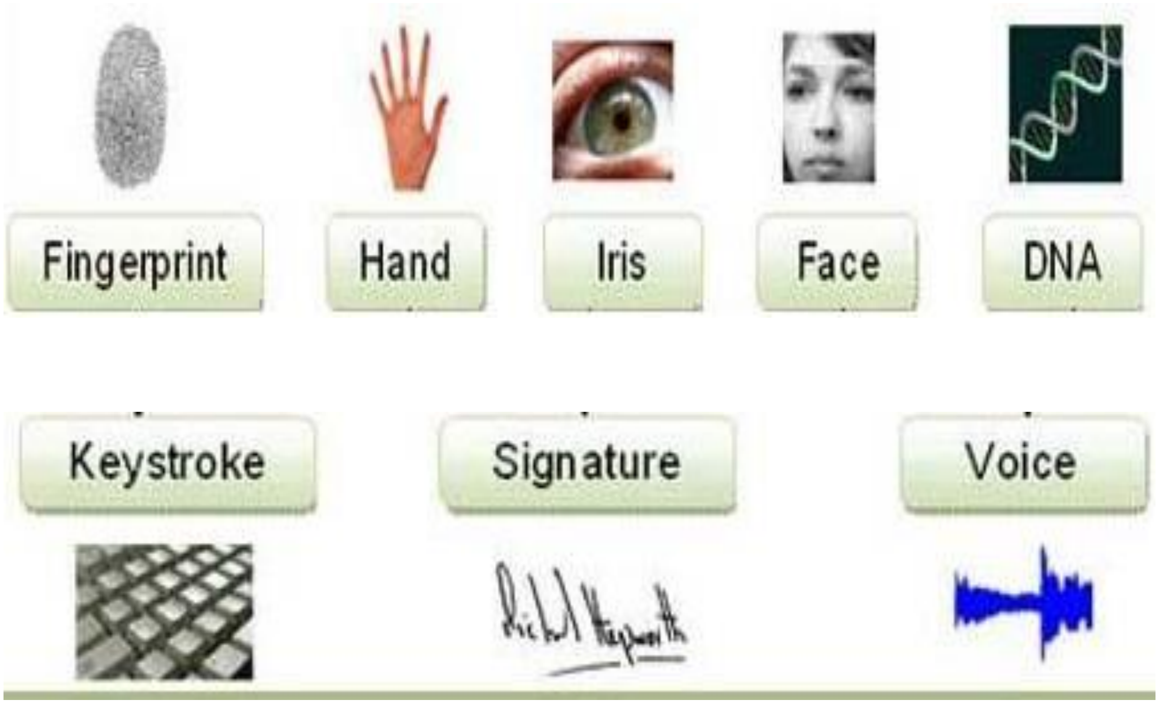


Figure 1.1 **Biometric types [1]**

Biometrics is an automated system which uniquely identifies a person on the basis of their characteristics. E.g. US-VISIT(a border security system of United States) etc[1]. In this project Biometrics will be used for both system and characteristics.

1.4.1 Use of Biometrics

With the proliferation of information exchange across the Internet, and the storage of sensitive data on open networks, cryptography is becoming an increasingly important feature of computer security [4]. Many cryptographic algorithms are available for securing information. In general, data will be secured using a symmetric cipher system, while public-key systems will be used for digital signatures and for secure key exchange between users. However, regardless of whether a user deploys a symmetric or a public-key system, the security is dependent on the secrecy of the secret or private key, respectively. Because of the large size of a cryptographically-strong key, it would clearly not be feasible to require the user to remember and enter the key each time it is required. Instead, the user is typically required to choose an easily remembered passcode that is used to encrypt the cryptographic key. This encrypted key can then be stored on a computer's hard drive. To retrieve the cryptographic key, the user is prompted to enter the passcode, which will then be used to decrypt the key.

There are two main problems with the method of passcode security. First, the security of the cryptographic key, and hence the cipher system, is now only as good as the passcode. Due to practical problems of remembering various passcodes, some users tend to choose simple words, phrases, or easily remembered personal data, while others resort to writing the passcode down on an accessible document to avoid data loss.

Obviously these methods pose potential security risks. The second problem concerns the lack of direct connection between the passcode and the user. Because a passcode is not tied to a user, the system running the cryptographic algorithm is unable to differentiate between the legitimate user and an attacker who fraudulently acquires the passcode of a legitimate user.

As an alternative to passcode protection, biometric authentication offers a new mechanism for key security by using a biometric to secure the cryptographic key. Instead of entering a passcode to access the cryptographic key, the use of this key is guarded by biometric authentication. When a user wishes to access a secured key, he or she will be prompted to allow for the capture of a biometric sample. If this verification sample matches the enrollment template, then the key is released and can be used to encrypt or decrypt the desired data. Thus, biometric authentication can replace the use of passcodes to secure a key. This offers both convenience, as the user no longer has to remember a passcode, and secure identity confirmation, since only the valid user can release the key.

So, in the traditional security system User secures his data by using very large secret key, which must be presented during the authentication. i.e. System recognizes a person on “what the user remembers”. Biometrics offers a natural and reliable solution to certain aspects of utilizing fully automated or semi-automated schemes to recognize individuals based on their biological characteristics. By using Biometrics It is possible to establish an identity based on “who you are”, rather than “what you remember”.

In some systems biometrics may be used with the passwords to increase the level of security. Such an arrangement is called dual factor scheme. E.g. Use of Id cards with the fingerprint reader etc.

Biometrics also offers additional advantages due to their high uniqueness, which can't be provided by the passwords. These advantages are Negative recognition and Non-repudiation. Negative recognition is the process by which system determines that a certain person is indeed enrolled in system although the individual may deny it. Non-repudiation is a way to guarantee that an individual who accesses a certain facility cannot later deny using it.

1.4.2 Biometric Characteristics

Various number of Biometrics are being used in various applications. Each Biometric has its pros and cons, and therefore choice of a Biometric for a particular application depends upon a variety of issues [1]. Following are the seven factors identified that determine the suitability of a physical or behavioral characteristics to be used in an application:

1. **Universality:** Biometric trait should be possessed by every person targeted by the application.
2. **Uniqueness:** Biometric trait should be highly unique per person.
3. **Permanence:** Biometric trait should be sufficiently invariant over the time for matching.
4. **Measurability:** It should be possible to acquire, digitize and extract features from the Biometric trait without causing inconvenience.
5. **Performance:** The recognition accuracy and the resources required to achieve that accuracy should meet the constraints imposed by application.
6. **Acceptability:** Users of application should be willing to provide the Biometric trait to the system.

7. **Circumvention:** It refers to the ease with which biometric trait can be faked or imitated.

1.4.4 Application of Biometrics

Establishing the identity of a person with high confidence is becoming critical in a number of applications in our vastly interconnected society. Questions like "Is she really who she claims to be?", "Is this person authorized to use this facility?" or "Is he in the watchlist posted by the government?" are routinely being posed in a variety of scenarios ranging from issuing a driver's license to gaining entry into a country. The need for reliable user authentication techniques has increased in the wake of heightened concerns about security, and rapid advancements in networking, communication and mobility. Thus, biometrics is being increasingly incorporated in several different applications. These applications can be categorized into three main groups:

1. Commercial applications such as computer network login, electronic data security, e-commerce, Internet access, ATM or credit card use, physical access control, mobile phone, PDA, medical records management, distance learning, etc.

2. Government applications such as national ID card, managing inmates in a correctional facility, driver's license, social security, welfare-disbursement, border control, passport control, etc.

3. Forensic applications such as corpse identification, criminal investigation, parenthood determination, etc.

Table 1.1 ***Authentication solutions employing Biometrics*** **[1]**

FORENSICS	**GOVERNMENT**	**COMMERCIAL**
Corpse Identification	National Id Card	ATM
Criminal Investigation	Driver license; Voter registration	Access control; Computer login
Parenthood determination	Welfare disbursement	Mobile Phone
Missing children	Border crossing	E-commerce; Internet; Banking; Smart card

Examples of few applications where biometrics are being used for authenticating individuals are presented below:

i. IBM made an addition of biometrics to personal mobile devices, which will enhance the security surrounding corporate “Bring your own device” (BYOD) policies.

ii. ZTE a global manufacturer of smartphones and mobile devices, recently launched its flagship ZTE Star 2 product, a new 4G LTE voice controlled smartphone. ZTE Star 2’s voice control functions allow users to unlock the device with their own unique voice tone, make phone calls, open third-party apps, play music, take photos and much more.

iii. Touch ID by apple for iOS 8, User can unlock phone by just putting finger on home button and it also enables user to make purchases from iTunes, iBooks and the App store. And for first time, developers can integrate Touch ID into third party apps.

1.4.5 Advantages of Biometrics

Biometrics have many advantages over traditional authentication system, some of them are mentioned below:

1. Biometrics cannot be lost, stolen or forgotten. Barring disease or serious physical injury, the biometric is consistent and permanent.

2. Biometrics are unique, It is also secure in that the biometric itself cannot be socially engineered, shared or used by others.

3. There is no requirement to remember passwords, or PINs, thus eliminating overhead cost. The biometric is always available to individual.

4. Biometrics provides a high degree of confidence in user identity.

1.4.6 Biometric Encryption

In the traditional Biometric systems, Biometric image or Biometric template of the user is stored for matching during the authentication. This database of Biometric template has to be highly secured because of its risk of misuse on being stolen. If the database has been stolen then user cannot use that biometric trait in system again because biometric trait is unique and someone else also has that now, which can be used to get access of the system. This type of threat can be solved if biometric template or image is not stored in database. Biometric encryption provides this solution by storing a random generated key in the database for matching.

Biometric Encryption (BE) is a group of emerging technologies that securely bind a digital key to a biometric or generate a digital key from the biometric, so that no biometric image or template is stored.

This innovative technique for securing a key using a biometric has been developed by Mytec Technologies Inc., based in Toronto Canada. The solution developed by Mytec does not use an independent, two-stage process to first authenticate the user and then release the key. Instead, the key is linked with the biometric at a more fundamental level during enrollment, and is later retrieved using the biometric during verification. Furthermore, the key is completely independent of the biometric data, which means that, firstly, the use of the biometric is not forfeited if the key is ever compromised, and secondly, the key can be easily modified or updated at a later date. The process developed by Mytec Technologies is called Biometric Encryption™. During enrollment, the Biometric Encryption process combines the biometric image with a digital key to create a secure block of data, known as a Bioscrypt™. The digital key can be used as a cryptographic key. The Bioscrypt is secure in that neither the fingerprint nor the key can be independently obtained from it. During verification, the Biometric Encryption algorithm retrieves the cryptographic key by combining the biometric image with the Bioscrypt. Thus, Biometric Encryption does not simply provide a yes/no response in user authentication to facilitate release of a key, but instead retrieves a key that can only be recreated by combining the biometric image with the Bioscrypt.

There are two BE approaches: key binding, when an arbitrary key (e.g., randomly generated) is securely bound to the biometric, and key generation, when a key is derived from the biometric. Both approaches usually store biometric dependent helper data. In the key binding mode, the digital key is randomly generated on enrollment so that neither the user nor anybody else knows it. The key itself is completely independent of biometrics, and therefore, can always be changed or updated. After a biometric sample is acquired, the BE algorithm securely and consistently binds the key to the biometric to create a biometrically encrypted key. The BE template provides privacy protection and can be stored either in a database or locally (smart card, token, laptop, cell phone, etc.). At the end of the enrollment, both the key and the biometric are discarded.

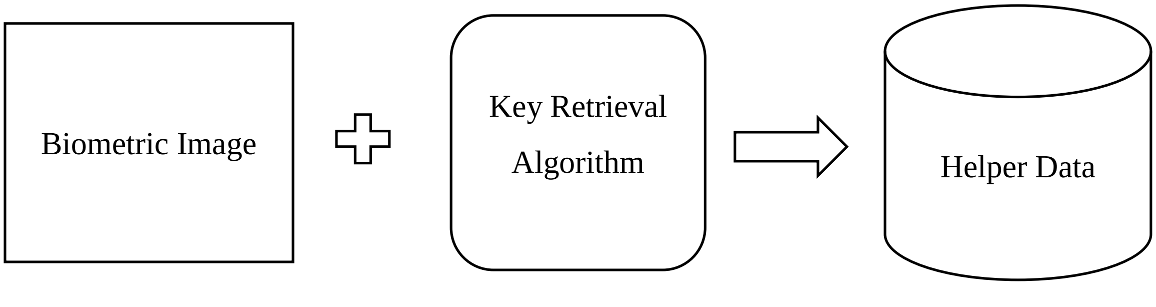


Figure 1.2 **Key Retrieval [2]**

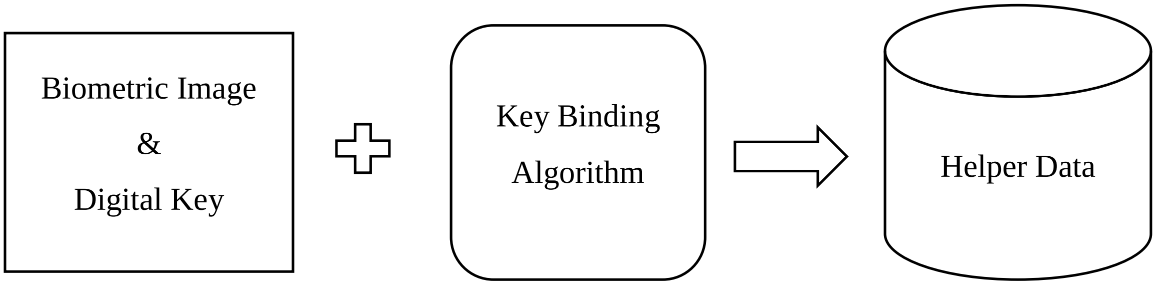


Figure 1.3 **Key Binding [2]**

1.4.7 Verification and Identification

Depending on the application context, a biometric system may operate either in verification or identification mode**[2]**.

In the verification mode, the system validates a person's identity by comparing the captured biometric data with her own biometric template(s) stored in the database. In such a system, an individual who desires to be recognized claims an identity, usually via a PIN, a user name or a smart card, and the system conducts a one-to-one comparison to determine whether the claim is true or not. Verification is typically used for positive recognition.

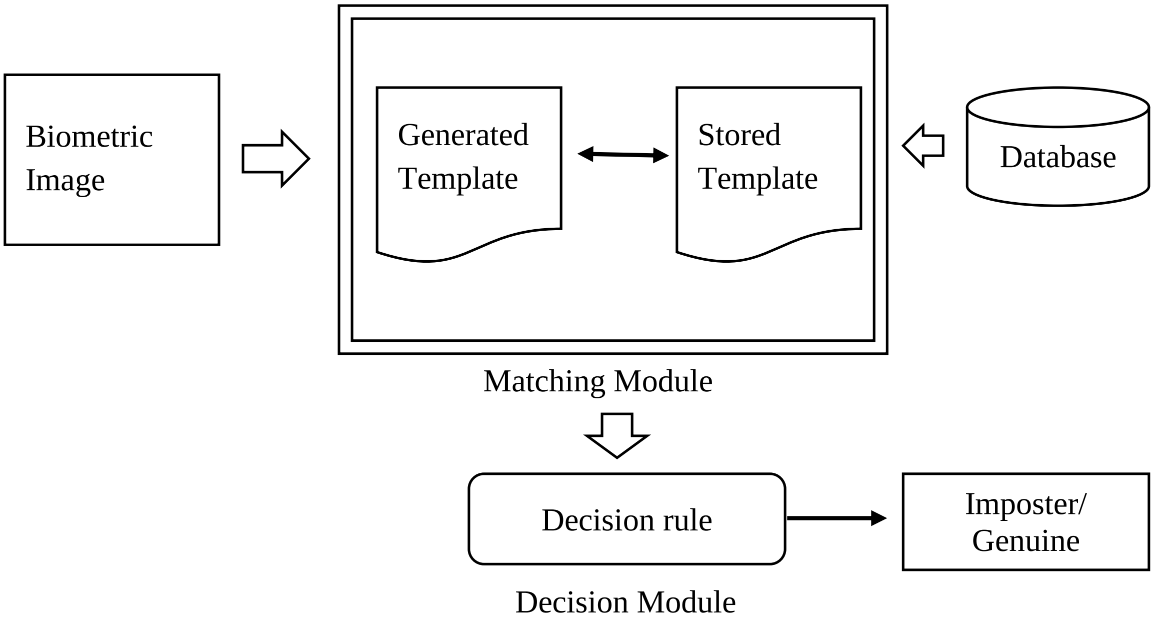


Figure 1.4 **Verification [2]**

In the identification mode, the system recognizes an individual by searching the templates of all the users in the database for a match. Therefore, the system conducts a one-to-many comparison to establish an individual's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity. Identification is a critical component in negative recognition applications where the system establishes whether the person is who she (implicitly or explicitly) denies to be.

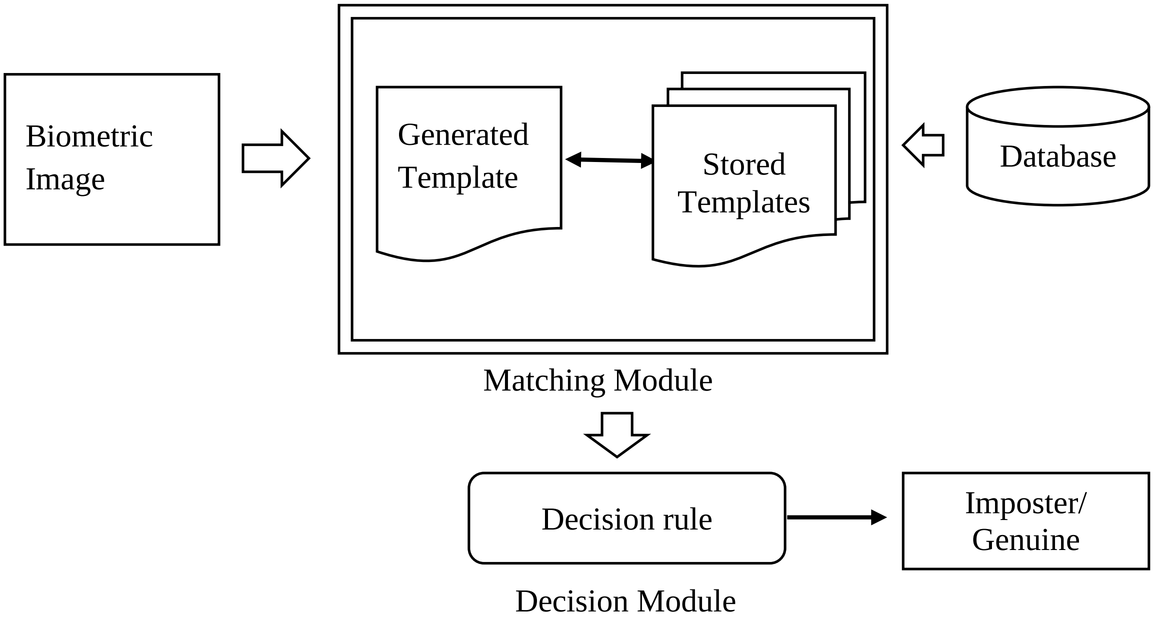


Figure 1.5 **Identification [2]**

1.4.7 Barcode

A barcode is an optical machine readable representation of data relating to the object to which it is attached. Originally barcodes systematically represented data by varying the widths and spacings of parallel lines, and may be referred to as linear or one-dimensional (1D). Later they evolved into rectangles, dots, hexagons and other geometric patterns in two dimensions (2D). Although 2D systems use a variety of symbols, they are generally referred to as barcodes as well. Barcodes originally were scanned by special optical scanners called barcode readers. Later, scanners and interpretive software became available on devices including desktop printers and smartphones.

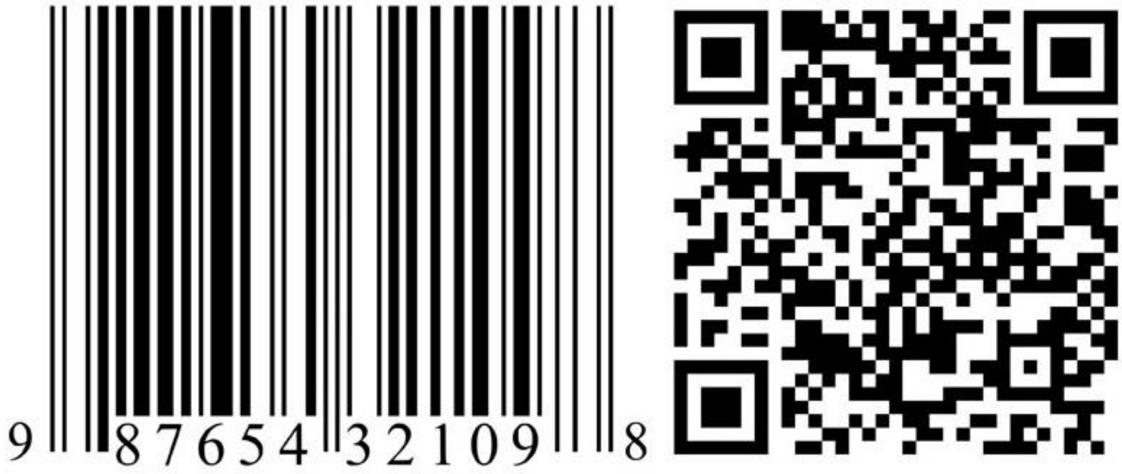


Figure 1.4 **1D and 2D Barcode**

CHAPTER 2

Literature Review

The overall goal of this chapter were firstly to establish the significance of biometric authentication system then identify a place where a new contribution could be made. The bulk of the chapter was on critically evaluating the different methodologies used in this field so as to identify the appropriate approach securing biometrics. Various papers and articles were studied for the understanding of system and approach for designing it. Some of the important ones are mentioned below.

1. Jiawei et al in [4] proposed to create a cost effective privacy preserving biometric authentication system, which can provide privacy protection against possible theft, loss or misuse of biometric data.

   Decomposition of system in three stages: biometric database processing, privacy preserving Fingercode comparison and final result generation. No storage of biometric template in the database. Matching of key on basis of Eucledian distance calculation.

2. Hogo et al in [5] proposed the generation of keys from fingerprints and pseudorandom number generator which increases the complexity space of key.

   Create fingerprint template from fingerprint using minutiae detection, generate a pseudorandom number and use the Fuzzy commitment for creation of the biometric key.

3. Upmanyu et al in [6] proposed a secure and blind biometric authentication protocol, which addresses the concerns of user's privacy, template protection and trust issues, while not affecting the performance and accuracy.

   Blind authentication protocol defined as "a biometric authentication protocol which does not reveal any information about the biometric samples to the authenticating server. It also does not reveal any information regarding the classifier, employed by the server or the client".

4. Bakhteri et al in [7] proposed a system for replacing cryptographic algorithms with a mix of biometric with cryptographic algorithm to generate biometric encrypted key. Fuzzy vault method for biometric encryption i.e. XOR of biometric template with random generated key.

5. Nagan et al in [8] proposed a method for using fuzzy vault with minutiae descriptors for better security. This method provides 16n+1 bit security. In this method fingerprint minutiae are extracted on basis of location So, orientation change does not affect the matching process.

6. Jin et al in [9] proposed a method for generating secure minutiae based fingerprint templates using Random Hash Triangle method and then from bit block coding it is converted into a string which takes less storage space.

Table 2.1 Existing Methods

Authors	**Technology**	**Results & Findings**
Jiawei et al[4], 2013	Biometric Identification	Less costly due to decomposition of System in three stages. Instead of Biometric template, encrypted key is stored in database.
Hogo, et al[5], 2012	Biometric encryption & random number generation	100% classification accuracy and generated keys are strong when compared with weak and semi weak DES keys.
Upmanyu et al [6], 2010	Client Server architecture	Blind authentication provided accuracy of 84.45%. Shows accuracy can be maintained while increasing the security by adding encryption.
Bakhteri et al [7], 2009	Biometric Encryption	Fuzzy vault method for biometric encryption and Image Processing methods.
Nagan et al[8], 2013	Biometric Template Security	Fuzzy vault provides security on basis of length of key. In the case of 16 bit key, 16n+1 bit security.
Jin et al[9], 2009	Biometric Template Generation	One way transformation of Biometric into Biometric template provides security.

CHAPTER 3

Proposed Design

3.1. Software Specifications

Software specifications is a set of requirements which enables a software developer to develop the software under some specific guidelines.

3.1.1 Functional Requirements

- Two types of user, Admin and Simple User referenced as User.
- Admin can create database, overwrite database & remove specific user.
- User can enroll himself using fingerprint into system.
- Generation of random key by random number algorithm.
- Barcode generation after enrollment of user from name, biometric template and random generated key.
- Store random key in the database and match biometric decrypted key.
- User can save the information into system and update it.
- User authentication using Fingerprint and barcode.

3.1.2 Non-Functional Requirements

- Multiple SQL databases compatibility.
- Positive sum privacy model.
- Scalable database and system architecture.
- Easy operability for users.
- High response time for accessing information.
- File format of fingerprint used is .jpeg.
- Small enrollment time.

3.1.3 Logical Database Requirements

- Only the users who have authentication can access and update their information.
- Admin can delete any specific user.
- Database can be created by admin directly.
- Tightly bound database system with the system.

3.1.4 User Interface Requirements

- Window interface for using the system.
- Display of fingerprint during enrollment and display of barcode after the enrollment.
- Accessing the required window by choice selection user interface.
- Homepage for system should be choice between types of users.
- Guided interface for interaction with system.

3.2 Proposed Methods

For the implementation of the biometric encrypted secure storage system, following methods has been decided:

3.2.1 Image Preprocessing

Before template can be created it is necessary to process fingerprint image. This process follows following steps:

- **Image Segmentation**

  Image segmentation is performed to differentiate between foreground and background of the fingerprint image. The foreground is the fingerprint with ridges and valleys, whereas the background is outside the fingerprint border.

- **Binarization (Otsu’s method)**

  Binarization is the process to convert the grey-scale fingerprint images to pure black-and-white images. This is done by using a threshold value and converting all pixel values below the threshold into ‘0’ and all pixel with values above the threshold are assigned the value of ‘1’. Otsu’s method will be used for calculating threshold and binarization [10].

- **Thinning (Holt’s method)**

  Thinning which is also known as skeletonization is the process to reduce all lines in an image to single pixel width. Thinning is implemented by repeatedly deleting edge point pixels as long as the pixel is not an endpoint and the connectivity of the skeleton is not interrupted. Holt’s method will be used with staircase elimination for thinning.

- **Feature Extraction (Cross number method for minutiae)**

  Feature extraction is the process of extracting unique characteristics of a biometric trait. Minutiae extraction will be used here by using cross number method, where two types of minutiae will be extracted ridge ending and ridge bifurcation. It is also called Fuzzy vault.

3.2.2 Biometric Encryption

It is the process of binding key with biometric data to create new key.

- **Random key generation (Random number)**

  Random number generator will be used for creation of random keys.

- **Key generation/retrieval (Fuzzy commitment)**

  In the Fuzzy commitment method key and biometric would be XORed to create another key which will be used for retrieval by again this method during matching.

3.2.3 Database Access (Hashing)

In order to access the database hashing will be used after matching. Hashing is the process of storing data by use of a preselected Hash function. It provides the O(n) search complexity.

3.3 Software Design

The main focus of the analysis phase of Software development is on “What needs to be done”. The objects discovered during the analysis can serve as the framework or Design. The class’s attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution.

Emphasis shifts from the application domain of implementation and computer such as user interfaces or view layer and access layer. During analysis, we look at the physical entities or business objects in the system, that is, which players and how they cooperate to do the work of the application. These objects represent tangible elements of the business.

During the Design phase, we elevate the model into logical entities, some of which might relate more to the computer domain as people or employees. Here his goal is to design the classes that we need to implement the system the difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way to interact with a user or present information.

Design process:

During the design phase the classes identified in object-oriented analysis Must be revisited with a shift focus to their implementation. New classes or attribute and Methods must be an added for implementation purposes and user interfaces.

The following are some of the viwes of software design life cycle. They are

- Data Flow Diagrams
- UML Diagrams
- Data Base Design

3.3.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. It can also be used for the visualization of data processing (structured design).

There are two types of DFDs. They are:

- Context Level DFD
- Top Level DFD

- **Context Level DFD**

In the Context Level the whole system is shown as a single process.

- No data stores are shown.
- Inputs to the overall system are shown together with data sources (as External entities).
- Outputs from the overall system are shown together with their destinations (as External entities).

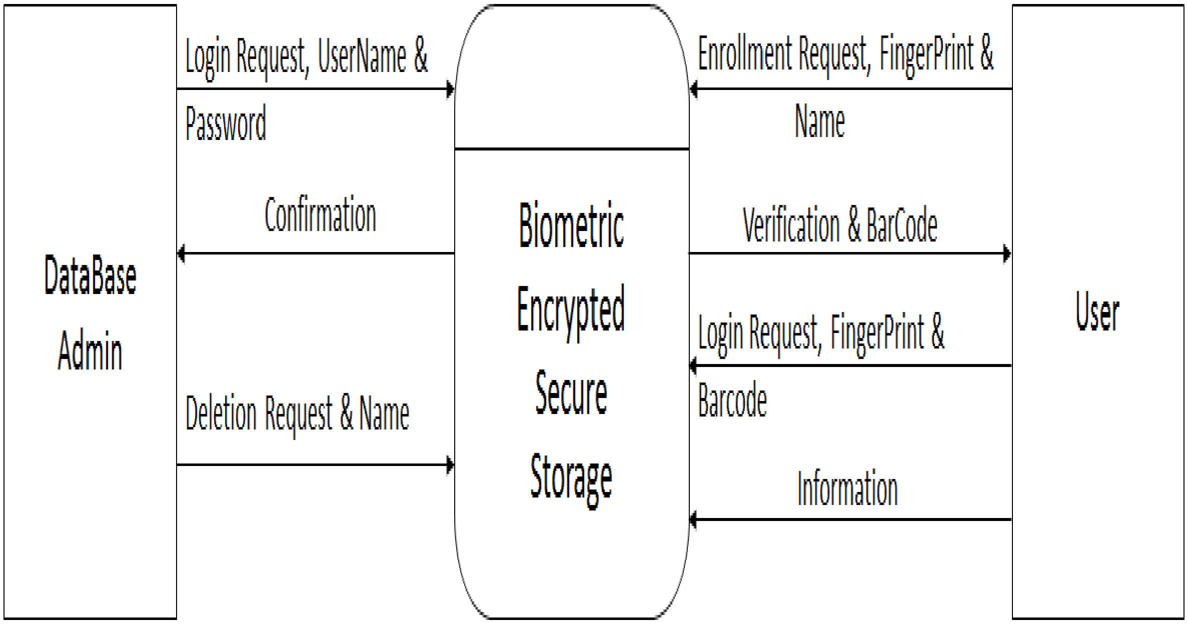


Figure 3.1 **Context Level DFD**

- **DFD Fragments**

  A DFD Fragment is a portion of event-partition system model that shows the process, external agent, data stores and data flow needed to respond to an event. Each fragment only shows those data stores that are actually needed to respond to the event.

- **Create Database**

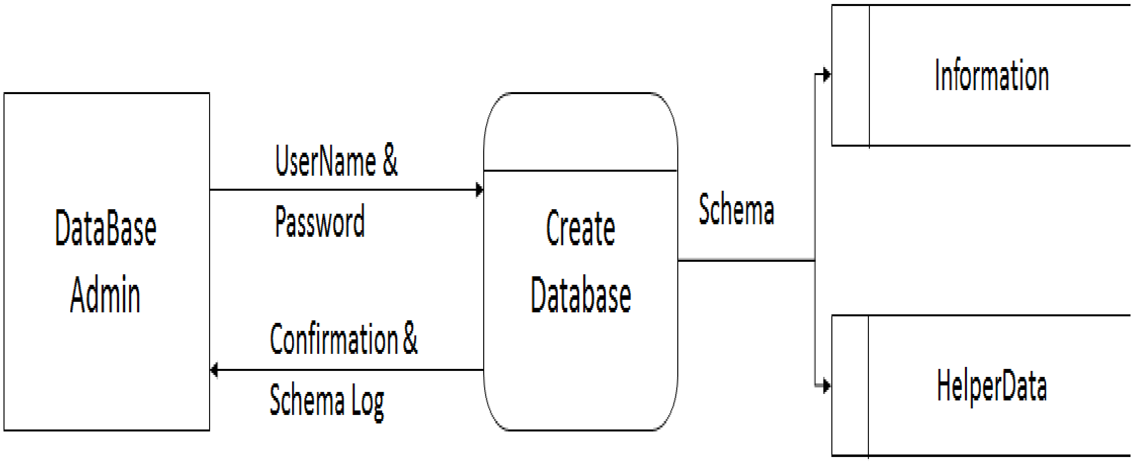


Figure 3.2 **Create Database**

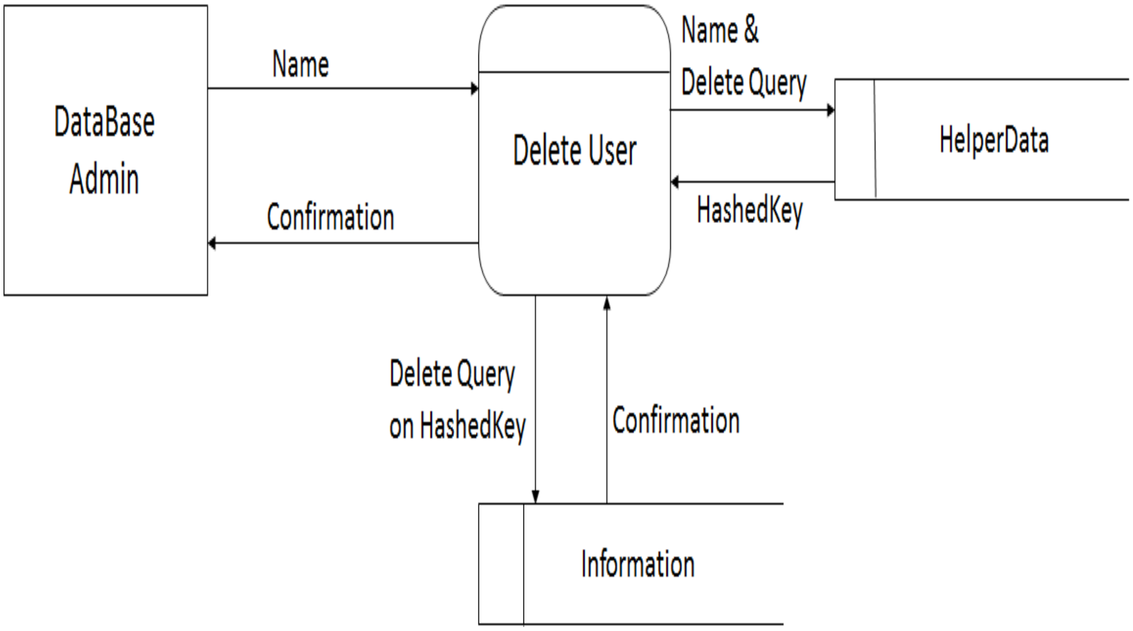


Figure 3.3 **Delete User**

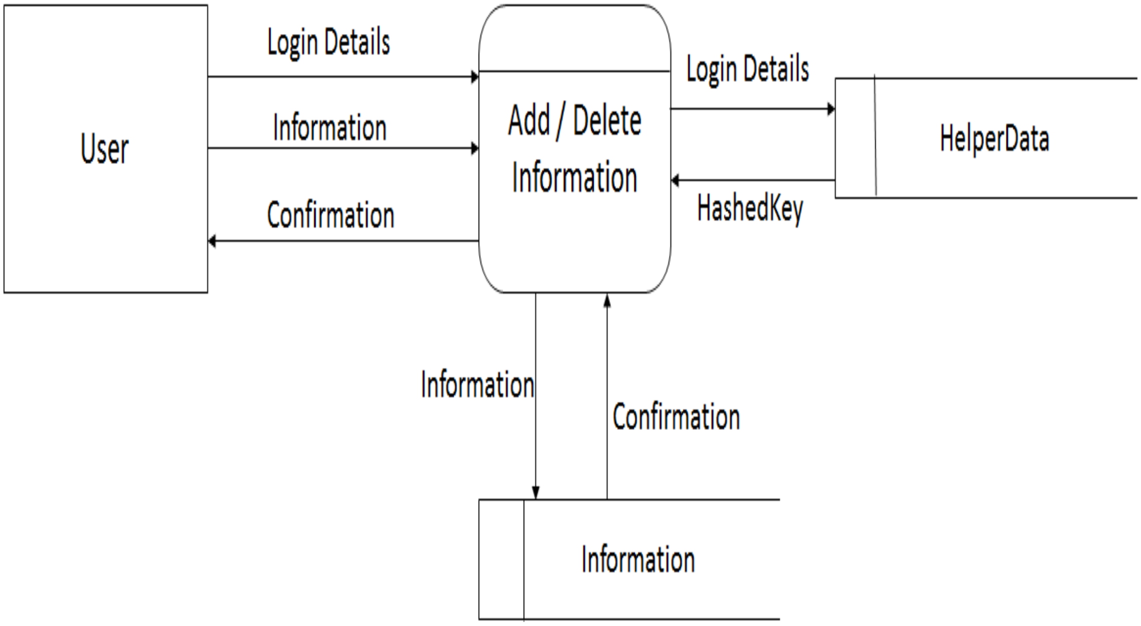


Figure 3.4 **Add/Delete Information**

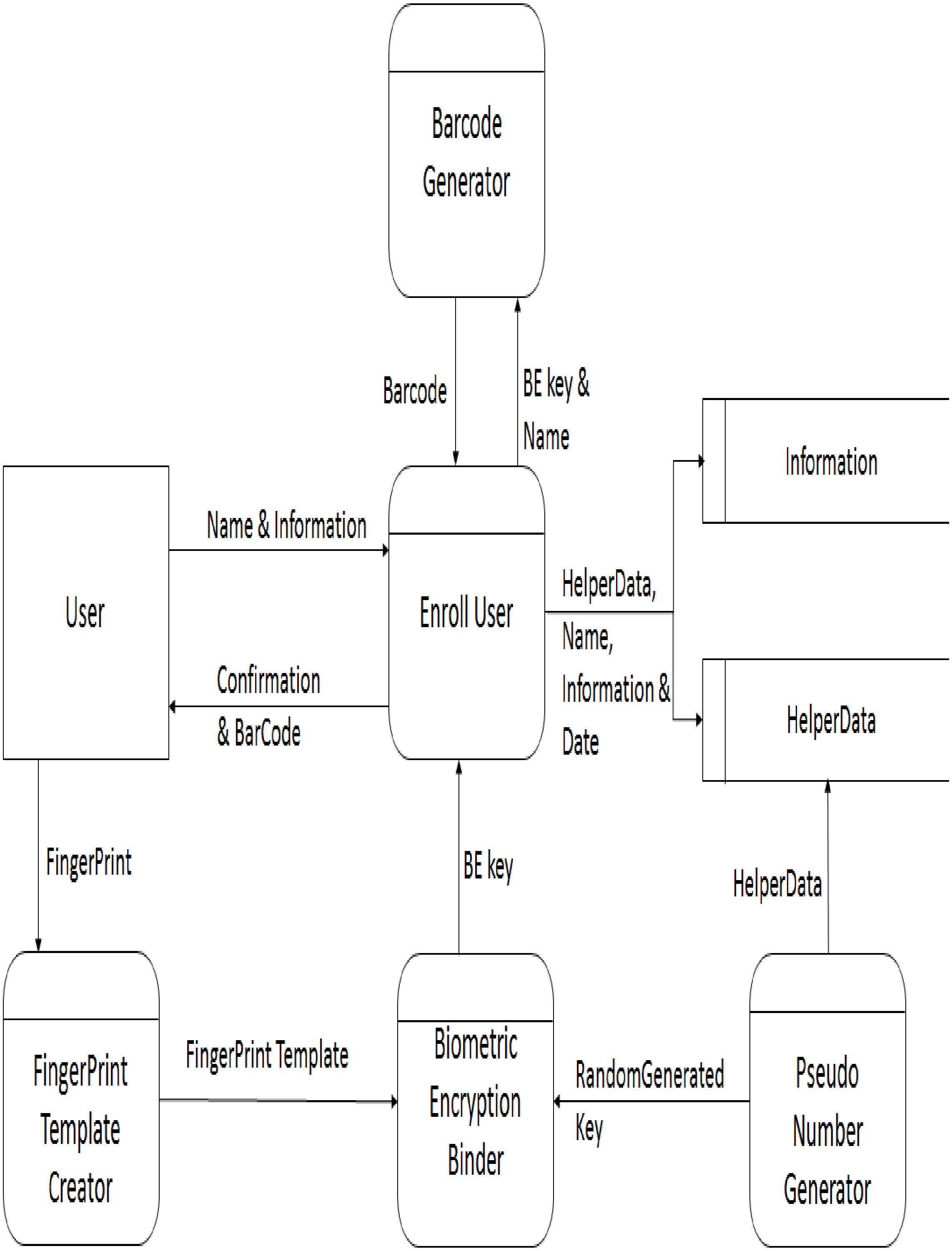


Figure 3.5 **Enroll User**

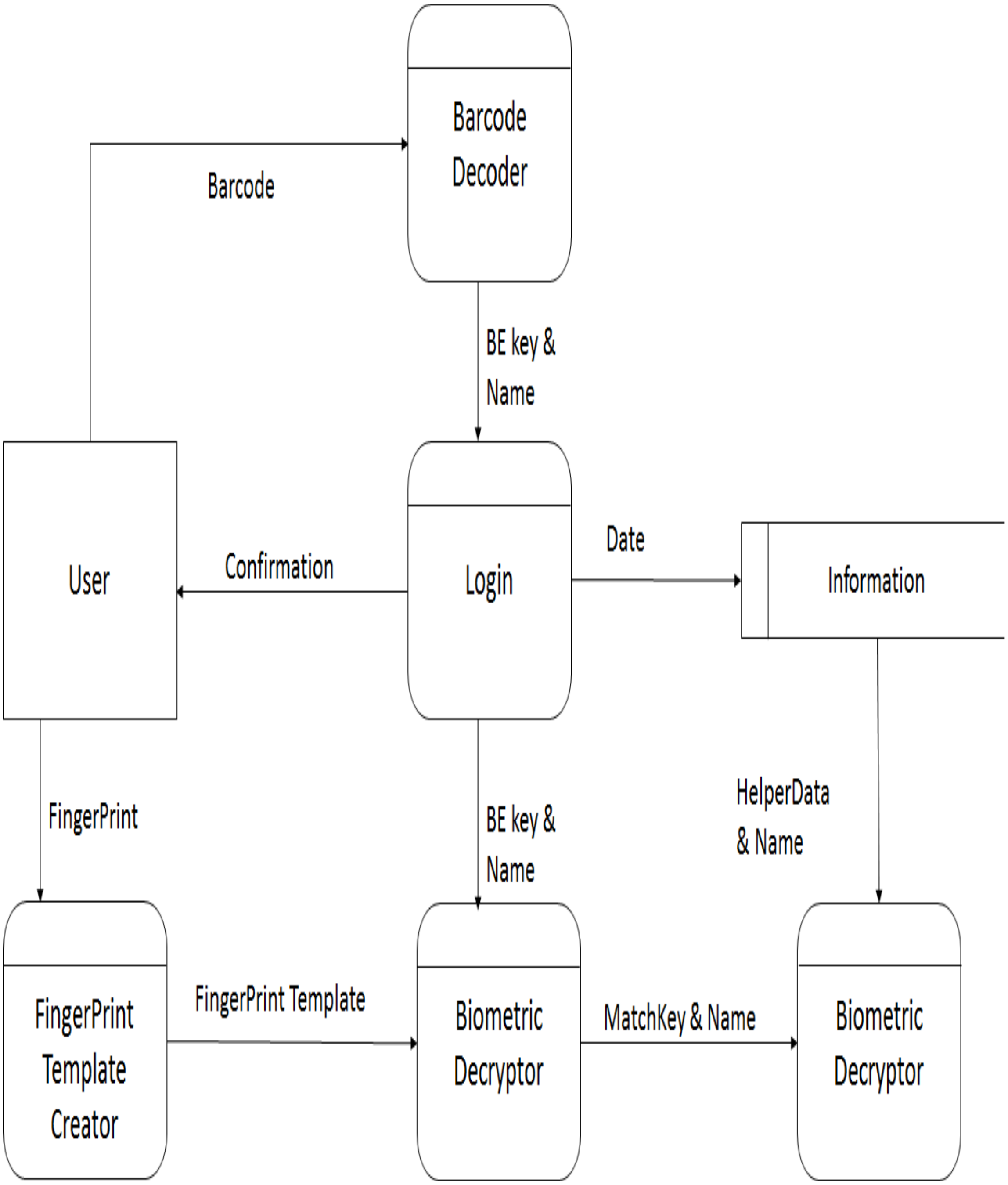


Figure 3.6 **Login User**

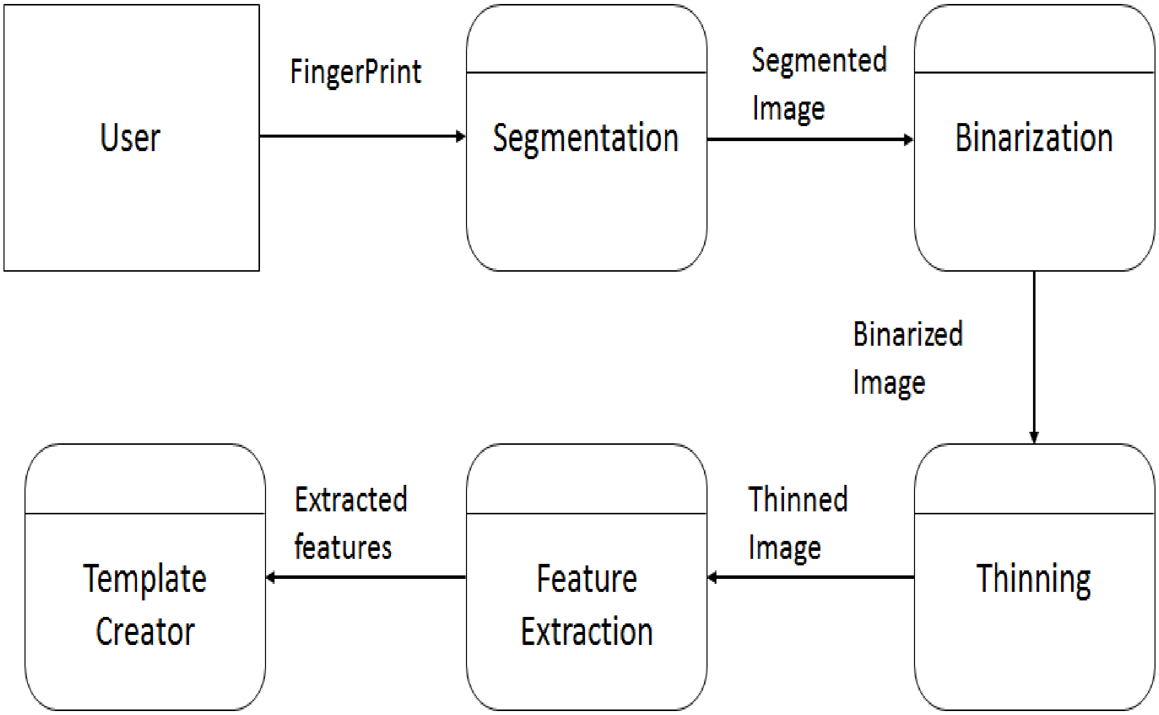


Figure 3.7 **Fingerprint Template Preprocessing**

3.3.2 UML Diagrams

Unified Modeling Language

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

- **Use Case Diagram**

  This is the overall Use Case diagram of system. Two types of actors are in the diagram user and admin. Admin has direct access to database by password inbuilt with system, which cannot be changed. Admin can create new database and delete user from the database. User can enroll in the system and after enrollment access his information. During enrollment barcode is generated which is used with fingerprint for authentication of user. Enrollment includes biometric encryption, Random number generation, barcode generation processes and authentication extends biometric decryption, barcode decoding and matching.

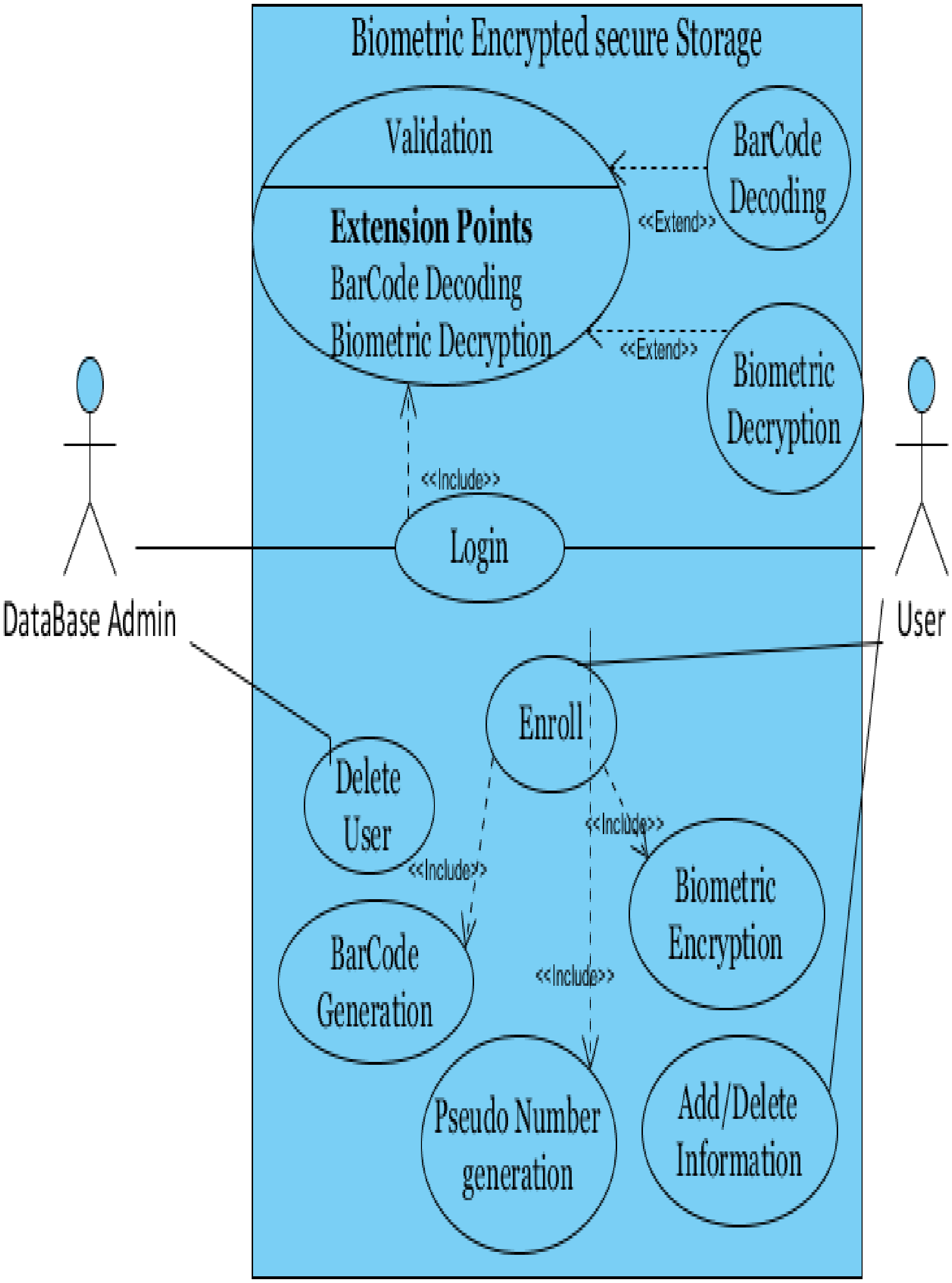


Figure 3.8 **Overall Use Case Diagram**

- **State Machine Diagram**

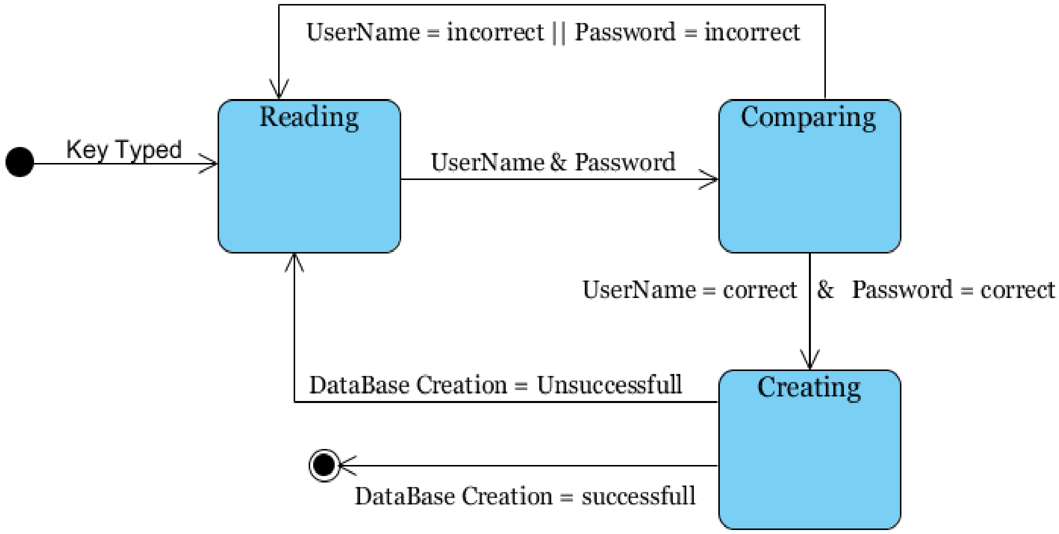

UserName = incorrect || Password = incorrect
Key Typed
Reading
UserName & Password
Comparing
UserName = correct
&
Password = correct
DataBase Creation = Unsuccessfull
Creating
DataBase Creation = successfull

Figure 3.9 **Create Database**

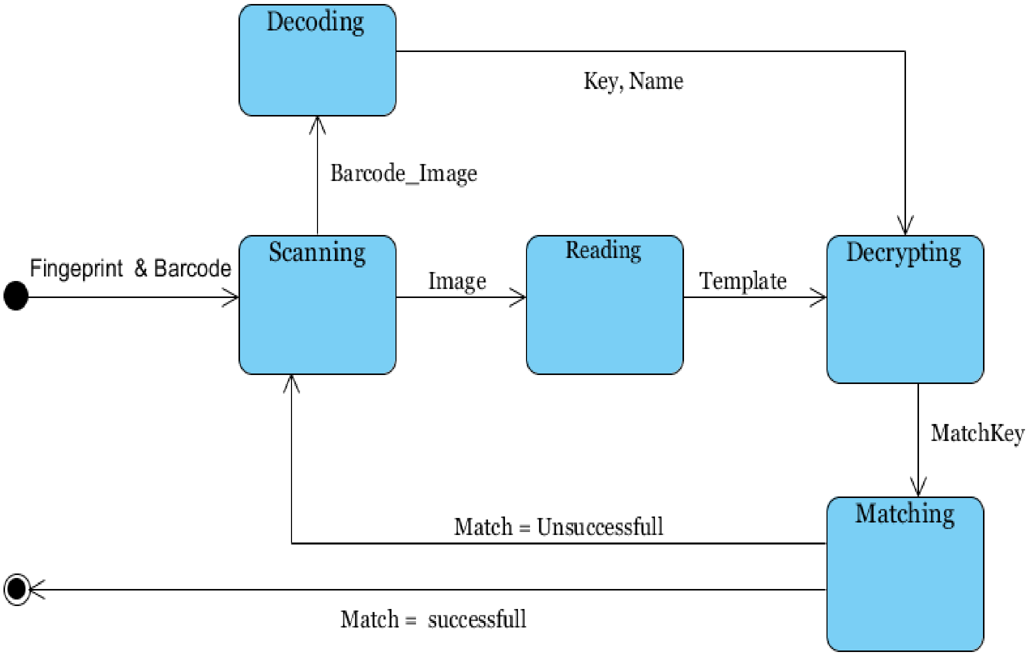


Figure 3.10 **Validation State**

- **Activity Diagram**

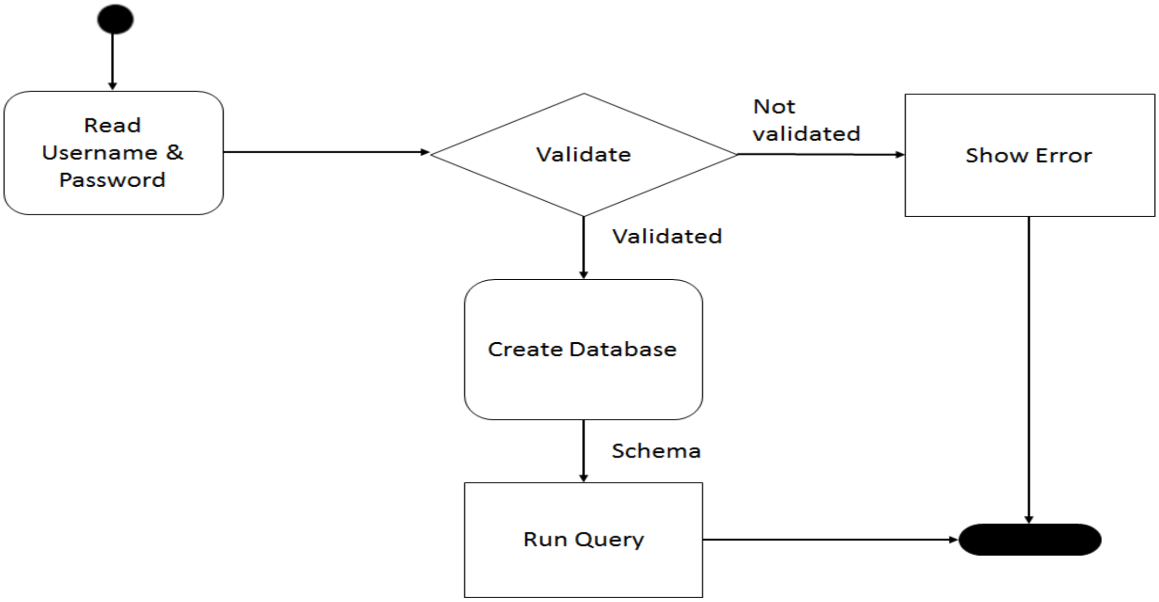


Figure 3.11 **Create Database**

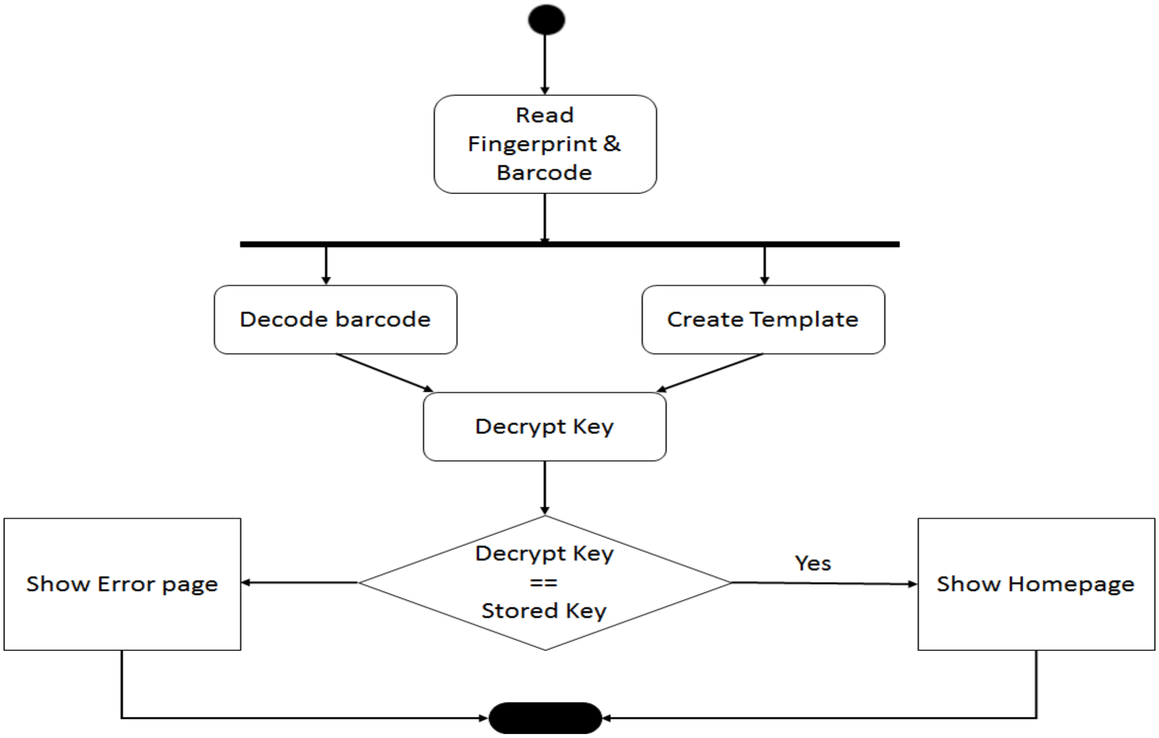


Figure 3.12 **Login User**

3.4 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views.

In an object database the entities and relationships map directly to object classes and named relationships.

3.3.1 ER Diagram

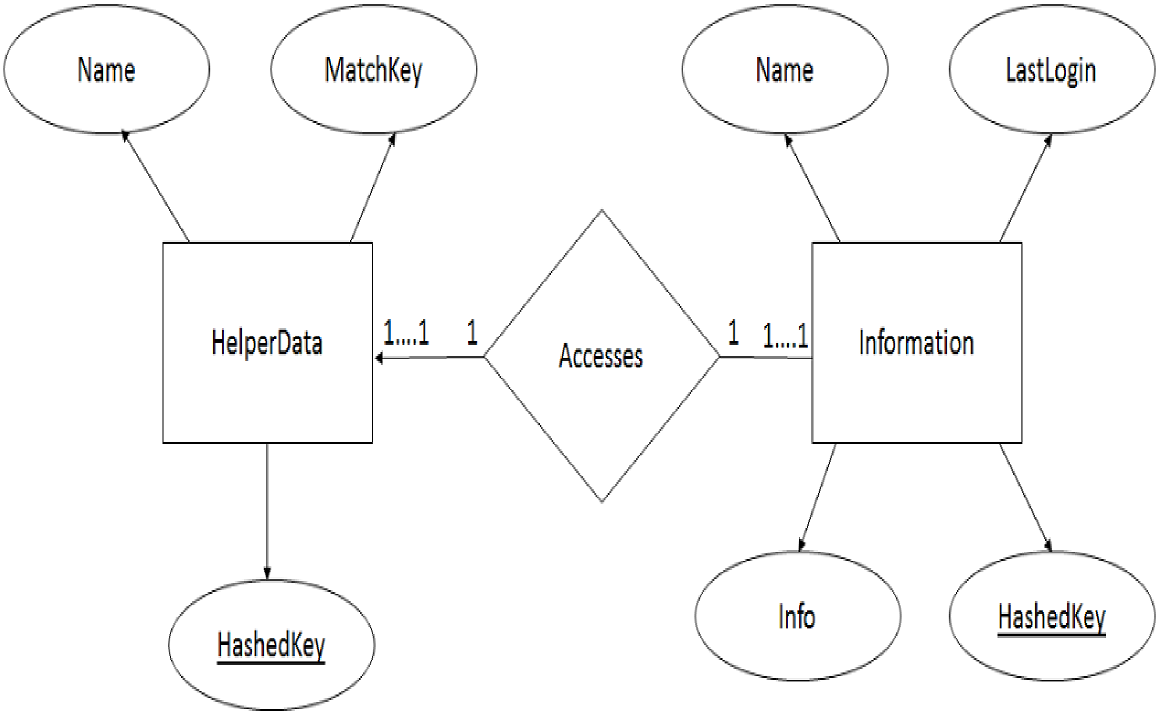


Figure 3.13 **Overall ER Diagram**

3.4 Technology Used

3.4.1 Java

Java is the foundation for virtually every type of networked application and is the global standard for developing and delivering embedded and mobile applications, games, Web-based content, and enterprise software. Latest version of java (java 8.1) has been used to develop the software for BESS.

3.4.2 Hibernate ORM

Hibernate ORM is an open-source high performance Object Relational Mapping library for the Java language, providing a framework for mapping an object-oriented model to a traditional domain model to a relational database. Hibernate solves the object relational mismatch problems by directly replacing direct persistence related database accesses with high-level object handling functions. Latest version of Hibernate (4.3.7) has been used to develop software for BESS.

3.4.3 Hibernate Query Language

Hibernate provides a SQL inspired language Hibernate Query Language (HQL). HQL is fully object-oriented and understands notions like inheritance, polymorphism and association. It can be used with any SQL database. HQL is used for querying the database for Creation, Retrieval, Update and Deletion.

3.4.4 J2EE

J2EE is abbreviated for Java 2 Platform Enterprise Edition. J2EE is a platform independent, Java centric from for developing, building and deploying Web based enterprise applications online. The J2EE platform consists of a set of services, APIs and protocols that provide the functionality for developing for developing multithread, web-based application. J2EE is used in BESS for making BESS architecture scalable and maintainable for future.

3.4.5 UML

UML is abbreviated for Unified Modelling Language. It is a general purpose modelling language in the field of software engineering, which is designed to provide a standard way to visualize the design of a system. It provides different types of modelling for system Flow model, Behavioral model, Interaction model and Structure model. Latest version of UML (2.0) is used for BESS.

3.5 Softwares Used

3.5.1 Netbeans IDE

Netbeans IDE is the official IDE for Java 8. It provides easy and efficient project management by providing versioning of code and management of folders. It provides rapid user interface development by drag-drop mechanism. It has been used for project management and user interface development of BESS. It will be used for testing and debugging in future.

3.5.2 Visual Paradigm

It is a design tool for UML supporting UML 2. In addition to modelling support, it provides report generation and code reverse engineering capabilities including code generation. It can reverse engineer diagrams from the code and provide round-trip engineering for various programming languages. It has been used in designing phase for creation of various UML diagrams like Use-case diagram, state machine diagram etc.

3.6 Database Used

Fingerprint image database used in BESS has been taken from FVC 2004 and FVC 2006, which will be used for checking the performance of the system with the results already available from these fingerprints and will be used for generation of performance matrix of BESS.

3.6.1 MySQL

MySQL database has been used for creation of project but it will work with any SQL database.

3.7 Performance Metric

For the proposed scheme Verification rate is used as the performance metric. Verification rate (VR) is defined as:

VR = (Number of continuous matches*100) / Key Length.

Verification rate defines whether a user is above threshold of verification or not. For the proposed scheme, 90% verification rate is set as threshold.

CHAPTER 4
Implementation

4.1 Algorithms

An algorithm is a specific set of instructions for carrying out a procedure or solving a problem, usually with the requirement that the procedure terminate at some point. Specific algorithms sometimes also go by the name method, procedure, or technique. The word "algorithm" is a distortion of al-Khwārizmī, a Persian mathematician who wrote an influential treatise about algebraic methods. The process of applying an algorithm to an input to obtain an output is called a computation.

4.1.1 Image Threshold

Converting a greyscale image to monochrome is a common image processing task. In computer vision and image processing , *Otsu's method* is used to automatically perform clustering-based image thresholding or, the reduction of a graylevel image to a binary image [10]. The algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels). Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum.

```
Thresold_Calculator ( BufferedImage original ) {
        histogram[] = imageHistogram ( BufferedImage original);
        total_no_pixels = ImageHeight * ImageWidth;

        for  i = 0 to 256 {
        sum = sum  + i * histogram[i]
        }
```

```
    for  i  = 0 to 256 {
    weight_Background = weight_Background + histogram[i]
    weight_Foreground = total_no_pixels – weight_Background
        if (weight_Background == 0)
        break;
    sumB = sumB + i * histogram[i]

    mean_Background = sumB / weight_Background
    mean_Foreground = (sum – sumB) / weight_Foreground
    class_variance = weight_Background * weight_Foreground * square(mean_Background – mean_Foreground)

    if ( class_variance > maximum_variance ){
        maximum_variance = class_variance
        threshold = i
    }
    }
    return threshold
}
```

4.1.2 Image Binarization

A binary image is a digital image that has only two possible values for each pixel. Typically the two colors used for a binary image are black and white though any two colors can be used. The color used for the object(s) in the image is the foreground color while the rest of the image is the background color.[1] In the document-scanning industry this is often referred to as "bi-tonal". Binarization is the process of converting grayscale image into binary image.

```
binarizeImage ( bufferedImage Image){

    threshold = Threshold_Calculator ( bufferedImage Image)
```

```
        for i=0 to image_width{
        for j=0 to image_height{
        color = Image.getRGB(x,y)
        red = color.getRed()
        blue = color.getBlue()
        green = color.getGreen()
        if ( red < threshold && blue < threshold && green < threshold)
        image[i][j] = 1
        else image[i][j] = 0
        }
        }
    }
```

4.1.3 Image Thinning or Skeletonization

Thinning is the process to remove outer points of an object shape in an image. The remaining points form skeleton of the object and so process is called skeletonization. The thinning process occasionally removes a large volume of data, and so improves the efficiency of sub-sequent tasks. These algorithms can be divided into two categories, namely, sequential and parallel algorithms. A sequential thinning algorithm performs the pixel removal work pixel by pixel in the image plane and the removal decision depends on the result obtained so far in the current iteration as well as of previous iteration. On the other hand parallel algorithm processes all the pixels of image simultaneously, and the removal of pixels depends on the result of the previous iterations. Mr. Christopher Holt proposed a parallel thinning algorithm [11].

```
holt_Thinning ( Fingerprint fingerprint){
    image[][] = fingerprint.getBinaryImage()
    while (no_points_deleted == false){
    no_points_deleted = true
    for c = 0 to 1{
    for y = 1 to fingerprint_height – 1{
```

```
    for x = 1 to fingerprint_width – 1{
    point1 = image[x][y]==1 ? true:false
    point2 = image[x][y-1]==1 ? true:false
    point3 = image[x+1][y-1]==1 ? true:false
    point4 = image[x+1][y]==1 ? true:false
    point5 = image[x+1][y+1]==1 ? true:false
    point6 = image[x][y+1]==1 ? true:false
    point7 = image[x-1][y+1]==1 ? true:false
    point8 = image[x-1][y]==1 ? true:false
    point9 = image[x-1][y-1]==1 ? true:false

    if(image[x][y]==1){
    if(!(!isEdge(x,y,image) || (points4 && points6 && (points2 || points8))){
    no_pixels_deleted = false
    points_to_remove.add(point(x,y))
    }
    }}}
    for i = 0 to points_to_remove.size(){
         point = points_to_remove.remove(i)
         image[p.x][p.y] = 0
    }
    //staircase Removal
for y = 1 to fingerprint_height – 1{
    for x = 1 to fingerprint_width – 1{
    point1 = image[x][y]==1 ? true:false
    point2 = image[x][y-1]==1 ? true:false
    point3 = image[x+1][y-1]==1 ? true:false
    point4 = image[x+1][y]==1 ? true:false
    point5 = image[x+1][y+1]==1 ? true:false
    point6 = image[x][y+1]==1 ? true:false
    point7 = image[x-1][y+1]==1 ? true:false
    point8 = image[x-1][y]==1 ? true:false
    point9 = image[x-1][y-1]==1 ? true:false
    if(image[x][y]==1){
    if(point2 &&(point4 && !point3 && !point7 && (!point8 || !point6)||
    (point8 && !point9 && !point5 && (!point4 || !point6)))){
    image[x][y]=0
    continue
    }
    }}
}
fingerprint.setSkeleton(image)
}
}
```

4.1.4 Cross Number method

Cross number method is used for feature extraction from the fingerprint image. In this a 3x3 window is taken and pixel in middle is checked for surrounding 1’s and 0’s by which it is decided whether it is a minutiae or not.

```
Fingerprint mapMinutiaes(Fingerprint fingerprint) {
outSkeleton = basicOperations.copy (fingerprint.getSkeleton())
img = FPManager.toImage(outSkeleton);
bfImg = new BufferedImage(width, height, Image.SCALE_SMOOTH);
g2d = bfImg.createGraphics();
 g2d.drawImage(img, 0, 0, null);
 g2d.dispose();

for i = (margin + 20) to (width - margin – 20) {
for j = margin to ( height – margin) {
patterns = BasicOperations.timesPattern (i, j, fingerprint.getSkeleton());
    if (fingerprint.getSkeleton()[i][j] == 1) {
        if (patterns == 1) {
        outSkeleton = drawRectangle(i, j, outSkeleton, 2);
          end_of_line++;
                }
                if (patterns == 3) {
                    outSkeleton = drawRectangle(i, j, outSkeleton, 3);
                    bifurcation++;
                }
            }
        }
    }

    fingerprint.setSkeleton(outSkeleton);
    fingerprint.setBifurcations(bif);
    fingerprint.setEndoflines(eol);
    return fingerprint;
}
```

4.1.5 Binder module

```
static public String Bind(String key, String rKey) {
        int l1 = key.length();
        int l2 = rKey.length();
        if (l2 < 19) {
            for (int i = 0; i < 19 - l2; i++) {
                rKey = rKey + "1";
                l2++;}}
             else if (l2 > 19) {
                         rKey = rKey.substring(0, 18); }
        String key1 = key.substring(0, l2);
        String key2 = key.substring(l2, (l2 * 2));
        long lkey = Long.parseLong(key1);
        long lfKey = Long.parseLong(rKey);
        lkey = lkey ^ lfKey;
        key1 = Objects.toString(lkey, null);
        lkey = Long.parseLong(key2);
        lkey = lkey ^ lfKey;
        key2 = Objects.toString(lkey, null);
        String key3 = key1 + key2;
        key = key3 + key.substring(40 - (l1 - (l2 * 2)));
        return key;
    }
```

4.1.6 Extractor module

```
static public String Extract(String key, String bKey){
        int l1 = key.length();
        int l2 = bKey.length();
        String key1 = key.substring(0, 19);
        String key2 = key.substring(19,38);
        long lkey =  Long.parseLong(key1);
        String bKey1 = bKey.substring(0, 19);
        String bKey2 = bKey.substring(19,38);
        long lfKey = Long.parseLong(bKey1);
        lkey = lkey ^ lfKey;
        key1 = Objects.toString(lkey, null);
        lkey =  Long.parseLong(key2);
        lfKey = Long.parseLong(bKey2);
        lkey = lkey ^ lfKey;
        key2 = Objects.toString(lkey, null);
        key  = key1 + key2 ;//+ key.substring(40-(l1-(l2*2)));
        if(key1.equals(key2)){
        return key1;}
        else
           return key; }
```

4.1.7 KeyGenerator module

```
static public String generate(){
        Random num = new Random();
        long random_num = num.nextLong();
        if(random_num<0){
        random_num=random_num*(-1);
        }
        String key = Objects.toString(random_num, null);
        return key;
    }
```

4.1.8 Database Module

- **HelperData**

  HelperData class is used to generate the HelperData table in the relational database with its data members as the attributes of table.

```
package Biometric.entity;
import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.Id;
@Entity
public class HelperData implements Serializable {
    @Id
    private String HashedKey;
    private String Name;
    private String MatchKey;
    public String getHashedKey() {
        return HashedKey;
    }
    public void setHashedKey(String HashedKey) {
        this.HashedKey = HashedKey;
    }
    public String getMatchKey() {
        return MatchKey;
    }
    public void setMatchKey(String MatchKey) {
        this.MatchKey = MatchKey;
    }
    public HelperData(){}
    public HelperData(String hash,String name, String key){
        this.HashedKey = hash;
        this.Name = name;
    }
    public String getName() {
        return Name;
    }
    public void setName(String Name) {
        this.Name = Name;
    }}
```

- **Create Database**

  This method creates the table from the HelperData class in the database.

```
public void createDB() {
        AnnotationConfiguration cfg = new AnnotationConfiguration();
        try {
            cfg.addAnnotatedClass(HelperData.class);
        } catch (Exception e) {
            System.out.println(e);
        }
        cfg.configure();
        SchemaExport se = new SchemaExport(cfg);
 se.create(true, true);//Run the schema creation script.
         JOptionPane.showMessageDialog(this,"Database created");
    }
```

4.1.9 Enrollment

```
//This code executes when the Enroll key on Enrollment form is pressed.
……………………………………………………………………………………..
Connection con = null;
Statement stmt;
ResultSet rs = null;
try {
Class.forName("com.mysql.jdbc.Driver");
 con  =  DriverManager.getConnection("jdbc:mysql://localhost:3306/test",  "root",
"password");
  stmt = con.createStatement();
   int count=0;
            if(jTextField2.getText().equals(null)||jTextField1.getText().equals(null))
            JOptionPane.showMessageDialog(this,"Enrollment failed");
            else{
count= stmt.executeUpdate("insert into helperdata (HashedKey,MatchKey, Name)
Values"+"(\""+jTextField2.getText()+"\","+"\"00\","+"\""+jTextField1.getText()+
"\")");
JOptionPane.showMessageDialog(this,"User Enrolled");}
        }
        catch (ClassNotFoundException e) {
            System.out.println("Could not load database driver:" + e.getMessage());
        } catch (SQLException e) {
            System.out.println("SQLException caught: " + e.getMessage());
        } finally {
            //Always close the database connection
            try {
                if (con != null) {
                    con.close();
                }
            } catch (SQLException e) {
```

```
System.out.println("Exception found"+e);            }
        }
……………….
……………….
```

4.1.10 Verification

```
// This code executes when Verify key is pressed in the Verification module.
………………………………………………..
        String rKey = "";
        String rKey1 = "";
        rKey = Extractor.Extract(jTextField1.getText(), jTextField2.getText());
        int len = rKey.length();
if(len>19){
rKey1 = rKey.substring(0, 19);
rKey = rKey.substring(19,len);
    System.out.println("rKey1 is: "+rKey);
    System.out.println("rKey is: "+rKey1);
}
else{}
        Connection con = null;
        Statement stmt;
        ResultSet rs = null;
        String match = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/test",       "root",
"password");
            stmt = con.createStatement();
            int count = 0;
            rs= stmt.executeQuery("Select * from helperdata where (name=\"" +
jTextField3.getText() + "\")");
            long num1 = Long.parseLong(rKey);
    while (rs.next()) {
        System.out.println(num1);
        String Hkey = rs.getString("HashedKey");
        long num = Long.parseLong(Hkey);
        System.out.println(num);
        num1 = num ^ num1;
        System.out.println(num1);
        for (int k = 0; k < rKey.length(); k++) {
            if (Hkey.charAt(k) == rKey.charAt(k)) {
                match = match + "1";
            } else {
                match = match + "0";
            }

        }
    }
    System.out.println("match is:"+match);
    count = match.length()-match.replace("1","").length();
    System.out.println("count is:"+count);
    String matchPercent = "Match Percentage is: ";
    matchPercent = matchPercent + (count*100)/match.length();
        JOptionPane.showMessageDialog(this, matchPercent);
} catch (ClassNotFoundException e) {
    System.out.println("Could not load database driver:" + e.getMessage());
} catch (SQLException e) {
    System.out.println("SQLException caught: " + e.getMessage());
} finally {
    //Always close the database connection
    try {
        if (con != null) {
            con.close();
```

```
                }
            } catch (SQLException e) {
System.out.println(“Exception found”+e);
            }
        }
……………………………
…………………………….
```

4.1.11 Biometric Encryption

Biometric encryption is process of encrypting biometric template with the random generated key. Fuzzy commitment algorithm is used for this.

```
String generate_key( random_key, template){
random_key  = random_key XOR template
}
```

4.1.12Biometric Decryption

Biometric decryption is process of decrypting biometric template with the stored key (helper data). Fuzzy commitment algorithm is used for this.

```
String retrieve_key( stored_key, template)
{
stored_key  = stored_key XOR template
}
```

4.1.13Barcode Encoder

Barcode encoder encodes the data into the 2D barcode. The barcode being used here is QRCode. For encoding the text, it is entered into the text field of form and on pressing encode button an image of QRCode is generated.

4.1.14 Barcode Decoder

Barcode decoder decodes the QRCode saved as image in the system. For decoding, press the decode button and select the area which has barcode. After decoding QRCode text will be displayed in text field of form.

4.2 Screenshots

4.2.1 Homepage

On starting the application Homepage will come up which has mainly four buttons, which links to new windows of relating functions.

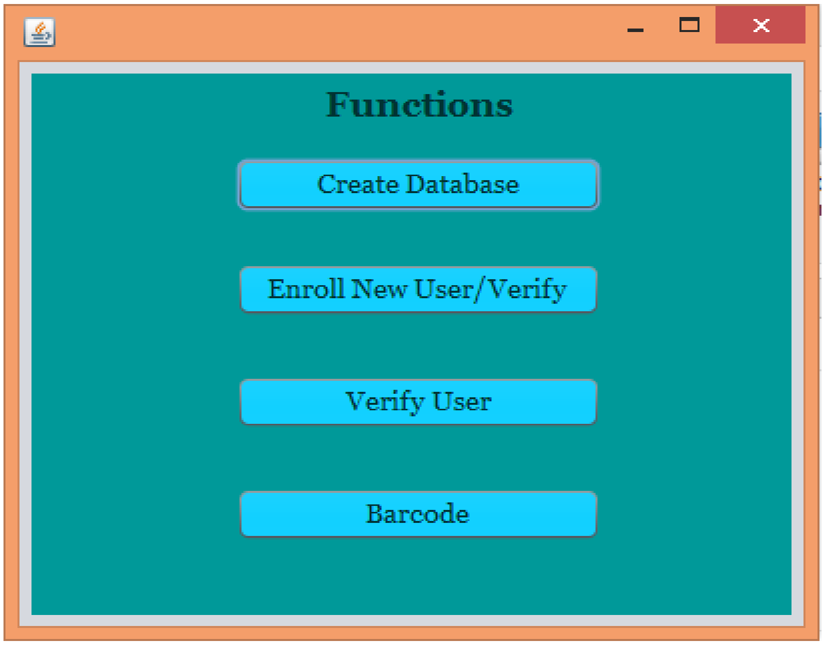


Figure 4.1 Homepage

4.2.2 Admin module

Database admin can create database directly by providing username and password. If database already exists, it will be overwritten by the create action. On completion of creation of database, a message will say database created.

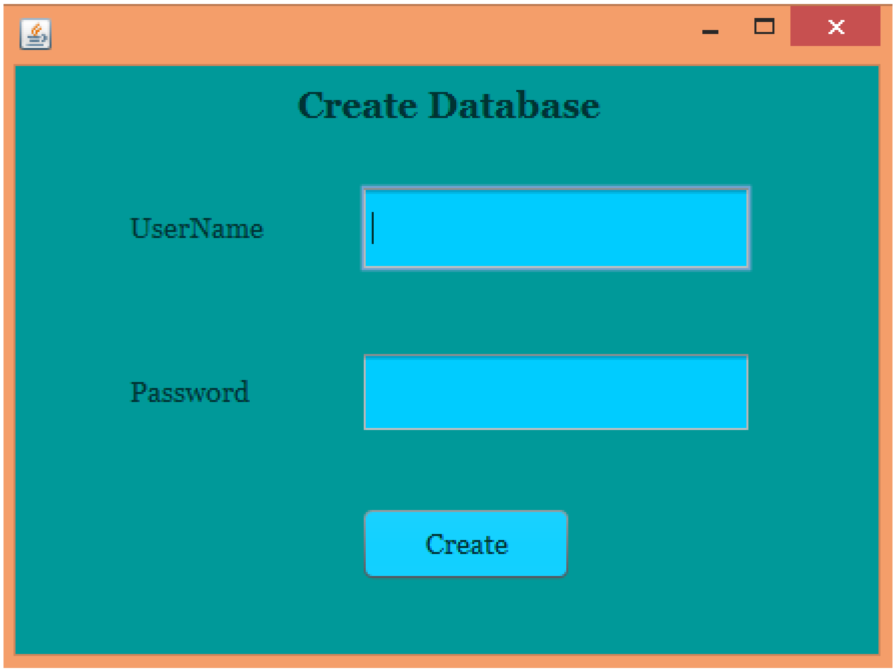
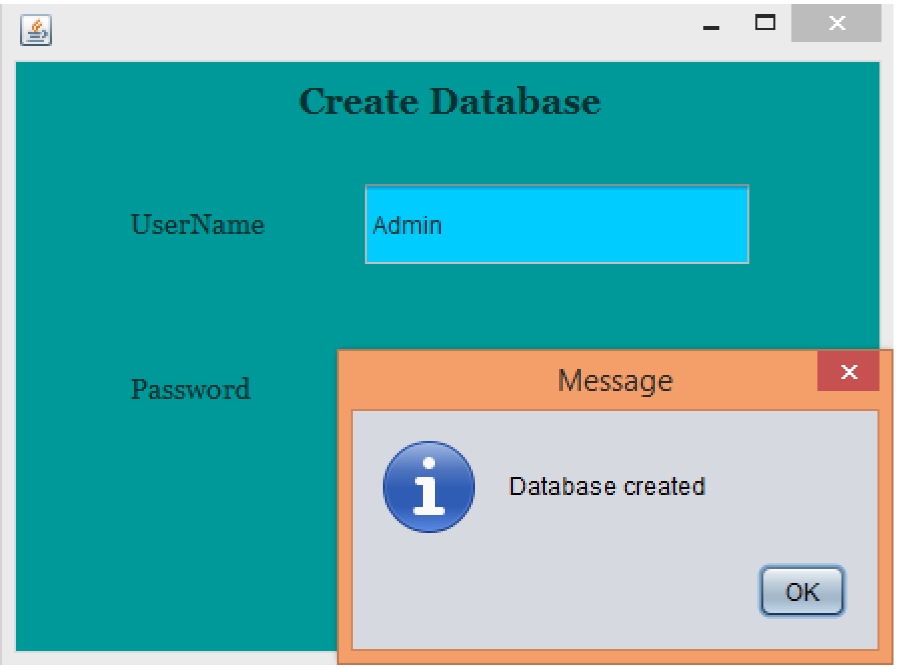


Figure 4.2 ***Create Database***



Admin

Database created

Figure 4.2 ***Create Database Message***

4.2.2 Fingerprint Module

On running the preprocessing module, a window for preprocessing of image comes up where an image can be selected using browse or directly pasting location in the textbox. The image chosen will be displayed on the darker green area. Second part contains four buttons for four function which has to be applied sequentially. Below buttons number of Bifurcation and EndOf Lines are shown which shows the number of detected minutiaes.

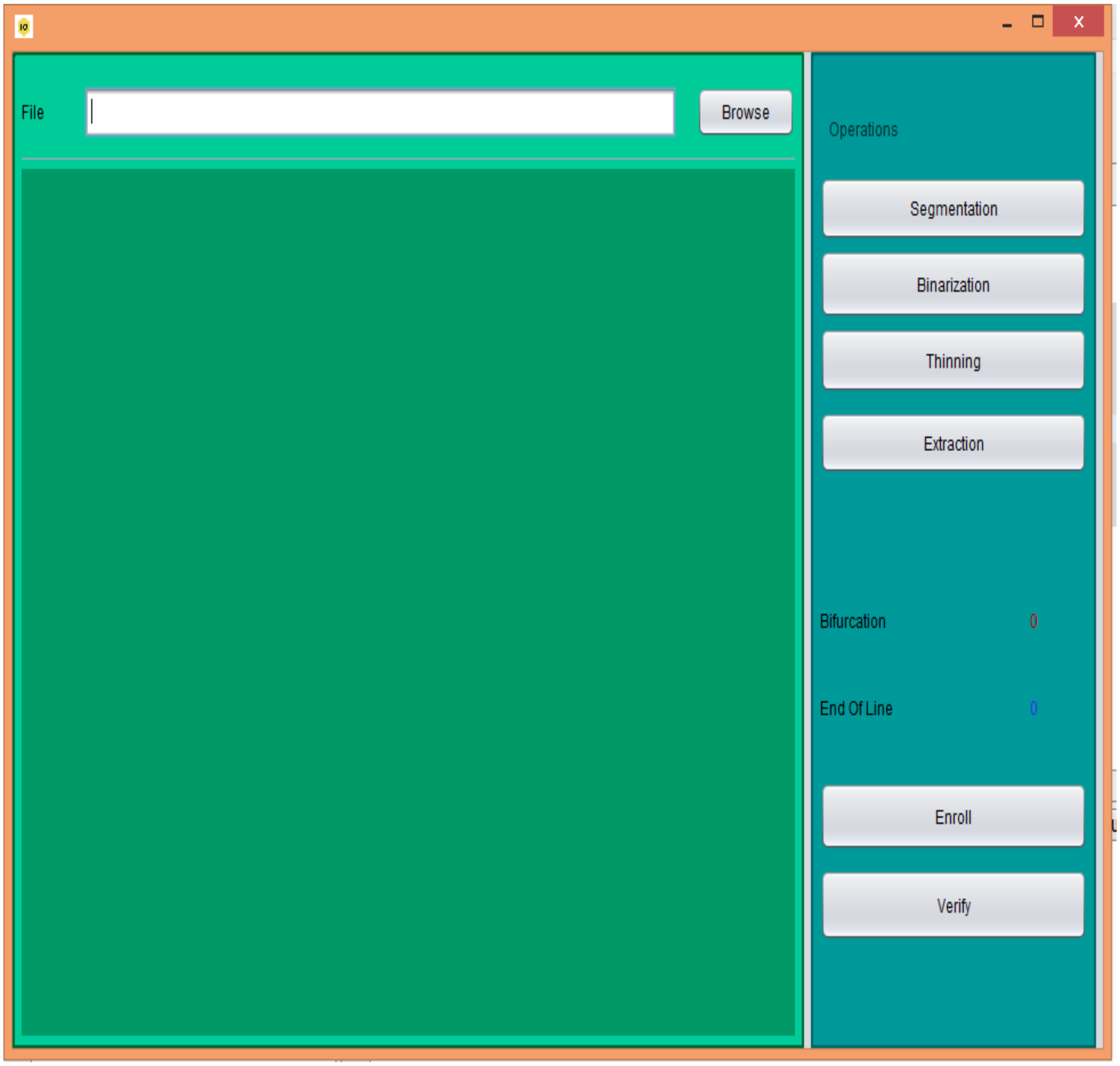
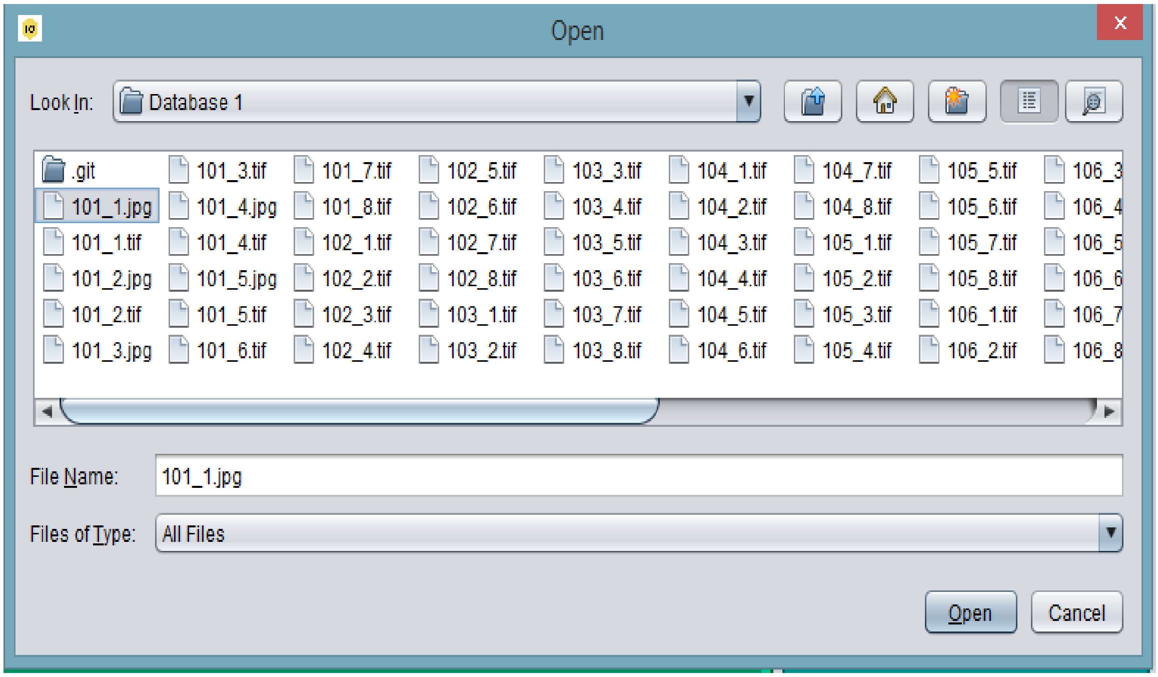


Figure 4.3 **Fingerprint Preprocessing**

On clicking the Browse button.

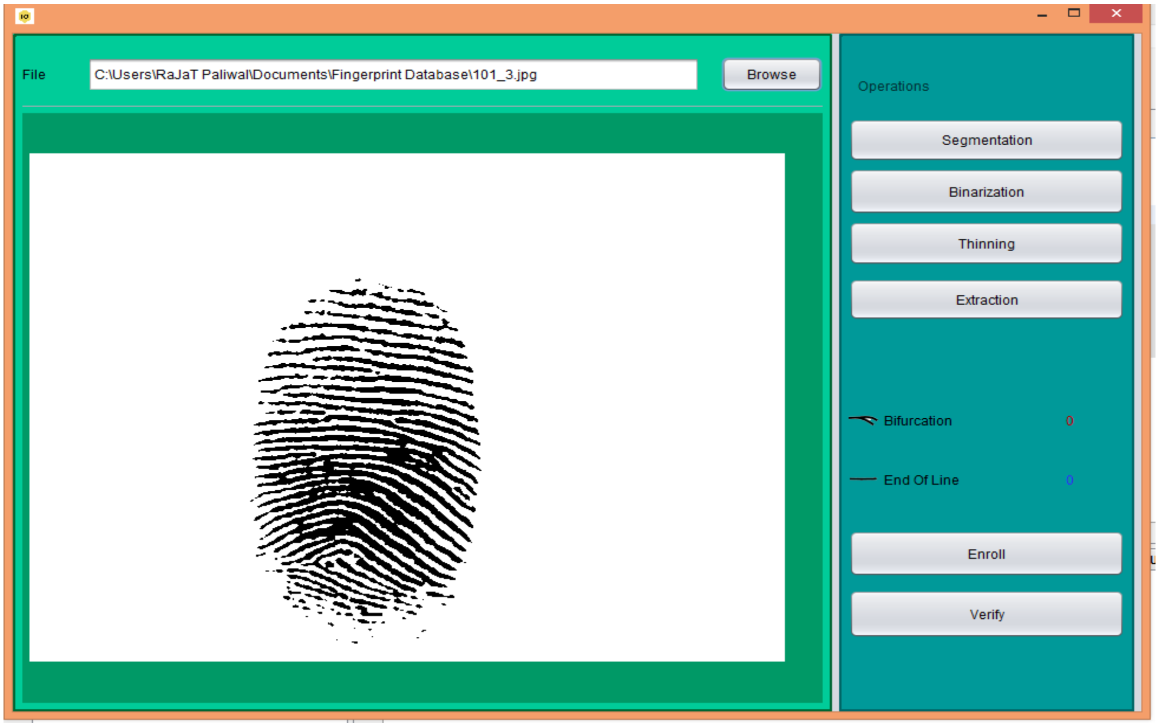


Look In: Database 1

File Name: 101_1.jpg

Files of Type: All Files

Figure 4.4 **Choose image from Database**

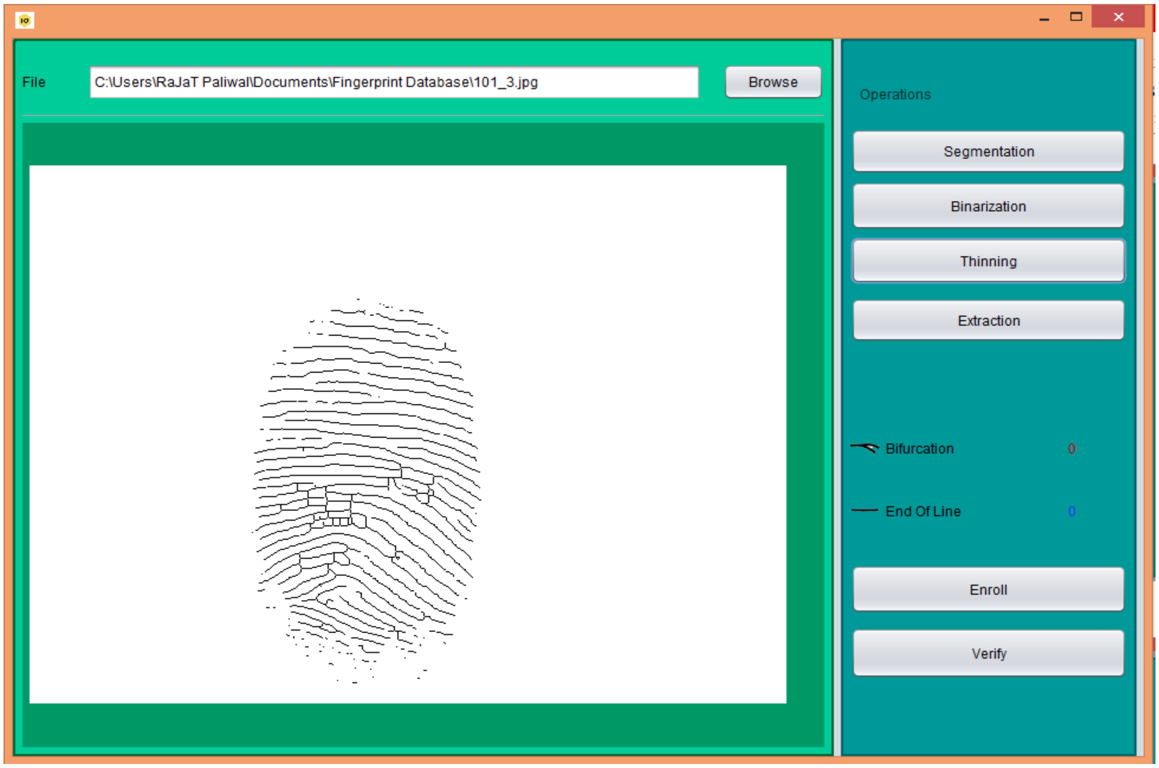


File C:\Users\RaJaT Paliwal\Documents\Fingerprint Database\101_3.jpg

Bifurcation 0

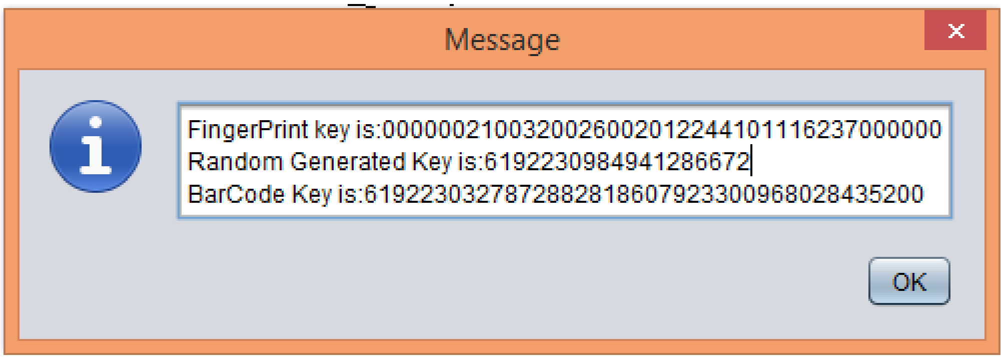
End Of Line 0

Figure 4.5 **After Binarization**



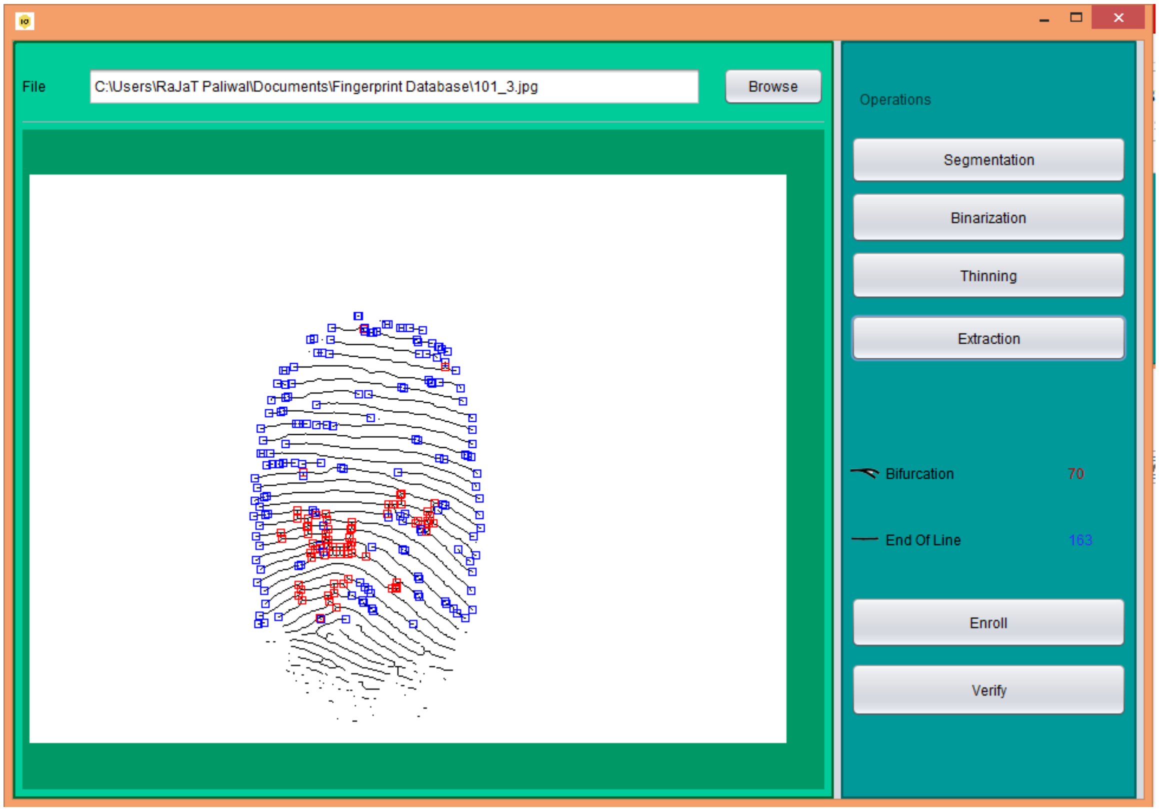
C:\Users\RaJaT Paliwal\Documents\Fingerprint Database\101_3.jpg

Figure 4.6 **After Thinning**



```
FingerPrint key is:00000021003200260020122441011162370000000
Random Generated Key is:6192230984941286672
BarCode Key is:6192230327872882818607923300968028435200
```

Figure 4.7 **On Extraction**



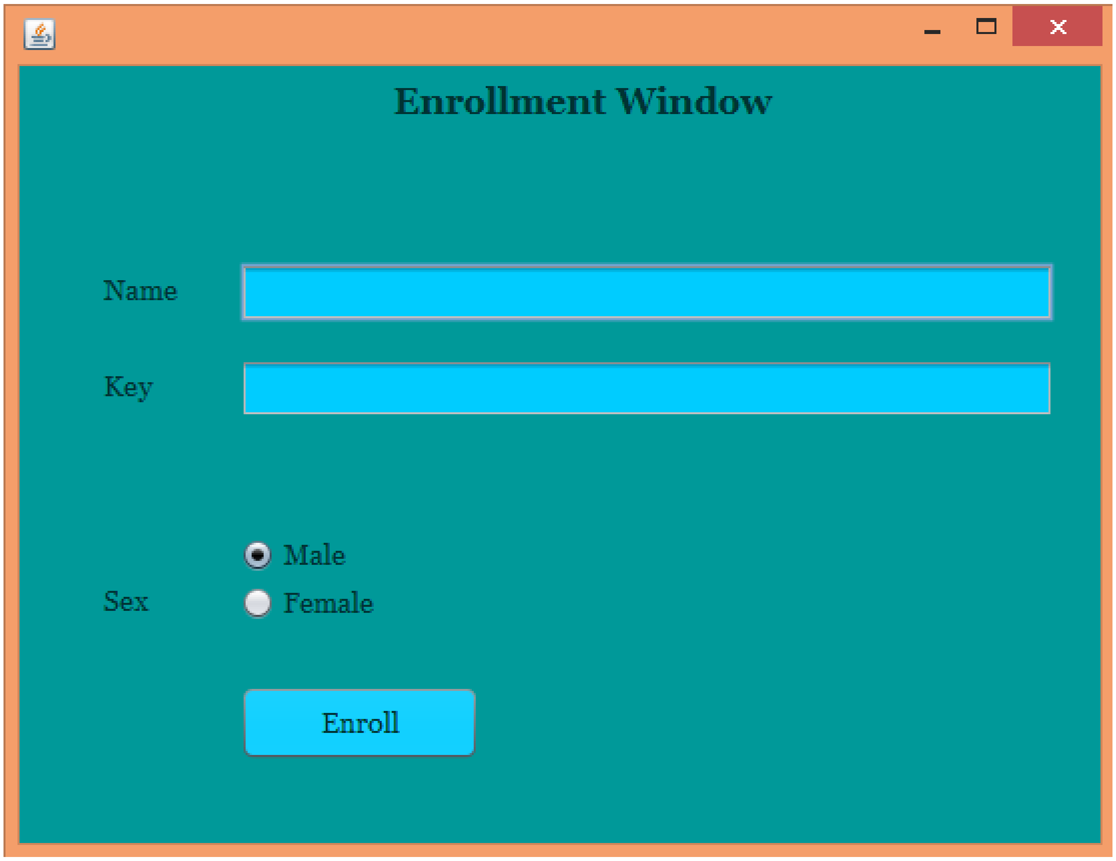
File C:\Users\RaJaT Paliwal\Documents\Fingerprint Database\101_3.jpg

Bifurcation 70

End Of Line 163

***Figure 4.8* After Extraction**

4.2.3 User Module



Sex ● Male ○ Female

Figure 4.9 **Enroll User**

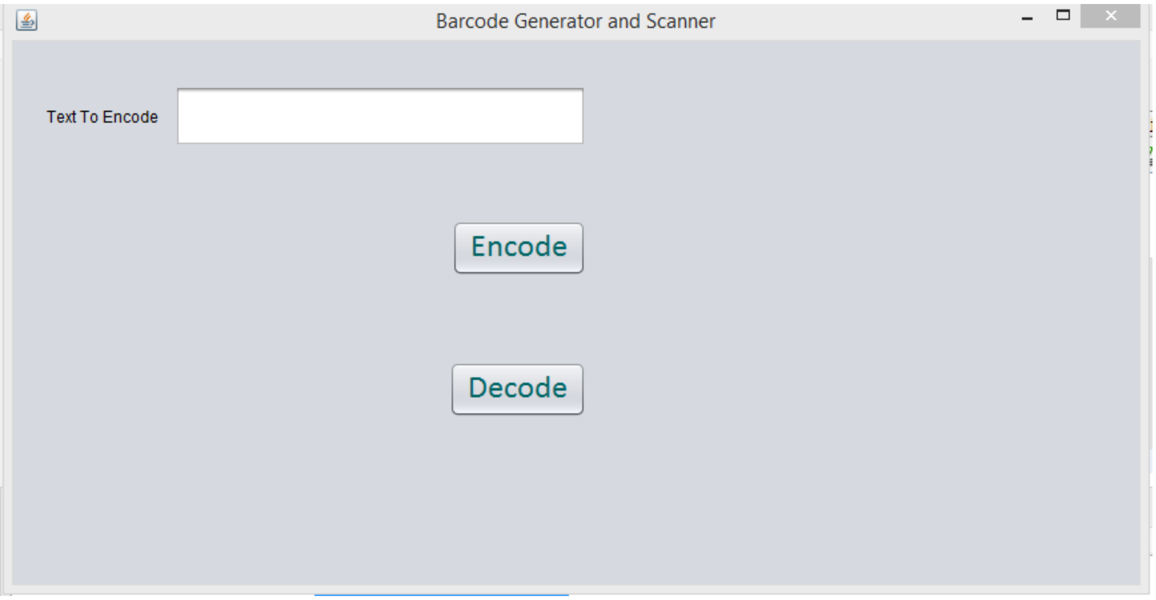
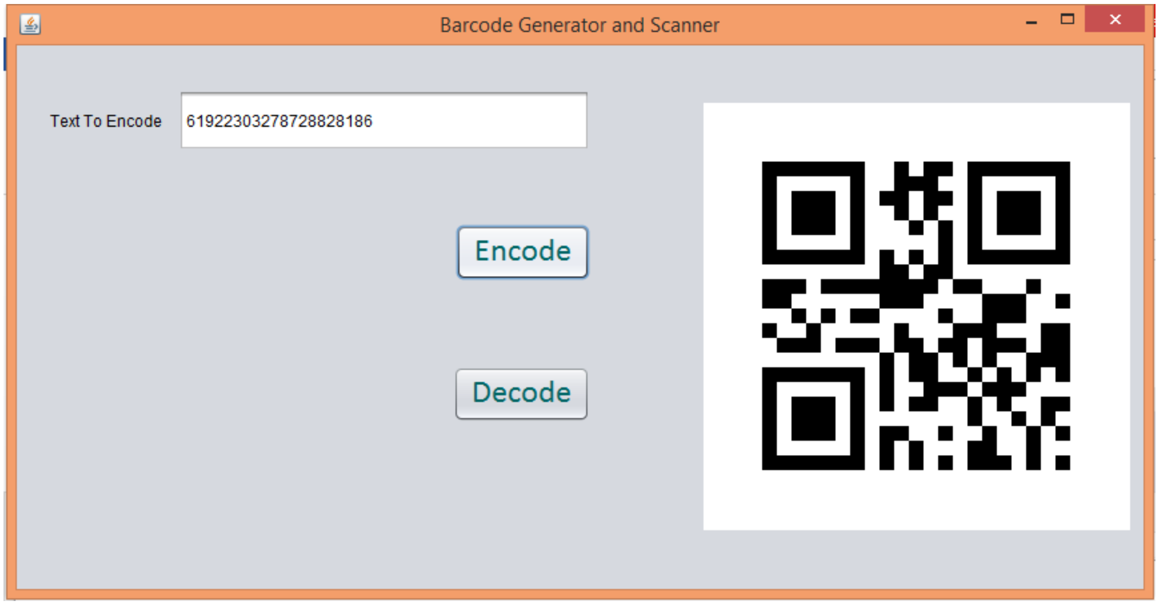


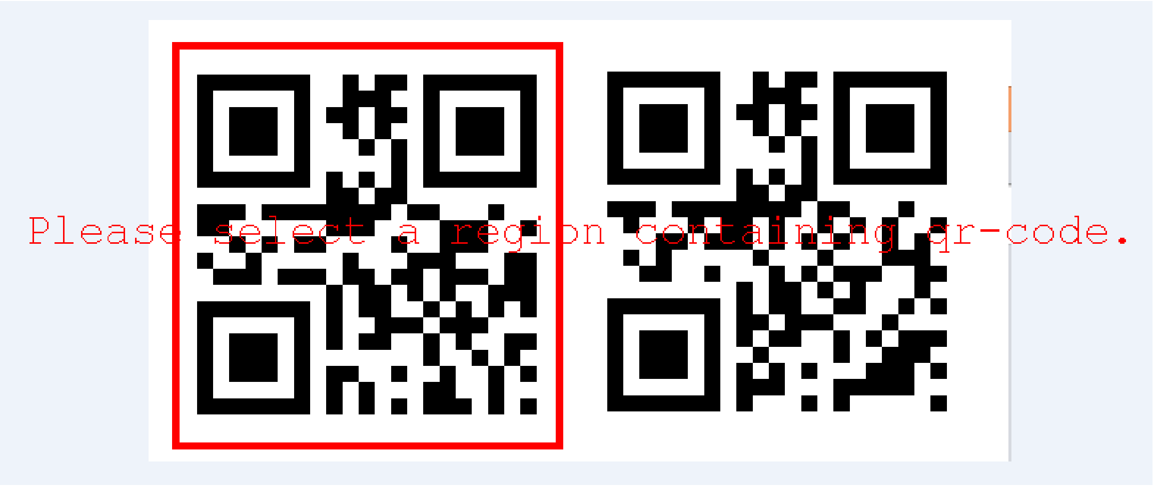
Figure 4.10 **Barcode Endode/Decode**



Text To Encode 61922303278728828186

Figure 4.11 **Encoding Key**

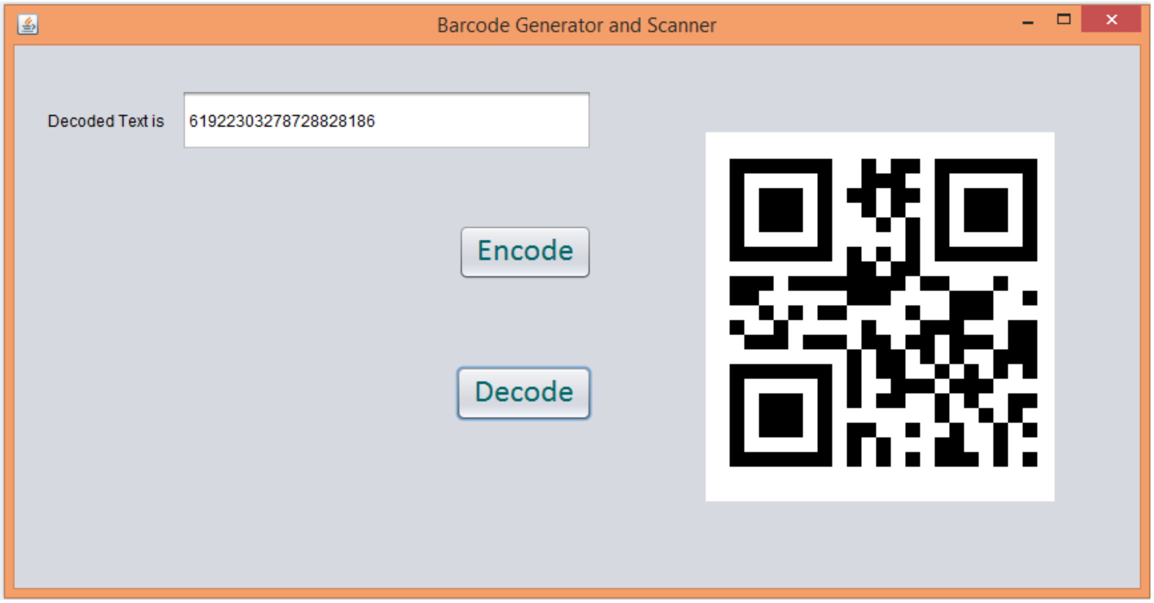
On Encoding the key a barcode is generated which can be saved and further used for getting the key for verification phase.



Please select a region containing qr-code.

Figure 4.12 **Decoding Barcode**

Both Barcodes on decoding provides keys which are combined to generate the Encrypted key.



Decoded Text is 61922303278728828186

Figure 4.13 After Decoding

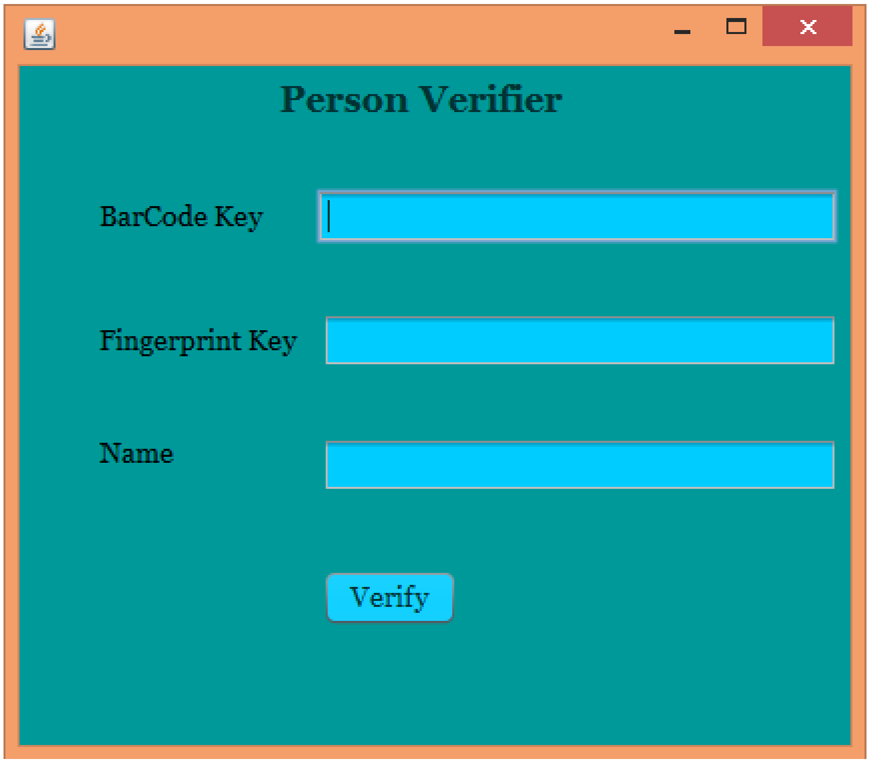
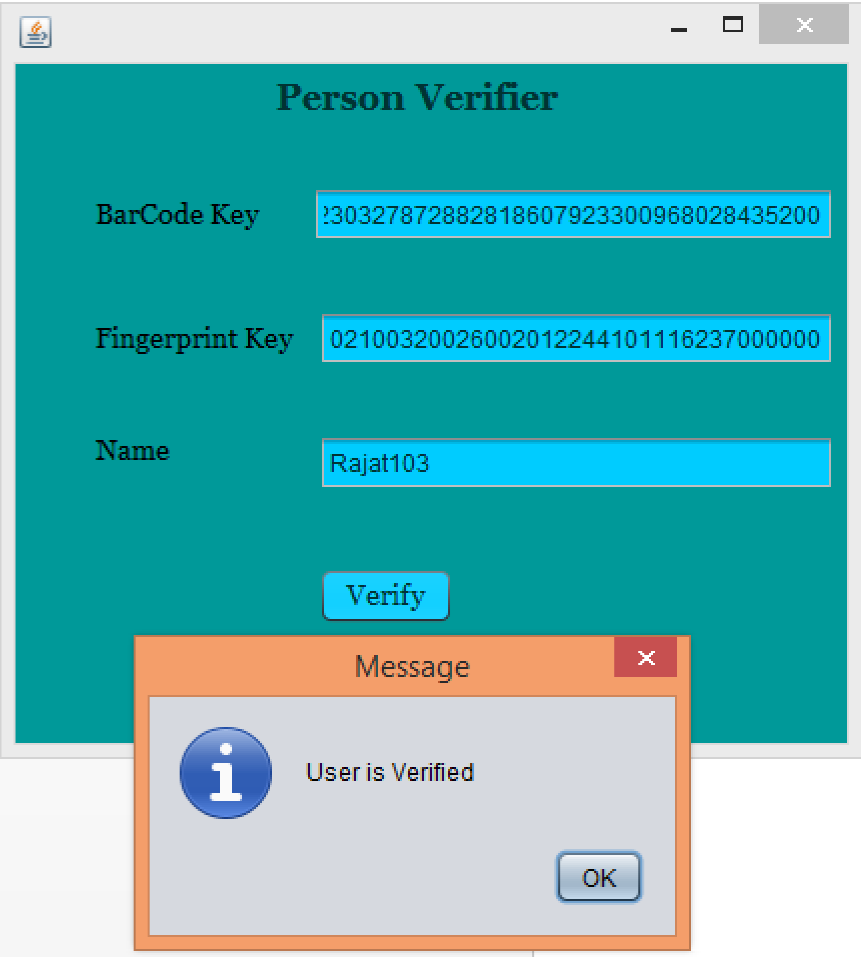


Figure 4.14 **Person Verifier**



Person Verifier

BarCode Key 230327872882818607923300968028435200

Fingerprint Key 02100320026002012244101116237000000

Name Rajat103

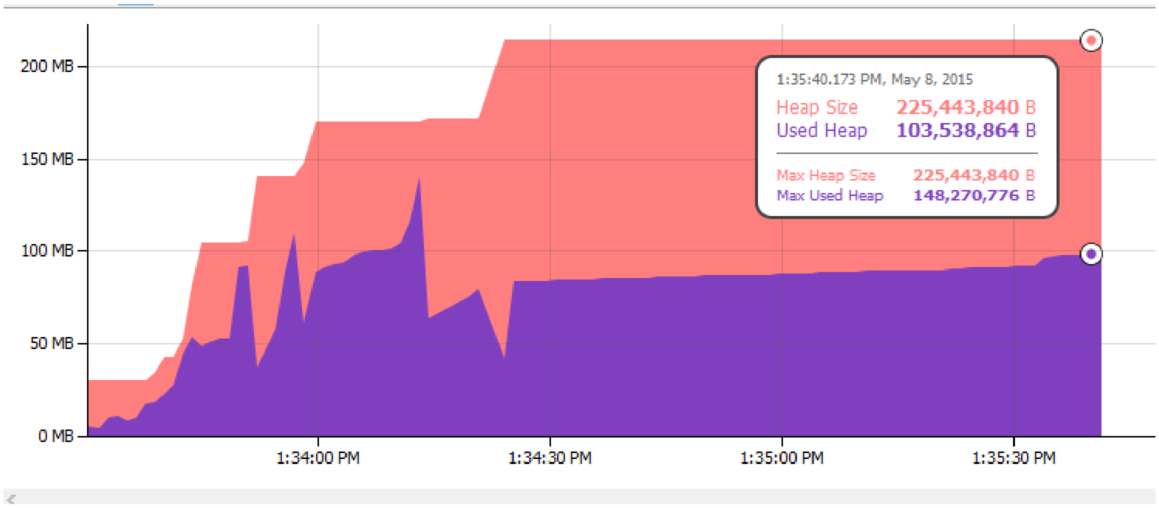
Message

User is Verified

Figure 4.15 **Verification of User**

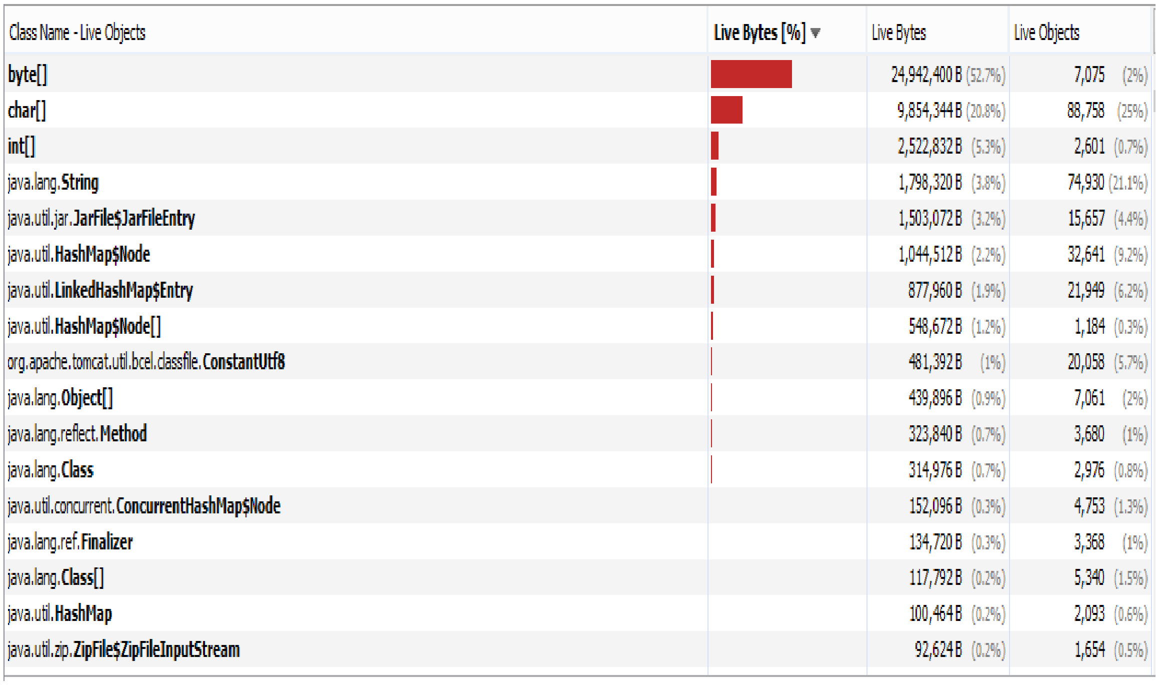
4.3 Memory Usage

Below are the memory requirement for running the application on a system.



1:35:40.173 PM, May 8, 2015
Heap Size 225,443,840 B
Used Heap 103,538,864 B
Max Heap Size 225,443,840 B
Max Used Heap 148,270,776 B

Figure 4.16 **Memory Requirement**



Class Name - Live Objects	Live Bytes [%] ▾	Live Bytes	Live Objects
byte[]		24,942,400 B (52.7%)	7,075 (2%)
char[]		9,854,344 B (20.8%)	88,758 (25%)
int[]		2,522,832 B (5.3%)	2,601 (0.7%)
java.lang.String		1,798,320 B (3.8%)	74,930 (21.1%)
java.util.jar.JarFile$JarFileEntry		1,503,072 B (3.2%)	15,657 (4.4%)
java.util.HashMap$Node		1,044,512 B (2.2%)	32,641 (9.2%)
java.util.LinkedHashMap$Entry		877,960 B (1.9%)	21,949 (6.2%)
java.util.HashMap$Node[]		548,672 B (1.2%)	1,184 (0.3%)
org.apache.tomcat.util.bcel.classfile.ConstantUtf8		481,392 B (1%)	20,058 (5.7%)
java.lang.Object[]		439,896 B (0.9%)	7,061 (2%)
java.lang.reflect.Method		323,840 B (0.7%)	3,680 (1%)
java.lang.Class		314,976 B (0.7%)	2,976 (0.8%)
java.util.concurrent.ConcurrentHashMap$Node		152,096 B (0.3%)	4,753 (1.3%)
java.lang.ref.Finalizer		134,720 B (0.3%)	3,368 (1%)
java.lang.Class[]		117,792 B (0.2%)	5,340 (1.5%)
java.util.HashMap		100,464 B (0.2%)	2,093 (0.6%)
java.util.zip.ZipFile$ZipFileInputStream		92,624 B (0.2%)	1,654 (0.5%)

Figure 4.17 Live Memory Space Snapshot

CHAPTER 5
Experimental Analysis

For the experimental purpose FVC 2004**[12]** database is used which is easily available for public domain. It consists of 100 different users and each user has 8 fingerprint images. We performed two sets of experiments, namely genuine test and impostor test. For the test on the genuine set, we compared the enroll fingerprint and the query fingerprint among the *same* user. For the test on imposters, the scores of imposter are generated by comparing the enroll fingerprint and the query fingerprint from the different users.

5.1 Security of Biometric

In the proposed system when we assume that the stored key is revealed, the adversary has to further reconstruct the hash vector. However, the key revealed is the encrypted and randomized version of the original data. To construct the biometric, it has to be decrypted with barcode, which is not available to adversary. Even if barcode is also revealed then the decrypted key has to be rearranged as hash vector. We have used 10 rectangles and key's minimum length is 40. It makes it computationally difficult to invert the original hash vector. In case the original hash vector has been reconstructed, the adversary will also have no clue to determine the exact location of each minutiae points since we just count the number of minutiae points in the rectangles based on their angles.

CHAPTER 5

Conclusion and Future work

Biometric Encryption is a fruitful area for research and is becoming sufficiently mature for prototype development and the consideration of applications. BE technologies exemplify the fundamental privacy and data protection principles that are endorsed around the world, such as data minimization, user empowerment, and security. Although introducing biometrics into information systems may result in considerable benefits, it can also introduce many new security and privacy vulnerabilities, risks, and concerns. Novel Biometric Encryption techniques can overcome many of those risks and vulnerabilities, resulting in a win-win, positive-sum model that presents distinct advantages to both security and privacy.

Biometric Encrypted Secure Storage (BESS) uses the best available algorithms for preprocessing of data, hence reducing the computation cost. It uses hibernate, hence making it easy to scale the database architecture w.r.t object oriented nature.

In the future blind authentication protocol will be implemented over this system to make it usable for client-server architecture. Also some other biometric encryption will be tested to work with the system and make it more secure than fuzzy commitment. A module for adding and deleting information in the database after authorization with biometric key and barcode will be added.

BIBLIOGRAPHY

[1] Cavoukian, Ann, and Alex Stoianov, "Biometric encryption chapter from the encyclopedia of biometrics." pp. 1-14, 2009.

[2] Jain, Anil K., Patrick Flynn, and Arun A. Ross, “Introduction to Biometrics”, “*Handbook of biometrics”*. Springer, pp. 1-23, 2007.

[3] Cavoukian, Ann, "Transformative Technologies Deliver Both Security and Privacy: Think Positive-Sum not Zero-Sum", *Information & Privacy Commissioner Ontario, Canada*, 2008.

[4] Jiawei Yuan; Shucheng Yu, "Efficient privacy-preserving biometric identification in cloud computing", *INFOCOM, 2013 Proceedings IEEE* , Vol., No., pp.2652-2660, 14-19 April 2013.

[5] Hogo, M.A., "Biometric keys based on pseudorandom sequences",*IEEE International Carnahan Conference on Security technology (ICCST),* Boston, MA, pp.110-118, October 2012.

[6] M. Upmanyu; A.M. Namboodiri; K. Srinathan; C.V. Jawahar, "Blind Authentication: A Secure Crypto-Biometric Verification Protocol", *IEEE Transactions on Information Forensics and Security,*Vol.5, No.2, pp.255-268, June 2010.

[7] R. Bakhteri; M. Khalil Hani, "Biometric encryption using fingerprint fuzzy vault for FPGA-based embedded systems", *TENCON 2009 - 2009 IEEE Region 10 Conference*, Singapore, pp.1,5, 23-26 Jan. 2009.

[8] A. Nagar; K. Nandakumar; A.K. Jain, "Securing fingerprint template: Fuzzy vault with minutiae descriptors", *19th International Conference on Pattern Recognition, 2008, ICPR 2008*,pp.1,4, 8-11 Dec. 2008.

[9] Zhe Jin, Andrew Beng Jin Teoh, Thion Song Ong, Connie Tee, "Secure minutiae-based fingerprint templates using random triangle hashing." *Visual Informatics: Bridging Research and Practice*. Springer Berlin Heidelberg, 2009. 521-531.

[10] Otsu, Nobuyuki. "A threshold selection method from gray-level histograms" *Automatica* 11. pp 285-296,1975.

[11] Holt, Christopher M.,"An improved parallel thinning algorithm" *Communications of the ACM* 30:2, pp.156-160, 1987.

[12] Third International Fingerprint Verification Competition (2004), http://bias.csr.unibo.it/fvc2004/